Proceedings of the 8th Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE 2023)

Magdalena Fuentes, Toni Heittola, Keisuke Imoto, Annamaria Mesaros,
Archontis Politis, Romain Serizel, Tuomas Virtanen (eds.)

September 20-22, 2023

DCASE 2023 WORKSHOP

This work is licensed under a Creative Commons Attribution 4.0 International License. To view a copy of this license, visit:

http://creativecommons.org/licenses/by/4.0/

Citation: Magdalena Fuentes, Toni Heittola, Keisuke Imoto, Annamaria Mesaros, Archontis Politis, Romain Serizel, Tuomas Virtanen (eds.), Proceedings of the 8th Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE 2023), Sep. 2023.

ISBN: 978-952-03-3171-9

Table of Content

Sound Event Classification with Object-Based Labels 1
James Afolaranmi, Irene Martín-Morató and Annamaria Mesaros

Learning in the Wild: Bioacoustics Few Shot Learning Without Using a Training Set 6
Víctor Aguado, Joan Navarro and Ester Vidaña-Vila

Multi-Resolution Conformer for Sound Event Detection: Analysis and Optimization 11
Sara Barahona, Diego de Benito-Gorron, Sergio Segovia, Daniel Ramos and Doroteo T. Toledano

Foley Sound Synthesis at the DCASE 2023 Challenge 16
Keunwoo Choi, Jaekwon Im, Laurie M. Heller, Brian McFee, Keisuke Imoto, Yuki Okamoto, Mathieu Lagrange and Shinnosuke Takamichi

STELIN-US: A Spatio-Temporally Linked Neighborhood Urban Sound Database 21
Snehit Chunarkar, Bo-Hao Su and Chi-Chun Lee

Foley Sound Synthesis Based on Generative Adversarial Networks Using Oneself-Conditioned Contrastive Learning 26
HaeChun Chung, Yuna Lee and JaeHoon Jung

Description and Discussion on DCASE 2023 Challenge Task 2: First-Shot Unsupervised Anomalous Sound Detection for Machine Condition Monitoring 31
Kota Dohi, Keisuke Imoto, Noboru Harada, Daisuke Niizumi, Yuma Koizumi, Tomoya Nishida, Harsh Purohit, Ryo Tanabe, Takashi Endo and Yohei Kawaguchi

Post-Processing Independent Evaluation of Sound Event Detection Systems 36
Janek Ebbers, Reinhold Haeb-Umbach and Romain Serizel

ToyADMOS2+: New Toyadmos Data and Benchmark Results of the First-Shot Anomalous Sound Event Detection Baseline 41
Noboru Harada, Daisuke Niizumi, Daiki Takeuchi, Yasunori Ohishi and Masahiro Yasuda

Evaluating Classification Systems Against Soft Labels with Fuzzy Precision and Recall 46
Manu Harju and Annamaria Mesaros

META-SELD: Meta-Learning for Fast Adaptation to the New Environment in Sound Event Localization and Detection 51
Jinbo Hu, Yin Cao, Ming Wu, Feiran Yang, Ziying Yu, Wenwu Wang, Mark D. Plumbley and Jun Yang

Leveraging Geometrical Acoustic Simulations of Spatial Room Impulse Responses for Improved Sound Event Detection and Localization 56
Christopher Ick and Brian McFee

Speech Obfuscation in Mel Spectra That Allows for Centralised Annotation and Classification of Sound Events 61
Michiel Jacobs, Lode Vuegen, Suraj Khan and Peter Karsmakers

FALL-E: A Foley Sound Synthesis Model and Strategies 66
Minsung Kang, Sangshin Oh, Hyeongi Moon, Kyungyun Lee and Ben Sangbae Chon

Label Filtering-Based Self-Learning for Sound Event Detection Using Frequency Dynamic Convolution with Large Kernel Attention 71
Ji Won Kim, Sang Won Son, Yoonah Song, Hong Kook Kim, Il Hoon Song and Jeong Eun Lim

Improving Automated Audio Captioning Fluency Through Data Augmentation and Ensemble Selection 76
Jaewon Kim, Yoon-Ah Park, Jae-Heung Cho and Joon-Hyuk Chang

Weakly-Supervised Automated Audio Captioning via Text Only Training 81
Theodoros Kouzelis and Vassilis Katsouros

Killing Two Birds with One Stone: Can an Audio Captioning System Also Be Used for Audio-Text Retrieval? 86
Étienne Labbé, Thomas Pellegrini and Julien Pinquier

Few Shot Bioacoustic Detection Boosting with Finetuning Strategy Using Negative-Based Prototypical Learning 91
Yuna Lee, HaeChun Chung and JaeHoon Jung

Masked Modeling Duo Vision Transformer with Multi-Layer Feature Fusion on Respiratory Sound Classification 96
Boxin Liu, Shiqi Zhang, Daiki Takeuchi, Noboru Harada and Shoji Makino

Efficient Evaluation Algorithms for Sound Event Detection 101
Vincent Lostanlen and Brian McFee

Aggregate or Separate: Learning From Multi-Annotator Noisy Labels for Best Classification Performance 106
Irene Martín-Morató, Paul Ahokas and Annamaria Mesaros

Active Learning in Sound-Based Bearing Fault Detection 111
Maarten Meire, Jeroen Zegers and Peter Karsmakers

Auditory Neural Response Inspired Sound Event Detection Based on Spectro-Temporal Receptive Field 116
Deokki Min, Hyeonuk Nam and Yong-Hwa Park

Creating a Good Teacher for Knowledge Distillation in Acoustic Scene Classification 121
Tobias Morocutti, Florian Schmid, Khaled Koutini and Gerhard Widmer

Pretraining Representations for Bioacoustic Few-Shot Detection Using Supervised Contrastive Learning 126
Ilyass Moummad, Romain Serizel and Nicolas Farrugia

Incremental Learning of Acoustic Scenes and Sound Events 131
Manjunath Mulimani and Annamaria Mesaros

Frequency & Channel Attention for Computationally Efficient Sound Event Detection 136
Hyeonuk Nam, Seong-Hu Kim, Deokki Min and Yong-Hwa Park

Unsupervised Domain Adaptation for the Cross-Dataset Detection of Humpback Whale Calls 141
Andrea Napoli and Paul R. White

Few-Shot Bioacoustic Event Detection at the DCASE 2023 Challenge 146
Ines Nolasco, Burooj Ghani, Shubhr Singh, Ester Vidaña-Vila, Helen Whitehead, Emily Grout, Michael Emmerson, Ivan Kiskin, Frants Jensen, Joe Morford, Ariana Strandburg-Peshkin, Lisa Gill, Hanna Pamuła, Vincent Lostanlen and Dan Stowell

Advancing Natural-Language Based Audio Retrieval with Passt and Large Audio-Caption Data Sets 151
Paul Primus, Khaled Koutini and Gerhard Widmer

Foley Sound Synthesis with a Class-Conditioned Latent Diffusion Model 156
Robin Scheibler, Takuya Hasumi, Yusuke Fujita, Tatsuya Komatsu, Ryuichi Yamamoto and Kentaro Tachibana

Distilling the Knowledge of Transformers and CNNs with CP-Mobile 161
Florian Schmid, Tobias Morocutti, Shahed Masoudian, Khaled Koutini and Gerhard Widmer

Device Generalization with Inverse Contrastive Loss and Impulse Response Augmentation 166
Lorenz P. Schmidt and Nils Peters

Multi-Label Open-Set Audio Classification 171
Sripathi Sridhar and Mark Cartwright

Spectral Transcoder : Using Pretrained Urban Sound Classifiers on Undersampled Spectral Representations 176
Modan Tailleur, Mathieu Lagrange, Pierre Aumond and Vincent Tourre

Audio Difference Captioning Utilizing Similarity-Discrepancy Disentanglement 181
Daiki Takeuchi, Yasunori Ohishi, Daisuke Niizumi, Noboru Harada and Kunio Kashino

Cross-Dimensional Interaction with Inverted Residual Triplet Attention for Low-Complexity Sound Event Detection 186
Khandelwal Tanmay and Rohan Kumar Das

Exploring Multi-Task Learning with Weighted Soft Label Loss for Sound Event Detection with Soft Labels 191
Khandelwal Tanmay and Rohan Kumar Das

Event Classification with Class-Level Gated Unit Using Large-Scale Pretrained Model for Optical Fiber Sensing 196
Noriyuki Tonami, Sakiko Mishima, Reishi Kondo, Keisuke Imoto and Tomoyuki Hino

Audio-Change Captioning to Explain Machine-Sound Anomalies 201
Shunsuke Tsubaki, Yohei Kawaguchi, Tomoya Nishida, Keisuke Imoto, Yuki Okamoto, Kota Dohi and Takashi Endo

Automatic Detection of Cow Vocalizations Using Convolutional Neural Networks 206
Ester Vidaña-Vila, Jordi Malé, Marc Freixes, Mireia Solís-Cifré, Miquel Jiménez, Cristian Larrondo, Raúl Guevara, Joana Miranda, Leticia Duboc, Eva Mainau, Pol Llonch and Rosa Ma

Alsina-Pagès

Low-Complexity Acoustic Scene Classification Using Deep Mutual Learning and Knowledge Distillation Fine-Tuning 211
Shilong Weng, Liu Yang, Binghong Xu and Xing Li

Two vs. Four-Channel Sound Event Localization and Detection 216
Julia Wilkins, Magdalena Fuentes, Luca Bondi, Shabnam Ghaffarzadegan, Ali Abavisani and Juan Pablo Bello

PLDISET: Probabilistic Localization and Detection of Independent Sound Events with Transformers 221
Peipei Wu, Jinzheng Zhao, Yaru Chen, Berghi Davide, Yi Yuan, Chenfei Zhu, Yin Cao, Yang Liu, Philip J.B. J Jackson, Mark D. Plumbley and Wenwu Wang

Crowdsourcing and Evaluating Text-Based Audio Retrieval Relevances 226
Huang Xie, Khazar Khorrami, Okko Räsänen and Tuomas Virtanen

Text-Driven Foley Sound Generation with Latent Diffusion Model 231
Yi Yuan, Haohe Liu, Xiyuan Kang, Peipei Wu, Mark D. Plumbley and Wenwu Wang

Preface

This volume gathers the papers presented at the Detection and Classification of Acoustic Scenes and Events 2023 Workshop (DCASE2023), Tampere, Finland, during 21–22 September 2023.

The DCASE 2023 Workshop was the eighth edition of the Workshop on Detection and Classification of Acoustic Scenes and Events, organized in conjunction with the DCASE Challenge. The aim of the workshop was to bring together researchers from many different universities and companies with interest in the topic, and provide the opportunity for scientific exchange of ideas and opinions.

The DCASE 2023 Workshop was jointly organized by researchers at Tampere University, Doshisha University, University of Lorraine, and New York University. The workshop received 68 submissions. Each paper was assigned to four reviewers, receiving at least three reviews. Of these, 47 papers were accepted for presentation.

The Organizing Committee was honored to have two keynote talks by leading experts in the field: Andrew Owens (The University of Michigan, Department of Electrical Engineering and Computer Science) and Björn Schuller (Imperial College London, Group on Language, Audio, & Music; University of Augsburg).

We wish to warmly thank our contributors for the success of the DCASE 2023. The workshop was the result of the hard work of many people, including all the authors and keynote speakers, the members of the Technical Program Committee, as well as all the organizers and participants of the DCASE Challenge tasks.

This edition of the workshop was supported by sponsorship from Google, Huawei, Reality Labs Research, Bose, Hitachi Ltd, Samsung, Adobe, Cochlear.ai, Gaudio, MARVEL Project H2020, and Mitsubishi Electric. We thank them for their valuable support to this workshop and the expanding topic area.

Annamaria Mesaros, Tuomas Virtanen, Keisuke Imoto,
Magdalena Fuentes, Romain Serizel, Archontis Politis, and Toni Heittola

Sponsors

Platinum sponsors

Google HUAWEI REALITY LABS RESEARCH

Gold sponsors

BOSE

Silver sponsors

HITACHI Inspire the Next SAMSUNG

Bronze sponsors

Adobe cochl. GAUDIO

MARVEL MITSUBISHI ELECTRIC Changes for the Better

SOUND EVENT CLASSIFICATION WITH OBJECT-BASED LABELS

James Afolaranmi, Irene Martín-Morató, Annamaria Mesaros

Computing Sciences, Tampere University, Tampere, FINLAND
james.afolaranmi@tuni.fi, irene.martinmorato@tuni.fi, annamaria.mesaros@tuni.fi

ABSTRACT

Availability of audio-visual datasets and increase of computational resources have made possible the use of deep learning techniques that exploit the relationship between audio and video. In this paper, we present an approach that makes use of pretrained models for object detection to label audio clips based on objects that are expected to make sound. The study consists of performing object detection for four target classes belonging to vehicle category and training sound classifiers in supervised way using the resulting labels. We conclude that object detection is a useful alternative for labeling audio-visual material for audio classification, with substantial improvements in different datasets. Results show that even for data provided with reference audio labels, labeling through video object detection can identify additional, non-annotated acoustic events, thus improving the quality of the labels in existing datasets. This promotes exploitation of video content not only as an alternative, but also to complement the available label information.

Index Terms— sound event classification, deep neural networks, object-based labels.

1. INTRODUCTION

Audio classification tasks have increased in popularity in recent years, due to applicability of methods for acoustic monitoring [1], environment monitoring [2], or emotion recognition [3] along with others. The analysis of acoustic scenes aims at recognizing different types of information in the environment, for example vehicles in urban scenes. The diversity of acoustic information in everyday environments increases the complexity of such task.

Deep learning methods allow obtaining high performance on classification tasks. Among the challenges that the traditional supervised learning scenario must overcome, one is data availability for training robust models. Supervised methods rely on the labeled data to train models effectively. These methods require accurate labelling, meaning that when incorrect labels are present in the training data, the learning process is compromised, leading to suboptimal performance of the model and reducing its generalization capabilities [4]. Proper data curation, label verification, and quality control mechanisms are essential to mitigate the impact of incorrect labels and ensure the training of robust models.

The release of AudioSet [5] has been a milestone for the audio datasets. It contains 527 sound classes from over 5000 hours of audio recordings collected from YouTube videos, provided as annotations for 10-second long clips. Following AudioSet, more datasets providing video and audio modalities have been published, e.g. MAVD-traffic [6] and TAU Urban Audio-Visual Scenes 2021 [7] datasets. While these are the main audio-visual datasets used in audio research, there is a much larger number of such datasets that are used in image/video research. Audio-visual datasets provide a rich source of information that combines auditory and visual modalities, offering valuable insights into the correlation and complementarity between audio and visual cues.

The task of annotating sound events within such datasets is time-consuming and expensive. As a result, the majority of audio-visual datasets are primarily annotated for visual content; some datasets have information on acoustic scene, while the annotation of sound events remains limited to a smaller subset. For example in EPIC-SOUNDS [8], the authors collected a large scale dataset of audio annotations as an extension of the original EPIC-KITCHEN dataset [9], which is originally aimed at computer vision research.

In this work we propose to investigate if labels derived through object detection methods based on the video modality are suitable for audio classification. Generally, the information in the audio and visual modalities is highly correlated, and sound-producing objects may be visible in the video, even though this is not guaranteed, for example in poor light conditions or in the presence of obstructions. We investigate how well YOLO (You Only Look Once) object detector [10] can be used to provide labels for audio content to ultimately train an audio classification model. Experiments performed on three different datasets show that even though the labels inferred based on objects are not fully corresponding to the audio ground truth, they provide a sufficient supervision signal for training a sound event classification system.

The rest of the paper is organized as follows: Section 2 introduces the approach used for obtaining the object-based labels and how they are used for audio classification purposes; Section 3 presents the datasets used in the experiments and introduces the classification system; it also includes an analysis of the results and discusses the comparison of the object-based labels with the reference audio labels; finally, Section 4 presents the conclusions and future work.

2. OBJECT-BASED AUDIO CLASSIFICATION

Figure 1 illustrates the workflow followed in this study. To obtain labels for the audio content, object detection using YOLO [10] is performed on video frames from the video clip. The pretrained model OpenL3 [11] is used to perform feature extraction and to obtain the embeddings for the corresponding aucio clip. The labels and the embeddings are used as input to the audio classification model. The target labels are the labels obtained from the object-detection model, and the input data are the embeddings from the pretrained OpenL3 model. The acoustic model is then trained using this information for classifying the selected target sounds.

This work was supported by Academy of Finland grant 332063 “Teaching machines to listen”. The authors wish to thank CSC-IT Centre of Science Ltd., Finland, for providing computational resources.

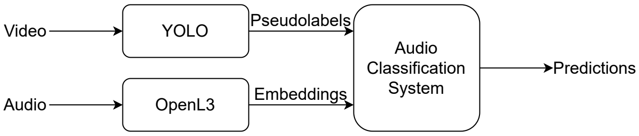


Figure 1: Proposed approach; Object detection is applied to frames of the video, and the resulting output labels are used in training the audio classification system.

2.1. Object detection framework

Object detection is a popular research task in computer vision. It involves localization of target objects into bounding boxes and classification of those objects. Object detectors can be classified into two categories: single-stage or two-stage object detectors, depending on the method used to locate and classify objects. YOLO falls under the category of single-stage detector that carries out object localization and classification in the same run [12].

YOLO architecture is based on multiple CNN layers followed by fully connected layers. It predicts bounding boxes and class probabilities simultaneously, making it efficient for real-time detection. In [10], the authors showed that YOLO was able to score 57.9% mean average precision on the PASCAL VOC 2012 test set on 20 labelled classes, and generalized better than other detectors when tested for person detection in two artwork datasets. In our study, we used a pretrained version of YOLOV5[13] capable of recognizing 80 classes to perform detection of four vehicle-related classes within the video data.

2.2. Audio classification framework

Because the scope of this work is to investigate feasibility of labeling audio through video, for the audio classification model employed in our study uses an existing architectures rather than designing and optimizing one for the task. We use embeddings from the pretrained L3-Net [14] implemented in OpenL3 [11] as a backbone, and three dense linear layers of 512, 128 and 4 neurons stacked upon each other; the network uses ReLU as an activation function for the first two dense layers and sigmoid activation function for the output layer to perform multi-label classification.

2.3. Datasets and baseline system

We use three different audio-visual datasets, namely: AudioSet [5], the MAVD dataset in Urban environments [15] and a subset of TAU Urban Audio-Visual Scenes 2021 Development Dataset [7]. In this work we use four target sound classes: *Bus*, *Car*, *Motorcycle* and *Truck*, which can all be found in these three datasets.

From AudioSet, a subset of 121.8 hours of data was selected based on the target classes. A 70/30 ratio is used to partition this subset into training and test set. The labels provided in AudioSet are used as ground truth in our comparative experiments. As documented in [16], some clips in AudioSet may have incorrect or missing labels. This is due to the annotation process which included a verification step for the candidate labels [5]; in this process the labels were manually verified, but no new labels were added. In general, AudioSet has a highly imbalanced class distribution which is prominent also in the subset used in our experiments, with the majority of the data examples belonging to the *Car* class.

Dataset	Training (hh:mm)	Test (hh:mm)
AudioSet	85:15	36:32
MAVD	01:03	00:27
TAU UrbanASC	02:00	01:30

Table 1: Amount of data available per dataset.

As a second source of annotated audio-visual data we use the Urbansas dataset [17], which consists of 3 hours of manually annotated data, compiled from two different datasets: MAVD [6] and TAU Urban Audio-Visual Scenes 2021 dataset (TAU UrbanASC) [7]. MAVD is an audio-visual dataset created to monitor urban noise in Montevideo, Uruguay, and consists of 1.5 hours of manually annotated data divided into train and test set. The TAU Urban Audio-Visual Scenes 2021 dataset (TAU UrbanASC) [7] consists of synchronized audio and video segments with a length of 10 seconds recorded in 12 different European cities. Of these, 1.5 h of the *street traffic* clips was annotated within the Urbansas dataset. We treat MAVD and TAU UrbanASC separately in our experiments. The total amount of data available in the training and test subsets used in our classification experiments is presented in Table 1.

We perform the classification experiments using the object-based labels and, for comparison, the reference audio labels, when available. Since audio reference labels are only available for two of the three datasets, AudioSet and MAVD, the comparative experiment is performed only for these two datasets.

3. EXPERIMENTAL RESULTS

We performed object detection on five image frames of the video clip (one frame every two seconds) using the pretrained YOLO model. To extract frames from the video we used the OpenCV library [1] in Python. For each of these five frames, YOLO returns labels corresponding to the four target classes, and coordinates for the bounding box of each object. The predicted labels include multiple instances of different classes for each 10 s video clip. To avoid losing any information about the detected objects, we create the set of labels inferred based on the video as the union set of the predicted object labels. The audio clip is then assigned the resulting set of labels for training a model as a multilabel classifier.

3.1. Comparison of inferred labels with audio reference labels

First of all, we verify to what extent the object-based inferred labels match the audio reference labels. To this end, we compare the obtained labels with the reference labels for all the data in each dataset (including training and test set, when available). The results are presented in Table 2. The object-based labels are most similar with the reference labels for the *Car* class in all three datasets, having a significantly higher F-score than any other class. We also observe that the *Truck* class has very low precision values for all three datasets. The discrepancy between the object-based and the audio reference labels is quite large for many cases. For example in the case of AudioSet the *Bus* class the precision is 0.32, meaning that only one third of clips labeled as *Bus* by YOLO are also annotated based on audio as containing the sound.

---

[1] https://github.com/opencv/opencv

	AudioSet			MAVD			TAU UrbanASC		
Class	P	R	F	P	R	F	P	R	F
Bus	0.32	0.73	0.45	0.86	0.55	0.67	0.43	0.90	0.59
Car	0.72	0.89	0.79	0.67	0.97	0.79	0.65	0.99	0.78
Motorcycle	0.46	0.90	0.61	0.50	0.35	0.41	0.71	0.42	0.53
Truck	0.36	0.87	0.51	0.12	0.80	0.21	0.13	0.84	0.23
Average	0.47	0.85	0.59	0.54	0.67	0.52	0.48	0.79	0.53

Table 2: Comparison of the object-based labels and reference audio labels for the three datasets.

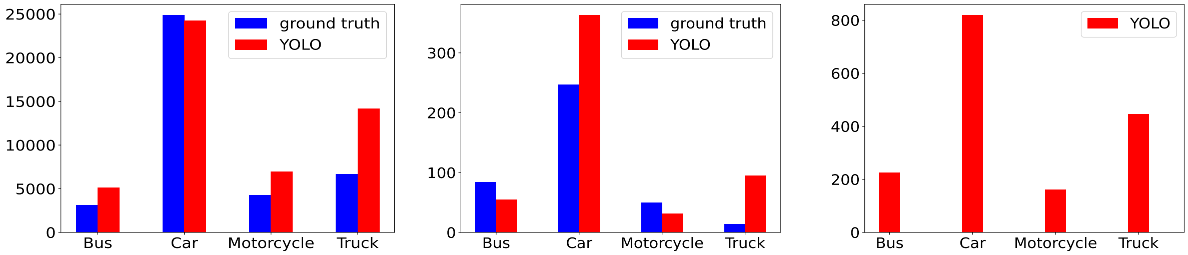


Figure 2: Training instances available for AudioSet, MAVD and TAU UrbanASC using the reference audio labels (blue) and object-based labels (red). For TAU UrbanASC, only the testing set is annotated, therefore we have no reference for comparison.

	AudioSet						MAVD						TAU UrbanASC		
	Reference labels			object-based			Reference labels			object-based			object-based		
Class	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F
Bus	0.00	0.00	0.00	0.18	0.08	0.11	0.73	0.46	**0.57**	0.82	0.22	0.35	0.23	0.27	0.25
Car	**0.73**	0.79	0.76	0.67	**0.96**	**0.79**	0.81	0.87	**0.84**	0.67	1.00	0.80	0.63	0.99	0.77
Motorcycle	0.51	0.32	0.39	0.54	0.43	**0.48**	0.50	0.32	**0.39**	0.67	0.09	0.16	0.00	0.00	0.00
Truck	0.00	0.00	0.00	0.28	0.68	**0.40**	0.00	0.00	0.00	0.00	0.00	0.00	0.07	0.34	0.11
Average	0.31	0.28	0.29	0.42	0.54	0.45	0.51	0.41	0.45	0.54	0.33	0.33	0.23	0.40	0.28

Table 3: Classification results for the three datasets, for the classifier trained with the reference labels and with the object-based labels.

Figure 2 illustrates the number of instances per class available for training the audio classification system for each case. It can be clearly seen that the *Car* class is the one with highest number of example instances for both label sets (reference and object-based labels) and all three datasets, while *Truck* has considerably less instances in the reference labels compared to the object-based labels.

3.2. Audio classification with object-based labels

We train a classifier using YOLO object-based labels. For comparison, we also train the same classifier structure using the audio reference labels. These models are then tested on the same test set and their performances are compared in terms of precision, recall, and F-score. The results are presented in Table 3.

For AudioSet, we observe that the classifier trained with object-based labels obtains a significantly higher recall for all the classes, and a higher F-score, despite the classifier output being evaluated against the reference annotations of the dataset itself. The system trained with object-based labels is able to recognize a higher number of event instances in the test data than if trained with the official reference labels provided in the dataset. However, this does not happen for the MAVD dataset. We hypothesize that the annotation process of MAVD was more efficient, and the quality of its reference labels is high. As seen from the results in Table 2, YOLO produces a lot of false positives, which in the case of MAVD are detrimental to the training process and consequently to the classification performance. Only for the *Car* class the classification performance is similar between the two training scenarios, but while recall for the model trained with the object-based labels reaches 100%, its precision suffers due to false positives. For the TAU UrbanASC we do not have audio reference labels, therefore we can only analyze the training with the object-based labels. The results in Table 3 show a high recall value for the *Car* class, which seems to be the dominant class among all datasets and label sets. At the same time. the system does not classify correctly any *Motorcycle* instances, which is the least represented class in the training data. Overall, the results on TAU UrbanASC are similar to those on MAVD.

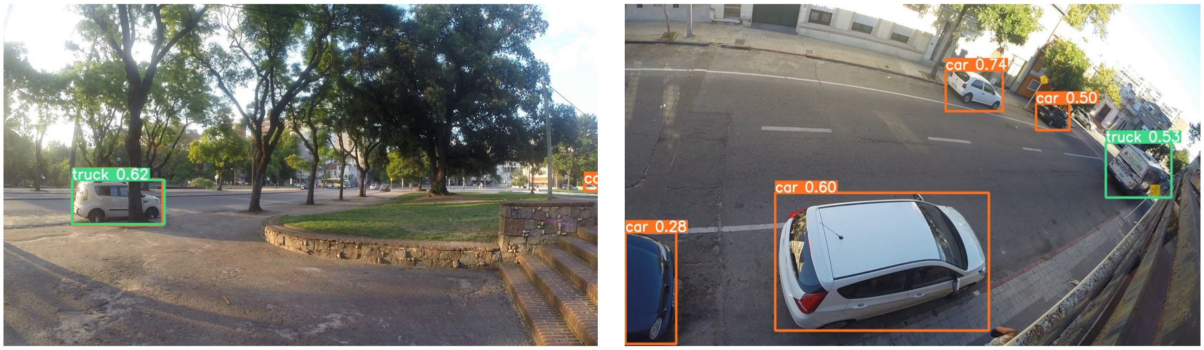


Figure 3: Example of mislabelled vehicles by YOLO.

3.3. Discussion

The *Truck* class is a very difficult case for all datasets, even though it is somewhat detected in AudioSet and TAU UrbanASC by the system trained with the object-based labels. In particular, the performance in AudioSet is very high, considering that the system trained with reference labels does not find any instance of this class. To understand this significant improvement in performance for AudioSet, we checked the clips for which YOLO indicated label *Truck* but the audio reference label did not contain it. We listened to 50 randomly selected clips and observed that 14% of them indeed contain truck sounds. In these cases, YOLO indicated a correct sound label based on the image, which were missing labels in the audio reference. There were also many false positives which add noise to the training process; nevertheless, the overall effect on the system performance was positive.

We visually inspected also the MAVD dataset *Truck* class, to understand the difference between the datasets. Looking at the predictions from the object detector, we observed that different types of vehicles (cars and buses) were mislabeled as trucks, which creates confusion between the categories. Two such examples are shown in Figure 3. In addition, in MAVD there were many scenes with parked vehicles which were visible and detected by the object detector, but did not produce any sound, therefore creating misleading information for the audio classifier during training.

This investigation revealed a very obvious drawback of using this method - objects in the image that do not produce sound (in this case parked vehicles) appear as false positives for the audio modality, and may be detrimental to performance. However, even with all these drawbacks and possible failure scenarios, the approach was shown to produce reasonable labels and in some cases lead to performance improvements. While this does not solve the problem of labeling audio content in audio-visual datasets, it can serve as a tool in more advanced training approaches; for example the object-based labels can be used as suggestions for methods that use active learning, or with a human-in-the-loop for verification; or can be treated as labels with some level of uncertainty to complement data that has been manually labeled by human annotators.

4. CONCLUSIONS

This work presented a novel approach of labelling audio data utilizing video information, to investigate the suitability of the method for creating reference labels for audio. The obtained labels were used afterwards in audio classification task. The method is based on an object detector model that takes as input a few frames of video corresponding to the audio clip, and predicting the target classes. Experiments performed on three different datasets showed the feasibility of using the audio-visual connection in the data to label audio content. However, the approach is unsuitable for situations when the sound sources are obscured/absent in the video frames, as they are not found by the object detector. In addition, some target sounding objects in the scene may actually not produce a sound in specific instances, leading to false positive labels. Despite these drawbacks, the method proves to be faster and lower-cost compared to the traditional annotation methods. Results from the experiment show that the method may outperform models trained with the provided reference audio labels, if they contain noisy or possibly incorrect information. We conclude that object-based labeling provides a suitable supervision signal for training and may be a useful tool in learning about audio content if handled as complementary information or to reinforce existing information about the data. Future work will focus on exploring more datasets for including a larger number of classes, and approaches for alleviating the effect of errors introduced by the object-based detector.

5. REFERENCES

[1] M. Ohlenbusch, A. Ahrens, C. Rollwage, and J. Bitzer, “Robust drone detection for acoustic monitoring applications,” in *2020 28th European Signal Processing Conference (EUSIPCO)*, 2021, pp. 6–10.

[2] E. B. Çoban, A. R. Syed, D. Pir, and M. I. Mandel, “Towards large scale ecoacoustic monitoring with small amounts of labeled data,” in *2021 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2021, pp. 181–185.

[3] Y. Xie, R. Liang, Z. Liang, C. Huang, C. Zou, and B. Schuller, “Speech emotion classification using attention-based lstm,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 11, pp. 1675–1685, 2019.

[4] C. Northcutt, A. Athalye, and J. Mueller, “Pervasive label errors in test sets destabilize machine learning benchmarks,” in *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, December 2021.

[5] J. F. Gemmeke, D. P. W. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, “Audio set: An ontology and human-labeled dataset for audio events,” in *Proc. IEEE ICASSP 2017*, New Orleans, LA, 2017.

[6] P. Zinemanas, P. Cancela, and M. Rocamora, "MAVD-traffic dataset," July 2019.

[7] S. Wang, A. Mesaros, T. Heittola, and T. Virtanen, "A curated dataset of urban scenes for audio-visual scene analysis," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 626–630.

[8] J. Huh, J. Chalk, E. Kazakos, D. Damen, and A. Zisserman, "EPIC-SOUNDS: A Large-Scale Dataset of Actions that Sound," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2023.

[9] D. Damen, H. Doughty, G. M. Farinella, S. Fidler, A. Furnari, E. Kazakos, D. Moltisanti, J. Munro, T. Perrett, W. Price, and M. Wray, "Scaling egocentric vision: The epic-kitchens dataset," in *European Conference on Computer Vision (ECCV)*, 2018.

[10] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779–788.

[11] A. L. Cramer, H.-H. Wu, J. Salamon, and J. P. Bello, "Look, listen, and learn more: Design choices for deep audio embeddings," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 3852–3856.

[12] Z.-Q. Zhao, P. Zheng, S.-T. Xu, and X. Wu, "Object detection with deep learning: A review," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 11, pp. 3212–3232, 2019.

[13] G. Jocher, "Yolov5 by ultralytics," 2020. [Online]. Available: https://github.com/ultralytics/yolov5

[14] R. Arandjelovic and A. Zisserman, "Look, listen and learn," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 609–617.

[15] P. Zinemanas, P. Cancela, and M. Rocamora, "Mavd: A dataset for sound event detection in urban environments," in *Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE*, 2019.

[16] E. Fonseca, M. Plakal, D. P. W. Ellis, F. Font, X. Favory, and X. Serra, "Learning sound event classifiers from web audio with noisy labels," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 21–25.

[17] M. Fuentes, B. Steers, P. Zinemanas, M. Rocamora, L. Bondi, J. Wilkins, Q. Shi, Y. Hou, S. Das, X. Serra, and J. P. Bello, "Urban sound & sight: Dataset and benchmark for audio-visual urban scene understanding," in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 141–145.

LEARNING IN THE WILD: BIOACOUSTICS FEW SHOT LEARNING WITHOUT USING A TRAINING SET

*Víctor Aguado, Joan Navarro, Ester Vidaña-Vila**

La Salle Campus Barcelona, University Ramon Llull, Barcelona, ES
* Corresponding author: ester.vidana@salle.url.edu

ABSTRACT

Few-shot learning is a machine learning approach in which a pre-trained model is re-trained for new categories with just a few examples. This strategy results very convenient for tasks with a dynamic number of categories as typically happens in acoustic data. The purpose of this paper is to explore the possibility of skipping this pre-training process and using as training data only the five first shots of an audio file together with the silence between them. For the experimental evaluation, data belonging to the Validation set of Task 5 DCASE Challenge 2023 is used, purposely neglecting the Training set. This challenge consists of detecting animal species using only five positive examples. In this exploratory work, three learning methods have been compared: a ResNet architecture with a prototypical loss, a ProtoNet and an XGBoost classifier. In all cases, spectrograms with different transformations are used as inputs. Obtained results are evaluated per audio file, enabling the obtention of particular conclusions about different animal species. While the detection for some species presents encouraging results using only these first 5-shots as training data, all the tested algorithms are unable to successfully learn how to properly detect the blackbird sounds of the dataset.

***Index Terms*—** Bioacoustics, Few-shot learning, Prototypical networks, Acoustic Event Detection, Sound Event Detection

1. INTRODUCTION

Supervised machine learning methods aim at categorizing data from a training set containing (extensive amounts of) labeled data [1]. The performance of these techniques is typically evaluated with a test dataset that incorporates data samples that not only belong to the same categories as the training set but also adhere to a similar statistical distribution [2]. Since the early stages of artificial intelligence in the 1950s, such approaches have demonstrated promising results across diverse fields, including healthcare, computer vision, robotics, and finance, among many others. However, pursuing better accuracy and performance results, building more robust systems and processing an ever-increasing amount of features, has driven modern approaches to supervised machine learning (i.e., deep learning [3]) to be astonishingly data hungry [4, 5]. This data hungriness is especially concerning in those applications in which obtaining a high volume of labeled data to build a training dataset is unfeasible and/or the computational resources for processing all the training data are unavailable [6]. Recently, this situation has motivated the conception of what has been coined as few-shot learning paradigm: an alternative approach to current data-hungry supervised learning techniques that aims at building reliable systems with a dramatically low number of labeled training examples [6].

Few-shot learning can be viewed as an effort to emulate the innate ability of humans to leverage previously acquired knowledge when learning new concepts [7, 6]. For instance, learning to ride a motorbike may require less training if an individual already knows how to ride a bicycle. Traditional methods for few-shot learning aim to take advantage of prior knowledge about certain categories (e.g., bicycle riding in the previous example) in order to learn new ones (e.g., motorbike riding in the previous example) [6]. Interestingly, this machine learning approach has attracted a lot of interest in the field of bioacoustics, particularly for tasks related to sound event detection or species classification [8]. In this domain it is very common to encounter large acoustic datasets that are very time consuming to annotate and contain highly imbalanced classes (i.e., events with with infrequent occurrences versus highly recurrent events) [8].

Typical approaches to few-shot learning consist of using pre-trained systems with a (large) set of known classes and re-training them with few—usually between two and five—shots (i.e., examples) by means of different algorithms such as meta-learning and/or prototypical networks [8, 9]. These algorithms are still data hungry [5] and strongly rely on the particular tricks and data used in this pre-training process [8]. The purpose of this work is to explore the benefits of skipping the pre-training stage in few-shot learning for acoustic data and solely training the system with five shots of data (as positive samples) plus the silence surrounding each of them (as negative samples). To obtain reference values, this work has been contextualized in the Task 5 [10] of the DCASE Challenge 2023 - Few-shot bioacoustic event detection[1] that challenges participants to detect and classify vocalizations of animals using five examples (i.e., shots) of each one of the species. For the sake of this work, the Training set provided by the challenge organizers has been left aside on purpose and different classifiers (i.e., ResNet, ProtoNet, and XGBoost) have been trained using the aforementioned five shots from the Validation set. More specifically, every audio file has been used to train a model. Obtained results have been compared to the DCASE baseline for Task 5 that features a prototypical network. This work enables researchers to assess and quantify the benefits—in terms of F1-score—of the pre-training process in few-shot learning for this particular challenge.

The remainder of this paper is organized as follows. Section 2 describes the methodology for data collection and the selection of the classifiers. Next, Section 3 presents the experimental results and their comparison with the DCASE baseline. Finally, Section 4 concludes the paper.

---

[1] https://dcase.community/challenge2023/task-few-shot-bioacoustic-event-detection

2. METHODOLOGY

This section delves into the methodology employed in this few-shot learning study. We start by exposing the data collection as well as the preprocessing steps. Moreover, we introduce the experimental setup and the learning methods implemented. To end up, we show the prototypical loss used in the experimentation and its mathematical sense in order to classify every event.

2.1. Data Collection and Preprocessing

Data used in this study are obtained from the Validation set of Task 5 DCASE Challenge 2023 - Few-shot bioacoustic event detection. The dataset specifically focuses on animal species detection using only five positive examples. The audio files are collected from various sources and are annotated with the corresponding species labels. Available data are split into three datasets: Training, Validation, and Test. Note that classes in the Validation set are not available in the Training set. In the Validation set, unlike in the Training Set, only positive or negative labels are considered. That is, there is no more than one species per audio file. Therefore, the objective will be to train a model that is able to discern whether a given event is a vocalization or not. Note that in this exploratory work, the Training set is intentionally neglected, and only the Validation set is used for training the learning methods. This leads us to an "extreme" few-shot learning where one model is created and trained for each audio file with the task of detecting the corresponding vocalization. Also, it is worth mentioning that the Test set has not been used as the complete annotations are not publicly available.

Before conducting the experiments, we have conducted some preprocessing steps. This involves computing the first five positive event spectrograms labeled with positive class, as well as five negative samples. Negative spectrograms are computed from intervals of silence or noise between the first five positive vocalizations of a given duration. All spectrograms are equally sized and computed using the duration of the smaller known positive or negative sample in the few-shot samples of each audio. Figure 1 illustrates an example of this preprocessing step. In Figure 1, the smallest sample is the 4th negative. As we are using the minimum duration event as window size for obtaining the spectrograms, larger events will result split in more spectrograms, so the model may be trained with more than 5 positive and negative spectrograms belonging to the same sample. To avoid class imbalance, the number of positive and negative spectrograms is always the same, being the class that presents less samples the one that limits the amount of data of each category.

2.2. Learning Methods

Three different learning methods have been employed in this study to explore the benefits of using only the initial five shots of audio data in the Validation set for training:

2.2.1. *ResNet Architecture with Prototypical Loss*

The ResNet architecture [11], a popular deep neural network, is utilized in combination with the prototypical loss function. This approach aims to learn a feature representation space where examples from the same category are grouped together. The ResNet model is initially pre-trained on a large-scale dataset (ImageNet) and then fine-tuned using the limited training data from the first five shots (positive and negative) of the audio files.

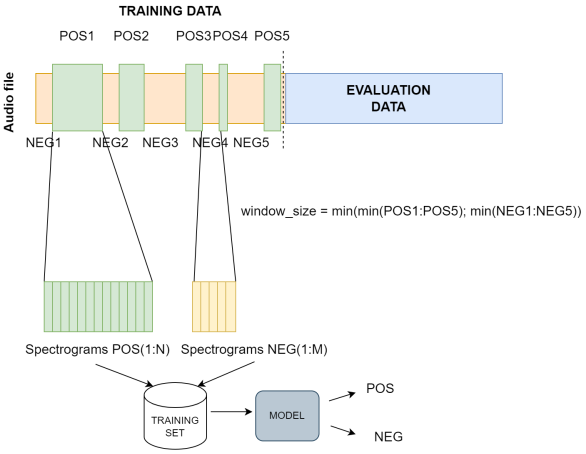


Figure 1: Extraction of positive (POS) and negative (NEG) samples from an audio file.

2.2.2. *ProtoNet*

ProtoNet [12] is a few-shot learning algorithm that builds prototype representations for each category based on a few labeled examples. It learns to classify new instances by computing similarity measures between the prototypes and the query samples. The ProtoNet uses an encoder, which is composed of multiple convolutional blocks. Each convolutional block includes a convolutional layer, batch normalization layer, ReLU activation function, and a max pooling layer. These layers are applied sequentially to the input data, transforming it and extracting meaningful features. The number of convolutional blocks can vary, but in this architecture, there are four convolutional blocks. In this study, a ProtoNet is trained with the initial five shots for each audio file and the first five computed negative samples.

2.2.3. *XGBoost Classifier*

XGBoost [13] is a gradient-boosting framework that is known for its high performance in various machine learning tasks. In this work, we aim to train a XGBoost classifier to learn from the first five shot spectrogram patterns as well as from the first five silences and make predictions on new instances.

2.3. Prototypical Loss

The prototypical loss, which has been used for training the ResNet and the ProtoNet, is a mathematical formulation used in few-shot learning tasks. Its objective is to train a model that can effectively classify new instances from unseen classes with only a small number of labeled examples. In this loss function, support examples are selected for each class in the Validation set. These support examples are used to define the characteristics of each class. The support examples of each class are averaged together to create a prototype representation, which serves as the centroid or central point of the support examples for that class (see Equation 1).

$$\mathbf{c}_j = \frac{1}{N_s} \sum_{i=1}^{N_s} \mathbf{x}_{ij} \tag{1}$$

Audio File	Precision (%)	Recall (%)	F1-score (%)
Overall ResNet	12.19	53.95	18.13
Overall ProtoNet	33.41	66.15	37.30
Overall XGBoost	31.19	64.26	36.53
Overall DCASE Baseline	22.1	49.01	28.31

Table 1: Overall percentage of Precision, Recall and F1-score of the 3 evaluated models and the DCASE Prototypical network baseline.

The remaining examples for each class, which were not used as support examples, are considered as query examples. The goal is to classify these query examples based on their similarity to the prototypes. In this case, for the similarity, Euclidean Distance is used. The prototypical loss is obtained by computing the mean log probability of the negative distances mentioned early for each ground truth class (see Equation 2). This loss encourages the model to assign high probabilities to the correct classes for the query examples. That is, this loss helps the model to project samples in an embedding space where query samples should lay near its ground truth prototype. In addition to the classification loss, a regularization term is added to the loss function. This term promotes compactness in the prototype representations by penalizing their norm.

In Equation 2 N_q represents the number of query samples. The numerator represents the exponential of the distance between the model output for query sample i and its corresponding prototype c_k. The denominator is formed by the sum of the exponential of all minus distances between query sample i and the rest of prototpyes. Finally λ is the regularization term that multiplies the norm of prototypes set.

$$\mathcal{L} = -\frac{1}{N_q}\sum_{i=1}^{N_q}\log\left(\frac{\exp(-\mathbf{d}(f_\phi(x_i), c_k)}{\sum_{k'}\exp(-\mathbf{d}(f_\phi(x_i), c_{k'})}\right) + \lambda\|\mathbf{c}\| \quad (2)$$

3. EXPERIMENTAL RESULTS

This section explains which metrics have been used and the obtained results of the experimental evaluation.

3.1. Performance Evaluation Metrics

To assess the performance of the learning methods, the following evaluation metrics are employed: precision, recall, and F1-score. Precision and recall assess the algorithm's ability to correctly classify positive instances and retrieve all relevant instances, respectively. The F1-score combines both precision and recall into a single metric. For computing those metrics, the True Positive, False Positive and False Negative rates of each audio file were obtained. The individual metrics of each audio file of the dataset have been calculated using the code provided for Task 5 2023 of the DCASE challenge, which is explained in [10]. After obtaining the individual metrics for every audio file, the metrics were averaged to obtain an overall score and thus be able to compare the different models.

3.2. Results and Analysis

Table 1 provides an overview of the performance of our three models. ResNet achieved a precision of 12.19%, indicating a poor ability to correctly identify positive instances. The recall score of 53.95% suggests that it captured a moderate proportion of relevant instances. The resulting F-measure was 18.13%, reflecting its overall performance.

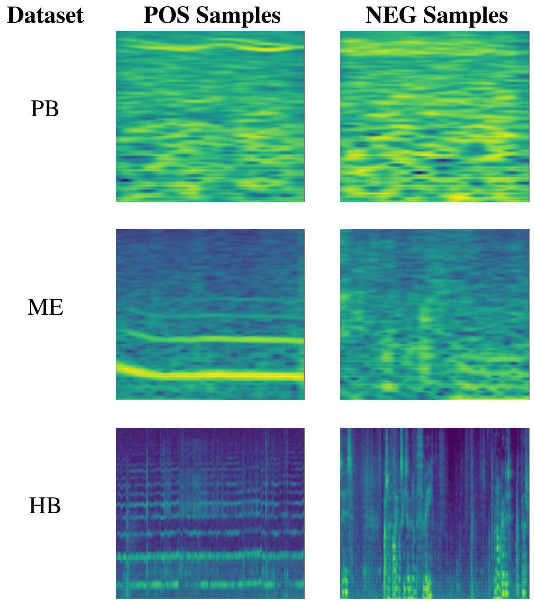


Figure 2: Positive (POS) and negative (NEG) spectrogram samples for each one of the datasets (PB, ME, HB) of the Validation set.

ProtoNet performed better than ResNet, with a precision of 33.41% and a recall of 66.15%. This means ProtoNet had a higher ability to identify positive instances and capture relevant instances. As a result, it achieved an F-measure of 37.36%.

XGBoost showed similar performance to ProtoNet, with a precision of 31.19% and a recall of 64.26%. Its F-measure was 36.53%, indicating a very similar effectiveness to ProtoNet. However it is important to point out that XGBoost requires less than the half of the training time than the ProtoNet and, also, it can be trained on a CPU with a reasonable amount of time.

Furthermore, it is worth noting that all the models presented in this study achieved higher recall metrics compared to the Prototypical network Baseline provided by DCASE Task 5 (49.01%). However, in terms of precision and F-measure, only the ProtoNet and XGBoost models outperformed the DCASE baseline that was pre-trained with the Training dataset.

In summary, ProtoNet and XGBoost outperformed ResNet in terms of precision, recall, and F-measure, with ProtoNet achieving the highest F-measure among the three models. Generally, these models are thought to be deployed in low-complexity edge devices that can be trained and used in a specific environment. For that purpose, ProtoNet is easier to deploy in a low-complexity device due to its simpler architecture and lower computational requirements. It has fewer parameters and can run efficiently on devices with limited resources. On the other hand, at inference time, XGBoost requires more computational resources and may be more challenging to deploy on low-complexity devices.

In terms of a per-audio analysis, the F1-scores for each model and audio file have been computed and summarized in Table 2. All three presented models struggle at detecting correctly the PB dataset (blackbirds). By far, this dataset presents the worst results out of the three datasets, As it can be observed, the highest score is obtained

Dataset	Animal	Audio File	F1-score ResNet(%)	F1-score ProtoNet (%)	F1-score XGBoost (%)	F1-score Baseline (%)
PB	Blackbirds	BUK1_20181011_001004.wav	0.53	1.26	0.41	2.09
		BUK1_20181013_023504.wav	0.11	0.22	0.14	5.72
		BUK4_20161011_000804.wav	0.17	0.37	0.14	0.35
		BUK4_20171022_004304a.wav	4.04	0.49	0.46	19.35
		BUK5_20161101_002104a.wav	7.68	1.88	1.88	7.67
	Song Thrush	BUK5_20180921_015906a.wav	0.14	0.21	0.21	3.38
ME	Meerkats	ME1.wav	13.45	4.32	2.04	3.48
		ME2.wav	56.25	29.14	46.38	19.51
HB	Mosquitos	R4_cleaned recording_13-10-17.wav	39.08	78.98	70.27	32.43
		R4_cleaned recording_16-10-17.wav	17.92	60.24	57.83	58.33
		R4_cleaned recording_17-10-17.wav	15.70	67.40	64.81	10.37
		R4_cleaned recording_TEL_19-10-17.wav	10.86	80.00	38.03	67.54
		R4_cleaned recording_TEL_20-10-17.wav	36.20	88.47	71.79	18.18
		R4_cleaned recording_TEL_23-10-17.wav	16.03	91.93	76.24	72.48
		R4_cleaned recording_TEL_24-10-17.wav	55.88	81.15	80.91	72.32
		R4_cleaned recording_TEL_25-10-17.wav	31.09	35.61	86.27	37.65
		file_423_487.wav	4.53	35.45	46.01	59.88
		file_97_113.wav	16.67	15.35	12.84	18.93
		Overall Scores	**18.13**	**37.30**	**36.53**	**28.31**

Table 2: Percentage (%) of F1-score per audio file of the Validation set.

by ResNet with a 7.68% of F-measure. This also happens when using the DCASE Baseline, even though in that case there is an audio file that achieved an F1-score of up to 19.35%. To motivate this behaviour, Figure 2 shows an example of positive (POS) and negative (NEG) spectrograms of this dataset. As it can be observed, the PB dataset is the one that presents more noise, with the bird vocalization being almost masked by the background noise. Visually, it is even hard to distinguish the difference between the two of them (it is the yellowest flat line on the top part of the spectrogram). With the obtained spectrograms, the presence of noise in the PB audio files might have affected the models' ability to extract relevant features and make accurate predictions, resulting in the obtained lower F1-scores. Covnersely, in the ME (Meerkats) category, the ResNet model obtained an F1-score of 13.45% in one audio file, which outperformed ProtoNet (F1-score of 4.32%) and XGBoost (F1-score of 2.04%) in the same file. On the other hand, the XGBoost model performed exceptionally well in the other file (ME2.wav) with an F1-score of 29.14%, surpassing the scores of ResNet (F1-score of 56.25%) and ProtoNet (F1-score of 46.38%). In average, the three presented models obtain better results than the DCASE baseline (except for the first audio file and the XGBoost model). Finally, for the HB (Mosquitos) category, the ResNet model achieved an F1-score of 39.08%, followed by ProtoNet with 78.98%, and XGBoost with 70.27%. The F1-scores in this category indicate that ProtoNet performed better than the other two models and the baseline.

When interpreting the results, it is crucial to consider the challenging nature of the PB audio files (very short vocalizations, background noise) and the impact they had on the models' performance. In noisy scenarios, it may be necessary to explore additional preprocessing techniques or consider using specialized models or algorithms specifically designed to handle such conditions. In this work, PCEN [14] was evaluated as a possible technique to mitigate noise, but it was discarded as it did not significantly improve the results. It is also important to consider that every audio has an independent model, so this approach is highly affected by the first initial five shots for building a solid basis to predict the rest of the audio.

4. CONCLUSION

In this study, we explored the task of bioacoustic events detection using few-shot learning techniques. Every model was trained using solely the first five positive examples of animal vocalizations as well as the first five silences (where silence means absence of the species to be detected) of every audio of the Validation set of Task 5 DCASE Challenge 2023, meaning that the Training set was not used.

Three learning methods have been evaluated: ResNet, ProtoNet, and XGBoost and compared to the DCASE baseline.

The results demonstrated that ProtoNet and XGBoost outperformed ResNet in terms of precision, recall, and F1-score. ProtoNet achieved the highest F-measure among the three models, indicating its effectiveness in discerning positive instances and capturing relevant examples. This leads us to think that simpler models in terms of parameters perform better than complex ones in few-shot learning scenarios where the training examples are limited. In general, obtained results surpass the DCASE baseline.

However, it is important to note that the presence of background noise, especially in the PB dataset, supposed a challenge to the models' performance. This highlights the need for additional preprocessing techniques and specialized models to handle such challenging conditions.

Future work should focus on improving data preprocessing techniques (e.g., filtering denoising algorithms) and exploring advanced few-shot learning methods. Moreover, it should be analysed whether expanding the dataset through data augmentation results in better performance.

5. ACKNOWLEDGMENTS

The authors would like to thank the Departament de Recerca i Universitats (Generalitat de Catalunya) under Grants Ref. 2021-SGR-01396 for Ester Vidaña-Vila and 2021-SGR-01398 for Joan Navarro for the funding of HER and SmartSociety Research Groups.

6. REFERENCES

[1] S. B. Kotsiantis, I. Zaharakis, P. Pintelas, *et al.*, “Supervised machine learning: A review of classification techniques,” *Emerging artificial intelligence applications in computer engineering*, vol. 160, no. 1, pp. 3–24, 2007.

[2] T. Hastie, R. Tibshirani, J. Friedman, T. Hastie, R. Tibshirani, and J. Friedman, “Overview of supervised learning,” *The elements of statistical learning: Data mining, inference, and prediction*, pp. 9–41, 2009.

[3] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[4] A. Adadi, “A survey on data-efficient algorithms in big data era,” *Journal of Big Data*, vol. 8, no. 1, p. 24, 2021.

[5] D. Stowell, “Computational bioacoustics with deep learning: a review and roadmap,” *PeerJ*, vol. 10, p. e13152, 2022.

[6] A. Parnami and M. Lee, “Learning from few examples: A summary of approaches to few-shot learning,” *arXiv preprint arXiv:2203.04291*, 2022.

[7] L. Fei-Fei, “Knowledge transfer in learning to recognize visual objects classes,” in *Proceedings of the International Conference on Development and Learning (ICDL)*, vol. 11, 2006.

[8] I. Nolasco, S. Singh, E. Vidała-Vila, E. Grout, J. Morford, M. Emmerson, F. H. Jensen, I. Kiskin, H. Whitehead, A. Strandburg-Peshkin, L. Gill, H. Pamuła, V. Lostanlen, V. Morfi, and D. Stowell, “Few-shot bioacoustic event detection at the dcase 2022 challenge,” in *Proceedings of the 7th Detection and Classification of Acoustic Scenes and Events 2022 Workshop (DCASE2022)*, Nancy, France, November 2022.

[9] Q. Sun, Y. Liu, T.-S. Chua, and B. Schiele, “Meta-transfer learning for few-shot learning,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 403–412.

[10] V. Morfi, I. Nolasco, V. Lostanlen, S. Singh, A. Strandburg-Peshkin, L. F. Gill, H. Pamula, D. Benvent, and D. Stowell, “Few-shot bioacoustic event detection: A new task at the dcase 2021 challenge.” in *DCASE*, 2021, pp. 145–149.

[11] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” 2015.

[12] J. Snell, K. Swersky, and R. S. Zemel, “Prototypical networks for few-shot learning,” *CoRR*, vol. abs/1703.05175, 2017. [Online]. Available: http://arxiv.org/abs/1703.05175

[13] T. Chen and C. Guestrin, “Xgboost: A scalable tree boosting system,” in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 2016, pp. 785–794.

[14] V. Lostanlen, J. Salamon, M. Cartwright, B. McFee, A. Farnsworth, S. Kelling, and J. P. Bello, “Per-channel energy normalization: Why and how,” *IEEE Signal Processing Letters*, vol. 26, no. 1, pp. 39–43, 2018.

MULTI-RESOLUTION CONFORMER FOR SOUND EVENT DETECTION: ANALYSIS AND OPTIMIZATION

Sara Barahona, Diego de Benito-Gorron, Sergio Segovia, Daniel Ramos, Doroteo T. Toledano

AUDIAS Research Group
Universidad Autónoma de Madrid
Calle Francisco Tomás y Valiente, 11, 28049 Madrid, SPAIN
sara.barahona@estudiante.uam.es, diego.benito@uam.es,
sergio.segoviag@estudiante.uam.es, daniel.ramos@uam.es, doroteo.torre@uam.es

ABSTRACT

The Conformer architecture has achieved state-of-the-art results in several tasks, including automatic speech recognition and automatic speaker verification. However, its utilization in sound event detection and in particular in the DCASE Challenge Task 4 has been limited despite winning the 2020 edition. Although the Conformer architecture may not excel in accurately localizing sound events, it shows promising potential in minimizing confusion between different classes. Therefore, in this paper we propose a Conformer optimization to enhance the second Polyphonic Sound Detection Score (PSDS) scenario defined for the DCASE 2023 Task 4A. With the aim of maximizing its classification properties, we have employed recently proposed methods such as Frequency Dynamic Convolutions in addition to our multi-resolution approach, which allow us to analyse its behaviour over different time-frequency resolution points. Furthermore, our Conformer systems are compared with multi-resolution models based on Convolutional Recurrent Neural Networks (CRNNs) to evaluate the respective benefits of each architecture in relation to the two proposed scenarios for the PSDS and the different time-frequency resolution points defined. These systems were submitted as our participation in the DCASE 2023 Task 4A, in which our Conformer system obtained a PSDS2 value of 0.728, achieving one of the highest scores for this scenario among systems trained without external resources.

Index Terms— DCASE 2023, Sound Event Detection, Conformer, PSDS, Multi-resolution, Model fusion

1. INTRODUCTION

Sound Event Detection (SED) is the task that aims to detect and classify different sound events present within an audio clip. Although research in SED has a long history, the last few years have witnessed an increasing interest in the field, motivated in part by the publication of Google Audio Set [1] and the yearly challenges and workshops organized by the DCASE community [2]. This paper is centered in the context of one of these challenges, in particular the DCASE Task 4A: Sound Event Detection with Weak Labels and Synthetic Soundscapes. The goal of this task is to evaluate SED systems by employing both real and synthetic recordings which contain 10 sound event classes that can be found in a domestic environment. Besides, it tackles the issue of employing unlabeled data as well as different types of annotations: strong labels that provide temporal information (timestamps) along with the sound event category, and weak labels which solely indicate the category.

The metric employed for evaluating SED systems in this task is the Polyphonic Sound Detection Score (PSDS) [3], that relies on the intersection between detected and annotated sound events. Considering that it can be tuned for evaluating different properties of a SED system, two PSDS scenarios are proposed for the DCASE Challenge 2023 Task 4A. Whereas the first one (PSDS1) focuses on a fast reaction upon a sound event, requiring highly accurate localization, the second scenario (PSDS2) aims to avoid the confusion between classes, and it is not strict about timing errors.

Over the last few years, different architectures have been proposed to address this task. Since 2018, the baseline is based on a Convolutional Recurrent Neural Network (CRNN) [4], which employs CNNs for extracting local characteristics and RNNs to exploit temporal dependencies. Architectures based on attention mechanisms such as the Transformer [5] or the Conformer (Convolution Augmented Transformer) [6] have also been explored for this task. The Conformer architecture has been successfully employed by recent state-of-the-art models in tasks such as automatic speech recognition (ASR) [7] and automatic speaker verification (ASV) [8]. In the field of sound event detection, it achieved promising results winning the DCASE Challenge Task 4 in 2020 [9]. However, in the subsequent editions it was scarcely used, to the extend that last year we were the only team that submitted systems based on this architecture [10]. Although our experiments revealed a better performance of CRNN-based systems in terms of PSDS1, we observed the potential of the Conformer at classifying sound events. Therefore, in this paper we propose a continuation to our previous research by optimizing the Conformer architecture towards the PSDS2 and analysing its performance following our multi-resolution approach.

For this purpose, we introduce the Conformer architecture and describe the methodologies employed for its optimization in Section 2. The results of our experiments are presented and analysed in Section 3. Finally, Section 4 highlights the salient conclusions derived from this investigation.

2. PROPOSED METHODS

The Conformer (Convolution-Augmented Transformer) was designed with the aim of building an attention-based network capable of extracting both local and global features. For this purpose, a convolution module is added to the Transformer backbone. To solve temporal confusion, the relative positional embedding proposed for the Transformer-XL [11] is added to the global content-based attention mechanism. While this approach initially appeared to be highly promising for addressing the detection and classification of sound events, the Conformer has exhibited limitations in accurately localizing timestamps, resulting in a lower performance in terms of PSDS1 when compared with CRNNs. However, the Conformer has shown a great ability at classifying correctly each sound event, even when two sounds are similar or noise is present in an audio clip.

Considering that the main weakness of the Conformer architecture is the lack of temporal resolution, we propose to optimize a Conformer-based system towards the PSDS2. To accomplish this objective, we employ a multi-resolution approach to assess the system's effectiveness across various time-frequency resolution settings. Considering the pronounced influence of median filtering on the temporal resolution of a SED system's output, we adapt this post-processing technique to the scenario we are targeting. To evaluate the proposed methods in the framework of the DCASE Challenge Task 4A, we compare the performance of our Conformer systems with a multi-resolution version of the official baseline model based on CRNNs.

2.1. Optimized Conformer for PSDS2

Our Conformer model is based on the DCASE 2020 Task 4 winner [9], which consist of a CNN for feature extraction with 4 conformer blocks stacked. Additionally, they employ a tagging token similar to the classification token used in BERT [12] to summarize the weak label predictions through the attention layers.

To improve the PSDS2 value, we perform a hyperparameter tuning setting as objective this metric, leading to an optimal configuration of 7 Conformer blocks with 4 attention heads each and an encoder dimension of 144. Additionally, we substitute the CNN-based feature extractor with a Frequency Dynamic Convolution Neural Network (FDY-CNN) [13] to improve the classification of non-stationary sound events. For the FDY-CNN we employ context gating as the activation function and define a time-resolution reduction of 8 by adding one more average-pooling layer along the temporal dimension. Data augmentation techniques have also been applied to avoid confusion between classes. By this means, we employ both Mixup and FilterAugment [14] with a probability of 50% of applying them to the training data.

As semi-supervised learning, we utilize the mean-teacher method [15] for training both architectures. This method employs two identical models: student and teacher, whose weights are the exponential average weights of the student. By minimizing a consistency cost between the predictions of the student and teacher, the model learns to generate targets from unlabeled data. Generally, the teacher model achieves a more consistent learning trajectory across epochs, leading to a superior performance during testing. Thus, model selection is performed over the teacher network, adjusting the objective metric based on the specific scenario we are targeting. Whereas for the CRNN we employ the one set for the baseline (F1-score based on intersection), our Conformer systems use the PSDS2.

2.2. Multi-resolution analysis

In previous research, we proposed a multi-resolution approach which consist on varying the parameters employed for the extraction of mel-spectrogram features. Our multi-resolution approach has demonstrated the advantages of employing distinct time-frequency resolutions that align with the characteristics of each PSDS scenario or sound event category. Given that the main weakness of the Conformer seems to be the time resolution of its detections, we will explore how the different time-frequency resolutions impact the performance of this architecture.

Resolution	$\mathbf{T_{++}}$	$\mathbf{T_{+}}$	**BS**	$\mathbf{F_{+}}$	$\mathbf{F_{++}}$
N	1024	2048	2048	4096	4096
L	1024	1536	2048	3072	4096
R	128	192	256	384	512
$\mathbf{n_{mel}}$	64	96	128	192	256

Table 1: FFT length (N), window length (L), window hop (R) and number of Mel filters (n_{mel}) of the five resolution points employed for the feature extraction. N, L, and R are reported in samples, using a sample rate $f_s = 16000$ Hz.

Considering the trade-off between time and frequency resolution of the Short Time Fourier Transform (STFT), we design a total of 5 resolution points such that they span a range from higher frequency resolution to higher time resolution, relative to the original resolution utilized by the baseline system.

As presented in Table 1, we establish the resolution of the baseline system as the intermediate one (referred to as BS). From this one we define four additional resolution points. Among these, two are designed to double the resolution in frequency (F_{++}) and in time (T_{++}), whereas the remaining two are halfway points between BS and F_{++} (F_{+}) or T_{++} (T_{+}).

Single-resolution models are obtained by training each system with one of the points mentioned above. They can be combined into multi-resolution systems by frame-wise averaging the sequences of scores. As this combination is performed frame-wise, the sequences must have the same length. However, the different time resolutions defined in Table 1 lead to different lengths of the score sequences: $T_1, T_2, ...T_N$. To handle this issue we perform a linear interpolation of the sequences to the maximum length, $T_{max} = \max\{T_1, T_2, ...T_N\}$.

2.3. Class-dependent median filtering

Our multi-resolution approach is based on the fact that each sound event class presents different temporal and spectral characteristics. Therefore, smoothing the decoded predictions employing the same median filter for every class would be counter-productive. Additionally, each PSDS scenario can benefit from different window lengths. Whereas shorter median filters can improve the localization of onsets and offsets, longer windows may be advantageous for avoiding potential cross-triggers and, therefore, enhance the PSDS2.

For this purpose, we have employed a class-dependent median filtering in which the optimal lengths of each class are computed based on one of the PSDS scenarios, iterating over a range from 1 to 29 frames on the DESED Validation set.

3. EXPERIMENTS AND RESULTS

3.1. Dataset

For our experimental results we will use the DESED (Domestic Environment Sound Event Detection) dataset [16], which is the data proposed for the DCASE Task 4A. This dataset contains both real recordings, which are obtained from Google AudioSet [1], and synthetically generated audios employing the Scaper library [17]. The training data is composed of a synthetic strongly-labeled set (10,000

PSDS	DTC	GTC	CTTC	$\alpha_{\mathbf{CT}}$	$\alpha_{\mathbf{ST}}$	$\mathbf{e_{max}}$
Scenario 1	0.7	0.7	0.0	-	1.0	100
Scenario 2	0.1	0.1	0.3	0.5	1.0	100

Table 2: Parameter configuration for the PSDS scenarios.

clips), a real weakly-labeled set (1,578 clips) and a real unlabeled set (14,412 clips).

To select the best model during the training procedure, the synthetic validation set (2,500 clips) together with a 10% of the weakly-labeled set is employed. For testing, we use the validation set, which was constructed to match the clip-per-class distribution of the weakly labeled training set. It consists of 1,168 real audio clips annotated with strong labels.

3.2. Evaluation framework

The Polyphonic Sound Detection Score (PSDS) [3] was proposed for the DCASE Challenge 2021 Task 4 to overcome the limitations of event-based metrics, which rely on the overlap of collars and depend on a unique operating point. For this purpose, they define the Detection Tolerance Criterion (DTC) and the Ground Truth Intersection Criterion (GTC), which measure percentages of intersection between ground-truth labels and detected sound events. Additionally, they introduce the Cross-Trigger Tolerance Criterion (CTTC) to consider data bias by distinguishing the subset of false positives that intersect with labeled events, named as *cross-trigger*.

By modifying the threshold of intersection to these criteria, different properties of a SED system can be evaluated. As it is shown in Table 2, PSDS1 is defined with higher values for the DTC and GTC to measure a high intersection between labels and predictions. Conversely, these values are lower for the PSDS2 but in this case, the CTTC is taken into account to penalize the confusion between classes, whose cost is influenced by α_{CT}.

Results are provided for the recently proposed threshold-independent PSDS [18] over the DESED Validation set. Each model has been trained with three different initializations with the aim of estimating the performance's standard deviation. Moreover, we have compared the complexity of individual systems by calculating the Multiply–Accumulate Operations (MACs) for 10 seconds of audio prediction, a metric that was introduced in this year's evaluation.

3.3. Single-resolution results

The performance of both architectures for the different time-frequency resolution points defined is presented in Table 3. It is clearly seen that CRNN-based systems achieve higher PSDS1 results, evidencing the Conformer's limited temporal precision, which is accentuated when employing features that are not temporally enhanced (F_{++}). However, the Conformer system clearly outperforms the CRNN model in terms of PSDS2. Moreover, our Conformer system exhibits a reduced level of complexity in terms of Multiply-Accumulate operations (MACs). This metric is also influenced by the different resolution points, with lower values observed for frequency enhanced points, as they present shorter input lengths. All Conformer results in Table 3 use FDY, which provides enhanced performance as shown in Table 4.

Figure 1a shows a prototypical example highlighting the advantages and limitations of the different architectures. The CRNN accurately predicts the location of each event but confuses the second one by predicting a *Blender* instead of a *Vacuum cleaner*. This detection is considered a cross-trigger and will downgrade the PSDS2 value. In contrast, the Conformer predicts correctly the presence of both sound events in the clip, but it lacks temporal precision, lowering the PSDS1 results. The effect of the low resolution in time of the Conformer is even more visible in Figure 1b, where the prediction of continuous short events such as *Alarm_bell_ringing* is grouped into a single one.

Additionally, results show that each PSDS scenario benefits from a particular resolution point independently of the architecture employed. As expected, PSDS1 benefits from higher temporal resolution, whereas an enhancement in frequency resolution improves the results for PSDS2.

CRNN	PSDS1	PSDS2	MACs
$\mathbf{F_{++}}$	0.316 ± 0.004	0.561 ± 0.012	**0.891G**
$\mathbf{F_{+}}$	0.347 ± 0.015	**0.583 ± 0.022**	0.905G
BS	0.369 ± 0.006	0.579 ± 0.015	0.930G
$\mathbf{T_{+}}$	0.368 ± 0.039	0.550 ± 0.066	1.772G
$\mathbf{T_{++}}$	**0.374 ± 0.003**	0.575 ± 0.015	1.824G
Conformer	**PSDS1**	**PSDS2**	**MACs**
$\mathbf{F_{++}}$	0.194 ± 0.022	0.688 ± 0.015	**0.588G**
$\mathbf{F_{+}}$	0.224 ± 0.030	**0.696 ± 0.030**	0.633G
BS	0.263 ± 0.020	0.688 ± 0.018	0.879G
$\mathbf{T_{+}}$	0.251 ± 0.019	0.682 ± 0.014	1.147G
$\mathbf{T_{++}}$	**0.349 ± 0.029**	0.668 ± 0.015	1.331G

Table 3: Average and standard deviation results of individual CRNN and Conformer systems trained with different resolution points and initialized with diverse seeds over the DESED Validation set. Independent median filter was applied.

Architecture	PSDS1	PSDS2
CNN + Conformer	0.220 ± 0.027	0.607 ± 0.018
FDY-CNN + Conformer	**0.263 ± 0.020**	**0.688 ± 0.018**

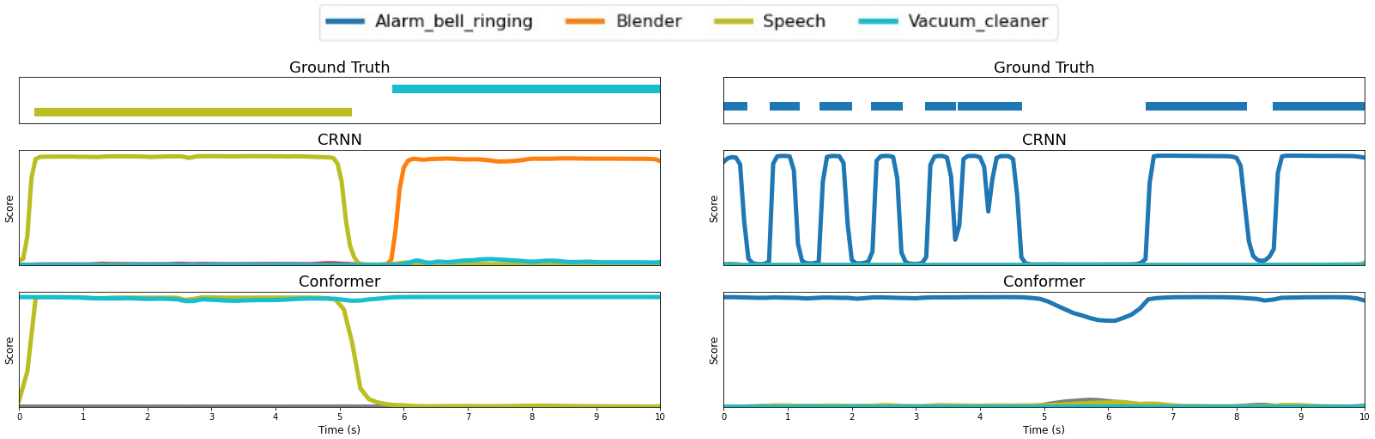
Table 4: Effects of employing FDY for the CNN-based feature extractor over the DESED Validation set.

3.4. Multi-resolution results

Single-resolution models are combined following the process described in Section 2.2 in order to obtain multi-resolution systems. In Table 5 the results of six combinations with up to five resolution points are presented individually for CRNNs and Conformers. Multi-resolution not only enhances the performance of single-resolution models, but also evidence that the combination of certain resolution points is more effective for a specific PSDS scenario. For both architectures, the PSDS1 is enhanced when employing a combination of resolutions enhanced in time. Conversely, the PSDS2 benefits from a combination of the five resolution points defined, which is logical as some sound events can be better distinguished by their spectral behaviour while others are better recognized based on their temporal properties.

3.5. Results with task-dependent median filtering

We have experimented with the class-dependent median filtering described in Section 2.3 in our best single-resolution systems (CRNN_T++ and Conformer_F+) and in our two optimal multi-



(a) YG3yk9UXVB9g_20.000_30.000.wav (b) Y9OqtuFGCCR8_30.000_40.000.wav

Figure 1: Ground truth, CRNN and Conformer predictions for two audio recordings from the DESED Validation set considering the baseline resolution.

		CRNN		**Confomer**	
Resolutions		**PSDS1**	**PSDS2**	**PSDS1**	**PSDS2**
3res	F$_{+}$, BS, T$_{+}$	0.397 ± 0.010	0.615 ± 0.012	0.275 ± 0.012	0.719 ± 0.017
3res-F	F$_{++}$, F$_{+}$, BS	0.375 ± 0.007	0.617 ± 0.013	0.255 ± 0.015	0.722 ± 0.014
3res-T	BS, T$_{+}$, T$_{++}$	0.401 ± 0.007	0.611 ± 0.014	**0.329 ± 0.013**	0.715 ± 0.017
4res-F	F$_{++}$, F$_{+}$, BS, T$_{+}$	0.390 ± 0.007	0.623 ± 0.012	0.268 ± 0.010	0.724 ± 0.015
4res-T	F$_{+}$, BS, T$_{+}$, T$_{++}$	**0.405 ± 0.005**	0.624 ± 0.013	0.309 ± 0.017	0.721 ± 0.016
5res	F$_{++}$, F$_{+}$, BS, T$_{+}$, T$_{++}$	0.398 ± 0.005	**0.632 ± 0.011**	0.306 ± 0.006	**0.727 ± 0.015**

Table 5: Average and standard deviations results for three initialization seeds of multi-resolution combinations of CRNN and Conformer systems over the DESED Validation set. Fixed median filter was applied.

Obj.	**Model**	**PSDS1**	**PSDS2**
PSDS1	**CRNN_T$_{++}$**	0.387 ± 0.004	0.585 ± 0.012
	CRNN_4res-T	**0.416 ± 0.005**	0.626 ± 0.016
PSDS2	**Conformer_F$_{+}$**	0.164 ± 0.018	0.740 ± 0.033
	Conformer_5res	0.243 ± 0.007	**0.781 ± 0.017**
-	**Baseline**	0.359 ± 0.006	0.562 ± 0.012
	ConformerSED [19]	0.341 ± 0.013	0.576 ± 0.015

Table 6: Effects of employing a class-dependent median filtering on our submitted systems. The Obj. column indicates the objective metric employed to optimize the median filter length of each class. The official baseline and a reproduction of the Miyazaki et al. Conformer system [9] are included for comparison purpose. Results are provided over the DESED Validation set.

resolution systems (CRNN_4res-T and Conformer_5res). Considering that the set of median filters learnt vary depending on which metric is set as objective, we have considered for each system the same PSDS scenario for which it has been designed: PSDS1 for CRNN models and PSDS2 for Conformers.

As we present in Table 6, the systems optimized for PSDS1 improve their results in this metric when the median filters are tuned according the best class-wise PSDS1 performance (from 0.374 to 0.387 in CRNN_T++, and from 0.405 to 0.416 in CRNN_4res-T). Additionally, this criterion is helpful for the PSDS2 as well.

When it comes to the systems optimized for the second scenario, their PSDS2 value is also enhanced when the median windows are tuned class-wise (from 0.696 to 0.740 in Conformer_F+, and from 0.727 to 0.781 in Conformer_5res). However, the median filters learnt with this criterion considerably downgrade the performance for PSDS1.

4. CONCLUSIONS

In this paper we presented the benefits of the Conformer architecture for sound event detection by optimizing a system towards the second scenario proposed for the DCASE Challenge 2023 Task 4A. Among the submitted systems without employing external data, our Conformer system achieves one of the best PSDS2 values over the evaluation set (0.729).

Following our previous multi-resolution approach, we were able to analyse its behaviour over different time-frequency resolutions and compare its performance with a CRNN-based system. Additionally, by employing this technique we not only demonstrate that a multi-resolution ensemble can considerably enhance the results, but also revealed that the different PSDS scenarios benefit from features that enhance either time or frequency resolution. Therefore, we obtain the best PSDS1 when combining CRNN systems trained with resolution points enhanced in time, while our best PSDS2 is obtained when combining the five resolutions defined for the Conformer.

5. REFERENCES

[1] J. F. Gemmeke, D. P. W. Ellis, *et al.*, "Audio set: An ontology and human-labeled dataset for audio events," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, New Orleans, LA, 2017.

[2] http://dcase.community.

[3] Bilen, G. Ferroni, F. Tuveri, J. Azcarreta, and S. Krstulović, "A framework for the robust evaluation of sound event detection," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 61–65.

[4] N. Turpault, R. Serizel, A. Parag Shah, and J. Salamon, "Sound event detection in domestic environments with weakly labeled data and soundscape synthesis," in *Workshop on Detection and Classification of Acoustic Scenes and Events*, New York City, United States, October 2019. [Online]. Available: https://hal.inria.fr/hal-02160855

[5] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ser. NIPS'17. Red Hook, NY, USA: Curran Associates Inc., 2017, p. 6000–6010.

[6] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu, *et al.*, "Conformer: Convolution-augmented transformer for speech recognition," *arXiv preprint arXiv:2005.08100*, 2020.

[7] J. Li, "Recent advances in end-to-end automatic speech recognition," *APSIPA Transactions on Signal and Information Processing*, vol. 11, no. 1, pp. –, 2022. [Online]. Available: http://dx.doi.org/10.1561/116.00000050

[8] Y. Zhang, Z. Lv, H. Wu, S. Zhang, P. Hu, Z. Wu, H. yi Lee, and H. M. Meng, "Mfa-conformer: Multi-scale feature aggregation conformer for automatic speaker verification," in *Interspeech*, 2022.

[9] K. Miyazaki, T. Komatsu, T. Hayashi, S. Watanabe, T. Toda, and K. Takeda, "Conformer-based sound event detection with semi-supervised learning and data augmentation," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2020 Workshop (DCASE2020)*, Tokyo, Japan, November 2020, pp. 100–104.

[10] D. de Benito-Gorron, S. Barahona, S. Segovia, D. Ramos, and T. Doroteo, "Multi-resolution combination of CRNN and conformers for dcase 2022 task 4," DCASE2022 Challenge, Tech. Rep., June 2022.

[11] Z. Dai, Z. Yang, Y. Yang, J. Carbonell, Q. V. Le, and R. Salakhutdinov, "Transformer-xl: Attentive language models beyond a fixed-length context," 2019.

[12] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 4171–4186. [Online]. Available: https://aclanthology.org/N19-1423

[13] H. Nam, S.-H. Kim, B.-Y. Ko, and Y.-H. Park, "Frequency dynamic convolution: Frequency-adaptive pattern recognition for sound event detection," *arXiv preprint arXiv:2203.15296*, 2022.

[14] H. Nam, S.-H. Kim, and Y.-H. Park, "Filteraugment: An acoustic environmental data augmentation method," in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 4308–4312.

[15] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," in *Advances in neural information processing systems*, 2017, pp. 1195–1204.

[16] R. Serizel, N. Turpault, A. Shah, and J. Salamon, "Sound event detection in synthetic domestic environments," in *ICASSP 2020 - 45th International Conference on Acoustics, Speech, and Signal Processing*, Barcelona, Spain, 2020. [Online]. Available: https://hal.inria.fr/hal-02355573

[17] J. Salamon, D. MacConnell, M. Cartwright, P. Li, and J. P. Bello, "Scaper: A library for soundscape synthesis and augmentation," in *2017 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2017, pp. 344–348.

[18] J. Ebbers, R. Haeb-Umbach, and R. Serizel, "Threshold independent evaluation of sound event detection scores," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 1021–1025.

[19] K. Miyazaki, "ConformerSED," https://github.com/mkoichi/ConformerSED, 2021.

FOLEY SOUND SYNTHESIS AT THE DCASE 2023 CHALLENGE

Keunwoo Choi[1*], *Jaekwon Im*[1,2*], *Laurie M. Heller*[3*], *Brian McFee*[4],
Keisuke Imoto[5], *Yuki Okamoto*[6], *Mathieu Lagrange*[7], *Shinnosuke Takamichi*[8]

[1] Gaudio Lab, Inc., Seoul, South Korea, {keunwoo, jaekwon}@gaudiolab.com
[2] KAIST, Daejeon, South Korea
[3] Carnegie Mellon University, USA, laurieheller@cmu.edu
[4] New York University, USA, brian.mcfee@nyu.edu
[5] Doshisha University, Japan, keisuke.imoto@ieee.org
[6] Ritsumeikan University, Japan, y-okamoto@ieee.org
[7] CNRS, Ecole Centrale Nantes, Nantes Université, France, mathieu.lagrange@ls2n.fr
[8] The University of Tokyo, Japan, shinnosuke_takamichi@ipc.i.u-tokyo.ac.jp

ABSTRACT

The addition of Foley sound effects during post-production is a common technique used to enhance the perceived acoustic properties of multimedia content. Traditionally, Foley sound has been produced by human Foley artists, which involves manual recording and mixing of sound. However, recent advances in sound synthesis and generative models have generated interest in machine-assisted or automatic Foley synthesis techniques. To promote further research in this area, we have organized a challenge in DCASE 2023: Task 7 - Foley Sound Synthesis. Our challenge aims to provide a standardized evaluation framework that is both rigorous and efficient, allowing for the evaluation of different Foley synthesis systems. We received 17 submissions, and performed both objective and subjective evaluation to rank them according to three criteria: audio quality, fit-to-category, and diversity. Through this challenge, we hope to encourage active participation from the research community and advance the state-of-the-art in automatic Foley synthesis. In this paper, we provide a detailed overview of the Foley sound synthesis challenge, including task definition, dataset, baseline, evaluation scheme and criteria, challenge result, and discussion.

***Index Terms*—** Generative models, DCASE, sound synthesis

1. INTRODUCTION

Recent years have seen remarkable progress in generative models, with applications in a variety of fields including image generation [1], text generation [2], music generation [3, 4, 5], and sound generation [6, 7]. Models like these are capable of generating high-quality and diverse samples, and have been widely adopted in both academia and industry. In particular, sound generation has gained increased attention in recent years, with advances in sound synthesis and generative models enabling the creation of realistic and diverse audio content.

Sound synthesis plays a crucial role in enhancing the auditory perception of multimedia content, such as movies, music, and videos. Automatic or machine-assisted Foley synthesis has the potential to greatly streamline the process of creating these sound effects, freeing up time and resources for multimedia content creators.

To encourage further research and development in the field of automatic Foley synthesis, we developed a challenge that aims to provide a standardized evaluation framework for different systems. Challenges have been shown to be an effective way to motivate the development of machine learning models, particularly in the early stages of a research area. We believe that this Foley sound synthesis challenge can play a critical role in advancing the state-of-the-art in automatic Foley synthesis. This challenge was held as part of the international Detection and Classification of Acoustic Scenes and Events 2023 Workshop. The topics discussed in this introduction are also covered in a proposal document [8].

2. PROBLEM AND TASK DEFINITION

We defined the problem of this challenge as ‘category-to-sound’ generation. The category is chosen in one of the selected seven categories - *dog bark*, *footstep*, *gunshot*, *keyboard*, *moving motor vehicle*, *rain*, and *sneeze/cough*. The sound is specified as a 4-second mono audio snippet with a sampling rate of 22,050 Hz.

As this was the first year of this challenge, we chose the input of the system to be a sound category rather than text input with natural language. This simplification was made to ease the organizing effort such as defining the problem and the evaluation scheme, collection of dataset, etc. We also intended this to lower the bar for participation, especially from academia, as category-based systems would require less data and computational resources than free text inputs. Similarly, limiting the problem to the seven categories clarified the subjective evaluation criteria. The seven categories were chosen so that i) the categories are useful for media creation, ii) it is feasible to collect a reasonable quantity of training/evaluation sounds with manual review, and iii) the generated sounds are easy to assess for the evaluators.

Despite this simplification, our intention for this challenge is to build towards generalizable and potentially useful approaches in the real world. In this regard, we specified the submitted systems should not simply copy-paste an existing sound, i.e., the systems should be generative, not retrieving.

Our goal is to motivate the development of new methods for Foley synthesis. Because the volume of data can be instrumental in qualitative improvements across many areas of ML, we created

---

[*] Equal contribution

two challenge tracks: one in which participants are free to augment their training data with external sources (Track A), and the other in which only the provided development dataset is allowed (Track B). To enhance the efficiency of the challenge, we also provided two pre-trained models, HiFi-GAN [9] and VQ-VAE [10], for Track B. These models were trained using the official dataset.

For a fair and correct evaluation, we required the participants to submit their model embedded in a Google Colab notebook template[1]. This provided an easy, familiar, and verifiable way for participants to share models while resolving any dependency issue for the organizers, at least within the time frame of the challenge.

3. OFFICIAL DATASET AND BASELINE

The development dataset used in this task consists of 6.1 hours of audio excerpts, each annotated with one of seven distinct sound classes: footstep, sneeze/cough, rain, dog bark, moving motor vehicle, gun shot, and keyboard. We selected the categories by considering an urban sound taxonomy [11]. The seven sound categories were selected evenly from each top-level group (‘human’, ‘nature’, ‘mechanical’), except for ‘music.’ There is no overlap in the low-level groups between the sound categories.

We collected the data from UrbanSound8K [11], FSD50K [12], and BBC Sound Effects.[2] To select the appropriate audio clips for our challenge, we followed a two-step process. First, we gathered audio samples that were annotated with labels closely related to one of the seven sound categories. Second, to ensure consistency in the challenge, we pre-processed the audio to mono 16-bit 22,050 Hz and either zero-padded or segmented it to a length of 4 seconds, a duration found sufficient for human recognition of class and audio quality. This pre-processing step was applied before selection, as the audio events comprise only a small portion of the total audio length.

To ensure the quality of the dataset, we carefully selected the audio clips for each category based on their relevance, variety, and clarity. One organizer manually selected the collection of excerpts, each of which was verified by a different organizer to ensure accuracy and clarity. Overall, we selected 5,550 labeled sound excerpts, with the number of sounds per category ranging from 681 to 900.

We divided the dataset into a development dataset and an evaluation dataset. Although the number of audio samples varies across sound classes, we ensured that the evaluation set had a consistent number of 100 audio samples per category. This decision was made to ensure that the evaluation set had a diverse range of sounds and was not too small. We also made sure that the partitions were stratified, so no source recording provided clips in both the development and evaluation sets, even if there were multiple excerpts from the same longer recording.

As a baseline system, we implemented a model [13] composed of three independently trained modules: PixelSNAIL [14], VQ-VAE [10], and HiFi-GAN [9]. The first module, PixelSNAIL, is an autoregressive model that maps a sound category input to a time-frequency representation. The second module, VQ-VAE, transforms the PixelSNAIL output into a Mel spectrogram through a compressed, latent vector encoding. The final module, HiFi-GAN, transforms the VQ-VAE output (Mel spectrogram) into a time-domain digital audio signal.

We selected the model as our baseline system for the following reasons. First, the modules were assigned the reconstruction task and the generation task separately, enhancing the whole architecture’s explainability. Second, the participants were allowed to reuse some of the modules. Since each module was trained independently, improving the performance of the system can be achieved by modifying the structure or scheme of specific modules while keeping the remaining modules unchanged.

4. EVALUATION

Even for objective tasks such as classification and detection tasks, it is challenging to provide unambiguous annotations and unbiased evaluation metrics. Multiple evaluation metrics may be necessary, but it can complicate the ranking of participants. [15]. With generative tasks such as the one considered in this challenge, the problem is even more difficult, as the produced data is not a set of labels, but audio, whose qualities must be assessed. This matter is far from being solved and is currently undergoing active research [16]. Recognizing this as a challenge, we opted for a pragmatic combination of objective and subjective evaluation protocols as proposed in [8].

In detail, we chose a two-step procedure. The first step considers objective metrics to get a first ranking of the proposed systems. Due to the constraints on human listening time for subjective ratings, in each track, only the top four entries were then considered for the second step with a subjective evaluation.

We decided to measure the following qualities:

1. **Perceptual Audio Quality**: The degree of clarity of sound, free from any artifacts, fuzziness, degradation, distortion, and noise.
2. **Fit-to-category**: The degree to which a sound is recognized as belonging in the intended category.
3. **Diversity**: The degree to which a system is able to produce a diverse set of sounds.

Evaluation of the above qualities typically involves high-level perceptual and cognitive processing by humans and thus cannot be evaluated by simple computational means. For this reason, we chose to complement the objective evaluation with subjective metrics. Although essential, subjective evaluation comes with some constraints. Humans can give different ratings depending upon the context of a sound they hear, and can experience fatigue. For the latter reason, only a subset of audio samples can be presented for subjective rating. To make the sure the context is similar across raters (and potentially, across future contests), the audio samples should include some “anchors,” i.e. sounds which clearly have a very low and/or high quality; anchors help to psychologically anchor the ratings and also serve as a check on the quality of the rater [17].

4.1. Step 1: Objective Evaluation

We adopted Fréchet Audio Distance (FAD) [18], a reference-free, lower-the-better, evaluation metric. FAD calculations were performed for each category. Systems were then ranked based on the average FAD across seven categories, and only the 4 top-performing systems per track were considered for the second step, due to time limitations of the subjective evaluation.

4.2. Step 2 : Subjective Evaluation

The subjective evaluation was operated in two steps. The first was an online survey that measured the fit-to-category and perceptual

---

[1] `https://colab.research.google.com`

[2] `https://sound-effects.bbcrewind.co.uk/`

audio quality. The fit-to-category asked the listener to use their general notion of the sound category and was not restricted to referencing the exact sounds in the development set, nor was it based on the number of sound events in a file. These tasks were performed on 20 sounds from each category, along with a set of anchors taken from the development set and baseline system. The second step was a measure of category diversity.

The selection of the 20 representative sounds was done as follows. OpenL3 embeddings of all the samples were computed and a k-means clustering with $k = 20$ was conducted on them [19]. The 20 "medoid" representative sounds are selected as the ones with the smallest Euclidean distance to the centroid in the embedding space.

After listening to each sound, the rater was asked to rate two scales to indicate both its perceptual audio quality and its fit-to-category, as defined in Section 4. For both scales, raters selected among 11 levels, with 0 being an unusable sound and 10 being the top of an absolute scale (the best possible, as opposed to the best of this contest). Re-listening to the sound was permitted. This procedure was more appropriate for category fit judgments than MUSHRA [17] because each sound was unique and different sounds could fit a category equally well.

Before rating a category, the rater listened to 6 representative sounds of the category from the development set. The high and low quality/fit anchor sounds, respectively, were hand-picked from the evaluation set and our baseline system. These sounds were not identified as anchors in the survey and were embedded in the main test at random locations. Entries from Track A and B were intermixed so that their relative quality would be apparent, even though the competition rankings are separated within each track. The order of trials was counterbalanced across test conditions.

4.3. Execution

All of the challenge participants performed the ratings on perceptual audio quality and fit-to-category for 4-7 categories, for a total duration of about 3-6 hours. After each category, the listener could take a break.

All participants listened to the same sounds. Thus, participants who submitted one of the finalist systems actually rated sounds from their own systems but their self-ratings were removed by the organizers before computing results. This allowed us to streamline the rating system while removing potential rating bias.

Rating at least 4 categories was required to be eligible for a prize. This requirement ensured that we had a fair distribution of teams doing ratings and enough ratings per sound. Additionally, some organizers rated sounds. This combined effort resulted in 10-15 independent ratings per sound. 93 separate category ratings were completed which took approximately 47 hours. Two of the 93 ratings were omitted at the start of the data analysis because they misrated 5 or more of the 12 quality-check trials (in both cases, giving a rating of low quality & good fit to an anchor sound that had a high quality & poor fit, indicating that they had confused the two scales). The anchors that had low quality tended to get a poor fit rating, so we did not use those as an exclusion criterion. Appropriate ratings were given for anchors that had high quality & low fit, high quality & good fit, and low quality & poor fit (4 of each type).

To validate the protocol as well as the software stack, a pilot study was carried out with the outputs of the baseline system in which the listeners were the organizers. During the evaluation phase, the test was advertised to relevant mailing lists. In this version, only one 30-minute category rating task was proposed to the listener, using a scheme to distribute the ratings across categories.

Finally, as our second step, we also performed a subjective test on Diversity. Diversity is a "set-based" quality, meaning that a set of generated audio files are mandatory for measuring it. For this reason, Diversity could not be evaluated within the above discussed listening test, whose stimuli are considered independently. For each system and each category, an organizer who did not participate in the ratings generated a continuous audio file sequencing the 20 representative sounds per system. Each file was given a name specifying the category and an obfuscated version of the system id. The diversity rating task took about 1.5 hours. Four other organizers, blind as to which systems they were rating, rated the diversity of the sounds per file from 0 (All the sounds appear to be identical) to 10 (Extremely large range of sounds).

Considering that 1) diversity may be less important than quality and fitness and 2) this quality has been not as rigorously tested in this edition of the challenge as the two other qualities, organizers decided in advance that the diversity ratings were weighted half as much as each of the audio quality and category fit ratings.

5. RESULTS

We provided a colab notebook as a starting point to implement submissions. We received 42 systems in total, including 11 systems in Track A [20, 21, 22, 23, 24] and 31 systems in Track B [25, 26, 27, 28, 29, 30, 31, 22, 32, 33, 34, 35, 36]. We removed disqualified submissions that failed to run on standard colab instances in a reasonable period of time. Before disqualification, we had a 4-day review period permitted for trivial bug fixes but did not allow changes in parameterization of the submitted systems.

With the remaining 36 working systems submitted by 17 teams, we generated 700 audio samples from 9 and 27 systems for tracks A and B, respectively. The audio samples are available online[3].

As all the scores (FAD scores per category, subjective test results on audio quality, fit-to-class, and diversity) were released on the DCASE official website,[4] we analyze the evaluation results in this section.

In Fig. 1, the FAD scores of 17 systems are plotted. The (x, y) position represents the average FAD score computed on the development set (FAD-Dev) and the evaluation set (FAD-Eval), respectively. The width and height of each rectangle represents the (scaled) standard deviation over 7 categories for both sets, respectively.

First, most of the systems show better (lower) FAD-Dev than FAD-Eval, with the exception of [20]. This is expected, as the training would be at least partially based on the development set. Second, it turns out that FAD-Dev is a noisy measure to predict FAD-Eval. This is not surprising as the final objective measure (FAD-Eval) contained new sounds to prevent overfitting. Third, comparing the top systems of track A and B, several systems in track B showed better performance on FAD-Dev, but not in FAD-Eval. This shows the difficulty of training a system with the limited amount of data permitted in track B.

In Fig. 2, the top 8 systems and the baseline system are plotted by their final ranking determined by a listening test as well as FAD-Eval and FAD-Dev. On the left, the scatterplot shows the importance of subjective tests. The Spearman's rank correlation coefficient of the ranking by FAD-Eval and the final ranking is only

---

[3] `https://zenodo.org/record/8091972`

[4] `https://dcase.community/challenge2023/task-foley-sound-synthesis-results`

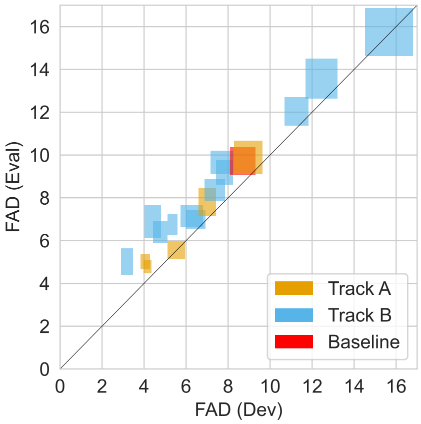


Figure 1: FAD Scores on the development set vs the evaluation set, computed on the 17 submitted systems and the baseline system.

‘0.238’. On the right, with FAD-Dev, the coefficient is somewhat higher, ‘0.524’.

We established that subjective perceptual sound qualities were not entirely predicted by objective FAD scores. In addition, we established that the three perceptual metrics were interrelated, but each had a unique contribution. Within each category, the correlations between average rating scores of finalist systems of audio quality and category fit were very strong (average across all categories was $r = 0.98$); however, when quality & fit ratings from individual trials were correlated within each category, the average correlation was less extreme ($r = .75$), showing that raters were not giving identical answers to both questions. Our anchor trials showed that the raters did know how to distinguish the two qualities, because they appropriately rated the category-inappropriate sounds with good audio quality. On the other hand, we also found that raters gave all-around low ratings to the category-appropriate sounds with poor audio quality. Because sound recognition was essential for judging category fit, it is plausible that good audio quality was required before being able to give a high category fit rating. Average diversity (within each category, across finalist systems) had a strong relationship to category fit ($r = 0.70$); nonetheless, half of the variance in diversity ratings was independent of quality/fit.

The perceptual ratings of the quality/fit of all the systems were plausible, with the highest average ratings obtained for the sounds from the development set, and the lowest for our baseline system. The submitted systems had intermediate ratings, showing that there is room for improvement in this challenge.

To summarize, there are expected mismatches between the objective evaluation for the provided sounds (FAD-Dev) and the sounds held back by the organizers (FAD-Eval); importantly, objective evaluation metrics did not completely align with subjective evaluation (final ranking). This justifies two of our choices for the evaluation scheme: i) receive submissions in the form of a system (code) instead of sounds, and ii) run a subjective evaluation.

6. CONCLUSION

In this paper, we presented a challenge for automatic Foley sound synthesis aimed at promoting further research and development in generative AI for sound. We have provided a detailed overview of the challenge, including task definition, dataset requirements, evaluation criteria, a baseline implementation, and analysis of the results. Through this challenge, we believe we have achieved our goal — to encourage active participation from the research community and advance the state-of-the-art in automatic Foley synthesis. Although it was the first year of the challenge, we received substantial submissions in both of the tracks. We also performed the generation and evaluation of the submitted systems successfully.

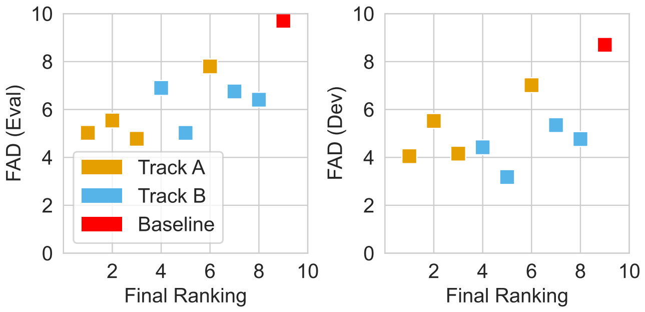


Figure 2: FAD scores on the development set and the evaluation set vs. the final ranking determined by the listening tests.

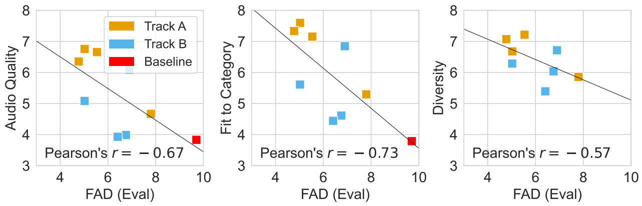


Figure 3: Relationships between objective measure (FAD-Eval) and subjective tests.

In both tracks, the best performing systems were based on deep learning, with a sequence of a diffusion model for spectrogram generation and HIFI-Gan [9] for phase reconstruction.

There have been difficulties as well. Our analysis showed the necessity of performing a subjective evaluation and running inference by ourselves. Unfortunately, both are costly; in total, about 47 hours were spent for the evaluation of 8 systems and about 471 A100 GPU hours for the inference. With permission, we released all the generated sounds as well as their subjective/objective scores on Zenodo, hoping to enable more analysis and even subjective quality prediction models based on the data.

In the future, we hope that the standardized evaluation framework provided by this challenge will help to facilitate comparisons between different Foley synthesis systems. It is already apparent that more sophisticated Foley sound synthesis will be possible in the near future with text-input, video-input, etc. We hope our challenge will ultimately lead to the development of more effective and efficient techniques.

7. ACKNOWLEDGEMENTS

Computing resources for sound generation were covered by Gaudio Lab, Inc. We thank all the raters who did the subjective evaluation.

8. REFERENCES

[1] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever, "Zero-shot text-to-image generation," in *International Conference on Machine Learning*. PMLR, 2021, pp. 8821–8831.

[2] OpenAI, "GPT-4 technical report," *arXiv preprint arXiv:2303.08774*, 2023.

[3] M. Pasini and J. Schlüter, "Musika! fast infinite waveform music generation," in *ISMIR 2022 Hybrid Conference*, 2022.

[4] A. Agostinelli, T. I. Denk, Z. Borsos, J. Engel, M. Verzetti, A. Caillon, Q. Huang, A. Jansen, A. Roberts, M. Tagliasacchi, *et al.*, "MusicLM: Generating music from text," *arXiv preprint arXiv:2301.11325*, 2023.

[5] C. Donahue, J. McAuley, and M. Puckette, "Adversarial audio synthesis," in *International Conference on Learning Representations (ICLR)*, 2019.

[6] Z. Borsos, R. Marinier, D. Vincent, E. Kharitonov, O. Pietquin, M. Sharifi, D. Roblek, O. Teboul, D. Grangier, M. Tagliasacchi, *et al.*, "Audiolm: a language modeling approach to audio generation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2023.

[7] H. Liu, Z. Chen, Y. Yuan, X. Mei, X. Liu, D. Mandic, W. Wang, and M. D. Plumbley, "AudioLDM: Text-to-audio generation with latent diffusion models," *arXiv preprint arXiv:2301.12503*, 2023.

[8] K. Choi, S. Oh, M. Kang, and B. McFee, "A proposal for foley sound synthesis challenge," *arXiv preprint arXiv:2207.10760*, 2022.

[9] J. Kong, J. Kim, and J. Bae, "HiFi-GAN: Generative adversarial networks for efficient and high fidelity speech synthesis," *Advances in Neural Information Processing Systems*, vol. 33, pp. 17 022–17 033, 2020.

[10] A. Van Den Oord, O. Vinyals, *et al.*, "Neural discrete representation learning," *Advances in neural information processing systems*, vol. 30, 2017.

[11] J. Salamon, C. Jacoby, and J. P. Bello, "A dataset and taxonomy for urban sound research," in *Proceedings of the 22nd ACM international conference on Multimedia*, 2014.

[12] E. Fonseca, X. Favory, J. Pons, F. Font, and X. Serra, "Fsd50k: an open dataset of human-labeled sound events," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 829–852, 2021.

[13] X. Liu, T. Iqbal, J. Zhao, Q. Huang, M. D. Plumbley, and W. Wang, "Conditional sound generation using neural discrete time-frequency representation learning," in *2021 IEEE 31st International Workshop on Machine Learning for Signal Processing (MLSP)*. IEEE, 2021, pp. 1–6.

[14] X. Chen, N. Mishra, M. Rohaninejad, and P. Abbeel, "Pixelsnail: An improved autoregressive generative model," in *International Conference on Machine Learning*. PMLR, 2018.

[15] A. Mesaros, T. Heittola, and T. Virtanen, "Metrics for polyphonic sound event detection," *Applied Sciences*, 2016.

[16] J. Vadillo and R. Santana, "On the human evaluation of universal audio adversarial perturbations," *Computers & Security*, 2022.

[17] B. Series, "Method for the subjective assessment of intermediate quality level of audio systems," *International Telecommunication Union Radiocommunication Assembly*, 2014.

[18] K. Kilgour, M. Zuluaga, D. Roblek, and M. Sharifi, "Fréchet audio distance: A reference-free metric for evaluating music enhancement algorithms." in *INTERSPEECH*, 2019.

[19] A. L. Cramer, H.-H. Wu, J. Salamon, and J. P. Bello, "Look, listen, and learn more: Design choices for deep audio embeddings," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 3852–3856.

[20] M. Kang, S. Oh, H. Moon, K. Lee, and B. S. Chon, "FALL-E: Gaudio foley synthesis system," Tech. Rep., June 2023.

[21] S. Fan, Q. Zhu, F. Xiao, H. Lan, W. Wang, and J. Guan1, "Foley sound synthesis with AudioLDM for dcase2023 task 7," Tech. Rep., June 2023.

[22] J. Lee1, H. Nam, and Y.-H. Park, "VIFS: An end-to-end variational inference for foley sound synthesis," Tech. Rep., 2023.

[23] R. Scheibler, T. Hasumi, Y. Fujita, T. Komatsu, R. Yamamoto, and K. Tachibana, "Class-conditioned latent diffusion model for DCASE 2023 foley sound synthesis challenge," Tech. Rep., 2023.

[24] Y. Yuan, H. Liu, X. Liu, X. Kang, M. D.Plumbley, and W. Wang, "Latent diffusion model based foley sound generation system for dcase challenge 2023 task 7," June 2023.

[25] S. Huang, J. Bai, Y. Jia, and J. Chen, "Jless submission to dcase2023 task7: Foley sound synthesis using non-autoagressive generative model," Tech. Rep., June 2023.

[26] W.-G. Choi and J.-H. Chang, "HYU submission for the dcase 2023 task 7: Diffusion probabilistic model with adversarial training for foley sound synthesis," Tech. Rep., June 2023.

[27] C.-W. Bang, N. K. Kim, and C. Chun, "High-quality foley sound synthesis using monte carlo dropout," Tech. Rep., 2023.

[28] Y. Chung, J. Lee, and J. Nam, "Foley sound synthesis in waveform domain with diffusion model," Tech. Rep., 2023.

[29] H. C. Chung, Y. Lee, and J. H. Jung, "Foley sound synthesis based on GAN using contrastive learning without label information," Tech. Rep., June 2023.

[30] P. Kamath, T. N. Islam, C. Gupta, L. Wyse, and S. Nanayakkara, "Dcase task-7: StyleGAN2-based foley sound synthesis," Tech. Rep., June 2023.

[31] K. Kim, J. Lee, H. Kim, and K. Lee, "Conditional foley sound synthesis with limited data: Two-stage data augmentation approach with stylegan2-ada," Tech. Rep., June 2023.

[32] A. Pillay, S. Betko, A. Liloia, H. Chen, and A. Shah, "DCASE task 7: Foley sound synthesis," Tech. Rep., June 2023.

[33] A. Qi, "Auto-bit for DCASE2023 task7 technical reports: Assemble system of bitdiffusion and PixelSNAIL," 2023.

[34] H. Zhang, K. Qian, L. Shen, L. Li, K. Xu, and B. Hu, "From noise to sound: Audio synthesis via diffusion models," 2023.

[35] T. Wendner, P. Hu, T. Jadidi, and A. Neuhauser, "Audio diffusion for foley sound synthesis," Tech. Rep., June 2023.

[36] Z. Xie, X. Xu, B. Li, M. Wu, and K. Yu, "The X-LANCE system for DCASE2023 challenge task 7: Foley sound synthesis track b," Tech. Rep., June 2023.

STELIN-US: A SPATIO-TEMPORALLY LINKED NEIGHBORHOOD URBAN SOUND DATABASE

Snehit Chunarkar, Bo-Hao Su, Chi-Chun Lee

Department of Electrical Engineering, National Tsing Hua University, Taiwan,
snehit@gapp.nthu.edu.tw, borrissu@gapp.nthu.edu.tw, cclee@ee.nthu.edu.tw

ABSTRACT

Automated acoustic understanding, e.g., sound event detection and acoustic scene recognition, is an important research direction enabling numerous modern technologies. Although there is a wealth of corpora, most, if not all, include acoustic samples of scenes/events in isolation without considering their interconnectivity with locations nearby in a neighborhood. Within a connected neighborhood, the temporal continuity and regional limitation (sound-location dependency) at distinct locations creates non-iid acoustics samples at each site across spatial-temporal dimensions. To our best knowledge, none of the previous data sources takes on this particular angle. In this work, we present a novel dataset, the Spatio-temporally Linked Neighborhood Urban Sound (STeLiN-US) database. The dataset is semi-synthesized, that is, each sample is generated by leveraging diverse sets of real urban sounds with crawled information of real-world user behaviors over time. This method helps create a realistic large-scale dataset, and we further evaluate it through perceptual listening tests. This neighborhood-based data generation opens up novel opportunities to advance user-centered applications with automated acoustic understanding. For example, to develop real-world technology to model a user's speech data over a day, one can imagine utilizing this dataset as the user's speech samples would modulate by diverse sources of acoustics surrounding linked across sites and temporally by natural behavior dynamics at each location over time.

***Index Terms*—** Audio Dataset, Sound Synthesis, Urban Sound, Connected

1. INTRODUCTION

Understanding acoustic surroundings seamlessly influences our daily life, e.g., recognizing different emergencies by distinct alerting sounds. Besides, acoustic sounds also affect human mental health, e.g., work productivity in a calm/noisy environment [1], and psychological impact on our well-being as the change in stress level [2]. Thus, understanding acoustic sounds plays a crucial role in our life, which provides plentiful information to uplift environmental awareness and life quality. Especially recent advanced techniques and the support of superior hardware in deep learning show a prominent performance on these acoustic contextual tasks.

Basically, these acoustic context tasks can be generally divided into two categories, which are sound event detection (SED) and acoustic scene classification. Specifically, a sound event detection task aims to predict a short-term and precise event, e.g., a dog barking, a car passing by, or a cell phone ringing. Differing from that, an acoustic scene classification task targets an environment-wise contextualization, e.g., on the street and in a coffee shop, which may compound multiple sound events. Recently, for sound event detection tasks, Turpault in [3] proposed to use weakly labeled data where a top-performed system using a convolutional neural network (CNN) model has achieved 42.7% F-measure. Besides, Ronchini et al. [4] integrated non-target events as auxiliary information while training and greatly impacted the SED task. As for acoustic scene classifications, DCASE has been predominantly focusing on scene classification in DCASE challenge Task 1 [5, 6, 7, 8, 9, 10] with constantly evolving their scope of interest within the task. Recently, they have been curious about the scope of this task on low-complexity approach [11] solutions, in which the top system competed with 48 submissions from 19 teams in the challenge and obtained 59.6% accuracy with 1.091 log loss. Both event detection and scene classification tasks manifest great accuracy in understanding the acoustic scenes/events with deep-learning-based models and provide insights for real-world applications.

However, most of them focus solely on scenes and events only. The currently published datasets used for similar tasks; only contain short-term audio from random locations and times isolatedly. None of them consider the inter-connectivity with locations nearby in a neighborhood. For instance, TAU Urban Acoustic Scenes 2020 Mobile [9] is one such designed for the scene classification task, but it lacks consistency in connectivity with its context of surrounding. The UrbanSound dataset [12] presents sound events compound with scrapped urban noises from the internet, which makes it diversely localized but poorly inter-linked. URBAN-SED dataset [13] having 11 events is a synthesized dataset aimed to compensate the sparsity of strongly annotated datasets; however, the same Brownian noise as background for all soundscape with predefined artificial synthesis settings barely justifies the real acoustic variation in an urban surrounding. ESC50 [14] with recordings in 2000 short clips emerge as one of the highest labeled environmental recording datasets bringing distinct 50 classes. Highlighting isolated high-quality sound events, the NIGENS dataset [15] brings 14 distinct sound event classes, including strong annotations. Whereas both ESC50 [14] and NIGENS [15] datasets are designed for SED tasks without the context of surrounding. SINS dataset [16] equipped with 16 activities aimed at activity detection in domestic environments for smart home applications. STARSS22 dataset [17] contains spatial recordings of real sound scenes collected in interiors, including temporal and spatial annotation of 13 sound events. However, both SINS [16] and STARSS22 [17] datasets sound recordings only in the interiors, which limits prominent datasets for diversity in the applications. Unlike the above-mentioned datasets, SONYC-UST [18] has attempted to build a dataset equipped with spatiotemporal metadata. The dataset contains real-world recordings in New York City with annotations defined using 23 tags based on New York City noise code. The highlight of SONYC-UST is the spatiotemporal context information that comforts monitoring the distribution of sound tags. But primarily focused on the events considered to be noise in the urban environment, eliminating common sound events (e.g., Birds chirping) which are not considered noise. Also, the recordings are intuitively at outdoor locations limiting the SED applications to deal with the events in outdoor scenarios. Alternatively, in this work, we bring a new perspective/angle to this field; we consider an application-wise scenario that can be applied in a user-contextualized, environment-aware closer to our daily life. That is, many applications incorporating speech are published as well, e.g., Speech Enhancement applications, Automatic Speech Recognition (ASR) applications, and Speech Separation Tasks. To mention, an unsupervised federated learning approach proposed by [19] for speech enhancement and separation with a release of LibriFSD50K dataset. And Darius Petermann et al. in [20] introduce the separation of an audio mixture into speech, music, and sound effects using their proposed dataset named Divide and Remaster. However, they integrate acoustic scenes/events into speech but in a non-realistic and artificial manner. Whereas to do so, continuous recording from real-life scenes is required, and even with the recordings subsequently, it needs to be annotated for the event's presence to be useful for SED tasks. Nevertheless, collecting new and large-scale recordings from the real world and annotating them is expensive, cumbersome, and time-consuming. Synthesis becomes a more feasible way to catch the scalability of existing speech datasets.

Hence, being a preliminary study to implement this idea, we develop a framework for synthesizing a continuous real-world acoustic distributed sound surrounding. Henceforth, we proposed this dataset with the inspiration to equip researchers with variable surrounding sound in an environment closely resembles realistic patterns. The proposed dataset models the small-scale connected surrounding in urban areas. The detail of the work is organized as follows: Section 2 presents the details of the synthesis, Section 3 summarizes the dataset and presents the analysis of the same, and with an end note, Section 4 discuss the dynamic scaling of the dataset with potential applications and concludes the present work.

2. METHODOLOGY

The proposed dataset is synthesized to represent a small-scale interconnected urban area. The synthesis framework is divided into Preconditions, Traffic, and Scene Synthesis. Here Preconditions deal with the requirements for the synthesis, Whereas the synthesis part is broadly divided into Traffic and Scene Synthesis.

2.1. Preconditions

Being an interconnected urban sound database, it is important to map the locations and patterns for scene-specific sound classes to conceptualize. Hence we presented a map in Fig. 1 for the proposed dataset; mapping both indoor and outdoor environments, 5 distinct locations were selected for synthesis representing a small-scale interconnected urban area. Street, Metro Station, Park, School Playground, and Cafe, represented by microphones M1, M2, M3, M4, and M5, respectively, are simulated with 14 acoustic sound classes. Of all classes, 6 represented the background, and 8 were the events. Vehicle, Train, Pedestrian, Cafe Crowd, Children Playing, Urban Park, Street Music, Phone Ring, School Bell, Car Horn, River, Bird, Fountain, and Dog Bark are considered acoustic sound classes. Train, Pedestrian, Cafe Crowd, Urban Park, River, and Fountain are considered as background, and the rest are the events. After a thorough review, the sound recordings for mentioned classes are adopted from a suitable published dataset, as in Table 1. At the same time, the pattern for the appearance of the sound classes Vehicle, Car Horn, Street Music, Pedestrian, and Dog Bark is inspired by the annotation from real-world distribution of the closely relevant events from the SONYC [18]. And as for the background sound at the synthesized locations Metro Station, Park, and Cafe follows the google maps popular time index using LivePopularTimes[1] python package for the respective sound class. Specifically, searching nearby Manhattan, e.g., "subway in Manhattan" prompt shows 18 results with the popular time index, which indicates the people's traffic at that location. That helps relate to the density of background sound of the location, and taking the average for the number of results gives a general idea about the trend of busyness. Since the SONYC [18] data is mainly concentrated around Manhattan, searching for google maps popular times around that area makes the distribution consistent with the base area. This distribution is obtained for a week in an hourly fashion, which makes it convenient to design the density of events or the crowded nature of the background in a similar fashion.

2.2. Traffic Synthesis

Temporal connection across microphone locations is shaped by Traffic Synthesis. Autonomous from overall Scene Synthesis, Traffic Synthesis synthesizes a controlled flow of vehicles by tracing each vehicle's course with calculated time for the appearance of the same vehicle at another microphone that comes under the vehicle's track. There are 4 entry nodes considered for each vehicle to enter the environment as EN1-EN4. Now considering the more or less busy route, the path for the vehicle is decided with a random distribution till it exits the environment at the diagonally opposite node to its entry node. IDMT dataset [21] enriched with 4 different vehicle sounds at 3 different known speeds is best suited for Traffic Synthesis. Since the map in Fig. 1 is conceived with an approximated distance for microphone locations, hence compiling the information of speed, a good approximation for the timing is achieved by using $Speed = \frac{Distance}{Time}$. Following this set of conditions has equipped us with a temporal correlation across the microphone locations.

2.3. Scene Synthesis

Audio at 5 different microphone locations is synthesized to assemble a scene that furnishes realistic event patterns with the temporal connection. A brief overview of the acoustic classes at each synthesized location is given in Table 2. Following a realistic distribution described in 2.1, the dense nature of the environment is compiled in the synthesis by adding more audio segments on top of the same

Table 1: Sound Classes and Dataset used for the synthesis

Sound Class	Source Dataset
Vehicle	IDMT Traffic [21]
Train, Cafe Crowd, Urban Park	TUT Rare Sound Events 2017 [22]
Pedestrian	TAU Urban Acoustic Scenes 2020 Mobile [9]
Children Playing, Street Music	UrbanSound [12]
Phone Ring	NIGENS [15]
School Bell, River, Fountain	FreeSound.org
Car Horn, Dog Bark	UrbanSound8K [12]
Bird	ESC-50 [14]

---

[1] `https://github.com/GrocerCheck/LivePopularTimes.git`

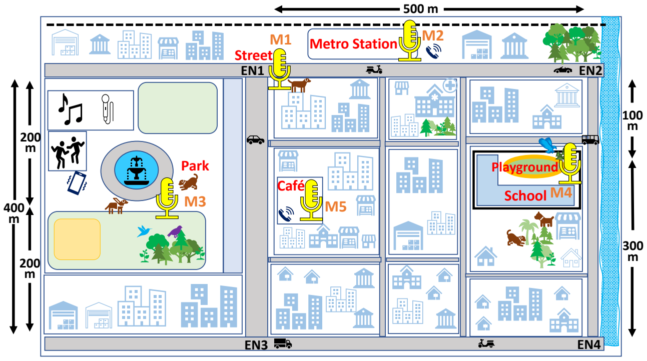


Figure 1: Acoustic Synthesis Map

sound class. A maximum of 3 audio segments have been added to represent the maximum dense structure. For the direct correlation of which popularity index from Google map has scaled down and quantized to an integer value to result in a range of 0-3. The dense scaling factor represents how many same-class audio segments to add for the background. On the other hand, the same factor for the event is inspired by a scaled version of SONYC [18] data for visualization of the event patterns, which indicates the number of sound events from the same classes added. However, events that are outside the SONYC [18] study are designed manually, e.g., Children playing and School bell class distribution in the School Playground microphone location (M4) are designed manually by considering school operating hours. The different sound classes merged to create a scene are scaled with the different intensities which is inspired by the inverse distance relationship with the sound intensity as in eq.(1). Whereas I_1, I_2 are the original and synthesis sound intensities, and d_1, d_2 are the respective distance of the recording from the source. Since the sounds taken from datasets do not contain the information regarding the distance of recording in detail, hence the chosen factor d_2 is scaled in terms of distance d_1 and then verified by manual listening for any resulting scaling change required. The data shown in Table 3 indicate the distance scale for the particular sound class used in synthesis, e.g., intensity scaling factor 2 indicates the audio event or background sound in synthesis audio will be twice as distance with respect to the one in the raw sound itself.

$$I_2 = I_1 \left(\frac{d_1}{d_2}\right)^2, \tag{1}$$

Overall, in the end, all the considered sound classes, after going through dense scaling and distance scaling processes, are added with each other to synthesize the scenario, which has the temporal pattern and interconnection with locations. Hence equipped us with one of its kind acoustic dataset designed to simulate the closely connected neighborhood urban area.

Table 2: Combination of sound classes present at different locations throughout the week.

Location	Day	Sound Classes
Street	Mon-Sun	Vehicle, Pedestrian, Phone Ring, Car Horn, Dog Bark
Metro Station	Mon-Sun	Train, Pedestrian, Phone Ring
Park	Mon-Sun	Vehicle, Pedestrian, Urban Park, Street Music, Phone Ring, Car Horn, Bird, Fountain, Dog Bark
School Play-ground	Mon-Sun	Vehicle, Pedestrian, Children Playing, School Bell, Car Horn, River, Bird, Dog Bark
	Sat - Sun	Vehicle, Car Horn, River, Bird, Dog Bark
Cafe	Mon-Sun	Vehicle, Cafe crowd, Phone Ring, Car Horn

3. EXPERIMENTAL RESULTS

A brief assessment of the proposed dataset is presented, divided into a summary and analysis of the dataset. In the following sections, we discuss the summary and distribution of STeLiN-US further; we analyze it from a visual and human listener's perspective.

3.1. Summary

Following our proposed semi-synthesis procedures, we generate a Spatio-temporally Linked Neighborhood Urban Sound (STeLiN-US) database and is made available online[2]. Containing interconnected acoustic surroundings and scene-specific events, the proposed dataset is equipped with 525 audio clips comprising 43 hr 45 min in total. Synthesized for location-specific scene surrounding and adjunct with strong annotations for the events have reinforced the proposed dataset to be equivalently used in both scene classification and event detection tasks. Besides, embedding the time and day information with the synthesized acoustic scene has lifted the applicability from traditional tasks.

3.2. Analysis

3.2.1. Dataset Distribution

To visualize the distribution of each sound class in the final synthesis with respect to synthesized microphone locations, a series of bar

Table 3: Scaling d2 = k*d1, considered k values for respective locations and sound class, where d1, d2 are the respective distance of the recording from the source.

	Locations				
Sound Class	Street	M.Station	Park	School-P.G.	Cafe
Vehicle	2	-	4	5	5
Train	-	1	-	-	-
Pedestrian	1	1	3	4	-
Cafe Crowd	-	-	-	-	0.5
Children Playing	-	-	-	3	-
Urban Park	-	-	1	-	-
Street Music	-	-	3	-	-
Phone Ring	9	9	9	-	15
School Bell	-	-	-	3	-
Car Horn	3	-	5	6	6
River	-	-	-	4	-
Bird	-	-	2	3	-
Fountain	-	-	2	-	-
Dog Bark	4	-	2	5	-

---

[2] `https://doi.org/10.5281/zenodo.8241539`

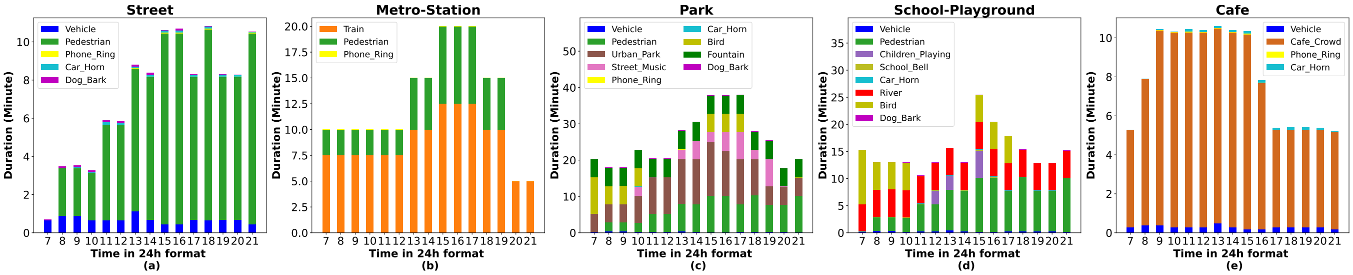


Figure 2: Average distribution of selected sound class in each synthesized location.

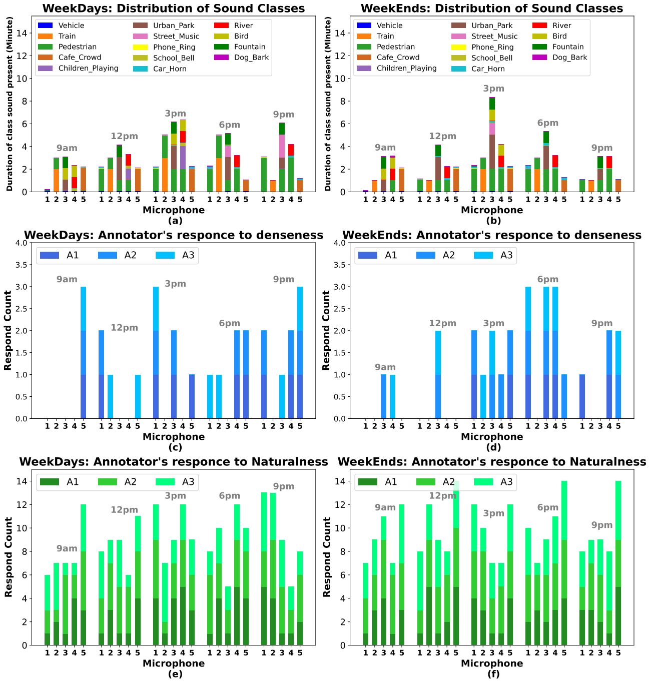


Figure 3: Sound class distribution at each synthesized location on (a) WeekDays (b) WeekEnds with corresponding annotators respond for dense/ sparseness in (c) and (d) and equivalent respond for naturalness in (e) and (f)

graphs are presented in Fig. 2. Following real environment distribution in sound classes, it showcases the distributed imbalance across time, location, and sound class e.g., from Fig. 2 (a), the Street is less busy during the morning than in late afternoon similarly in (e) cafe is busier from morning to afternoon than in late evenings, which are indeed the case in real life. Analogously, Fig. 2 can be compared with Table 2, which explains the presence of each class at synthesized microphone locations.

3.2.2. *Listening Test*

To validate the naturalness and sparseness of the proposed STeLiN-US, we further conducted a listening test by human annotation. In this experiment, a total of 50 audio samples (50 minutes in total) are selected randomly from STeLiN-US but evenly distributed in all synthesized microphone locations and times for this test. Precisely, all locations and time slots should present at least one time in the listening test set. During the listening test, we define two questions for annotators, including naturalness and sparseness. Naturalness is annotated on a 5-Likert scale, where 1 represents strong disagreement on the naturalness of audio (i.e., the audio sounds artificial), and 5 means strong agreement on naturalness (i.e., the audio sounds natural). Similarly, sparseness is labeled by asking whether the audio sounds in rush hour, which is a binary(yes/no) question for them, and 0 for sparse, 1 for dense. In the overall listening test, we include 6 unique annotators (2 females, and 4 males) in total.

Henceforth, to analyze the distribution in a systematic way, we divide them into weekdays and weekends, as shown in Fig. 3 (a) & (b), respectively. Fig. 3 (c) & (d) represent the Dense/Sparseness results from three annotators (A1, A2, and A3) divided into weekdays and weekends, respectively. Similarly, Fig. 3 (e) & (f) is for naturalness result. To have statistical results, we further compute the average annotation among all the annotators and present their standard deviation as well. Notably, we get 0.36 and 3.12 average results for dense and naturalness, respectively, and similarly, we get 0.22 and 0.91 standard deviations. This conveys annotators agree closely for dense and naturalness results with lower deviation at the same time. It is amazing to observe the average naturalness result is more than half of the max on the scale with the least deviation among annotators depicting that even if the dataset is synthesized one is still inclined to feel natural alike.

4. DISCUSSION AND CONCLUSION

The proposed synthesis approach cultivated with real-world user behavior can be dynamically scaled to model any required environment. Such wide adaptability can elevate application-specific research solutions. Furnished with the real surrounding pattern distribution of sound classes, the proposed STeLiN-US dataset simulates the acoustic appearance of closely interconnected neighborhoods in urban areas. This help in not only identifying the scenes but also predicting acoustic scenarios. This accommodates the user-centered applications, e.g., If combined with the ASR, the ASR performance can be analyzed based on the location and time more than that possible performance can be predicted beforehand based on the prediction of the scene busyness. Hence this dataset can unveil many possible applications for the researcher. In contrast with previously published datasets, portraying diversity across locations yet interconnected and diverse events to truly justify the surrounding environment and still sound natural alike from the listening test has made the proposed dataset unique and unmatched. Such incorporation of scene-specific events to replicate the real surrounding environments facilitates researchers in testing trailblazing event detection systems.

5. ACKNOWLEDGMENT

The work was supported by the National Science and Technology Council (NSTC), Taiwan.

6. REFERENCES

[1] J. Lim, K. Kweon, H. W. Kim, S. W. Cho, J. Park, and C. S. Sim, "Negative impact of noise and noise sensitivity on mental health in childhood," *Noise Health*, vol. 20, no. 96, pp. 199–211, 2018.

[2] L. I. Yankoty, P. Gamache, C. Plante, S. Goudreau, C. Blais, S. Perron, M. Fournier, M. S. Ragettli, M. Hatzopoulou, Y. Liu, and A. Smargiassi, "Relationships between long-term residential exposure to total environmental noise and stroke incidence," *Noise Health*, vol. 24, no. 113, pp. 33–39, 2022.

[3] N. Turpault, R. Serizel, A. Shah, and J. Salamon, "Sound event detection in domestic environments with weakly labeled data and soundscape synthesis," in *Acoustic Scenes and Events 2019 Workshop (DCASE2019)*, 2019, p. 253.

[4] F. Ronchini, R. Serizel, N. Turpault, and S. Cornell, "The impact of non-target events in synthetic soundscapes for sound event detection," in *Proceedings of the 6th Detection and Classification of Acoustic Scenes and Events 2021 Workshop (DCASE2021)*, Barcelona, Spain, November 2021, pp. 115–119.

[5] D. Stowell, D. Giannoulis, E. Benetos, M. Lagrange, and M. D. Plumbley, "Detection and classification of acoustic scenes and events," *IEEE Transactions on Multimedia*, vol. 17, no. 10, pp. 1733–1746, Oct 2015.

[6] A. Mesaros, T. Heittola, and T. Virtanen, "Assessment of human and machine performance in acoustic scene classification: Dcase 2016 case study," in *2017 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2017, pp. 319–323.

[7] ——, "Acoustic scene classification: An overview of dcase 2017 challenge entries," in *2018 16th International Workshop on Acoustic Signal Enhancement (IWAENC)*, 2018, pp. 411–415.

[8] ——, "A multi-device dataset for urban acoustic scene classification," in *Scenes and Events 2018 Workshop (DCASE2018)*, p. 9.

[9] T. Heittola, A. Mesaros, and T. Virtanen, "Acoustic scene classification in dcase 2020 challenge: generalization across devices and low complexity solutions," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2020 Workshop (DCASE2020)*, 2020, pp. 56–60. [Online]. Available: https://arxiv.org/abs/2005.14623

[10] I. Martín-Morató, T. Heittola, A. Mesaros, and T. Virtanen, "Low-complexity acoustic scene classification for multi-device audio: analysis of dcase 2021 challenge systems," 2021.

[11] I. Martín-Morató, F. Paissan, A. Ancilotto, T. Heittola, A. Mesaros, E. Farella, A. Brutti, and T. Virtanen, "Low-complexity acoustic scene classification in dcase 2022 challenge," 2022. [Online]. Available: https://arxiv.org/abs/2206.03835

[12] J. Salamon, C. Jacoby, and J. P. Bello, "A dataset and taxonomy for urban sound research," in *22nd ACM International Conference on Multimedia (ACM-MM'14)*, Orlando, FL, USA, Nov. 2014, pp. 1041–1044.

[13] J. Salamon, D. MacConnell, M. Cartwright, P. Li, and J. P. Bello, "Scaper: A library for soundscape synthesis and augmentation," in *2017 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2017, pp. 344–348.

[14] K. J. Piczak, "Esc: Dataset for environmental sound classification," in *Proceedings of the 23rd ACM International Conference on Multimedia*, ser. MM '15. New York, NY, USA: Association for Computing Machinery, 2015, p. 1015–1018. [Online]. Available: https://doi.org/10.1145/2733373.2806390

[15] I. Trowitzsch, J. Taghia, Y. Kashef, and K. Obermayer, "The nigens general sound events database," 2020.

[16] G. Dekkers, S. Lauwereins, B. Thoen, M. W. Adhana, H. Brouckxon, T. van Waterschoot, B. Vanrumste, M. Verhelst, and P. Karsmakers, "The SINS database for detection of daily activities in a home environment using an acoustic sensor network," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2017 Workshop (DCASE2017)*, November 2017, pp. 32–36.

[17] A. Politis, K. Shimada, P. Sudarsanam, S. Adavanne, D. Krause, Y. Koyama, N. Takahashi, S. Takahashi, Y. Mitsufuji, and T. Virtanen, "STARSS22: A dataset of spatial recordings of real scenes with spatiotemporal annotations of sound events," in *Proceedings of the 8th Detection and Classification of Acoustic Scenes and Events 2022 Workshop (DCASE2022)*, Nancy, France, November 2022, pp. 125–129. [Online]. Available: https://dcase.community/workshop2022/proceedings

[18] M. Cartwright, A. E. M. Mendez, J. Cramer, V. Lostanlen, G. Dove, H.-H. Wu, J. Salamon, O. Nov, and J. Bello, "SONYC urban sound tagging (SONYC-UST): A multilabel dataset from an urban acoustic sensor network," in *Proceedings of the Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE)*, October 2019, pp. 35–39. [Online]. Available: http://dcase.community/documents/workshop2019/proceedings/DCASE2019Workshop_Cartwright_4.pdf

[19] E. Tzinis, J. Casebeer, Z. Wang, and P. Smaragdis, "Separate but together: Unsupervised federated learning for speech enhancement from non-IID data," in *2021 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. IEEE, oct 2021. [Online]. Available: https://doi.org/10.1109%2Fwaspaa52581.2021.9632783

[20] D. Petermann, G. Wichern, Z.-Q. Wang, and J. L. Roux, "The cocktail fork problem: Three-stem audio separation for real-world soundtracks," 2022.

[21] J. Abeßer, S. Gourishetti, A. Kátai, T. Clauß, P. Sharma, and J. Liebetrau, "Idmt-traffic: An open benchmark dataset for acoustic traffic monitoring research," 2021.

[22] A. Mesaros, T. Heittola, A. Diment, B. Elizalde, A. Shah, E. Vincent, B. Raj, and T. Virtanen, "DCASE 2017 challenge setup: Tasks, datasets and baseline system," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2017 Workshop (DCASE2017)*, November 2017, pp. 85–92.

FOLEY SOUND SYNTHESIS BASED ON GENERATIVE ADVERSARIAL NETWORKS USING ONESELF-CONDITIONED CONTRASTIVE LEARNING

HaeChun Chung, Yuna Lee, JaeHoon Jung

AI2XL Lab.,
Institute of Convergence Technology,
KT Corporation

ABSTRACT

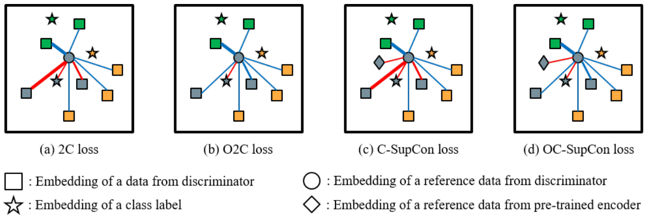
The creation of sound effects, such as foley sounds, for radio or film has traditionally relied on the expertise of skilled professionals. However, synthesizing these sounds automatically without expert intervention presents significant challenge. Particularly, when the available data is limited, this challenge becomes even more compounded. This often leads to a lack of diversity in the generated data. In this paper, we propose effective GAN frameworks, O2C-GAN and OC-SupConGAN for foley sound synthesis in this situation. The proposed frameworks use a new learning method, oneself-conditioned contrastive learning (OCC learning), to solve problems encountered in small dataset. The OCC learning is a method that aims to expand the diversity of data while preserving the inherent attributes of each class within the data. Experiments show that the proposed framework outperforms baseline schemes, ranking 2nd in DCASE2023-T7 Track B with a FAD score of 5.023 on the evaluation set.

Index Terms— Foley sound synthesis, Generative Adversarial Network, Contrastive Learning

1. INTRODUCTION

In recent years, there have been significant advancements in the field of generative models, leading to a growing interest in generating images or sounds that fulfill specific user-defined conditions across various domains. While the audio domain has seen substantial advancements in voice synthesis for singing, text-to-speech (TTS), and music generation, the focus on generating in other acoustic domains, such as sound effects or background noises, has been relatively limited [1, 2, 3]. Notably, foley sound synthesis [4], crucial for enriching auditory experiences in narratives like radio or movies, has been received relatively little attention. Foley sounds are meticulously crafted to synchronize with on-screen events and actions, adding realism and depth to the overall sound design. However, the creation of foley sounds traditionally relies on skilled professionals manually performing and recording the necessary sounds. This expert-driven approach restricts the scalability, flexibility, and creative exploration in sound production. As a result, there is a clear need to explore automated approaches for generating user-desired foley sounds. However, tackling this challenge is accompanied by various difficulties due to the complex nature of foley sounds. Specifically, the problem is further exacerbated when the available data for training models is scarce.

To promote research in the aforementioned field, task 7: Foley sound synthesis was introduced in the DCASE challenge. This aimed to pioneer a new ground of audio synthesis and generate user-desired sounds tailored to custom environments [5]. The following task was divided into two sub-tasks: A and B. Participants were challenged to generate 4-second audio clips with a dataset consisting of about 800 data per class given from the challenge. Task B allows only a dataset given from the challenge, while Task A allows the use of external datasets. We participated in Task B. The requirement to train models with such a limited amount of data imposes a critical flaw for the generative models. The scarcity of data is likely to lead to problem with a lack of diversity in the generated data.


(a) 2C loss (b) O2C loss (c) C-SupCon loss (d) OC-SupCon loss

: Embedding of a data from discriminator
: Embedding of a class label
: Embedding of a reference data from discriminator
: Embedding of a reference data from pre-trained encoder

Figure 1: Schematic figure of the application of oneself-conditioned contrastive learning to different conditional contrastive losses. The color of each shape represents a class. The color of line implies the push-and-pull between the embeddings. The red line represents pulling each embedding while the blue line represents pushing each other. The thickness of the line expresses the strength of the pushing and pulling force. The thicker the line, the stronger the pull or push.

In this paper, we propose oneself-conditioned contrastive learning (OCC learning) that selectively applies label information in conditional contrastive learning methods. The OCC learning uses label information of the data itself but does not use label information between data. This extends the diversity between data while maintaining the class-specific characteristics of the data. In small dataset situations, OCC learning intentionally makes training of GAN difficult, increasing the stability of learning and solving the mode collapse problem. This can be applied to models using conditional contrastive learning method, among which we applied it to ContraGAN [6] and C-SupConGAN [7], thereby proposing O2C-GAN and OC-SupConGAN respectively. The schematic difference between applying and not applying OCC learning to the contrastive loss of each model is depicted in Figure 1.

The rest of this paper is structured as follows: In Section 2, we provide a detailed description of our proposed methods, O2C-GAN and OC-SupConGAN. Section 3 outlines the dataset used and presents the experimental setup to compare with the baseline and other variants of our approach. In Section4, we discuss the results of our experiments, and finally, the last section concludes this paper.

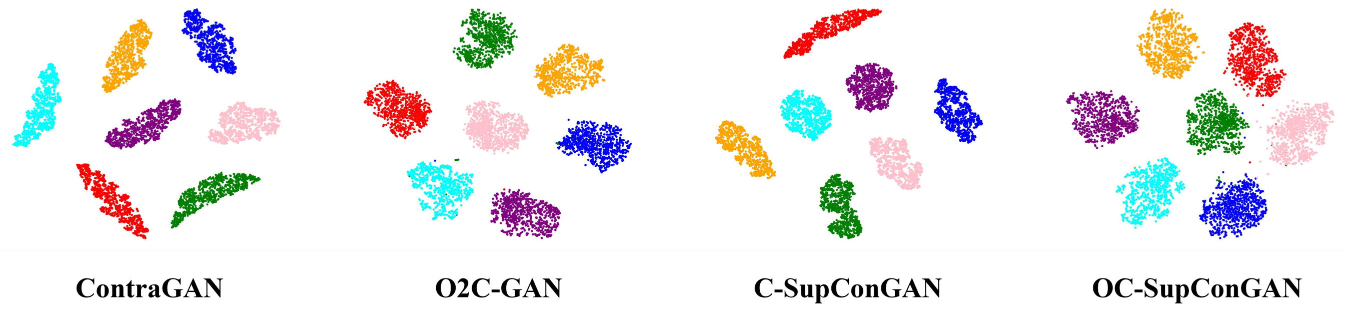


Figure 2: Illustration of a visual comparison of clusters of data embeddings generated by different trained GANs using t-SNE. The points of each t-SNE are the embeddings of the data generated by the generator in the latent space of the discriminator.

2. METHODS

We build the 2-stage system to obtain high performance of FAD score in the DCASE2023-T7 Track B. We denote the first stage as the 'Category-to-Mel spectrogram' section and the second as the 'Mel spectrogram-to-Sound' section for straightforward explanation. In the first step, we explain adversarial loss and introduce two contrastive loss functions which proposed oneself-conditioned contrastive learning (OCC learning) was fed into. We apply the following OCC learning to ContraGAN and C-SupConGAN and demonstrate effectiveness.

2.1. Category-to-Mel spectrogram

Adversarial Loss GAN [8] is composed of a generator and a discriminator. Generator G intends to deceive the discriminator D with a synthetic Mel spectrogram generated from the given label information. On the other hand, the discriminator D must establish the validity of the generated Mel spectrogram and the real Mel spectrogram using label information. G takes noise z_i with label information of class i, c_i, and D takes real Mel spectrogram x_i or fake Mel spectrogram $G(z_i, c_i)$ based on the same label information c_i. We use the hinge loss function as the adversarial loss function, and each objective function for D and G are shown in the equation below.

$$l_D = -\min\left(0, -1 + D\left(x_i, c_i\right)\right) - \min\left(0, -1 - D\left(G\left(z_i, c_i\right), c_i\right)\right) \tag{1}$$

$$l_G = -D\left(G\left(z_i, c_i\right), c_i\right)$$

Oneself-Conditioned Contrastive Loss (O2C loss) We first use ContraGAN, which introduced conditional contrastive loss (2C loss) to GAN. 2C loss is a supervised method that minimizes data-to-data distances belonging to the same class and data-to-class distance and maximizes data-to-data distances belonging to the different classes using data embeddings and class embeddings. To extract embeddings for contrastive learning, we divided the discriminator D into two separate networks: D_1 and D_2. Firstly, $D(\cdot) = D_2(D_1(\cdot))$ is used for calculating adversarial loss. To extract data embeddings d_i, features of real or fake data extracted from $D_1(\cdot)$ are additionally feedforward to the projection head $h(\cdot)$. Thus, we can term $d_i = h(D_1(x_i, c_i))$ for simplicity. The class embedding is extracted by the embedding function $e(\cdot)$ and can be denoted as $e(c_i)$. Further, these features are mapped to the unit hypersphere for cosine similarity computation.

Although the 2C loss function itself produces decent performance, the small number of data per class leads to an unexpected situation. We discovered that the adversarial loss of the discriminator D falls too quickly when we implement the 2C loss function as it is in the current task. This occurrence leads to poor GAN training, further to mode collapse problem [9] that produces similar outputs within the class. To resolve this tragic event, we introduce oneself-conditioned contrastive learning (OCC learning) to the original 2C loss function, and term this oneself-conditioned contrastive loss (O2C loss). As aforementioned above, 2C loss uses label information for both data-to-data and data-to-class relations. O2C loss ignores label information for data-to-data relations and uses label information only for data-to-class relations. The training guidelines for 2C loss and the O2C loss are outlined in (a) and (b) of Figure 1. As shown in the figure 1, the O2C loss maximizes distances between all data embeddings, regardless of whether the data belong to the same class or different classes, and only minimizes data-to-class distance. This optional use of label information distributes data within a class while maintaining the class's distinctiveness. The effect of O2C loss is shown in Figure 2. This solves the mode collapse problem by securing the diversity of data while generating well-classified data according to class and shows tremendous performance improvement. The following data-to-data distance $d2d_{i,j}$ and data-to-class distance $d2c_{i,i}$ can be denoted as the equation 3.

$$d2d_{i,j} = \exp\left(d_i \cdot d_j / \tau_d\right), \; d2c_{i,i} = \exp\left(d_i \cdot e(c_i) / \tau_c\right) \tag{2}$$

With the aforementioned notation, the O2C loss function is defined as follows:

$$l_{O2C}\left(d_i, c_i\right) = -\log\left(\frac{d2c_{i,i}}{d2c_{i,i} + \sum_{k=1}^{N} 1_{i \neq k} \cdot d2d_{i,j}}\right) \tag{3}$$

The $\cdot$ symbol denotes the inner (dot) product, and N is batch size. The hyperparameter τ is applied to control the pushing and pulling forces for distance between embeddings; the larger τ, the weaker the force, and the smaller τ, the stronger the force. C-SupConGAN differentiates the temperature hyperparameter for data-to-data distance τ_d and data-to-class distance τ_c to boost performance. We set $\tau_d = 0.1, \tau_c = 0.1$ by default, but we also conducted the experiment with different values of the two variables, which leads to better results.

Oneself-Conditioned Supervised Contrastive loss (OC-SupCon loss) C-SupConGAN, an advanced version of ContraGAN, uses pre-trained data features to support the feature learning of the discriminator. The conditional supervised contrastive loss (C-SupCon loss) appends data-to-source relation to prior 2C loss. For data-to-source relation, C-SupCon loss uses reference data embedding extracted from the pre-trained encoder $f(\cdot)$. This aids GAN's feature learning, thereby reduces the instability of the training process and enable long-term training, and ultimately improved performance. Nonetheless, mode collapse still occurs when C-SupConGAN is applied to the current task as it is. Therefore, we also apply the OCC learning to C-SupCon loss and call it OC-SupCon loss.

$$d2s_{i,i} = \exp\left(d_i \cdot f(x_i)/\tau_c\right) \tag{4}$$

In the same way, the OC-SupCon loss can be described as follows:

$$l_{OC-SupCon}\left(d_i, c_i\right) = -\log\left(\frac{d2s_{i,i} + d2c_{i,i}}{d2s_{i,i} + d2c_{i,i} + \sum_{k=1}^{N} 1_{i \neq k} \cdot d2d_{i,k}}\right) \tag{5}$$

The conceptual difference between the C-SupCon loss and the OC-SupCon loss can be schematically confirmed in Figure 1.

We used ResNet18 [10] as the encoder network $f(\cdot)$, and it was pretrained with Supervised Contrastive Learning (SupCon) [11] loss function. For audio augmentation, we used fade in/out and time masking during the pretraining process. After the pretraining process is completed, we proceed with classification finetuning and classification evaluation. Since additional dataset such as the evaluation dataset was not open to the public, we could only evaluate the performance of classification on the training set. The classification accuracy achieved 100%, which may appear as overfitting, but we can infer that the pretrained encoder network $f(\cdot)$ is capable of extracting high-quality audio embeddings from the training set. Thus, we use the data embedding $f(x_i)$ extracted from the pretrained encoder $f(\cdot)$ as a reference to the data embedding d_i extracted from the discriminator.

Our total system is optimized through two types of loss function, which is the combination of adversarial loss and O2C loss function and the combination of adversarial loss and OC-SupCon loss function. O2C loss or OC-SupCon loss is expressed as l_C. In this way, total loss function $\mathcal{L}$ can be described:

$$\mathcal{L}_D = \frac{1}{N}\sum_{k=1}^{N} l_D + \frac{1}{N}\sum_{k=1}^{N} l_C,\ \mathcal{L}_G = \frac{1}{N}\sum_{k=1}^{N} l_G + \frac{1}{N}\sum_{k=1}^{N} l_C \tag{6}$$

$$\mathcal{L} = \mathcal{L}_D + \mathcal{L}_G \tag{7}$$

We term GAN using O2C loss as O2C-GAN, and OC-SupCon loss as OC-SupConGAN.

2.2. Mel spectrogram-to-sound

After the training on the first stage, the trained generator network G have the ability to generate Mel-spctrograms from class categories. During the second stage, a pre-trained vocoder network transforms the generated Mel-spectrogram into a time-domain digital audio signal. Instead of proposing a new vocoder network, we leverage the pre-trained vocoder network, HiFi-GAN [12], provided by the DCASE challenge.

3. EXPERIMENT

We design our experiments for three purposes. First, we demonstrate the effectiveness of oneself-conditioned contrastive learning (OCC learning). The performances of models with or without OCC learning are compared. Second, we examine the effect of the temperature hyperparameter τ on the frameworks by adjusting τ_c. Third, we verify the two proposed models exceed the performance of the baseline system.

3.1. Experiment metrics

We use Frechet Audio Distance (FAD) [13]. FAD is a standard metric for music enhancement and is very useful in that it is a reference-free evaluation metric. FAD can be employed even in the absence of a ground truth reference audio because it is calculated from collections of hidden representations of created and real samples. The FAD score can be computed by multivariate Gaussians between the generated data set and the actual audio data set, which can be referred to as the reference embeddings.

3.2. Implementation Details

We use the log mel-band energies of input audio as an audio feature. We set the frame length to 1024, and hop size to 256. All the models we train are devised to generate 80×344 mel spectrogram. Initially, we employed the learning rates used in C-SupConGAN to train our proposed models. The generator was trained with a learning rate of 0.0001, while the discriminator was trained a learning rate of 0.0004. However, the small amount of dataset led to the circumstance of discriminator D learning too quickly. Thus, we set both learning rates equally to 0.0001. For all models, we use Adam optimizer [14] with $\beta_1 = 0.5$ and $\beta_2 = 0.999$ for training. For contrastive learning, we build a 2-layer projection layer $h(\cdot)$ which embeds the output of the portion of discriminator network D_1 to 128-dimension. During training, we freeze the weight of pretrained encoder network $E(\cdot)$.

3.3. Dataset

The DCASE2023-T7 Track B development set contains 4,850 labeled sound fragments, classified into 7 categories: dog bark, footstep, gunshot, keyboard, moving motor vehicle, rain, and sneeze/cough. Each sound was fitted to a length of 4 seconds, and zero-padded or segmented if necessary. All audio was transferred to mono 16-bit 22,050 Hz sampling rate [5]. As we are participating in subtask B, we do not use any external sources.

4. RESULTS

	w/o OCC learning		w. OCC learning	
	ContraGAN	C-SupConGAN	O2C-GAN	OC-SupConGAN
FAD	12.667	12.552	5.480	5.230

Table 1: The comparison of FAD score on two baselines with and without OCC learning.

Method	DT	CT	DogBark	Footstep	GunShot	Keyboard	Vehicle	Rain	Sneeze	Average
O2C-GAN	0.1	0.1	2.784	4.370	4.667	3.555	17.511	3.899	1.577	5.480
O2C-GAN	0.1	1.0	3.348	3.990	3.495	4.074	14.861	3.529	1.865	5.023
OC-SupConGAN	0.1	0.1	2.616	3.739	6.322	4.089	14.172	4.304	1.371	5.230
OC-SupConGAN	0.1	1.0	4.854	3.103	4.790	3.665	13.604	3.727	1.435	5.026

Table 2: The comparison of FAD score on our proposed methods submitted to the DCASE2023-T7 Track B.

4.1. Effectiveness of OCC learning

We demonstrate the effectiveness of oneself-conditioned contrastive learning (OCC learning) by comparing the cases with and without OCC learning for two different GAN. We use t-SNE [15] to visually compare clusters of embeddings of data generated by different trained GANs. In Figure 2, the points of each t-SNE are the embeddings of the data generated by the generator in the latent space of the discriminator. Generated data are compared quantitatively using FAD and it is shown in Table 1.

The GANs which do not use OCC learning train the model in a way that the distance between the data embedding and the data's own condition embeddings as well as the distance between data embeddings belonging to the same class becomes close. In ContraGAN, class embedding is used as condition, and in C-SupConGAN, class embedding and pretrained data embedding are used as condition. When the dataset with a large amount of data per class is used, this helps the data to cluster for class distinction. When a small dataset with fewer data per class is used as in the current task, this causes the data belonging to the class to clump too much, resulting in a decrease in the diversity of data. As a result, the loss of discriminator D drops rapidly, resulting in poor training of GAN. The GANs using OCC learning, O2C-GAN, and OC-SupConGAN, optimize the model so that the distance between the data embedding and the data's own condition embeddings becomes close as in the previous loss function, but the distance between data embeddings belonging to the same class becomes far. This expands data clustering, amplifying the diversity among data belonging to the same class while maintaining class distinctiveness by retaining class attributes in data. As a result, by making learning task difficult, GAN training becomes stable, and various and higher-quality data are generated. Moreover, OC-SupConGAN leverages additionally pre-trained data embeddings as the condition to enhance the subjectivity of the data. Consequently, it leads to a broader dispersion of data and improves performance compared to O2C-GAN. These effects are visually illustrated in Figure 2 and shows a significant performance improvement in Table 1.

4.2. Performance Comparison

Unlike ContraGAN, C-SupConGAN uses the different temperature hyperparameters τ, which controls the strength of pulling or pushing between embeddings, for data-to-data distance τ_d and data-to-condition distance τ_c. The higher the τ value, the weaker the strength, the lower the τ value, the stronger. The temperature hyperparameter τ_d, which controls the strength of the data-to-data distance, is called DT, and the temperature hyperparameter τ_c, which controls the strength of the data-to-condition distance, is called CT. In C-SupConGAN, experiments using various values of τ_d and τ_c were conducted, and the best performance was achieved at $\tau_d = 0.1$ and $\tau_c = 1.0$. We also performed experiments not only with $\tau_d = 0.1$ and $\tau_c = 0.1$, which were used by default, but also with $\tau_d = 0.1$ and $\tau_c = 0.1$. In OCC learning, this leads the distance between all data embeddings to be strongly far, and the distance between the data embeddings and the data's own condition embeddings to be weakly close. This encourages data to maintain the unique characteristics of the class, but weaken the binding force of the class, and secure more diversity by widening the distance from other data. As a result, as shown in Table 2, the generation performance is further improved.

Class	Baseline	Ours	
		O2C-GAN	**OC-SupConGAN**
DogBark	13.412	3.348	4.854
Footstep	8.108	3.990	3.103
GunShot	7.952	3.495	4.790
Keyboard	5.230	4.074	3.665
Vehicle	16.107	14.861	13.604
Rain	13.338	3.529	3.727
Sneeze	3.771	1.865	1.435
Average	9.702	5.023	5.026

Table 3: The FAD score on each class of baseline scheme, O2C-GAN, and OC-SupConGAN.

Table 3 refers to the performance comparison between baseline method with our proposed methods: O2C-GAN and OC-SupConGAN. Our two techniques outperform baseline methods in every way. In particular, in 'DogBark' and 'Rain' classses, our baseline frameworks performed 4 to 5 times better than the existing baseline. We speculate that this remarkable performance is due to the proposed frameworks' ability to enhance variance of data features within the class while keeping distinct characteristic of class using our proposed OCC learning. In Table 3, we can see that improvement of FAD performance of class 'Moving Motor Vehicle' is rather low. We infer this outcome is based on insufficient variance of audio data within the class. This trait induce generation of similar data in the class regardless of the methods. To sum up, our proposed frameworks achieve the average FAD score of 5.023 and 5.026, which is the half of the baseline.

5. CONCLUSION

In this paper, we propose new GAN frameworks, O2C-GAN and OC-SupConGAN, for foley sound synthesis introduced by DCASE challenge. The proposed frameworks use a new learning method, oneself-conditioned contrastive learning (OCC learning), to solve problems encountered in small dataset. The OCC learning is a method that aims to expand the diversity of data while maintaining the class properties in the data. Our proposed frameworks achieved FAD scores of 5.023 and 5.026, outperformed the baseline framework, and ranked 2nd in the DCASE2023-T7 Track B.

6. REFERENCES

[1] Y. Hono, K. Hashimoto, K. Oura, Y. Nankaku, and K. Tokuda, "Singing voice synthesis based on generative adversarial networks," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6955–6959.

[2] N. Kalchbrenner, E. Elsen, K. Simonyan, S. Noury, N. Casagrande, E. Lockhart, F. Stimberg, A. Oord, S. Dieleman, and K. Kavukcuoglu, "Efficient neural audio synthesis," in *International Conference on Machine Learning*. PMLR, 2018, pp. 2410–2419.

[3] J. Engel, C. Resnick, A. Roberts, S. Dieleman, M. Norouzi, D. Eck, and K. Simonyan, "Neural audio synthesis of musical notes with wavenet autoencoders," in *International Conference on Machine Learning*. PMLR, 2017, pp. 1068–1077.

[4] http://dcase.community/challenge2023/.

[5] K. Choi, J. Im, L. Heller, B. McFee, K. Imoto, Y. Okamoto, M. Lagrange, and S. Takamichi, "Foley sound synthesis at the dcase 2023 challenge," *In arXiv e-prints: 2304.12521*, 2023.

[6] M. Kang and J. Park, "Contragan: Contrastive learning for conditional image generation," *Advances in Neural Information Processing Systems*, vol. 33, pp. 21 357–21 369, 2020.

[7] H. Chung and J.-K. Kim, "C-supcongan: Using contrastive learning and trained data features for audio-to-image generation," in *Proceedings of the 2022 5th Artificial Intelligence and Cloud Computing Conference*, 2022, pp. 135–142.

[8] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, and Y. Bengio, "Generative adversarial networks, 1–9," *arXiv preprint arXiv:1406.2661*, 2014.

[9] H. Thanh-Tung and T. Tran, "Catastrophic forgetting and mode collapse in gans," in *2020 international joint conference on neural networks (ijcnn)*. IEEE, 2020, pp. 1–10.

[10] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*. Springer, 2016, pp. 630–645.

[11] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan, "Supervised contrastive learning," *Advances in neural information processing systems*, vol. 33, pp. 18 661–18 673, 2020.

[12] J. Kong, J. Kim, and J. Bae, "Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis," *Advances in Neural Information Processing Systems*, vol. 33, pp. 17 022–17 033, 2020.

[13] K. Kilgour, M. Zuluaga, D. Roblek, and M. Sharifi, "Fréchet audio distance: A reference-free metric for evaluating music enhancement algorithms." in *INTERSPEECH*, 2019, pp. 2350–2354.

[14] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[15] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." *Journal of machine learning research*, vol. 9, no. 11, 2008.

DESCRIPTION AND DISCUSSION ON DCASE 2023 CHALLENGE TASK 2: FIRST-SHOT UNSUPERVISED ANOMALOUS SOUND DETECTION FOR MACHINE CONDITION MONITORING

Kota Dohi[1], *Keisuke Imoto*[2], *Noboru Harada*[3], *Daisuke Niizumi*[3], *Yuma Koizumi*[4], *Tomoya Nishida*[1], *Harsh Purohit*[1], *Ryo Tanabe*[1], *Takashi Endo*[1], *and Yohei Kawaguchi*[1]

[1] Hitachi, Ltd., Japan, `kota.dohi.gr@hitachi.com`
[2] Doshisha University, Japan, `keisuke.imoto@ieee.org`
[3] NTT Corporation, Japan, `noboru.harada.pv@hco.ntt.co.jp`
[4] Google, Japan, `koizumiyuma@google.com`

ABSTRACT

We present the task description of the Detection and Classification of Acoustic Scenes and Events (DCASE) 2023 Challenge Task 2: "First-shot unsupervised anomalous sound detection (ASD) for machine condition monitoring". The main goal is to enable rapid deployment of ASD systems for new kinds of machines without the need for hyperparameter tuning. In the past ASD tasks, developed methods tuned hyperparameters for each machine type, as the development and evaluation datasets had the same machine types. However, collecting normal and anomalous data as the development dataset can be infeasible in practice. In 2023 Task 2, we focus on solving the first-shot problem, which is the challenge of training a model on a completely novel machine type. Specifically, (i) each machine type has only one section (a subset of machine type) and (ii) machine types in the development and evaluation datasets are completely different. Analysis of 86 submissions from 23 teams revealed that the keys to outperform baselines were: 1) sampling techniques for dealing with class imbalances across different domains and attributes, 2) generation of synthetic samples for robust detection, and 3) use of multiple large pre-trained models to extract meaningful embeddings for the anomaly detector.

Index Terms— anomaly detection, acoustic condition monitoring, domain shift, first-shot problem, DCASE Challenge

1. INTRODUCTION

Anomalous sound detection (ASD) [1–7] is the task of identifying whether the sound emitted from a target machine is normal or anomalous. Automatic detection of mechanical failure is essential for the artificial intelligence (AI)–based factory automation. Use of machine sounds for promptly detecting machine anomalies is useful for monitoring a machine's condition.

One fundamental challenge regarding the application of ASD systems is that anomalous samples for training can be insufficient both in number and type. In 2020, we organized the first ASD task in Detection and Classification of Acoustic Scenes and Event (DCASE) Challenge 2020 Task 2 [8]; "*unsupervised ASD*" that was intended to detect unknown anomalous sounds using only normal sound samples as the training data [1–7].

For the wide-spread application of ASD systems, advanced tasks such as handling of domain shifts should be tackled [9]. Domain shifts are differences between the source and target domain data caused by a machine's operational conditions or environmental noise. Since methods developed in the task in 2020 fail to distinguish normal sounds subject to domain shifts and anomalous sounds, the detection performance of these methods can degrade under domain-shifted conditions. To reflect domain-shifted conditions, we organized DCASE 2021 Task 2 [9], "*unsupervised ASD under domain shifted conditions*" and DCASE Challenge 2022 Task 2 [10], "*unsupervised ASD applying domain generalization techniques*". The task in 2021 focused on handling domain shifts using domain adaptation techniques, and the task in 2022 focused on handling domain shifts using domain generalization techniques.

Previous tasks from 2020 to 2022 had premises such as multiple machine IDs or section IDs for each machine type and the same set of machine types for the development and evaluation datasets. As a result, developed methods made use of multiple IDs within a machine type or tuned hyperparameters using normal and anomalous data from the development dataset. However, these premises could pose a barrier when attempting to apply methods developed in the past tasks to real-world scenarios, as preparing multiple IDs for each machine type or collecting normal and anomalous data for the development dataset can be time-consuming or even infeasible.

To solve the problem described above, we designed DCASE Challenge 2023 Task 2, "*First-Shot Unsupervised Anomalous Sound Detection for Machine Condition Monitoring*". This task is aimed at developing methods for solving the first-shot problem and rapidly deploying ASD systems, while the task also focuses on developing domain generalization techniques for handling domain shifts. Specifically, only one section is provided for each machine type, and the sets of machine types are completely different between the development and evaluation datasets.

We received 86 submissions from 23 teams. By analyzing these submissions, we found techniques several top-rankers used in common: 1) sampling techniques for dealing with class imbalances , 2) generation of synthetic samples for robust detection, and 3) use of multiple large pre-trained models to extract meaningful embeddings for the anomaly detector.

2. FIRST-SHOT UNSUPERVISED ANOMALOUS SOUND DETECTION UNDER DOMAIN SHIFTED CONDITIONS

Let the L-dimensional time-domain observation $\boldsymbol{x}_i \in \mathbb{R}^L$ be an audio clip that includes a sound emitted from a machine with a specific ID i. The ID serves as a unique identifier that indicates the machine's class on the basis of its model number or other identifying specifications. The goal of the ASD task is to classify the machine as normal or anomalous by computing the anomaly score $\mathcal{A}_\theta(\boldsymbol{x}_i)$ by using an anomaly score calculator $\mathcal{A}$ with parameters θ. $\mathcal{A}$ is trained to assign higher scores to anomalous samples and lower scores to normal samples. The input to $\mathcal{A}$ can be the audio clip $\boldsymbol{x}_i$ or $\boldsymbol{x}_i$ with additional information such as the ID. The machine is classified as anomalous if $\mathcal{A}_\theta(\boldsymbol{x}_i)$ exceeds a pre-defined threshold ϕ

$$\text{Decision} = \begin{cases} \text{Anomaly} & (\mathcal{A}_\theta(\boldsymbol{x}_i) > \phi) \\ \text{Normal} & (\text{otherwise}). \end{cases} \tag{1}$$

The primary difficulty in this task is to train $\mathcal{A}$ using only normal sounds (unsupervised ASD). The DCASE 2020 Challenge Task 2 was designed to address this issue.

In real-world scenarios, the domain-shift problem also needs to be solved. Domain shifts are variations in conditions between training and testing phases that impact the distribution of normal sound data. These shifts can arise from differences in operating speed, machine load, viscosity, heating temperature, environmental noise, signal-to-noise ratio, and other factors. Two domains, **source domain** and **target domain**, are defined: the former refers to the original condition with sufficient training data and the latter refers to another condition with only a few samples. The 2021 Task 2 aimed to develop domain adaptation techniques, assuming the domain information (source/target) of each sample is known. However, in practice, obtaining domain information is challenging due to the difficulty in detecting domain shifts.

To address the challenges of applying domain adaptation techniques in real-world scenarios, the 2022 Task 2 focused on developing domain generalization techniques. Domain generalization techniques for ASD aim at detecting anomalies from different domains with a single threshold. These techniques, unlike domain adaptation techniques, do not require detection of domain shifts or adaptation of the model during the testing phase.

Although several novel ASD methods have been proposed in past tasks, we have recognized that their application in real-world scenarios remains challenging. This is because certain assumptions in previous tasks may not hold in practice. One such assumption is that participants were allowed to tune the hyperparameters of the model by using the test data of the development dataset. However, this is often infeasible in real-world applications where the machine type can be completely new or the amount of test data can be insufficient for tuning hyperparameters. Another assumption is the existence of multiple IDs for a machine type. This assumption has facilitated the development of outlier exposure approaches [11], where sound clips from different machines are used as anomalies. However, in many practical cases, the number of machines for a machine type can be limited. This limitation arises because the customers may not possess multiple machines of the same machine type, or they may initially plan to install the system for only a few machines. As a result, the developed methods in the previous tasks may not be immediately applicable in practice.

To overcome these new challenges, the organizers designed the 2023 Task 2 with two main features: (i) completely different set of machine types between the development and evaluation dataset and (ii) Only one section for each machine type. Because the machine types are completely different between the development and evaluation dataset, tuning hyperparameters using the test data from the development dataset is no longer feasible. Furthermore, since only one section is available for each machine type, multiple IDs within a machine type cannot be used. As a result, participants are expected to develop ASD methods without tuning hyperparameters using the test data and without relying on multiple IDs within a machine type. We name these challenges the "first-shot problem", as these challenges replicate practical cases where the ASD system has to be deployed for a novel machine type or with a limited number of example measurements.

3. TASK SETUP

3.1. Dataset

The data for this task comprises three datasets: **development dataset**, **additional training dataset**, and **evaluation dataset**. Each dataset includes seven machine types, with one section per machine type. **Machine type** means the type of machine such as fan, gearbox, bearing, etc. **Section** is a subset or whole data within each machine type.

Each recording is a single-channel audio with a duration of 6 to 18 s and a sampling rate of 16 kHz. We mixed machine sounds recorded at laboratories and environmental noise samples recorded at factories and in the suburbs to create each sample in the dataset. For the details of the recording procedure, please refer to the papers on ToyADMOS2 [12] and MIMII DG [10].

The **development dataset** consists of seven machine types (fan, gearbox, bearing, slide rail, ToyCar, ToyTrain), and each machine type has one section that contains a complete set of the training and test data. Each section provides (i) 990 normal clips from a source domain for training, (ii) 10 normal clips from a target domain for training, and (iii) 100 normal clips and 100 anomalous clips from both domains for the test. We provided domain information (source/target) in the test data for the convenience of participants. Attributes that represent operational or environmental conditions are also provided in the file names and attribute csvs.

The **additional training dataset** provides novel seven machine types (Vacuum, ToyTank, ToyNscale, ToyDrone, bandsaw, grinder, shaker). Each section consists of (i) 990 normal clips in a source domain for training and (ii) 10 normal clips in a target domain for training. Attributes are provided in this dataset.

The **evaluation dataset** provides the same machine types as the additional training dataset. Each section consists of 200 test clips, none of which have a condition label (i.e., normal or anomaly) or the domain information. Attributes are not provided.

The data for this task differs from the 2022 version in two main aspects: reduced number of sections per machine type (from six in 2022 to one in this task) and a completely different set of machine types between the development and evaluation datasets. As a result, participants are required to train a model for a novel machine type using only one section for each machine type and without hyperparameter tuning using the development dataset.

3.2. Evaluation metrics

For evaluation, the area under the receiver operating characteristic curve (AUC) was employed as a metric to assess the overall detection performance, while the partial AUC (pAUC) was utilized to measure performance in a low false-positive rate (FPR) range $[0, p]$. In this task, we used $p = 0.1$. In domain generalization task, the AUC for each domain and pAUC for each section are calculated as

$$\mathrm{AUC}_{m,n,d} = \frac{1}{N_d^- N_n^+} \sum_{i=1}^{N_d^-} \sum_{j=1}^{N_n^+} \mathcal{H}(\mathcal{A}_\theta(x_j^+) - \mathcal{A}_\theta(x_i^-)), \tag{2}$$

$$\mathrm{pAUC}_{m,n} = \frac{1}{\lfloor pN_n^- \rfloor N_n^+} \sum_{i=1}^{\lfloor pN_n^- \rfloor N_n^+} \sum_{j=1}^{N_n^+} \mathcal{H}(\mathcal{A}_\theta(x_j^+) - \mathcal{A}_\theta(x_i^-)), \tag{3}$$

where m represents the index of a machine type, n represents the index of a section, $d = \{\mathrm{source}, \mathrm{target}\}$ represents a domain, $\lfloor \cdot \rfloor$ is the flooring function, and $\mathcal{H}(x)$ returns 1 when $x > 0$ and 0 otherwise. Here, $\{x_i^-\}_{i=1}^{N_d^-}$ are normal test clips in domain d in section n and $\{x_j^+\}_{j=1}^{N_n^+}$ are anomalous test clips in section n in machine type m. The N_d^- is the number of normal test clips in domain d, N_n^- is the number of normal test clips in section n, and N_n^+ is the number of anomalous test clips in section n.

The official score Ω is given by the harmonic mean of the AUC and pAUC scores over all machine types and sections:

$$\begin{aligned} \Omega \;=\;& h\left\{\mathrm{AUC}_{m,n,d},\ \mathrm{pAUC}_{m,n} \quad \middle| \right. \\ & \left. m \in \mathcal{M},\ n \in \mathcal{S}(m),\ d \in \{\mathrm{source}, \mathrm{target}\}\right\}, \end{aligned} \tag{4}$$

where $h\{\cdot\}$ represents the harmonic mean (over all machine types, sections, and domains), $\mathcal{M}$ represents the set of machine types, and $\mathcal{S}(m)$ represents the set of sections for machine type m.

3.3. Baseline systems and results

The organizers provided an Autoencoder (AE)-based baseline system with two different ways of calculating the anomaly scores. We present the baseline system and its detection performance. For details, please refer to [13].

3.3.1. Autoencoder-based baseline

First, the log-mel-spectrogram of the input $X = \{X_k\}_{k=1}^{K}$ is calculated, where $X_k \in \mathbb{R}^F$, and F and K are the number of mel-filters and time-frames, respectively. Then, the acoustic feature at k is obtained by concatenating consecutive frames of the log-mel-spectrogram as $\psi_k = (X_k, \cdots, X_{k+P-1}) \in \mathbb{R}^D$, where $D = P \times F$, and P is the number of frames of the context window.

3.3.2. Simple Autoencoder mode

In this mode, the anomaly score is calculated as

$$A_\theta(X) = \frac{1}{DK} \sum_{k=1}^{K} \|\psi_k - r_\theta(\psi_k)\|_2^2, \tag{5}$$

where r_θ is the vector reconstructed by the AE, and $\|\cdot\|_2$ is ℓ_2 norm.

3.3.3. Selective Mahalanobis mode

In this mode, the Mahalanobis distance between the observed sound and reconstructed sound is used to calculate the anomaly score. The anomaly score is given as

$$A_\theta(X) = \frac{1}{DK} \sum_{k=1}^{K} min\{D_s(\psi_k, r_\theta(\psi_k)), D_t(\psi_k, r_\theta(\psi_k))\}, \tag{6}$$

$$D_s(\cdot) = Mahalanobis(\psi_k, r_\theta(\psi_k), \Sigma_s^{-1}), \tag{7}$$

$$D_t(\cdot) = Mahalanobis(\psi_k, r_\theta(\psi_k), \Sigma_t^{-1}), \tag{8}$$

where Σ_s^{-1} and Σ_t^{-1} are the covariance matrices calculated with the source domain data and target domain data of each section, respectively.

Table 1: Results with Simple Autoencoder mode

Machine type	Section	AUC [%]		pAUC [%]
		Source	Target	
ToyCar	00	70.10 ± 0.46	46.89 ± 2.67	52.47 ± 1.28
ToyTrain	00	57.93 ± 2.12	57.02 ± 0.79	48.57 ± 0.32
bearing	00	65.92 ± 0.73	55.75 ± 0.76	50.42 ± 0.79
fan	00	80.19 ± 2.43	36.18 ± 3.71	59.04 ± 1.24
gearbox	00	60.31 ± 0.56	60.69 ± 0.63	53.22 ± 0.60
slider	00	70.31 ± 0.20	48.77 ± 0.12	56.37 ± 0.31
valve	00	55.35 ± 1.18	50.69 ± 1.12	51.18 ± 0.35

Table 2: Results with Selective Mahalanobis mode

Machine type	Section	AUC [%]		pAUC [%]
		Source	Target	
ToyCar	00	74.53 ± 1.55	43.42 ± 2.53	49.18 ± 0.49
ToyTrain	00	55.98 ± 2.41	42.45 ± 1.06	48.13 ± 0.17
bearing	00	65.16 ± 0.76	55.28 ± 0.57	51.37 ± 0.81
fan	00	87.10 ± 2.20	45.98 ± 4.43	59.33 ± 0.90
gearbox	00	71.88 ± 0.66	70.78 ± 0.62	54.34 ± 0.30
slider	00	84.02 ± 1.10	73.29 ± 0.60	54.72 ± 0.25
valve	00	56.31 ± 1.38	51.40 ± 0.40	51.08 ± 0.13

3.3.4. Results

The AUC and pAUC for each machine type are shown in Tables 1 and 2. The results are average of five independent runs.

4. CHALLENGE RESULTS

We received 86 submissions from 23 teams. Eleven teams outperformed the simple Autoencoder baseline, and eight teams outperformed the selective Mahalanobis baseline. The number of teams was significantly fewer than for the task in 2022, where 22 out of 31 teams outperformed the baselines. This observation suggests that the new features in this year's task, such as having only one section for each machine type and novel machine types in the evaluation dataset, have increased the task's difficulty level. Despite these challenges, several top-ranked teams significantly outperformed the baselines. Figure 1 illustrates the harmonic means of the AUCs for the top 10 teams. Notably, all eight teams that outperformed the baselines in the official scores also surpassed the baselines in the harmonic mean of the AUCs in the target domain. This indicates that higher AUCs in the target domain were crucial for higher ranks.

Since the task this year focused on developing ASD methods that work for novel machine types, we compared the AUCs between the development and evaluation datasets. Figure 2 shows the AUCs from the top 20 teams for the source domain, while Figure 3 displays the AUCs for the target domain. From Figure 2, it can be observed that approximately half of the teams achieved higher source-domain AUCs in the evaluation dataset compared to the development dataset. This indicates that, with a sufficient amount of training data, detection for a novel machine type can be possible without significant degradation in performance. However, Figure 3 reveals that the target-domain AUCs were lower in the evaluation dataset for most teams. This underscores the difficulty of dealing with domain shifts for novel machine types. The lower AUCs observed in the evaluation dataset for the target domain can be attributed to the fact that the variations induced by domain shifts can differ significantly for each machine type. In this case, when domain generalization techniques are developed for maximizing the AUCs in the development dataset, using the same techniques for the evaluation dataset will degrade the performance. Addressing these variations becomes more challenging when only a limited number of samples are available, further complicating the problem.

We summarize approaches used by top-ranked teams in the following.

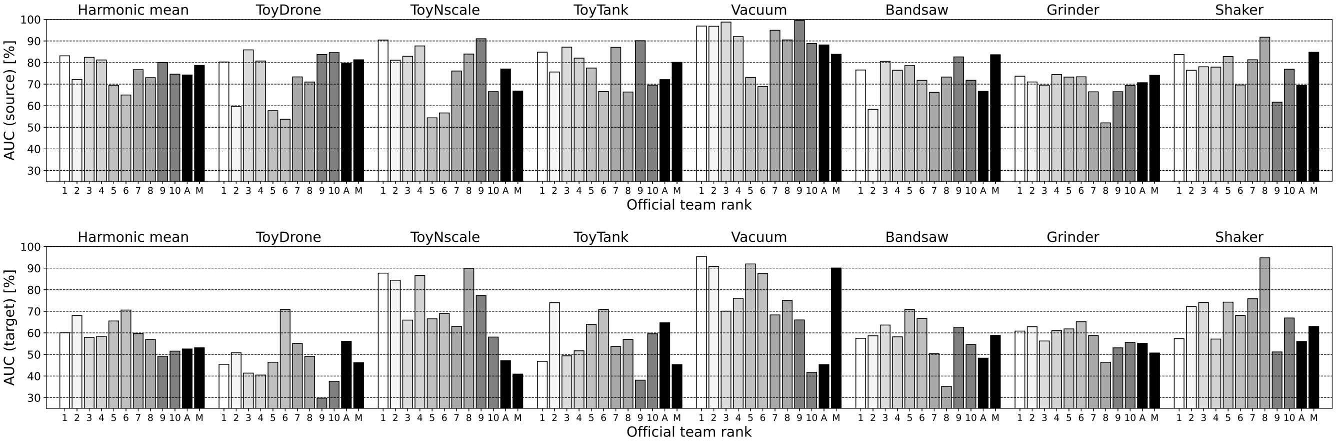


Figure 1: Evaluation results of top 10 teams in the ranking. Average source-domain AUC (Top) and target-domain AUC (bottom) for each machine type. Label "A" and "M" on the x-axis denote simple Autoencoder mode and selective Mahalanobis mode, respectively.

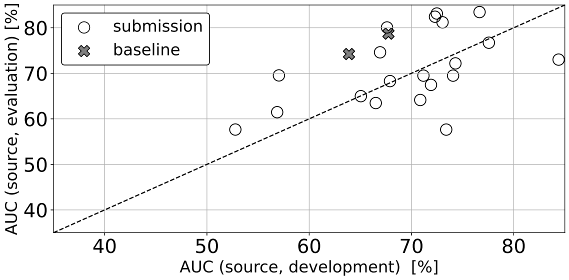


Figure 2: Comparison of average source-domain AUC for the development dataset and evaluation dataset across teams.

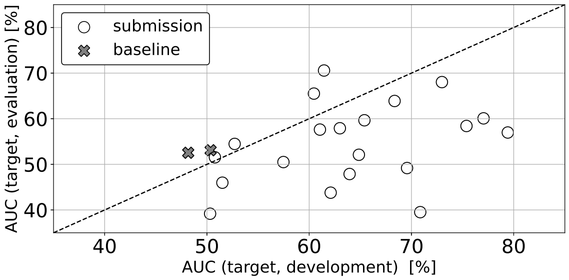


Figure 3: Comparison of average target-domain AUC for the development dataset and evaluation dataset across teams.

a. Oversampling for imbalance compensation

Because the number of samples in the datasets is imbalanced across domains and attributes, compensating for these class imbalances can improve the detection performance. The 6th team [14] duplicated samples from classes with fewer samples, while the 1st and 2nd teams [15, 16] oversampled target-domain data using SMOTE [17]. These approaches are only seen among top-rankers, and can be one of the key factors for outperforming the baselines.

b. Synthetic data generation for robust detection

Synthetic data can be utilized to accurately model the distribution of normal data and enhance the robustness of the detection model. The 1st, 4th, 5th, 10th, and 19th teams employed Mixup [18] including its variants [15, 19–22], and obtained higher source-domain AUCs. Other papers used other data augmentation techniques such as speed perturbation, noise injection, and pitch shift [14, 16, 23]. The treatment of generated synthetic data varies among teams. While the 4th team [19] treated them as anomalous samples that belong to a new class, the 1st and 5th teams [15, 20] treated them as normal samples. Mixup can be one of the key factors for outperforming the baselines, as this technique was used by several top-rankers and teams that achieved higher source-domain AUCs.

c. Attribute ID classification using pre-trained models

Although only one section was provided for each machine type, attributes were included in the development and additional training dataset. As a result, many participants trained attribute classifiers or machine type classifiers to obtain embeddings that could be used for outlier detectors [14–16, 19, 24, 25]. For the outlier detector, k-nearest neighbors algorithm (kNN) was used by most of the teams.

Pre-trained models are used [16, 24, 26] for attribute classifiers or machine type classifiers. Although pre-trained models have been used by participants in previous tasks, the 2nd and 3rd teams [16,24] are the first teams that used multiple large pre-trained models to achieve higher official scores. These pre-trained models were fine-tuned with classification objectives, i.e., attribute or machine type classification.

d. Other novel approaches

The 3rd team [24] grouped machine types into several categories so that generalization ability on novel machine types can be obtained. The 7th team [23] used AudioLDM [27], a text-to-audio model, to generate pseudo anomalous sounds from the text input.

5. CONCLUSION

This paper presented an overview of the task and analysis of the solutions submitted to DCASE 2023 Challenge Task 2. The task was aimed to develop an ASD system that works for a novel machine type with a single section for each machine type. Analysis of the submission revealed that, for novel machine types, detection in the target domain can be of significant difficulty compared to the source domain. The analysis also revealed useful methods for outperforming the baselines: 1) sampling techniques for dealing with class imbalances, 2) generation of synthetic samples by mix-up and its variants, and 3) use of multiple large pre-trained models for attribute ID classification.

6. REFERENCES

[1] Y. Koizumi, S. Saito, H. Uematsu, and N. Harada, “Optimizing acoustic feature extractor for anomalous sound detection based on Neyman-Pearson lemma,” in *EUSIPCO*, 2017, pp. 698–702.

[2] Y. Kawaguchi and T. Endo, “How can we detect anomalies from subsampled audio signals?” in *MLSP*, 2017.

[3] Y. Koizumi, S. Saito, H. Uematsu, Y. Kawachi, and N. Harada, “Unsupervised detection of anomalous sound based on deep learning and the Neyman-Pearson lemma,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 1, pp. 212–224, Jan. 2019.

[4] Y. Kawaguchi, R. Tanabe, T. Endo, K. Ichige, and K. Hamada, “Anomaly detection based on an ensemble of dereverberation and anomalous sound extraction,” in *ICASSP*, 2019, pp. 865–869.

[5] Y. Koizumi, S. Saito, M. Yamaguchi, S. Murata, and N. Harada, “Batch uniformization for minimizing maximum anomaly score of DNN-based anomaly detection in sounds,” in *WASPAA*, 2019, pp. 6–10.

[6] K. Suefusa, T. Nishida, H. Purohit, R. Tanabe, T. Endo, and Y. Kawaguchi, “Anomalous sound detection based on interpolation deep neural network,” in *ICASSP*, 2020, pp. 271–275.

[7] H. Purohit, R. Tanabe, T. Endo, K. Suefusa, Y. Nikaido, and Y. Kawaguchi, “Deep autoencoding GMM-based unsupervised anomaly detection in acoustic signals and its hyperparameter optimization,” in *DCASE Workshop*, 2020, pp. 175–179.

[8] Y. Koizumi, Y. Kawaguchi, K. Imoto, T. Nakamura, Y. Nikaido, R. Tanabe, H. Purohit, K. Suefusa, T. Endo, M. Yasuda, and N. Harada, “Description and discussion on DCASE2020 challenge task2: Unsupervised anomalous sound detection for machine condition monitoring,” in *DCASE Workshop*, 2020, pp. 81–85.

[9] Y. Kawaguchi, K. Imoto, Y. Koizumi, N. Harada, D. Niizumi, K. Dohi, R. Tanabe, H. Purohit, and T. Endo, “Description and discussion on DCASE 2021 challenge task 2: Unsupervised anomalous detection for machine condition monitoring under domain shifted conditions,” in *DCASE Workshop*, 2021, pp. 186–190.

[10] K. Dohi, T. Nishida, H. Purohit, R. Tanabe, T. Endo, M. Yamamoto, Y. Nikaido, and Y. Kawaguchi, “MIMII DG: Sound dataset for malfunctioning industrial machine investigation and inspection for domain generalization task,” in *DCASE Workshop*, 2022.

[11] R. Giri, S. V. Tenneti, F. Cheng, K. Helwani, U. Isik, and A. Krishnaswamy, “Self-supervised classification for detecting anomalous sounds,” in *DCASE Workshop*, 2020, pp. 46–50.

[12] N. Harada, D. Niizumi, D. Takeuchi, Y. Ohishi, M. Yasuda, and S. Saito, “ToyADMOS2: Another dataset of miniature-machine operating sounds for anomalous sound detection under domain shift conditions,” in *DCASE Workshop*, 2021, pp. 1–5.

[13] N. Harada, N. Daisuke, T. Daiki, O. Yasunori, and Y. Masahiro, “First-shot anomaly detection for machine condition monitoring: a domain generalization baseline,” *arXiv preprint arXiv:2303.00455*, 2023.

[14] Y. Zhou and Y. Long, “Attribute classifier with imbalance compensation for anomalous sound detection,” DCASE2023 Challenge, Tech. Rep., June 2023.

[15] J. Jie, “Anomalous sound detection based on self-supervised learning,” DCASE2023 Challenge, Tech. Rep., June 2023.

[16] Z. Lv, B. Han, Z. Chen, Y. Qian, J. Ding, and J. Liu, “Unsupervised anomalous detection based on unsupervised pretrained models,” DCASE2023 Challenge, Tech. Rep., June 2023.

[17] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, “Smote: Synthetic minority over-sampling technique,” vol. 16, no. 1, 2002.

[18] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, “mixup: Beyond empirical risk minimization,” in *International Conference on Learning Representations*, 2018. [Online]. Available: https://openreview.net/forum?id=r1Ddp1-Rb

[19] K. Wilkinghoff, “Fraunhofer fkie submission for task 2: First-shot unsupervised anomalous sound detection for machine condition monitoring,” DCASE2023 Challenge, Tech. Rep., June 2023.

[20] J. Yafei, B. Jisheng, and H. Siwei, “Unsupervised abnormal sound detection based on machine condition mixup,” DCASE2023 Challenge, Tech. Rep., June 2023.

[21] W. JiaJun, “Self-supervised representation learning for first-shot unsupervised anomalous sound detection,” DCASE2023 Challenge, Tech. Rep., June 2023.

[22] T. Fujimura, I. Kuroyanagi, T. Hayashi, and T. Toda, “Anomalous sound detection by end-to-end training of outlier exposure and normalizing flow with domain generalization techniques,” DCASE2023 Challenge, Tech. Rep., June 2023.

[23] J. Tian, H. Zhang, Q. Zhu, F. Xiao, H. Liu, X. Mei, Y. Liu, W. Wang, and J. Guan, “First-shot anomalous sound detection with gmm clustering and finetuned attribute classification using audio pretrained model,” DCASE2023 Challenge, Tech. Rep., June 2023.

[24] A. Jiang, Q. Hou, J. Liu, P. Fan, J. Ma, C. Lu, Y. Zhai, Y. Deng, and W.-Q. Zhang, “Thuee system for first-shot unsupervised anomalous sound detection for machine condition monitoring,” DCASE2023 Challenge, Tech. Rep., June 2023.

[25] L. Wang, F. Chu, Y. Zhou, S. Wang, Z. Yan, S. Xu, Q. Wu, M. Cai, J. Pan, Q. Wang, J. Du, T. Gao, X. Fang, and L. Zou, “First-shot unsupervised anomalous sound detection using attribute classification and conditional autoencoder,” DCASE2023 Challenge, Tech. Rep., June 2023.

[26] Y. Zeng, H. Liu, and Y. Zhou, “General anomalous sound detection using sound event classification and detection,” DCASE2023 Challenge, Tech. Rep., June 2023.

[27] H. Liu, Z. Chen, Y. Yuan, X. Mei, X. Liu, D. Mandic, W. Wang, and M. D. Plumbley, “AudioLDM: Text-to-audio generation with latent diffusion models,” 2023.

POST-PROCESSING INDEPENDENT EVALUATION OF SOUND EVENT DETECTION SYSTEMS

Janek Ebbers, Reinhold Haeb-Umbach

Paderborn University,
Department of Communications Engineering,
33098 Paderborn, Germany,
{ebbers,haeb}@nt.upb.de

Romain Serizel

Université de Lorraine, CNRS,
Inria, Loria,
F-54000 Nancy, France,
romain.serizel@loria.fr

ABSTRACT

Due to the high variation in the application requirements of sound event detection (SED) systems, it is not sufficient to evaluate systems only in a single operating point. Therefore, the community recently adopted the polyphonic sound detection score (PSDS) as an evaluation metric, which is the normalized area under the PSD-ROC. It summarizes the system performance over a range of operating points. Hence, it provides a more complete picture of the overall system behavior and is less biased by hyper parameter tuning. So far PSDS has only been computed over operating points resulting from varying the decision threshold that is used to translate the system output scores into a binary detection output. However, besides the decision threshold there is also the post-processing that can be changed to enter another operating mode. In this paper we propose the post-processing independent PSDS (piPSDS) which computes PSDS over operating points with varying post-processings and varying decision thresholds. It summarizes even more operating modes of an SED system and allows for system comparison without the need of implementing a post-processing and without a bias due to different post-processings. While piPSDS can in principle also combine different types of post-processing, we here, as a first step, present median filter independent PSDS (miPSDS) results for this year's DCASE Challenge Task4a systems. Source code is publicly available in our sed_scores_eval package[1].

Index Terms— sound event detection, polyphonic sound detection, evaluation, post-processing, median filter

1. INTRODUCTION

Machine listening is recently attracting increased interest not only from academia but also from industry. It is the field of developing machines which can replicate the human ability of recognizing and processing a large number of different sounds. There are many sub-disciplines to machine listing, with sound event detection (SED) [1] being one of them. Its aim is to recognize, classify and temporally localize sounds within an input audio. Due to the large number of possible applications, sounds and environments, one particular challenge is that there is often no or only little training data that perfectly matches the target application. Therefore, there is a particular interest in approaches for model training which can exploit imperfect data, such as weakly labeled learning [2, 3] and/or training with mismatched or unlabeled data [4, 5], as investigated by the Detection and Classification of Acoustic Scenes and Events (DCASE) Challenge Task 4 [6] for several years now.

Another more fundamental challenge for successful SED system development is the meaningful evaluation and comparison of system performance, where the choice of the evaluation metric can have a large impact [7]. Firstly, there is the complexity of the event matching between detected and ground truth events. Currently there exist three different approaches namely segment-based, collar-based and intersection-based [8, 9]. The DCASE Challenge Task 4 recently moved to intersection-based evaluation as it is more robust w.r.t. ambiguities in the ground truth labeling. Secondly, due to the high variation in application requirements, there is often not a single optimal system behavior as, e.g., expressed by the F_1-score. In some applications, missed hits may, e.g., be much more severe than false alarms. Therefore, system evaluation must ideally represent all different operating modes equally to capture the overall system behavior. The polyphonic sound detection score (PSDS) [9, 10] has been employed to capture performance over the range of decision thresholds, which are used to translate soft system output scores[2] into binary decisions. Therefore, system comparison using PSDS is also less biased by threshold tuning w.r.t. to a certain operating point.

However, the post-processing [11] (e.g. median filtering), that is applied to the classifier output either before or after thresholding, has also a large impact on the system performance, which is mostly underinvestigated. In particular, system comparisons may be biased due to the employment of different post-processings. Also tuning of the post-processing hyper-parameters may overfit to a certain scenario while performing badly in mismatched scenarios which can give misleading information on the system itself. Similar to the decision threshold, the type and parameters of the post-processing can be understood as operating parameters of the system and may be adjusted to enter another operating mode which better suits the current scenario and application requirements.

In this paper we propose post-processing independent PSDS (piPSDS) which summarizes performance over both different post-processings and decision thresholds. Hence, it gives an even more complete picture of the system's performance over different operating modes and furthermore is less biased by hyper-parameter tuning. We perform investigations on this year's DCASE Challenge Task 4 submissions and show that 1) there is indeed a large impact on evaluation results due to post-processing 2) for different operating points there are different optimal post-processings and 3) the proposed piPSDS allows SED system evaluation unbiased from threshold and post-processing tuning.

The rest of the paper is structured as follows. First, we recapitulate the preliminaries of SED, its evaluation and the PSDS in Sec. 2.1, Sec. 2.2 and Sec. 2.3, respectively. Our proposed piPSDS is presented in Sec. 3. Finally, we show results in Sec. 4 and draw conclusions in Sec. 5.

Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) - 282835863.

[1] `https://github.com/fgnt/sed_scores_eval`

[2] Note the ambiguity of the term score here, where PSD score refers to a metric value while output scores refer to soft class activity predictions of a model/neural network.

2. PRELIMINARIES

2.1. Sound Event Detection

To not only recognize but also temporally localize sound events, SED systems perform multi-label classification within smaller time-windows of an audio clip, e.g., at short-time Fourier transform (STFT) frame-level. For each window n a system provides soft classification scores $y_{n,c}$ for each event class c out of a set of C predefined sound event classes of interest. These scores represent the predicted activity of the event within a particular time-window. To obtain a hard decision, soft classification scores can be binarized using a certain decision threshold γ_c, where the class c is assumed active in the n-th window if $y_{n,c} \geq \gamma_c$, else it is assumed inactive. Connected active windows are then merged into a detected event $(\hat{t}_{\text{on},i}, \hat{t}_{\text{off},i}, \hat{c}_i)$ defined by onset time $\hat{t}_{\text{on},i}$, offset time $\hat{t}_{\text{off},i}$ and class label $\hat{c}_i$, respectively, where i represents the event index. Usually it is beneficial to run some kind of post-processing before or after binarization to obtain meaningfully connected event predictions and be more robust w.r.t. outliers. Common post-processings are, e.g., median filtering [12] and Hidden Markov Model smoothing [13]. The type and hyper-parameters of the post-processing, as well as the decision threshold and any other hyper-parameters that may be easily changed during application are summarized as a system's operating parameters τ in the following.

2.2. Evaluation of Detected Events

The evaluation of the detected events of event class c for specific operating parameters τ is, in accordance with other classification tasks, based on counting the intermediate statistics $N_{\text{TP},c,\tau}$, $N_{\text{FN},c,\tau}$ and $N_{\text{FP},c,\tau}$, which refer to the numbers of

- ground truth (GT) events that have been correctly detected by the system a.k.a. true positive (TP) detections,
- GT events that have not been detected by the system a.k.a. false negative (FN) detections,
- detected events that do not match any GT event a.k.a. false positive (FP) detections,

accumulated over the whole evaluation set, respectively. Bilen et al. [9] have further taken cross triggers (CTs) into account, a.k.a. substitutions, with $N_{\text{CT},c,k,\tau}$ being the number of FPs of class c matching GT events from another event class k, which may impair user experience more than standalone FPs.

When counting above intermediate statistics, different approaches exist for the temporal matching between detected events and GT events. As the definitions of PSDS and piPSDS, however, do not depend on the temporal matching that is used, we here only briefly recap intersection-based evaluation which has recently been used for PSDS computation as it is more robust w.r.t. ambiguities in the labeling of the evaluation data. Note, however, that one could instead also compute segment-based and collar-based [8] (pi)PSDS.

Intersection-based evaluation requires detected events to intersect with GT events by at least a fraction ρ_{DTC} to be not counted as a FP detection. Moreover, it requires a GT event to intersect with non-FP events by at least a fraction ρ_{GTC} to be counted as a TP detection. Further, if an FP event intersects with a GT event of another class by at least a fraction ρ_{CTTC} it is counted as a CT.

Of particular interest are in the following the TP rate (TPR) defined as $r_{c,\tau} = \frac{N_{\text{TP},c,\tau}}{N_{\text{TP},c,\tau}+N_{\text{FN},c,\tau}}$, and the effective FP rate (eFPR)

$$e_{c,\tau} = \frac{N_{\text{FP},c,\tau}}{T_{\text{ds}}} + \alpha_{\text{CT}} \frac{1}{C-1} \sum_{\substack{k \\ k \neq c}} \frac{N_{\text{CT},c,k,\tau}}{T_k} \quad . \tag{1}$$

which consists of the FPR $\frac{N_{\text{FP},c,\tau}}{T_{\text{ds}}}$ plus an additional penalty on CT rates (CTRs) $\frac{N_{\text{CT},c,k,\tau}}{T_k}$ averaged over all other classes $k \neq c$ and weighted by α_{CT}. Note that, with intersection-based evaluation, there is not a countable number of negative events, which is why the FPR is computed w.r.t. the total duration of the evaluation dataset T_{ds}, whereas CTRs are computed w.r.t. the total duration of activity T_k of the k-th class within the evaluation dataset.

2.3. Polyphonic Sound Detection Score

To compute PSDS [9], one starts with the computation of single-class PSD-ROC curves $r_c(e)$ for each event class c. $r_c(e)$ is obtained as a continuous "staircase-type" interpolation of true positive rates $r_{c,\tau}$ plotted over corresponding eFPRs $e_{c,\tau}$ for different operating parameters $\tau \in \widehat{\mathcal{T}}_c$.

While τ may be any (set of) hyper-parameter(s) that may change system behavior, it has so far, in accordance with the standard definition of ROC curves [14], only been considered to be the decision threshold used to translate soft prediction scores into binary detections. Here, an algorithm for the efficient joint evaluation of all possible decision thresholds has been proposed in [10]. Note that, in contrast to standard ROC curves, it is here not always guaranteed that $r_{c,\tau}$ is monotonically increasing with $e_{c,\tau}$, when, e.g., sophisticated intersection-based evaluation is employed. As in operation, however, one would always prefer the operating point with a higher true positive rate at lower or equal false positive rate if available, $\widehat{\mathcal{T}}_c$ represents only best case operating parameters:

$$\widehat{\mathcal{T}}_c = \left\{\tau \middle| \nexists\, \lambda \,\text{with}\, e_{c,\lambda} \leq e_{c,\tau} \,\text{and}\, r_{c,\lambda} > r_{c,\tau}\right\}. \tag{2}$$

Having the single-class PSD-ROC curves $r_c(e)$, the overall PSD-ROC curve is defined as the effective true positive rate

$$r(e) = \mu_{\text{TP}}(e) - \alpha_{\text{ST}} \sigma_{\text{TP}}(e) \tag{3}$$

which is average per-class true positive rate minus a penalty on standard deviation over classes weighted by a metric parameter α_{ST} with

$$\mu_{\text{TP}}(e) = \frac{1}{C} \sum_{c=1}^{C} r_c(e); \quad \sigma_{\text{TP}}(e) = \sqrt{\frac{1}{C} \sum_{c=1}^{C} (r_c(e) - \mu_{\text{TP}}(e))^2}.$$

Finally, the PSDS is the normalized area under the PSD-ROC:

$$\text{PSDS} = \frac{1}{e_{\max}} \int_0^{e_{\max}} r(e)de \tag{4}$$

with the maximal false positive rate $e_{\max}$ being a metric parameter, which controls up to which false positive rate the operating points may still be relevant.

3. POST-PROCESSING INDEPENDENT POLYPHONIC SOUND DETECTION SCORE

Besides the decision threshold there is also the post-processing that we could change to enter another operating mode. As an example, Fig. 1 shows the single-class PSD-ROC curves for "Speech" from this year's "Baseline_BEATS" system [15] when using post-processing median filtering with lengths of $0.1\,\text{s}$ and $1.0\,\text{s}$, respectively. It appears that when the system is operated in low eFPR mode, than it is better to use the larger median filter window size. When the system should be operated in high TPR mode, it is better to use a smaller window size. Thus, it is reasonable and also fairly easy to choose the post-processing depending on the requirements of a given application. To account for this in the system evaluation, which is supposed to capture overall system behavior, we propose

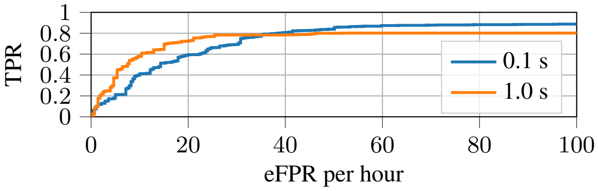


Figure 1: Baseline Speech ROCs with different median filters

to incorporate the variation of post-processing into the computation of the PSDS to get a post-processing independent PSDS (piPSDS).

To do so, we consider the operating parameters $\tau = (l, \gamma)$ to be a tuple of the post-processing l and the decision threshold γ. Here, l defines which postprocessing is used out of a predefined set of L possible post-processings.

The definition of the PSD-ROC according to Sec. 2.3 with $\tau \in \mathbb{L} \times \mathbb{R}$, where $\mathbb{L} = \{l \in \mathbb{N} \,|\, l < L\}$, gives us the pi-PSD-ROC. Due to the restriction to best case operating points in Eq. 2 the single-class pi-PSD-ROCs can be computed as

$$r_c(e) = \max_l r_{c,l}(e) \tag{5}$$

where $r_{c,l}(e)$ is the single-class PSD-ROC for a single post-processing l, which results from variation of the decision threshold and can be efficiently computed using the algorithm from [10]. Hence, the single-class pi-PSD-ROC chooses, for a given eFPR e, the post-processing with the highest TPR. piPSDS is then, analogously to Eq. 4, the normalized area under the pi-PSD-ROC.

Overall, piPSDS has two major advantages over only threshold-independent PSDS. Firstly, it better captures real-world SED applications, where it is natural to choose the post-processing that best suits the current application requirements. Secondly, for research it allows for system comparison without a bias being introduced by different post-processings.

4. RESULTS

Investigations are done with the baseline and submissions of this year's DCASE challenge Task4a. Participants have been asked to, in addition to their post-processed submission, also share the raw prediction scores as provided by their model/neural network without any further post-processing. This allows us to investigate 1) the impact of the post-processing, 2) post-processing independent evaluation. All following evaluations are performed on the DESED [16] public eval set, which is a part of the challenge evaluation data.

There are two intersection-based PSDS evaluated in the challenge, which refer to different scenarios. PSDS1 ($\rho_{\mathrm{DTC}} = 0.7$, $\rho_{\mathrm{GTC}} = 0.7$, $\alpha_{\mathrm{CT}} = 0$, $\alpha_{\mathrm{ST}} = 1$, $e_{\max} = 100/\mathrm{hour}$) particularly evaluates the model's capability of temporally localizing sound events, whereas PSDS2 ($\rho_{\mathrm{DTC}} = 0.1$, $\rho_{\mathrm{GTC}} = 0.1$, $\rho_{\mathrm{CTTC}} = 0.3$, $\alpha_{\mathrm{CT}} = 0.5$, $\alpha_{\mathrm{ST}} = 1$, $e_{\max} = 100/\mathrm{hour}$) is more focused on evaluating the reliable recognition of event classes within an audio clip. Due to space constraints and with post-processing being particularly relevant for the temporal localization of sound events, we only consider PSDS1 evaluation in the following.

With median filtering being the most popular type of post-processing for SED systems, we here consider median filter independent PSDS (miPSDS) as an instance of piPSDS, where the set of possible post-processings consists of median filters with different filter lengths. As the set of median filter lengths we use 21 filter lengths linearly spaced from 0.0 s (no filtering) to 1.0 s, 10 from 1.1 s to 2.0 s, 5 from 2.2 s to 3.0 s and 4 from 3.5 s to 5.0 s overall totaling 40 different filter lengths. The implementation of the median filter equals a time continuous filtering of a piece-wise constant signal that is defined by the timestamped prediction scores[3] submitted by the participants. This way it is ensured the systems employ the very same post-processing regardless of the system's output resolution which may vary across systems. Implementations of the median filter, miPSDS and piPSDS, with the latter taking any list of differently post-processed scores, are publicly available in the sed_scores_eval package[1], that is, in accordance with the challenge, used for evaluation.

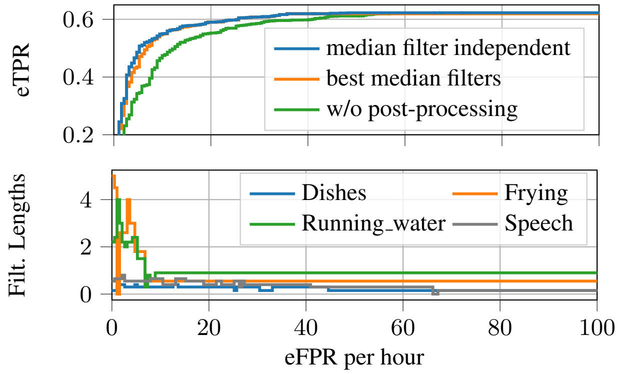


Figure 2: Upper Plot: PSD-ROCS for different post-processing setups. Lower Plot: Optimal median filter lengths over operating points as tracked by median filter independent PSD-ROC.

We first run investigations on the baseline system Baseline_BEATS [15] (Baseline). In the upper subplot of Fig. 2 we compare the following PSD-ROCs:

1. median filter independent: as defined in Eq. 3
2. best median filters: choosing best performing median filter per class as follows
   $\tilde{r}_c(e) = r_{c,b}(e)$ with $b = \mathrm{argmax}_l \mathrm{auc}(r_{c,l}(e))$,
3. without any post-processing.

It can be seen, that by applying (best) median filtering the PSD-ROC can be significantly improved over the unprocessed case. It can be further observed, that there are operating points, especially for low eFPRs, where the mi-PSD-ROC (mi-PSD-ROC) is higher than best median filter PSD-ROC. This indicates that best median filters are, although giving best overall performance, not the best choice for each individual operating point and better performance can be achieved by choosing operating point dependent filter lengths as the mi-PSD-ROC does. In the lower subplot of Fig. 2 we plot, for some event classes, the optimal filter lengths over operating points. We can see that for lower eFPRs optimal median filters tend to be longer than for higher eFPRs, which can be explained by the fact that longer median filters better suppress short duration FPs. Further, event classes with longer per-event durations, such as "Frying" and "Running Water", tend to have overall longer median filters than short duration event classes, which makes intuitively sense.

Next, we evaluate challenge submissions[4] with, without and independent of post-processing. As submitting unprocessed scores was optional, we evaluate only systems from the 12 teams that did provide them. We limit evaluation to the one single-model system per team that gave best PSDS1 performance in the challenge (with original post-processing). These systems

---

[3] Note that median filtering and thresholding are permutation invariant, i.e., applying the median filter before binarization yields the same result as applying it afterwards

[4] `https://zenodo.org/record/8248774`

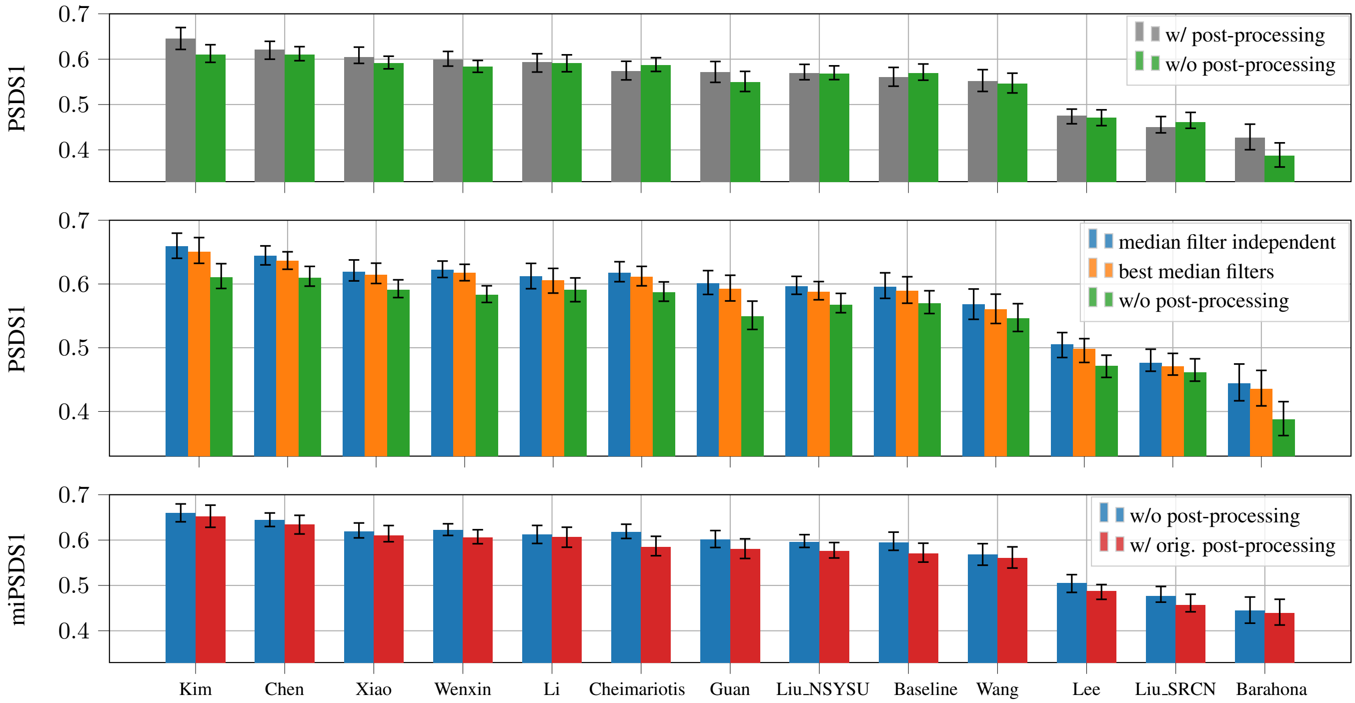


Figure 3: System Evaluation. Upper Plot: original post-processing vs. no post-processing. Middle Plot: miPSDS vs. PSDS with optimal median filter lengths per class vs. no post-processing. Lower Plot: miPSDS computed with unprocessed vs. post-processed data.

are Barahona-AUDIAS-2 [17], Cheimariotis-DUTH-1 [18], Chen-CHT-2 [19], Guan-HIT-3 [20], Kim-GIST-HanwhaVision-2 [21], Lee-CAUET-1 [22], Li-USTC-6 [23], Liu-NSYSU-7 [24], Liu-SRCN-4 [25], Wang-XiaoRice-1 [26], Wenxin-TJU-6 [27], Xiao-FMSG-4 [28].

To be able to evaluate the variance of system performance over different runs of system training, participants submitted prediction scores for three runs of training for each system. To further track variance of results due to variations in the evaluation data, we perform bootstrapped evaluation, where evaluation is performed on 20 different $80\,\%$ fractions of the eval data. In total we evaluate $3 \cdot 20 = 60$ different setups and report the mean and $5\,\% - 95\,\%$ confidence interval of the system's performances. This evaluation procedure is the same as we used for official challenge evaluation.

We first want to investigate the impact of the post-processing on the systems' performances in the upper subplot of Fig. 3. by comparing the performance with and without the post-processing as used by the participants. It appears that for some systems, e.g., Kim-GIST-HanwhaVision-2, the performance significantly degrades when removing the post-processing, whereas for other systems the performance does not degrade or even improves. When evaluating the unprocessed scores, the ranking also changes at multiple positions to Kim, Chen, Li, Xiao, Cheimariotis, Wenxin, Baseline, Liu_NSYSU, Guan, Wang, Lee, Liu_SRCN, Barahona. This suggests that there is some bias introduced by the post-processing, particularly, whether a sophisticated post-processing is employed or not. To some extent, however, it may also be a system property that it can benefit from post-processing more than other systems.

We next evaluate our proposed miPSDS and compare it to ”no processing” and ”best median filters” in the middle subplot of Fig. 3. It can be seen that for all systems performance can be improved by best median filters and further improved by operating point specific median filters as considered by miPSDS. Some systems, e.g., Kim and Barahona, benefit more from best median filters / median filter independent evaluation than others, which can be explained by our previous assumption that the effectiveness of post-processing is to some extent also a system property. Here, miPSDS evaluation gives again a different ranking which is Kim, Chen, Wenxin, Xiao, Cheimariotis, Li, Guan, Liu_NSYSU, Baseline, Wang, Lee, Liu_SRCN, Barahon.

Note, that it is still possible to run additional post-processing before piPSDS evaluation to improve performance. However, it can be assumed that the possible gain is rather small and it is more likely that an additional post-processing degrades piPSDS. To investigate this, we compare miPSDS evaluated on unprocessed scores vs. scores with participants' original post-processing in the lower subplot of Fig. 3. It can be seen that in all cases the additional post-processing degrades miPSDS performance.

5. CONCLUSIONS

Due to the high variation of SED system application requirements, SED evaluation has to capture the overall system behavior over various operating points. Therefore, the community recently moved to decision threshold independent evaluation using PSDSs to capture performance over different decision thresholds used for binarization of system output scores. In this paper we proposed piPSDS which further evaluates performance over different post-processings and effectively choosing the post-processing that is best suited for a certain operating mode. It has been shown that piPSDS indeed overcomes the bias introduced due to different post-processings but still accounts for system-specific effectiveness of post-processing. It further allows for system comparison without the need of employing a sophisticated post-processing, e.g., during system development.

6. REFERENCES

[1] A. Mesaros, T. Heittola, T. Virtanen, and M. D. Plumbley, "Sound event detection: A tutorial," *IEEE Signal Processing Magazine*, vol. 38, no. 5, pp. 67–83, 2021.

[2] Y. Wang, J. Li, and F. Metze, "A comparison of five multiple instance learning pooling functions for sound event detection with weak labeling," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2019, pp. 31–35.

[3] K. Miyazaki, T. Komatsu, T. Hayashi, S. Watanabe, T. Toda, and K. Takeda, "Weakly-supervised sound event detection with self-attention," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2020, pp. 66–70.

[4] L. JiaKai, "Mean teacher convolution system for dcase 2018 task 4," Detection and Classification of Acoustic Scenes and Events Challenge, Tech. Rep., September 2018.

[5] J. Ebbers and R. Haeb-Umbach, "Pre-training and self-training for sound event detection in domestic environments," DCASE2022 Challenge, Tech. Rep., June 2022.

[6] DCASE 2023 Challenge Task 4a description. [Online]. Available: https://dcase.community/challenge2023/task-sound-event-detection-with-weak-labels-and-synthetic-soundscapes

[7] G. Ferroni, N. Turpault, J. Azcarreta, F. Tuveri, R. Serizel, Ç. Bilen, and S. Krstulović, "Improving sound event detection metrics: insights from dcase 2020," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2021, pp. 631–635.

[8] A. Mesaros, T. Heittola, and T. Virtanen, "Metrics for polyphonic sound event detection," *Applied Sciences*, vol. 6, no. 6, p. 162, 2016.

[9] Ç. Bilen, G. Ferroni, F. Tuveri, J. Azcarreta, and S. Krstulović, "A framework for the robust evaluation of sound event detection," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2020, pp. 61–65.

[10] J. Ebbers, R. Haeb-Umbach, and R. Serizel, "Threshold independent evaluation of sound event detection scores," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2022, pp. 1021–1025.

[11] E. Benetos, D. Stowell, and M. D. Plumbley, "Approaches to complex sound scene analysis," *Computational analysis of sound scenes and events*, pp. 215–242, 2018.

[12] E. Cakir, T. Heittola, H. Huttunen, and T. Virtanen, "Polyphonic sound event detection using multi label deep neural networks," in *Proc. International Joint Conference on Neural Networks*, 2015, pp. 1–7.

[13] S. Cornell, M. Olvera, M. Pariente, G. Pepe, E. Principi, L. Gabrielli, and S. Squartini, "Domain-adversarial training and trainable parallel front-end for the dcase 2020 task 4 sound event detection challenge," in *Proc. Workshop on Detection and Classification of Acoustic Scenes and Events*, 2020.

[14] J. Davis and M. Goadrich, "The relationship between precision-recall and roc curves," in *Proc. 23rd international conference on Machine learning*. ACM Press, 2006, pp. 233–240.

[15] DCASE 2023 Challenge Task 4a baseline. [Online]. Available: https://github.com/DCASE-REPO/DESED_task/tree/master/recipes/dcase2023_task4_baseline

[16] N. Turpault, R. Serizel, A. P. Shah, and J. Salamon, "Sound event detection in domestic environments with weakly labeled data and soundscape synthesis," in *Proc. Workshop on Detection and Classification of Acoustic Scenes and Events*, 2019.

[17] S. Barahona, D. de Benito-Gorron, S. Segovia, D. Ramos, and D. Toledano, "Optimizing multi-resolution conformer and crnn models for different PSDS scenarios in DCASE challenge 2023 task 4a," DCASE2023 Challenge, Tech. Rep., June 2023.

[18] G.-A. Cheimariotis and N. Mitianoudis, "Sound event detection of domestic activities using frequency dynamic convolution and BEATS embeddings," DCASE2023 Challenge, Tech. Rep., June 2023.

[19] W.-Y. Chen, C.-L. Lu, H.-F. Chuang, Y.-H. C. Cheng, and B.-C. Chan, "Sound event detection system using pre-trained model for dcase 2023 task 4," DCASE2023 Challenge, Tech. Rep., June 2023.

[20] Y. Guan and Q. Shang, "Semi-supervised sound event detection system for DCASE 2023 task 4," DCASE2023 Challenge, Tech. Rep., June 2023.

[21] J. W. Kim, S. W. Son, Y. Song, . Kim, Hong Kook1, I. H. Song, and J. E. Lim, "Semi-supervised learning-based sound event detection using frequency dynamic convolution with large kernel attention for DCASE challenge 2023 task 4," DCASE2023 Challenge, Tech. Rep., June 2023.

[22] S. Lee, N. Kim, J. Lee, C. Hwang, S. Jang, and I.-Y. Kwak, "Sound event detection using convolution attention module for DCASE 2023 challenge task4a," DCASE2023 Challenge, Tech. Rep., June 2023.

[23] K. Li, P. Cai, and Y. Song, "Li USTC team's submission for DCASE 2023 challenge task4a," DCASE2023 Challenge, Tech. Rep., June 2023.

[24] C.-C. Liu, T.-H. Kuo, C.-P. Chen, C.-L. Lu, B.-C. Chan, Y.-H. Cheng, and H.-F. Chuang, "Cht+nsysu sound event detection system with pretrained embeddings extracted from beats model for dcase 2023 task 4," DCASE2023 Challenge, Tech. Rep., June 2023.

[25] M. Chen, Y. Jin, J. Shao, Y. Liu, B. Peng, and J. Chen, "DCASE 2023 challenge task4 technical report," DCASE2023 Challenge, Tech. Rep., June 2023.

[26] Y. Wang, H. Dinkel, Z. Yan, J. Zhang, and Y. Wang, "Pepe: Plain efficient pretrained embeddings for sound event detection," DCASE2023 Challenge, Tech. Rep., June 2023.

[27] X. Duo, Wenxin1 Fang and J. Li, "Semi-supervised sound event detection system for DCASE 2023 task4a," DCASE2023 Challenge, Tech. Rep., June 2023.

[28] Y. Xiao, T. Khandelwal, and R. K. Das, "FMSG submission for DCASE 2023 challenge task 4 on sound event detection with weak labels and synthetic soundscapes," DCASE2023 Challenge, Tech. Rep., June 2023.

TOYADMOS2+: NEW TOYADMOS DATA AND BENCHMARK RESULTS OF THE FIRST-SHOT ANOMALOUS SOUND DETECTION BASELINE

Noboru Harada, Daisuke Niizumi, Yasunori Ohishi, Daiki Takeuchi, Masahiro Yasuda

NTT Corporation, Japan
noboru.harada.pv@hco.ntt.co.jp

ABSTRACT

This paper introduces the newly recorded ToyADMOS dataset for the DCASE 2023 Challenge Task 2, First-shot anomalous sound detection for machine condition monitoring (DCASE2023T2). New machine types, such as ToyDrone, ToyNscale, Vacuum, and ToyTank, were newly recorded as a part of the Additional training and Evaluation datasets. This paper also shows benchmark results of the First-shot baseline implementation (with simple autoencoder and selective Mahalanobis modes) on the DCASE2023T2 Evaluation dataset and the previous DCASE Challenge Task 2 datasets in 2020, 2021, and 2022, compared with the baselines of those years.

Index Terms— DCASE 2023 Challenge Task 2, First-Shot Anomalous sound detection, ToyADMOS dataset

1. INTRODUCTION

In recent years, exhaustive research has been done on anomalous sound detection (ASD) for machine condition monitoring. Several challenge tasks related to ASD were organized in the Detection and Classification of Acoustic Scenes and Events (DCASE).

The first ASD challenge was DCASE 2020 Challenge Task 2: Unsupervised Detection of Anomalous Sounds for Machine Condition Monitoring (DCASE2020T2) [1]. The task setting required systems to have only the normal samples of machine operating sound for training. No anomalous sound sample was available for training because getting enough number of anomaly samples is extremely difficult in real application scenarios. The organizers of DCASE2020T2 had to create specific datasets for the task by intentionally adding damage to machines, e.g., toys. The datasets are called ToyADMOS and MIMII dataset [2, 3]. Since then, the ASD challenge task has been extended to represent more realistic application scenarios, such as DCASE 2021 Challenge Task 2: Unsupervised Anomalous Sound Detection for Machine Condition Monitoring under Domain Shifted Conditions (DCASE2021T2) [4], and DCASE 2022 Challenge Task 2: Unsupervised Anomalous Sound Detection for Machine Condition Monitoring Applying Domain Generalization Techniques (DCASE2022T2) [5]. ToyADMOS2, MIMII DUE, and MIMII DG datasets had been developed [6, 7, 8].

In DCASE 2023 Challenge Task 2: "First-shot Anomalous Sound Detection for Machine Condition Monitoring" (DCASE2023T2) [9], new datasets and a baseline implementation complying with the First-shot requirements were introduced [10, 6, 8, 11]. First-shot means the system can use only the given training data for the target. The First-shot ASD is characterized as follows:

- No use of data from different machine instances (not given)
- No hyperparameter tuning nor tool ensemble enabled for dedicated machine type (by analyzing ground truth results with both normal and anomaly samples)

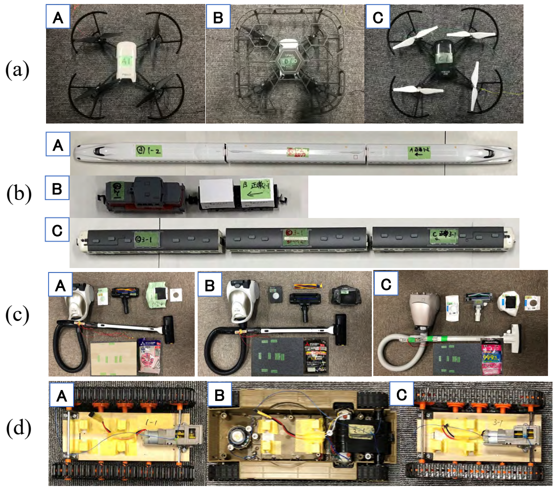


Figure 1: Images of toy-model configurations A, B, and C for (a) ToyDrone, (b) ToyNscale, (c) Vacuum, and (d) ToyTank.

This paper describes the newly added ToyADMOS data for the Additional training and Evaluation datasets of the DCASE2023T2. In addition, this paper also shows benchmark results of the First-shot baseline implementation [12] on the previous DCASE Challenge Task 2 datasets in 2020, 2021, 2022, and 2023, compared with the baselines of those years [13, 14, 15, 16, 17]. The source code and data are available at GitHub [12] and Zenodo [9, 18, 19].

2. TOYADMOS2+: ADDITIONAL DATA FOR THE DCASE2023 CHALLENGE TASK 2

To provide the First-shot training data for DCASE2023T2, the following four machine types, (a) ToyDrone, (b) ToyNscale, (c) Vacuum, and (d) ToyTank, were newly recorded. Each machine type has three model configurations (A, B, C) shown in Fig. 1. The machine operating sounds were recorded with the room layout and the microphone settings shown in Figs. 2 and 3.

ToyDrone: The ToyDrone flies in a guide frame as shown in Figs. 2(a) and 3(a). There are three flying patterns.

ToyNscale: The N-scale toy train runs on a railway track. Sound data were collected with eight microphones surrounding the track, as shown in Figs. 2(b) and 3(b).

Vacuum: The Vacuum is set on a guide frame in Figs. 2(c) and 3(c). A floor mat or a wooden floor plate was used.

ToyTank: The ToyTank runs in a guide frame as shown in Figs. 2(d) and 3(d).

To generate anomaly samples, some of the parts were intentionally damaged. Anomaly conditions for each machine type are shown in Table 1.

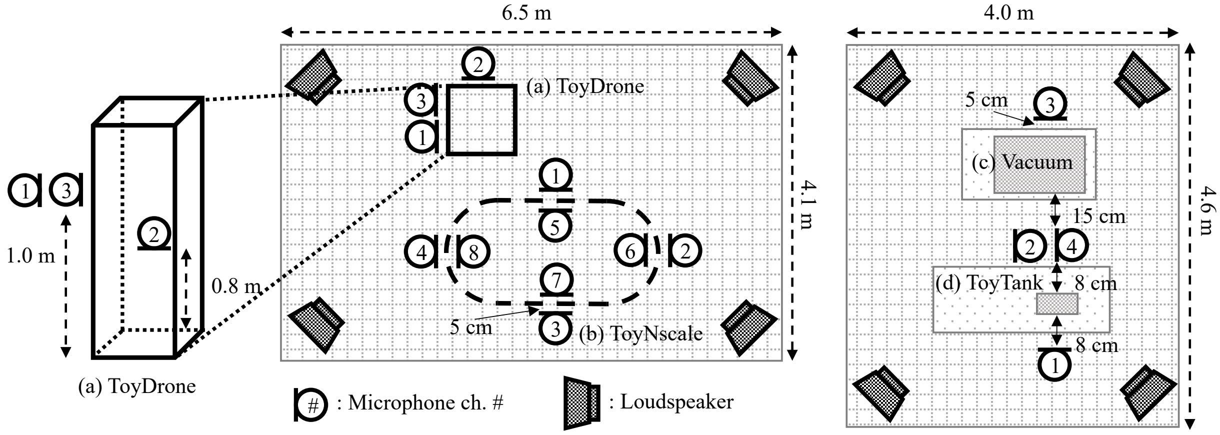


Figure 2: Recording-room layouts and microphone arrangements, (a) ToyDrone, (b) ToyNscale, (c) Vacuum, and (d) ToyTank.

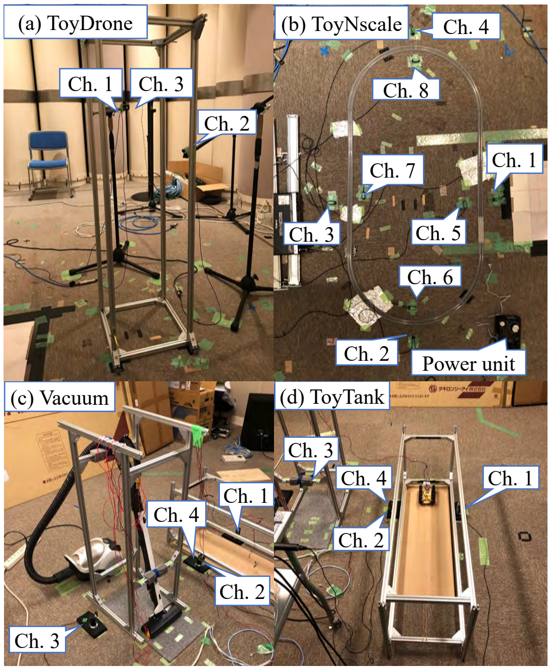


Figure 3: Images of microphone arrangements (a) ToyDrone, (b) ToyNscale, (c) Vacuum, and (d) ToyTank.

Table 2 shows the recording setting. All the operating sound and noise samples were recorded with 48 kHz sampling, 24-bit for each channel, and then downsampled to 16 kHz, 24-bit, monaural. Sample duration varies from 6 sec to 18 sec, depending on the machine type, as shown in Table 2. Domain shift conditions were controlled by changing machine instances (ID), operating speed, mic position, and mixed background noise samples. Table 3 shows the domain shift conditions of source and target domains.

For training data (Additional training dataset), there are 1000 normal samples given for training, where 990 samples are from the source domain, and 10 samples are from the target domain. For evaluation data (Evaluation dataset), 50 normal and 50 anomaly samples are from each source and target domain. In total, there are 200 samples for each machine type. In total, 960 min of data was prepared. The data is available at the Zenodo links [18, 19] under the Creative Commons Attribution 4.0 International Public License [20].

Table 1: Anomaly conditions for each machine type.

(a) **ToyDrone**		(b) **ToyNscale**	
Part	Condition	Part	Condition
Propeller	- Cut one side	Carriage	- Flat tire
	- Cut two sides		- Broken shaft
Guard	- Guard missing	Railway track	- Disjointed
Unbalance	- Offset weight		- Obstructing stone

(c) **Vacuum**		(d) **ToyTank**	
Part	Condition	Part	Condition
Air filter	- No filter	Trackbelt	- Damaged belt
Nozzle	- Thread jam		- Foreign object
Dust bag	- Bag full	Trackbelt wheel	- Wheel lock
	- No dust bag		- Missing wheel

Table 2: Recording conditions for DCASE2023T2 Eval. dataset.

	ToyDrone	ToyNscale	Vacuum	ToyTank
Model variations	A, B, C	A, B, C	A, B, C	A, B, C
Speed levels	Three patterns	Five	Lb, M, Mb, H^{*1}	Five
Mic. config	Ch. 1 - 3	Ch. 1 - 8	Ch. 1 - 4	Ch. 1 - 4
Noise type^{*2}	N1	N1	N2	N2
Sample duration	18 sec	6 sec	15 sec	8 sec

^{*1}*L, M, H* mean Lo, Middle, and High power, and *Lb, Mb* means with a brush on.
^{*2}*N1*: windy outside noise and *N2*: room air conditioning noise.

Table 3: Domain shift settings for DCASE2023T2 Eval. dataset.

Source domain	ToyDrone	ToyNscale	Vacuum	ToyTank
Machine ID	A, B	A, B	A, B	A, B
Speed	2, 3	2, 3	Lb, M, Mb, H^{*1}	3, 4
Mic.	Dynamic 1	Dynamic 1 - 4	Dynamic 1	Dynamic 1
Noise^{*2}	N1	N1	N2	N2
Target domain	**ToyDrone**	**ToyNscale**	**Vacuum**	**ToyTank**
Machine ID	C	C	C	C
Speed	1	1, 5	Mb, H^{*1}	1, 5
Mic.	Dynamic 2	Dynamic 5 - 8	Dynamic 2	Dynamic 1
Noise^{*2}	N1	N1	N2	N2

^{*1}*L, M, H* mean Lo, Middle, and High power, and *Lb, Mb* means with a brush on.
^{*2}*N1*: windy outside noise and *N2*: room air conditioning noise.

3. BENCHMARK RESULTS WITH THE DCASE2023T2 FIRST-SHOT BASELINE

3.1. The DCASE2023T2 First-shot baseline

DCASE2023 Challenge Task 2 baseline has the following two operating modes. For the details, see Sec. 3.2 and [11].

Table 4: AUC results of the DCASE 2023 Challenge Task 2 Evaluation dataset.

System	metric	hmean^{*1}	amean^{*1}	**ToyDrone**	**ToyNscale**	**ToyTank**	**Vacuum**	**Bandsaw**	**Grinder**	**Shaker**
First-shot compliant simple Autoencoder mode (FS-AE) [12]	AUC (source)	0.7421	0.7484	0.8121	0.7650	0.7196	0.8815	0.6710	0.7039	0.6859
	AUC (target)	0.5357	0.5436	0.5375	0.4891	0.6655	0.4488	0.5264	0.5590	0.5790
	pAUC (src & tgt)	0.5551	0.5575	0.5316	0.5223	0.5998	0.6040	0.5098	0.5858	0.5489
	TOTAL score	**0.5981**	**0.6165**							
Selective Mahalanobis AE mode [12]	AUC (source)	0.7877	0.7934	0.8495	0.6684	0.8085	0.8031	0.8328	0.7396	0.8519
	AUC (target)	0.5377	0.5707	0.4218	0.4377	0.4879	0.8727	0.6014	0.5169	0.6566
	pAUC (src & tgt)	0.5722	0.5779	0.5140	0.5107	0.5470	0.6795	0.5684	0.6033	0.6227
	TOTAL score	**0.6151**	**0.6474**							
DCASE2022T2 baseline AE [16]	AUC (source)	0.7732	0.7835	0.9018	0.7705	0.8325	0.9047	0.6708	0.7051	0.6993
	AUC (target)	0.3644	0.4049	0.2611	0.3940	0.3430	0.2397	0.4554	0.5494	0.5920
	pAUC (src & tgt)	0.5323	0.5346	0.4970	0.5112	0.5807	0.5116	0.5023	0.5789	0.5605
	TOTAL score	**0.5071**	**0.5744**							
cf. DCASE2023T2 Top 1 Jie_IESEFPT_task2_2 [21] (operating condition label classification)	AUC (source)	0.8313	0.8377	0.8026	0.9042	0.8480	0.9690	0.7655	0.7367	0.8376
	AUC (target)	0.6008	0.6444	0.4544	0.8768	0.4682	0.9548	0.5749	0.6082	0.5734
	pAUC (src & tgt)	0.6203	0.6399	0.5158	0.7774	0.6153	0.8532	0.5335	0.6245	0.5597
	TOTAL score	**0.6697**	**0.7073**							

^{*1}*hmean* denotes harmonic mean, and *amean* denotes arithmetic mean.

First-shot-compliant simple Autoencoder mode (FS-AE): This is a simple autoencoder. For training, the model parameter θ of the AE is trained to minimize the mean square error (MSE) between a normal input sample x^- and its reconstruction $\hat{x}^-$ using

$$Loss \quad = MSE(x^-, \hat{x}^-), \qquad (1)$$

$$where \quad \hat{x}^- = Dec_\theta(Enc_\theta(x^-)). \qquad (2)$$

For the testing phase, the anomaly score A_θ is calculated with the reconstruction error of the given query sample x using

$$Anomaly\ Score\ \ A_\theta \quad = MSE(x, \hat{x}), \qquad (3)$$

$$where\ \hat{x} \quad = Dec_\theta(Enc_\theta(x)). \qquad (4)$$

When the anomaly score exceeds the pre-set threshold, the sample is detected as an anomaly sample.

Selective Mahalanobis Autoencoder mode: The anomaly score A_θ is calculated using the covariance matrixes Σ_s^{-1} and Σ_t^{-1} of distance between normal samples x^- and its reconstruction $\hat{x}^-$ for the source and target domains with:

$$Anomaly\ Score\ \ A_\theta \quad = min\{D_s(x, \hat{x}), D_t(x, \hat{x})\}, \qquad (5)$$

$$where\ \ D_s(\cdot) \quad = Mahalanobis(x, \hat{x}, \Sigma_s^{-1}), \qquad (6)$$

$$D_t(\cdot) \quad = Mahalanobis(x, \hat{x}, \Sigma_t^{-1}). \qquad (7)$$

3.2. Experimental setup and evaluation criterion

For the First-shot compliant baseline, the model hyperparameters were set to the values described in [11].

The frame size for STFT was 64 ms with 50 % hop size translated into 128 frequency bands Log-mel energies. Five consecutive frames were concatenated to formulate 640 dimensions (128×5) as input to the system. In the autoencoder model, there were three layers of 128 dimensions linear, Batch normalization, and Activation with ReLU each, in encoder and decoder. The bottleneck layer had eight dimensions. The number of epochs for training was 100. The batch size was 256, and the Adam optimizer used a 0.001 learning ratio.

The performances of the two operating modes of the DCASE2023T2 baseline [11, 12] were compared with the previous baselines, such as DCASE2020T2 baseline AE [13], DCASE2021T2 baseline AE [14], DCASE2021T2 baseline MobileNetV2AE [15], DCASE2022T2 baseline AE [16], and DCASE2022T2 MobileNetV2 [17]. The Evaluation datasets and baseline systems dedicated to the DCASE Challenges were used. For those baseline systems, hyperparameters were set to the ones used in the corresponding previous DCASE Challenges [1, 4, 5, 10, 9, 22, 23, 24]. The total scores Ω for evaluating the systems are calculated based on Area Under the Receiver Operating Characteristic (ROC) curve (AUC) and partial AUC (pAUC) with a harmonic mean ($hmean$) and an arithmetic mean ($amean$) of AUC and pAUC. All the results were the averaged score of systems trained with three different random seeds except for Jie_IESEFPT_task2_2 [21] that were copied from the official score.

3.3. Experimental results

Experimental results of the DCASE2023T2 First-shot baseline on the DCASE2023T2 Evaluation datasets are shown in Table 4. Some other results of the DCASE2023T2 baseline compared with the previous baseline systems on the Evaluation datasets of DCASE 2020, 2021, and 2022 Challenge tasks are shown in Tables 5, 6, and 7, respectively. The DCASE2023T2 baseline with the selective Mahalanobis AE mode performed better than others in the DCASE2021T2, DCASE2022T2, and DCASE2023T2. For DCASE2020T2, the DCASE2023T2 baseline with the FS-AE mode performs better because of no domain shift. The DCASE2023T2 baseline can be a performance benchmark for the tasks.

4. CONCLUSION

This paper introduces the newly recorded ToyADMOS dataset for the DCASE 2023 Challenge Task 2, First-shot anomalous sound detection for machine condition monitoring. New machine types, such as ToyDrone, ToyNscale, Vacuum, and ToyTank, are newly recorded as a part of the Additional training and Evaluation datasets. This paper also shows benchmark results of the First-shot baseline implementation (with simple autoencoder and selective Mahalanobis modes) on the previous DCASE Challenge Task 2 datasets in 2020, 2021, 2022, and 2023, compared with the baselines of those years. ToyADMOS2+ dataset (DCASE 2023 Challenge Task 2 Additional Training Dataset and Evaluation Dataset) is available at [9, 18, 19] with the Creative Commons Attribution 4.0 International Public License [20]. The updated source code of the DCASE2023T2 baseline supporting all the previous task settings is available at [12].

5. ACKNOWLEDGMENT

The authors thank Mr. Hideaki Sanada, Mr. Akito Karasawa, Mr. Keisuke Hasegawa, and Mr. Yuri Musashijima for their hard work implementing the systems and conducting experiments.

Table 5: AUC results of the DCASE 2020 Challenge Task 2 Evaluation dataset.

System	metric	hmean[*1]	amean[*1]	**ToyCar**	**ToyConvayor**	**fan**	**pump**	**slider**	**valve**
First-shot compliant simple Autoencoder mode (FS-AE) [12]	AUC	0.7817	0.8003	0.8399	0.7859	0.8601	0.8618	0.7975	0.6567
	pAUC	0.6187	0.6342	0.6936	0.6651	0.6903	0.6499	0.5872	0.5191
	TOTAL score	**0.6907**	**0.7173**						
Selective Mahalanobis AE mode [12]	AUC	0.7434	0.7845	0.8101	0.7727	0.9478	0.9175	0.7280	0.5305
	pAUC	0.6115	0.6366	0.5892	0.6700	0.7886	0.7333	0.5292	0.5095
	TOTAL score	**0.6710**	**0.7105**						
DCASE2020T2 baseline AE [13]	AUC	0.7774	0.7959	0.7973	0.8822	0.8593	0.8408	0.8148	0.5810
	pAUC	0.6189	0.6312	0.6630	0.7075	0.6768	0.6443	0.5869	0.5085
	TOTAL score	**0.6892**	**0.7135**						

[*1] *hmean* denotes harmonic mean, and *amean* denotes arithmetic mean.

Table 6: AUC results of the DCASE 2021 Challenge Task 2 Evaluation dataset.

System	metric	hmean[*1]	amean[*1]	**ToyCar**	**ToyTrain**	**fan**	**gearbox**	**pump**	**slider**	**valve**
First-shot compliant simple Autoencoder mode (FS-AE) [12]	AUC (source)	0.6421	0.6506	0.7049	0.7009	0.6818	0.6605	0.6336	0.6483	0.5245
	AUC (target)	0.5716	0.5817	0.6211	0.5808	0.5850	0.6278	0.5699	0.5592	0.5279
	pAUC (source)	0.5224	0.5244	0.5232	0.5265	0.5225	0.5541	0.5175	0.5166	0.5104
	pAUC (target)	0.5206	0.5253	0.5680	0.5082	0.5081	0.5712	0.5098	0.5189	0.4926
	TOTAL score	**0.5601**	**0.5705**							
Selective Mahalanobis AE mode [12]	AUC (source)	0.6488	0.6696	0.8438	0.5677	0.7773	0.6581	0.6774	0.6593	0.5033
	AUC (target)	0.5664	0.5885	0.6560	0.4706	0.6060	0.6333	0.6284	0.5849	0.5401
	pAUC (source)	0.5418	0.5507	0.6896	0.4937	0.5601	0.5485	0.5408	0.5063	0.5159
	pAUC (target)	0.5279	0.5328	0.6009	0.5093	0.5308	0.5717	0.5067	0.5101	0.5001
	TOTAL score	**0.5676**	**0.5854**							
DCASE2021T2 baseline AE [14]	AUC (source)	0.6468	0.6556	0.7490	0.7141	0.6624	0.6736	0.6312	0.6444	0.5146
	AUC (target)	0.5693	0.5834	0.6232	0.6451	0.5568	0.6330	0.5612	0.5408	0.5236
	pAUC (source)	0.5272	0.5305	0.5409	0.5660	0.5176	0.5516	0.5143	0.5149	0.5082
	pAUC (target)	0.5318	0.5386	0.5692	0.5959	0.4982	0.5807	0.5101	0.5219	0.4945
	TOTAL score	**0.5650**	**0.5770**							
DCASE2021T2 baseline MobileNetV2 [15]	AUC (source)	0.5351	0.6034	0.4279	0.5215	0.7505	0.5620	0.7013	0.7246	0.5360
	AUC (target)	0.5236	0.5892	0.5800	0.3852	0.6396	0.4889	0.7107	0.7517	0.5681
	pAUC (source)	0.5569	0.5736	0.5299	0.5312	0.6467	0.5522	0.6342	0.6060	0.5149
	pAUC (target)	0.5617	0.5780	0.6505	0.4921	0.6288	0.4962	0.6236	0.6241	0.5304
	TOTAL score	**0.5431**	**0.5860**							

[*1] *hmean* denotes harmonic mean, and *amean* denotes arithmetic mean.

Table 7: AUC results of the DCASE 2022 Challenge Task 2 Evaluation dataset.

System	metric	hmean[*1]	amean[*1]	**ToyCar**	**ToyTrain**	**fan**	**gearbox**	**bearing**	**slider**	**valve**
First-shot compliant simple Autoencoder mode (FS-AE) [12]	AUC (source)	0.6515	0.6803	0.8331	0.4998	0.6562	0.7116	0.7479	0.7594	0.5539
	AUC (target)	0.5123	0.5418	0.6537	0.5243	0.3580	0.6026	0.5621	0.5264	0.5657
	pAUC (src & tgt)	0.5344	0.5420	0.6658	0.4973	0.5136	0.5152	0.5474	0.5355	0.5192
	TOTAL score	**0.5599**	**0.5880**							
Selective Mahalanobis AE mode [12]	AUC (source)	0.6650	0.7138	0.9401	0.5031	0.6820	0.8510	0.7195	0.7637	0.5371
	AUC (target)	0.5557	0.5999	0.7965	0.5075	0.4055	0.7724	0.6280	0.5595	0.5296
	pAUC (src & tgt)	0.5623	0.5763	0.7738	0.5067	0.5133	0.6071	0.5763	0.5408	0.5164
	TOTAL score	**0.5903**	**0.6300**							
DCASE2022T2 baseline AE [16]	AUC (source)	0.6478	0.6762	0.7749	0.5973	0.6433	0.7006	0.7281	0.7548	0.5346
	AUC (target)	0.4451	0.4771	0.4761	0.3915	0.3200	0.5762	0.5321	0.4931	0.5505
	pAUC (src & tgt)	0.5263	0.5314	0.5939	0.4969	0.5046	0.5204	0.5519	0.5377	0.5146
	TOTAL score	**0.5272**	**0.5616**							
DCASE2022T2 baseline MobileNetV2 [17]	AUC (source)	0.5758	0.6535	0.6457	0.5340	0.6914	0.5996	0.6333	0.7692	0.7014
	AUC (target)	0.4542	0.5248	0.4621	0.5916	0.3842	0.4433	0.5239	0.5052	0.7629
	pAUC (src & tgt)	0.5345	0.5413	0.5419	0.5046	0.5354	0.4820	0.5151	0.5498	0.6601
	TOTAL score	**0.5163**	**0.5732**							

[*1] *hmean* denotes harmonic mean, and *amean* denotes arithmetic mean.

6. REFERENCES

[1] Y. Koizumi, Y. Kawaguchi, K. Imoto, T. Nakamura, Y. Nikaido, R. Tanabe, H. Purohit, K. Suefusa, T. Endo, M. Yasuda, and N. Harada, "Description and discussion on DCASE2020 challenge task2: Unsupervised anomalous sound detection for machine condition monitoring," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2020 Workshop (DCASE2020)*, November 2020, pp. 81–85. [Online]. Available: http://dcase.community/documents/workshop2020/proceedings/DCASE2020Workshop_Koizumi_3.pdf

[2] Y. Koizumi, S. Saito, H. Uematsu, N. Harada, and K. Imoto, "ToyADMOS: A dataset of miniature-machine operating sounds for anomalous sound detection," in *Proceedings of IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, November 2019, pp. 308–312. [Online]. Available: https://ieeexplore.ieee.org/document/8937164

[3] H. Purohit, R. Tanabe, T. Ichige, T. Endo, Y. Nikaido, K. Suefusa, and Y. Kawaguchi, "MIMII Dataset: Sound dataset for malfunctioning industrial machine investigation and inspection," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2019 Workshop (DCASE2019)*, November 2019, pp. 209–213. [Online]. Available: http://dcase.community/documents/workshop2019/proceedings/DCASE2019Workshop_Purohit_21.pdf

[4] Y. Kawaguchi, K. Imoto, Y. Koizumi, N. Harada, D. Niizumi, K. Dohi, R. Tanabe, H. Purohit, and T. Endo, "Description and discussion on dcase 2021 challenge task 2: Unsupervised anomalous detection for machine condition monitoring under domain shifted conditions," in *Proceedings of the 6th Detection and Classification of Acoustic Scenes and Events 2021 Workshop (DCASE2021)*, Barcelona, Spain, November 2021, pp. 186–190.

[5] K. Dohi, K. Imoto, N. Harada, D. Niizumi, Y. Koizumi, T. Nishida, H. Purohit, R. Tanabe, T. Endo, M. Yamamoto, and Y. Kawaguchi, "Description and discussion on dcase 2022 challenge task 2: Unsupervised anomalous sound detection for machine condition monitoring applying domain generalization techniques," in *Proceedings of the 7th Detection and Classification of Acoustic Scenes and Events 2022 Workshop (DCASE2022)*, Nancy, France, November 2022.

[6] N. Harada, D. Niizumi, D. Takeuchi, Y. Ohishi, M. Yasuda, and S. Saito, "ToyADMOS2: Another dataset of miniature-machine operating sounds for anomalous sound detection under domain shift conditions," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events Workshop (DCASE)*, Barcelona, Spain, November 2021, pp. 1–5.

[7] R. Tanabe, H. Purohit, K. Dohi, T. Endo, Y. Nikaido, T. Nakamura, and Y. Kawaguchi, "MIMII DUE: Sound dataset for malfunctioning industrial machine investigation and inspection with domain shifts due to changes in operational and environmental conditions," *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pp. 21–25, 2021.

[8] K. Dohi, T. Nishida, H. Purohit, R. Tanabe, T. Endo, M. Yamamoto, Y. Nikaido, and Y. Kawaguchi, "MIMII DG: Sound dataset for malfunctioning industrial machine investigation and inspection for domain generalization task," in *Proceedings of the 7th Detection and Classification of Acoustic Scenes and Events 2022 Workshop (DCASE2022)*, Nancy, France, November 2022.

[9] "DCASE 2023 challenge task 2: First-shot unsupervised anomalous sound detection for machine condition monitoring," 2023, https://dcase.community/challenge2023/task-first-shot-unsupervised-anomalous-sound-detection-for-machine-condition-monitoring.

[10] K. Dohi, K. Imoto, N. Harada, D. Niizumi, Y. Koizumi, T. Nishida, H. Purohit, R. Tanabe, T. Endo, and Y. Kawaguchi, "Description and discussion on DCASE 2023 challenge task 2: First-shot unsupervised anomalous sound detection for machine condition monitoring," *In arXiv e-prints: 2305.07828*, 2023.

[11] N. Harada, D. Niizumi, D. Takeuchi, Y. Ohishi, and M. Yasuda, "First-shot anomaly detection for machine condition monitoring: A domain generalization baseline," *In arXiv e-prints: 2303.00455*, 2023.

[12] N. Harada, Y. Musashijima, and D. Niizumi, "dcase2023_task2_baseline_ae," 2023, https://github.com/nttcslab/dcase2023_task2_baseline_ae.

[13] Y. Kawaguchi, "dcase2020_task2_baseline," 2020, https://github.com/y-kawagu/dcase2020_task2_baseline.

[14] ——, "dcase2022_task2_baseline_ae," 2021, https://github.com/y-kawagu/dcase2021_task2_baseline_ae.

[15] ——, "dcase2022_task2_baseline_mobile_net_v2," 2021, https://github.com/y-kawagu/dcase2021_task2_baseline_mobile_net_v2.

[16] K. Dohi, "dcase2022_task2_baseline_ae," 2022, https://github.com/Kota-Dohi/dcase2022_task2_baseline_ae.

[17] ——, "dcase2022_task2_baseline_mobile_net_v2," 2022, https://github.com/Kota-Dohi/dcase2022_task2_baseline_mobile_net_v2.

[18] "DCASE 2023 challenge task 2 additional training dataset," 2023, https://zenodo.org/record/7830345.

[19] "DCASE 2023 challenge task 2 evaluation dataset," 2023, https://zenodo.org/record/7860847.

[20] "Creative commons attribution 4.0 international public license," https://creativecommons.org/licenses/by/4.0/legalcode.

[21] J. Jie, "Anomalous sound detection based on self-supervised learning," DCASE2023 Challenge, Tech. Rep., June 2023.

[22] "DCASE 2020 challenge task 2: Unsupervised detection of anomalous sounds for machine condition monitoring," 2020, https://dcase.community/challenge2020/task-unsupervised-detection-of-anomalous-sounds.

[23] "DCASE 2021 challenge task 2: Unsupervised anomalous sound detection for machine condition monitoring under domain shifted conditions," 2021, https://dcase.community/challenge2021/task-unsupervised-detection-of-anomalous-sounds.

[24] "DCASE 2022 challenge task 2: Unsupervised anomalous sound detection for machine condition monitoring applying domain generalization techniques," 2022, https://dcase.community/challenge2022/task-unsupervised-anomalous-sound-detection-for-machine-condition-monitoring.

EVALUATING CLASSIFICATION SYSTEMS AGAINST SOFT LABELS WITH FUZZY PRECISION AND RECALL

Manu Harju, Annamaria Mesaros

Computing Sciences, Tampere University, Tampere, Finland

ABSTRACT

Classification systems are normally trained by minimizing the cross-entropy between system outputs and reference labels, which makes the Kullback-Leibler divergence a natural choice for measuring how closely the system can follow the data. Non-binary references can arise from various sources, and it is often beneficial to use the soft labels for training instead of the binarized data. In addition to the cross-entropy based measures, precision and recall provide another perspective for measuring the performance of a classification system. However, the existing definitions for precision and recall require binary reference labels, and binarizing the data can cause erroneous interpretations and loss of information about the underlying data distributions. We present a novel method to calculate precision, recall and F-score without quantizing the data. The proposed metrics are based on fuzzy theory and extend the well established metrics, as the definitions coincide when used with binary labels. To understand the behavior of the proposed metrics we show numerical example cases and an evaluation of different sound event detection models trained on real data with soft labels.

Index Terms— soft labels, soft precision and recall, sound event detection

1. INTRODUCTION

The target labels in classification tasks are usually presented in one-hot or multi-hot encoded form, indicating target classes being either present or absent. This leads naturally to using binary representations for the data; however, nothing prevents us for using non-binary values. The terms hard and soft labels are often used to make the distinction between the binary and non-binary cases. Soft labels can be derived from binary data e.g. by using label smoothing [1] or data augmentation with mixup [2]. Furthermore, non-binary values can be used to represent uncertainty in the original data. Using soft labels to present the annotators confidence can improve the model performance [3, 4] and help with ambiguous classes [5].

One particular case is in the DCASE 2023 Challenge, where the organizers provide soft labels in the sound event detection (SED) task. The data was collected by splitting approximately 3-minute long recordings into 10-second clips with a one-second stride, and the clips were annotated on Amazon Mechanical Turk for temporally weak labels. Each clip was annotated by five different annotator, and therefore a single one-second segment in the middle part in the longer files got annotation from 50 workers. The collected labels were used to estimate competence values for the individual annotators, which were then used to compute weighted averages of the annotator opinions, with the resulting numbers designated as the *soft labels* for one-second segments of audio [6].

The challenge task is about training a sound event detection system using the soft labels, to investigate if leveraging information from the soft labels is beneficial for the acoustic models. However, the evaluation is done using hard labels and hard metrics. Converting soft labels into binary requires choosing a threshold value, and finding a good one is not a trivial task. The most straightforward way is to use 0.5 as the threshold, and this is also how the reference data for the challenge is binarized. However, as a consequence, six event classes out of 17 are left out from the evaluation, as there are not enough segments with a soft label value above the threshold.

Converting the soft reference labels into binary means discarding part of the information about the data. On the other hand, using soft labels for training was shown to improve accuracy and yield more structure for the more ambiguous classes in the learned feature space [5]. Furthermore, training models with soft labels was shown to improve the performance on out-of-sample data and to make the models more robust against adversarial attacks [3].

In [7], soft labels have been used for training a SED, and the Kullback-Leibler divergences in the results indicate that the system output is closer to the reference labels than for the case when the same system is trained using the hard labels. Furthermore, using the midpoint of the value range as the threshold for each class independently improves the performance on less represented classes. Nevertheless, if the system performance is measured from binary references, we are missing out the uncertainty levels in the data. In an extreme case such harsh quantization renders the data unusable by converting all the labels into the same value.

In this paper we propose a novel method to calculate precision and recall from softly-labeled data. The proposed metrics coincide with the hard metrics when used with binary labels, and thus can be seen as an extension of the hard precision and recall for handling soft references and system outputs. Using the soft labels directly allows measuring the system's ability to track the reference data, taking the metrics closer to the cross-entropy based ones. However, the soft precision and recall also provide similar information about the bias of the classifier as the hard metrics.

Measuring precision and recall requires counting set sizes for the predicted outputs and the references. For example, in multilabel classification, for a single input sample we may have a set of predicted labels and a set of reference labels. Previous work for soft metrics involve the idea of the set elements having soft boundaries for several different scenarios. Such elements may overlap with each other, making measuring the set size more difficult. The work in [8] presents a number of examples, including comparing sets of words and audio segmentation. In the word set comparison, syntactic and semantic similarities are measured between words to determine the overlap and *soft cardinality* of the sets. In the segmentation example the onset and offset times are from continuous scale but the segments themselves are hardly labeled. In [9], audio segments are converted into fuzzy sets with soft boundaries and fuzzy

---

This work was supported by Academy of Finland grant 332063 "Teaching machines to listen".

set theory is used to compute fuzzy precision and recall. However, in all these examples, the elements activities in the prediction and reference sets are binary. In contrast, the elements of interest in our case are well defined audio segments with clear boundaries, but the intensities of the predictions and reference data, i.e. the soft labels, come from the unit interval.

This paper is organized as follows: Section 2 introduces the theoretical definitions of the soft precision and recall and a few numerical examples for understanding the behavior of the metrics in various scenarios. Section 3 shows a comparison of hard and soft metrics using the DCASE 2023 Challenge SED with soft labels task data, and the method and results are discussed in Section 4. Finally, Section 5 presents conclusions and future work.

2. SOFT PRECISION AND RECALL

2.1. Definitions

Let X be the set of all elements to be classified, and $L \subseteq X$, $G \subseteq X$ be the sets of predicted labels and ground truth references, respectively. In the binary case, the sets L and G contain the positively labeled elements. Precision and recall can be defined using set theoretical notation [8] with:

$$\begin{aligned} \text{Precision} &= \frac{|L \cap G|}{|L|}, \\ \text{Recall} &= \frac{|L \cap G|}{|G|}, \\ \text{F}_1 - \text{score} &= 2\frac{|L \cap G|}{|L| + |G|}. \end{aligned} \tag{1}$$

A fuzzy set is defined by its membership function $\mu : X \to [0, 1]$ mapping each element of X to its grade of membership. We interpret L and G as fuzzy sets through the soft labels: given the reference label value y_i and prediction $\hat{y}_i$ for the i'th item x_i, the membership function values are:

$$\begin{aligned} \mu_L(x_i) &= \hat{y}_i, \\ \mu_G(x_i) &= y_i. \end{aligned} \tag{2}$$

We use the standard definition of intersection for fuzzy sets, namely the minimum of the membership function values. Finally, the fuzzy set cardinality is just the sum of the membership values, resulting in the following definitions for the soft precision, recall, and F_1-score

$$\begin{aligned} \text{Precision} &= \frac{\sum_i \min(\hat{y}_i, y_i)}{\sum_i \hat{y}_i}, \\ \text{Recall} &= \frac{\sum_i \min(\hat{y}_i, y_i)}{\sum_i y_i}, \\ \text{F}_1 - \text{score} &= 2\frac{\sum_i \min(\hat{y}_i, y_i)}{\sum_i (\hat{y}_i + y_i)}. \end{aligned} \tag{3}$$

It is possible to use some other T-norm for the fuzzy intersection, but the advantage of using minimum is its idempotency, i.e. $\min(x, x) = x$. From idempotency it follows that if $\hat{y}_i = y_i$ for any value in the unit interval, then also the precision and recall for that item will be equal to 1.

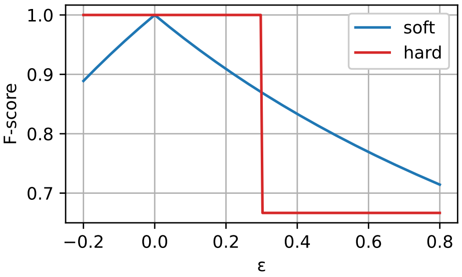


Figure 1: F-scores for a two-point example, comparing a 0.2 reference with a $0.2 + \varepsilon$ prediction.

2.2. Examples

Table 1 shows three small example cases. For each case, four data points are illustrated as a bar plot, where the blue bars represent predictions and the red bars correspond to reference labels. The dashed grey line denotes the threshold value of 0.5. All three examples contain 50 % of positive references in the binarized case. The table presents for each case the hard and soft metrics, and the last column shows Kullback-Leibler divergence for a measure of difference between the predictions and references.

For the first two examples the reference labels are the same and they correspond to a more clear annotation of categories. In the first case, half of the predicted positives are correct, and half of the positive references are found; in the second case, both positive references are found. For these two examples, the values for the hard metrics are as expected. However, the third example represents a more ambiguous case, where the labels are close to the 0.5 threshold value. In such a case the label quantization may turn the prediction and reference into complete opposites, resulting in very low scores despite the small difference between the system outputs and the reference labels. Furthermore, according to the KL-divergence the predictions are closest to the references in the last example, even though the hard metrics indicate the opposite.

To get a better insight about the behavior of the metric, we construct an example of two data points. The first data point is a true positive with value 0.8 for both prediction and reference. For the second point we set the reference label to 0.2 and use $0.2 + \varepsilon$ for our prediction. The effect of perturbing the prediction around the ground truth can be seen in Figure 1. The figure shows that the soft F-score has a sharp peak at the correct value. For $\varepsilon > 0.3$ the hard metric interprets the label as incorrect, while the soft metrics still finds some common mass between the prediction and reference label, resulting in higher soft F-score than the hard one.

3. EXPERIMENTAL RESULTS

3.1. System evaluations

We use the baseline system from the DCASE 2023 Challenge task 4B and two different modifications to explore the behavior of the proposed metrics. The baseline system is a standard CRNN with three convolutional blocks and one bidirectional GRU layer of size 32. Each convolutional block consists of a single 2D convolution with 128 filters and 3x3 kernels followed by batch normalization, rectified linear unit, max pooling, and dropout layers. The second system, denoted TinyCRNN, has two convolutional blocks, the number of filters is reduced to 32, and the GRU size is halved in

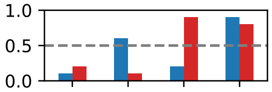
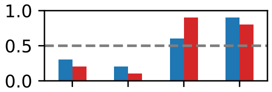
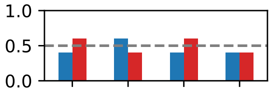
	Hard			Soft			
Data points	P	R	F	P	R	F	KLD
	50.0	50.0	50.0	66.7	60.0	63.2	0.446
	100.0	100.0	100.0	85.0	85.0	85.0	0.083
	0.0	0.0	0.0	88.9	80.0	84.2	0.061

Table 1: Simple example cases. Blue is predicted value, red is the reference label value. The dashed grey line is the threshold value at 0.5. The predictions are closest to the reference in the third case, even though the classical hard metrics indicate the opposite.

comparison to the baseline. The third system, denoted DWSNet, uses a similar GRU layer and the same number of blocks as the baseline, but each block consists of four depthwise separable convolutions. The number of parameters for the baseline, TinyCRNN, and DWSNet models are 380k, 28k, and 215k, respectively. The models are trained with soft labels and regression setup using the code provided with the baseline model.

In addition to the trained systems, we compute scores for three data-driven sampling methods for generating a randomized system output based on the training data. In the first case we fit beta distributions for each class in the training data and produce the system output by sampling from each distribution individually. The second sampled output is generated in a similar manner to the first one, but the fitted distributions are shuffled so that the classes are assigned a randomly chosen distributions of other classes. The original data comes from five different scenes with partially overlapping sets of classes, and the labels for absent classes are set to zero. As a result, the fitted distributions are biased towards zero, and for a single segment the method can output labels that are never seen co-occurring in the data. The third output is generated by sampling individual time steps from the training data, i.e. taking all outputs for all the classes at the same time.

For an extensive analysis, we use two different hard label evaluation methods that are supplied with the task baseline code. In the first case the system outputs are binarized using 0.5 as the threshold value. In the second case the system output values are automatically class-wise thresholded for the best macro F-score [10]. We call this method *optimal threshold* (OT). The soft metrics are calculated over the set of 11 classes included in the challenge evaluation for a direct comparison with the above-mentioned metrics, and also for all the 17 classes that are present in the training data.

Table 2 shows micro and macro F-scores of the trained systems and sampling methods. Micro F-score (F_m) is the global average over all the data points, whereas macro F-score (F_M) is the unweighted average of the class-wise F-scores. The numbers in the table are jackknife estimates of means over ten runs. The confidence intervals are omitted from the table for brevity. In addition to the F-scores, we include the Kullback-Leibler divergences to have another perspective for the system outputs. The results show that the soft metrics correlate well with the hard metrics. However, the soft metrics tend to give higher values since all the common content between the predictions and the references is taken into account. Both soft and hard F-scores agree that DWSNet has the best performance, with the exception of the hard macro F-score, for which the confidence intervals of DWSNet and baseline overlap. Similarly, all the F-scores for TinyCRNN are lower than those of the baseline. This ranking of the trained systems is also in agreement with the KL-divergences. Furthermore, the soft F-scores are also placing the random methods in the same order as the KL-divergence. The OT F-scores for all three random cases are close to each other, therefore it is difficult to draw any conclusion on their difference.

Interestingly, the OT F-score is very similar irrespective of the beta sampling method. The data consists of five different scenes having different sets of classes, whereas the sampling is done independently for each class throughout all the data. As expected, shuffling the order of the distributions takes the predicted labels even further from the references, as shown by the KLD values. The hard and soft F-scores are lower for the shuffled case, but because the optimal threshold follows the distribution of the predictions, the OT F-score fails to indicate that these predictions are worse than the class-wise beta ones.

The confidence intervals of the optimal threshold hard metric and the soft metric are shown in Figure 2. The figure indicates that the soft F-scores are more stable than the optimal threshold ones, having narrower CIs, particularly for the micro F-score. The OT method is optimizing threshold values for each class individually, which is causing some variations in the micro F-scores. In addition, there is some overlapping in the confidence intervals of the OT F-scores.

3.2. Random system output

We generate several different random predictions from arbitrarily chosen distributions to investigate the stability of the measures. The predicted labels are sampled from beta distributions with $\alpha = \beta = r$ with $r \in \{0.01, 0.1, 1, 5, 20\}$ and evaluated against the challenge reference data. Increasing the parameter r lowers variance and makes the probability mass more concentrated around the middle point. In addition to the random samples we also include a system output of a constant value 0.5. In all the cases, the distribution is symmetric and has expectation of 0.5.

Due to the symmetry, the hard F-score gives the same result for

	Betas		Shuffled betas		Data sampling		Baseline		TinyCRNN		DWSNet	
	F_m	F_M	F_m	F_M	F_m	F_M	F_m	F_M	F_m	F_M	F_m	F_M
Hard	23.2	9.1	7.2	4.6	30.9	10.9	70.4	35.0	64.8	26.2	72.8	35.7
Hard / OT	27.4	18.0	26.4	18.1	25.0	17.8	51.7	41.5	49.2	39.7	60.2	43.6
Soft	35.5	23.0	17.1	14.1	39.0	25.9	72.6	65.5	69.4	60.2	77.0	70.3
Soft (all classes)	33.3	18.6	15.7	12.6	36.8	21.1	70.4	58.8	66.8	53.2	75.1	64.5
KLD	0.746		1.138		0.074		0.034		0.044		0.028	

Table 2: F-scores and KL-divergences for the system outputs measured against the reference labels. F_m and F_M denote micro F-score and macro F-score, respectively.

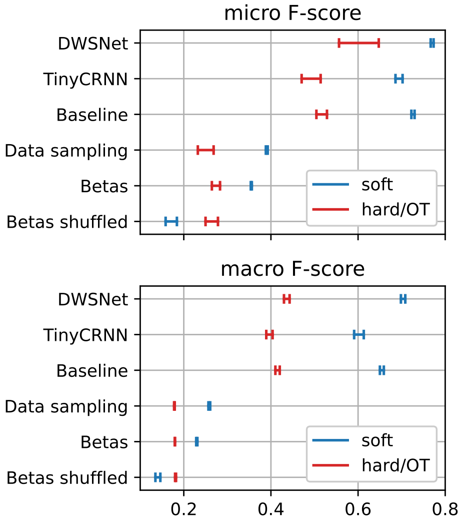


Figure 2: 95% confidence intervals for the average F-score of the trained models and generated random system outputs.

all the distributions, and is therefore left out from the comparison. Figure 3 shows that also the OT hard macro F-score stays constant, but optimizing the class-wise thresholds introduces some variation to the micro F-score. In contrast to the hard F-scores, the soft F-score decreases as the KL-divergence between the predictions and reference labels increases. Despite that the setup is very particular and the generated random samples are far from a real model output, this experiment shows that there are cases where the hard metrics are not able to make the distinction between system outputs with different KL-divergences.

4. DISCUSSION

The hard metrics work well if the reference labels are clearly clustered into positives and negatives. However, when the label values come from a wider scale, converting the data into binary can make the predictions look worse than they are. In practice, the errors from converting the data into binary are not very likely to mislead in the overall interpretation of the results, since averaging over a large number of segments will smooth out the values. However, a non-informative evaluation result is still possible if most of the data lie close to the threshold value. Nevertheless, hard metrics are not useful if we want to measure how well a system learns the uncertainties in the reference labels.

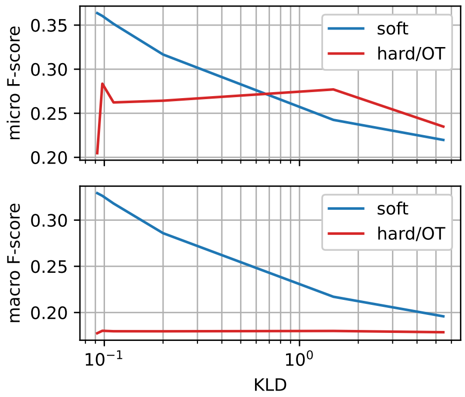


Figure 3: F-score against KL-divergence for the random system predictions evaluated against real data reference labels.

The soft metrics tend to give higher values than the hard ones since all the segments with nonzero reference or nonzero prediction contribute to the total scores. However, even if the soft F-scores are pushed higher in comparison with the hard ones, the differences between the trained systems are more visible, as the confidence intervals are separated by wider margins.

Finally, the definitions of the soft precision and recall make them differentiable, which makes it possible to train a model using a loss based on the soft metrics. Based on brief experiments with the baseline model, the soft F-score based loss did not bring any significant advantage compared to the MSE loss used in the baseline. Nevertheless, this direction may require more extensive experiments and system architecture design that was out of the scope of this work.

5. CONCLUSIONS

This paper introduced soft definitions for the precision and recall based on fuzzy sets, extending the classical hard metrics to evaluate system outputs against soft labels. The experiments show that the proposed metrics correlate well with the hard F-scores, and ranking system outputs according to the soft F-score agrees with the ranking based on KL-divergence. Furthermore, the confidence intervals of the results suggest that the soft F-score is more stable than the optimal threshold method. In future work, we plan to investigate the behavior of the proposed metrics with more real system outputs along with the existing metrics. In addition, the soft precision and recall based loss for model training could be studied in more detail.

6. REFERENCES

[1] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2818–2826.

[2] H. Zhang, M. Cissé, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," in *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, 2018.

[3] J. C. Peterson, R. M. Battleday, T. L. Griffiths, and O. Russakovsky, "Human uncertainty makes classification more robust," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 9616–9625.

[4] Q. Nguyen, H. Valizadegan, and M. Hauskrecht, "Learning classification models with soft-label information," *Journal of the American Medical Informatics Association*, vol. 21, pp. 501–508, 2014.

[5] V. Grossmann, L. Schmarje, and R. Koch, "Beyond hard labels: Investigating data label distributions," in *Proceedings of the 39th International Conference on Machine Learning (ICML)*, 2022.

[6] I. Martín-Morató and A. Mesaros, "Strong labeling of sound events using crowdsourced weak labels and annotator competence estimation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 31, pp. 902–914, 2023.

[7] I. Martín-Morató, M. Harju, P. Ahokas, and A. Mesaros, "Training sound event detection with soft labels from crowdsourced annotations," in *ICASSP 2023 - IEEE International Conference on Acoustics, Speech and Signal Processing*, 2023.

[8] P. Fränti and R. Mariescu-Istodor, "Soft precision and recall," *Pattern Recognition Letters*, vol. 167, pp. 115–121, 2023.

[9] B. Ziółko, "Fuzzy precision and recall measures for audio signals segmentation," *Fuzzy Sets and Systems*, vol. 279, pp. 101–111, 2015, theme: Data, Audio and Image Analysis.

[10] J. Ebbers, R. Haeb-Umbach, and R. Serizel, "Threshold independent evaluation of sound event detection scores," in *ICASSP 2022 - IEEE International Conference on Acoustics, Speech and Signal Processing*, May 2022.

META-SELD: META-LEARNING FOR FAST ADAPTATION TO THE NEW ENVIRONMENT IN SOUND EVENT LOCALIZATION AND DETECTION

Jinbo Hu[1,2], *Yin Cao*[3], *Ming Wu*[1], *Feiran Yang*[1], *Ziying Yu*[1], *Wenwu Wang*[4],
Mark D. Plumbley[4], *Jun Yang*[1,2]

[1]Key Laboratory of Noise and Vibration Research, Institute of Acoustics,
Chinese Academy of Sciences, Beijing, China,
{hujinbo, mingwu, feiran, yuziying, jyang}@mail.ioa.ac.cn
[2]University of Chinese Academy of Sciences, Beijing, China
[3]Department of Intelligent Science, Xi'an Jiaotong Liverpool University, China, yin.k.cao@gmail.com
[4]Centre for Vision, Speech and Signal Processing (CVSSP), University of Surrey, UK,
{w.wang, m.plumbley}@surrey.ac.uk

ABSTRACT

For learning-based sound event localization and detection (SELD) methods, different acoustic environments in the training and test sets may result in large performance differences in the validation and evaluation stages. Different environments, such as different sizes of rooms, different reverberation times, and different background noise, may be reasons for a learning-based system to fail. On the other hand, acquiring annotated spatial sound event samples, which include onset and offset time stamps, class types of sound events, and direction-of-arrival (DOA) of sound sources is very expensive. In addition, deploying a SELD system in a new environment often poses challenges due to time-consuming training and fine-tuning processes. To address these issues, we propose Meta-SELD, which applies meta-learning methods to achieve fast adaptation to new environments. More specifically, based on Model Agnostic Meta-Learning (MAML), the proposed Meta-SELD aims to find good meta-initialized parameters to adapt to new environments with only a small number of samples and parameter updating iterations. We can then quickly adapt the meta-trained SELD model to unseen environments. Our experiments compare fine-tuning methods from pre-trained SELD models with our Meta-SELD on the Sony-TAU Realistic Spatial Soundscapes 2023 (STARSSS23) dataset. The evaluation results demonstrate the effectiveness of Meta-SELD when adapting to new environments.

Index Terms— SELD, MAML, unseen environments, fast adaptation, meta-learning, few-shot

1. INTRODUCTION

Sound event localization and detection (SELD) refers to detecting categories, presence, and spatial locations of different sound sources. SELD characterizes sound sources in a spatial-temporal manner. SELD was first introduced in Task 3 of the Detection and Classification of Acoustics Scenes and Events (DCASE) 2019 Challenge [1]. After three iterations of Task 3 of the DCASE Challenge, types of data transform from computationally generated spatial recordings to real-scene recordings [2].

SELD can be regarded as a Multi-Task Learning problem. Adavanne et al. [3] proposed SELDnet for a joint task of sound event detection (SED) and regression-based direction-of-arrival (DOA) estimation. SELDnet is unable to detect homogeneous overlap, which refers to overlapping sound events of the same type but with different locations. The Event-Independent Network V2 (EINV2), with a track-wise output format and permutation invariant training, was proposed to tackle the homogeneous overlap detection problem [4–6]. Different from two outputs of SED and DOA in SELDnet and EINV2, the Activity-coupled Cartesian DOA (ACCDOA) approach merges two subtasks into a single task [7, 8]. The Cartesian DOA vectors contain the activity information of sound events in the ACCDOA method.

In practical SELD system deployment, unseen complex environments may lead to performance degradation. In the STARSS22 dataset [2], there are no duplicated recording environments in the training and validation sets. Our previous system submitted to Task 3 of the DCASE 2022 Challenge obtained the second rank in the team ranking [9]. However, we found unsatisfactory generalization performance for fold4_room2 recordings in the *dev-test-tau* set of STARSS22 [9]. Experimental results show that class-dependent localization error $\mathrm{LE}_{\mathrm{CD}}$ is high and location-dependent F-score $\mathrm{F}_{\leq 20^\circ}$ is low, but class-dependent localization recall $\mathrm{LR}_{\mathrm{CD}}$ is high. This suggests there may be the weak localizing performance of our system in fold4_room2. In addition, manually annotated spatial sound event recordings are very expensive. Taking the STARSS22 dataset for example [2], each scene was captured with a 32-channel spherical microphone array, a 360° camera, a motion capture (mocap) system, and wireless microphones. Onset, offset, and class information of sound events were manually detected and classified by annotators through listening to wireless microphone recordings and watching video recordings, while positional annotations were extracted for each event by masking the tracker data with the temporal activity window of the event. In the end, 360° video recordings are utilized to validate those annotations. This type of complex recording and annotation process means that large datasets of the annotated spatial recording might be expensive.

Few-shot learning can act as a test bed for learning like humans, allowing a system to learn from small samples and reducing data gathering effort and computation [10]. Meta-learning, which facilitates few-shot learning, learns a general-purpose learning algorithm that generalizes across tasks and ideally enables each new task to be learned well from the task-distribution view [11]. Meta-learning has advanced few-shot learning significantly in computer

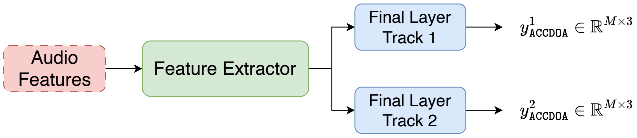

Audio Features
Feature Extractor
Final Layer Track 1
Final Layer Track 2
$y^1_{\texttt{ACCDOA}} \in \mathbb{R}^{M\times 3}$
$y^2_{\texttt{ACCDOA}} \in \mathbb{R}^{M\times 3}$

Figure 1: The multi-ACCDOA representation of the SELD model. There is no track dimension in the ACCDOA representation.

vision [12,13]. One of the most successful meta-learning algorithms is model-agnostic meta-learning (MAML) [14]. MAML tries to learn general initial parameters that can be rapidly adapted to another task. The method is model-agnostic and compatible with any model trained with gradient descent. It can be applicable to a variety of different learning problems, including classification, regression, and reinforcement learning. In audio signal processing, the meta-learning method has recently attracted interest as a way to solve few-shot learning problems recently. Meta-TTS [15] is proposed to build personalized speech synthesis systems with few enrolled recordings of unseen users' voices using MAML. In [16], MAML is utilized to allow sound source localization models to adapt to different environments and conditions.

In this paper, we propose Meta-SELD, applying meta-learning to SELD models with activity-coupled Cartesian DOA (ACCDOA) representation [7] to improve performance, especially in localization. We use MAML to find general initial parameters to minimize the loss across several tasks in Meta-SELD so that it can quickly adapt to an unseen environment. We take recordings in different environments as different tasks and aim to improve the performance of a specific unseen environment with a few samples recorded in the same environment. The experimental results demonstrate that Meta-SELD outperforms the fine-tuning method from the pre-trained SELD model in the STARSS23 dataset.

2. RELATED WORK

Activity-coupled Cartesian DOA (ACCDOA) representation [7] assigns a sound event activity to the length of a corresponding Cartesian DOA. When inferring, the threshold is set for the length of class-wise Cartesian DOA vectors to determine whether an event class is active. In contrast to EINV2, the ACCDOA representation merges SED and DOA branches into a single branch, decreasing the model parameters and avoiding the necessity of balancing the loss measuring on the SED task and the DOA task.

The ACCDOA representation can not detect homogenous overlaps. Therefore, multi-ACCDOA which still contains a single branch and combines class-wise output format and track-wise output format, is proposed to overcome the problem [8]. While each track in the track-wise output format of EINV2 only detects one event class and a corresponding location, each track in the multi-ACCDOA predicts activities and corresponding locations of all target classes. Auxiliary duplicating permutation invariant training (ADPIT) is also proposed to train each track of the multi-ACCDOA with original targets and duplicated targets, enabling each track to regard the same target as the single one. The multi-ACCDOA representation is shown in Fig. 1. Its outputs are track-wise and class-wise Cartesian DOA vectors. Each vector length indicates the activity of the event. Besides the activity threshold, multi-ACCDOA employs angle thresholds to determine whether the predicted objects are the same or different.

3. META-SELD

3.1. The SELD model

Without loss of generality, in this study, we adopt a simple Convolutional Recurrent Neural Network (CRNN) as our network, which is similar to the baseline of Task 3 of DCASE 2022 Challenge [2] but with ACCDOA format. The network has three convolution blocks followed by a one-layer bidirectional gated recurrent unit (BiGRU). The network takes the concatenation of log-mel spectrograms and intensity vectors as input and predicts active sound events with corresponding Cartesian DOA vectors for each time step. The network architecture of CRNN is shown in Table 1.

Table 1: The network architecture of CRNN

Log-mel spectrogram & Intensity vectors
(Conv2d 3 × 3 @ 32, BatchNorm2d, ReLU) × 2, Avg Pooling 2 × 2
(Conv2d 3 × 3 @ 64, BatchNorm2d, ReLU) × 2, Avg Pooling 2 × 2
(Conv2d 3 × 3 @ 128, BatchNorm2d, ReLU) × 2, Avg Pooling 2 × 2
(Conv2d 3 × 3 @ 256, BatchNorm2d, ReLU) × 2, Avg Pooling 1 × 2
Global average pooling @ frequency
1-layer BiGRU of 128 hidden size, 256 × 39 linear layer, Tanh
Mean Square Error

3.2. Meta-SELD training

Given a model represented by a parameterized function f_Θ with parameters Θ, MAML [14] learns the initial parameters Θ_0 from general tasks $\mathcal{T}_i$ sampled from the training set $\mathcal{D}_{\texttt{train}}$ and is expected to perform well on unseen tasks from the test set $\mathcal{D}_{\texttt{test}}$ after a few iterations of parameters update with a small number of samples from the corresponding task. These initial parameters are very sensitive to being further optimized on a specific task. Each task $\mathcal{T}_i$ consists of a labeled support set $\mathcal{S}_i$ of K samples and a labeled query set $\mathcal{Q}_i$ of Q samples. A new task is expected to be quickly adapted with K samples, which is known as K-shot learning. The loss function of MAML is defined as

$$\mathcal{L} = \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}(f_\Theta) \tag{1}$$

where $p(\mathcal{T})$, which is sampled from $\mathcal{D}_{\texttt{train}}$, is a distribution over tasks that we want our model to be able to adapt to. In contrast to supervised deep learning methods, the objective of which is to find optimal parameters to minimize the loss function across all training samples, MAML tries to find generalized initial parameters for different tasks. MAML will then update the initial parameters after several iterations of training on data of new tasks.

There are two groups of parameters in the MAML algorithm, meta-parameters and adapt-parameters. In the meta-training phase, MAML starts with randomly initialized meta-parameters Θ and then adapts to a new specific task $\mathcal{T}_i$ with several update iterations using $\mathcal{S}_i$. The meta-parameters Θ become adapt-parameters Θ_i':

$$\Theta_i' = \Theta - \alpha \nabla_\Theta \mathcal{L}_{\mathcal{T}_i}\left(f_\Theta, \mathcal{S}_i\right) \tag{2}$$

where α is the adaptation learning rate for adapt-parameters updates. After updates across a batch of tasks, the meta-parameters are updated as:

$$\Theta = \Theta - \beta \nabla_\Theta \sum_{\mathcal{T}_i} \mathcal{L}_{\mathcal{T}_i}\left(f_{\Theta_i'}, \mathcal{Q}_i\right) \tag{3}$$

Algorithm 1 Meta-training of MAML for Meta-SELD

```
Require: Distribution over all rooms $p(\mathcal{T})$, adaptation step size $\alpha$, meta step size $\beta$
 1: randomly initialize meta-parameters $\Theta$
 2: while not done do
 3:     Sample a batch of rooms $\mathcal{T}_i \sim p(\mathcal{T})$
 4:     for each room $\mathcal{T}_i$ do
 5:         Sample disjoint examples $(\mathcal{S}_i, \mathcal{Q}_i)$ from $\mathcal{T}_i$
 6:         Let $\Theta_{i,0} \leftarrow \Theta$
 7:         for gradient descent step $j := 0$ to $N-1$ do
 8:             Perform gradient descent to update adapt-parameters:
                $\Theta_{i,j+1} \leftarrow \Theta_{i,j} - \alpha \nabla_{\Theta_i} \mathcal{L}_{\mathcal{T}_i}(\Theta_{i,j}, \mathcal{S}_i)$
 9:         end for
10:         Compute $\mathcal{L}_{\mathcal{T}_i}(f_{\Theta_{i,N}}, \mathcal{Q}_i)$
11:     end for
12:     Perform gradient descent to update meta-parameters:
        $\Theta \leftarrow \Theta - \beta \nabla_{\Theta} \sum_{\mathcal{T}_i} \mathcal{L}_{\mathcal{T}_i}\left(f_{\Theta_{i,N}}, \mathcal{Q}_i\right)$
13: end while
```

where β is the meta step size. The loss $\mathcal{L}_{\mathcal{T}_i}$ is calculated by the parameterized function $f_{\Theta'_i}$ on the query set $\mathcal{Q}_i$. After updating Θ on the query set, Θ will be used as initial parameters for the following meta-training steps.

We aim to adapt to an unseen environment with K samples (K-shot). The objective of MAML is to find optimal initial parameters across several tasks, so we need to construct a set of tasks from the training set $\mathcal{D}_{\texttt{train}}$. $\mathcal{D}_{\texttt{train}}$ is split according to the different recording rooms. Audio clips recorded in different rooms belong to different tasks. We first sample a batch of tasks from all tasks and then sample $K + Q$ samples in each task, where K samples for a support set $\mathcal{S}_i$ and Q samples for a query set $\mathcal{Q}_i$. The overall training procedure of MAML is summarized in Algorithm 1. Step 8 in Algorithm 1 is an inner-loop update for adapt-parameters, while Step 12 is outer-loop updates for meta-parameters.

3.3. Meta-SELD test

In the meta-testing phase, a specific unseen task $\mathcal{T}_j^{\texttt{test}}$ created using $\mathcal{D}_{\texttt{test}}$ is used. $\mathcal{T}_j^{\texttt{test}}$ consists of a labeled support set $\mathcal{S}_j^{\texttt{test}}$ of K samples, and an unlabeled query set $\mathcal{Q}_j^{\texttt{test}}$ of Q samples. After training the model using well-trained parameter Θ from the meta-training phase as the initial parameters on $\mathcal{S}_j^{\texttt{test}}$, we get updated parameters ${\Theta_j}'$. We then use $f_{\Theta'_j}$ to evaluate on $\mathcal{Q}_j^{\texttt{test}}$.

The meta processes for testing and training are slightly different. Similar to the training, the test set $\mathcal{D}_{\texttt{test}}$ is split according to the recording room of each audio clip. For clips of each room, we also chose K samples for meta-test support set $\mathcal{S}_j^{\texttt{test}}$ and all remaining samples for meta-test query set $\mathcal{Q}_j^{\texttt{test}}$. After N iterations of parameters update on $\mathcal{S}_j^{\texttt{test}}$, the meta-parameters Θ are updated to $\Theta_{j,N}$. The final performance is evaluated on $\mathcal{Q}_j^{\texttt{test}}$ with $f_{\Theta_{j,N}}$.

4. EXPERIMENTS

4.1. Dataset

There are 16 different recording rooms in total in the development set of the STARSS23 dataset, including nine recording rooms in *dev-train-set* and seven recordings rooms in *dev-test-set*. The development set of STARSS23, which contains roughly 7.5 hours of recordings, has less data than the development set in DCASE 2021, which contains roughly 13 hours of synthetic recordings [17]. Considering the complexity of the real-scene environment, we use additional datasets to improve the performance. We generated simulated data using the generator code provided by DCASE[1]. We synthesize multi-channel spatial recordings by convolving monophonic sound event examples with multi-channel Spatial Room Impulse Responses (SRIRs). Samples of sound events are selected from AudioSet [18] and FSD50K [19], based on the affinity of the labels in those datasets to target classes in STARSS23. PANNs [20] are then employed to clean the selection of the clips. We use pre-trained PANNs to infer these clips and select high-quality clips based on output probability above 0.8. We extracted SRIRs from the TAU Spatial Room Impulse Response Database (TAU-SRIR DB)[2], which contains SRIRs captured in 9 rooms at Tampere University. It was used for official synthetic datasets in DCASE 2019-2021 [1, 17, 21].

The 2700 1-minute audio clips that we synthesized using the abovementioned SRIRs from 9 rooms are used for $\mathcal{D}_{\texttt{train}}$, and all of *dev-set* of STARSS23, recorded in 16 rooms, are used for $\mathcal{D}_{\texttt{test}}$.

4.2. Experimental setup

The sampling rate of the dataset is 24 kHz. We extracted 64-dimensional log mel spectrograms from four-channel first-order ambisonics (FOA) signals with a Hanning window of 1024 points, and a hop size of 320. Each audio clip is segmented to a fixed length of five seconds with no overlap for training and inference.

In the meta-training phase, the training set and test set are divided into 9 tasks and 16 tasks, respectively, corresponding to 9 rooms and 16 rooms. We first sample a batch of rooms randomly and then sample a batch of examples from each of the rooms. The batch of samples of each room constructs a task, and a part of the samples are support samples while the remaining samples are query samples. The batch size of rooms and samples is 4 and 64, respectively. A batch of samples contains 30 support samples and 34 query samples. In the meta-test phase, we sort the audio clips according to the filename, and select the first 30 samples of recordings of each room as samples from the support set $\mathcal{S}_j^{\texttt{test}}$. The remaining samples of each room are as samples from the test set $\mathcal{Q}_j^{\texttt{test}}$. The AdamW optimizer is used for updates of meta-parameters of MAML, while the SGD optimizer is used to update adapt-parameters. The meta step size β begins with 0.001 in the first 100 epochs out of 150 epochs in total and is then decreased by 10% every 20 epochs. The adaptation step size and the number of update iterations are always kept at 0.01 and 5, respectively.

To demonstrate the effectiveness of Meta-SELD, we compare Meta-SELD with the fine-tuning method from the pre-trained SELD model. Firstly, we train a SELD model with AdamW optimizer in $\mathcal{D}_{\texttt{train}}$ from scratch. The learning rate is 0.0003 for the first 70 epochs and then decreases to 0.00003 for the following 20 epochs. Secondly, we initialize the parameters from the previously trained SELD model and then use $\mathcal{S}_i^{\texttt{test}}$ and $\mathcal{Q}_i^{\texttt{test}}$ as the training set and the test set of the i-th room to fine-tune. Similar to the process of the adapt-parameters updates in MAML, the SGD optimizer with a step size of 0.01 and update iterations of 5 are used for fine-tuning.

A joint metric of localization and detection [22, 23] is used: location-dependent F-score ($\mathrm{F}_{\leq 20^\circ}$) and error rate ($\mathrm{ER}_{\leq 20^\circ}$), and class-dependent localization recall ($\mathrm{LR}_{\mathrm{CD}}$) and localization error

---

[1] https://github.com/danielkrause/DCASE2022-data-generator

[2] https://zenodo.org/record/6408611

Table 2: The performance of the Meta-SELD and fine-tuning methods from pre-trained SELD models. Both two methods are evaluated in $\mathcal{Q}_i^{\texttt{test}}$. Note that *overall* scores of the fine-tuning method and Meta-SELD compute the fast adaptation performance of each individual room and then micro-average.

Room	$\mathrm{ER}_{20^\circ}\downarrow$			$\mathrm{F}_{20^\circ}\uparrow$			$\mathrm{LE}_{\mathrm{CD}}\downarrow$			$\mathrm{LR}_{\mathrm{CD}}\uparrow$			$\mathcal{E}_{\texttt{SELD}}\downarrow$		
	Pre-train	Fine-tune	Meta	Pre-train	Fine-tune	Meta	Pre-train	Fine-tune	Meta	Pre-train	Fine-tune	Meta	Pre-train	Fine-tune	Meta
fold3_room4	0.624	**0.574**	0.603	**44.5%**	40.4%	29.8%	17.8°	**17.6°**	21.5°	**64.6%**	61.2%	54.4%	**0.408**	0.414	0.470
fold3_room6	0.639	0.607	**0.594**	38.0%	**40.5%**	40.4%	18.0°	**17.2°**	17.4°	**65.3%**	63.8%	61.1%	0.427	**0.415**	0.419
fold3_room7	0.610	**0.606**	0.660	**31.1%**	30.7%	20.8%	23.6°	24.1°	**22.5°**	59.9%	**60.5%**	48.3%	0.458	**0.457**	0.523
fold3_room9	0.673	**0.601**	0.608	43.7%	46.6%	**47.5%**	19.1°	18.6°	**18.3°**	**78.7%**	78.2%	73.3%	0.389	**0.364**	0.375
fold3_room12	0.685	**0.659**	0.689	28.0%	29.8%	**33.0%**	26.8°	**26.1°**	33.3°	43.1%	43.6%	**46.3%**	0.531	**0.518**	0.520
fold3_room13	0.650	0.599	**0.594**	37.7%	**39.4%**	36.1%	17.5°	16.9°	**15.9°**	**50.9%**	48.8%	37.1%	0.465	**0.453**	0.488
fold3_room14	0.633	**0.582**	0.613	**40.2%**	37.4%	28.6%	**23.2°**	23.7°	24.8°	**55.3%**	54.0%	47.2%	0.452	**0.450**	0.498
fold3_room21	0.757	0.750	**0.735**	19.3%	**21.6%**	18.9%	20.5°	**18.9°**	20.6°	39.3%	31.4%	**43.8%**	0.571	0.581	**0.556**
fold3_room22	0.850	0.818	**0.800**	11.4%	12.8%	**16.7%**	31.6°	29.5°	**29.0°**	45.6%	43.8%	**48.8%**	0.614	0.604	**0.577**
fold4_room2	0.809	0.774	**0.753**	6.2%	8.2%	**15.4%**	47.8°	41.3°	**33.0°**	72.4%	72.4%	**75.7%**	0.572	0.550	**0.506**
fold4_room8	0.716	0.716	**0.702**	31.7%	**33.6%**	30.7%	22.5°	**21.0°**	23.2°	**54.0%**	49.4%	49.4%	**0.496**	0.501	0.507
fold4_room10	0.792	0.708	**0.651**	36.3%	**41.7%**	35.8%	23.8°	21.5°	**20.2°**	66.1%	72.0%	**78.2%**	0.475	0.423	**0.406**
fold4_room15	0.582	0.563	**0.539**	33.3%	33.5%	**43.4%**	16.5°	**15.5°**	19.3°	42.8%	42.6%	**59.0%**	0.478	0.472	**0.406**
fold4_room16	0.601	**0.584**	0.607	39.8%	**40.5%**	34.3%	21.7°	21.9°	**21.6°**	**55.1%**	54.9%	48.7%	0.443	**0.438**	0.474
fold4_room23	0.813	0.746	**0.676**	25.4%	26.5%	**31.8%**	26.2°	**24.9°**	25.8°	40.4%	43.6%	**47.3%**	0.575	0.546	**0.507**
fold4_room24	0.828	**0.779**	0.782	26.2%	25.7%	**30.8%**	**19.4°**	19.7°	24.4°	41.0%	**43.6%**	42.7%	0.566	0.549	**0.546**
Overall	0.707	0.677	**0.672**	23.0%	24.2%	**26.0%**	22.8°	22.3°	**21.9°**	39.5%	40.2%	**41.0%**	0.552	0.539	**0.531**

($\mathrm{LE}_{\mathrm{CD}}$). $\mathrm{F}_{\leq 20^\circ}$ and $\mathrm{ER}_{\leq 20^\circ}$ consider true positives predicted under a spatial threshold 20° from the ground truth. $\mathrm{LE}_{\mathrm{CD}}$ and $\mathrm{LR}_{\mathrm{CD}}$ are computed for localization predictions in the case that the types of sound events are predicted correctly. A macro-average of $\mathrm{F}_{\leq 20^\circ}$, $\mathrm{LR}_{\mathrm{CD}}$ and $\mathrm{LE}_{\mathrm{CD}}$ is used.

We use an aggregated SELD metric which was computed as

$$\mathcal{E}_{\texttt{SELD}} = \frac{1}{4}\left[\mathrm{ER}_{\leq 20^\circ} + (1 - \mathrm{F}_{\leq 20^\circ}) + \frac{\mathrm{LE}_{\mathrm{CD}}}{180^\circ} + (1 - \mathrm{LR}_{\mathrm{CD}})\right]. \tag{4}$$

4.3. Experimental results

Table 2 shows the performance of the Meta-SELD method compared with the fine-tuning method from the pre-trained SELD models. The pre-trained SELD models are trained without using samples from $\mathcal{D}_{\texttt{test}}$.

According to the last row of Table 2, the *overall* score, which is a micro average across all rooms, shows that all of $\mathrm{ER}_{\leq 20^\circ}$, $\mathrm{F}_{\leq 20^\circ}$, $\mathrm{LE}_{\mathrm{CD}}$, and $\mathrm{LR}_{\mathrm{CD}}$ are improved using Meta-SELD compared with the fine-tuning method. We observe a drop in $\mathcal{E}_{\texttt{SELD}}$ in fold3_room4 and fold4_room8 even though some new samples of unseen environments are used for training. This may be due to the fact that the new samples do not have valid information for training. We also observe the Meta-SELD method improves $\mathcal{E}_{\texttt{SELD}}$ by a large margin in fold3_room22, fold4_room2, and fold4_room23 where the pre-trained model has poor performance across all rooms. Specifically, $\mathrm{ER}_{\leq 20^\circ}$, $\mathrm{F}_{\leq 20^\circ}$, and $\mathrm{LR}_{\mathrm{CD}}$ of fold3_room22 and fold4_room23 outperform other methods. Meta-SELD mainly improves the performance of SED in fold3_room22 and fold4_room23. All metrics of fold4_room2 are improved in Meta-SELD compared with the fine-tuning method, especially in DOA estimation. In fold4_room2, all of the pre-trained model, the fine-tuning method, and Meta-SELD achieve $\mathrm{LR}_{\mathrm{CD}}$ of over 70%, but $\mathrm{LE}_{\mathrm{CD}}$ of three methods is always high compared with $\mathrm{LE}_{\mathrm{CD}}$ of other rooms. Meta-SELD decreases 14.8° and 8.3° of $\mathrm{LE}_{\mathrm{CD}}$ compared with the pre-trained model and the fine-tuning method in fold4_room2, hence directly leading to the increase of $\mathrm{F}_{\leq 20^\circ}$ and the decrease of $\mathrm{ER}_{\leq 20^\circ}$. However, performance degradation happens in fold3_room4, fold3_room7, fold3_room14, and fold4_room16, where Meta-SELD has the worst metric scores. There is no significant change in $\mathrm{LE}_{\mathrm{CD}}$, and the decline in SED performance is the main factor. One of the possible reasons for this observation could be that there are some conflicts in optimizing Meta-SELD across a batch of rooms.

Experimental results demonstrate that Meta-SELD can find better initial parameters across a batch of tasks than the fine-tuning method, especially in rooms where the pre-trained model and the fine-tuning method perform worse. Meta-SELD reduces the risk of overfitting when using a small number of samples, which usually happens in the fine-tuning method.

5. CONCLUSION

In this paper, we presented Meta-SELD, which employed Model-Agnostic Meta-Learning (MAML) to the sound event localization and detection task to achieve fast adaptation to unseen environments. The method only utilizes a small number of samples and a few update iterations of training. We use the STARSS23 dataset and synthesized 2700 1-minute samples that are convolved using monophonic sound event clips with multi-channel spatial room impulse responses. The sound event clips are extracted from FSD50K and AudioSet and are further filtered by the PANNs model through a probability threshold. The SRIRs used are from TAU-SRIR DB. Our methods are trained on synthetic datasets and evaluated on all development sets of the STARSS23 dataset. Audio clips recorded from the same room or synthesized using SRIRs collected from the same room are regarded as the same task for MAML. The experimental results show that the Meta-SELD method improves $\mathcal{E}_{\texttt{SELD}}$ significantly in those rooms where both the pre-trained model and the fine-tuning method perform unsatisfactorily. The overall score demonstrates that the Meta-SELD method outperforms the fine-tuning method on average.

6. ACKNOWLEDGEMENT

This work was supported in part by Grant "XJTLU RDF-22-01-084", UK Engineering and Physical Sciences Research Council (EPSRC) Grant EP/T019751/1 "AI for Sound (AI4S)". For the purpose of open access, the authors have applied a Creative Commons Attribution (CC BY) licence to any Author Accepted Manuscript version arising.

7. REFERENCES

[1] S. Adavanne, A. Politis, and T. Virtanen, "A multi-room reverberant dataset for sound event localization and detection," in *Proc. DCASE 2019 Workshop*, 2019, pp. 10–14.

[2] A. Politis, K. Shimada, P. Sudarsanam, S. Adavanne, D. Krause, Y. Koyama, N. Takahashi, S. Takahashi, Y. Mitsufuji, and T. Virtanen, "STARSS22: A dataset of spatial recordings of real scenes with spatiotemporal annotations of sound events," in *Proc. DCASE 2022 Workshop*, 2022, pp. 161–165.

[3] S. Adavanne, A. Politis, J. Nikunen, and T. Virtanen, "Sound event localization and detection of overlapping sources using convolutional recurrent neural networks," *IEEE J. Sel. Top. Signal Process.*, vol. 13, pp. 34–48, 2018.

[4] Y. Cao, T. Iqbal, Q. Kong, Y. Zhong, W. Wang, and M. D. Plumbley, "Event-independent network for polyphonic sound event localization and detection," in *Proc. DCASE 2020 Workshop*, 2020, pp. 11–15.

[5] Y. Cao, T. Iqbal, Q. Kong, F. An, W. Wang, and M. D. Plumbley, "An improved event-independent network for polyphonic sound event localization and detection," in *Proc. IEEE ICASSP 2021*, 2021, pp. 885–889.

[6] J. Hu, Y. Cao, M. Wu, Q. Kong, F. Yang, M. D. Plumbley, and J. Yang, "A track-wise ensemble event independent network for polyphonic sound event localization and detection," in *Proc. IEEE ICASSP 2022*, 2022, pp. 9196–9200.

[7] K. Shimada, Y. Koyama, N. Takahashi, S. Takahashi, and Y. Mitsufuji, "ACCDOA: Activity-coupled cartesian direction of arrival representation for sound event localization and detection," in *Proc. IEEE ICASSP 2021*. IEEE, 2021, pp. 915–919.

[8] K. Shimada, Y. Koyama, S. Takahashi, N. Takahashi, E. Tsunoo, and Y. Mitsufuji, "Multi-ACCDOA: Localizing and detecting overlapping sounds from the same class with auxiliary duplicating permutation invariant training," in *Proc. IEEE ICASSP 2022*, 2022, pp. 316–320.

[9] J. Hu, Y. Cao, M. Wu, Q. Kong, F. Yang, M. D. Plumbley, and J. Yang, "Sound event localization and detection for real spatial sound scenes: Event-independent network and data augmentation chains," in *Proc. DCASE 2022 Workshop*, 2022, pp. 46–50.

[10] Y. Wang, Q. Yao, J. T. Kwok, and L. M. Ni, "Generalizing from a few examples: A survey on few-shot learning," *ACM Computing Surveys*, vol. 53, no. 3, pp. 1–34, 2020.

[11] T. Hospedales, A. Antoniou, P. Micaelli, and A. Storkey, "Meta-learning in neural networks: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 9, pp. 5149–5169, 2021.

[12] W.-Y. Chen, Y.-C. Liu, Z. Kira, Y.-C. F. Wang, and J.-B. Huang, "A closer look at few-shot classification," in *Proc. ICLR 2019*, 2019.

[13] N. Gao, H. Ziesche, N. A. Vien, M. Volpp, and G. Neumann, "What matters for meta-learning vision regression tasks?" in *Proc. CVPR 2022*, 2022, pp. 14 776–14 786.

[14] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proc. ICML 2017*, 2017, pp. 1126–1135.

[15] S.-F. Huang, C.-J. Lin, D.-R. Liu, Y.-C. Chen, and H.-y. Lee, "Meta-TTS: Meta-learning for few-shot speaker adaptive text-to-speech," *IEEE/ACM Trans. on Audio, Speech, and Lang. Process.*, vol. 30, pp. 1558–1571, 2022.

[16] M. Barhoush, A. Hallawa, A. Peine, L. Martin, and A. Schmeink, "Localization-driven speech enhancement in noisy multi-speaker hospital environments using deep learning and meta learning," *IEEE/ACM Trans. on Audio, Speech, and Lang. Process.*, vol. 31, pp. 670–683, 2022.

[17] A. Politis, S. Adavanne, D. Krause, A. Deleforge, P. Srivastava, and T. Virtanen, "A dataset of dynamic reverberant sound scenes with directional interferers for sound event localization and detection," in *Proc. DCASE 2021 Workshop*, 2021, pp. 125–129.

[18] J. F. Gemmeke, D. P. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio Set: An ontology and human-labeled dataset for audio events," in *Proc. IEEE ICASSP 2017*, 2017, pp. 776–780.

[19] E. Fonseca, X. Favory, J. Pons, F. Font, and X. Serra, "FSD50K: An open dataset of human-labeled sound events," *IEEE/ACM Trans. on Audio, Speech, and Lang. Process.*, vol. 30, pp. 829–852, 2021.

[20] Q. Kong, Y. Cao, T. Iqbal, Y. Wang, W. Wang, and M. D. Plumbley, "PANNs: Large-scale pretrained audio neural networks for audio pattern recognition," *IEEE/ACM Trans. on Audio, Speech, and Lang. Process.*, vol. 28, pp. 2880–2894, 2020.

[21] A. Politis, S. Adavanne, and T. Virtanen, "A dataset of reverberant spatial sound scenes with moving sources for sound event localization and detection," in *Proc. DCASE 2020 Workshop*, 2020, pp. 165–169.

[22] A. Mesaros, S. Adavanne, A. Politis, T. Heittola, and T. Virtanen, "Joint measurement of localization and detection of sound events," in *Proc. IEEE WASPAA 2019*, 2019, pp. 333–337.

[23] A. Politis, A. Mesaros, S. Adavanne, T. Heittola, and T. Virtanen, "Overview and evaluation of sound event localization and detection in DCASE 2019," *IEEE/ACM Trans. on Audio, Speech, and Lang. Process.*, vol. 29, pp. 684–698, 2020.

LEVERAGING GEOMETRICAL ACOUSTIC SIMULATIONS OF SPATIAL ROOM IMPULSE RESPONSES FOR IMPROVED SOUND EVENT DETECTION AND LOCALIZATION

Christopher Ick *Brian McFee*

Music and Audio Research Laboratory, New York University
Brooklyn, NY 11201, United States

ABSTRACT

As deeper and more complex models are developed for the task of sound event localization and detection (SELD), the demand for annotated spatial audio data continues to increase. Annotating field recordings with 360° video takes many hours from trained annotators, while recording events within motion-tracked laboratories are bounded by cost and expertise. Because of this, localization models rely on a relatively limited amount of spatial audio data in the form of spatial room impulse response (SRIR) datasets, which limits the progress of increasingly deep neural network based approaches. In this work, we demonstrate that simulated geometrical acoustics can provide an appealing solution to this problem. We use simulated geometrical acoustics to generate a novel SRIR dataset that can train a SELD model to provide similar performance to that of a real SRIR dataset. Furthermore, we demonstrate using simulated data to augment existing datasets, improving on benchmarks set by state of the art SELD models. We explore the potential and limitations of geometric acoustic simulation for localization and event detection. We also propose further studies to verify the limitations of this method, as well as further methods to generate synthetic data for SELD tasks without the need to record more data.

Index Terms— Acoustic Simulation, Localization, Data Augmentation

1. INTRODUCTION

Sound event detection and localization (SELD) is the union of two active fields of research; sound event detection (SED), and localization, or direction-of-arrival (DoA) estimation. Expanding the scene-description capabilities of SED with the spatiotemporal characterization of localization sees applications ranging from autonomous robot navigation [1] and urban monitoring [2], to speaker diarization [3] and immersive experiences in virtual and augmented reality devices.

While earlier techniques for SELD have focused on traditional signal-processing or parametric models such as [4, 5, 6, 7], recent literature is dominated by deep neural network (DNN) approaches, which have been shown superior performance in both pure localization [8, 9] as well as joint SELD tasks [10]. A surge of interest in this field can be attributed to the introduction of SELD as a task in the DCASE2019 challenge [11]. This challenge included the release of several 4-channel audio datasets with spatial and temporal annotations for sound events. The audio was generated by convolving sound events with spatial room impulse responses (SRIRs) recorded in 5 separate rooms at 504 unique azimuth-elevation-distance combinations. This was further iterated upon by the SELD challenge in DCASE2020 [12] with 13 unique rooms. Recently, DCASE2022 reintroduced the challenge with hand-annotated real-world recordings in the STARSS22 dataset [13], providing one of the first datasets with real-world data upon which to evaluate SELD models. This dataset uses a combination of 360° video, and motion capture to extract spatiotemporal annotations that were manually validated. In addition to this, the DCASE2022 dataset also included a release of the SRIRs used to generate the training data for the SELD task, as well as the code for the generator itself, allowing users to generate their own annotated spatial data [14]. This data included SRIRs measured over a wide range of positions over 9 different rooms in Tampere University's campus.

These datasets are unique in the density of SRIR measurements across particular paths, the variety of acoustic enclosures, and the large amount of SRIRs. Because of their scale, visibility, and quality, these datasets have become some of the most cited SRIR datasets for DNN-based approaches to SELD, because of their ability to meet the data requirements of these highly-parametrized models. Despite this, these datasets are still severely limited by the recording procedure of SRIRs, which require time, expertise, and a low-noise environment to produce at a high quality. Increasing the spatial density, variety of trajectories types, and number of trajectory paths becomes multiplicatively time consuming to develop. Furthermore, the range of rooms in which these measurements can be recorded is inherently limited by the limitations of the recording facilities, usually limited to a dozen rooms or so in the best of cases. However, without a wide range of acoustic environments to perform these measurements, generalization to a variety of unseen acoustic environments becomes impossible.

Physical acoustic simulations provide an attractive solution to the limitations of field-recorded SRIRs. Acoustic simulation is typically split into two categories: wave based methods, which simulate the propagation of sound waves through physical media, and geometric modeling methods, which model the transportation of acoustic energy through acoustic rays, mimicking popular methods for modeling optical rays. Geometrical acoustics approximate the wavelength of the propagating sound to have wavelength relatively small compared to the room geometries of interest, and neglects wave effects such as diffraction or scattering. Nevertheless because of ease of implementation and computational efficiency, geometrical acoustic modeling methods have seen wide success in several tasks, including modeling architectural acoustics [15] and room parameter estimation [16].

In this work, we propose utilizing one method of geometrical acoustics modeling, the image source method, to generate simulated SRIRs for training DNN models for SELD. We demonstrate

---

This material is based upon work supported by the National Science Foundation under NSF Award 1922658.

the effectiveness of this simulated method for SRIR generation using the framework and data provided in previous DCASE SELD challenges. By creating an audio dataset from simulated SRIRs, we train a SELD model with similar performance to one utilizing real-world SRIRs. By directly compares simulated SRIRs with a datasets of recorded SRIRs of the similar size, room geometries, and DoA distributions, we demostrate the downstream effects of simulation in place of recording as being relatively minimal, differing our work from prior studies [17]. Furthermore, we augment a typical SRIR dataset with simulated SRIRs, training models that outperform those trained solely on recorded SRIRs. Finally we propose further experiments to explore the use of simulated SRIRs for training SELD models. The code associated with this work is released in an open-source github repository [1] to further work in using synthetic SRIRs for training DNN models.

2. ACOUSTIC SIMULATIONS

2.1. The Image Source Method

The image source method (ISM) is a technique used in architectural acoustics and room modeling to predict the sound field in enclosed spaces [18]. The ISM considers the primary sound source and virtual images reflected by the room's boundaries. These virtual sources are assumed to emit sound with the same magnitude and phase as the primary source, but with a delay due to the additional path length traveled. Typically, this starts with defining the room geometry, including the positions and shapes of the walls, ceiling, and floor. For each reflecting surface, virtual image sources are "mirrored" across the boundary. The number of virtual sources depends on the order of reflections considered. From here, the interaction between the primary sound source and the virtual image sources is calculated by determining the path lengths, time delays, and attenuation factors associated with each source-receiver combination, as well as the material properties of each surface through which the sound path is reflected upon. By summing the contributions of the primary source and its image sources, the sound field at various locations in the room can be predicted, providing an estimation of the sound pressure level, arrival times, and directivity patterns.

It's important to note several limitations of the ISM model. The ISM implicitly assumes that all surfaces are perfectly reflective and flat, with idealized acoustic properties, which fails to account for acoustic effects such as scattering or diffraction. Furthermore, the order of reflections/virtual sources scales the computational cost exponentially, meaning compute late-stage reflections in an SRIR prohibitively expensive.

Despite its limitations, the image source method is widely used due to its computational efficiency and effectiveness in predicting sound fields in enclosed spaces. Because localization is more reliant on direct sounds/early reflections, the limitations caused by use of the the ISM for computing SRIRs can be expected to be relatively minimal.

2.2. The TAU-SRIR Dataset

To validate the use of ISM-generated SRIRs in a direct-comparison, we take the existing TAU-SRIR database [14] as an example database for which well established metrics for SELD have been measured.

	Azimuth (ϕ)	Elevation (θ)
M1	45°	35°
M2	-45°	-35°
M3	135°	-35°
M4	-135°	35°

Table 1: Microphone Geometry for TAU-SRIR dataset. Each microphone is 4.2cm from the center, and is modeled with a hypercardioid response.

Room Name	Traj. type	N_t	N_h	N_{SRIRs}
Bomb shelter	Circular	2	9	6480
Gym	Circular	2	9	6480
PB132	Circular	2	9	6480
PC226	Circular	2	9	6480
SA203	Linear	6	3	1594
SC203	Linear	4	5	1592
SE203	Linear	4	4	1760
TB103	Linear	4	3	1184
TC352	Circular	2	9	6480

Table 2: Trajectory information for rooms contained in the TAU-SRIR dataset [14]. Each room contains trajectories across a number of trajectory groups (N_t) and a number of heights (N_h), for a total of $N_t \times N_h$ trajectories per room. Each trajectory is sampled in roughly 1° increments.

The TAU-SRIR database contains SRIRs recorded in 9 different rooms throughout Tampere University's campus. Each SRIR is computed by recording a maximum-length sequence (MLS) played through a loudspeaker, recorded on an Eigenmike spherical microphone array. Each SRIR was downsampled to 24kHz and truncated at 300ms, resulting in 7200 samples per RIR. The data is stored in a 4-channel audio corresponding to a tetrahedral microphone array with the geometry in spherical coordinates (ϕ, θ), specified in Table 1. For each room, the position of the microphone array was provided.

SRIRs were measured along either circular or linear traces at fixed distance from the microphone array along the z-axis at a number of trajectory groups, separated by distance and reflection across the axis of the microphone array in the case of linear traces. Circular trajectory groups had a specified radius of orbit, whereas linear trajectories had a specified start and end point in 3D space. Each trajectory was repeated at a number of different heights, and each trajectory had a fixed number SRIR measurements and corresponding DoA measurements recorded as Cartesian components of a unit vector. The number of SRIR measurements vary across different trajectories/heights, spaced in roughly 1° increments. The total number measurements can be seen in Table 2

2.3. Room Simulation

We recreate this dataset using the python package *pyroomacoustics* [19], a pythonic implementation of the ISM, that has demonstrated use in implementations of various algorithms for beamforming, direction finding, adaptive filtering, source separation, and single channel denoising.

To replicate the acoustic conditions of each of the rooms in the TAU-SRIR dataset, we randomly sampled the RIRs uniformly in each room until we had a sample of 5 single-channel RIRs. Using the Schroeder method [20], we estimated the RT_{60} of each room by

---

[1] `https://github.com/ChrisIck/DCASE_Synth_Data`

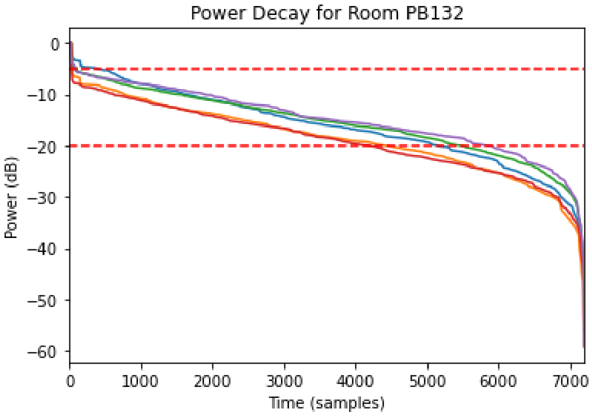


Figure 1: The log-scale energy decay from a sample of RIRs from a singular room. The region in the dashed lines is roughly linear, suggesting it corresponds to mid-late reflections, and is used for an estimate of RT$_{60}$

hand-selecting the early decay of the energy-decay function of the RIR samples and computing the linear fit. This can be see in in Figure 1. Using the inverse Sabine formula, we used this to estimate the mean absorption coefficient of the rooms and the number of required reflection orders to approximate a room of a similar RT$_{60}$. We combined these parameters with the geometry estimations from the TAU-SRIR dataset to construct virtual rooms matching those of the 9 rooms in the TAU-SRIR dataset. To this room, we added a virtual tetrahedral microphone with the geometry specified in Table 1, with each virtual microphone using a hypercardioid response pattern centered at the position specified in the TAU-SRIR dataset.

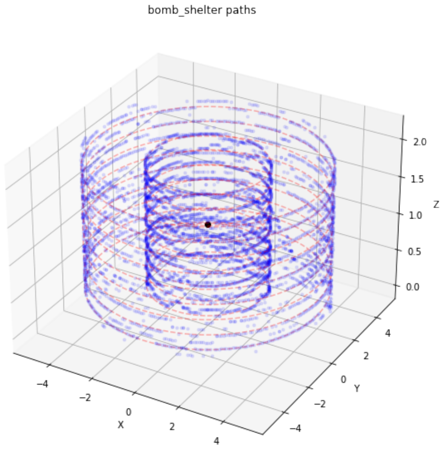


Figure 2: SRIR measurement positions reconstructed from the DoAs provided in the TAU-SRIR dataset (blue), compared to the path specified by the height, radius, and position labels provided (red)

To estimate the positions of the SRIRs in space corresponding to the DoA measurements in the TAU-SRIR dataset, we chose points along the DoA that most closely matched the corresponding path on the line specified by the trajectory in the dataset. Circular datasets specified their height and radius, and were centered along the same z-line as the microphone array. Linear trajectories specified the start point and end points of their traces. These matched points were estimated by projecting the DoA vectors onto a cylinder that matched the radius of the trajectory, after translation by the position of the microphone array and the height of the trajectory of interest (see Figure 2). Once these points were estimated, they were placed into the simulated room in a position relative to the virtual microphone array, and the 4-channel SRIR was computed with the ISM for each point, at the sample rate of 2400kHz. To match the dimensions of the original TAU-SRIR dataset, we truncate the SRIRs to 300ms, providing us with an SRIR with dimensions of 7200×4 for each point.

3. METHODOLOGY

To evaluate the performance of this dataset in SELD tasks, we generated a dataset of audio events consistent with the methodology of dataset generation for training the baseline SELD model in the DCASE 2022 challenge [21]. We generated 3 datasets, one using the original SRIR database, which we will refer to as the TAU-SRIR datset. We also generated a dataset using only the synthetic SRIRs, which we refer to as the SIM-SRIR dataset. Finally, we generated a third dataset that equally samples both the original and simulated SRIRs, which we will refer to as the augmented SRIR dataset, or AUG-SRIR.

3.1. Data Generation

To generate our annotated spatialized audio, we followed the procedure used in DCASE2019-2021, by convolving various sound events with SRIRs.

The sound events were drawn from the FSD50k audioset [22], a subset containing over 20k sound events of 13 classes selected for the DCASE challenge. These sound events were spatialized into virtual recordings, each corresponds to a singular room, allowing for up to three concurrently active sources. The sources can be static or dynamic, with equal probability, and the dynamic sources can move at slow (10°/sec), moderate (20°/sec), or fast (40°/sec) angular speeds. Each sample lasted 60 seconds, 40 of which had at least one active event class.

For each of the 3 SRIR datasets, 1200 recordings were created in separate folds, 900 for training and 300 for validation. The training and validation sets used 6 and 3 rooms respectively, such that none of the same rooms overlapped both folds.

3.2. Model

The model architecture is identical to the one used in the DCASE2022 Task 3 challenge baseline [13]; a SELDnet style CRNN with multi-ACCDOA representation for co-occuring events [23]. The model takes T frames of an STFT time-frequency representation of the multichannel features, and outputs $T/5 \times N \times C \times 3$ vector coordinates, and N is the assumed maximum co-occurring events, in our case 3.

The input features are 4-channel 64-band log-mel spectrograms combined with SALSA-lite spatial features, all of which are truncated to include bins up to 9kHz, without mel-band aggregation following [24]).

Each model was trained on the training folds generated from the SRIR datasets described above. In addition to this, data from the

	ER_{20°	F_{20°	LE_{CD}	LR_{CD}
TAU-SRIR	**0.71**	14.4%	55.1°	39.2%
SIM-SRIR	0.73	13.0%	79.6°	34.8%
AUG-SRIR	0.75	**16.3%**	**52.3°**	**42.3%**

Table 3: The cross-class evaluation metrics for models trained on data generated from different SRIR datasets

Sony-TAu Realistic Spatial Soundscapes 2022 (STARSS22) [13] was added for training, using the 54 development sound mixtures for training, but witholding the remaining 52 clips for evaluation of the models, ensuring the results were exclusively on real recorded data from unseen rooms. The models were trained for 100 epochs each,

3.3. Evaluation

Evaluation was completed using join localization-detection metrics established in the DCASE 2020 challenge. The detection metrics used were error-rate and F1 score for a spatial threshold within 20° (ER_{20° and F_{20°). F1 score was macro-averaged to account for class distribution differences in the FSD50k audio subset used. Localization metrics are class dependent localization error and recall (LE_{CD} and LR_{CD}).

4. RESULTS

Despite the coarse physical approximations made by the ISM, the entirely synthetic SRIRs generated with this process performed nearly as well as the SRIRs recorded in real world settings. Furthermore, the dataset of real SRIRs augmented by synthetic SRIRs outperformed both by a narrow margin, showing benefits of geometrical acoustics simulation for data augmentation for SELD tasks.

Regarding the cross-class average performance of the models in Table 3, we can see that for our classification metrics, all three models perform relatively similarly, with a slight performance edge to the AUG-SRIR dataset trained models. Looking into the per-class results in figure 3, we can see that generally, all three models struggle with similar classes (telephone, laughter, door), but the AUG-SRIR dataset outperforms both in certain classes for which both other models perform poorly on (Water tap/Faucet, and Knock).

Looking at the localization based results, it appears that some amount of the performance differences between the SIM-SRIR trained models and the TAU-SRIR trained models can possibly be attributed to model fine-tuning; while SIM-SRIR trained models had poor localization performance on certain results (Water tap/Faucet, and Knock), the AUG-SRIR model outperformed the baseline TAU-SRIR dataset. This suggests that the SIM-SRIR datasets are actually providing beneficial information for these sound classes missing from the TAU-SRIR datasets. With more thorough model tuning, it's possible that the performance for SIM-SRIR trained models it even closer to that of the baseline.

5. CONCLUSION

In this work we demonstrated the potential of using acoustic simulation to generate spatial audio data for training SELD models. We've shown that simulated SRIR data can improve the performance of SELD models as a form of data augmentation. In addition to this, we've shown that simulated SRIRs, while not as effective as those recorded in real acoustic environments, can be used to effectively train SELD models, removing the relatively high cost of producing additional data for similarly performing results in a relatively limited setting. Generating larger volumes of SRIRs over a wider range of acoustic conditions could provide even better results than these baselines, potentially demonstrating greater robustness over varying acoustic environments. Furthermore, using a high-volume of simulated SRIRs to train a model, and using a hold-out of limited high-quality real-world data to fine the model could produce SoTA results. This result is promising for future experiments involve SRIRs for use in acoustic simulation data. Understanding the requirements for angular density in dynamic SRIR recordings can help inform future dataset collection practices, as well as the robustness of these models to noise; limited work was done exploring the effect of noise on the models trained with simulated SRIRs. Further ablation studies are necessary to understand the limitations of geometrical acoustic methods for SELD-based tasks, but these early experiments suggest that these can provide a low-resource alternative to real-world SRIR recordings.

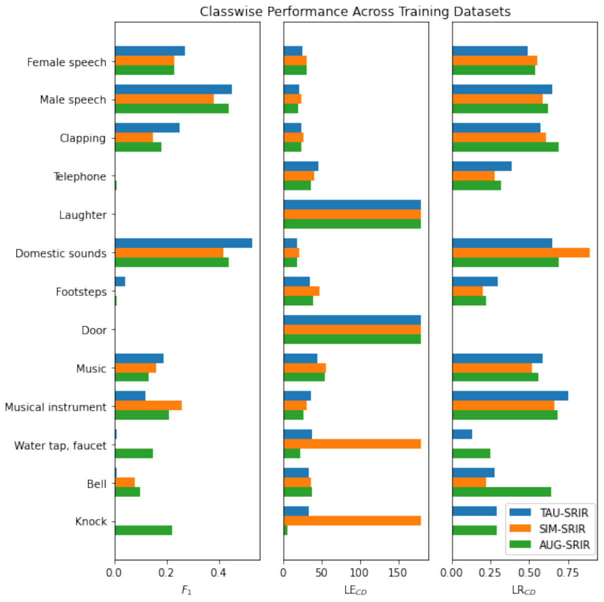


Figure 3: Per-class results of models trained on each dataset for F-measure within 20°, localization error, and localization recall.

6. REFERENCES

[1] L. Bondi, G. Chuang, C. Ick, A. Dave, C. Shelton, B. Coltin, T. Smith, and S. Das, "Acoustic imaging aboard the international space station (iss): Challenges and preliminary results," in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 5108–5112.

[2] F. Stevens and D. Murphy, "Spatial impulse response measurement in an urban environment," in *Audio Engineering Society Conference: 55th International Conference: Spatial Audio*, Aug 2014. [Online]. Available: http://www.aes.org/e-lib/browse.cfm?elib=17355

[3] J. Wang, Y. Liu, B. Wang, Y. Zhi, S. Li, S. Xia,

J. Zhang, F. Tong, L. Li, and Q. Hong, "Spatial-aware speaker diarizaiton for multi-channel multi-party meeting," in *Interspeech 2022*. ISCA, sep 2022. [Online]. Available: https://doi.org/10.21437%2Finterspeech.2022-11412

[4] T. Butko, F. G. Pla, C. Segura, C. Nadeu, and J. Hernando, "Two-source acoustic event detection and localization: Online implementation in a smart-room," in *2011 19th European Signal Processing Conference*, 2011, pp. 1317–1321.

[5] C. J. Grobler, C. P. Kruger, B. J. Silva, and G. P. Hancke, "Sound based localization and identification in industrial environments," in *IECON 2017 - 43rd Annual Conference of the IEEE Industrial Electronics Society*, 2017, pp. 6119–6124.

[6] R. Chakraborty and C. Nadeu, "Sound-model-based acoustic source localization using distributed microphone arrays," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 619–623.

[7] K. Lopatka, J. Kotus, and A. Czyzewski, "Detection, classification and localization of acoustic events in the presence of background noise for acoustic surveillance of hazardous situations," *Multimedia Tools Appl.*, vol. 75, no. 17, p. 10407–10439, sep 2016. [Online]. Available: https://doi.org/10.1007/s11042-015-3105-4

[8] S. Adavanne, A. Politis, and T. Virtanen, "Direction of arrival estimation for multiple sound sources using convolutional recurrent neural network," *CoRR*, vol. abs/1710.10059, 2017. [Online]. Available: http://arxiv.org/abs/1710.10059

[9] L. Perotin, R. Serizel, E. Vincent, and A. Guérin, "Crnn-based multiple doa estimation using acoustic intensity features for ambisonics recordings," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 1, pp. 22–33, 2019.

[10] S. Adavanne, A. Politis, J. Nikunen, and T. Virtanen, "Sound event localization and detection of overlapping sources using convolutional recurrent neural networks," *CoRR*, vol. abs/1807.00129, 2018. [Online]. Available: http://arxiv.org/abs/1807.00129

[11] S. Adavanne, A. Politis, and T. Virtanen, "Localization, detection and tracking of multiple moving sound sources with a convolutional recurrent neural network," *CoRR*, vol. abs/1904.12769, 2019. [Online]. Available: http://arxiv.org/abs/1904.12769

[12] A. Politis, S. Adavanne, and T. Virtanen, "A dataset of reverberant spatial sound scenes with moving sources for sound event localization and detection," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2020 Workshop (DCASE2020)*, Tokyo, Japan, November 2020, pp. 165–169. [Online]. Available: https://dcase.community/workshop2020/proceedings

[13] A. Politis, K. Shimada, P. Sudarsanam, S. Adavanne, D. Krause, Y. Koyama, N. Takahashi, S. Takahashi, Y. Mitsufuji, and T. Virtanen, "STARSS22: A dataset of spatial recordings of real scenes with spatiotemporal annotations of sound events," in *Proceedings of the 8th Detection and Classification of Acoustic Scenes and Events 2022 Workshop (DCASE2022)*, Nancy, France, November 2022, pp. 125–129. [Online]. Available: https://dcase.community/workshop2022/proceedings

[14] A. Politis, S. Adavanne, and T. Virtanen, "TAU Spatial Room Impulse Response Database (TAU- SRIR DB)," Apr. 2022. [Online]. Available: https://doi.org/10.5281/zenodo.6408611

[15] J. B. Allen and D. A. Berkley, "Image method for efficiently simulating small-room acoustics," *The Journal of the Acoustical Society of America*, vol. 65, no. 4, pp. 943–950, 04 1979. [Online]. Available: https://doi.org/10.1121/1.382599

[16] C. Ick, A. Mehrabi, and W. Jin, "Blind acoustic room parameter estimation using phase features," in *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023, pp. 1–5.

[17] F. B. Gelderblom, Y. Liu, J. Kvam, and T. A. Myrvoll, "Synthetic data for dnn-based doa estimation of indoor speech," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 4390–4394.

[18] J. B. Allen and D. Berkley, "Image method for efficiently simulating small-room acoustics," *The Journal of the Acoustical Society of America*, vol. 65, no. 4, pp. 943–950, 04 1979. [Online]. Available: https://doi.org/10.1121/1.382599

[19] R. Scheibler, E. Bezzam, and I. Dokmanic, "Pyroomacoustics: A python package for audio room simulations and array processing algorithms," *CoRR*, vol. abs/1710.04196, 2017. [Online]. Available: http://arxiv.org/abs/1710.04196

[20] M. R. Schroeder, "New method of measuring reverberation time," *The Journal of the Acoustical Society of America*, vol. 37, no. 3, pp. 409–412, 1965. [Online]. Available: https://doi.org/10.1121/1.1909343

[21] K. Shimada, Y. Koyama, S. Takahashi, N. Takahashi, E. Tsunoo, and Y. Mitsufuji, "Multi-accdoa: Localizing and detecting overlapping sounds from the same class with auxiliary duplicating permutation invariant training," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Singapore, Singapore, May 2022.

[22] E. Fonseca, X. Favory, J. Pons, F. Font, and X. Serra, "Fsd50k," Oct. 2020. [Online]. Available: https://doi.org/10.5281/zenodo.4060432

[23] K. Shimada, Y. Koyama, S. Takahashi, N. Takahashi, E. Tsunoo, and Y. Mitsufuji, "Multi-accdoa: Localizing and detecting overlapping sounds from the same class with auxiliary duplicating permutation invariant training," 2022.

[24] T. N. T. Nguyen, D. L. Jones, K. N. Watcharasupat, H. Phan, and W.-S. Gan, "SALSA-lite: A fast and effective feature for polyphonic sound event localization and detection with microphone arrays," in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, may 2022. [Online]. Available: https://doi.org/10.1109%2Ficassp43922.2022.9746132

SPEECH OBFUSCATION IN MEL SPECTRA THAT ALLOWS FOR CENTRALISED ANNOTATION AND CLASSIFICATION OF SOUND EVENTS

Michiel Jacobs^{1,2,3}, *Lode Vuegen*^{1,2,3}, *Suraj Khan*^{1,2,3}, *Peter Karsmakers*^{1,2,3*}

^{1} KU Leuven, Dept. of Computer Science, Kleinhoefstraat 4, B-2440 Geel, Belgium
^{2} Flanders Make @ KU Leuven
^{3} Leuven.AI - KU Leuven Institute for AI
Corresponding authors: {michiel.jacobs, peter.karsmakers}@kuleuven.be

ABSTRACT

Nowadays, computerised Sound Event Classification (SEC) aids in several applications, e.g. monitoring domestic events in smart homes. SEC model development typically requires data collected from a diverse set of remote locations. However, this data could disclose sensitive information about uttered speech that might have been present during the acquisition.

In this work, three data preprocessing techniques are investigated that obstruct recognising semantics in speech, but retain the required information in the data for annotating sound events and SEC model development. At the remote location, the data are first preprocessed before transferring to a central place. At the central location, speech should not be interpretable anymore, while still having the opportunity to annotate data with relevant sound event labels. For this purpose, starting from a log-mel representation of the sound signals, three speech obfuscation techniques are assessed: 1) calculating a moving average of the log-mel spectra, 2) sampling a few of the most energetic log-mel spectra and 3) shredding the log-mel spectra. Both intelligibility and SEC experiments were carried out. All considered techniques proved effective in obfuscating speech, while still allowing SEC. For stationary sound events, calculating the moving average of the log-mel spectra is recommended, as well as shredding the log-mel spectra. For impulsive sound events, sampling a few of the most energetic log-mel spectra is recommended.

Index Terms— Speech obfuscation, sound event classification, log-mel spectra

1. INTRODUCTION

Since the past couple of years, there is a growing trend in utilising sound to monitor certain processes. Sound monitoring allows to continuously perceive environments in an automated manner using machine or deep learning (resp. ML, DL) models, e.g. to predict when machine components are about to fail [1], to classify home activities [2] or to automatically interpret auscultation sounds for disease monitoring [3]. Next to the signals of interest, the microphones used are prone to picking up speech signals as well. These speech signals could potentially contain private and/or sensitive information. Therefore, speech obfuscating techniques have to be applied to the recorded sound.

In this work, three preprocessing techniques are evaluated that make the speech present in the sound data unintelligible. Each of these techniques can be run on an extreme edge device with minimal computational overhead, prior to communicating the sound data to the central location. This way, speech is obfuscated while still retaining enough information to centrally perform the tasks of Sound Event Classification (SEC). Moreover, although no words can be recognised, the data still allow human annotators to add class labels to the restored sound data at the central location for refining the ML models.

The main contributions of this work are the following: a) study the effects of three different preprocessing techniques for speech obfuscation, b) perform speech and event intelligibility experiments to study the effect of these speech obfuscation techniques on sound intelligibilty (speech should be obfuscated, while sound events should still be identifiable), and c) perform domestic SEC experiments to study the effect of these speech obfuscation techniques on SEC model classification performance.

The remainder of this paper is organised as follows: Section 2 discusses the related work. Section 3 explains the obfuscation techniques studied. Next, Section 4 describes the intelligibility and classification experiments. Section 5 then covers the results of the intelligibility and the classification experiments. Next, Section 6 discusses the results and findings. Finally, Section 7 summarises this work and forms the conclusion.

2. RELATED WORK

Kumar, Nguyen, Zeng et al. [4] presented subsampling and sound shredding techniques, both applied on mel-frequency cepstral coefficients (MFCC). With subsampling, MFCC feature vectors get thrown out of the sequence in a nonrandom manner. When applying subsampling, the event duration plays an important factor. The time duration of the shortest event (class) has to exceed the subsampling period, since otherwise this event can completely get lost when the corresponding frames are thrown away. Therefore, subsampling MFCCs at regular time intervals is considered not relevant in this research, since it could remove events with short duration. With sound shredding, blocks of MFCCs (units) get randomly shuffled inside of a so-called snippet. The authors indicated that subsampling and sound shredding are valid speech obfuscation techniques that still allow context, gender and speaker recognition using both a k-nearest neighbour (kNN) and a support vector machine (SVM).

When opting to interpret the sound events centrally, another possibility would be to add a speech filter to the edge device. This way, detected speech is simply not sent to the central server. One example would be a so-called Voice Activity Detector (VAD). The drawback of such a VAD is that when an event of interest overlaps a speech signal, it will be discarded and the event gets lost [5]. Also,

---

[*] Funding provided by Flanders Innovation & Entrepreneurship Agency (VLAIO) and Flanders Make.

the use of such speech filters is more complex in terms of computation as compared to the techniques considered in this work.

The work of Larson, Lee, Liu et al. [6] focussed on detecting cough, while disguising speech. The authors proved that ten principal components suffice for classifying the cough sounds. When the number of principal components was increased to 25, the quality of the coughing sound was good and 84% of the spoken words was concealed. However, in the context described in this work such preprocessing is likely to hinder the post-hoc labelling of any audio events that are present in the data.

Chen, Adcock and Krishnagiri [7] used a methodology that identified the vocalic regions using a vocalic syllable detector and replaced the local vocalic linear predictive coefficients (LPC) with those of pre-recorded vowels. Speech intelligibility experiments showed that this methodology can reduce the word recognition rate to 7%. Furthermore, Liaqat, Nemati, Rahman et al. [5] applied this methodology to detect coughs. The mean classification accuracy of the raw audio was 75.86%, while the mean classification accuracy of the filtered audio was 75.75%. The t-test p-value equalled 0.985, thus showing no significant difference between the raw and filtered classification accuracies.

3. METHODOLOGY

This section starts by discussing the calculation of the log-mel spectra, which are the features most commonly used for sound event classification using DL techniques. Next, it discusses which speech obfuscation techniques were applied in the experiments: 1) calculating the moving average of the log-mel spectra, 2) sampling the most energetic log-mel spectra and 3) shredding the log-mel spectra. Finally, this section outlines the steps required to transform the processed log-mel spectra back into a time-domain sound signal (sound restoration).

3.1. Feature extraction: log-mel spectra

The most popular choice for acoustical features in combination with DL are the so-called log-mel spectra [8]. To calculate the short-time Fourier transform (STFT), the following parameters were used: 32 milliseconds (ms) window length, 16 ms hop length (50% overlap), and Hamming window. Finally, the STFT frames were converted into mel frames using a 64-dimensional mel filterbank and the logarithm was taken.

3.2. Speech obfuscation techniques

When averaging consecutive log-mel spectra, nonstationary speech signals get diffused when they are represented by an aggregated log-mel spectrum spanning a larger time horizon. The larger the number of frames being averaged, the more difficult it becomes to restore the original speech afterwards. As can be seen in Table 1, three moving average (MA) configurations were tested. First, *MA-light* refers to averaging over a sliding window having a length of 4 frames (80 ms) and a step of 3 frames (48 ms). Second, *MA-medium* refers to averaging over a sliding window having a length of 8 frames (144 ms) and a step of 5 frames (80 ms). Third, *MA-heavy* refers to averaging over a sliding window having a length of 12 frames (208 ms) and a step of 7 frames (112 ms).

When sampling the most energetic windows, a sliding window is moved over the mel frequency domain audio signal within a nonoverlapping larger segment. At each position, the energy (sum of squares inside smaller, overlapping segment) is calculated and only those segments having highest energy are retained. As can be seen in Table 1, again three configurations were tested. First, with *ENERGY-light* a block of 56 contiguous (912 ms) frames is taken and is replaced by the block of 18 contiguous (304 ms) frames having the highest energy. Second, with *ENERGY-medium* a block of 112 contiguous frames (1,808 ms) is taken and is replaced by the block of 18 contiguous frames having the highest energy. Third, with *ENERGY-heavy* a block of 168 contiguous frames (2,704 ms) is taken and is replaced by the block of 18 contiguous frames having the highest energy.

With sound shredding, log-mel frames get randomly shuffled inside of a so-called snippet, which is a region of contiguous log-mel spectra wherein sound shredding is applied [4]. Two parameters have to be defined, i.e. the unit size refers to the number of frames that are seen as a whole (a block of contiguous, adjoining frames), and the number of units inside one snippet. Again, three configurations were tested and can be found in Table 1. First, *SHRED-light* uses a snippet size of 3 units (208 ms). Second, *SHRED-medium* uses a snippet size of 6 units (400 ms). Third, *SHRED-heavy* uses a snippet size of 16 units (1,040 ms). In all three configurations the unit size was kept at 4 contiguous frames (80 ms), since this was required by the convolutional kernels of the DL-based SEC models in the automated classification experiments. One limitation of sound shredding is that all sound information is kept, i.e. given sufficient effort an attacker could still rearrange the units in the correct order again.

3.3. Sound restoration

In order to assess the speech and event intelligibility of the obfuscated sound features, these features have to be restored to the time-domain signal. Recall that prior to obfuscation the sound data were transformed to log-mel spectra. To return to the time domain, the log, mel and STFT operations have to be reversed. The logarithm can be perfectly undone without introducing artefacts. The inversion from mel frequency scale back to regular frequency scale can be achieved by equation (1):

$$|\hat{X}[k]|^2 = \sum_{b=0}^{B-1} M^{\dagger}_{bk} m_b \approx |X[k]|^2 \tag{1}$$

where $M^{\dagger}$ is the Moore-Penrose pseudo-inverse of the mel matrix M, $|X[k]|^2$ the magnitude spectrogram and m_b the mel value of bin b. This equation guarantees that $|\hat{X}[k]|^2$ is the best solution with minimum norm [9]. This inversion of mel frequency might introduce minor artefacts. The inverse STFT operation was performed using NumPy's `numpy.fft.irfft` function and corresponding phase information [10].

4. EXPERIMENTS

4.1. Speech intelligibility experiment

The speech intelligibility experiment aimed to evaluate the level of obfuscation by having participants grade the restored audio. The data used in this experiment were derived from the Mozilla Common Voice Dutch Subset (v10.0) [11]. A subset of 27 sound files was taken from the Dutch (NL) dataset. Messages of varying length (6 to 9 words) and of both male and female speakers were included.

Twelve native Dutch-speaking participants each got one of three sets of 27 sound recordings obfuscated with varying techniques and configurations. The participants had to grade their obfuscated recordings on an ordinal scale from one to three. Herein,

Table 1: The abbreviations used in the experiments, alongside a brief description of the corresponding configuration. “Light” always refers to the least obfuscating technique, while “heavy” refers to the most obfuscating configuration.

Group	**Abbreviation**	**Description**
	baseline	32 ms STFT window size and 16 ms STFT step size, 64 log-mel bins.
Moving average of log-mel spectra	MA-light	Moving average over 4 log-mel frames, step size of 3 log-mel frames.
	MA-medium	Moving average over 8 log-mel frames, step size of 5 log-mel frames.
	MA-heavy	Moving average over 12 log-mel frames, step size of 7 log-mel frames.
Sampling log-mel spectra	ENERGY-light	For each block of 56 frames, apply a sliding window with length 18 frames and hop size 1 frame and retain the 18 frames having highest energy.
	ENERGY-medium	For each block of 112 frames, apply a sliding window with length 18 frames and hop size 1 frame and retain the 18 frames having highest energy.
	ENERGY-heavy	For each block of 168 frames, apply a sliding window with length 18 frames and hop size 1 frame and retain the 18 frames having highest energy.
Shredding log-mel spectra	SHRED-light	Sound shredding with unit size: 4 log-mel frames, snippet size: 3 units.
	SHRED-medium	Sound shredding with unit size: 4 log-mel frames, snippet size: 6 units.
	SHRED-heavy	Sound shredding with unit size: 4 log-mel frames, snippet size: 16 units.

a score of ’1’ represented sound that is completely incomprehensible, a score of ’2’ represented sound that had a portion of the words comprehensible, while a score of ’3’ referred to perfectly understandable audio. The mean intelligibility score then represented the mean grade for each preprocessing obfuscation method.

Next to the ordinal score, each participant had to write down the message he/she understood. By comparing the understood message and the true transcription, an objective measure of the obfuscated sound quality could be made. In case a participant noted the sound recording as ’2’ but none of the words in the message he/she understood were correct, then the score was altered afterwards to ’1’. In case the participant assessed the recording as ’3’ but the sentence understood was different or incomplete, then the score was changed to ’2’. If the assigned score equalled ’3’ and the understood message differed by only a single word as compared to the transcription and the meaning of the sentence did not become very different, only then the score was kept as ’3’.

In our speech intelligibility experiment, phase information was not discarded when calculating the STFT to simulate the best reconstruction possible (worst-case scenario from the point of speech intelligibility). Therefore, this information could be used during reconstruction. The baseline had the same transformation and reconstruction applied.

4.2. Sound event intelligibility experiment

The sound event intelligibility experiment is similar to the speech intelligibility experiment and differs only in the type of sound to label, i.e. the same participants had to recognise varying domestic sound events in 18 obfuscated recordings. The event classes are summarised in Table 2 and originate from the same dataset as used in the classification experiment (Section 4.3). The labelled events were graded in a binary true/false manner, i.e. a correct label received score ’1’, while an incorrect label received ’0’. The participants did not receive any prior knowledge about the recording procedure (e.g. microphone location) that could help them.

In our event intelligibility experiment, the phase information was discarded after calculating the STFT and was replaced by a random Gaussian noise phase (worst-case scenario from the point of event intelligibility). The baseline had the same transformation and reconstruction applied.

4.3. Sound event classification experiment

For the domestic event classification task, the data and classifier model from Vuegen and Karsmakers [12] were used. The considered dataset contains domestic sound events collected from 72 home environments. In total, data for eight different domestic sound events are available. The recordings were made using a sampling frequency equal to 32 kHz and each sample had a 16-bit resolution. In total, 47.7 hours of data were recorded, spread out over 1519 recordings. Table 2 gives an overview of the dataset distribution.

Table 2: Overview of the dataset used in the SEC task and the event intelligibility experiment. “*Background*” refers to silence and sounds that do not belong to any of the other classes. [12].

Class	**Hours**	**Recordings**
Background	10.5	205
Door & window	5.3	141
Faucet & shower	9.3	386
Footstep	4.2	220
Kitchen hood	4.0	140
Speech	4.9	217
Toilet	5.5	136
Radio & television	4.0	74

As a classifier model a convolutional neural network (CNN) is used. Its performance is evaluated in a 4-fold cross-validation scheme using the previously discussed log-mel features. The model consists of three convolutional layers having 32 filters and ReLU activation (no pooling), followed by one fully-connected layer having 64 neurons with ReLU activation and finally one fully-connected output layer of 8 neurons with softmax activation (8 classes). The dimensions of the convolutional kernel were 4×4, with a stride of 1×4. Note that in case sound shredding is used as an speech obfuscation technique, the horizontal stride of the convolutional kernel was modified to have a value of 4. As such the kernel always spanned a single shredding unit (4 log-mel frames). This way, a kernel never covered a mix of two neighbouring shredded units which are expected to have an unnatural transient from one unit to the other. Zero padding was added to keep the correct dimensions.

The input dimension of the CNN models can be found in Table 3 and was set to one second for the baselines of MA and SHRED techniques. For the ENERGY techniques, the input of the baseline CNN was thrice the segment length.

5. RESULTS

Two types of experiments were carried out to test both the comprehensibility of speech and events, and the SEC model performance on the obfuscated log-mel spectra. The first set of experiments tried to assess the level of speech obfuscation through intelligibility experiments, while the second experiment assessed SEC performance. Table 1 lists the abbreviations used, alongside a brief description of each of the nine tested configurations.

5.1. Speech intelligibility experiment

The results of the speech intelligibility experiment can be found in Table 4a. As was mentioned in Section 4.1, a mean intelligibility score equal to 1 represents a perfect obfuscation, while a mean intelligibility score equal to 3 represents perfectly comprehensible audio. It can be seen that both MA-medium and MA-heavy are able to achieve the best obfuscation in this preliminary speech intelligibility experiment. Furthermore, ENERGY-heavy and SHRED-heavy can be recommended as well, since both have a mean opinion score below 1.20. SHRED-light performs worst in obfuscating speech.

5.2. Event intelligibility experiment

In the event intelligibility experiment, participants had to label obfuscated domestic events. As can be seen in Table 4b, all of the MA and SHRED obfuscation techniques (with the exception of MA-medium and SHRED-medium) score above 0.70, while all ENERGY obfuscating techniques score less.

5.3. Classification

The results of the domestic event classification experiment are presented in Table 3. It can be seen that all models have comparable classification results.

Table 3: Results of the CNN classification experiment. The baseline always spanned the same time horizon at the network's input.

	Macro average recall ± SD (4 folds; in %)		**Macro F1 ± SD (4 folds; in %)**		**Nr. of input frames**	
	Obfuscated	**Baseline**	**Obfuscated**	**Baseline**	**Obfuscated**	**Baseline**
MA-light	84 ± 0.0	85 ± 0.0	83 ± 0.0	84 ± 0.0	20	61
MA-medium	83 ± 0.0	85 ± 0.0	82 ± 0.5	84 ± 0.5	12	63
MA-heavy	82 ± 0.5	85 ± 0.0	81 ± 0.5	84 ± 0.0	8	61
ENERGY-light	83 ± 0.5	87 ± 0.5	82 ± 0.5	87 ± 0.0	54	168
ENERGY-medium	82 ± 0.0	89 ± 0.0	82 ± 0.6	89 ± 0.5	54	336
ENERGY-heavy	81 ± 0.8	90 ± 0.6	81 ± 1.0	89 ± 0.5	54	504
SHRED-light	84 ± 0.0	84 ± 0.5	83 ± 0.5	83 ± 0.5	60	60
SHRED-medium	85 ± 0.5	85 ± 0.5	85 ± 0.6	85 ± 0.6	72	72
SHRED-heavy	84 ± 0.5	85 ± 0.5	83 ± 0.0	84 ± 0.6	64	64

6. DISCUSSION

The results of the speech intelligibility experiment (Table 4a) are as expected, with the exception of MA-medium and MA-heavy. The MA-medium speech appeared to be less intelligible as compared to MA-heavy, but this could be explained by the limited number of participants and assessments. For all ENERGY techniques, the impulsive events all had perfect classifications and can therefore be recommended for this kind of events.

The results of the event intelligibility experiment (Table 4b) are as expected as well, with the exception of MA-medium and SHRED-medium. Possible reasons for these inconsistencies are the possibility for the participants to choose "I don't know", the limited number of participants and the limited diversity in combinations of obfuscation techniques and event types. In practice, the annotators could also have access to additional information, e.g. the microphone location and the spectrogram representation. This would aid them in annotating the events. Furthermore, when looking at the participants' annotations it can be noted that most mistakes were between "speech" and "radio & television", and "footsteps" and "door & window" (impulsive sounds), and between "faucet & shower", "toilet", "background" and "kitchen hood" (stationary sounds). More experienced annotators would be better at distinguishing these different types of events. Note that our participants were not trained beforehand, which could also explain why the baseline is lower than MA-light and SHRED-light.

In the results of the CNN classification experiment (Table 3), a decrease in performance can be seen with all three ENERGY techniques as compared to their corresponding baselines. This decrease could be explained by the fact that the CNN model has less information at its input. For ENERGY-light, the time at the model's input is reduced by 66.7% as compared to its baseline. For ENERGY-medium this reduction is equal to 83.2%, and for ENERGY-heavy this reduction is equal to 88.8%. A smaller decrease in performance is also noticeable with MA, due to the reduced resolution at the network's input. SHRED does not suffer from a decrease in performance, because the same information is still present at the network's input.

Table 4: Results of the intelligibility experiments.

(a) Speech, lower is better, range $[1, 3]$.

Obfuscation technique	**Mean opinion score ± SD**
Baseline	2.96 ± 0.09
MA-light	1.69 ± 0.43
MA-medium	1.03 ± 0.08
MA-heavy	1.06 ± 0.17
ENERGY-light	1.64 ± 0.33
ENERGY-medium	1.22 ± 0.23
ENERGY-heavy	1.19 ± 0.27
SHRED-light	1.97 ± 0.29
SHRED-medium	1.36 ± 0.42
SHRED-heavy	1.17 ± 0.22

(b) Events, higher is better, range $[0, 1]$.

Obfuscation technique	**Mean score ± SD**
Baseline	0.77 ± 0.20
MA-light	0.83 ± 0.20
MA-medium	0.58 ± 0.20
MA-heavy	0.71 ± 0.19
ENERGY-light	0.67 ± 0.26
ENERGY-medium	0.67 ± 0.30
ENERGY-heavy	0.50 ± 0.35
SHRED-light	0.88 ± 0.14
SHRED-medium	0.58 ± 0.20
SHRED-heavy	0.75 ± 0.22

7. CONCLUSION

In this work, three techniques based on the log-mel spectra were investigated for the purpose of speech obfuscation. A requisite was that sound data could be labelled by human raters at a later point in time, without having intelligible speech in the recordings. The first technique was calculating the moving average of 4 (MA-light), 8 (MA-medium) or 12 (MA-heavy) log-mel frames. The second technique was sampling those windows of log-mel frames having the highest energy, where 18 out of 56 frames (ENERGY-light), 18 out of 112 frames (ENERGY-medium) or 18 out of 168 frames (ENERGY-heavy) were kept. The final technique was sound shredding, where 4 contiguous log-mel frames were kept in a so-called unit. These units were then randomised inside of a snippet of length 3 units (SHRED-light), 6 units (SHRED-medium) or 16 units (SHRED-heavy).

Both a speech and event intelligibility experiment (12 participants) and a SEC classification experiment were carried out. The intelligibility experiment demonstrated that both MA-heavy and SHRED-heavy achieved good speech obfuscation levels, while still having the possibility to label the data. Furthermore, the proposed techniques only had minor impact on the classification performance when evaluating on a dataset with sounds from domestic events, except for the ENERGY techniques.

All considered techniques proved effective in obfuscating speech, while still allowing SEC. For stationary sound events, calculating the moving average or shredding the log-mel spectra is recommended. For impulsive sound events, sampling a few of the most energetic log-mel spectra is recommended.

8. ACKNOWLEDGEMENT

The authors would like to thank Flanders Innovation & Entrepreneurship Agency (VLAIO) and Flanders Make for providing research funding.

9. REFERENCES

[1] B. Boons, M. Verhelst, and P. Karsmakers, "Low power online machine monitoring at the edge," in *2021 International Conference on Applied Artificial Intelligence (ICAPAI)*, 2021, pp. 1–8.

[2] L. Vuegen, P. Karsmakers, B. Vanrumste, and H. Van hamme, "Acoustic event classification using low-resolution multi-label non-negative matrix deconvolution," *Journal of the Audio Engineering Society*, vol. 66, no. 5, pp. 369–384, May 2018.

[3] S. H. Lee, Y.-S. Kim, M.-K. Yeo, M. Mahmood, N. Zavanelli, C. Chung, J. Y. Heo, Y. Kim, S.-S. Jung, and W.-H. Yeo, "Fully portable continuous real-time auscultation with a soft wearable stethoscope designed for automated disease diagnosis," *Science Advances*, vol. 8, no. 21, p. eabo5867, 2022. [Online]. Available: https://www.science.org/doi/abs/10.1126/sciadv.abo5867

[4] S. Kumar, L. T. Nguyen, M. Zeng, K. Liu, and J. Zhang, "Sound shredding: Privacy preserved audio sensing," in *Proceedings of the 16th International Workshop on Mobile Computing Systems and Applications*, ser. HotMobile '15. New York, NY, USA: Association for Computing Machinery, 2015, pp. 135–140.

[5] D. Liaqat, E. Nemati, M. Rahman, and J. Kuang, "A method for preserving privacy during audio recordings by filtering speech," in *2017 IEEE Life Sciences Conference (LSC)*, 2017, pp. 79–82.

[6] E. C. Larson, T. Lee, S. Liu, M. Rosenfeld, and S. N. Patel, "Accurate and privacy preserving cough sensing using a low-cost microphone," in *Proceedings of the 13th International Conference on Ubiquitous Computing*, ser. UbiComp '11. New York, NY, USA: Association for Computing Machinery, 2011, pp. 375–384.

[7] F. Chen, J. Adcock, and S. Krishnagiri, "Audio privacy: Reducing speech intelligibility while preserving environmental sounds," in *Proceedings of the 16th ACM International Conference on Multimedia*, ser. MM '08. New York, NY, USA: Association for Computing Machinery, 2008, pp. 733–736.

[8] A. Politis, A. Mesaros, S. Adavanne, T. Heittola, and T. Virtanen, "Overview and evaluation of sound event localization and detection in dcase 2019," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 684–698, 2021.

[9] L. E. Boucheron and P. L. De Leon, "On the inversion of mel-frequency cepstral coefficients for speech enhancement applications," in *2008 International Conference on Signals and Electronic Systems*, 2008, pp. 485–488.

[10] NumPy. Numpy v1.23 `numpy.fft.irfft` function. [Online]. Available: https://numpy.org/doc/1.23/reference/generated/numpy.fft.irfft.html

[11] R. Ardila, M. Branson, K. Davis, M. Henretty, M. Kohler, J. Meyer, R. Morais, L. Saunders, F. M. Tyers, and G. Weber, "Common voice: A massively-multilingual speech corpus," in *Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020)*, 2020, pp. 4211–4215.

[12] L. Vuegen and P. Karsmakers, "Real-time on-edge classification: an application to domestic acoustic event recognition," in *Proceedings of the 29th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, 2021.

FALL-E: A FOLEY SOUND SYNTHESIS MODEL AND STRATEGIES

Minsung Kang, Sangshin Oh*, Hyeongi Moon, Kyungyun Lee, Ben Sangbae Chon*

Gaudio Lab, Inc., Seoul, South Korea. `bc@gaudiolab.com`

ABSTRACT

This paper introduces FALL-E, a foley synthesis system and its training/inference strategies. The FALL-E model employs a cascaded approach comprising low-resolution spectrogram generation, spectrogram super-resolution, and a vocoder. We trained every sound-related model from scratch using our extensive datasets, and utilized a pre-trained language model. We conditioned the model with dataset-specific texts, enabling it to learn sound quality and recording environment based on text input. Moreover, we leveraged external language models to improve text descriptions of our datasets and performed prompt engineering for quality, coherence, and diversity. FALL-E was evaluated by an objective measure as well as listening tests in the DCASE 2023 challenge Task 7. The submission achieved the second place on average, while achieving the best score for diversity, second place for audio quality, and third place for class fitness.

Index Terms— Sound synthesis, foley, generative audio

1. INTRODUCTION

Generative AI has seen significant progress in recent years, particularly in the domains of images and text. However, the progress in sound generation has been comparatively slower.

In the field of sound generation, numerous impressive works have been introduced including text-to-sound models such as AudioGen [1] and AudioLDM [2]. In addition, several works can be used as modules of the whole system such as Hifi-GAN [3], SoundStream, EnCodec [4, 5], latent diffusion [6], and spectrogram super-resolution [7].

Furthermore, in text-input and text-conditioned generation, models such as T5 [8], GPT [9, 10], text prompt engineering [11, 12], and diffusion with conditioned generative models [1, 2, 13, 14] have been introduced. As the behavior of large deep learning models is somewhat difficult to analyze, these works enable us as users to steer the model using carefully selected text inputs.

In this context, we present a novel approach to foley synthesis that utilizes a cascade system composed of low-resolution spectrogram generation, a super-resolution module, and a vocoder. Our system represents our submission to the DCASE 2023 Task 7 - Foley Synthesis Challenge (Track A) [15]. While we report objective measures with respect to the official evaluation set, our ultimate goal is to develop sound generation models that extend beyond the challenge's scope.

In Section 2, we introduce our model architecture, FALL-E, detailing the function of each module and how they work in tandem. In Section 3, we provide an in-depth analysis of our evaluation results, showcasing the effectiveness of our approach in various settings. Lastly, in Section 4, we summarize our contributions and highlight future directions for our work.

2. FALL-E

2.1. Architecture

The cascade system, which involves generating low-resolution images or features and subsequently obtaining higher-resolution results, has been extensively utilized in generation models[14, 16, 17]. We adopt this approach to generate foley sound. Our proposed system, FALL-E, consists of three separately-trained models: diffusion-based low-resolution spectrogram generation model and upsampling model, and a GAN-based mel-spectrogram inversion network.

Text Encoder of FALL-E is a pre-trained Flan-T5, an instruction finetuned-variant of a T5 model which shows better performance for various applications [18]. The class category is mapped to predefined text prompts from the prompt corpus. Then Flan-T5 converts the text prompts into a sequence of text embedding, which is input to the Low-resolution Spectrogram Generator.

Low-resolution Spectrogram Generator is based on Glide, a diffusion generative model for text-to-image generation [14]. This module produces a low-resolution spectrogram. Specifically, it generates a 32×128 feature map for a 128-bin, 512-frame mel-spectrogram. The module employs a U-Net shaped architecture with 5 residual blocks in both the encoder and decoder. In the encoder, each block comprises 2 convolution layers and an additional upsampling layer with the number of convolution channels in each block increasing linearly from 192. The decoder is a mirrored version of the encoder.

Spectrogram Upsampler is another diffusion-based

---

*Equal contribution.

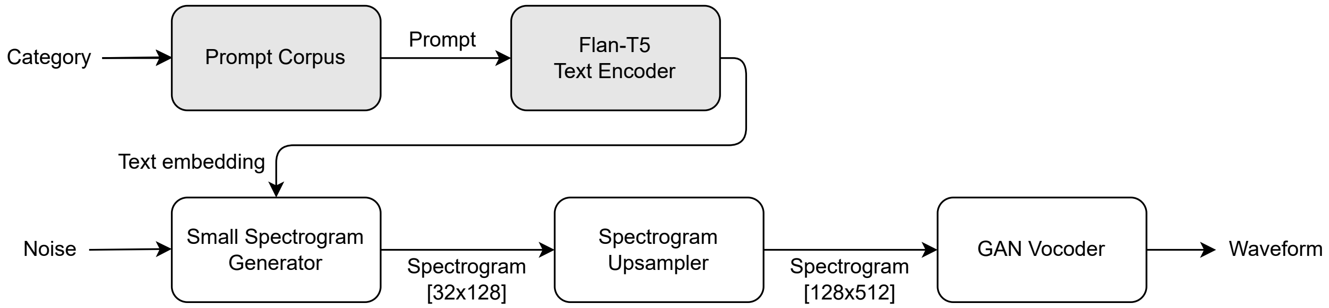


Figure 1: The overall system. Shaded blocks indicate the rule-based or pretrained models.

Module name	**Num. of Parameters**
Text Encoder	110 M
Low-res. Spec. Generator	318 M
Spectrogram Upsampler	89 M
Mel Inversion Network	125 M
Total	642 M

Table 1: The number of parameters in each module.

generative model that synthesizes mel-spectrograms from a given low-resolution spectrogram. The overall architecture of this model is a U-Net that is similar to Low-resolution Spectrogram Generator but with a different number of blocks and channels. Its encoder and decoder consists of 4 blocks and the number of convolution channel in the first block is 128. Unlike Low-resolution Spectrogram Generator, it isn't conditioned on text; it is only conditioned by the low-resolution mel-spectrogram feature.

Mel Inversion Network converts the generated mel-spectrograms into waveforms. Based on HiFi-GAN [19] and BigVGAN [20], we add FiLM [21] layers as a residual connection. The additional layer helps the model to preserve signal characteristics of the conditioned spectrogram and improves the phase reconstruction quality. We open-sourced this mel inversion network, GOMIN.[1]

The whole system has 642M parameters in total. Its details are described in Table 1.

2.2. Datasets

Training datasets include various sources across private and public audio datasets, including AudioSet [22], CLOTHO [23], FreeToUseSounds.[2], Sonniss,[3] WeSoundEffects,[4] and ODEON.[5] To prevent data imbalances or the potential risks of model misbehavior, samples with speech or musical contents are filtered out based on their metadata. After the filtering, we used 3,815 hours of audio signals for training.

Name	**AQ**	**Dataset Size**		**Modality**		
		Dura.	**N. Files**	**Lb**	**Cp**	**Vd**
Public dataset						
AudioSet	*noisy*	5420 *h*	1,951,460	✓		✓
Clotho	*noisy*	37.0 *h*	5,929		✓	
Free To Use Sounds	*noisy*	175.7 *h*	6,370		✓	
Sonnis Game Effects	*clean*	84.6 *h*	5,049		△	
WeSoundEffects	*clean*	12.0 *h*	488		△	
Odeon Sound Effects	*clean*	19.5 *h*	4,420		△	
Private dataset						
Private dataset	*clean*	3829 *h*	371,116		△	

Table 2: A list of audio datasets. AQ: audio quality, Dura.: duration, N. Files: number of files. Modality columns refer to the existence of labels, captions, and videos, respectively. *Clean* recording: Audio is recorded in well-treated environments and mastered for professional content production. *Noisy*: dataset contains environmental noises or interference signals. △: Textual information included, not necessarily captions. This table is partially from [1] and [24]

2.3. Prompting Strategy

Text conditioning can be optimized or engineered to improve the model behavior. One of our focuses was to control the recording condition/environment of the generated signals so that the model can learn from crowd-sourced, noisy datasets (low recording SNR) as well, while being able to produce high-quality audio. Among the datasets we used, AudioSet, Clotho, and Free To Use Soounds were "*noisy*" dataset. We append a special token that indicates *noisy dataset* to the text input during training. For the other datasets, we append *clean dataset* token. The impact of this additional token will be discussed in Section 3. We also clean the text label (i.e., text normalization) by dropping some stop words and numbers.

Our model is designed to process natural language text. When we directly use the sound class name as input, we

---

[1] https://github.com/ryeoat3/gomin

[2] https://www.freetousesounds.com/all-in-one-bundle/

[3] https://sonniss.com/gameaudiogdc

[4] https://wesoundeffects.com/we-sound-effects-bundle-2020

[5] https://www.paramountmotion.com/odeon-sound-effects

Sound class	**WAS ↑**	**Qual. ↑**	**Fit. ↑**	**Div. ↑**	**FAD ↓**
Dog bark	7.984	7.612	8.223	8.250	11.456
Footstep	6.865	6.455	7.082	7.250	5.959
Gun shot	7.255	6.814	7.573	7.500	3.021
Keyboard	6.989	6.814	7.157	7.000	4.090
Motor vehicle	6.881	6.446	7.131	7.250	6.173
Rain	6.243	5.928	6.306	6.750	5.738
Sneeze & cough	6.553	6.528	6.606	6.500	2.340
Average	6.967	6.657	7.154	7.214	5.540

Table 3: DCASE 2023 task 7 official results across all sound classes. **WAS** indicates "Weighted Average Score", **Qual.** refers to audio quality, **Fit.** to category fitness, and **Div.** to diversity within the class.

Model	**WAS ↑**	**Qual. ↑**	**Fit. ↑**	**Div. ↑**	**FAD ↓**
Surrey	7.886	7.546	**8.419**	7.500	**3.621**
LINE	7.339	6.444	7.529	**8.750**	3.679
HEU	4.877	3.800	5.142	6.500	5.685
Baseline	2.688	2.930	2.447	-	13.412
Ours	**7.984**	**7.612**	8.223	8.250	11.456

Table 4: Comparison of the official results for the "Dog Bark" sound class in DCASE 2023 Task 7 with other submission models.

have observed that the diversity of the generated sound is not as sufficient as that of real sound samples from the training dataset. On the other hand, by employing a variety of text prompts for each class, our model is capable of generating a more diverse range of sounds. For example, for footstep sound class, we can provide prompts such as:"*clean* recording, footsteps on snow", "*clean* recording, footsteps, running", and "*clean* recording, footsteps in a large room".

3. EVALUATION AND ANALYSIS

In DCASE 2023 Task 7, our model achieved 2nd place in subjective scores and 3rd place in FAD scores, with a specific breakdown of 2nd place in Audio Quality, 3rd place in Category Fit, and 1st place in Diversity. Table 3 presents the details of each sound class. In this section, we will delve deeply into the topics of objective and subjective evaluations.

The right column in Table 3 presents FAD scores across all sound classes using the official evaluation repositories.[6] Our approach outperforms the baseline approach in all classes, with notable improvements observed in the rain and moving motor vehicle classes. Furthermore, the subjective quality is significantly improved by our model in all classes. It should be acknowledged that FAD scores may not be indicative of other important aspects of audio quality such as clarity, high-SNR, and high-frequency components. Also, as FAD measures similarity between a reference set and a test set, improvement beyond reference is mismeasured as a degradation, including quantization noise and codec noise. As evidenced in Table 3 and Table 4, our performance in the "Dog Bark" sound class received the worst score in FAD, while achieving the highest score in the Weighted Average Score (WAS).

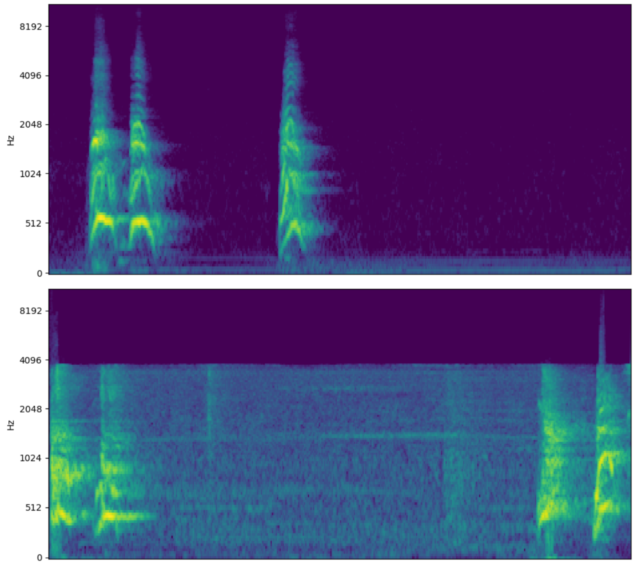


Figure 2: Mel-spectrograms of the generated audio samples using different recording environment prefixes. Prompts for images are (*top*) "*clean* recording, puppy bark," and (*botton*) "*noisy* recording, puppy bark," respectively

Our model was developed to generate high-quality audio suitable for real-world scenarios using the environment and audio quality prefixes. Despite most of the audio samples in our training dataset exhibiting poor audio quality due to background noise, babble noise, wind noise, device noise, and codec distortion, we confirmed our model produces high-quality audio. As discussed in Section 2.3, we controlled the audio sample quality by adding a special token as a prefix to the original text. Given that audio quality cannot be evaluated objectively, we conducted a informal listening test for the same text with both *clean* and *noisy* prefixes. Depending on the prefix used, we observed impressive improvements in sound quality across all sound classes. As illustrated in Figure 2, we can clearly observe that the use of the *clean* prefix had a discernible impact on the audio quality, as indicated by the mel-spectrogram images. This type of model steering by prompting has been popular in other domains, and to our best knowledge, our work is the first work that successfully shows it in audio generation.

To improve quality for mel-spectrogram inversion, we trained our own network based on HiFi-GAN [19] and BigV-

---

[6] https://github.com/DCASE2023-Task7-Foley-Sound-Synthesis

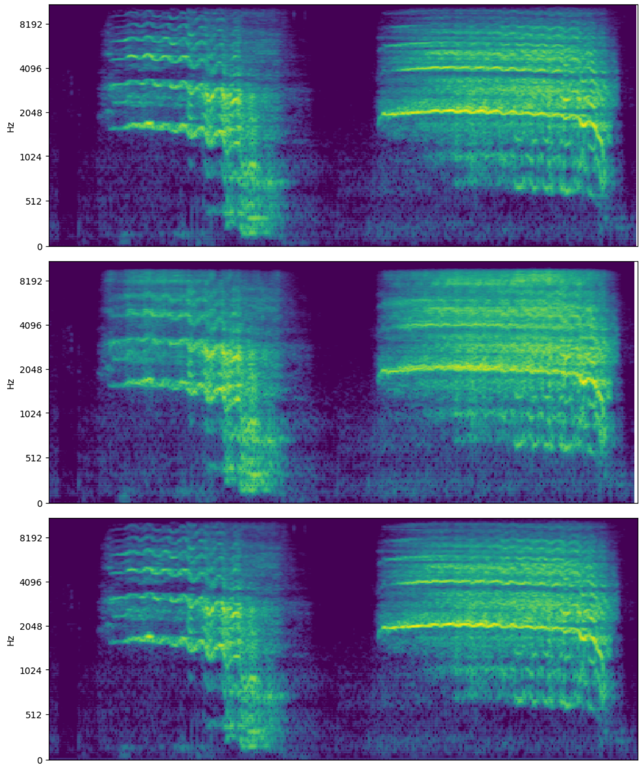


Figure 3: Mel-spectrogram for reconstructed samples. (*top*) ground-truth waveform, (*middle*) reconstructed with challenge baseline model, and (*bottom*) Our GOMIN system.

GAN [20] as explained above. Thanks to its bigger capacity and other architectural improvements, it showed better performance for overall sound categories. Compared to the baseline model,[6] our model well reconstructs tonal or harmonic components in the signal especially when the input mel-spectrograms include complex composition.

4. CONCLUSION

In this paper, we have presented FALL-E, Gaudio's foley synthesis system. FALL-E employs a cascaded approach with low-resolution spectrogram generation, a super-resolution module, and a vocoder. Our system was submitted to the DCASE 2023 Task 7 - Foley Synthesis Challenge (Track A), and we have reported the objective measure with respect to the official evaluation set. Through our extensive dataset and language model conditioning, as well as prompt engineering, we have achieved high-quality, diverse, and coherent sound generation results.

There is a vast potential for the development of generative AI in the audio domain. As technology continues to advance, new possibilities for sound generation arise, and the potential applications of this technology are vast. For example, in film and game production, foley synthesis could be used to produce more realistic sound effects, saving time and resources compared to traditional foley artistry. We believe that FALL-E, along with other works in the field, will pave the way for future advancements in generative audio technology, and we look forward to the continued development of this exciting area of research.

Acknowledgement

This research was supported by Culture, Sports and Tourism R&D Program through the Korea Creative Content Agency grant funded by the Ministry of Culture, Sports and Tourism in 2022 (Project Name: R&D on AI Text-to-Sound Generation, Project Number: RS-2023-00229204, Contribution Rate: 100%). The authors are grateful for Naver D2 Startup Factory and Naver Cloud Platform for supporting the GPU resources for this research.

We would like to highlight the clear arrangement implemented to ensure fairness and prevent any unfair advantage in the task. The conflict of interest with one of the organizers of this task was openly disclosed to the organizers, and the co-organizer affiliated with the institution in question remained uninvolved once the finalists were objectively determined. Additionally, during the subjective evaluation phase, other organizers were kept blind to the submission numbers to maintain impartiality. These measures were put in place to uphold the integrity and impartiality of the task evaluation process.

5. REFERENCES

[1] F. Kreuk, G. Synnaeve, A. Polyak, U. Singer, A. Défossez, J. Copet, D. Parikh, Y. Taigman, and Y. Adi, "Audiogen: Textually guided audio generation," *arXiv preprint arXiv:2209.15352*, 2022.

[2] H. Liu, Z. Chen, Y. Yuan, X. Mei, X. Liu, D. Mandic, W. Wang, and M. D. Plumbley, "Audioldm: Text-to-audio generation with latent diffusion models," 2023.

[3] J. Kong, J. Kim, and J. Bae, "Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis," *Advances in Neural Information Processing Systems*, vol. 33, pp. 17 022–17 033, 2020.

[4] N. Zeghidour, A. Luebs, A. Omran, J. Skoglund, and M. Tagliasacchi, "Soundstream: An end-to-end neural audio codec," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 495–507, 2021.

[5] A. Défossez, J. Copet, G. Synnaeve, and Y. Adi, "High fidelity neural audio compression," *arXiv preprint arXiv:2210.13438*, 2022.

[6] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 10 684–10 695.

[7] L. Sheng, D.-Y. Huang, and E. N. Pavlovskiy, "High-quality speech synthesis using super-resolution mel-spectrogram," *arXiv preprint arXiv:1912.01167*, 2019.

[8] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *The Journal of Machine Learning Research*, vol. 21, no. 1, pp. 5485–5551, 2020.

[9] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, *et al.*, "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.

[10] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[11] V. Liu and L. B. Chilton, "Design guidelines for prompt engineering text-to-image generative models," in *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, 2022, pp. 1–23.

[12] J. White, Q. Fu, S. Hays, M. Sandborn, C. Olea, H. Gilbert, A. Elnashar, J. Spencer-Smith, and D. C. Schmidt, "A prompt pattern catalog to enhance prompt engineering with chatgpt," *arXiv preprint arXiv:2302.11382*, 2023.

[13] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever, "Zero-shot text-to-image generation," 2021.

[14] T. A. Halgren, R. B. Murphy, R. A. Friesner, H. S. Beard, L. L. Frye, W. T. Pollard, and J. L. Banks, "Glide: a new approach for rapid, accurate docking and scoring. 2. enrichment factors in database screening," *Journal of medicinal chemistry*, vol. 47, no. 7, pp. 1750–1759, 2004.

[15] K. Choi, J. Im, L. Heller, B. McFee, K. Imoto, Y. Okamoto, M. Lagrange, and S. Takamichi, "Foley sound synthesis at the dcase 2023 challenge," *In arXiv e-prints: 2304.12521*, 2023.

[16] J. Ho, C. Saharia, W. Chan, D. J. Fleet, M. Norouzi, and T. Salimans, "Cascaded diffusion models for high fidelity image generation," *The Journal of Machine Learning Research*, vol. 23, no. 1, pp. 2249–2281, 2022.

[17] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen, "Hierarchical text-conditional image generation with clip latents," *arXiv preprint arXiv:2204.06125*, 2022.

[18] H. W. Chung, L. Hou, S. Longpre, B. Zoph, Y. Tay, W. Fedus, Y. Li, X. Wang, M. Dehghani, S. Brahma, A. Webson, S. S. Gu, Z. Dai, M. Suzgun, X. Chen, A. Chowdhery, A. Castro-Ros, M. Pellat, K. Robinson, D. Valter, S. Narang, G. Mishra, A. Yu, V. Zhao, Y. Huang, A. Dai, H. Yu, S. Petrov, E. H. Chi, J. Dean, J. Devlin, A. Roberts, D. Zhou, Q. V. Le, and J. Wei, "Scaling instruction-finetuned language models," 2022.

[19] J. Kong, J. Kim, and J. Bae, "Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis," *Advances in Neural Information Processing Systems*, vol. 33, pp. 17 022–17 033, 2020.

[20] S.-g. Lee, W. Ping, B. Ginsburg, B. Catanzaro, and S. Yoon, "Bigvgan: A universal neural vocoder with large-scale training," *arXiv preprint arXiv:2206.04658*, 2022.

[21] E. Perez, F. Strub, H. De Vries, V. Dumoulin, and A. Courville, "Film: Visual reasoning with a general conditioning layer," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[22] J. F. Gemmeke, D. P. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio set: An ontology and human-labeled dataset for audio events," in *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2017, pp. 776–780.

[23] K. Drossos, S. Lipping, and T. Virtanen, "Clotho: An audio captioning dataset," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 736–740.

[24] Y. Wu, K. Chen, T. Zhang, Y. Hui, T. Berg-Kirkpatrick, and S. Dubnov, "Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

LABEL FILTERING-BASED SELF-LEARNING FOR SOUND EVENT DETECTION USING FREQUENCY DYNAMIC CONVOLUTION WITH LARGE KERNEL ATTENTION

Ji Won Kim[1], Sang Won Son[1], Yoonah Song[1], Hong Kook Kim[1,2,]

[1] AI Graduate School, [2] School of EECS
Gwangju Institute of Science and Technology
Gwangju 61005, Korea
{{jiwon.kim, ssw970519, yyaass0531}@gm., hongkook@}gist.ac.kr

Il Hoon Song[3], Jeong Eun Lim[3]

[3]AI Lab.
Hanwha Vision
Seongnam-si, Gyeonggi-do 13488, Korea
{ilhoon, je04.lim}@hanwha.com

ABSTRACT

This paper proposes a convolutional recurrent neural network (CRNN)-based sound event detection (SED) model. The proposed model utilizes frequency dynamic convolution (FDY) with a large kernel attention (LKA) for convolution operations within the CRNN. This is designed to effectively capture time-frequency patterns and long-term dependencies for non-stationary audio events. In addition, we concatenate a pre-trained bidirectional encoder representation from audio transformers (BEATs) embedding with the output of FDY–LKA. This provides the FDY-based feature maps with semantic information. Given the limited labeled data condition of the DCASE Challenge dataset, we first employ the mean-teacher-based semi-supervised learning. Then, we propose label filtering-based self-learning for audio event data selection, when their pseudo labels predicted from the mean-teacher model are strong correlated with given weakly labels. This strategy applies weakly labeled and unlabeled data, and then extends to the AudioSet. We evaluate its performance of the proposed SED model on DCASE 2023 Challenge Task 4A, measuring the F1-score and polyphonic sound detection scores, namely PSDS1 and PSDS2. The results indicate that the proposed CRNN-based model with FDY–LKA improves the F1-score, PSDS1, and PSDS2 in comparison to the baseline for DCASE 2023 Challenge Task 4A. When we apply the BEATs embedding via average pooling to both the baseline and the proposed model, we find that the performance of the proposed model significantly outperforms the baseline, with an F1-score of 6.2%, a PSDS1 score of 0.055, and a PSDS2 score of 0.021. Consequently, our model is ranked first in the DCASE 2023 Challenge Task 4A evaluation for a single model track, and second for an ensemble model.

Index Terms**—**Sound event detection, semi-supervised learning, label filtering-based self-learning, frequency dynamic convolution, large kernel attention, BEATs embedding

1. INTRODUCTION

The objective of sound event detection (SED) is to recognize and classify individual sound events originating from acoustic signals, along with their corresponding time stamps. The potential applications of the SED model have been attracted from audio captioning [1] to various domains, such as wildlife tracking [2], equipment monitoring [3], and medical monitoring [4]. In recent years, SED has been extensively researched using deep learning models [5]. However, a significant challenge in using deep learning for SED is the requirement of strong labels, which are expensive and time-consuming. This problem has led to develop weakly supervised or semi-supervised learning techniques to mitigate such label requirement.

To address this problem, we apply a self-learning strategy based on label filtering to train the proposed SED model when the quantity of labeled training data is limited. The proposed model is based on a convolutional recurrent neural network (CRNN), where the convolution is realized with frequency dynamic convolution (FDY) [6] with large kernel attention (LKA) [7].

As a remedy for limited resources, we use select data from the AudioSet [8] as additional training material. In this context, the audio class of each data item from AudioSet is mapped into that of the DCASE Challenge Task 4A and data belonging to the DCASE audio class are selected. However, even though this approach of using additional AudioSet data improves SED performance [9], it leads to a data imbalance issue. Furthermore, this method tends to include audio data whose characteristics differ from those in the DCASE training set. Thus, we propose an alternative in the form of a label filtering-based self-learning method to select appropriate data from AudioSet by examining the inference probability during model training.

Next, one of the most successful components in detection models is the application of an attention mechanism, which emphasizes semantic knowledge in the feature map. Of late, there have been several types of attention mechanisms, like squeeze-and-excitation (SE) [10] and convolutional block attention module (CBAM) [11], which are designed to accommodate channel and/or spatial information for attention. These mechanisms alter or reshape an image to obtain attention weights, given that images are shift-invariant for classification or detection. However, the spectrogram image of an audio event signal is neither shift-invariant nor stationary, necessitating an attention mechanism with unaltered attention weights.

Inspired by image classification and detection [7], we incorporate LKA into the sound event detection model. Combining this enables us to maintain long-term dependency for the attention, even when the audio signals are non-stationary. To the best of our

---

[*] This work was supported in part by Hanwha Vision Co. Ltd., and by Institute of Information & communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT) (No.2022-0-00963).

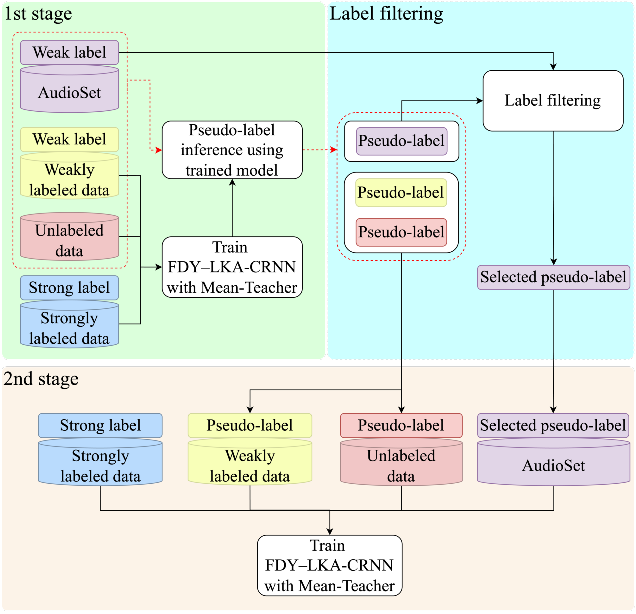


Figure 1: Illustration of the training procedure for the proposed FDY–LKA-CRNN-based SED model, including a label filtering-based self-learning strategy.

knowledge, we are the first to apply this to sound event detection tasks.

Our contributions can be summarized as follows:

- We have developed a label-filtering approach to supplement training data from weakly-labeled out-of-domain sources, such as AudioSet, within a self-learning-based model training framework. As a result, we have improved the detection accuracy of sound event detection.
- Additionally, we have integrated an attention mechanism, the large kernel attention (LKA), proposed for image classification and detection, into the sound event detection model. This is vital as audio signals are inherently non-stationary, necessitating the retention of long-term dependency for the attention.
- We have applied our proposed training strategy to designing SED models for the DCASE 2023 Challenge Task 4A and achieved the best performance in terms of F1-score and PSDSs [12] without an ensemble. Moreover, our ensemble model ranked second.

Following this introduction, Section 2 describes the dataset and input features of the SED model. Section 3 proposes a label filtering-based self-learning strategy applied to the FDY–LKA-CRNN model. Next, Section 4 evaluates the performance of the proposed SED models on the validation dataset task of DCASE 2023 Task 4A. Finally, Section 5 concludes this paper.

2. DATASET

The DCASE 2023 Challenge Task 4A consists of four datasets: weakly labeled data, unlabeled in-domain training data, strongly labeled synthetic data, and strongly labeled real data. All the audio clip data span 10 seconds each. The strongly labeled synthetic dataset is unique in that it is generated by Scraper [13]. The weakly labeled dataset only has class labels and is annotated for 1,578 clips. The unlabeled in-domain training dataset includes 14,412 audio clips. Meanwhile, the real strongly labeled and synthetic datasets comprise 3,470 and 10,000 clips, respectively. In addition to the DCASE dataset, we utilize a subset of AudioSet that includes 18,000 clips with in-domain weak labels.

The following preprocessing steps are employed to prepare the data for input to the model: First, the mono-channel signals are resampled from 44.1 to 16 kHz. Subsequently, the audio signals are divided into frames of 2,048 samples each, with a hop length of 160 samples. Each frame first undergoes a 2,048-point fast Fourier transform (FFT), followed by a 128-dimensional mel-filterbank analysis. This results in input feature dimensions of (1001x128). The extracted mel-spectrogram features are then normalized using the mean and standard deviation of all the training audio samples.

3. PROPOSED FDY–LKA-CRNN-BASED SED MODEL

Fig. 1 illustrates the training procedure of the proposed FDY–LKA-CRNN-based SED model, which employs a label filtering-based self-learning strategy. As depicted in the upper-left arm of the figure, an SED model is initially trained using the mean-teacher approach, where the entire DCASE Challenge Task 4A dataset is utilized. For a detailed procedure of this first-stage training, please refer to the training description in [14]. Subsequently, label filtering is carried out to select audio event data from AudioSet for the training of the second-stage SED model. This selection process is designed to choose audio event data for which the pseudo-labels, predicted from the first-stage SED model, strongly correlate with the weak labels provided by the AudioSet data descriptors. Finally, the second-stage SED model as shown in the lower arm of Fig. 1 is retrained using both the entire DCASE challenge data and the selected AudioSet data.

The following subsections explains the network architecture of the proposed LKA-CRNN-based SED model, LKA-based attention, and label filtering-based self-learning.

3.1. Network architecture

Table 1 displays the network architecture of this proposed model. The model comprises one stem block, six FDY–LKA blocks, one optional fusion block, and one RNN block. Initially, all input features for each audio clip are grouped to form a spectral image of dimensions (1001×128×1), which serves as the input to the stem block. In detail, the stem block consists of one convolutional block with 32 kernels of size (3×3) and a stride of (1×1), which is further processed by batch normalization (BN), gated linear unit (GLU) activation, and a 2×2 average pooling layer. Note that (x×y×z) and (x×y) indicate (frame×frequency×channel) and (frame×channel), respectively.

Next, the output from the stem block is processed by the first FDY–LKA block. This block is made up of FDY, LKA, BN, GLU, and an average pooling layer, as indicated in the table. The output of each FDY–LKA block is then passed to the next FDY–LKA block. Consequently, the output from the last FDY–LKA block, which is also the output of FDY–LKA-CNN, becomes a feature map with a dimension of (250×1×256).

In the fusion block, we optionally use the bidirectional encoder representation from audio transformers (BEATs) encoder [15] which is pretrained with AudioSet. The BEATs encoder ex-

Table 1. Network architecture of the proposed FDY–LKA-CRNN-based SED model, where the Fusion Block is optionally performed when BEATs embedding is applied.

Name	Layers	Output shape
Input Layer	Input: log-mel spectrogram	1001×128×1
Stem Block	3x3, Conv2D, @32 GLU, BN 2x2 average pooling layer	500×64×32
FDY–LKA Blocks	(FDY(K=4), @64, GLU, BN LKA 2x2 average pooling layer)	250×32×64
	(FDY(K=4), @128, GLU, BN LKA 1x2 average pooling layer)	250×16×128
	(FDY(K=4), @256, GLU, BN LKA 1x2 average pooling layer)	250×8×256
	(FDY(K=4), @256, GLU, BN LKA 1x2 average pooling layer)	250×4×256
	(FDY(K=4), @256, GLU, BN LKA 1x2 average pooling layer)	250×2×256
	(FDY(K=4), @256, GLU, BN LKA 1x2 average pooling layer)	250×1×256
Fusion Block (optional)	Average pooling or interpolation on BEATs embedding	250×768
	Channel-wise Concatenation (Output of FDY–LKA blocks (250×256) BEATs embedding (250×768))	250×1024
	Fully connected layer (1024×256)	250×256
RNN Block	(256 Bi-GRU cells) x 2	250×512

tracts the embedding corresponding to high-level semantic information. To align the dimensions between the output of the FDY–LKA-CNN and the BEATs embedding, we employ either average pooling or nearest neighbor interpolation. This results in four distinct models, constructed by applying one of these methods at the first or second stage for model diversity. The aligned BEATs embedding is then concatenated with the output of the FDY–LKA-CNN, followed by a fully connected (FC) layer to produce a feature map with a dimension of (250×256).

Finally, this feature map is processed by the RNN block, which comprises two bidirectional gated recurrent units (Bi-GRUs) designed to learn temporal context information. To perform SED, the output from the RNN block is processed by an FC layer and then a sigmoid function, generating an output with a dimension of (250×10), where 10 indicates the number of sound events to be detected.

The following subsection provides detailed explanations of our contributions, such as the LKA-based attention and label-filtering-based self-learning strategy, which are two key factors in achieving state-of-the-art SED performance.

3.2. LKA-based attention

The FDY, in each FDY–LKA block, is designed to capture the specific frequency characteristics associated with each event class category in the DCASE challenge. However, it is not enough to only use FDY; we also need to represent the long-term dependency of audio signals. Audio signals are inherently non-stationary, which means that we need to apply LKA-based attention, as illustrated in the FDY–LKA block in Table 1. Originally, LKA was proposed for image classification and detection tasks [7] to assign attention to a pixel by considering its adjacent pixels. In this paper, we interpret the spectrogram of an audio event sound as an image. Therefore, the attention for a specific time-frequency bin should be assigned by taking into account its adjacent time-frequency bins or bands.

The LKA attention mechanism comprises three distinct convolution layers: a depth-wise convolution layer, a depth-wise dilation convolution layer, and a (1x1) convolutional layer. The depth-wise convolution layer utilizes the local time-frequency information derived from the feature map procured by FDY. Following this, the depth-wise dilation convolution layer extracts essential long-range time-frequency band information. The final convolutional layer focuses on a channel that represents audio events as the functionality of the attention mechanism.

3.3. Label filtering-based self-learning

We propose a label filtering method to address the scarcity of strongly labeled data provided by the DCASE challenge. First, we prepare the data for label filtering, which includes 1) all the weakly labeled and unlabeled data from the DCASE dataset, and 2) a segment of AudioSet data that corresponds to one of the DCASE audio classes. We then use the first-stage SED model to infer these data and obtain the class prediction probabilities.

Next, we generate a strong pseudo-label, l_c^F, of the c-th class at the F-th frame for a given audio data using the following equation:

$$l_c^F = \begin{cases} 1, \text{ if } (p_c^F > \alpha) \text{ and } (p_c > \beta) \\ 0, \text{ otherwise,} \end{cases} \quad \text{for all } c \qquad (1)$$

where p_c^F represents the probability of the c-th class at the F-th frame of the audio signal for the strong pseudo-label, and p_c represents the probability of the c-th class for the weak pseudo-label. If p_c^F and p_c exceed the given thresholds, α and β, respectively, then the strong pseudo-label is assigned as class c. If (1) is not met, the audio data is discarded. Note that we set α and β to 0.5 and 0.7, respectively, from the exhaustive search.

After completing the label filtering process, all audio data with strong pseudo-labels are utilized as the second-stage training data. Here, the strongly labeled data from the DCASE dataset is also incorporated in the second stage.

4. PERFORMANCE EVALUATION

4.1. Model training

In the first training stage, the FDY–LKA-CRNN-based SED model parameters were initialized using the Xavier initialization [16]. The Adam optimization technique [17] was employed with a dropout rate [18] of 0.5. The learning rate was determined according to the ramp-up strategy [19], with the maximum learning rate reaching 0.001 after 50 epochs. Various augmentation techniques were applied to the training data, including time-frequency shift [20], time mask [21], mix-up [22], and filter augmentation

Table 2: Performance comparison of the baseline and different versions of the proposed SED models on the validation and evaluation dataset of the DCASE 2023 Challenge Task 4A.

Model	AudioSet	BEATs embedding	Ensemble	Validation dataset			Evaluation dataset		
				F1-score (%)	PSDS1	PSDS2	F1-score (%)	PSDS1	PSDS2
Baseline [25]	–	–	–	40.7	0.359	0.562	37.7	0.327	0.538
	√	Average pooling	–	57.6	0.491	0.787	56.7	0.510	0.798
Wenxin-TJU [26]	√	√	–	–	0.512	0.808	58.2	0.546	0.831
FDY–LKA-CRNN (Stage 1)	–	–	–	58.3	0.471	0.715	54.5	0.459	0.701
	–	Interpolation	–	63.3	0.527	0.782	–	–	–
	–	Average pooling	–	62.9	0.525	0.776	61.2	0.576	0.809
FDY–LKA-CRNN (Stage 2)	√	Interpolation	–	63.4	0.543	0.806	63.8	0.581	0.835
	√	Average pooling	–	63.8	0.546	0.808	64.6	0.591	0.831
FDY–LKA-CRNN (Stages 1 & 2)	√	Both	√	65.6	0.567	0.815	65.5	0.611	0.846

[23]. In the second stage, all training hyperparameters were set identically to those in the first stage.

4.2. Experimental results

The performance of the proposed SED model was evaluated using the measures defined in the DCASE 2023 Challenge Task 4A [24]: an event-based F1-score and PSDSs. Table 2 compares the performance between the baseline and various versions of the proposed SED models on the validation and evaluation datasets of the DCASE 2023 Challenge Task 4A. The performance on the validation dataset was drawn from the results released by DCASE 2023 Challenge Task 4A [25]. Note that there are blanks in the performance on the evaluation dataset for the first-stage SED model with interpolation since we did not submit this version to the DCASE challenge. Additionally, all the numbers in the table were averaged over three evaluations for each model, according to the DCASE challenge guideline.

We first compared the performance of our proposed model trained in the first stage with the baseline; both models were trained with the DCASE 2023 Challenge dataset without BEATs embeddings. As shown in the first and fourth rows of the table, the proposed FDY–LKA-CRNN-based SED model achieved a higher F1-score, PSDS1, and PSDS2 by 17.6%, 0.112, and 0.153, respectively, than the baseline. Upon applying BEATs embedding in the form of either interpolation or average pooling to the first-stage SED model, we observed increased F1-score, PSDS1, and PSDS2, compared to the first-stage model without BEATs embedding. The superior performance of the first-stage SED model over the baseline can be attributed to the contribution of FDY–LKA to the representation learning for this sound event detection task.

Second, we examined the effectiveness of expanding the training data from AudioSet on the SED performance. From the second and eighth rows in the table, it is clear that the addition of AudioSet data via the proposed label filtering significantly improved the SED performance. Specifically, the second-stage SED model with average pooling provided higher F1-score, PSDS1, and PSDS2 by 6.2%, 0.055, and 0.021, respectively, than the baseline with average pooling. Moreover, the second-stage SED model outperformed the first-stage SED model, indicating that label filtering is an efficient method for expanding training data.

Next, we constructed an ensemble model by combining 24 different models from each of the first- and second-stage SED models, which were taken according to different training epochs. This ensemble outperformed the baseline and individual stage models, due to inherent benefits of ensemble modeling such as reducing overfitting and improving model robustness.

Lastly, we compared our results with those of the Wenxin-TJU system [26] that was ranked the third place in the single model system track of DCASE 2023 Challenge Task 4A. As shown in the third and eighth rows of the table, the second stage of the proposed FDY–LKA-CRNN model provided higher PSDS1 for both the validation and evaluation dataset than Wenxin-TJU system, while two models had similar PSDS2.

5. CONCLUSION

We proposed an FDY–LKA-CRNN-based SED model with BEATs embedding for sound event detection. To achieve state-of-the-art performance in the DCASE 2023 Challenge Task 4A, we integrated the LKA-based attention to capture long-term dependency within the convolutional architecture. Additionally, we proposed a label filtering approach to select data from another public domain dataset—AudioSet. Accordingly, we developed a two-stage model training approach; the first-stage model was trained using DCASE 2023 Challenge data, while the second-stage model was trained using both DCASE 2023 Challenge data and selected AudioSet data. Finally, we constructed several versions of SED models based on the first- or second-stage training and their ensemble, which included models constructed by BEATs embedding using two different methods—interpolation and average pooling.

Various versions of the proposed FDY–LKA-CRNN-based SED models were evaluated on the validation dataset for DCASE 2023 Task 4A, and their performance was compared with the baseline. The results revealed that the proposed second-stage SED model, featuring LKA-based attention and label filtering-based data selection, significantly improved the SED performance compared to the baseline and the first-stage SED models. Moreover, an ensemble model consisting of the first- and second-stage models outperformed other versions of the proposed models.

6. REFERENCES

[1] K. Drossos, S. Adavanne, and T. Virtanen, “Automated audio captioning with recurrent neural networks,” in *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA),* 2017, pp. 374–378.

[2] Z. Zhao, S.-H. Zhang, Z.-Y. Xu, K. Bellisario, N.-H. Dai, H. Omrani, and B. C. Pijanowski, “Automated bird acoustic event detection and robust species classification,” *Ecological Informatics*, vol. 39, pp. 99–108, 2017.

[3] S. Grollmisch, J. Abeßer, J. Liebetrau, and H. Lukashevich, “Sounding industry: Challenges and datasets for industrial sound analysis,” in *Proc. European Signal Processing Conference (EUSIPCO)*, 2019, pp. 1–5.

[4] N. C. Phuong and T. D. Dat, “Sound classification for event detection: Application into medical telemonitoring,” in *Proc. International Conference on Computing, Management and Telecommunications (ComManTel)*, 2013, pp. 330–333.

[5] T. K. Chan and C. S. Chin, “A comprehensive review of polyphonic sound event detection,” *IEEE Access*, vol. 8, pp. 10333–103373, 2020.

[6] H. Nam, S.-H. Kim, B.-Y. Ko, and Y.-H. Park, “Frequency dynamic convolution: Frequency-adaptive pattern recognition for sound event detection,” *arXiv preprint*, arXiv:2203.15296, 2022.

[7] M.-H. Guo, C.-Z. Lu, Z.-N. Liu, M.-M. Cheng, and S.-M. Hu, “Visual attention network,” *arXiv preprint*, arXiv:2202.09741, 2022.

[8] J. F. Gemmeke, D. P. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, “Audio set: An ontology and human-labeled dataset for audio events,” in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 776–780.

[9] S. Suh and D. Y. Lee, “Data engineering for noisy student model in sound event detection,” *Tech. Rep. in DCASE 2022 Challenge,* 2022.

[10] J. Hu, L. Shen, and G. Sun, “Squeeze-and-excitation networks,” in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 7132–7141.

[11] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, “CBAM: Convolutional block attention module,” in *Proc. European Conference on Computer Vision (ECCV)*, 2018, pp. 3–9.

[12] J. Ebbers, R. Haeb-Umbach, and R. Serizel, “Threshold independent evaluation of sound event detection scores,” in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 1021–1025.

[13] J. Salamon, D. MacConnell, M. Cartwright, P. Li, and J. P. Bello, “Scaper: A library for soundscape synthesis and augmentation,” in *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2017, pp. 344–348.

[14] N. K. Kim and H. K. Kim, “Polyphonic sound event detection based on residual convolutional recurrent neural network with semi-supervised loss function,” *IEEE Access*, vol. 9, pp. 7564–7575, 2021.

[15] S. Chen, Y. Wu, C. Wang, S. Liu, D. Tompkins, Z. Chen, and F. Wei, “BEATs: Audio pre-training with acoustic tokenizers,” *arXiv preprint*, arXiv:2212.09058, 2022.

[16] X. Glorot and Y. Bengio, “Understanding the difficulty of training deep feedforward neural networks,” in *Proc. International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2010, pp. 249–256.

[17] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint*, arXiv:1412.6980, 2014.

[18] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A simple way to prevent neural networks from overfitting,” *Journal of Machine Learning Research*, vol. 15, no. 56, pp. 1929–1958, 2014.

[19] A. Tarvainen and H. Valpola, “Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results,” in *Proc. International Conference on Neural Information Processing Systems (NIPS)*, 2017, pp. 1195–1204.

[20] L. Delphin-Poulat and C. Plapous, “Mean teacher with data augmentation for DCASE 2019 Task 4,” *Tech. Rep. in DCASE 2019 Challenge*, 2019.

[21] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, “Specaugment: A simple data augmentation method for automatic speech recognition,” *arXiv preprint*, arXiv:1904.08779, 2019.

[22] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, “Mixup: Beyond empirical risk minimization,” *arXiv preprint*, arXiv:1710.09412, 2017.

[23] H. Nam, S.-H. Kim, and Y.-H. Park, “Filteraugment: An acoustic environmental data augmentation method,” in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 4308–4312.

[24] https://dcase.community/challenge2023/task-sound-event-detection-with-weak-labels-and-synthetic-soundscapes.

[25] https://dcase.community/challenge2023/task-sound-event-detection-with-weak-labels-and-synthetic-soundscapes-results.

[26] W. Duo, X. Fang, and J. Li, “Semi-supervised sound event detection system for DCASE 2023 task4A,” *Tech. Rep. in DCASE 2023 Challenge*, 2023.

IMPROVING AUTOMATED AUDIO CAPTIONING FLUENCY THROUGH DATA AUGMENTATION AND ENSEMBLE SELECTION

Jaewon Kim[*], *Yoon-Ah Park*[*], *Jae-Heung Cho*[*], *and Joon-Hyuk Chang*

Department of Electronic Engineering, Hanyang University,
Seoul, Republic of Korea

ABSTRACT

Automated audio captioning is a task of generating descriptions corresponding to audio clips. The training process of AAC typically consists of a pre-training, fine-tuning, and reinforcement learning. While reinforcement learning enhances the evaluation metrics for captions, it has the drawback of potentially lowering the quality of the captions, such as incomplete sentence or repetitive words. In this study, we propose an ensemble selection technique that combines models before and after reinforcement learning to improve evaluation metrics while maintaining caption quality. Furthermore, we apply several data augmentation techniques to complement the characteristics of WavCaps, which predominantly consists of single events, and improve generalization property. In particular, proposed approaches can reach impressive scores both an existing metric SPIDE$_r$, and a new fluency metric SPIDE$_r$-FL, 0.344 and 0.315, respectively. This resulted in a 2nd place ranking in DCASE 2023 task 6a, while the baseline system achieved SPIDE$_r$ of 0.271 and SPIDE$_r$-FL of 0.264.

Index Terms— Automated audio captioning, pre-training, data augmentaion, reinforcement learning

1. INTRODUCTION

Automated audio captioning (AAC) is an audio-to-text generation task that first introduced by K. Drossos *et al.* [1]. It is an multimodal task combines audio processing and natural language processing to describe audio clips using natural language. Unlike sound event detection [2] and audio classification tasks [3], AAC aims to capture spatio-temporal relationships in audio clips and perform advanced interpretation of audio. The detection and classification of acoustic scenes and events (DCASE) challenge has played a significant role in promoting research on AAC, particularly with the use of audio-caption pair datasets like Clotho [4] and AudioCaps [5].

During the initial development of AAC models, recurrent neural network (RNN)-based approaches [1, 6, 7] were commonly proposed. Moreover, as attention-mechanism language models [8] with superior performance emerged, transformer-based models gained significant popularity. Various transformer-based architectures, including convolution neural network (CNN)-transformer [9, 10], transformer [11], and CNN-RNN-transformer [12] with encoder-decoder structures, were widely adopted. These models establish a crucial connection between audio and transformer-based language models. CNN-based encoders have particularly demonstrated outstanding performance in audio representation as audio feature extractors. This combination of transformers and CNN-based encoders has significantly advanced the field of AAC.

In this study, we employ a bidirectional auto-regressive transformer (BART) [13] based CNN-BART model. In addition, we used data augmentation techniques such as SpecAugment [14], PairMix [15], and synonym substitution in the pre-training and fine-tuning process to enhance the generalization characteristics of the model and complement the characteristics of the dataset. SpecAugment is a widely used technique that applies random transformations to the log mel-spectrogram of the audio input, thereby enhancing robustness and generalization. PairMix is a multimodal data augmentation technique that mixes two audio clips and captions. The WavCaps [16] we used in the pre-training process mostly consisted of single event audio clips; therefore, model could not be sufficiently training about the spatial-temporal features. To address these issues, we used PairMix in the pre-training phase. Additionally, to enhance the model's universality and prevent overfitting during fine-tuning, we conducted synonym substitution, which entailed replacing random words with their synonym within the caption.

Reinforcement learning (RL) was adopted to futher enhance the model's performance. Specifically, we utilized RL based on self-critical sequence training, which has been proposed as a supplementary method to directly improve evaluation metrics. Throughout the RL process, we monitored the CIDE$_r$ [17] score, resulting in significant improvements in SPIDE$_r$. However, it is worth noting that RL models often generate captions of lower quality, such as incomplete sentences or repetitive words, as their primary objective is to improve the CIDE$_r$ score. In this study, we proposed an ensemble selection technique that can maintain the advantages of RL while enhancing caption quality. By combining models trained without RL and models trained with RL, we observed improvements in both SPIDE$_r$ and SPIDE$_r$-FL scores compared to using the pre-trained model alone. Also, the proposed method showed higher performance than the existing models in terms of SPIDE$_r$ and SPIDE$_r$-FL.

2. RELATED WORKS

AAC task employs various data augmentation techniques to enhance model performance and improve generalization capabilities. These techniques include SpecAugment, mix up, time stretching, white noise injection, and more. Among them, SpecAugment is widely used as a key data augmentation technique in AAC. It involves transforming spectrogram data in the frequency domain to increase data diversity. Frequency domain transformations can be performed in various ways, such as time masking, where a portion of the time axis is masked or duplicated, and frequency masking, where certain frequency ranges are masked. These transformations

---

[*]: Equal contributions.

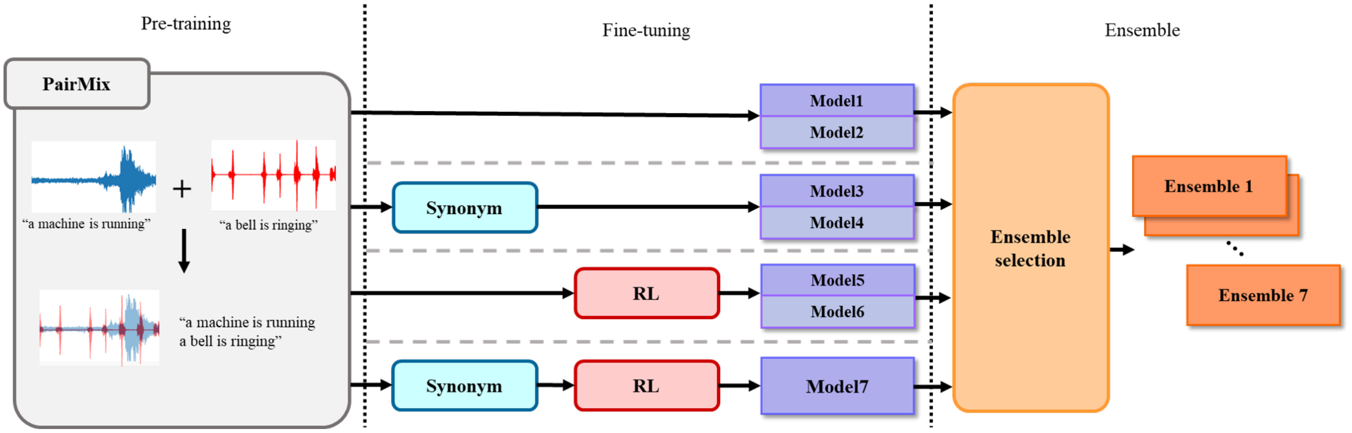


Figure 1: The flow of overall system. Synonym and RL refer synonym substitution and reinforcement learning.

allow the model to learn from a wider range of frequency patterns and enhance its robustness to noise and environmental changes, thereby improving both performance and generalization abilities. SpecAugment is recognized as a vital component of data augmentation and is widely adopted in research and applications. By utilizing this technique, AAC models can effectively operate in various environments and noise conditions. Additionally, mix up [18] is a technique where two audio samples are linearly combined to create a new sample. This allows the model to learn features from different audio sources and diversify the training data. PairMix linearly combines two captions to generate a new caption, which helps improve the model's learning by creating diverse combinations of sentence structures and content.

Captioning models are usually trained using cross-entropy loss. However, it should be noted that minimizing the loss function does not always improve the evaluation metric. To address these challenges, we employ a technique called self-critical sequence training [19]. This approach allows us to optimize the evaluation metrics directly, leading to improved scores in terms of these metrics. The model generates captions, and rewards are computed based on the metrics (such as SPICE [20], BLEU, CIDE$_r$) between the generated captions and the ground truth captions. The model is trained to maximize these rewards, aiming to generate superior captions.

3. METHODS

3.1. Data augmentation

3.1.1. PairMix

PairMix is an efficient and straightforward multimodal data augmentation technique in AAC task. It is first introduced in image captioning field named MixGen [21]. PairMix combines two audio clips and concatenates their corresponding captions. The formula for this process can be represented as follows:

$$\hat{\mathbf{a}} = \sum_{i=1}^{N} \lambda_i \mathbf{a}_i, \tag{1}$$

$$\hat{\mathbf{t}} = Concat(\mathbf{t}_{i=1}^{N}), \tag{2}$$

where $\mathbf{a}$, $\mathbf{t}$, $\hat{\mathbf{a}}$, and $\hat{\mathbf{t}}$ represent the audio waveform, caption, augmented audio, and augmented caption, respectively. $\lambda_i \in [0, 1]$ for i = 1, 2, ..., N is a hyperparameter that controls the degree of mixing.

Although data augmentation in the multimodal domain often poses challenges, PairMix provides an uncomplicated solution for audio-text datasets. By merging two audio clips, the model trained using PairMix data augmentation can extract multiple simultaneous sound events. This capability is crucial in AAC because detecting multiple sound events significantly improves the accuracy of the resulting captions. Simultaneously, the concatenation of two captions provides the model with the potential to generate more detailed and extended descriptions of audio clips. Hence, PairMix effectively enhances both audio feature extraction of detecting multiple sound events and caption quality of generating longer, specific descriptions.

3.1.2. Synonym substitution

Synonym substitution is a simple but effective data augmentation technique [22] derived from WordNet-based synonym substitution. This method entails substituting certain words in a sentence with their synonyms, thus allowing the model to express audio clips using rich vocabulary. During the fine-tuning process, we select individual words from the target captions, particularly nouns, and replaced them with their synonyms at random. This strategy can improve the generalization property of the model and the semantic properties of generated captions by ensuring that a single audio clip does not correspond to a single caption, but to various captions with the same meaning.

3.2. Audio feature extractor

In our model, we employ a 14-layer CNN derived from the pre-trained audio neural networks (PANNs) [23] architecture for the extraction of audio features. The choice of PANNs as an audio feature extractor through transfer learning is both rational and effective, given its pre-training on an audio tagging dataset. Audio tagging involves a multi-label classification task, necessitating the model to identify overlapping events occurring simultaneously within an audio clip. This requirement aligns well with the AAC task, which also needs to discern overlapping sound events. This particular CNN architecture is acknowledged for its great performance in capturing audio representations. It comprises six convolutional blocks, each containing two CNN layers with a kernel size of 3 × 3. Following each CNN layer, batch normalization [24] is used to standardize the inputs, and a rectified linear unit (ReLU) activation function [25] is incorporated to enhance performance.

3.3. Language model

We incorporated BART as our language model, motivated by its impressive track record in text generation tasks. BART comprises an encoder and a decoder, each constructed from 12 transformer layers. The BART encoder receives the audio features produced by the audio feature extractor. In contrast, the BART decoder ingests both the output of the BART encoder and the reference caption. An attention mechanism is employed between the BART encoder and decoder, facilitating the model in capturing the semantic nuances and contextual information within the input sentence. Within each transformer block of the decoder, self-attention is applied to model the interactions among all the words in the input sentence. This strategy enables the model to generate precise predictions for the subsequent word, leading to high-quality text generation. The application of self-attention aids the model in capturing long-range dependencies and complex contextual relationships between words.

3.4. Ensemble selection

When choosing models for an ensemble, the conventional approach is to select those that perform well on particular evaluation metric. In the context of AAC, one of CIDE$_r$ or SPIDE$_r$-FL is often considered when forming ensemble combinations. However, this can lead to an imbalance, where one metric's increses while the other remains unchanged or decrease. This situation is particularly evident when there's a large difference between CIDE$_r$ and SPIDE$_r$-FL scores, often occurring when the model is trained using RL. The RL method, SCST, specifically targets the CIDE$_r$ evaluation metric score. While this approach elevates the CIDE$_r$ score, it tends to lower the SPIDE$_r$-FL score. In order to simultaneously boost both scores, we strategically select the models for the ensemble. Some of these models are already trained using RL, while others are not. Given that CIDE$_r$ score can be elevated sufficiently due to RL, we exclude models achieve low scores on SPIDE$_r$-FL for attaining greater scores of it. This method aims to ensure high performance on both the CIDE$_r$ and SPIDE$_r$-FL metrics. We will describe about the metrics in subsection 4.4.

4. EXPERIMENTS

4.1. Training

Our learning process consists of three stages: pre-training, fine-tuning and ensemble selection. During the pre-training phase, we employed the WavCaps, AudioCaps, and Clotho datasets to train the model, integrating the PairMix augmentation technique. Subsequently, in the fine-tuning phase, we froze the audio feature extractor and fine-tuned the model using the Clotho dataset via various methods. Some of the experiments utilized data augmentation techniques, while others did not. Similarly, a subset of the models was fine-tuned using a RL approach, while others were not. In the final phase, we created several combinations of the outcomes from the fine-tuning step to form an ensemble. Fig. 1 shows the overview of our proposed methods.

4.2. Dataset

4.2.1. WavCaps

The WavCaps dataset[1] is a large-scale, weakly-labelled audio captioning dataset, encompassing approximately 400,000 audio clips paired with captions. This dataset is including BBC Sound Effects, FreeSound [26], SoundBible and AudioSet [27]. To reduce the challenges associated with noisy and unsuitable raw descriptions, a three-stage processing pipeline leveraging ChatGPT is employed. The average duration of the audio clips is 67.59 seconds, and captions primarily consist of single-event descriptions, with an average caption length of 7.8 tokens. However, due to the unavailability of some data from FreeSound, we focused exclusively on the publicly accessible data for our research.

4.2.2. AudioCaps

AudioCaps is a dataset composed of 46,000 audio clips, each 10 seconds in duration and paired with text descriptions. The dataset is divided into three subsets: development-training, development-validation, and development-testing, which contain 38,118, 500, and 979 audio clips, respectively. While the training set provides a single caption per audio clip, the validation and testing sets offer five captions for each clip.

4.2.3. Clotho

Clotho v2.1 is divided into three subsets within its published development sets: development-training, development-validation, and development-testing. The development-training subset comprises 3,839 audio clips, and the development-validation and development-testing subsets each consist of 1,045 audio clips. All audio files in this dataset fall within a duration of 15 to 30 seconds. For each audio clip, there are five accompanying captions, each ranging from 8 to 20 words in length.

4.3. Experiment setup

The proposed model was trained using Adam [28] optimizer with batch size of 16 in both pre-training and fine-tuning phases. In pre-training phase, the learning rate was fixed to 1×10^{-6}, and in fine-tuning phases, we used two different learning rates of 5×10^{-5} and 1×10^{-6}. We adopted PairMix technique during pre-training process and we set $\lambda = 0.5$ and $N = 2$ in Eq. (1) and Eq. (2). With regard to synonym substitution, we randomly selected 8 captions in mini batch and substituted one nouns to another similar meaning nouns. In terms of ensemble selection, we selected models as following rules. First, exclude two models attaining the lowest and second lowest scores on SPIDE$_r$ (PairMix 1, PairMix 2) and also SPIDE$_r$-FL (PairMix+RL 1, PairMix+S·S+RL), respectively. Second, exclude two models attaining the lowest scores on SPIDE$_r$ and SPIDE$_r$-FL (PairMix 1, PairMix+RL 1). Finally, exclude none of them.

4.4. Evaluation metrics

We evaluated the models trained by our methods through one machine translation metric, METEOR [29], and four captioning metrics. CIDE$_r$, SPICE, SPIDE$_r$ and SPIDE$_r$-FL are those. METEOR assesses translation quality through exact word matches,

---

[1] `https://github.com/XinhaoMei/WavCaps`

Model	METEOR	CIDEr	SPICE	SPIDEr	SPIDEr-FL
PairMix 1	*0.179*	*0.458*	*0.125*	*0.291*	0.290
PairMix 2	0.183	0.468	0.130	0.299	0.295
PairMix+S·S 1	0.182	0.473	0.129	0.301	0.298
PairMix+S·S 2	0.188	0.483	0.137	0.310	**0.306**
PairMix+RL 1	0.192	0.505	0.135	0.320	*0.154*
PairMix+RL 2	0.193	0.518	0.142	0.330	0.227
PairMix+S·S+RL	**0.195**	**0.526**	**0.143**	**0.335**	0.226

Table 1: Performances of each data augmentation techniques and RL on Clotho evaluation split. For all metrics, higher values indicate better performance. S·S refers synonym substitution. The difference between the number of models is the learning rate. Models possessing number 1 in their names are trained with learning rate of 5×10^{-5} and the others are 1×10^{-6}.

stem matches, synonym matches and phrase matches. Then it computes the harmonic mean of precision and recall according those matches. CIDE$_r$ measures weighted sum of cosine similarity between predicted and reference captions by term frequency and inverse document frequency so that it shows how created caption is well related to audio clip. SPICE metric calculates F-score using semantic scene graphs in sense of words relations in the captions. This means SPICE score can indicate model ability to generate semantically correct captions. SPIDE$_r$ is the average of CIDE$_r$ score and SPICE score, which is able to estimate the balance between two metrics. SPIDE$_r$-FL is an evaluation metrics that includes the fluency of captions. It is calculated by dividing the SPIDE$_r$ score by 10 for each individual example with an error.

5. RESULTS

The results of data augmentation techniques and RL on Clotho test set are shown in Table 1. We observed synonym substitution slightly enhances both SPIDE$_r$ and SPIDE$_r$-FL scores. Additionally, we compared the models trained with RL and those that are not. The models trained with RL were scoring higher values of SPIDE$_r$ than those models without, however, one of the captions of the highest SPIDE$_r$ score model was *'a fishing line is being wound up and a keys in'* which was not fluent enough since the sentence was not terminated. This results in the SPIDE$_r$-FL scores were significantly lower than the models not trained with RL. As a result, the model trained with PairMix and synonym substitution with learning rate 1×10^{-6} appeared the highest score of SPIDE$_r$-FL. Meanwhile, the model trained with PairMix, synonym substitution and RL was seem to be the top SPIDE$_r$ score model. In the context of ensemble selection, we analyzed the relations of ensembles with and without RL. The ensemble model excluding the RL scored similar in both SPIDE$_r$ and SPIDE$_r$-FL metrics with the top SPIDE$_r$-FL single model. However, when at least one model trained through RL was included in the ensemble, there was a notable increase in SPIDE$_r$ scores. Furthermore, for SPIDE$_r$-FL, some of these models achieved higher scores compared to ensembles without RL. Especially ensemble 3 model was achieving the highest score on SPIDE$_r$-FL metric. We also observed the caption improvement like following with the same audio clip we stated above: *'a fishing reel is being wound up and a bell is ringing'*. This caption is clearly more fluent. From Fig. 2, we noticed 4:1 ratio of non-RL models and RL models was performing the best. The model combination of each ensemble model is described below.

Model	METEOR	CIDEr	SPICE	SPIDEr	SPIDEr-FL	# of model
Ensemble 1	0.185	0.485	0.132	0.308	0.305	4
Ensemble 2	0.196	0.537	0.144	0.341	*0.256*	5
Ensemble 3	0.195	0.539	0.144	0.341	**0.332**	5
Ensemble 4	0.195	*0.529*	0.144	0.336	0.279	5
Ensemble 5	0.196	**0.543**	0.146	**0.345**	0.277	6
Ensemble 6	0.195	0.535	0.145	0.340	0.311	6
Ensemble 7	0.196	0.542	**0.147**	0.344	0.298	7

Table 2: Results of ensemble selection.

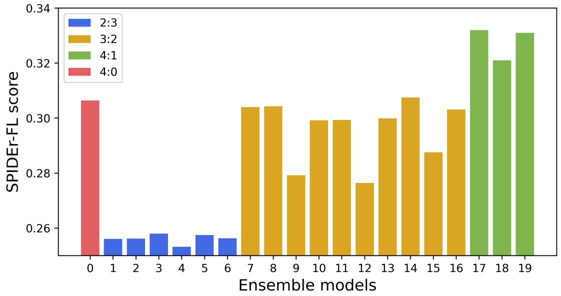


Figure 2: Ensemble SPIDE$_r$-FL scores according to ratio of non-RL and RL models. The legend shows the ratio according to colors.

- **Ensemble 1**: 4 models trained without RL.
- **Ensemble 2**: Top 5 SPIDE$_r$ models.
- **Ensemble 3**: Top 5 SPIDE$_r$-FL models.
- **Ensemble 4**: Excluding the lowest SPIDE$_r$ model and SPIDE$_r$-FL model.
- **Ensemble 5**: Top 6 SPIDE$_r$ models.
- **Ensemble 6**: Top 6 SPIDE$_r$-FL models.
- **Ensemble 7**: All 7 models.

6. CONCLUSION

In this study, we presented data augmentation, RL, and ensemble selection to boost both evaluation metrics, SPIDE$_r$ and SPIDE$_r$-FL. PairMix successfully rose the performance during the pre-training phase. This was considered the result of PairMix effect of developing the ability to detect multiple sound events at the same time stamps. Synonym substitution, likewise, elevated the model capability to express in various vocabulary. In terms of RL, it only concentrated on increasing the value of metric score, the actual fluency of captions decrease. This led to conclude removing RL models for ensemble was reasonable choice, however, those ensemble models including RL models were showing better performance when they were evaluated with both SPIDE$_r$ and SPIDE$_r$-FL. The chosen RL models played a role of regularization on ensemble, leading to generate well-related and more fluent captions.

7. ACKNOWLEDGEMENT

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.RS-2023-002255630, Development of Artificial Intelligence for Text-based 3D Movie Generation)

8. REFERENCES

[1] K. Drossos *et al.*, “Automated audio captioning with recurrent neural networks,” in *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics* (*WASPAA*), 2017, pp. 374–378.

[2] A. Mesaros, T. Heittola, T. Virtanen, and M. D. Plumbley, “Sound event detection: A tutorial,” *IEEE Signal Processing Magazine*, vol. 38, no. 5, pp. 67–83, 2021.

[3] D. Stowell *et al.*, “Detection and classification of acoustic scenes and events,” *IEEE Transactions on Multimedia*, vol. 17, no. 10, pp. 1733–1746, 2015.

[4] K. Drossos, S. Lipping, and T. Virtanen, “Clotho: An audio captioning dataset,” in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing* (*ICASSP*), 2020, pp. 736–740.

[5] C. D. Kim *et al.*, “Audiocaps: Generating captions for audios in the wild,” in *Proc. Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2019, pp. 119–132.

[6] X. Xu *et al.*, “Audio caption in a car setting with a sentence-level loss,” in *Proc. International Symposium on Chinese Spoken Language Processing* (*ISCSLP*), 2021, pp. 1–5.

[7] M. Wu *et al.*, “Audio caption: Listen and tell,” in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing* (*ICASSP*), 2019, pp. 830–834.

[8] A. Vaswani *et al.*, “Attention is all you need,” *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[9] K. Chen *et al.*, “Audio captioning based on transformer and pre-trained cnn.” in *Proc. Detection and Classification of Acoustic Scenes and Events* (*DCASE*), 2020, pp. 21–25.

[10] X. Mei *et al.*, “Diverse audio captioning via adversarial training,” in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing* (*ICASSP*), 2022, pp. 8882–8886.

[11] X. Mei *et al.*, “Audio captioning transformer,” *arXiv preprint arXiv:2107.09817*, 2021.

[12] A. Ö. Eren and S. Sert, “Audio captioning based on combined audio and semantic embeddings,” in *Proc. IEEE International Symposium on Multimedia* (*ISM*), 2020, pp. 41–48.

[13] M. Lewis *et al.*, “Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension,” *arXiv preprint arXiv:1910.13461*, 2019.

[14] D. S. Park *et al.*, “Specaugment: A simple data augmentation method for automatic speech recognition,” *arXiv preprint arXiv:1904.08779*, 2019.

[15] E. Kim *et al.*, “Improving audio-language learning with mixgen and multi-level test-time augmentation,” *arXiv preprint arXiv:2210.17143*, 2022.

[16] X. Mei *et al.*, “WavCaps: A chatgpt-assisted weakly-labelled audio captioning dataset for audio-language multimodal research,” *arXiv preprint arXiv:2303.17395*, 2023.

[17] R. Vedantam, Z. C. Lawrence, and D. Parikh, “Cider: Consensus-based image description evaluation,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR*), 2015, pp. 4566–4575.

[18] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, “mixup: Beyond empirical risk minimization,” *arXiv:1710.09412*, 2017.

[19] S. J. Rennie *et al.*, “Self-critical sequence training for image captioning,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR*), 2017, pp. 7008–7024.

[20] P. Anderson, B. Fernando, M. Johnson, and S. Gould, “Spice: Semantic propositional image caption evaluation,” in *Proc. 14th European Conference Computer Vision*, 2016, pp. 382–398.

[21] X. Hao *et al.*, “Mixgen: A new multi-modal data augmentation,” in *Proc. IEEE/CVF Winter Conference on Applications of Computer Vision*, 2023, pp. 379–389.

[22] J. Wei and K. Zou, “Eda: Easy data augmentation techniques for boosting performance on text classification tasks,” *arXiv preprint arXiv:1901.11196*, 2019.

[23] Q. Kong *et al.*, “Panns: Large-scale pretrained audio neural networks for audio pattern recognition.” *IEEE Trans. Audio, Speech, and Language Processing.*, vol. 28, pp. 2880–2894, 2020.

[24] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in *Proc. International Conference on Machine Learning*, 2015, pp. 448–456.

[25] A. F. Agarap, “Deep learning using rectified linear units (relu),” *arXiv preprint arXiv:1803.08375*, 2018.

[26] F. Font *et al.*, “Freesound technical demo,” in *Proc. International Conference on Multimedia*, 2013, pp. 411–412.

[27] J. Gemmeke *et al.*, “Audio set: An ontology and human-labeled dataset for audio events,” in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing* (*ICASSP*), 2017, pp. 776–780.

[28] Kingma *et al.*, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.

[29] A. Agarwal and A. Lavie, “METEOR: An automatic metric for mt evaluation with high levels of correlation with human judgments,” in *Proc. Workshop on Machine Translation*, 2007.

WEAKLY-SUPERVISED AUTOMATED AUDIO CAPTIONING VIA TEXT ONLY TRAINING

Theodoros Kouzelis

Institute for Language and Speech Processing
Athena Research Center
15125, Marousi, Greece
theodoros.kouzelis@athenarc.gr

Vassilis Katsouros

Institute for Language and Speech Processing
Athena Research Center
15125, Marousi, Greece
vsk@athenarc.gr

ABSTRACT

In recent years, datasets of paired audio and captions have enabled remarkable success in automatically generating descriptions for audio clips, namely Automated Audio Captioning (AAC). However, it is labor-intensive and time-consuming to collect a sufficient number of paired audio and captions. Motivated by the recent advances in Contrastive Language-Audio Pretraining (CLAP), we propose a weakly-supervised approach to train an AAC model assuming only text data and a pre-trained CLAP model, alleviating the need for paired target data. Our approach leverages the similarity between audio and text embeddings in CLAP. During training, we learn to reconstruct the text from the CLAP text embedding, and during inference, we decode using the audio embeddings. To mitigate the modality gap between the audio and text embeddings we employ strategies to bridge the gap during training and inference stages. We evaluate our proposed method on Clotho and AudioCaps datasets demonstrating its ability to achieve a relative performance of up to 83% compared to fully supervised approaches trained with paired target data. [1] Our code is available at: `https://github.com/zelaki/wsac`

***Index Terms*—** Automated audio captioning, multi-modal learning, contrastive learning.

1. INTRODUCTION

Audio-Language tasks have recently gained the attention of the audio community with the introduction of Automated Audio Captioning and Language-Based Audio Retrieval in the DCASE Challenge and the release of publicly available Audio-Language datasets such as Clotho [1] and AudioCaps [2]. The intrinsic relationship between Audio and Language presents an opportunity for the development of models that can effectively establish a shared semantic space for the two modalities. Such an approach has recently achieved great success with models like COALA [3], AudioClip [4], and CLAP [5, 6, 7]. These models use parallel audio-text data to train a joint representation, where the embeddings of audio-text pairs are similar. Such models achieve high accuracy in a zero-shot setting in a variety of tasks including Sound Event Classification, Music tasks, and Speech-related tasks [5].

Automated Audio Captioning (AAC) is a multimodal task that aims to generate textual descriptions for a given audio clip. In order to generate meaningful descriptions, a method needs to capture the sound events present in an audio clip and generate a description in natural language. Training audio captioning models requires large datasets of audio-caption pairs, and these are challenging to collect. While great effort has been done, the data scarcity issue of audio captioning still withholds. The common datasets in AAC, AudioCaps and Clotho, contain together 50k captions for training, whereas 400k captions are provided in COCO caption [8] for image captioning. Kim et al. [9] observe that due to the limited data, prior arts design decoders with shallow layers that fail to learn generalized language expressivity and are fitted to the small-scaled target dataset. Due to this issue, their performance radically decreases when tested on out-of-domain data. Motivated by these limitations we present an approach to AAC that only requires a pre-trained CLAP model and unpaired captions from a target domain. This alleviates the need for paired audio-text data, and also allows for simple and efficient domain adaptation.

Our approach is inspired by recent advances in zero-shot image captioning [10, 11], that leverage the aligned multi-modal latent space provided by CLIP [12] obviating the need for image data during training and by the recent success of Contrastive Language-Audio models such as CLAP [5] in many downstream tasks. We train a lightweight decoder model to reconstruct texts from their respective CLAP embeddings, and at inference use this decoder to decode the audio embeddings. Our findings align with prior studies in image captioning suggesting that such an approach is suboptimal due to the presence of a phenomenon known as *modality gap* [13].

The *modality gap* suggests that embeddings from different data modalities are located in two completely separate regions of the embedding space of multi-modal contrastive models [13]. To mitigate this issue we employ strategies that have been shown to effectively condense the gap in CLIP embeddings [10, 11] and show that they can be effectively utilized for CLAP models. These strategies can be divided into two categories, strategies that condense the gap during *training* and during *inference*.

Experiments on Clotho and AudioCaps datasets show that our weakly-supervised approach can achieve comparable performance to prior fully supervised arts, without requiring any target audio data during training. Our contributions can be summarized as follows: (1) We propose **WSAC: W**eakly-**S**upervised **A**udio **C**aptioning an AAC approach that requires no auditory in-domain data for training, (2) we demonstrate that the *modality gap* phenomenon is present in CLAP models, and (3) employ methods that effectively mitigate it.

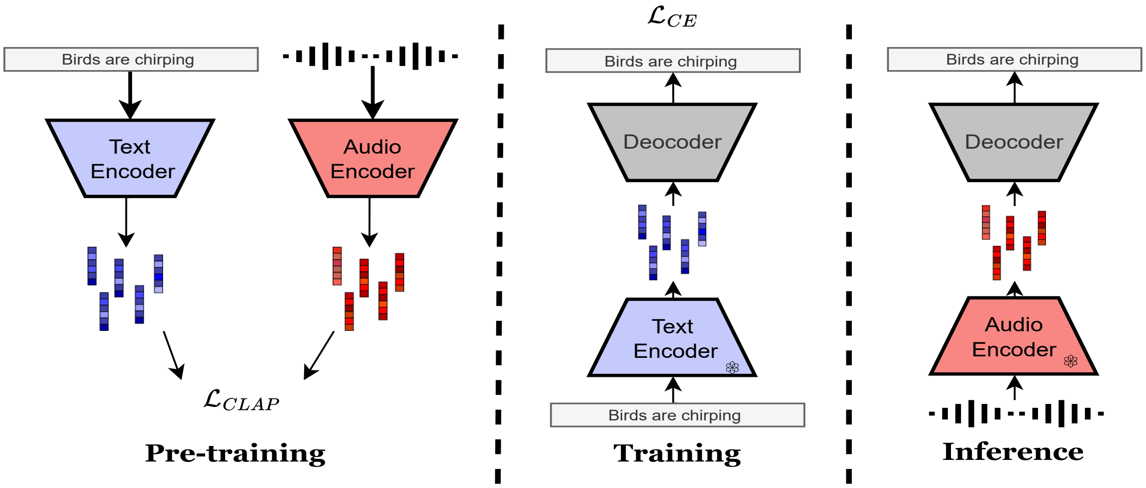
2. TEXT-ONLY TRAINING

Our goal is to learn a model that produces a caption for a given audio clip. Unlike fully supervised approaches, during training we only assume that we have access to a set of target domain captions $\mathcal{C}$. We further assume a pre-trained CLAP model with an audio en-

---

[1] This work was conducted in the framework of the PREMIERE project (No. 101061303) that is funded by the European Union.

Figure 1: Overview of our proposed approach. **Left:** An illustration of the CLAP training paradigm. The encoders are trained to map semantically similar audio-caption pairs to similar embeddings in a joint representation space. **Middle:** Our proposed weakly supervised training. A frozen CLAP text encoder embeds a caption and a decoder learns to reconstruct the caption from its embedding. **Right:** At inference, we decode the audio embedding extracted from a frozen CLAP audio encoder, using the trained decoder.



coder $\mathcal{A}_{clap}$ and a text encoder $\mathcal{T}_{clap}$ trained to project semantically similar audio-text pairs into similar embeddings in a shared embedding space as presented in Fig. 1 (Left). Given an audio clip x_a and text x_t let $\mathbf{z_a} = \mathcal{A}_{clap}(x_a) \in \mathbb{R}^d$ and $\mathbf{z_t} = \mathcal{T}_{clap}(x_t) \in \mathbb{R}^d$ be their embeddings.

First we extract text embeddings $\mathbf{z}_t$ for all $x_t \in \mathcal{C}$, keeping $\mathcal{T}_{clap}$ frozen. During training, our goal is to learn a network that inverts the CLAP text encoder $\mathcal{T}_{clap}$. We use a textual decoder D consisting of a mapping network f and an auto-regressive language model, to reconstruct the original text x_t from the CLAP text embedding $\mathbf{z_t}$. Following recent work [9], we train our decoder using the prefix language modeling paradigm. Specifically, after passing the text embedding through the mapping network f we regard $\mathbf{p} = f(\mathbf{z_t})$ as a prefix to the caption. Given a text $t = \{w_1, w_2, ..., w_T\}$, our objective is to minimize the autoregressive cross-entropy loss:

$$\mathcal{L} = -\sum_{i=1}^{T} \log D(w_i | w_{<i}, \mathbf{p}) \tag{1}$$

Since the CLAP text embedding is optimized to be similar to the CLAP audio embedding, we can directly infer the text decoder using the audio embeddings $\mathbf{z}_a$ without any pairwise training on the target dataset. The training and inference stages are presented in Fig. 1 (middle) and (right) respectively.

3. STRADEGIES TO BRIDGE THE MODALITY GAP

Directly employing the audio embeddings to infer D is not optimal due to the presence of the modality gap. Fig. 2 is a visualization of generated embeddings from the pre-trained CLAP model from the Clotho training set. Paired inputs are fed into the pre-trained model and the embeddings are visualized in 2D using T-SNE [14]. This visualization clearly demonstrates the presence of the modality gap phenomenon, as a noticeable gap separates the paired audio and text embeddings. To address this issue, we utilize strategies that have demonstrated success in bridging the modality gap in CLIP embedding space [10, 11, 13]. We show that these strategies can be adopted for CLAP and show their effectiveness in mitigating the modality gap. These approaches can be divided into two categories: Bridging the gap either during the training phase or during the inference phase.

3.1. Training strategies

Attempting to reduce the modality gap during training we adopt the following strategies: (a) Noise injection [10], and Embedding Shift [13]. These strategies aim to narrow the disparity between the modality used to train the decoder, which is text, and the target modality, which is audio.

3.1.1. Noise injection

In [10], the authors show that injecting the text embedding with Gaussian noise during training has the effect of creating a region in the embedding space that will map to the same caption. This method assumes that the corresponding audio embedding is more likely to be inside this region. Following [10], we add zero-mean Gaussian noise of standard deviation σ to the text embedding before feeding it to the decoder. We set σ to the mean L_{inf} norm of embedding differences between five captions that correspond to the same audio. Since we assume no access to target audio data we estimate σ using 50 audio-caption pairs from the WavCaps dataset [7]. Thus the prefix in Eq. 1 becomes $\mathbf{p} = f(\mathbf{z_t} + \mathbf{n})$, where $\mathbf{n} \in \mathbb{R}^d$ is a random standard Gaussian noise with standard deviation σ.

3.1.2. Embedding shift

Building upon the findings of [13], who investigated the impact of shifting embeddings in various multi-modal contrastive learning models on downstream tasks, we propose a method to align the text embeddings with the audio embeddings during training. First, we define the modality gap following [13], as the difference between the center of audio embeddings and text embeddings:

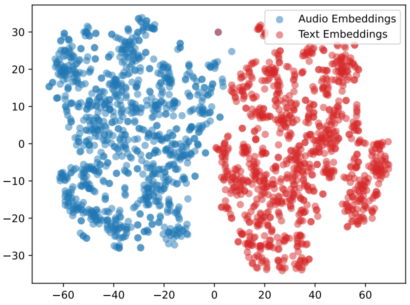


Figure 2: Visualization of audio and text embedding pairs randomly sampled from the Clotho training set. The modality gap phenomenon is present as the audio and text modalities are embedded in two completely separate regions.

$$\mathbf{\Delta_{gap}} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{z_{a_i}} - \frac{1}{n}\sum_{i=1}^{n}\mathbf{z_{t_i}} \tag{2}$$

Then, we shift every text embedding toward closing the modality gap, and thus the prefix in Eq. 1 becomes $\mathbf{p} = f(\mathbf{z_t} + \mathbf{\Delta_{gap}})$.

3.2. Inference strategies

At inference, we adopt two training-free strategies proposed in [11], and map an audio embedding extracted from the CLAP audio encoder $\mathcal{A}_{clap}$ into the text embedding space. For both strategies, we will assume a decoder D trained on some target data as described in Section 2 and a set of text embeddings obtained from the target training set that we will refer to as *Memory*, $\mathcal{M} = \{\mathbf{z_t^1}, \mathbf{z_t^2}, ...\mathbf{z_t^N}\}$, where N is the size of the training set.

3.2.1. Nearest-neighbor decoding

A straightforward strategy that can be adopted at inference time to mitigate the modality gap is to use the nearest text embedding as the prefix, instead of the audio embedding. We calculate the cosine similarity between the audio embedding $\mathbf{z_a}$ and the text embeddings in $\mathcal{M}$ and decode with the most similar:

$$\mathbf{p} = \mathbf{z_i} \;\mid\; i = \underset{\mathbf{z_t} \in \mathcal{M}}{argmax}\; sim(\mathbf{z_a}, \mathbf{z_t}) \tag{3}$$

Where $sim(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x}\cdot\mathbf{y}}{\|\mathbf{x}\|\cdot\|\mathbf{y}\|}$. Since the decoder is trained to reconstruct the original text conditioned on the text embedding, nearest-neighbor decoding can be successful if a sufficiently similar text embedding is present in $\mathcal{M}$.

3.2.2. Projection-based decoding

A better approach is to project the audio embedding into the text embedding space. This involves obtaining the representation of the audio embedding, by combining the embeddings in $\mathcal{M}$ through a weighted combination.

$$\mathbf{p} = \sum_{i=1}^{|\mathcal{M}|} w_i * \mathbf{z_{t_i}} \tag{4}$$

The weights w_i for these text embeddings are determined by calculating the cosine similarity between the audio embedding $\mathbf{z_a}$ and each embedding in $\mathcal{M}$. Following [11] the similarity is then scaled by a temperature parameter τ and normalized using a softmax function:

$$w_i = \frac{\exp(sim(\mathbf{z_a}, \mathbf{z_{t_i}})/\tau)}{\sum_{j=1}^{|\mathcal{M}|}\exp(sim(\mathbf{z_a}, \mathbf{z_{t_j}})/\tau)} \tag{5}$$

4. EXPERIMENS

4.1. Data

We conduct experiments using two benchmarks, AudioCaps and Clotho. AudioCaps contains 50k, 10-second audio clips sourced from Audioset [15]. Each audio is annotated with one caption in the training set and five captions in the evaluation set. Clotho consists of 4981 audio samples of 15 to 30 seconds duration. Each audio is annotated with five captions. We follow the standard recipes of training, validation, and test splits on each dataset for our experiments. To adhere to a weakly-supervised setting we assume no access to audio data in the training and validation sets.

4.2. Experimental setup

To extract audio and text embeddings we employ a frozen CLAP model[2] trained on WavCaps [7]. The audio encoder is a CNN14 from Pre-trained Audio Neural Networks (PANNs) [16], and the text encoder is a BERT-based model [17]. We choose this model as the embedding extractor because AudioCaps and Clotho datasets were not included in its training set. This choice is made under the assumption that target audio data are unavailable for training purposes. The decoder D consists of a mapping network f which is a 2-layered MLP, and the language model which is a 4-layer Transformer [18] with 4 attention heads. The size of the hidden state is 768. The decoder D is trained from scratch on the target captions. The noise variance for *Noise Injection* training is set to $\sigma^2 = 0.013$. We train the proposed model for 30 epochs using Adam optimizer [19] and a batch size of 64. The learning rate is linearly increased to 2×10^{-5} in the first five epochs using warm-up, which is then multiplied by 0.2 every 10 epochs. We use greedy search for decoding.

4.3. Compared methods and evaluation metrics

Since no previous work has addressed AAC in similar supervision settings we compare our methods against fully supervised approaches trained on paired data. `Koh et al.` [23] use a latent space similarity objective and train a model with a PANNs encoder and a transformer decoder. `Xu et al.` [22] design a GRU for the decoder. `Mei et al.` [20] propose a full transformer encoder-decoder architecture. `Gontier et al.` [21] utilize a pre-trained language model based on BART [21], and finetune it for AAC using guidance from Audioset tags. `Kim et al.` [9] propose prefix tuning for AAC learning a prefix to guide the caption generation of a frozen GPT-2 [24]. `Mei et al.` [7] utilize a CLAP audio encoder pre-trained on WavCaps and a BART decoder achieving state-of-the-art results in both Clotho and AudioCaps. All the methods in this work are evaluated by the metrics widely used in the captioning tasks, including BLEU [25], METEOR [26], ROUGE-L [27], CIDEr [28], SPICE [29], and SPIDEr [30].

---

[2] https://github.com/XinhaoMei/WavCaps/tree/master

Table 1: **Results on AudioCaps and Clotho**. We report results for fully supervised methods trained on audio-caption pairs, and our proposed methods trained only on captions. `WSAC` is our baseline approach presented in Section 2. We refer to *Noise injection* as `NI`, *Embedding shift* as `ES`, *Nearest-neighborhood decoding* as `NND` and, *Projection-based decoding* as `PD`. We highlight the best results for fully and weakly supervised methods with underline and **bold** respectively.

Dataset	Supervision	Method	BLEU_1	BLEU_2	BLEU_3	BLEU_4	METEOR	ROUGE_L	CIDEr	SPICE	SPIDEr
Audiocaps	Audio-Caption Pairs	`Mei et al.` [20]	0.647	0.488	0.356	0.252	0.222	0.468	0.679	0.160	0.420
		`Kim et al.` [9]	0.713	0.552	0.421	0.309	0.240	0.503	0.733	0.177	0.455
		`Gontier et al.` [21]	0.699	0.523	0.380	0.266	0.241	0.493	0.753	0.176	0.465
		`Mei et al.` [7]	0.707	-	-	0.283	0.250	0.507	0.787	0.182	0.485
	Captions Only	`WSAC`	0.574	0.398	0.267	0.167	0.222	0.426	0.493	0.155	0.324
		`WSAC+NI`	0.662	0.477	0.328	0.216	0.223	0.46	0.579	0.155	0.367
		`WSAC+ES`	0.653	0.458	0.300	0.185	0.214	0.451	0.540	0.154	0.347
		`WSAC+NND`	0.643	0.457	0.312	0.198	0.231	0.454	0.548	0.166	0.357
		`WSAC+PD`	**0.698**	**0.511**	**0.357**	**0.232**	**0.241**	**0.479**	**0.633**	**0.173**	**0.403**
Clotho	Audio-Caption Pairs	`Xu et al.` [22]	0.556	0.363	0.242	0.159	0.169	0.368	0.377	0.115	0.246
		`Koh et al.` [23]	0.551	0.369	0.252	0.168	0.165	0.373	0.380	0.111	0.246
		`Kim et al.` [9]	0.560	0.376	0.253	0.160	0.170	0.378	0.392	0.118	0.255
		`Mei et al.` [7]	0.601	-	-	0.180	0.185	0.400	0.488	0.133	0.310
	Captions Only	`WSAC`	0.462	0.282	0.173	0.102	0.166	0.343	0.265	0.113	0.189
		`WSAC+NI`	0.525	0.314	0.193	0.118	0.164	0.352	0.315	0.113	0.214
		`WSAC+ES`	**0.546**	**0.332**	**0.203**	**0.120**	0.159	0.353	0.301	0.109	0.205
		`WSAC+NND`	0.498	0.294	0.179	0.106	0.166	0.338	0.332	0.113	0.222
		`WSAC+PD`	0.532	0.324	0.200	0.118	**0.174**	**0.354**	**0.371**	**0.123**	**0.247**

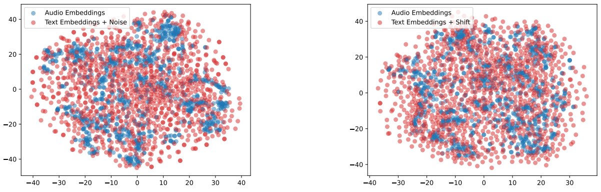
4.4. Results and Discussion

In this section, we present the results of our proposed methods on the performance metrics and compare them with fully supervised arts. Additionally, we illustrate the effectiveness of each strategy in reducing the modality gap. As shown in Table 1 our methods demonstrate comparable performance to prior state-of-the-art models despite never encountering in-domain audio data during training. We present the results of our baseline approach described in Section 2 and the results of the baseline approach in conjunction with the strategies presented in Section 3. It is evident that all the strategies boost the performance of our baseline approach in both evaluation sets. Interestingly the *inference strategies* outperform the *training strategies* in most cases. We hypothesize that this is because they utilize the *Memory* $\mathcal{M}$ which consists of in-domain text embeddings in order to bridge the modality gap. Our best-performing method, namely *Projection-based decoding* achieves 80% and 83% of the SPIDEr performance of the current fully supervised state-of-the model in Clotho and AudioCaps evaluation sets respectively. Additionally *Projection-based decoding* matches the performance of the of fully-supervised approaches proposed by Kim et al. [9]. Koh et al. [23] and Xu et al. [22] in the Clotho evaluation set.

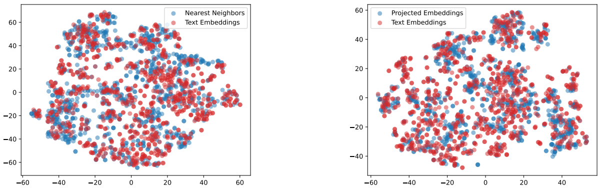
Visualization of embeddings: To further examine the effectiveness of the proposed strategies we illustrate the embeddings in 2D space using t-SNE in Fig. 3. In Fig. 3a and 3b we randomly sample audio and text embeddings from the Clotho training set after applying *Noise Injection* and *Embedding Shift* to the text embeddings. Fig. 3c and 3d illustrate randomly selected text embeddings from the Clotho evaluation set, alongside the embeddings utilized for decoding, namely the nearest neighbors and the projections, rather than the paired audio embeddings. It is evident that all strategies are effective in condensing the modality gap showcased in Fig. 2, where the audio and text modalities are embedded at arm's length in their shared representation space.

5. CONCLUSION AND FEATURE WORK

In this work, we propose a weakly-supervised approach for Automated Audio Captioning that requires a pre-trained CLAP model and only additional text data to train on a target domain. Our method



(a) Noise Injection (b) Embedding Shift



(c) Nearest-neighbor decoding (d) Projection-based decoding

Figure 3: TSN-E visualizations of the embedding space after applying the strategies presented in Section 3.

alleviates the necessity of paired data in a target domain, which are hard to collect. We demonstrate that by leveraging the shared embedding space of CLAP we can learn to reconstruct the text from the CLAP text embedding and during inference decode using the audio embeddings. We show that such an approach is suboptimal due to the presence of a modality gap and adopt strategies that effectively mitigate it. Our best-performing method achieves comparable results to prior arts trained in a fully supervised manner. For future work, we plan to study the effectiveness of our proposed approach on other tasks, such as Music Captioning and Audio Question Answering. We further aim to train a mapping network to learn the gap between the two modalities in a supervised manner.

6. REFERENCES

[1] K. Drossos, S. Lipping, and T. Virtanen, "Clotho: an audio captioning dataset," *ICASSP 2020 - 2020 IEEE Interna-*

tional Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 736–740, 2019.

[2] C. D. Kim, B. Kim, H. Lee, and G. Kim, "AudioCaps: Generating captions for audios in the wild," in *In Proc. NAACL.* Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 119–132. [Online]. Available: https://aclanthology.org/N19-1011

[3] X. Favory, K. Drossos, T. Virtanen, and X. Serra, "Coala: Co-aligned autoencoders for learning semantically enriched audio representations," *arXiv preprint arXiv:2006.08386*, 2020.

[4] A. Guzhov, F. Raue, J. Hees, and A. R. Dengel, "Audioclip: Extending clip to image, text and audio," *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 976–980, 2021.

[5] B. Elizalde, S. Deshmukh, M. A. Ismail, and H. Wang, "Clap learning audio concepts from natural language supervision," in *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023, pp. 1–5.

[6] Y. Wu, K. Chen, T. Zhang, Y. Hui, T. Berg-Kirkpatrick, and S. Dubnov, "Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation," *In Proc. ICASSP*, vol. abs/2211.06687, 2022.

[7] X. Mei, C. Meng, H. Liu, Q. Kong, T. Ko, C. Zhao, M. . Plumbley, Y. Zou, and W. Wang, "Wavcaps: A chatgpt-assisted weakly-labelled audio captioning dataset for audio-language multimodal research," *ArXiv*, vol. abs/2303.17395, 2023.

[8] X. Chen, H. Fang, T. Lin, R. Vedantam, S. Gupta, P. Dollár, and C. L. Zitnick, "Microsoft COCO captions: Data collection and evaluation server," *CoRR*, vol. abs/1504.00325, 2015. [Online]. Available: http://arxiv.org/abs/1504.00325

[9] M.-K. Kim, K. Sung-Bin, and T.-H. Oh, "Prefix tuning for automated audio captioning," *In Proc. ICASSP 2023*, vol. abs/2303.17489, 2023.

[10] D. Nukrai, R. Mokady, and A. Globerson, "Text-only training for image captioning using noise-injected clip," in *Conference on Empirical Methods in Natural Language Processing*, 2022.

[11] W. Li, L. Zhu, L. Wen, and Y. Yang, "Decap: Decoding clip latents for zero-shot captioning via text-only training," *In Proc. ICLR*, vol. abs/2303.03032, 2023.

[12] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning transferable visual models from natural language supervision," in *In Proc ICML*, 2021.

[13] W. Liang, Y. Zhang, Y. Kwon, S. Yeung, and J. Y. Zou, "Mind the gap: Understanding the modality gap in multi-modal contrastive representation learning," *ArXiv*, vol. abs/2203.02053, 2022.

[14] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 2008. [Online]. Available: http://www.jmlr.org/papers/v9/vandermaaten08a.html

[15] J. F. Gemmeke, D. P. W. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio set: An ontology and human-labeled dataset for audio events," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 776–780.

[16] Q. Kong, Y. Cao, T. Iqbal, Y. Wang, W. Wang, and M. D. Plumbley, "Panns: Large-scale pretrained audio neural networks for audio pattern recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 2880–2894, 2019.

[17] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *In Proc. ACL, Volume 1 (Long and Short Papers)*, June 2019.

[18] A. Vaswani, N. M. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *NIPS*, 2017.

[19] I. Loshchilov and F. Hutter, "Fixing weight decay regularization in adam," *ArXiv*, vol. abs/1711.05101, 2017.

[20] X. Mei, X. Liu, Q. Huang, M. D. Plumbley, and W. Wang, "Audio captioning transformer," in *DCASE Workshop*, 2021.

[21] F. Gontier, R. Serizel, and C. Cerisara, "Automated audio captioning by fine-tuning bart with audioset tags," in *DCASE Workshop*, 2021.

[22] X. Xu, H. Dinkel, M. Wu, Z. Xie, and K. Yu, "Investigating local and global information for automated audio captioning with transfer learning," *In Proc. ICASSP*, pp. 905–909, 2021.

[23] A. Koh, X. Fuzhao, and C. E. Siong, "Automated audio captioning using transfer learning and reconstruction latent space similarity regularization," in *In Proc. ICASSP.* IEEE, 2022, pp. 7722–7726.

[24] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," 2019.

[25] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *ACL.* Philadelphia, Pennsylvania, USA: Association for Computational Linguistics, July 2002, pp. 311–318. [Online]. Available: https://aclanthology.org/P02-1040

[26] S. Banerjee and A. Lavie, "METEOR: An automatic metric for MT evaluation with improved correlation with human judgments," in *ACL.* Ann Arbor, Michigan: Association for Computational Linguistics, June 2005, pp. 65–72. [Online]. Available: https://aclanthology.org/W05-0909

[27] C.-Y. Lin, "ROUGE: A package for automatic evaluation of summaries," in *Text Summarization Branches Out.* Barcelona, Spain: Association for Computational Linguistics, July 2004, pp. 74–81. [Online]. Available: https://aclanthology.org/W04-1013

[28] R. Vedantam, C. L. Zitnick, and D. Parikh, "Cider: Consensus-based image description evaluation," in *In Proc. CVPR*, 2015, pp. 4566–4575.

[29] P. Anderson, B. Fernando, M. Johnson, and S. Gould, "Spice: Semantic propositional image caption evaluation," in *In Proc ECCV.* Springer, 2016, pp. 382–398.

[30] S. Liu, Z. Zhu, N. Ye, S. Guadarrama, and K. P. Murphy, "Optimization of image description metrics using policy gradient methods," *ArXiv*, vol. abs/1612.00370, 2016.

KILLING TWO BIRDS WITH ONE STONE: CAN AN AUDIO CAPTIONING SYSTEM ALSO BE USED FOR AUDIO-TEXT RETRIEVAL?

Étienne Labbé[1], *Thomas Pellegrini*[1,2], *Julien Pinquier*[1]

[1]IRIT, Université Paul Sabatier, CNRS, Toulouse, France
[2]Artificial and Natural Intelligence Toulouse Institute (ANITI)
{etienne.labbe,thomas.pellegrini,julien.pinquier}@irit.fr

ABSTRACT

Automated Audio Captioning (AAC) aims to develop systems capable of describing an audio recording using a textual sentence. In contrast, Audio-Text Retrieval (ATR) systems seek to find the best matching audio recording(s) for a given textual query (Text-to-Audio) or vice versa (Audio-to-Text). These tasks require different types of systems: AAC employs a sequence-to-sequence model, while ATR utilizes a ranking model that compares audio and text representations within a shared projection subspace. However, this work investigates the relationship between AAC and ATR by exploring the ATR capabilities of an unmodified AAC system, without fine-tuning for the new task. Our AAC system consists of an audio encoder (ConvNeXt-Tiny) trained on AudioSet for audio tagging, and a transformer decoder responsible for generating sentences. For AAC, it achieves a high SPIDEr-FL score of 0.298 on Clotho and 0.472 on AudioCaps on average. For ATR, we propose using the standard Cross-Entropy loss values obtained for any audio/caption pair. Experimental results on the Clotho and AudioCaps datasets demonstrate decent recall values using this simple approach. For instance, we obtained a Text-to-Audio R@1 value of 0.382 for AudioCaps, which is above the current state-of-the-art method without external data. Interestingly, we observe that normalizing the loss values was necessary for Audio-to-Text retrieval.

Index Terms— Automated audio captioning, audio-text retrieval, ConvNeXt, DCASE Workshop

1. INTRODUCTION

In recent years, audio-language tasks have received greater attention due to advances in machine learning for text processing. For example, the Automated Audio Captioning (AAC) task aims to create machine learning systems that produce a sentence describing an audio file, while the Audio-Text Retrieval (ATR) task aims to use a caption to extract an audio from its database (Text-to-Audio, T2A) or use an audio to retrieve a caption from its database (Audio-to-Text, A2T). Research on these tasks is also boosted by the DCASE Challenge and Workshop[1], which proposed two tasks dedicated to AAC and T2A. Although these tasks appear to be closely related, they are usually performed by two different systems and architectures. Those systems can sometimes share common weights [1], but they need to be trained differently on several phases. In the image captioning task, the authors of [2] proposed to use a captioning system by describing each image and compare these descriptions to the captions instead of the images. In this paper, we propose another method for using an AAC system to perform the ATR task, and we investigate the implications of using this system in this way.

2. SYSTEM DESCRIPTION

2.1. AAC system architecture

To achieve the AAC task, we employ a deep neural network with an encoder-decoder architecture. We trained a ConvNeXt [3] (CNext) model for audio tagging and used it as an encoder to produce frame-level features to overcome the lack of audio-language data. The ConvNeXt was trained on the AudioSet [4] audio tagging dataset without the AudioCaps [5] audio captioning dataset files to avoid biases. This encoder achieves a high mAP score of 0.462 on AudioSet. The details of the architecture and training hyperparameters are given in [6]. The encoder gives a list of features of shape 768×31 for a 10-seconds audio clip, which are projected by a sequence of dropout set to 0.5, dense layer, a ReLU activation and another dropout set to 0.5. The decoder is a standard transformer decoder architecture [7] with six decoder layers blocks, four attention heads per block, a feedforward dimension of 2048, a GELU [8] activation function and a global dropout set to 0.2. Unlike a lot of AAC and ATR systems, no pre-trained weight has been used for the decoder/word modelling part. We found that freezing the ConvNeXt encoder leads to lower variances, so we decided to pre-compute all its embeddings to train only the decoder part. The whole model contains 28M frozen parameters and 12M trainable parameters.

2.2. Data augmentation

During our training with the decoders, we added three different augmentations on audio and input word embeddings to reduce overfitting and improve model generalization. Mixup [9] modifies the input audio and word embeddings during training, with α set to 0.4. Each embedding is mixed with another one in the current batch, except for the target, which remains unmixed. Label Smoothing [10] is applied to the target one-hot vectors to reduce the maximal probability of each word and limit the confidence of the model. Finally, SpecAugment [11] masks a part of the audio frame embeddings, with 6 stripes dropped with a maximal size of 4 in time axis and 2 stripes dropped with a maximal size of 2 in feature axis.

2.3. Using a captioning system for retrieval

The first idea to use an AAC system for ATR is to generate predictions to describe each audio file and compare each text query to each description using a metric like BLEU, CIDEr-D or SBERT, as proposed in [2], but we found low results using this strategy. We

---

[1] `https://dcase.community/`

Table 1: AAC results on Clotho and AudioCaps testing subsets. Our results are averaged over 5 seeds. WC stands for WavCaps [12] dataset. Best values for each dataset/metric are in **bold**, and best values without external data are underlined.

Dataset	System	Train data	#params	METEOR	CIDEr-D	SPICE	SPIDEr	SPIDEr-FL
CL	BEATs+Conformer [13]	CL+AC	127M	**.193**	**.506**	**.146**	**.326**	**.326**
	CNN14-trans [14]	CL	88M	.177	.441	.128	.285	N/A
	CNext-trans (ours)	CL	40M	.189	.464	.136	.300	.298
AC	HTSAT-BART [12]	AC+WC	171M	**.250**	**.787**	.182	**.485**	N/A
	Multi-TTA [15]	AC	108M	.242	.769	.181	.475	N/A
	CNext-trans (ours)	AC	40M	.246	.763	**.183**	.473	.472

believe that AAC systems tend to produce less detailed and diversified sentences than references, which leads to a loss of information when using it to summarize the audio content into a single sentence. Typically, the vocabulary size used during inference is only around 617 distinct words over the 1839 words present on average in the references for the Clotho development-testing subset. AAC systems are usually trained to predict the next token of a sentence using previous words and the audio file. This means that the model actually takes as input an audio and a caption, and the loss could be used to score this input. We decided to simply use the Cross-Entropy (CE) loss used in training to score each pair, and expecting that an AAC system should be able to give a higher loss value when the input caption does not match the input audio file than when they match. Equations 1a and 1b describe how an audio and text element are retrieved using the CE.

$$\mathrm{T2A}(t, A, f) = \operatorname*{argmin}_{a \in A} \mathrm{CE}(f(a, t_{\text{prev}}), t_{\text{next}}) \tag{1a}$$

$$\mathrm{A2T}(a, T, f) = \operatorname*{argmin}_{t \in T} \mathrm{CE}(f(a, t_{\text{prev}}), t_{\text{next}}) \tag{1b}$$

where t corresponds to a caption, T is the list of all captions, a is an audio file from the A list of audio files. f is the AAC system which produces the distributions of probabilities for the next words t_{next} given the previous words t_{prev} in the context of an audio file.

3. EXPERIMENTAL SETUP

3.1. Datasets

AudioSet [4] is the largest audio tagging dataset publicly available and contains 2M pairs of audio/tag. The audio files last for 10 seconds extracted from YouTube videos and the dataset contains 527 different sound events tags. Clotho [16] (CL) is an AAC dataset containing 6974 audio files ranging from 15 to 30 seconds in length extracted from the FreeSound website. The dataset is divided into three splits used respectively for training, validation and testing, containing five captions per audio file. In our experiments, each audio file is resampled from 44.1 kHz to 32 kHz. During training, we randomly select one of five captions for each audio file. AudioCaps [5] (AC) is the largest AAC dataset written only by humans, containing 51308 audio files from the AudioSet dataset. Since original YouTube videos are removed or unavailable for various reasons, our version of the train split contains 46230 out of 49838 files, 464 out of 495 in the validation split and 912 out of 975 files in the test split. In addition, we slightly improve caption correctness in the training subset by manually fixing 996 invalid captions with grammatical and typographic errors. For the two AAC datasets, captions are put in lowercase and all punctuation characters are removed. The codebase used to download, read and extract data is a package named aac-datasets[2].

3.2. Metrics

For the AAC task, we report the five metrics used in the DCASE Challenge task 6a. METEOR [17] is based on the precision and recall of the words. CIDEr-D [18] uses the TF-IDF scores of the shared n-grams between candidates and references. SPICE [19] builds a graph representing the scene described by the captions and computes an F-score with its common edges. SPIDEr [20] averages the two previous metrics and finally, SPIDEr-FL[3] is a combination of the SPIDEr metric with a pre-trained system designed to detect fluency errors. When one of them is detected, the SPIDEr score is divided by a factor of 10. The codebase for AAC metrics is available as a public Pip package[4] named aac-metrics. For the ATR task, we use the Recall@k metric, which measures if a relevant (ground truth) element is in the top-k retrieved elements.

3.3. Hyperparameters

The number of training epochs K is set to 400 with a batch size set to 512. The optimizer used is AdamW with an initial learning rate (lr_0) set to $5 \cdot 10^{-4}$, β_1 set to 0.9, β_2 set to 0.999, ϵ set to 10^{-8} and weight decay set to 2. Weight decay is not applied to the bias contained in the network. The learning rate is decreased during training at the end of each epoch k using a cosine scheduler rule: $\mathrm{lr}_k = \frac{1}{2}\big(1 + \cos(\frac{k\pi}{K})\big)\mathrm{lr}_0$. The gradient L_2-norm is clipped to 1 to avoid collapsing across seeds, the label smoothing reduces maximal target probability by 0.2 and the mixup α hyperparameter is set to 0.4. Since only the projection and the decoder part are trained, a single AAC experiment runs in one hour on AC and three hours on CL datasets with one V100 graphics card. To validate our model, we used the FENSE metric [27] which is based on the cosine similarity of the embeddings produced by a pre-trained Sentence-BERT model combined with the same fluency error detector used in SPIDEr-FL. During validation and inference, we used the standard beam search algorithm to generate better sentences. In order to limit the number of repetition tokens, we forced the model to avoid generating the same word twice in a single sentence, except for stop words defined in the NLTK package.

---

[2] `https://pypi.org/project/aac-datasets/0.3.3/`

[3] `https://dcase.community/challenge2023/task-automated-audio-captioning`

[4] `https://pypi.org/project/aac-metrics/0.4.2/`

Table 2: Audio-language retrieval results on Clotho and AudioCaps testing subsets. Our results are averaged over five seeds. WC stands for WavCaps dataset. Best values for each dataset/task/metric are in **bold**, and best values without external data are underlined. The asterisk * denotes the results scaled by a min-max strategy described in 4.2.

Retrieval dataset	**System**	**Training dataset(s)**	**#params**	**Text-to-audio**			**Audio-to-text**		
				R@1	**R@5**	**R@10**	**R@1**	**R@5**	**R@10**
CL	PaSST-N[4] [21]	CL+AC+WC	441M	**.261**	**.553**	**.693**	N/A	N/A	N/A
	CNN14-BERT [12]	CL+WC	214M	.215	.479	.663	**.271**	**.527**	**.663**
	CNN14-BERT [22]	CL	192M	.167	.410	.539	N/A	N/A	N/A
	Triplet-weighted [23]	CL	185M	.142	.366	.497	.169	.381	.514
	TAP+PMR [24]	CL	185M	.171	.396	N/A	.182	.399	N/A
	CNext-trans (ours)	CL	40M	.137	.349	.480	.148*	.404*	.541*
AC	HTSAT-BERT [12]	AC+WC	141M	.422	.765	.871	**.546**	**.852**	**.924**
	ONE-PEACE [25]	CL+AC+7 others	2B	**.425**	**.775**	**.884**	.510	.819	.920
	MMT [26]	AC	290M	.361	.720	.845	.396	.768	.867
	Multi-TTA [15]	AC	187M	.347	.703	.832	.402	.740	.872
	TAP+PMR [24]	AC	185M	.368	.727	N/A	.417	.762	N/A
	CNN14+TAP+PMR [24]	AC	192M	.334	.688	N/A	.431	.733	N/A
	CNext-trans (ours)	AC	40M	.382	.733	.853	.398*	.814*	.919*

4. RESULTS

4.1. AAC and ATR results

The AAC results are given in Table 1. We also reported the SOTA scores for each dataset, without reinforcement learning, without ensemble method and with or without external captioning datasets. On CL, our model performs better than the previous SOTA without external data (CNN14-trans) in all metrics and uses more than twice fewer parameters (40M instead of 88M). We believe this is mainly due to our stronger pretrained encoder, which has a higher mAP score on AudioSet and produces better features for AAC. On AC, the model reach a score very close to the Multi-TTA method, with only 0.002 absolute difference in SPIDEr despite having an unbiased encoder not trained on the testing files of AC.

Retrieval results are shown in Table 2. Just as AAC results, we reported the SOTA methods without ensemble methods and with or without external captioning datasets. Since all values are not always reported, we added several SOTA methods to compare our system with at least one other methods for each column. For the T2A task on the CL dataset, our model performs better than the DCASE baseline, but worse than most SOTA methods. However, the system achieves the highest scores on AudioCaps without external data. Somewhat surprisingly, our system outperforms other methods without external data on the A2T task on R@5 and R@10, but not on the R@1 metric on both datasets.

4.2. A2T min-max scaling

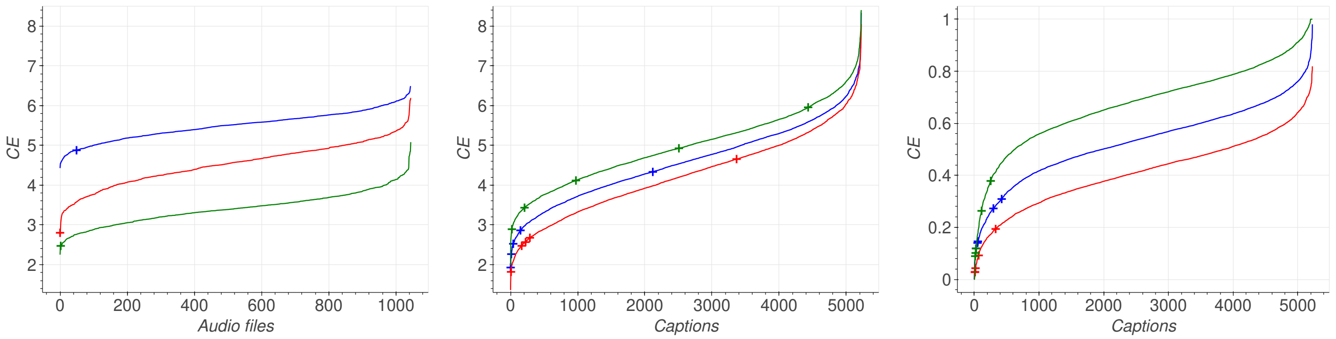
We found that even if our system performs well on T2A task, the results on A2T one were really low compared to the SOTA ones. The system reaches an R@1 of 0.146 on AC and 0.038 on CL when using raw loss values. We found that this is caused by a subset of the captions, where the loss values are almost always lower than the others for all audio files. For instance, in Figure 1a, the vertically lowest green curve corresponds to the loss of a query with all the other audio files, and is almost always lower than the other curves. In particular, only 120 unique captions are retrieved for 1045 queries during the A2T task with raw losses, but we did not find a strong correlation between these captions and the frequencies of their words or their length. In order to clarify why it impacts only the A2T task and not T2A, we provide a simple example in Table 3. This example shows the loss values for three different audio A_i with their corresponding captions C_i. When we perform the T2A task, we select the retrieved audio A_i with the lowest loss value in the column i, which achieves a perfect score in that case. However, when we perform A2T, only the caption C_1 is retrieved, because its column has a range of value different from the others, which explains the poor results when using raw loss values. To tackle this problem, we propose a post-processing which scales each "column" (i.e., each series of values corresponding to a single caption). In particular, we tried to normalize and standardize, but a simple min-max scaling has led to the best results. We also added a rule when two retrieved captions has the same score (zero when they are the minimal value of their column) by using their original losses to decide which one will be used. The impact of this scaling on the A2T losses are given in figures 1b and 1c.

Table 3: Real loss values over 3 audio files and captions.

	C_1	C_2	C_3
A_1	**1.7**	8.4	8.1
A_2	2.1	**7.6**	8.5
A_3	2.0	8.3	**6.5**

5. BENEFITS AND DOWNSIDES OF USING AAC SYSTEM

Recently, the authors of paper [28] showed that ATR systems usually fail to capture high-level relations between sounds by showing corrupted captions to an ATR system. More precisely, they propose to replace in caption the word "*after*" by "*before*" and vice versa to invert the sequence of sound events described and name this the Before-After Test (BAT). The ATR system should be able to give a lower score for an incorrect input caption than for a correct one. We believe that audio-language systems should be able to capture that kind of information better than audio event classes, but the actual metrics do not usually reflect the model performance on it. In ad-



(a) Sorted losses for 3 captions over audios.

(b) Sorted losses for 3 audios over captions.

(c) **Scaled** sorted losses for 3 audios over captions.

Figure 1: Losses for 3 queries over all retrieved items. The position of the relevant (ground truth) elements are shown with a cross.

dition to the perturbation proposed by them, we proposed to switch the relation type from sequence to superposition and vice versa by replacing some words or inverting the propositions of the sentence. For example, the sentence "*a man speaks* ***then*** *a dog barks*" can become "*a man speaks* ***as*** *a dog barks*" if we replace "*then*", or become "*a dog barks* ***then*** *a man speaks*" if we invert the propositions between "*then*". We detailed the different words tested in Table 4.

Table 4: Detailed words used for Replace. BAT stands for Before-After-Test, seq2sup for sequence-to-superposition and sup2seq for superposition-to-sequence.

Set	Words	Replaced by one of
BAT	before	after
	after	before
seq2sup	followed by, and then, then, before, after	as, while
sup2seq	as, while	followed by, and then, then, before, after

Table 5: Accuracy over different perturbations on Clotho development-testing subset. 0.5 is the score of a random model.

System	Type	Set	Accuracy
MLP [28]	Replace	BAT	.496
MLP+ACBA [28]			.554
TFMER [28]			.509
TFMER+ACBA [28]			.685
CNext-trans (ours)	Replace	BAT	.768
		seq2sup	.825
		sup2seq	.903
CNext-trans (ours)	Invert	BAT	.892
		seq	.906
		sup	.778

The Table 5 shows that our model performs very well at discriminating sound events relations, with 76.8% for the BAT, higher than the best of the compared study (68.5%). We can also see that our model performs very well on other tests which perturb the relations, with 90.6% It could imply that our model effectively captures the sequence and superposition relations. We also noticed for the Invert test with superposition words that our model is still able to detect the correct caption, probably because the first sounds described in those sentences are the loudest or longest ones in the audio.

Nevertheless, an AAC system requires computing the whole decoder pass-forward for each pair audio/caption, while usually ATR systems compute separate embeddings for each modality. For the A2T task, the post-processing is required to achieve an acceptable performance, necessitating to keep the minimal and maximal value of the loss for each caption, or an estimation of them. If a new caption is added to the database, the minimal and maximal value also need to be computed or estimated with several audio files. This scaling should also be required for zero shot experiments, which is close to the A2T task.

6. CONCLUSIONS

In this study, we propose a straightforward method for leveraging any standard AAC system for A2T. We demonstrate that despite not being specifically trained for it, an AAC system can achieve reasonable performance on both the T2A and A2T subtasks. Furthermore, it can even attain state-of-the-art scores compared to ATR methods that do not employ external data. We also observed that our model often overestimates the loss value for a subset of captions in the A2T task, resulting in poor results in the initial configuration. To address this issue, we introduced a post-processing strategy based on min-max scaling to mitigate bias in the scores. This adjustment significantly improved the results, for instance, increasing R@1 from 0.038 to 0.148 on Clotho. Finally, we evaluated our system by perturbing the input captions and found that it outperforms another ATR method in distinguishing various sound event relations. In the future, potential research directions could involve modifying AAC training using a contrastive-based loss to enhance ATR performance or developing new benchmarks and test databases to refine the evaluation of ATR systems.

7. ACKNOWLEDGMENT

This work was partially supported by the Agence Nationale de la Recherche LUDAU (Lightly-supervised and Unsupervised Discovery of Audio Units using Deep Learning) project (ANR-18-CE23-0005-01). This work was granted access to the HPC resources of IDRIS under the allocation 2022-AD011013739 made by GENCI.

8. REFERENCES

[1] X. Xu, Z. Xie, M. Wu, and K. Yu, "The SJTU system for DCASE2022 challenge task 6: Audio captioning with audio-text retrieval pre-training," DCASE2022 Challenge, Tech. Rep., July 2022.

[2] A. Krishnan, S. Rajesh, and S. SS, "Text-based image retrieval using captioning," in *2021 Fourth International Conference on Electrical, Computer and Communication Technologies (ICECCT)*, 2021, pp. 1–5.

[3] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A ConvNet for the 2020s," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 11 966–11 976.

[4] J. F. Gemmeke, D. P. W. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio Set: An ontology and human-labeled dataset for audio events," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 776–780.

[5] C. D. Kim, B. Kim, H. Lee, and G. Kim, "AudioCaps: Generating captions for audios in the wild," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 119–132.

[6] T. Pellegrini, I. Khalfaoui-Hassani, E. Labbé, and T. Masquelier, "Adapting a ConvNeXt model to audio classification on AudioSet," 2023.

[7] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is All you Need," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017.

[8] D. Hendrycks and K. Gimpel, "Gaussian Error Linear Units (GELUs)," 2016.

[9] H. Zhang, M. Cissé, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond Empirical Risk Minimization," in *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.

[10] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2818–2826.

[11] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "SpecAugment: A simple data augmentation method for automatic speech recognition," in *Interspeech 2019*. ISCA, sep 2019.

[12] X. Mei, C. Meng, H. Liu, Q. Kong, T. Ko, C. Zhao, M. D. Plumbley, Y. Zou, and W. Wang, "WavCaps: A ChatGPT-assisted weakly-labelled audio captioning dataset for audio-language multimodal research," *arXiv preprint arXiv:2303.17395*, 2023.

[13] S.-L. Wu, X. Chang, G. Wichern, J.-w. Jung, F. Germain, J. L. Roux, and S. Watanabe, "BEATs-based audio captioning model with INSTRUCTOR embedding supervision and ChatGPT mix-up," DCASE2023 Challenge, Tech. Rep., May 2023.

[14] H. Won, B. Kim, I.-Y. Kwak, and C. Lim, "CAU submission to DCASE 2021 task6: Transformer followed by transfer learning for audio captioning," DCASE2021 Challenge, Tech. Rep., July 2021.

[15] E. Kim, J. Kim, Y. Oh, K. Kim, M. Park, J. Sim, J. Lee, and K. Lee, "Exploring train and test-time augmentations for audio-language learning," 2023.

[16] K. Drossos, S. Lipping, and T. Virtanen, "Clotho: an Audio Captioning Dataset," in *2020 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2020, Barcelona, Spain, May 4-8, 2020*. IEEE, 2020, pp. 736–740.

[17] M. Denkowski and A. Lavie, "Meteor Universal: Language Specific Translation Evaluation for Any Target Language," in *Proceedings of the Ninth Workshop on Statistical Machine Translation*. Baltimore, Maryland, USA: Association for Computational Linguistics, 2014, pp. 376–380.

[18] R. Vedantam, C. L. Zitnick, and D. Parikh, "CIDEr: Consensus-based image description evaluation," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 4566–4575.

[19] P. Anderson, B. Fernando, M. Johnson, and S. Gould, "SPICE: Semantic Propositional Image Caption Evaluation," in *Computer Vision – ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham: Springer International Publishing, 2016, pp. 382–398.

[20] S. Liu, Z. Zhu, N. Ye, S. Guadarrama, and K. Murphy, "Improved Image Captioning via Policy Gradient optimization of SPIDEr," in *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*. IEEE Computer Society, 2017, pp. 873–881.

[21] P. Primus, K. Koutini, and G. Widmer, "Cp-jku's submission to task 6b of the dcase2023 challenge: Audio retrieval with passt and gpt-augmented captions," DCASE2023 Challenge, Tech. Rep., June 2023.

[22] C.-C. Wang, J. Du, and J.-S. R. Jang, "Dcase 2023 task 6b: Text-to-audio retrieval using pretrained models," DCASE2023 Challenge, Tech. Rep., June 2023.

[23] X. Mei, X. Liu, J. Sun, M. D. Plumbley, and W. Wang, "On metric learning for audio-text cross-modal retrieval," 2022.

[24] Y. Xin, D. Yang, and Y. Zou, "Improving text-audio retrieval by text-aware attention pooling and prior matrix revised loss," 2023.

[25] P. Wang, S. Wang, J. Lin, S. Bai, X. Zhou, J. Zhou, X. Wang, and C. Zhou, "One-peace: Exploring one general representation model toward unlimited modalities," 2023.

[26] A. S. Koepke, A.-M. Oncescu, J. Henriques, Z. Akata, and S. Albanie, "Audio retrieval with natural language queries: A benchmark study," *IEEE Transactions on Multimedia*, pp. 1–1, 2022.

[27] Z. Zhou, Z. Zhang, X. Xu, Z. Xie, M. Wu, and K. Q. Zhu, "Can audio captions be evaluated with image caption metrics?" in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 981–985.

[28] H.-H. Wu, O. Nieto, J. P. Bello, and J. Salamon, "Audio-text models do not yet leverage natural language," 2023.

FEW SHOT BIOACOUSTIC DETECTION BOOSTING WITH FINETUNING STRATEGY USING NEGATIVE-BASED PROTOTYPICAL LEARNING

Yuna Lee, HaeChun Chung, JaeHoon Jung

AI2XL Lab.,
Institute of Convergence Technology,
KT Corporation

ABSTRACT

Sound event detection involves the identification and temporal localization of sound events within audio recordings. Bioacoustic sound event detection specifically targets animal vocalizations, which necessitate substantial time and resources for manual annotation of temporal boundaries. This paper aims to address the challenges associated with bioacoustic sound event detection by proposing a novel prototypical learning framework. Our approach fuses contrastive learning and prototypical learning to use the limited amount of dataset at its utmost. Further, our framework leverages finetuning strategy with a novel loss function to develop a robust framework. Experimental results on a benchmark dataset demonstrate the effectiveness of our proposed method in accurately detecting and localizing bioacoustic sound events, improving the F1 score from 29.59% to 83.08%.

***Index Terms*—** Few-shot Learning, Contrastive Learning, finetuning, bioacoustic sound Event Detection

1. INTRODUCTION

Sound event detection is the task of recognizing the sound events and their respective temporal start and end times in a recording [1]. In the case of bioacoustic sound event detection, the task focuses on animal vocalizations, which demand time and resources to annotate each time stamp [2]. Few-shot learning (FSL) is a supervised learning method that can achieve high performance on data from completely different domains even with a small amount of data. As all of these tasks encounter data scarcity and the difficulty of building a framework generalized in the acoustic domain, FSL has come into the limelight. In the previous DCASE-T5 challenges, submitted systems achieved great performance by using the transductive inference method [3, 4, 5], improved prototypical learning [6], contrastive learning [7], and multi-class classification learning via splitting the audio segment into frame-level [8]. Nevertheless, proposed methods showed relatively low performance on the evaluation dataset compared to the performance obtained on the validation set. The majority of existing methods adopted prototypical learning to identify positive classes from negative classes. Prototypical learning itself demonstrated high performance, there were two limitations to taking the performance to another level. Firstly, the capability of high-level feature learning was challenging since the model was trained on classifying binary classes, which are positive and negative. Second, the loss function of current prototypical learning [9] focuses on pulling positive classes, which we refer to as “positive-based prototypical loss function (PPL)”. It may be promising on the training dataset, which contains a sufficient amount of positive class data, but it can lead to overfitting when the amount of negative class data is much greater than that of positive class. If the model is trained in the standard prototypical learning manner, the embedding features of negative classes are highly likely to be dispersed, while those of positive classes are well-clustered in the embedding space. As the class imbalance problem is prevalent in the bioacoustic domain, we propose a fine-tuning strategy with a negative-based prototypical loss function (NPL) to ameliorate this issue. The proposed method suggests additional training on negative class data to enhance the ability to aggregate negative classes in the embedding space. By applying the proposed strategy, the pretrained model can attain the superior capability to discriminate between positive and negative classes. Through this strategy, the pretrained model can achieve a higher F-measure on the validation dataset.

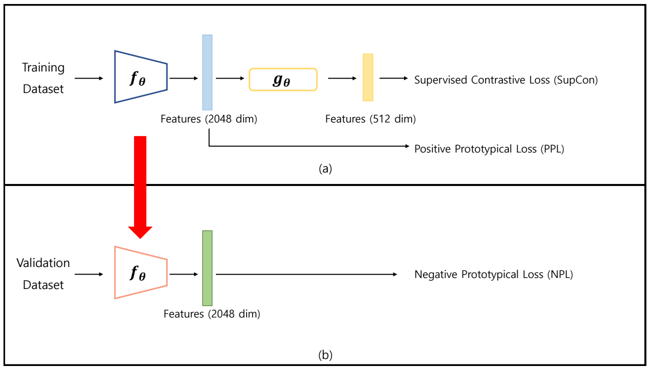


Figure 1: Overview of the proposed framework. The framework consists of a pretraining stage and finetuning stage. The pretraining stage is described in (a). The encoder f_θ is trained on the training dataset through supervised contrastive loss (SupCon) and PPL functions. Also, the finetuning stage can be seen in (b). Pretrained encoder f_θ is finetuned on the validation dataset. We exploit NPL function throughout finetuning process.

2. METHODS

2.1. Outline

Our overall framework can be shown in Figure 1. We utilize our method in N-way K-shot task. Prior to previous systems [3, 4, 5, 6, 7, 8], we denote the positive segment as the target sound event and the negative segment as the audio segments that do not contain the target sound event in each audio file. Given the fact that training dataset contains 45 classes and task 5 is regarded as 5-shot learning problem, we set $N = 45$ and $K = 5$. As each audio file in the validation dataset should be considered independently, we define negative segments from a single audio file as solitary negative classes instead of grouping negative segments into a single 'unknown' class. Simply put, each audio file contains a single positive class and a single negative class. Also, our system has 45 negative classes along with 45 positive classes. This enables encoder network $f_\theta(\cdot)$ to cluster positive segments more densely, maximizing the gap between positive segments and negative segments.

2.2. Pretraining Stage

In the pretraining stage, we train the encoder network $f_\theta(\cdot)$. We select each $2 \times K$ positive segments and negative segments from the dataset and set K segments as support segments and the other as query segments. We denote the positive support set of class i as S_i^p and the query set as Q_i^p, and the negative support set and the query set of class i can be expressed as S_i^n, Q_i^n where $|S| = |Q| = K$. The prototype of each set defined in class i, which is represented by the mean embedding vectors, is defined as the equation below.

$$s_i^* = \frac{1}{|S_i^*|}\sum_{(x_i,y_i)\in S_i^*} f_\theta(x_i), q_i^* = \frac{1}{|Q_i^*|}\sum_{(x_i,y_i)\in Q_i^*} f_\theta(x_i) \tag{1}$$

where (x_i, y_i) are the segment and its label of the class i in each set. Equation 2 describes PP_j^i, which is the euclidean distance between positive embedding vectors of Q_i^p and positive support prototype of class j, s_j^p.

$$PP_j^i = \left(\sqrt{\sum_{x\in Q_i^p} (f_\theta(x) - s_j^p)^2}\right) \tag{2}$$

In the same way, we denote PN_j^i, which is the euclidean distance between embedding vectors of Q_i^p and negative support prototype s_j^n.

$$PN_j^i = \left(\sqrt{\sum_{x\in Q_i^p} (f_\theta(x) - s_j^n)^2}\right) \tag{3}$$

We can formulate positive-based loss for class i as the equation below.

$$ppl_i = -log\left(\frac{exp\left(-PP_i^i\right)}{\sum_{j=1}^{N}\left(exp\left(-PP_j^i\right) + exp\left(-PN_j^i\right)\right)}\right) \tag{4}$$

Using equation 4, PPL is described as the equation 5.

$$PPL = \frac{1}{N}\sum_{i=1}^{N} ppl_i \tag{5}$$

We pretrain $f_\theta(\cdot)$ with PPL function and supervised contrastive (SupCon) loss function [10] to enhance the feature representation capacity of $f_\theta(\cdot)$. We build a 2-layer projection layer $g_\theta(\cdot)$ to create embedding vectors for each audio segment in the following step. Thus, our total loss function for pretraining step is $\mathcal{L}_{train} = \mathcal{L}_{PPL} + \mathcal{L}_{SupCon}$. We adopt 3-layer ResNet [11] network from previous years' method [3] as $f_\theta(\cdot)$. We set the output embedding dimension to 2048 for $\mathcal{L}_{PPL}$, and downsize the dimension to 512 for $\mathcal{L}_{SupCon}$. Through the pretraining stage, the encoder network $f_\theta(\cdot)$ can attain the ability to embed positive classes well in the embedding space.

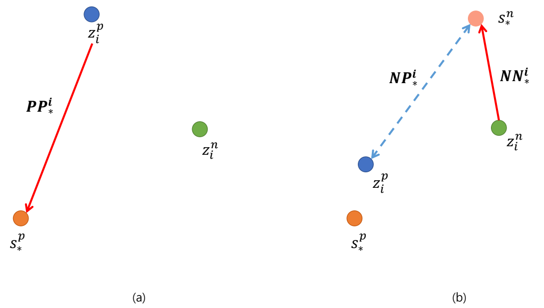


Figure 2: Let $z_i^p = f_\theta(x_i^p)$ be the positive embedding vector in the query set of class i, and $z_i^n = f_\theta(x_i^n)$ as the negative embedding vector in the following set. (a) depicts PPL function, which seeks to minimize PP_*^i. (b) describes the NPL function, minimize NN_* while maximize NP_*^i. Given that the encoder network already possesses the capability to cluster positive classes, we utilize NPL during the fine-tuning stage to increase the distance between s_*^n and s_*^p. The red line infers pull force, and the blue dotted line refers to push force.

2.3. Finetuning Stage

After the pretraining stage, $f_\theta(\cdot)$ is capable of detecting positive segments from negative segments. Nonetheless, the dataset is comprised of a large number of negative segments and a scarce amount of positive segments in the bioacoustic domain. This fact may not guarantee the good performance of $f_\theta(\cdot)$ on the general bioacoustic domain. In order to resolve data scarcity and performance maintenance issues, we figured that a sole training stage was not enough. Based on the unique characteristic of the bioacoustic dataset, we finetune $f_\theta(\cdot)$ to aim at negative-based feature learning, which is the opposite of the aforementioned stage. We display a comparison of PPL and NPL in Figure 2. Further, we propose a further developed Distance-based NPL function by incorporating the Furthest Point Sampling (FPS) algorithm [12] into the NPL function.

Negative-based Prototypical Loss We add an additional definition of distances between embedding vectors of Q_*^n and support prototypes. Unlike PPL, NPL minimizes the distance between negative embedding vectors and s^n while maximizing the distance between the positive embedding vectors. We redesign the positive-based loss ppl_i as the equation 6.

$$pnl_i = -log\left(\frac{exp\left(PN_i^i\right)}{\sum_{j=1}^{N}\left(exp\left(PP_j^i\right) + exp\left(PN_j^i\right)\right)}\right) \tag{6}$$

Following the equations 2 and 3, we define NP and NN as euclidean distance of negative query embedding vectors between the positive support prototype and negative support prototype. Equation 7 and 8 describes NP and NN specifically.

$$NP_j^i = \left(\sqrt{\sum_{x\in Q_i^n} (f_\theta(x) - s_j^p)^2}\right) \tag{7}$$

$$NN_j^i = \left(\sqrt{\sum_{x\in Q_i^n} (f_\theta(x) - s_j^n)^2}\right) \tag{8}$$

And we add new negative-based loss nnl_i to minimize the gap between negative embedding vectors and s^n. The following distance function is described below.

$$nnl_i = -log\left(\frac{exp\left(-NN_i^i\right)}{\sum_{j=1}^{N}\left(exp\left(-NP_j^i\right)+exp\left(-NN_j^i\right)\right)}\right) \quad (9)$$

With equations 6 and 9, NPL function can be summarized as equation 10.

$$NPL = \frac{1}{N}\sum_{i=1}^{N}(pnl_i + nnl_i) \quad (10)$$

By finetuning $f_\theta(\cdot)$ with $\mathcal{L}_{NPL}$, $f_\theta(\cdot)$ learns the ability to cluster negative embedding vectors and negative prototype more densely, giving the effect of separating positive segments and negative segments.

Distance-based Negative-based Prototypical Loss While NPL loss function randomly picks K support features and K query features from $2 \times K$ arbitrarily chosen features, we extend NPL loss function by adopting the idea of the Furthest Point Sampling (FPS) algorithm. FPS algorithm is a classic method used in 3D point clouds [12]. Since we aim to clump negative embedding vectors and negative prototype, we believe the distance-based selection of query and support features can maximize the efficacy of NPL loss function. All distances between $2 \times K$ randomly extracted positive features and $2 \times K$ negative features are calculated. The positive and negative features with the shortest distance are selected as a pair of reference features. Nearest-neighbor sampling method [13] is attempted based on the selected positive reference feature and negative reference feature. Thus, we set negative features placed close to the positive features as a negative support set, and positive features closely located to the negative features as a positive query set. Then, we optimize the loss function to maximize the distance between the negative prototype and positive query set so that we can ultimately maximize PN. We conduct the furthest sampling based on the prior negative reference feature in negative features. Through this process, negative features located on the outskirts will be selected from negative features, and non-selected features will be located on the inner side among negative features. We set the selected features to a negative query set and the unselected features to a negative support set. The negative prototype created from negative support set is used to minimize the distance between negative query set, eventually minimizing NN. In this way, we can boost the initial goal of NPL by optimizing the maximization of positive-negative distance and minimization of negative-to-negative distance at the same time. The following procedures are illustrated in Figure 3. For post-processing and inference, we applied methods proposed in the DCASE 2022 challenge [3].

3. EXPERIMENT

3.1. Experimental setup

We conducted the experiments for two purposes. First, we prove that our novel framework is more applicable in the few-shot learning domain than baseline methods. In the previous DCASE challenges, transductive inference (TI) method adapted from [14] played a crucial role in challenge [15, 16, 17, 18]. Here, we apply part of the TI method as a variant to our scheme. Thus, we compare variants with our method to analyze the impact of our novel finetuning strategies as an ablation study. Second, we intend to prove the efficacy of our proposed method by comparing the results of grafting the finetuning strategy. We set contrastive learning and few-shot learning as our basic framework. In all experiments, the learning rate was set to 0.001 and the input length was fixed at 0.2 seconds. To prevent overfitting on any dataset, we implemented early stopping. We did not use any augmentation or additional acoustic features. We adopted the official evaluation metric[1] as our evaluation metric. Since the full annotation of the evaluation set was not released in public, we considered the validation set of the DCASE 2023 task 5 dataset as the evaluation set.

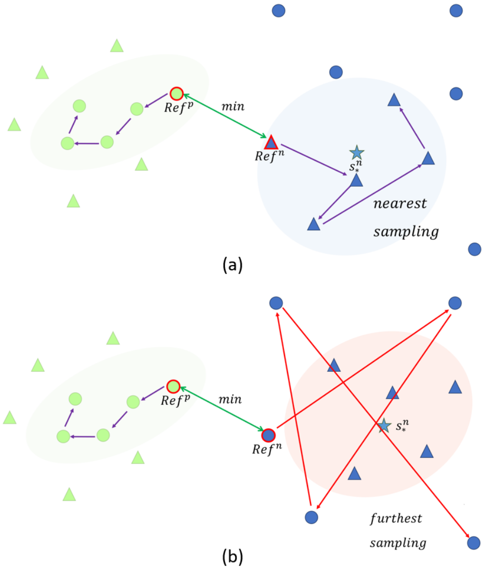


Figure 3: We denote each positive and negative reference feature as Ref^p and Ref^n. The triangle, circle, and star-shaped figure each represent the feature vectors of the support set, the query set, and the prototype respectively. (a) shows the process of maximizing PN based on Ref^* through nearest-neighbor sampling. (b) is the process of minimizing the NN via the furthest point sampling.

		Precision (%)	Recall (%)	F-measure (%)
Template Matching		2.42	18.32	4.28
Prototypical Network		36.34	24.96	29.59
DCASE2022 Winning Team [8]		77.50	71.50	74.40
Ours	Pretraining	74.27	56.70	64.31
	Finetuning	89.93	77.20	**83.08**

Table 1: The precision, recall, and F-measure of the validation set.

4. RESULTS

4.1. Performance Comparison

We compare our methods with baseline schemes and the winning team of DCASE 2022 [8]. we describe our basic framework as the "backbone" for convenience. Pretraining denotes the performance of the encoder $f_\theta(\cdot)$ after the pretraining stage, and Finetuning denotes the performance after the finetuning stage. As can

---

[1] https://github.com/c4dm/dcase-few-shot-bioacoustic

System	PB	ME	HB	Overall		
	F-measure (%)			Pre (%)	Rec (%)	F-measure (%)
Backbone	45.27	74.58	80.66	66.39	59.28	62.64
w. TI method	47.18	85.71	72.50	74.27	56.70	64.31
w. Distance-based NPL finetuning	63.04	95.41	95.60	90.17	74.38	81.52
w. TI method & Distance-based NPL finetuning	63.45	99.05	97.53	89.93	77.20	**83.08**

Table 2: The precision, recall, and f-measure of each subset in the validation set.

be seen in Table 1, our proposed method outnumbers both baseline and 2022 challenge-winning team by a large margin. We also evaluated our encoder network $f_\theta(\cdot)$ after each stage to confirm the impact of the distance-based NPL function. The performance disparity between the two stages clearly verifies distance-based NPL function actually have a meaningful impact on developing the capacity to detect positive sound event even in the highly imbalanced dataset, increasing the performance up to 18.77%. In Table 2, we compare our basic scheme and its variants. We select the condition where the distance-based NPL finetuning strategy and TI method are additionally applied to our backbone for comparison. We select systems with different conditions as mentioned in section 3.1. Our system showed relatively low performance on the PB dataset relative to other datasets in general. We assume this phenomenon is due to the drastic ratio between the positive segment and the negative segment as it contains a relatively short duration of the positive segment. Since the features extracted from positive segments are limited, the encoder network $f_\theta(\cdot)$ finds it more difficult to detect positive segments. This phenomenon was consistently observed in the performance of the DCASE2023 evaluation dataset [19]. All of our submitted systems showed relatively low performance on the CT dataset, in which the majority of positive segments are less than 0.2 seconds, which is the minimum input length of our method.

4.2. Ablation Study

In the ablation study, we compare our baseline scheme and the combination of two different novel finetuning strategies. We compared the case where only the basic training stage was performed for each baseline and the case where original NPL finetuning and distance-based NPL finetuning was applied.

System	F-measure
Backbone	62.64
w. NPL finetuning	79.79
w. Distance-based NPL finetuning	**81.52**

Table 3: Ablation study of the proposed method.

Table 3 states that finetuning strategy with the NPL function and the distance-based NPL function shows a noticeable numerical difference. We presume the following difference is based on the prototype selection. While typical NPL selects support features and query features randomly, distance-based NPL is based on euclidean distance, which is more definite. This induces the network to finetune in a way that estimates the position of the positive prototype and escalates PN, increasing performance more intuitively. The effect of distance-based NPL finetuning is visualized with t-SNE [20] in Figure 4. As shown in figure 4, t-SNE of the same class tend to cluster more densely after distance-based NPL is exploited. The fact that the distance-based NPL performed better than the conventional NPL was also confirmed in the performance of the evaluation set. It was confirmed that the systems finetuned with distance-based NPL performed better than the systems finetuned with typical NPL. The performance gap was more prominent in the case where the positive class and the negative class were very similar, such as MGE dataset in the DCASE 2023 evaluation set [19].

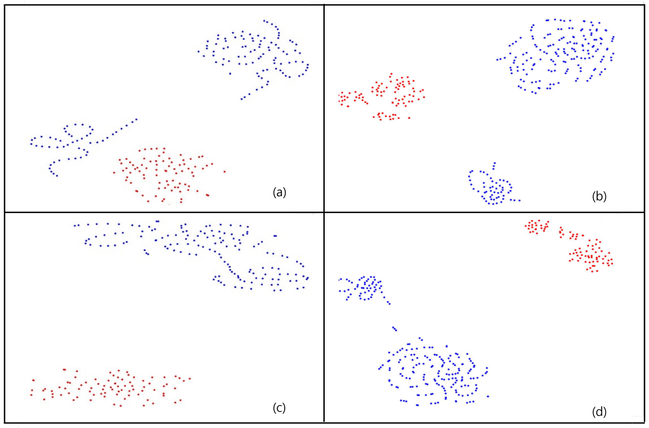


Figure 4: (a) and (c) are extracted from the same audio, and (b) and (d) are extracted from the same audio file. (a), (b) are visualizations extracted after pretraining stage. and (c), and (d) are extracted after finetuning stage. Red dot represents embedding vectors extracted from positive segments, and blue dot represents vectors extracted from negative segments.

5. CONCLUSION

In this paper, we presented a novel framework for few-shot bioacoustic event detection. Our method combines the contrastive learning method and prototypical learning and uses the novel finetuning strategy of using a modified prototypical loss function. The proposed pretraining process enables embedding positive class data on the embedding space, NPL finetuning strategy enables pretrained network to detect sound events in the environment where positive sound events were unseen in the training stage or fine-tuning stage. Experiments showed that the proposed framework can robustly separate positive and negative segments in highly imbalanced datasets. Further, the fact that all of the submitted systems achieve high F-measure scores on two new subsets proves its ability to generalize to new classes [19].

6. REFERENCES

[1] A. Mesaros, T. Heittola, T. Virtanen, and M. D. Plumbley, “Sound event detection: A tutorial,” *IEEE Signal Processing Magazine*, vol. 38, no. 5, pp. 67–83, 2021.

[2] http://dcase.community/challenge2023/.

[3] H. Liu, X. Liu, X. Mei, Q. Kong, W. Wang, and M. D. Plumbley, “Surrey system for dcase 2022 task 5 : Few-shot bioacoustic event detection with segment-level metric learning technical report,” DCASE2022 Challenge, Tech. Rep., June 2022.

[4] Y. Tan, L. Xu, C. Zhu, S. Li, H. Ai, and X. Shao, “A new transductive framework for few-shot bioacoustic event detection task,” June 2022.

[5] Q. Huang, Y. Li, W. Cao, and H. Chen, “Few-shot bio-acoustic event detection based on transductive learning and adapted central difference convolution,” June 2022.

[6] D. Yang, Y. Zou, F. Cui, and Y. Wang, “Improved prototypical network with data augmentation,” June 2022.

[7] B. Zgorzynski and M. Matuszewski, “Siamese network for few-shot bioacoustic event detection,” June 2022.

[8] J. Tang, X. Zhang, T. Gao, D. Liu, J. P. Xin Fang and, Q. Wang, J. Du, K. Xu, and Q. Pan, “Few-shot embedding learning and event filtering for bioacoustic event detection,” June 2022.

[9] J. Snell, K. Swersky, and R. Zemel, “Prototypical networks for few-shot learning,” *Advances in neural information processing systems*, vol. 30, 2017.

[10] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan, “Supervised contrastive learning,” *Advances in neural information processing systems*, vol. 33, pp. 18 661–18 673, 2020.

[11] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[12] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, “Pointnet++: Deep hierarchical feature learning on point sets in a metric space,” in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2017/file/d8bf84be3800d12f74d8b05e9b89836f-Paper.pdf

[13] B. W. Silverman and M. C. Jones, “E. fix and jl hodges (1951): An important contribution to nonparametric discriminant analysis and density estimation: Commentary on fix and hodges (1951),” *International Statistical Review/Revue Internationale de Statistique*, pp. 233–238, 1989.

[14] M. Boudiaf, I. Ziko, J. Rony, J. Dolz, P. Piantanida, and I. Ben Ayed, “Information maximization for few-shot learning,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 2445–2457, 2020.

[15] D. Yang, H. Wang, Y. Zou, Z. Ye, and W. Wang, “A mutual learning framework for few-shot sound event detection,” in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 811–815.

[16] M. Boudiaf, H. Kervadec, Z. I. Masud, P. Piantanida, I. Ben Ayed, and J. Dolz, “Few-shot segmentation without meta-learning: A good transductive inference is all you need?” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 13 979–13 988.

[17] Y. Liu, J. Lee, M. Park, S. Kim, E. Yang, S. J. Hwang, and Y. Yang, “Learning to propagate labels: Transductive propagation network for few-shot learning,” *arXiv preprint arXiv:1805.10002*, 2018.

[18] D. Yang, H. Wang, Z. Ye, and Y. Zou, “Few-shot bioacoustic event detection= a good transductive inference is all you need,” DCASE2021 Challenge, Tech. Rep, Tech. Rep., 2021.

[19] I. Nolasco, S. Singh, E. Vidana-Villa, E. Grout, J. Morford, M. Emmerson, F. Jensens, H. Whitehead, I. Kiskin, A. Strandburg-Peshkin, *et al.*, “Few-shot bioacoustic event detection at the dcase 2022 challenge,” *arXiv preprint arXiv:2207.07911*, 2022.

[20] L. Van der Maaten and G. Hinton, “Visualizing data using t-sne.” *Journal of machine learning research*, vol. 9, no. 11, 2008.

MASKED MODELING DUO VISION TRANSFORMER WITH MULTI-LAYER FEATURE FUSION ON RESPIRATORY SOUND CLASSIFICATION

Boxin Liu[1], *Shiqi Zhang*[1], *Daiki Takeuchi*[2], *Daisuke Niizumi*[2], *Noboru Harada*[2], *Shoji Makino*[1]

[1]Waseda University, Japan [2] NTT Corporation, Japan

ABSTRACT

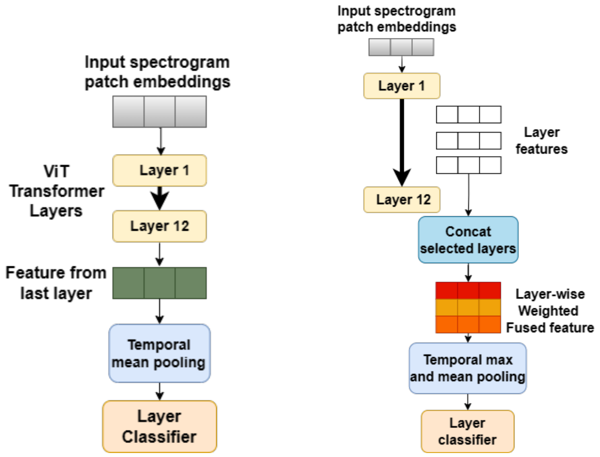
Respiratory sounds are significant relevant indicators for respiratory health and conditions. Classifying the respiratory sounds of patients can assist doctors' diagnosis of lung diseases. For this purpose, many deep learning-based automatic analysis methods have been developed. However, it is still challenging due to the limited medical sound datasets. In this study, we apply a pre-trained Vision Transformer (ViT) based model from the Masked Modeling Duo (M2D) framework for this task. While the M2D ViT pre-trained model provides effective features, we think combining features from different layers can improve the performance in this task. We propose a multi-layer feature fusion method using learnable layer-wise weights and validate its effectiveness in experiments and an analysis of pre-trained model layers. Our approach achieves the best ICBHI score of 60.68, 2.39 higher than the previous state-of-the-art method.

Index Terms— Respiratory Sound Classification, ICBHI, Pre-trained Model, Feature Fusion, Masked Modeling Duo

1. INTRODUCTION

Respiratory diseases have recently become the third cause of death worldwide [1]. And due to the impact of the COVID-19 global pandemic, the need for diagnosing lung disease with efficient methods with accuracy and lower work burden for physicians and medical experts has been increasing. Respiratory sound classification is a task to identify whether a breathing cycle of a recorded sound sample contains adventitious sounds related to potential disease in the respiratory system. Conventional respiratory sound classification requires medical experts to utilize stethoscopes to conduct auscultations for patients in person, which is highly demanding for hospitals and other medical institutions [2].

International Conference on Biomedical Health Informatics (ICBHI) Respiratory Sound Database [3] is a public database for developing the algorithms on respiratory classification tasks recorded by microphones and electronic stethoscopes. The audio samples in this dataset consist of respiratory cycles in variant lengths with four kinds of annotations: normal, crackles, wheeze, and the combination of both anomalies. Crackles are discontinuous adventitious sounds in breathing cycles and can be an early sign of cardiorespiratory conditions. At the same time, wheezes are continuous and musical sounds of anomaly, indicating the patient's obstructive airway conditions. The classification for these types of breath sounds can be the basis for diagnosing or monitoring diseases such as asthma, Chronic Obstructive Pulmonary Disease (COPD) [4], and pneumonia. With the release of this dataset, more and more research attention has been drawn to the respiratory sound classification task and further the automatic assistance for doctors' diagnoses.



(a) Conventional (b) Proposed

Figure 1: Overview of the conventional method and the multi-layer feature fusion workflow.

The sounds in the ICBHI dataset were recorded from various positions and by many types of equipment, making distinguishing different respiratory cycles difficult. Besides, the dataset scale is limited. Until now, several previous studies have proposed models to tackle this task, and many novel structures or algorithms and data augmentations have been introduced [5–11]. With the addition of the limited dataset, pre-training visual models with large-scale datasets have been widely used in ICBHI task [7–11].

The models from the self-supervised learning framework pre-training on large-scale audio datasets have recently achieved competitive performance in the image field and several audio tasks [12–15]. In this study, we used pre-trained ViT [16] models from the Masked Modeling Duo (M2D) framework [15] (M2DViT). We adopted the M2DViT without changing the backbone configurations, such as patch size and grid size.

While the features from the M2DViT can perform well in various tasks, we expect to achieve even better performance by combining features from different layers to form effective representations of audio samples. In this study, we explore the possibility of the feature fusion available from the M2DViT layers for solving the ICBHI task and propose methods for fusing effective features. We experiment with our methods on the ICBHI task and validate the effectiveness. In addition, we analyze the contribution of layer features and show that the later layers contribute more.

In summary, the main contributions of this paper are as follows:

- Proposing to compose effective representations for the respiratory sound classification task using M2D layer features.

- Conducting experiments with our multi-layer feature fusion methods and comparing ours with the previous methods.
- Analyzing the performance of different layers of M2DViT and their combinations in the respiratory sound classification task.

2. RELATED WORKS

2.1. Respiratory Sound Classification

Ever since the ICBHI2017 challenge and the release of the open access dataset, researchers have trained and evaluated many deep learning-based respiratory classifying methods to have better solutions for this task. Early models like LungRN+NL [5] combine ResNet-based architecture and mix-up augmentation method, and then the attention mechanism was introduced with LungAttn [6]. The works after 2020 are widely presented with ImageNet [17] or AudioSet [18] pre-training. And for RespireNet [8], the authors also use a device-specified fine-tuning strategy to improve the performance. The previous works are mainly based on ResNet structure except for a recent work [11], which uses a simple CNN backbone from PANNs [19] and contrastive learning with metadata strategy. The self-supervised methods, such as contrastive learning, show their validity in [11]. A concurrent work based on Audio Spectrogram Transformer (AST) [20] and contrastive learning with Patch-Mix augmentation [21] shows that the pre-trained attention-based model has the potential for better performance than other conventional models.

2.2. Masked Modeling Duo

The adopted self-supervised learning framework of M2D [15] is an effective method for general-purpose pre-training using a masked prediction task. This method was originally inspired by the Masked Autoencoder (MAE) [22] approach utilized in Masked Image Modeling (MIM), along with the Bootstrap Your Own Latent (BYOL) [23] framework, which enables the direct acquisition of latent representations through a target network.

In the two divided networks of M2D, the framework learns to predict the output of the target network with the output of the online network. At the same time, visible patches serve as input for the online network, and masked patches for the target network. While the online network weights are optimized to minimize the loss, the weights of the encoder in target network ξ are updated based on the exponential moving average (EMA) of the online network θ with a decay rate τ.

M2D learns effective representations in the online encoder. After training, only the trained parameter of the online encoder f_θ is transferred as a pre-trained ViT model, which we call M2DViT, for downstream tasks. The M2DViT pre-trained weights are available online[1] and used in our experiments, which are pre-trained on AudioSet [18]. Unlike previous works, we combine multi-layer feature outputs.

2.3. Feature Fusion

The method of Feature Fusion was broadly proposed to deal with multi-modal tasks [24, 25]. There are numerous approaches to extracting features from different levels of deep learning models. The skip connection structure and multi-scale attention mechanism have been widely used. For example, in the work, MS-CAM [26], an iterative attentional feature fusion method performs excellently in vision models. And for another instance, in the work of MFVT [27], the authors proved that the fused features in the ViT-based model are a potent strategy in the fine-grained visual categorization task. Besides, in audio-related tasks, the multi-layer feature fusion serves as a powerful method, as reported in [28]. The mechanism of multi-layer feature fusion is similar to the skip connection essential for convolutional networks such as ResNet [29] and DenseNet [30], and various methods for connecting layers are proposed. The skip connections encourage the networks to obtain semantic features from the early layers of the model [31]. The fusion is usually performed by operations of addition or concatenation with a fixed weight of the features [29, 30].

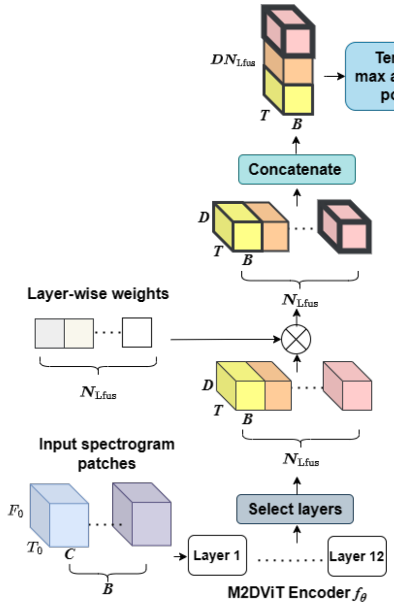


Figure 2: The multi-layer feature fusion calculation flow. F_0, T_0, and C are frequency bins, time frames, and channels in a spectrogram, respectively.

3. METHODOLOGY

The encoder in M2DViT is based on the ViT backbone, consisting of 12 transformer blocks as layers. The ViT first patchifies the input mel-spectrogram and then processes it with a projection of a linear layer, transferring the spectrogram into patch embeddings. Then the fixed sinusoidal positional embedding is added to the input. The multi-head attention is applied , followed by the MLP containing 2 linear layers with a Gaussian error linear unit (GELU) activation. We denote the transformer blocks as transformer layers for simplicity. The outputs from all transformer layers have the same shape, and all the layer outputs are available for later use, such as classification.

The conventional M2DViT, shown in Fig. 1(a), takes a spectrogram input, processes the input in the transformer layers, then outputs the last layer feature $z \in R^{B \times T \times D}$, where B is the input batch size, T is the length of the sequence composed by encoded spectrotemporal patches, and D is the embedded patch feature dimension. Then, only the output z is used afterward.

---

[1] `https://github.com/nttcslab/m2d`

One drawback of the conventional method is that the information from different layers is not used as the representation for audio [32]. The performance of the features from pre-trained ViT layers can be imbalanced due to the structure and training metrics [28]. To address this problem, we introduce multi-layer feature fusion methods to combine the layer features. We also use learnable layer-wise weights to balance layers with better performance automatically, which is optimized as the training epoch proceeds. Our approach enables arbitrary combinations of the layer features as effective representations for later use.

The pipeline for multi-layer feature fusion is shown in Fig. 1(b). And Fig. 2 shows the details of the feature calculation flow. The output features of all the layers can be defined as $\{z_i \in R^{B\times T\times D} | i \in L\}$, where z_i is the i-th layer output and L is the number of layers.

Then, we calculate the multi-layer feature fusion $\tilde{z}$ as follows:

$$\tilde{z} = \text{concat}(\{w_i z_i | i \in L_{\text{fus}}\}) \tag{1}$$

where the concat is a function that concatenates features on the dimension of D, L_{fus} is the set of layer indexes of desired fusion, and w_i is a learnable layer-wise weight of the i-th layer in the fused feature. Note that the $\tilde{z}$ forms the shape of $R^{B\times T\times DN_{\text{Lfus}}}$, where N_{Lfus} is the number of the fused layers. As a result, the multi-layer feature fusion enables us to utilize useful information in the features from all the desired layers.

As a final operation, we apply temporal poolings to summarize time-framed features in a feature vector:

$$z' = \text{mean}(\tilde{z}) + \max(\tilde{z}) \tag{2}$$

where $z' \in R^{B\times DN_{\text{Lfus}}}$ is the final fused feature vector used as the input for later use (e.g., classification) and mean/max are temporal operations each. We follow [28] for the effective temporal pooling operation.

4. EXPERIMENTS

We conducted experiments to validate our proposals. The following sections explain the dataset (Section 4.1), evaluation metrics (Section 4.2), and experimental setup (Section 4.3). Then we show experimental results with vanilla M2DViT (Section 4.4) and results with our proposals as well as previous studies (Section 4.5).

4.1. Dataset

ICBHI Respiratory Sound Database [3] consists of 920 annotated respiratory audio samples recorded from 126 patient subjects of the labs and hospitals in Portugal and Greece. The samples are officially split into a train set (539 samples, 60%) and a test set (381 samples, 40%). The database contains two sets of annotations. One is for whether a cycle contains crackles, wheezes, or a combination of both, and some with no adventitious respiratory sounds. The other is the annotation of the locations of the adventitious respiratory sounds. In the 6898 respiratory cycles, whose lengths vary from 0.2s to 16.2s, 1864 contain crackles, 886 contain wheezes, and 506 contain both crackles and wheezes. The others are normal ones. The chest locations from which the recordings were acquired are also provided. Noise levels in some respiration cycles are high, which simulates real-life conditions. The recordings were collected using heterogeneous equipment, and their duration ranged from 10s to 90s. The average time duration of the cycles is 2.7s, and the total length is 5.5h.

Method	Masking ratio r	S_p	S_e	S_c
M2DViT	0.6	71.59±2.64	43.25±1.60	57.42±0.79
	0.7	75.78±5.84	39.51±4.64	57.64±0.86

Table 1: The ICBHI performance comparison between different pre-training masking ratios of M2DViT.

4.2. Evaluation Metrics

The evaluation metrics in our experiments are adopted from the original ICBHI2017 challenge, which is common in the previous papers. There are three scores, sensitivity S_e, specificity S_p, and the average of these two metrics ICBHI score S_c. They are calculated as the following formulas:

$$S_e = \frac{P_c + P_w + P_b}{T_c + T_w + T_b} \tag{3}$$

$$S_p = \frac{P_n}{T_n} \tag{4}$$

$$S_c = \frac{S_e + S_p}{2} \tag{5}$$

where P_c, P_w, P_b, and P_n are numbers of right prediction for the cycles containing crackles, wheezes, both of the two adventitious sounds and none of them. While T_c, T_w, T_b, and T_n are the total numbers of four categories respectively.

4.3. Experimental Setup

We used an Adam optimizer with a learning rate of 1e-4 and weight decay of 1e-4, cosine scheduled in the M2DViT model. The batch size is set as 64. The classifier used in our study is a 4-class linear classifier. The input spectrograms are patchified with a patch size of (16, 16), and the grid size is (5, 38). The number of encoder embedding dimensions is 768. The ViT-base pre-trained model consists of 12 transformer blocks with the same number of attention heads [16]. We fine-tuned all the pre-trained ViT model weights in 150 epochs and used weighted cross-entropy as our training and evaluating loss.

We used the same ViT and training settings described above for all the setups. For the adaptive weight for all layers, we initialized them as all layers weighted the same value of 1.0 and updated them after training every epoch. All our experiments run five times with random seeds, and we provide statistics of results.

4.3.1. Preprossessing and Data Augmentation

We followed the basic experiments settings with [11], in which the authors also used mel-spectrogram as input. Due to the dataset's recording conditions, the audio samples' sampling rates vary in an extensive range from 4 kHz to 44.1 kHz. We first resampled them into a fixed sampling rate of 16kHz. And for different durations of the samples, we ensure that all samples have the same desirable length of 8s. For longer samples, we limited them to 8s from the beginning of each clip. While for the shorter samples, we circularly pad them until we get the standard length. Because most of the respiratory cycles are shorter than 8s.

In this length of time, the model can compose representations for most of the respiratory cycles. The spectrogram transform settings in our experiments are the default in M2D. The samples are converted into a time-frequency representation of a log-mel spectrogram with 80 mel filterbanks, a window length of 400, and a hop length of 160. The minimum and maximum frequencies are 50 Hz

Method	Architecture	Pre-training dataset	Fused Layers	Layer-wise weights	Results		
					S_p	S_e	S_c
LungRN+NL [5]	ResNet	-	-	-	63.20	41.32	52.26
LungAttn [6]	ResNet	-	-	-	71.44	36.36	53.90
Wang et al. [7]	ResNeSt [33]	ImageNet	-	-	70.40	40.20	55.30
RespireNet [8]	ResNet34	ImageNet	-	-	72.30	40.10	56.20
ARSC-Net [9]	ResNet	-	-	-	67.13	**46.38**	56.76
Nguyen et al. [10]	ResNet50	ImageNet	-	-	79.34	37.24	58.29
Moummad et al. [11]	CNN6 [19]	AudioSet	-	-	75.95	39.15	57.55
M2DViT	M2D ViT [15, 16] masking ratio=0.7	AudioSet	-	-	75.78±5.84	39.51±4.64	57.64±0.86
M2DViT-Fusion			(i) 5th & 12th	Fixed (1.0 for all)	75.43±5.22	41.18±5.80	58.30±0.97
			(ii) 5th & 11th	Fixed (1.0 for all)	**82.05±4.16**	38.06±3.24	60.05±1.00
			(iii) 4th & 7th & 10th	Fixed (1.0 for all)	79.69±2.68	39.96±1.56	59.83±0.72
			(iv) All	Fixed (1.0 for all)	79.71±3.58	40.34±2.55	59.97±0.69
			(v) All	Learnable	79.48±4.99	41.87±4.27	**60.68±0.49**

Table 2: The overall comparison of ICBHI performance of our methods and the previous ones. Except for the last result with learnable layer-wise weight, all fused layer features are of the fixed weight of 1.0. All the results are presented with the mean values and standard deviations.

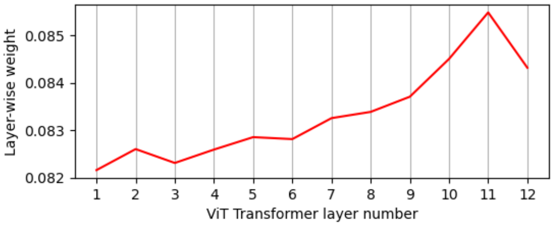


Figure 3: The learned layer-wise feature weights of the M2DViT-Fusion model with the best ICBHI score S_c. The weights are the average of best epochs from 5 runs and normalized to the sum of 1.

and 8000 Hz. Then the transformed spectrogram is normalized and standardized into a mean value of 0.3690 and a standard deviation of 0.2550. We also used the augmentation method SpecAugment [34] as in [11]. The mask sizes for time and frequency are 20 and 40, while stripes are 2 for both time and frequency in SpecAugment. We shuffled the train samples, masking the blocks of the frequency and time steps with time-wrapping augmentation to encourage the network to learn robust features from the spectrogram.

4.4. Experiments with Vanilla M2DViT

We first compared M2DViT weights pre-trained with masking ratios of 0.6 and 0.7 by fine-tuning them without our proposals. Table 1 shows that pre-trained M2DViT with a 0.7 masking ratio (M2D r=0.7) performs slightly better than M2D r=0.6, and the S_c of these two options are almost identical. While the original M2D r=0.6 showed better results on S_e, M2D r=0.7 weight was better on S_p. The original M2D r=0.6 showed better results on the speech tasks, and M2D r=0.7 was better on music tasks [15]. ICBHI with respiratory sounds is supposed to be more speech-like breathing noise [3]. Therefore, we used M2D r=0.7 in the following experiments.

4.5. Experiments with Proposals

We applied our multi-layer feature fusion methods with various layer combinations, denoted as M2DViT-Fusion, and compared them with the previous methods. Table 2 shows the results of the best-performing layer combinations in the brute-force parameter search, and Fig. 3 shows the learned layer-wise weights in the M2DViT-Fusion of the epoch with the best ICBHI score S_c. It is worth noting that the representation dimensionality varies from D in M2DViT to $D \times 2$ in (i) and (ii), $D \times 3$ in (iii), and $D \times 12$ in (iv) and (v). For the sake of experiment time constraints, we learned layer-wise weights only when using all layers.

We find that the 11th layer would provide the most significant features for the ICBHI task; Fig. 3 shows that the 11th is the best for (v) in Table. 2 with all layer fusion with learnable layer-wise weights, and the (ii) 5th & 11th layer fusion shows the best S_p result. Fig. 3 also shows a trend that the later layers perform better, though the performance drops at the last layer.

We also found that learning the layer-wise weights is better than the fixed weights, showing the effectiveness of the layer feature weighting; while the results of (v) with learnable weights and (iv) with fixed weights are highly overlapping.

Compared with the previous studies, (v) fusing all layers with learnable weights shows the best average ICBHI score of 60.68±0.49, about 2 point improvement from Nguyen et al. [10], with a score of 58.29. For the S_p, (ii) 5th & 11th shows the best result of 82.05±4.16, more than 2 point improvement from Nguyen et al., with a score of 79.34. However, for the S_e, ARSC-Net shows the best result of 46.38. Overall, we think the results validated the effectiveness of our proposals.

5. CONCLUSION

We introduced a novel feature fusion method to the classification task on the ICBHI dataset. And in the experiments, our M2DViT-Fusion methods showed a better performance than the vanilla M2DViT. The results proved that multi-layer feature fusion is an effective way to extract effective audio representations, including the proposed learnable layer-wise weight. In the layer weight analysis, we also found the later layers contribute more. The fine-tuned model got the best ICBHI score of 60.68 on the ICBHI dataset, which is improved by 2.39 compared to the previous SOTA method. While we exhibited improvements, the result would still need further improvement for practical diagnosis assistance. Possible directions may include effective augmentation techniques and new large-scale respiratory sound datasets to help models achieve desirable performance in the future.

6. ACKNOWLEDGEMENT

This work was supported by JSPS KAKENHI Grant Number 23H03423.

7. REFERENCES

[1] S. M. Levine and D. D. Marciniuk, “Global impact of respiratory disease: What can we do, together, to make a difference?” *Chest*, vol. 161, no. 5, pp. 1153–1154, 2022.

[2] M. A. Fernandez-Granero, D. Sanchez-Morillo, and A. Leon-Jimenez, “Computerised analysis of telemonitored respiratory sounds for predicting acute exacerbations of COPD,” *Sensors*, vol. 15, no. 10, pp. 26 978–26 996, 2015.

[3] B. M. Rocha, D. Filos, L. Mendes, G. Serbes, S. Ulukaya, Y. P. Kahya, N. Jakovljevic, T. L. Turukalo, I. M. Vogiatzis, E. Perantoni, *et al.*, “An open access database for the evaluation of respiratory sound classification algorithms,” *Physiological measurement*, vol. 40, no. 3, p. 035001, 2019.

[4] R. X. A. Pramono, S. Bowyer, and E. Rodriguez-Villegas, “Automatic adventitious respiratory sound analysis: A systematic review,” *PloS one*, vol. 12, no. 5, p. e0177926, 2017.

[5] Y. Ma, X. Xu, and Y. Li, “Lungrn+ nl: An improved adventitious lung sound classification using non-local block resnet neural network with mixup data augmentation.” in *Interspeech*, 2020, pp. 2902–2906.

[6] J. Li, J. Yuan, H. Wang, S. Liu, Q. Guo, Y. Ma, Y. Li, L. Zhao, and G. Wang, “LungAttn: Advanced lung sound classification using attention mechanism with dual TQWT and triple STFT spectrogram,” *Physiological Measurement*, vol. 42, no. 10, p. 105006, 2021.

[7] Z. Wang and Z. Wang, “A domain transfer based data augmentation method for automated respiratory classification,” in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 9017–9021.

[8] S. Gairola, F. Tom, N. Kwatra, and M. Jain, “Respirenet: A deep neural network for accurately detecting abnormal lung sounds in limited data setting,” in *2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, 2021, pp. 527–530.

[9] L. Xu, J. Cheng, J. Liu, H. Kuang, F. Wu, and J. Wang, “Arsc-net: Adventitious respiratory sound classification network using parallel paths with channel-spatial attention,” in *2021 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, 2021, pp. 1125–1130.

[10] T. Nguyen and F. Pernkopf, “Lung sound classification using co-tuning and stochastic normalization,” *IEEE Transactions on Biomedical Engineering*, vol. 69, no. 9, pp. 2872–2882, 2022.

[11] I. Moummad and N. Farrugia, “Learning audio features with metadata and contrastive learning,” *arXiv preprint arXiv:2210.16192*, 2022.

[12] A. Baade, P. Peng, and D. Harwath, “MAE-AST: Masked Autoencoding Audio Spectrogram Transformer,” in *Interspeech*, 2022, pp. 2438–2442.

[13] D. Niizumi, D. Takeuchi, Y. Ohishi, N. Harada, and K. Kashino, “Byol for audio: Self-supervised learning for general-purpose audio representation,” in *IJCNN*, Jul 2021.

[14] Y. Gong, C.-I. Lai, Y.-A. Chung, and J. Glass, “SSAST: Self-Supervised Audio Spectrogram Transformer,” in *AAAI*, vol. 36, no. 10, 2022, pp. 10 699–10 709.

[15] D. Niizumi, D. Takeuchi, Y. Ohishi, N. Harada, and K. Kashino, “Masked modeling duo: Learning representations by encouraging both networks to model the input,” in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023, pp. 1–5.

[16] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, *et al.*, “An image is worth 16x16 words: Transformers for image recognition at scale,” *arXiv preprint arXiv:2010.11929*, 2020.

[17] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in *2009 IEEE conference on computer vision and pattern recognition*, 2009, pp. 248–255.

[18] J. F. Gemmeke, D. P. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, “Audio set: An ontology and human-labeled dataset for audio events,” in *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, 2017, pp. 776–780.

[19] Q. Kong, Y. Cao, T. Iqbal, Y. Wang, W. Wang, and M. D. Plumbley, “Panns: Large-scale pretrained audio neural networks for audio pattern recognition,” *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 28, pp. 2880–2894, 2020.

[20] Y. Gong, Y.-A. Chung, and J. Glass, “AST: Audio Spectrogram Transformer,” in *Interspeech*, 2021, pp. 571–575.

[21] S. Bae, J.-W. Kim, W.-Y. Cho, H. Baek, S. Son, B. Lee, C. Ha, K. Tae, S. Kim, and S.-Y. Yun, “Patch-mix contrastive learning with audio spectrogram transformer on respiratory sound classification,” *arXiv preprint arXiv:2305.14032*, 2023.

[22] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, “Masked autoencoders are scalable vision learners,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 16 000–16 009.

[23] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Avila Pires, Z. Guo, M. Gheshlaghi Azar, *et al.*, “Bootstrap your own latent-a new approach to self-supervised learning,” *Advances in neural information processing systems*, vol. 33, pp. 21 271–21 284, 2020.

[24] X. Xu and J. Hao, “Multi-layer feature fusion convolution network for audio-visual speech enhancement,” *arXiv preprint arXiv:2101.05975*, 2021.

[25] B. Yu, Z. Zhang, D. Zhao, and Y. Wang, “Audio-visual speech enhancement with deep multi-modality fusion,” in *2022 5th International Conference on Information Communication and Signal Processing (ICICSP)*. IEEE, 2022, pp. 143–147.

[26] Y. Dai, F. Gieseke, S. Oehmcke, Y. Wu, and K. Barnard, “Attentional feature fusion,” in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2021, pp. 3560–3569.

[27] X. Lv, H. Xia, N. Li, X. Li, and R. Lan, “Mfvt: Multilevel feature fusion vision transformer and ramix data augmentation for fine-grained visual categorization,” *Electronics*, vol. 11, no. 21, p. 3552, 2022.

[28] D. Niizumi, D. Takeuchi, Y. Ohishi, N. Harada, and K. Kashino, “Composing general audio representation by fusing multilayer features of a pre-trained model,” in *2022 30th European Signal Processing Conference (EUSIPCO)*, 2022, pp. 200–204.

[29] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[30] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[31] A. E. Orhan and X. Pitkow, “Skip connections eliminate singularities,” *arXiv preprint arXiv:1701.09175*, 2017.

[32] A. Pasad, J.-C. Chou, and K. Livescu, “Layer-wise analysis of a self-supervised speech representation model,” in *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2021, pp. 914–921.

[33] H. Zhang, C. Wu, Z. Zhang, Y. Zhu, H. Lin, Z. Zhang, Y. Sun, T. He, J. Mueller, R. Manmatha, *et al.*, “Resnest: Split-attention networks,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 2736–2746.

[34] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, “Specaugment: A simple data augmentation method for automatic speech recognition,” *arXiv preprint arXiv:1904.08779*, 2019.

EFFICIENT EVALUATION ALGORITHMS FOR SOUND EVENT DETECTION

Vincent Lostanlen[1]* *and Brian McFee*[2]†

[1] Nantes Université, École Centrale Nantes, CNRS, LS2N, UMR 6004, F-44000 Nantes, France
[2] New York University, Music and Audio Research Lab & Center for Data Science, New York, NY, USA

ABSTRACT

The prediction of a sound event detection (SED) system may be represented on a timeline by intervals whose bounds correspond to onset and offset respectively. In this context, SED evaluation requires to find all non-empty intersections between predicted and reference intervals. Denoting by M and N the number of predicted events and reference events, the time complexity of exhaustive search is $O(MN)$. This is particularly inefficient when the acoustic scene of interest contains many events (typically above 10^3) or when the detection threshold is low. Our article presents an algorithm for pairwise intersection of intervals by performing binary search within sorted onset and offset times. Computational benchmarks on the BirdVox-full-night dataset confirms that our algorithm is significantly faster than exhaustive search. Moreover, we explain how to use this list of intersecting prediction–reference pairs for the purpose of SED evaluation: the Hopcroft-Karp algorithm guarantees an optimal bipartite matching in time $O((M+N)^{3/2})$ in the best case (all events are pairwise disjoint) and $O((M+N)^{5/2})$ in the worst case (all events overlap with each other). The solution found by Hopcroft-Karp unambiguously defines a number of true positives, false positives, and false negatives; and ultimately, information-retrieval metrics such as precision, recall, and F-score.

Index Terms— Evaluation procedures, sound event detection.

1. INTRODUCTION

Given a sound category of interest, the task of sound event detection (SED) aims to identify occurrences of this sound category within an audio recording. SED systems are optimized to pinpoint each instance of the target sound over the time axis. This is known as "strong" labeling, as opposed to "weak" labeling which only reports presence versus absence. In recent years, the renewed interest for deep learning in SED has found many fruitful applications to conservation biology, urban science, industry, and healthcare.

Evaluating the performance of an SED system is not so simple as evaluating a classifier of acoustic scenes. Let us denote the prediction of the system by $\boldsymbol{x}$ and the reference by $\boldsymbol{y}$. We use symbols $\wedge$, $\vee$, and $\neg$ for conjunction (AND), disjunction (OR), and negation (NOT) respectively. With weak labels, $\boldsymbol{x}$ and $\boldsymbol{y}$ boil down to a single bit, and may be compared with elementary logical operations: $(\boldsymbol{x} \wedge \boldsymbol{y})$ for a true positive (TP), $(\boldsymbol{x} \wedge \neg\boldsymbol{y})$ for a false positive (FP), and $(\neg\boldsymbol{x} \wedge \boldsymbol{y})$ for a false negative (FN). The time complexity of this evaluation is independent of the content of $\boldsymbol{x}$ and $\boldsymbol{y}$, i.e., $O(1)$.

The situation is different with strong labels since they are localized in time and potentially repeated over multiple instances. For this matter, we may express the prediction $\boldsymbol{x}$ in terms of a list of M time intervals over $\mathbb{R}$: $(\boldsymbol{x}_1, \dots, \boldsymbol{x}_M) = ([a_1, b_1], \dots, [a_M, b_M])$. Each of these intervals represents a different predicted instance of the target sound, with the lower and upper bound denoting sound onset (start time) and offset (end time) respectively. Likewise, we define the reference $\boldsymbol{y}$ as a list of N intervals: $\boldsymbol{y} = (\boldsymbol{y}_1, \dots, \boldsymbol{y}_N) = ([u_1, v_1], \dots, [u_N, v_N])$. The evaluation procedure specifies a binary operator, later denoted by $\approx$, which determines whether a predicted interval $\boldsymbol{x}_m$ may be matched to a reference interval $\boldsymbol{y}_n$. An important example of such operator consists in checking whether $\boldsymbol{x}_m$ and $\boldsymbol{y}_n$ have a non-empty intersection:

$$
\begin{aligned}
(\boldsymbol{x}_m \approx \boldsymbol{y}_n) &\iff (\boldsymbol{x}_m \cap \boldsymbol{y}_n) \neq \emptyset \\
&\iff (a_m \leq v_n) \wedge (b_m \geq u_n). \qquad (1)
\end{aligned}
$$

The predicted number of events is equal to M and the true number of events is equal to N. Meanwhile, computing the number of true positives (TP) is more challenging because it posits that each interval cannot be matched more than once during evaluation. Formally speaking, we consider a graph $\mathcal{G}$ whose vertices are partitioned into two subsets, $\boldsymbol{x}$ and $\boldsymbol{y}$, and whose edges $\mathcal{E}$ correspond to all interval pairs $(\boldsymbol{x}_m, \boldsymbol{y}_n)$ satisfying Equation (1). We seek three subsets $\mathcal{X} \subseteq \boldsymbol{x}$, $\mathcal{Y} \subseteq \boldsymbol{y}$, and $\mathcal{Z} \subseteq \mathcal{E}$ of highest cardinal, under the constraint that each vertex $\boldsymbol{x}_m \in \mathcal{X}$ and each $\boldsymbol{y}_n \in \mathcal{Y}$ must be incident to at most one of the edges in $\mathcal{Z}$. Hence, SED evaluation comprises two stages:

1. Construct the full edge set

$$\mathcal{E} = \{(\boldsymbol{x}_m, \boldsymbol{y}_n) \subseteq \mathcal{X} \times \mathcal{Y} \mid \boldsymbol{x}_m \approx \boldsymbol{y}_n\}, \qquad (2)$$

2. Identify a maximal subset $\mathcal{Z} \subseteq E$ such that each interval of $(\boldsymbol{x} \cup \boldsymbol{y})$ appears at most once. In mathematical terms:

$$
\begin{aligned}
&\forall \boldsymbol{x}_m \in \boldsymbol{x}, \forall \boldsymbol{x}_{m'} \in \boldsymbol{x} \setminus \{\boldsymbol{x}_m\}, \forall \boldsymbol{y}_n \in \boldsymbol{y}, \forall \boldsymbol{y}_{n'} \in \boldsymbol{y} \setminus \{\boldsymbol{y}_n\}, \\
&(\boldsymbol{x}_m, \boldsymbol{y}_n) \in \mathcal{Z} \implies \big((\boldsymbol{x}_{m'}, \boldsymbol{y}_n) \notin \mathcal{Z}\big) \wedge \big((\boldsymbol{x}_m, \boldsymbol{y}_{n'}) \notin \mathcal{Z}\big) \\
&\qquad (3)
\end{aligned}
$$

The second stage is efficiently solved by the Hopcroft-Karp algorithm [1]. Meanwhile, our work focuses on the first stage: i.e., to efficiently identify all candidate matchings. This problem is solvable in time $O(MN)$ by comparing all pairs of intervals. However, in SED settings which cover long time periods, the number of detected intervals may range in the thousands, making the all-pairs matching approach inefficient in practice.

The key observation is that most comparisons of the form $\boldsymbol{x}_m \approx \boldsymbol{y}_n$ will evaluate to false and can be discarded in advance. Indeed, if we know a lower bound b on the onset u_n of $\boldsymbol{y}_n$, then we can conclude that any interval $\boldsymbol{x}_m = [a_m, b_m]$ such that $b_m < b$ is necessarily disjoint from $\boldsymbol{y}_n$; and likewise if we know an upper bound a on the offset v_n such that $a_m > a$.

---

[*] VL is supported by CNRS grant CAPTEO, WeAMEC grant PETREL, and Horizon Europe BioacAI.

[†] BM is supported by NSF award 1955357.

In this article, we propose an algorithm for evaluating SED efficiently; i.e., without examining all pairs. More precisely, our algorithm implements maximum cardinal matching on interval bigraphs [2], and has an asymptotic time complexity of

$$O\left((M+N)(\log M+\log N)+|\mathcal{E}|\sqrt{M+N}\right). \tag{4}$$

We begin by explaining why a greedy approach, in which all intervals are visited once, is not guaranteed to return the optimal number of true positives, and thus should not be used. Then, we present the two stages of our algorithm: construction of the interval bigraph and maximum cardinality matching. We discuss the use of our algorithm since 2021 as part of Task 5 of the DCASE challenge on few-shot bioacoustic event detection; and its connection with an existing algorithm for efficient evaluation of sound event detection in mir_eval. We conclude with a performance benchmark on a realistic use case, namely, automatic detection of avian flight calls in the BirdVox-full-night dataset [3].

2. SUBOPTIMAL GREEDY ALGORITHM

Algorithm 1 Exhaustive search of matching pairs $(\boldsymbol{x}_m, \boldsymbol{y}_n)$. The length of $\mathcal{E}$ is an upper bound on TP. Complexity: $O(MN)$.

```
$\mathcal{E}$ = list()
for $m = 1$ to $M$ do
    for $n = 1$ for $N$ do
        if $\boldsymbol{x}_m \approx \boldsymbol{y}_n$ then
            append $(\boldsymbol{x}_m, \boldsymbol{y}_n)$ to $\mathcal{E}$
        end if
    end for
end for
return $\mathcal{E}$
```

Algorithm 2 Greedy search of matching pairs $(\boldsymbol{x}_m, \boldsymbol{y}_n)$. Lists $\mathcal{X}$ and $\mathcal{Y}$ have the same length as $\mathcal{Z}$ and contain non-repeating elements only. This algorithm gives a lower bound on TP and should not be used for SED evaluation. Worst-case complexity: $O(MN)$.

```
$\mathcal{Z}$ = list()
$\mathcal{X}$ = list()
$\mathcal{Y}$ = list()
for $m = 1$ to $M$ do
    for $n = 1$ for $N$ do
        if $(\boldsymbol{x}_m \approx \boldsymbol{y}_n) \wedge (\boldsymbol{y}_n \notin \mathcal{Y})$ then
            append $(\boldsymbol{x}_m, \boldsymbol{y}_n)$ to $\mathcal{Z}$
            append $\boldsymbol{x}_m$ to $\mathcal{X}$
            append $\boldsymbol{y}_n$ to $\mathcal{Y}$
            break
        end if
    end for
end for
return $\mathcal{X}, \mathcal{Y}, \mathcal{Z}$
```

At first glance, the evaluation of an SED system may seem easy. Since the number of true positives (TP) involves non-disjoint pairs between a prediction interval $\boldsymbol{x}_m$ and a reference interval $\boldsymbol{y}_n$, one could list those pairs exhaustively with a double loop, as in Algorithm 1. Yet, this algorithm involves $O(MN)$ comparisons and only produces an upper bound on TP, since the same interval may appear in multiple pairs of the list $\mathcal{E}$. A potential workaround consists in defining a list $\mathcal{X}$ containing all the prediction intervals that have been matched so far, and grow it as we traverse the list $\boldsymbol{x}$; and likewise for $\mathcal{Y}$ and $\boldsymbol{y}$. By only admitting a new pair $(\boldsymbol{x}_m, \boldsymbol{y}_n)$ if $\boldsymbol{x}_m$ does not already belong to $\mathcal{X}$ nor $\boldsymbol{y}_n$ to $\mathcal{Y}$, one guarantees that each prediction interval is matched to at most one reference interval and vice versa: see Algorithm 2. This is a form of “greedy” search: it makes a locally optimal choice at each stage, yet is globally suboptimal. In this instance, the inner loop in Algorithm 2 consists in looking for an unmatched reference interval ($\boldsymbol{y}_n \notin \mathcal{Y}$) such that $\boldsymbol{x}_m \approx \boldsymbol{y}_n$, given an unmatched prediction interval ($\boldsymbol{x}_m \notin \mathcal{X}$).

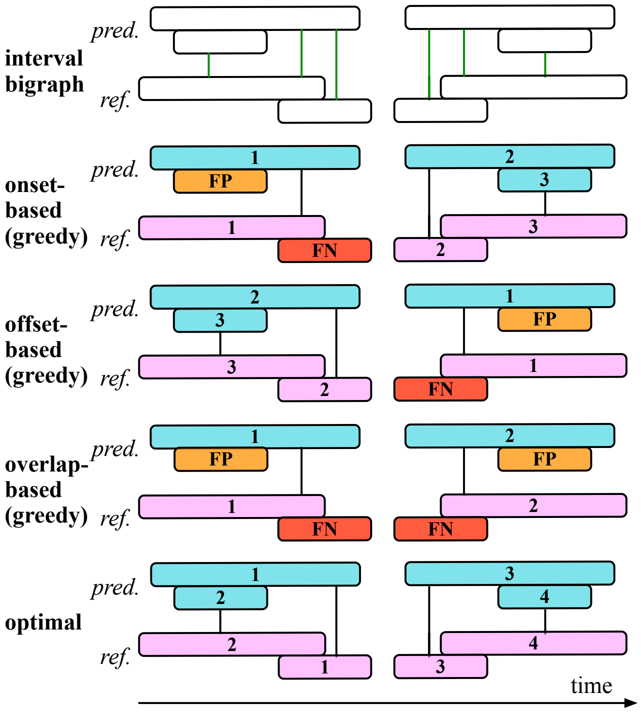


Figure 1: Top: exhaustive search (Algorithm 1) overestimates the number of true positives (TP) of sound event detection, defined as the maximum cardinality matching between prediction (blue) and reference (orange). Center: greedy search (Algorithm 2) underestimates TP. Bottom: our algorithm (Algorithm 3) returns the correct value of TP. See Section 2 for details.

Figure 2 illustrates the problem of SED evaluation between a prediction of $M = 4$ intervals and a reference of $N = 4$ intervals. The top row, in white, implements Algorithm 1 on this example, yielding a list $\mathcal{E}$ of six pairs, each of the form $(\boldsymbol{x}_m, \boldsymbol{y}_n)$. Greedy algorithms, such as Algorithm 2, achieve this by traversing the list $\mathcal{E}$ once, in a predefined order. In this way, they define a sublist $\mathcal{Z} \subset \mathcal{E}$, which is initialized as the empty list and grown progressively until all pairs in $\mathcal{E}$ have been examined. The second, third, and fourth rows in Figure 2 show instances of such greedy algorithms, with variations in priority: i.e., based on earlier onset time a_m, on later offset time b_m, or on greatest overlap (as done in MIREX). We observe that, even though these greedy algorithms differ in terms of pairing sublist $\mathcal{Z}$, both of them leave one prediction event and one reference event unmatched. Hence, they evaluate the prediction as yielding three TP, one FP, and one FN.

Yet, all the greedy algorithms mentioned above are suboptimal. Indeed, in Figure 2, there exists a matching which yields four TP and no FP nor FN: see bottom row. In the next section, we present an algorithm which finds this optimal solution in polynomial time.

3. MAIN CONTRIBUTION

Our proposed method comprises two stages: construction of the interval bigraph $\mathcal{G} = (\boldsymbol{x} \cup \boldsymbol{y}, \mathcal{E})$ and maximum cardinality matching. For the first stage, we write a custom iterative algorithm which is based on dichotomic search: see Algorithm 3. For the second stage, we reuse the algorithm of Hopcroft and Karp for maximum cardinality matching of bipartite graphs [1]. The novelty of our method resides in the faster construction of the set of edges $\mathcal{E}$ in the first stage, by exploiting the temporal logic of intervals in Equation 1.

Algorithm 3 presents our main contributions, i.e., the fast construction of the edge set $\mathcal{E}$. We begin by sorting predicted offsets $(b_1, \ldots, b_M)$, yielding a permutation σ satisfying $b_{\sigma(1)} \leq \ldots \leq b_{\sigma(M)}$. Thanks to divide-and-conquer algorithms such as quicksort, this operation incurs an average-case time complexity of $O(M \log M)$. Likewise, we sort reference onsets $(u_1, \ldots, u_N)$, yielding a permutation ϕ satisfying $u_{\phi(1)} \leq \ldots \leq u_{\phi(N)}$ with average-case $O(N \log N)$ time complexity. Then, we initialize two integers n and μ, pointing to the first element of ϕ and σ respectively. We define the partial list of sorted predicted offsets $L = (b_{\sigma(\mu)}, \ldots b_{\sigma(M)})$ and increment μ by the index of the first element in L exceeding $u_{\phi(n)}$. Since L is already sorted, this element may be found via binary search, whose number of comparisons is logarithmic in M in the worst case.

After having updated μ, we store the set $\{\sigma(\mu), \ldots, \sigma(M)\}$ at the entry $\phi(n)$ of a set-valued array $\mathcal{S}$. We increment n by one and repeat the operation of dichotomic search over the list L with the new value $u_{\phi(n)}$. We halt this procedure as soon as $u_{\phi(n)}$ exceeds $b_{\sigma(M)}$: indeed, for greater values of n, $\mathcal{S}(\phi(n))$ is known to be empty. By construction, each set $\mathcal{S}(\phi(n))$ contains all the indices of the predicted events whose offsets happen after the onset of the reference event $\phi(n)$. Formally:

$$\begin{aligned}\mathcal{S}(\phi(n)) &= \left\{1 \leq \sigma(m) \leq M \,\middle|\, b_{\sigma(m)} \geq u_{\phi(n)}\right\} \\ &= \left\{1 \leq m \leq M \,\middle|\, b_m \geq u_n\right\} \qquad (5)\end{aligned}$$

The latter formula in the equation above is obtained after applying inverse permutations σ^{-1} and ϕ^{-1} to indices m and n respectively. Going back to Equation 1, we observe that $(\boldsymbol{x}_m \approx \boldsymbol{y}_n)$ implies $(m \in \mathcal{S}(n))$ but the converse is not necessarily true. However, if $(m \in \mathcal{S}(n))$ for some pair (m, n), a necessary and sufficient condition for $(\boldsymbol{x}_m \approx \boldsymbol{y}_n)$ is $(a_m \leq v_n)$. Thus, we propose to refine each set $\mathcal{S}(n)$ by intersecting it with the set of all indices m such that $a_m \leq v_n$.

We sort predicted onsets $(a_1, \ldots, a_M)$, yielding a permutation π which satisfies $a_{\pi(1)} \leq \ldots \leq a_{\pi(M)}$. Likewise, we sort reference offsets $(v_1, \ldots, v_N)$, yielding a permutation ψ which satisfies $v_{\psi(1)} \leq \ldots \leq v_{\psi(N)}$. Similarly to σ and ϕ in the paragraph above, these sorting operations incur a cumulated asymptotic cost of $O(M \log M + N \log N)$ in the average case. We reset the integer to n to N. We set μ to the maximum value of m such that $a_{\pi(m)}$ is below $v_{\psi(n)}$. Thanks to sorting, this may be achieved by binary search, whose worst-case complexity is $O(\log M)$. We update the list $\mathcal{S}(\psi(n))$ by intersecting it with $(\pi(1), \ldots, \pi(\mu))$. This intersection may be implemented efficiently with a hash table, as it does not involve any numerical comparison. We decrement n by one and repeat the process until $v_{\psi(n)}$ falls below $a_{\pi(1)}$. Finally, we build $\mathcal{E}$ incrementally by looping through every value $m \in \mathcal{S}(n)$ for n from 1 to N and constructing the pair $(\boldsymbol{x}_m, \boldsymbol{y}_n)$. In practice, since $\mathcal{S}(n)$ has much fewer than M elements, the number of iterations in this nested loop is typically much smaller than $O(MN)$. Thus, most of the computational cost of Algorithm 3 is spent in binary search.

	naïve	proposed
graph construction	MN	$(M+N)(\log M + \log N) + \lvert\mathcal{E}\rvert$ (Algorithm 3)
event matching	$2^{\lvert\mathcal{E}\rvert}$	$\lvert\mathcal{E}\rvert\sqrt{M+N}$ (Hopcroft-Karp)

Table 1: Upper bounds on the asymptotic time complexities of algorithms for constructing non-disjoint interval pairs $\mathcal{E}$ (left column) and maximum cardinality matching $\mathcal{Z}$ (right column) in the worst case, up to a constant multiplicative factor. See Section 3 for details.

Algorithm 3 Our algorithm lists all edges of an interval bigraph.

```
$\mathcal{E} \leftarrow \emptyset$
$\mathcal{S}(1), \ldots, \mathcal{S}(N) \leftarrow \emptyset$ {initialize list of matching indices}
$\sigma \leftarrow \arg\mathrm{sort}(b_1, \ldots, b_M)$ {sort predicted offsets}
$\phi \leftarrow \arg\mathrm{sort}(u_1, \ldots, u_N)$ {sort reference onsets}
$n \leftarrow 1$
$\mu \leftarrow 1$
while $u_{\phi(n)} \leq b_{\sigma(M)}$ {up to last predicted offset} do
  $L \leftarrow (b_{\sigma(\mu)}, \ldots, b_{\sigma(M)})$ {sublist of predicted offsets}
  $\mu \leftarrow \mu + \min\left\{0 \leq i \leq (M - \mu) \,\middle|\, u_{\phi(n)} \leq L_{i+1}\right\}$
  $\mathcal{S}(\phi(n)) \leftarrow \{\sigma(\mu), \ldots, \sigma(M)\}$
  $n \leftarrow n + 1$
end while
$\pi \leftarrow \arg\mathrm{sort}(a_1, \ldots, a_M)$ {sort predicted onsets}
$\psi \leftarrow \arg\mathrm{sort}(v_1, \ldots, v_N)$ {sort reference offsets}
$n \leftarrow N$
while $v_{\psi(n)} \geq a_{\pi(1)}$ {down to first predicted onset} do
  $\mu \leftarrow \max\left\{1 \leq m \leq \mu \,\middle|\, v_{\psi(n)} \geq a_{\pi(m)}\right\}$
  $\mathcal{S}(\psi(n)) \leftarrow \mathcal{S}(\psi(n)) \cap (\pi(1), \ldots, \pi(\mu))$
  $n \leftarrow n - 1$
end while
for $n$ from 1 to $N$ {for every reference event} do
  for $m \in \mathcal{S}(n)$ {for every matching prediction} do
    $\mathcal{E} \leftarrow \mathcal{E} \cup \{(I_m, J_n)\}$ {include edge}
  end for
end for
return $\mathcal{E}$
```

As shown in Table 1, Algorithm 3 accelerates graph construction from $O(MN)$ to $O(M \log M + N \log N + |\mathcal{E}|)$. If the detector is perfect ($\boldsymbol{x} = \boldsymbol{y}$) and if the reference consists of disjoint intervals, one has $M = N = |\mathcal{E}|$: under this important special case, the time complexity of graph construction is $O(N^2)$ for exhaustive search (Algorithm 1) versus $O(N \log N)$ for binary search (Algorithm 3). Furthermore, the complexity of event matching is 2^N for exhaustive search versus $N\sqrt{N}$ for Hopcroft-Karp. The interest behind our contribution is that $N \log N$, unlike N^2, is dominated by $N\sqrt{N}$; thus, after replacing Algorithm 1 by Algorithm 3, the cost of graph construction is asymptotically negligible, and most of SED evaluation is spent in event matching.

4. PRACTICAL CONSIDERATIONS

4.1. Selecting pairs based on intersection-over-union ratio

We have implemented Algorithm 3 as part of the official evaluation toolkit[1] of the DCASE challenge task on "Few-shot bioacoustic event detection" [4]. This task was inaugurated in 2021 and maintained through 2022 and 2023. As part of the challenge rules, we have stated that, in order to be considered a valid matching, an interval pair $(\boldsymbol{x}_m, \boldsymbol{y}_n)$ should not only overlap but also have at least 50% of intersection-over-union ratio (IoU). This is a refinement of Equation 1 in the criterion $\boldsymbol{x}_m \approx \boldsymbol{y}_n$. To accelerate the construction of the bipartite graph $\mathcal{G}$, we run Algorithm 3 as a prefiltering stage and then evaluate IoU explicitly on all non-disjoint pairs.

4.2. Interoperability with sed_eval

Running Algorithm 3 in conjunction with Hopcroft-Karp yields a maximal matching $\mathcal{Z}$ for $\mathcal{G}$. The set cardinal of $\mathcal{Z}$ corresponds to the number of true positives (TP) of the detector. From this number, we deduce the following information-retrieval metrics:

$$\text{Precision} = \frac{\text{TP}}{\text{TP}+\text{FP}} = \frac{|\mathcal{Z}|}{M}, \tag{6}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP}+\text{FN}} = \frac{|\mathcal{Z}|}{N}, \tag{7}$$

$$F_1\text{-score} = \frac{2}{(\text{Precision})^{-1} + (\text{Recall})^{-1}} = \frac{2|\mathcal{Z}|}{M+N}, \tag{8}$$

which are already in widespread use in the DCASE community thanks to the sed_eval toolbox [5]. While sed_eval relies on exhaustive search (Algorithm 1) for graph construction, we rely on Algorithm 3. For event matching, both sed_eval and our implementation rely on the Hopcroft-Karp algorithm, as made available by the SciPy toolbox. Therefore, our implementation returns the same output as sed_eval while being more computationally efficient.

4.3. An important special case: evaluating onset detection

Algorithm 3 generalizes another algorithm, implemented under the name of "_fast_hit_windows" in mir_eval v0.5 and later [6]. In combination with Hopcroft-Karp, this other algorithm serves to evaluate sound onset detection efficiently. For future reference, we present its pseudocode in Algorithm 4. The premise of Algorithm 4 is that a predicted onset a_m may be matched to a reference onset u_n if and only if they are within a certain time lag δ of each other. Formally:

$$\begin{aligned}(a_m \approx u_n) &\iff |a_m - u_n| \leq \delta \\ &\iff (u_n \geq a_m - \delta) \wedge (u_n \leq a_m + \delta)\end{aligned} \tag{9}$$

Like Algorithm 3, Algorithm 4 begins by sorting reference onsets, which incurs an $O(N \log N)$ time complexity. Then, for every predicted onset u_m, it performs binary search over the sorted list of reference onsets $(u_{\phi(1)}, \dots, u_{\phi(N)})$, while accounting for the maximum admissible lag δ. This later stage incurs a time complexity of $O(M \log N)$; hence, the time complexity of Algorithm 4 is $O((N+M)\log N)$. Note that, if there are many more predicted onsets than reference onsets ($N \gg M$), the algorithm may be accelerated by a factor $(\log N / \log M)$ by swapping the roles of prediction and reference.

Algorithm 4 An efficient evaluation algorithm for sound onset detection, as implemented in the mir_eval v0.5 and later.

```
$\mathcal{E} \leftarrow \emptyset$
$\phi \leftarrow \arg\mathrm{sort}(u_1, \dots, u_N)$
for $m$ from 1 to $M$ do
    $n_{\min} \leftarrow \min\{1 \leq n \leq N \mid u_{\phi(n)} \geq a_m - \delta\}$
    $n_{\max} \leftarrow \max\{1 \leq n \leq N \mid u_{\phi(n)} \leq a_m + \delta\}$
    for $n$ from $n_{\min}$ to $n_{\max}$ do
        $\mathcal{E} \leftarrow \mathcal{E} \cup (a_m, u_{\phi(n)})$
    end for
end for
return $\mathcal{E}$
```

5. EXAMPLE APPLICATION

To evaluate the speed of exhaustive search (Algorithm 1) versus our algorithm (Algorithm 3), we evaluate a deep convolutional network for automatic detection of avian flight calls on an audio recording from the BirdVox-full-night dataset [3]. This audio recording lasts for roughly 11 hours and has been annotated by an expert ornithologist. The reference $\boldsymbol{y}$ contains $N = 9113$ events. We set the threshold of the convnet detector to a value such that the prediction $\boldsymbol{x}$ contains $M = 2N = 18226$ events.

In the Python programming language, Algorithm 1 takes 65 ± 1 seconds to find all matching pairs between $\boldsymbol{x}$ and $\boldsymbol{y}$ on a personal computer (2.3 GHz Quad-Core Intel Core i7). On the same computer, Algorithm 3 returns the same output within 11.7 ± 0.1 seconds. Beyond the raw comparison, we note that the speed could be improved further by resorting to a high-performance compiler such as Numba. We should also keep in mind that, in practice, computing the area under the precision–recal curve (AUPRC) requires to recompute the bipartite graph $\mathcal{G}$ for many values of the detection threshold, including low values when $M \gg N$. Furthermore, SED evaluation is typically performed over several initializations of the system and across several hyperparameter choices, as in [7]. Hence, the gain in speed by switching from Algorithm 1 to Algorithm 3 becomes significant when conducting a full-scale benchmark.

6. CONCLUSION

With this article, we have stressed the difficulty of making SED evaluation both correct and computationally efficient, by pointing out the shortcomings of greedy methods and of exhaustive search. We have presented an algorithm evaluating sound event detection, which generalizes an evaluation algorithm for onset detection in mir_eval. Our theoretical analysis and speed benchmark on a long-duration audio recording demonstrate the interest of this algorithm.

7. ACKNOWLEDGMENT

We thank Colin Raffel for maintaining the mir_eval package, which serves as an inspiration to this work. We also thank Veronica Morfi, Inês Nolasco, and Shubhr Singh, and Dan Stowell for maintaining the evaluation software of DCASE Task 5.

8. REFERENCES

[1] J. E. Hopcroft and R. M. Karp, "An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs," *SIAM Journal on comput-*

---

[1] Source code: https://github.com/c4dm/dcase-few-shot-bioacoustic. The metrics module implement functions slow_intersect (Algorithm 1) and fast_intersect (Algorithm 3).

ing, vol. 2, no. 4, pp. 225–231, 1973.

[2] A. K. Das and R. Chakraborty, “New characterizations of proper interval bigraphs,” *AKCE International Journal of Graphs and Combinatorics*, vol. 12, no. 1, pp. 47–53, 2015.

[3] V. Lostanlen, J. Salamon, A. Farnsworth, S. Kelling, and J. P. Bello, “Birdvox-full-night: A dataset and benchmark for avian flight call detection,” in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 266–270.

[4] V. Morfi, I. Nolasco, V. Lostanlen, S. Singh, A. Strandburg-Peshkin, L. F. Gill, H. Pamula, D. Benvent, and D. Stowell, “Few-shot bioacoustic event detection: A new task at the DCASE 2021 challenge,” in *Proceedings of the International Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE)*, 2021, pp. 145–149.

[5] A. Mesaros, T. Heittola, and T. Virtanen, “Metrics for polyphonic sound event detection,” *Applied Sciences*, vol. 6, no. 6, p. 162, 2016.

[6] C. Raffel, B. McFee, E. J. Humphrey, J. Salamon, O. Nieto, D. Liang, D. P. Ellis, and C. C. Raffel, “mir_eval: A transparent implementation of common mir metrics,” in *Proceedings of the International Society of Music Information Retrieval (ISMIR) Conference*, 2014, pp. 367–372.

[7] V. Lostanlen, J. Salamon, A. Farnsworth, S. Kelling, and J. P. Bello, “Robust sound event detection in bioacoustic sensor networks,” *PLOS ONE*, vol. 14, no. 10, p. e0214168, 2019.

AGGREGATE OR SEPARATE: LEARNING FROM MULTI-ANNOTATOR NOISY LABELS FOR BEST CLASSIFICATION PERFORMANCE

Irene Martín-Morató, Paul Ahokas, Annamaria Mesaros

Computing Sciences, Tampere University, Tampere, FINLAND
irene.martinmorato@tuni.fi, paul.ahokas@tuni.fi, annamaria.mesaros@tuni.fi

ABSTRACT

While there is the saying of two heads are better than one, having multiple opinions brings the problem of finding a common ground. For data, multiple annotator opinions are usually aggregated into a single set of labels, regarded as the ground truth. With this ground truth, classification models can be trained in a supervised way to learn the annotated data categories. Finding a suitable aggregation for multiple annotator opinions is the topic of research in many domains. In this work we investigate the use of raw data obtained from multiple annotators with various levels of reliability, to train a model for audio classification. The model sees all the individual annotator opinions and learns the categories without the need of aggregating the information. The results show that using a fully-connected layer that models individual annotators, it is possible to leverage the data distribution and learn to classify sounds without the need for aggregation of labels.

Index Terms**—** audio tagging, multi-annotator, crowd layer.

1. INTRODUCTION

Identifying what sounds are present in an audio clip can be used in multiple applications such as surveillance [1], environment monitoring [2], health care monitoring [3] or music tagging [4] among others. The most simple definition for this task is audio tagging in which a classifier aims to identify the active sounds in a clip, given a set of classes it has been trained to recognize. The effectiveness of supervised machine learning heavily depends on the availability of good quality and extensive labeled datasets. A way to establish a good quality of the data is to have an expert annotate it carefully. At the same time, having a unique expert annotating everything will, in practice, teach the classifier to behave like this specific expert. However, some experts may disagree on the categories in the data, which creates the problem of establishing the ground truth based on multiple expert opinions, which is a time-consuming and expensive way to annotate data, and brings the additional problem of finding the common ground.

A simple and often used alternative is to have multiple annotators that are not necessarily experts on the task [5]. By using the knowledge of crowds it is possible to dispose of the experts, reducing the cost. Applying the same principle, to reduce annotation time, each non-expert annotator sees a subset of the data, which results in an sparse annotation, where a single annotator does not have to see all the data, but still each instance is annotated by more than one annotator. To obtain large amounts of annotated data, crowdsourcing has been used as a convenient solution [6, 7], despite its obvious drawbacks of uncontrolled data quality.

Several works in different domains have attempted to study how to best utilize the information and learn from multiple annotators. In [8] the competence of a large pool of annotators, who partially annotate the same data, is estimated. The method, called MACE - Multi-Annotator Competence Estimation, uses an unsupervised model that learns from redundant information and is able to identify the trustworthy annotators and predict the correct underlying labels. The drawback of the method is that needs a specific data structure and careful selection of parameters. In [9] authors proposed selection of an optimal subset of annotators from a pool of workers. They studies three real-world datasets: question-answering dataset; disambiguity dataset and image dataset. However, in the case of a large number of annotators, the computational demands of such combinatorial procedures are notably high. A simpler approach is to allow multiple annotators to verify and make corrections of previously annotated data, although not always successfully; for example, in [10] the authors mention that even with five curation stages there was almost never a consensus among annotators.

Selecting subsets or aggregating opinions requires a preprocessing step controlled by design choices and parameters which entangle the interpretation of the final results for the task at hand. In previous work, we used MACE to aggregate annotator opinions for crowdsourced audio tags [11], and observed that it produces a larger amount of labels than the majority vote approach.

In this work, we perform a systematic study of how the deep learning model itself can cope with the multiple opinions instead of providing it with the single, aggregated, label per training item, for the task of audio tagging. We include both simple and state-of-the-art architectures trained for audio tagging, and investigate if aggregation brings any advantage in training. We follow the setup of the crowd layer proposed by Rodriges et. al. [12], a fully-connected layer that learns from the crowd. The authors show how this approach works for multiple tasks, e.g. binary classification, multi-class classification and regression. In [13], the authors model individual annotators, weighting them differently based on the experts reliability in a network, "doctor net", modeling medical doctors. However, Rodriges et.al. show that the crowd layer outperforms the "doctor net" approach, albeit on a different dataset collected using MTurk. Here we investigate the effect on performance of the crowd layer, in addition to training with labels generated by MACE, and training directly with the raw data.

The paper is organized as follows: Section 2 introduces the multi-annotator dataset and explains the crowd layer implementation and how it is used in this work; Section 3 presents the audio tagging systems tested and introduces the combinations of aggre-

---

This work was supported by Academy of Finland grant 332063 "Teaching machines to listen". The authors wish to thank CSC-IT Centre of Science Ltd., Finland, for providing computational resources.

gation considered; it also includes an analysis of the results, and discussion of the benefits of using a crowd layer instead of label aggregation methods; finally, Section 4 presents conclusions and future work.

2. LEARNING FROM MULTI-ANNOTATOR DATA

In [11] we presented a study of annotator and annotations reliability for crowd-sourced audio tags for real-life acoustic scenes[1]. We showed that the aggregation of the multi-annotator labels using annotator competence estimation and true label prediction through MACE produces a plausible and trustable ground truth. We observed that by gradually eliminating the less trustworthy annotators from the aggregation, the level of inter-annotator agreement in the predicted aggregated labels gradually improved. Nonetheless, discarding annotators should be limited to the outliers only, in order to retain as much information and opinions as possible.

When annotating real-world data and aggregating the information, it is not possible to evaluate the correctness of the resulting labels. However, we conducted a subsequent study that included synthetic data, and observed that the labels produced through MACE aggregation are faithfully representing the ground truth [14], with an 86% F-score, (97% precision and 77% recall), much better than the typical majority vote aggregation (68% F-score, with 98% precision, 52% recall). We therefore consider the labels produced using MACE a sufficiently accurate representation of reality, and use them as reference in the evaluation of the classifiers.

2.1. Dataset

The dataset used in our experiments is the MATS (Multi-Annotator Tagged Soundscapes) data, published with the study in [11]. It is a subset of TAU Urban Acoustic Scenes 2019 [15], consisting of audio from three acoustic scenes (airport, public square, and park). The audio clips are 10 seconds long, and some of them are consecutive segments of one long recording from a single location. A total of 133 annotators, students taking an audio signal processing course, annotated a randomly assigned set of 131 files each, such that each audio clip was annotated by five different annotators. The complete details about the data annotation process and its postprocessing is explained in [11]. The unbalanced nature of the MATS dataset can be observed based on the numbers from Table 1, with the most dominant sounds in the data being related to human presence and traffic.

For the experiments, we partition the data into training, validation and test sets following the DCASE 2019 Task 1 split that respects the location ID of the original recordings, ensuring that all clips of the same long recording are placed into one single subset (training, validation or test). The partitioning results in sets containing 1772 clips for training; 762 for validation and 1099 for test.

2.2. Crowd layer

A general-purpose crowd layer was proposed in [12], which allows training of neural networks directly using the labels produced by multiple annotators. The authors showed that the model is able to capture the reliability and biases of different annotators, achieving state-of-the-art results for three different tasks. In this study, we used the author's code[2] and adapted it from TensorFlow to PyTorch.

class labels	MACE	majority vote
adults talking	2728	2190
footsteps	1853	828
traffic noise	1580	634
birds singing	979	648
children voices	917	446
music	152	69
announcement/speech	148	73
siren	98	37
dog barking	84	25
announcement/jingle	35	8

Table 1: Statistics of class labels in the data used for experiments resulting from combining the multiple annotations

We use the PaSST model [16] and extend it with the crowd layer for the purpose of our study. The PaSST architecture is first extended with a fully-connected layer for the multilabel classification of the ten sound classes. Then the crowd layer is added as the very last layer, having as inputs the actual classification layer. The crowd layer learns to map the probabilities of the classification layer to the raw labels, assumed to being capable of capturing the bias and reliabilities of the annotators. The classification layer of the network becomes a shared layer among the annotators, a bottleneck that during training receives adjusted gradients from the different opinions, aggregates them and backpropagates to the rest of the network.

Given the output of a model denoted as σ, the activation of the crowd layer for each of the annotators r can be defined as $\mathbf{a}_r = f_r(\sigma)$, f_r being the annotator-mapping function. The original publication proposes a few different implementations of the annotator-mapping function, ranging from a matrix function with per-class biases to a single vector function without bias. In this work, we considered the more simplistic implementation, and use the linear transformation of the input, without per-class bias. The layer is defined in the following equation:

$$f_r(\sigma) = \mathbf{w}^r \odot \sigma, \tag{1}$$

where $\mathbf{w}$ is the annotator specific vector. The raw annotation is sparse, with only five opinions per clip in a large pool of annotators, therefore it is not necessary to propagate information from all outputs; a mask is used to set to zero the gradient contributions of the missing labels (corresponding to annotators that did not provide an opinion to the current clip).

Two different scenarios involving PaSST models are used: one that uses PaSST only to produce embeddings, which are used as input of a simpler model with a fully connected layer; and another one in which the weights of the entire PaSST architecture are fine-tuned during training. Once the model has been trained, the crowd layer is removed, and the remaining architecture is used as a classifier on the test set, with the weights of the model expected to have learned the true distribution of the classes. Note that, to evaluate the model performance, the labels of the test set were processed using MACE, as described in the previous subsection.

---

[1] The MATS (Multi-Annotator Tagged Soundscapes) dataset is available at https://doi.org/10.5281/zenodo.4774959

[2] https://github.com/fmpr/CrowdLayer

Model	setup	MACE			Majority Vote		
		Macro-F1	Micro-F1	mAP (95% CI)	Macro-F1	Micro-F1	mAP (95% CI)
mel_CNN	baseline	32.51%	72.40%	0.41 (0.39, 0.43)	26.95%	63.94%	0.30 (0.29, 0.32)
	raw	29.13%	65.60%	0.40 (0.38, 0.41)	26.93%	63.03%	0.29 (0.27, 0.30)
	crowd	35.13%	73.99%	0.41 (0.39, 0.43)	30.33%	62.06%	0.31 (0.27, 0.35)
PaSST_emb	baseline	47.03%	79.51%	0.61 (0.56, 0.65)	51.37%	69.75%	0.63 (0.50, 0.77)
	raw	38.73%	64.14%	0.61 (0.57, 0.65)	47.02%	72.87%	0.59 (0.54, 0.65)
	weighed	46.18%	71.79%	0.60 (0.56, 0.64)	49.15%	71.01%	0.58 (0.53, 0.64)
	crowd	51.42%	**80.38**%	0.62 (0.58, 0.67)	60.12%	69.77%	0.65 (0.52, 0.79)
PaSST	baseline	45.73%	79.42%	0.67 (0.62, 0.71)	51.28%	73.68%	0.68 (0.55, 0.81)
	crowd	**53.15%**	77.19%	**0.69** (0.65, 0.74)	**63.21%**	**73.79**%	**0.68** (0.58, 0.78)

Table 2: Comparison of the different considered setups, evaluated against MACE and majority vote aggregated reference.

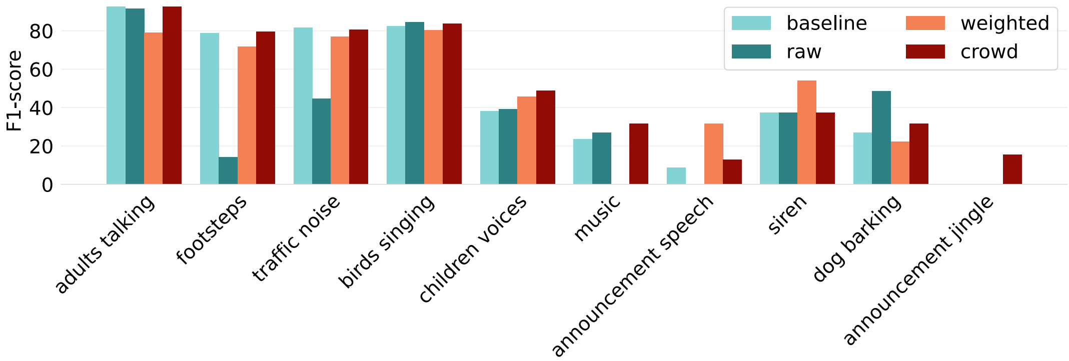


Figure 1: Class-wise F1-score comparison for the PaSST_emb systems with different training setups, evaluated against MACE labels.

2.3. Baseline systems

To evaluate the suitability of the crowd layer we test it under different conditions, in order to observe its effect independently of the used model and features. We use as a baseline system a CNN with three convolutional layers, each followed by batch normalization and ReLu activation layer, denoted as mel_CNN. This system is a typical multiclass classification system having ten output neurons (for the ten classes to be classified). It is trained using the MACE aggregated labels as targets, and a feature representation consisting of mel energies calculated using a window size of 2048 samples with a hop length equivalent to 20 ms, and 64 mel filter banks, with the lower and upper frequencies set to 50 and 14kHz. We train the same system with the raw labels, by considering each clip as an independent data point which we provide to the network with the labels provided by one annotator. In practice, this means that one audio clip is provided to the network five times, with five label sets as available from the annotators pool. We denote this training setup as "raw". The same architecture is also trained using the crowd layer, hence denoted by "crowd". We use similar baselines also for the PaSST architecture, indicated as PaSST_emb (using PaSST only to produce the embeddings feature representation) and PaSST (full training of the entire network).

3. EXPERIMENTAL SETUP AND RESULTS

The evaluation of the systems is done by calculating standard audio tagging metrics. The macro-average and micro-average metrics (Precision, Recall and F1-score) and the Mean Average Precision (mAP) are calculated for each system against the reference labels obtained using MACE and against a second set of reference labels obtained using majority vote. The results are presented in Table 2 and include the 95% confidence interval for mAP, calculated using the jackknife estimation method.

3.1. Aggregate or separate: performance evaluation

The baseline system mel_CNN obtains the lowest performance when trained with the aggregated target labels, among the three training setups. Its performance decreases considerably in terms of F1-score when training with the raw data, indicating that the training pairs likely contain incorrect or contradictory labels which are presented as targets to the same audio clip, creating fluctuations in the loss function. On the other hand, the crowd layer successfully uses the redundant information to correct for the labeling errors, noticeable in particular in the macro-F1 performance; micro-average F1-score and mAP do not change significantly, which seems to indicate better performance for minority classes. The trend is seen in both evaluation procedures, though, based on our experience and previous work, we trust more the MACE labels as a reference.

Using the PaSST embeddings with the aggregated targets brings a considerable improvement in performance, which is further increased when using the crowd layer. Similar to the simple CNN, using raw data in the training is detrimental, while the crowd layer brings a significant boost to the class-wise scores. As an additional experiment, we investigate the use of annotator competence as ad-

Model	MACE P	MACE R	MACE F1	Majority Vote P	Majority Vote R	Majority Vote F1
PaSST_emb baseline	80.61%	49.91%	57.98%	56.97%	66.33%	58.50%
PaSST_emb crowd	89.97%	54.43%	62.57%	66.50%	74.30%	64.64%
PaSST baseline	95.60%	**55.32%**	66.25%	64.42%	74.51%	67.06%
PaSST crowd	94.81%	53.44%	66.82%	66.04%	**76.37%**	70.20%

Table 3: Macro-averaged metrics calculated for the training data for PaSST architectures, with and without the crowd layer .

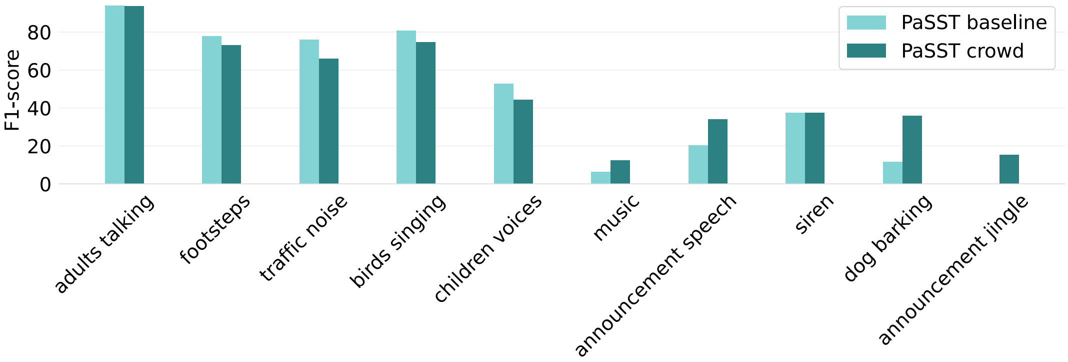


Figure 2: Class-wise F1-score comparison for the PaSST systems in baseline and crowd layer training setups, evaluated against MACE labels.

ditional information in the raw data training setup. Here, we calculate the competence estimates for each annotator from the training data using MACE, and multiply the network target vector (binary indicators of class presence) of each annotator by its competence. This results in a weighted target information, which can be seen as a form of data augmentation. The method, denoted as "weighted" in Table 2, brings a significant improvement compared to the raw labels training scenario, but does not outperform other setups.

A detailed analysis of the class-wise F1-scores can be seen in Figure 1: only the system trained with the crowd layer is able to identify all 10 classes. The performance is quite similar between baseline and crowd layer setups for the classes with higher number of examples, but the less represented classes show large fluctuation depending on the training setup. Here we can observe the advantage of using the competence-weighted augmentation, which is beneficial for the *announcement speech* and *siren* classes, but inconsistent over the entire set of classes. The smallest class, *announcement jingle*, is only detected in the crowd layer training case.

The best macro F1-score and MAP among all experimental setups is obtained with the fine-tuning of the entire PaSST architecture, including the classification layer and the crowd layer. In this setup, the model is initialized with the pretrained weights and trained for 30 epochs with the MATS data. Note that micro-F1 is higher when evaluated against the majority vote reference, which contains a lesser amount of labels, according to [14]. An illustration of the class-wise F1-scores is shown in Fig. 2, with classes arranged in order of their size. We can clearly see that the crowd layer network has better performance for the under-represented classes, even though the MACE aggregation is designed to override the majority vote result if a minority of the annotators are highly reliable [8]. This shows that no matter how sophisticated aggregation method is used, the loss of information from the separate labels to the aggregate ones may have a significant effect on the task where the data under discussion are used.

3.2. Learning distributions

To study how the crowd layer learns the label distribution, we calculate the macro-average metrics against the training data for the PaSST systems (baseline and crowd training setup). The results, presented in Table 3, show that the crowd layer helps the network learn to mimic somewhat the distribution of the MACE labels, more in the setup that uses embeddings. Continuing to train the whole network instead of just extract embeddings from the pretrained network is, as expected, a better way to learn the distribution of the training data. Moreover, the very similar values of the metrics show that the crowd layer does not lead to overfitting either. When classifying the training data, the scores against the majority vote aggregates are generally better than against the MACE aggregates, but this is due to the smaller amount of labels to compare, which is reflected in a high recall. On the other hand, the precision of the models is considerably higher for the MACE aggregation, showing more robustness of the model in its predictions.

4. CONCLUSIONS

Performance of supervised models rely on the quality of the annotated data, which can be obtained from multiple annotators to avoid bias and leverage information from multiple annotators. In this work, we investigated different methods to use the multiple opinions, training different audio classifiers with aggregated or separate labels. In our experiments, letting the model learn from multiple annotators using a simple crowd layer had the best performance. By adding this linear transformation to the model, we can avoid the manual intervention into the dataset, and remove the influence of the aggregation method on the model performance. However, a question remains on the scalability of the approach, with extreme combinations like binary classification (single neuron) and large number of annotators (e.g. thousands) requiring closer examination.

5. REFERENCES

[1] S. Ntalampiras, "Adversarial attacks against audio surveillance systems," in *2022 30th European Signal Processing Conference (EUSIPCO)*, 2022, pp. 284–288.

[2] E. Vidaña-Vila, J. Navarro, D. Stowell, and R. M. Alsina-Pagès, "Multilabel acoustic event classification using real-world urban data and physical redundancy of sensors," *Sensors*, vol. 21, no. 22, 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/22/7470

[3] S. Yu, Y. Ding, K. Qian, B. Hu, W. Li, and B. W. Schuller, "A glance-and-gaze network for respiratory sound classification," in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 9007–9011.

[4] J. Pons and X. Serra, "musicnn: Pre-trained convolutional neural networks for music audio tagging," 2019.

[5] V. C. Raykar, S. Yu, L. H. Zhao, G. H. Valadez, C. Florin, L. Bogoni, and L. Moy, "Learning from crowds," *Journal of Machine Learning Research*, vol. 11, no. 43, pp. 1297–1322, 2010. [Online]. Available: http://jmlr.org/papers/v11/raykar10a.html

[6] M. Cartwright, G. Dove, A. E. Méndez Méndez, J. P. Bello, and O. Nov, "Crowdsourcing multi-label audio annotation tasks with citizen scientists," in *Proceedings of the 2019 Conference on Human Factors in Computing Systems*, ser. CHI '19. NY, USA: Association for Computing Machinery, 2019, p. 1–11.

[7] E. Fonseca, M. Plakal, D. P. W. Ellis, F. Font, X. Favory, and X. Serra, "Learning sound event classifiers from web audio with noisy labels," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 21–25.

[8] D. Hovy, T. Berg-Kirkpatrick, A. Vaswani, and E. Hovy, "Learning whom to trust with MACE," in *Conf. NAACL HLT*. Atlanta, Georgia: Association for Computational Linguistics, jun 2013, pp. 1120–1130.

[9] H. Li and Q. Liu, "Cheaper and better: Selecting good workers for crowdsourcing," in *AAAI Conference on Human Computation & Crowdsourcing*, 2015.

[10] S. Hershey, D. P. W. Ellis, E. Fonseca, A. Jansen, C. Liu, R. C. Moore, and M. Plakal, "The benefit of temporally-strong labels in audio event classification," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021.

[11] I. Martín-Morató and A. Mesaros, "What is the ground truth? reliability of multi-annotator data for audio tagging," in *29th European Signal Processing Conference 2019 (EUSIPCO 2019)*, 2021.

[12] F. Rodrigues and F. C. Pereira, "Deep learning from crowds," ser. AAAI'18/IAAI'18/EAAI'18. AAAI Press, 2018.

[13] M. Y. Guan, V. Gulshan, A. M. Dai, and G. E. Hinton, "Who said what: Modeling individual labelers improves classification," in *AAAI Conference on Artificial Intelligence*, 2017.

[14] I. Martín-Morató, M. Harju, and A. Mesaros, "Crowdsourcing strong labels for sound event detection," in *2021 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2021, pp. 246–250.

[15] A. Mesaros, T. Heittola, and T. Virtanen, "Acoustic scene classification in DCASE 2019 challenge: closed and open set classification and data mismatch setups," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2019 Workshop (DCASE2019)*, New York, Nov 2019.

[16] K. Koutini, J. Schlüter, H. Eghbal-zadeh, and G. Widmer, "Efficient Training of Audio Transformers with Patchout," in *Proc. Interspeech 2022*, 2022, pp. 2753–2757.

ACTIVE LEARNING IN SOUND-BASED BEARING FAULT DETECTION

Maarten Meire^{1,2,3}, *Jeroen Zegers*^{4}, *Peter Karsmakers*^{1,2,3}

^{1} KU Leuven, Dept. of Computer Science, ADVISE-DTAI, Kleinhoefstraat 4, B-2440 Geel, Belgium
{maarten.meire, peter.karsmakers}@kuleuven.be
^{2} Leuven.AI - KU Leuven institute for AI
^{3} Flanders Make @ KU Leuven
^{4} Flanders Make vzw, CoreLab MotionS, 3001 Leuven, Belgium

ABSTRACT

Sound Event Classification (SEC) for fault detection of bearings in rotating machinery has recently shown good results. Bearing fault detection via microphones has advantages over the more traditional accelerometer-based solutions, in terms of ease of sensor deployment, non-intrusiveness and hardware cost. These novel SEC methods often use deep learning (DL), which require large amounts of labeled data. As events of faulty bearings are rare in practical scenarios, it can be time consuming to manually find and label examples of faults. Rather than labeling a complete dataset, active learning (AL) methods present the expert labeler with unlabeled samples that are expected to be the most informative in the learning process. This way the most interesting samples are labeled first, which allows to only annotate a subset of the dataset, while still retaining (close to) maximal accuracy. In this work a novel data set, that contains acoustic data from accelerated life time tests for bearings, is used to investigate the performance of two AL methods in terms of classification accuracy and number of additionally selected and annotated examples.

Index Terms— Active learning, Fault detection, Bearing monitoring, Transfer learning

1. INTRODUCTION

An important part in industrial applications is rotating machinery. Rolling Element Bearings (REB) are a common element in this machinery and most system failures can be attributed to these REB [1]. It is important to detect faults in these REB to prevent critical failures and this is most commonly done based on vibration analysis [2], however research has also been done towards using sound signals for REB fault detection [3]. Various data-driven approaches, usually Deep Learning (DL) based, have been investigated for the purpose of REB fault detection in the last few years. As these approaches are data-driven, and often based on DL, large amounts of data are required to train the associated models. For vibration analysis acquiring this data requires an accelerometer to be attached directly to the REB, which is not always trivial, especially in complex machines. By using a microphone this can be done without needing direct contact, making the data acquisition process easier, and it has been shown that using sound signals for fault detection is a promising alternative to vibration analysis [4, 5, 6, 7]. However, even if a lot of data can be acquired, the process of annotating this data remains time and cost intensive. Active Learning (AL) methods have been developed to reduce this cost by only annotating samples that are the most informative for learning algorithms [8, 9]. The focus of this work will be to use sound signals captured by a microphone in combination with AL methods for fault detection in REB.

In literature AL methods have already been used for fault detection in industrial applications. In [10] an extension of the entropy measure of model uncertainty was used to select the most informative samples to train a model that was learned on a data set with isolated and compound faults for REB fault detection. A combination of entropy and complexity was used in [11] to select samples for fault diagnosis in a gearbox showing a better performance using this combination. AL was applied to cellular networks in [12], with a comparison of 3 uncertainty based sampling methods, demonstrating their effectiveness. In [13] it is mentioned that using a single criterion strategy might not be stable and a new criterion is proposed that combines multiple commonly used criteria. A best versus second best uncertainty metric was used in [14] in combination with label propagation and ensembles to improve the performance of bearing fault diagnosis using a small training set.

The previously discussed works use vibration signals as data. However, as mentioned earlier, this work will focus on using sound signals. To the best of our knowledge, in the literature no prior work regarding AL for REB fault detection using sound is found. Nonetheless, AL in combination with sound has shown promising results in other fields. In [15] AL methods were evaluated using 2 synthetic sound event datasets for sound event detection and it was shown that training while keeping the original training set along with the annotated samples resulted in a better performance. A combination of AL and semi-supervised learning methods was used in [16] on a total of 3 datasets containing sound data for gender identification, speaker identification, and emotion detection. In [17] an alternating certainty sampling method was proposed where sometimes samples with high confidence were chosen instead of low confidence to improve the robustness against incorrect annotations. This method was evaluated on an urban sound dataset.

To compare AL methods a novel and unique accelerated bearing life time test dataset is used. It contains data captured using an accelerometer and 2 microphones. To the best of our knowledge, there is no public dataset that contains sound signals from bearing life time tests, as there are for vibration signals, e.g. IMS [18].

The rest of this paper is structured as follows. In Section 2 the AL methods that will be compared are explained. A detailed description of the experimental setup is given in Section 3, this includes the dataset, the preprocessing, the architecture and learning parameters of the models, and a description of the performed experiments. The results of the experiments are discussed in Section 4. Finally, conclusions and future work are given in Section 5.

2. ACTIVE LEARNING

In this section the (AL) methods that will be used in the experiments are discussed.

2.1. Uncertainty sampling

The first method, which is commonly used for AL, is based on selecting the samples for which the model predictions are most uncertain. For a classifier this often means that these samples are located close to the decision boundary. A simple yet commonly used metric to quantify predictive uncertainty is the information entropy,

$$H(X) = -\sum_{i=1}^{k} p_i log(p_i), \tag{1}$$

where X is the sample under evaluation, k is the amount of classes, and p_i is the estimated posterior probability for the $i-th$ class as predicted by a classifier. In this work a Convolutional Neural Network (CNN) is used to provide the probabilities for each class, the model will be described in Section 3.3.

2.2. Hybrid sampling

To avoid sampling multiple similar samples that have high prediction uncertainty the sampling criterion can be augmented with a novelty metric. The latter is referred to as hybrid sampling. In this way there is a potential to further decrease the annotation cost [8]. The Semi-Supervised Detection of Outliers (SSDO) [19] algorithm is used to calculate the novelty metric. This algorithm is based on k-means, but does not only take into account the distance to a cluster center, but also the size and relative position of the considered cluster. Formally, the hybrid sampling strategy is defined as follows:

$$S(X) = H(X) + \alpha N(X), \tag{2}$$

where $N(X)$ is the score obtained by SSDO for a sample X and α is a hyperparameter that balances both individual scores. Both scores have ranges $H(X) \in [0,1]$ and $N(X) \in [0,1]$. The sample X with the highest $S(X)$ is then selected as the most interesting.

3. EXPERIMENTAL SETUP

This section describes the dataset that was used in the experiments, as well as the preprocessing that was performed on the data, the model architecture and associated learning parameters that were used, and finally the performed experiments.

3.1. Dataset

This work uses a novel data set, collected by Flanders Make, that consists of data collected from multiple accelerated bearing life time tests. Data was collected using an accelerometer, a microphone inside the safety cover of a setup, and a microphone outside the cover, hereafter called the internal and external microphone, respectively. The setup can be seen in Figure 1, with the placement of both the internal and external microphone. The accelerometer was attached directly to the bearing housing. All sensors captured data with a sampling frequency of 50 kHz. A total of 64 accelerated life time tests, or runs, were performed. For each bearing a small indent was created on the inner races (IR) using a Rockwell-C indenter. The lifetime was further accelerated by applying a radial load of 9 kN. The test was stopped when either the stopping criteria of 20g peak vibrations was reached or the test had to be stopped due to safety concerns, e.g. overheating. The life time tests were performed with varying settings, e.g. fixed or varying rpm, and other setups running next to the test setup. It was also determined that the various life time tests not only resulted in inner race faults, but also in outer race and ball faults or in some cases it was considered as not faulty. The experiments in this work use a subset of this dataset, more specifically the data captured by the external microphone from only the runs with an inner race fault, with no additional running setups, and a fixed rpm during the run. Note that this rpm can vary between runs, e.g. 1800 rpm for one run and 2000 rpm for another. In total the considered subset contains 10 runs that match these criteria. The same setup was already used in previous works for accelerometer based fault detection [20].

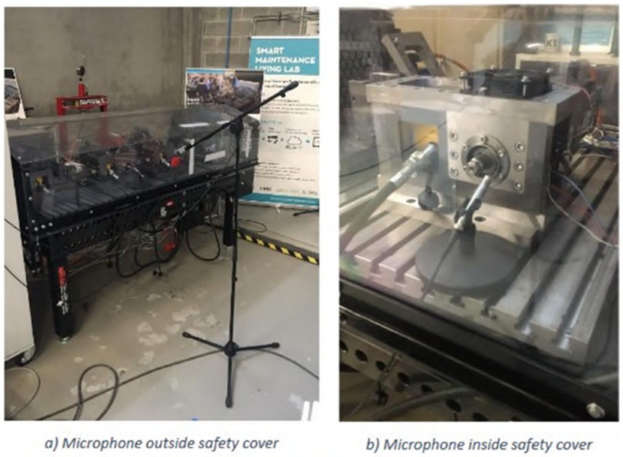

a) Microphone outside safety cover b) Microphone inside safety cover

Figure 1: The microphone setup used in the bearing life time tests.

In addition to the captured data, two sets of ground truth labelling are also provided. It should be noted here that this labelling is not based directly on the physical state of the bearing, but based on analysis of the data captured by either the accelerometer or the internal microphone. Using this labelling the moment in time p_f where the bearing starts having faulty behavior is determined. Data prior to p_f is then considered as healthy and data after p_f is considered faulty. As there are two sets of labelling, p_f is determined separately for each.

3.2. Preprocessing

In this work the raw audio data is first transformed to log mel spectra. This transformation is done using a window and hop size of 1s. A total of 64 mel filterbanks are then extracted. This leads to an input frame with shape (64,10), as each audio fragment is 10s long, which can then be passed to the models. As the data consists of multiple different runs, each run is separately standardized, using a running mean and standard deviation, to have, approximately, zero mean and unit variance. For the CNN the input frames are used directly, while for SSDO the mean and standard deviation for each filterbank are calculated over 10s and then stacked, resulting in a 128 dimensional feature vector.

After this preprocessing the data is split into 3 parts: 1) a training run that will be used to train an initial model, 2) a sampling set that will be used to sample points from, and 3) a test run that will be used to assess the generalization performance of the model. This split is done in a leave-one-run-out scheme, meaning that there will be 10 folds, as there are 10 available runs, with a single run in each test set. From the remaining 9 runs one is chosen as the training run and the other 8 are used as the sampling set. In this training run data is taken so that the amount of healthy and faulty samples is roughly equal. More specifically, this is done by taking all the data after p_f and taking the same amount of data directly prior to p_f. Then a maximum of 100s of data from the start of the run is also added. From the training run 20% will be used as validation.

3.3. Model architecture and learning parameters

The CNN model used in this work consists of 3 convolutional blocks, using 64, 64, and 32 filters, respectively, 2 fully connected blocks, both using 20 neurons, and a final fully connected layer as output. A single convolutional block is a sequence of a convolutional, batch normalization [21], maxpooling, and dropout [22] layer. The fully connected block contains the same sequence without the maxpooling. The leaky ReLu activation function was used for all layers, except the final layer, which uses a softmax activation. The filters in the convolutional layers are all size (7,7) and move with a stride of 1 in each direction. The maxpooling layers use a (2,2) window and move with a stride of 2 in each direction. All dropout layers use the default drop rate of 0.5.

A model was trained using the data described in Section 3.2 to serve as starting point for the AL methods. This model was trained for 100 epochs with the Adam optimizer [23] and a learning rate of $1e^{-3}$. If the validation loss did not improve for 10 consecutive epochs, the learning rate was halved. The weights were further regularized by the L_2 norm with a factor (λ) of $5e^{-6}$. All hyperparameters were empirically tuned independently from the test data.

The SSDO model, used for the hybrid sampling, was fitted using a contamination factor of $1e-3$, meaning that 0.1% of the training data is considered as novel. The amount of clusters used by the algorithm is set to 5% of the amount of training data.

3.4. Experiments

In this work a comparison of AL methods will be made to investigate the model performance on an independent test set in terms of the employed number of additionally annotated examples. For this purpose, the same experiment was repeated twice, once with labels based on information from the internal microphone (Y_{Mic}) and once with labels based on the accelerometer (Y_{Acc}).

3 methods were compared to each other: 1) random sampling, where samples are chosen at random to annotate, this will serve as a baseline method , 2) uncertainty sampling, which uses the prediction probabilities of the CNN to determine what samples to annotate, as described in Section 2.1, and 3) hybrid sampling, which further incorporates a novelty metric, as described in Section 2.2. To evaluate the methods, first an initial CNN model was trained on the data from a single run that is available in the training partition, as described in Section 3.3. Next a first sample is selected for annotation using the considered sampling strategy. After the annotation, the sample was added to the training set and the CNN model was updated for 20 epochs and, if the hybrid sampling method was being used, the SSDO method was refitted. This process was repeated 200 times for each sampling method.

To quantify the performance the F1 score was used as a metric. The faulty class is considered to have a positive label. As the leave-one-run-out scheme was used, the mean and standard deviation of this metric are computed over the folds. However, it was noticed that the standard deviation was similar across the results, ranging from 0.15 to 0.2, hence it will not be shown for reasons of clarity.

4. RESULTS

In this section first the results of the experiment where the Y_{Mic} were used will be discussed. Thereafter, the same experiment is repeated but this time with Y_{Acc}. For both experiments the F1 score in terms of the additionally annotated samples, ranging from 0 to 200, will be tracked. As a baseline, the F1 score of a model trained using the full set of training samples (on average around 15000 samples) for 100 epochs was added.

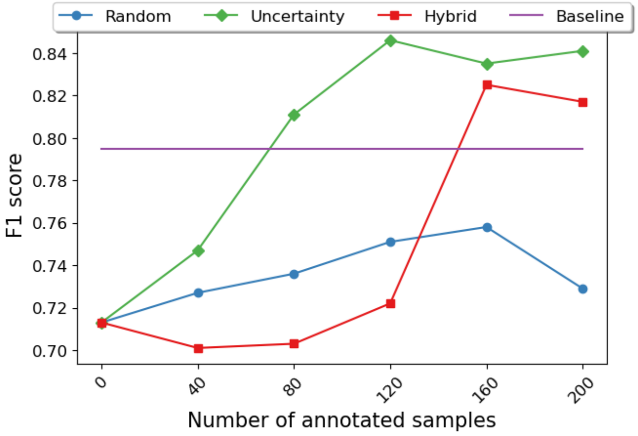


Figure 2: The F1 score comparison of the considered AL methods and the baseline on the test set in steps of 40 annotated samples using Y_{Mic}.

4.1. Microphone labels

The F1 scores attained by the CNN model on the test set using the AL methods described earlier and Y_{Mic} are shown in Figure 2. It can be seen that, while hybrid sampling does not perform as well with low amounts of annotated samples, both uncertainty and hybrid sampling outperform both random sampling and the baseline when respectively 80 and 160 samples are additionally annotated. This indicates that by using AL the amount of samples that need to be annotated can be significantly reduced, in this work by around 75 times or more. The difference between random sampling and uncertainty and hybrid sampling can likely be attributed to the sample selection. By inspecting these samples it can be seen that data around p_f for the various runs in the sampling set is chosen significantly more for uncertainty sampling, and to a lesser extent for hybrid sampling, while random sampling follows a more even distribution across the entire set, as is to be expected. This is empirically verified and will be discussed in Section 4.3. By choosing samples around p_f the model can learn a boundary between what is healthy and faulty. However, as there is a domain shift to the unknown bearing, it is expected that, while the boundary is likely to be improved, it will not be a perfect match. The lower performance of hybrid sampling up to 120 samples, could potentially be due to a smaller similarity between chosen samples increasing the complexity of the data in comparison to the other methods.

4.2. Accelerometer labels

The results attained on the test set when using Y_{Acc} are shown in Figure 3. It can be seen that these results are similar to the results attained using Y_{Mic}, with increasing F1 scores when more samples are annotated. Uncertainty sampling also slightly surpasses the baseline with 80 annotated samples. However, it does stagnate, and even performs slightly worse, afterwards. The difference between the 3 methods is smaller compared to the labels based on the microphone, especially towards higher amounts of annotated samples. It can be seen that hybrid sampling once again performs worse with

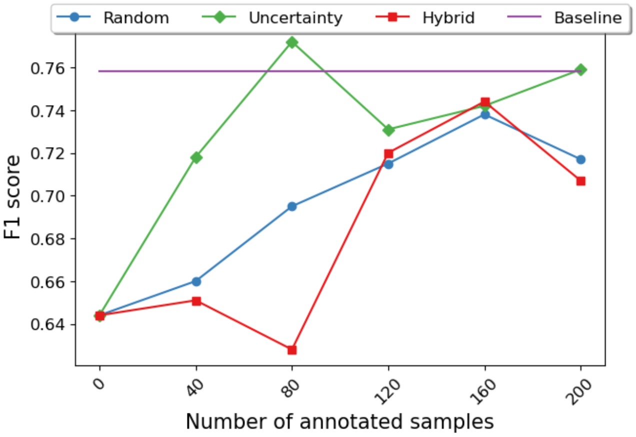


Figure 3: The F1 score comparison of the considered AL methods and the baseline on the test set in steps of 40 additionally annotated samples using Y_{Acc}.

up to 80 annotated samples. This is likely due to the same reasons as explained earlier. Additionally, as the accelerometer is expected to detect changes earlier, and thus p_f for Y_{Acc} is slightly earlier than p_f for Y_{Mcc}, the samples around p_f have an increased similarity, which will cause less of these samples to be chosen, as they will have a lower novelty metric. This possibly results in a too rough decision boundary with a worse performance as a consequence. The overall worse performance can likely also be attributed to the difference in p_f and corresponding labels. With Y_{Acc} the distribution of the healthy and faulty data is expected to overlap more. From the moment the accelerometer signals are starting to change the sound signals might still be very similar. This makes the problem more difficult with worse performance as a consequence.

4.3. Sample selection

As mentioned earlier, it was noticed that uncertainty, and to a lesser extent hybrid, sampling selected a significant amount of samples around p_f of a run. To illustrate this, an experiment was performed where only a single run was available in the sampling set and 50 samples, selected using uncertainty and hybrid sampling, were annotated following the same process as described in Section 3.4. The log mel spectrum of the specific run and a histogram of the selected samples is shown in Figure 4. It can be seen that uncertainty sampling selected 28 samples within 100 samples of p_f while 20 were selected, within the same group of samples, by hybrid sampling, indicating that indeed a significant amount of samples are selected around p_f by uncertainty, and to a lesser extent hybrid, sampling. Furthermore, it is indicated that samples are also selected with a larger selection by hybrid sampling, around noise events that pop-up in the healthy data, e.g. around 5000s or 500 samples, which would cause the model to learn the data is healthy, regardless of the disturbing noise events. This could have also contributed to the improved performance of AL compared to random sampling.

5. CONCLUSION AND FUTURE WORK

In this work we compared two AL methods, more specifically uncertainty and hybrid sampling, and a random sampling baseline method to evaluate the performance with regards to generalization to an unknown bearing when additional samples from known bearings are annotated. This was done using a novel dataset that contains accelerated bearing life time tests with data captured from an accelerometer, a microphone inside a safety cover, and a microphone outside the safety cover. Labels were provided based on the microphone inside the safety cover and based on the accelerometer.

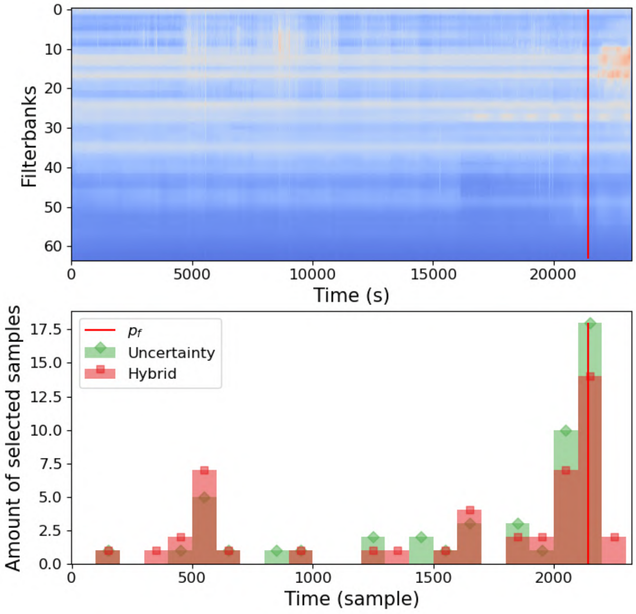


Figure 4: The log mel spectrum of the sampling run (top) and a histogram, with bins of 100 samples, of the selected samples (bottom) by uncertainty and hybrid sampling. The red line indicates p_f.

It is indicated that, for the labels based on the microphone inside the safety cover, both AL methods outperform random sampling and also outperform the baseline that uses all data. Furthermore, the uncertainty sampling method does show a better performance compared to hybrid sampling. This is likely due to more samples being selected in the close vicinity of p_f. When looking at results for the labels based on the accelerometer, the difference between the methods is not as clear. However, uncertainty sampling still shows the best performance, also attaining a score similar to the baseline. The hybrid sampling method does not perform as well, likely due to smaller novelty metric between points near p_f, as the accelerometer can detect the fault earlier than the microphone. The results on both sets of labels indicate that it is possible to attain a similar, or better, performance to a method that does not use AL, while the amount of annotated samples was reduced by a factor of around 75.

In future research we will include different faults, e.g. outer race faults, into the experiments. We will also investigate the combination of label propagation with the AL methods.

6. ACKNOWLEDGMENT

The authors would like to thank the ACMON ICON project for funding the research presented in this paper. This research has received support of Flanders Make, the strategic research centre for the manufacturing industry. This research received funding from the Flemish Government (AI Research Program).

7. REFERENCES

[1] A. Nabhan, N. Ghazaly, A. Samy, and M. M.O, "Bearing fault detection techniques - a review," *Turkish Journal of Engineering, Sciences and Technology*, vol. 3, 01 2015.

[2] D.-T. Hoang and H.-J. Kang, "A survey on deep learning based bearing fault diagnosis," *Neurocomputing*, vol. 335, pp. 327 – 335, 2019.

[3] M. Altaf, M. Uzair, M. Naeem, A. Ahmad, S. Badshah, J. A. Shah, and A. Anjum, "Automatic and efficient fault detection in rotating machinery using sound signals," *Acoustics Australia*, vol. 47, pp. 125–139, 2019.

[4] J. Pacheco-Chérrez, J. A. Fortoul-Díaz, F. Cortés-Santacruz, L. María Aloso-Valerdi, and D. I. Ibarra-Zarate, "Bearing fault detection with vibration and acoustic signals: Comparison among different machine leaning classification methods," *Engineering Failure Analysis*, vol. 139, p. 106515, 2022.

[5] M. Iqbal and A. K. Madan, "CNC machine-bearing fault detection based on convolutional neural network using vibration and acoustic signal," *Journal of Vibration Engineering &amp Technologies*, vol. 10, no. 5, pp. 1613–1621, Mar. 2022.

[6] J. Grebenik, Y. Zhang, C. Bingham, and S. Srivastava, "Roller element bearing acoustic fault detection using smartphone and consumer microphones comparing with vibration techniques," in *2016 17th International Conference on Mechatronics - Mechatronika (ME)*, 2016, pp. 1–7.

[7] D. Xiao, C. Qin, H. Yu, Y. Huang, C. Liu, and J. Zhang, "Unsupervised machine fault diagnosis for noisy domain adaptation using marginal denoising autoencoder based on acoustic signals," *Measurement*, vol. 176, p. 109186, 2021.

[8] P. Ren, Y. Xiao, X. Chang, P.-Y. Huang, Z. Li, B. B. Gupta, X. Chen, and X. Wang, "A survey of deep active learning," *ACM Comput. Surv.*, vol. 54, no. 9, oct 2021.

[9] E. Lughofer, "Hybrid active learning for reducing the annotation effort of operators in classification systems," *Pattern Recognition*, vol. 45, no. 2, pp. 884–896, 2012.

[10] Y. Jin, C. Qin, Y. Huang, and C. Liu, "Actual bearing compound fault diagnosis based on active learning and decoupling attentional residual network," *Measurement*, vol. 173, p. 108500, 2021.

[11] J. Chen, D. Zhou, Z. Guo, J. Lin, C. Lyu, and C. Lu, "An active learning method based on uncertainty and complexity for gearbox fault diagnosis," *IEEE Access*, vol. 7, pp. 9022–9031, 2019.

[12] M. Chen, K. Zhu, R. Wang, and D. Niyato, "Active learning-based fault diagnosis in self-organizing cellular networks," *IEEE Communications Letters*, vol. 24, no. 8, pp. 1734–1737, 2020.

[13] Z. Liu, J. Zhang, X. He, Q. Zhang, G. Sun, and D. Zhou, "Fault diagnosis of rotating machinery with limited expert interaction: A multicriteria active learning approach based on broad learning system," *IEEE Transactions on Control Systems Technology*, vol. 31, no. 2, pp. 953–960, 2023.

[14] C. Jian, K. Yang, and Y. Ao, "Industrial fault diagnosis based on active learning and semi-supervised learning using small training set," *Engineering Applications of Artificial Intelligence*, vol. 104, p. 104365, 2021.

[15] Z. Shuyang, T. Heittola, and T. Virtanen, "Active learning for sound event detection," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 2895–2905, 2020.

[16] S. Karlos, C. Aridas, V. G. Kanas, and S. Kotsiantis, "Classification of acoustical signals by combining active learning strategies with semi-supervised learning schemes," *Neural Computing and Applications*, vol. 35, no. 1, pp. 3–20, Feb. 2021.

[17] Y. Wang, A. E. Mendez Mendez, M. Cartwright, and J. P. Bello, "Active learning for efficient audio annotation and classification with a large amount of unlabeled data," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 880–884.

[18] H. Qiu, J. Lee, J. Lin, and G. Yu, "Wavelet filter-based weak signature detection method and its application on rolling element bearing prognostics," *Journal of Sound and Vibration*, vol. 289, pp. 1066–1090, 02 2006.

[19] V. Vercruyssen, W. Meert, G. Verbruggen, K. Maes, R. Bäumer, and J. Davis, "Semi-supervised anomaly detection with an application to water analytics," in *2018 IEEE International Conference on Data Mining (ICDM)*, 2018, pp. 527–536.

[20] T. Ooijevaar, K. Pichler, Y. Di, S. Devos, B. Volckaert, S. V. Hoecke, and C. Hesch, "Smart machine maintenance enabled by a condition monitoring living lab," in *8th IFAC Symposium on Mechatronic Systems MECHATRONICS 2019*, vol. 52, no. 15. IFAC, 2019, pp. 376–381.

[21] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *CoRR*, vol. abs/1502.03167, 2015.

[22] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 56, pp. 1929–1958, 2014.

[23] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *International Conference on Learning Representations*, 12 2014.

AUDITORY NEURAL RESPONSE INSPIRED SOUND EVENT DETECTION BASED ON SPECTRO-TEMPORAL RECEPTIVE FIELD

Deokki Min, Hyeonuk Nam, Yong-Hwa Park

Korea Advanced Institute of Science and Technology, South Korea
{minducky, frednam, yhpark}@kaist.ac.kr

ABSTRACT

Sound event detection (SED) is one of tasks to automate function by human auditory system which listens and understands auditory scenes. Therefore, we were inspired to make SED recognize sound events in the way human auditory system does. Spectro-temporal receptive field (STRF), an approach to describe the relationship between perceived sound at ear and transformed neural response in the auditory cortex, is closely related to recognition of sound. In this work, we utilized STRF as a kernel of the first convolutional layer in SED model to extract neural response from input sound to make SED model similar to human auditory system. In addition, we constructed two-branched SED model named as Two Branch STRFNet (TB-STRFNet) composed of STRF branch and baseline branch. While STRF branch extracts sound event information from auditory neural response, baseline branch extracts sound event information directly from the mel spectrogram just as conventional SED models do. TB-STRFNet outperformed the DCASE baseline by 4.3% in terms of threshold-independent macro F1 score, achieving 4th rank in DCASE Challenge 2023 Task 4b. We further improved TB-STRFNet by applying frequency dynamic convolution (FDYConv) which also leveraged domain knowledge on acoustics. As a result, two branch model applied with FDYConv on both branches outperformed the DCASE baseline by 6.2% in terms of the same metric.

***Index Terms*—** Sound event detection, STRF, Auditory scene analysis, Human auditory system, auditory neural response

1. INTRODUCTION

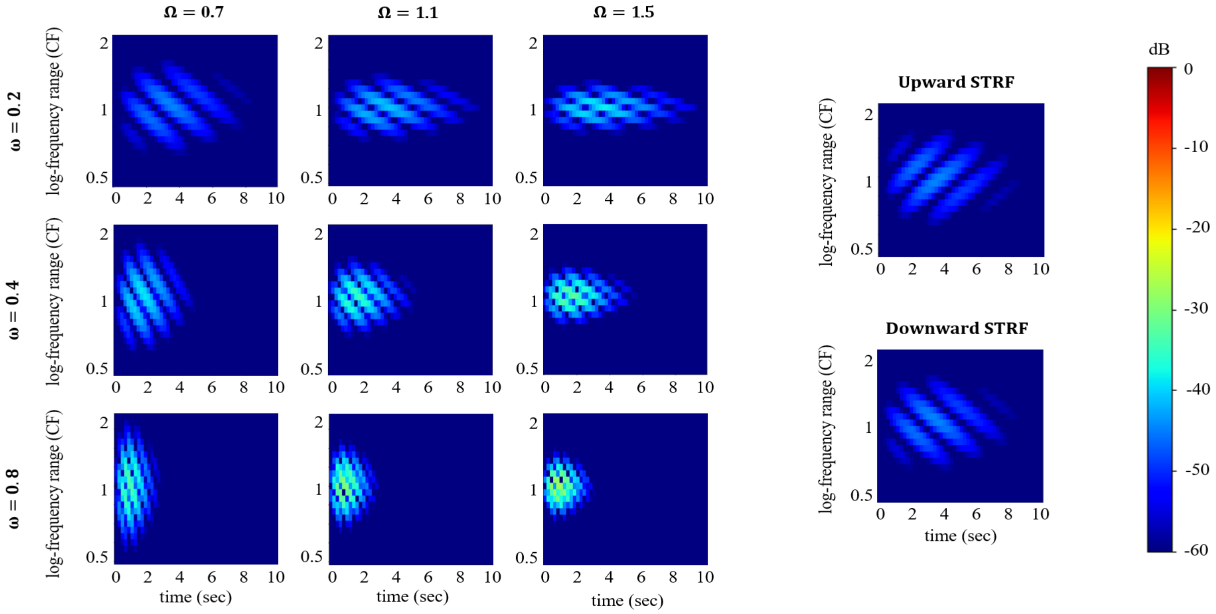
Sound event detection (SED) is a task for recognition of sound event class and their corresponding time onset and offset [1-4]. SED is closely related to human auditory perception, in that recognizing sound events and their respective time information is essential for the understanding of surrounding acoustic context. Therefore, we were inspired to improve SED by exploiting findings from auditory scene analysis (ASA), a field that aims to translate complex acoustic scene into auditory perception representations within human brain [5]. As sound passes through each part of auditory system, it is transformed into meaningful neural responses by which the auditory cortex can comprehend the perceptual meaning through several steps [6]. The processes include nonlinear amplification, frequency analysis, transformation from vibration into electric signal and higher-order neural computation [7-8]. While aforementioned steps are widely studied and applied to various audio and speech processing works [9-11], the transformation of sound stimulus into auditory cortical neural response is still not entirely comprehended and remains as the subject of ongoing research [12-14].

One approach to simulate the process of transformation from sound stimulus to auditory cortical neural response is to use spectro-temporal receptive field (STRF). STRF is defined as descriptive linear function which predicts primary auditory cortex (A1) cell response for given time-frequency representation of the sound [15, 16]. To estimate STRF, several methods such as spike-triggered average [17], boosting [18] and machine-learning method such as SVM [19] have been applied to experimental data. Observation on A1 cell response and estimated STRF has revealed that A1 cells have modulation-reactive characteristic that they are easily activated by ripple stimulus which is temporally and spectrally modulated signal [20]. Such spectro-temporal modulation are known to mediate analysis of sound such as speech so that we can obtain the sound intelligibility [21]. In Chi et al [20], STRF is constructed considering such reactive characteristic to dynamic modulation so that STRF captures the spectro-temporal modulation. While some works used the constructed STRF on deep learning applications to extract perceptually important characteristic [22-24], STRF is yet to be applied on SED to the best of our knowledge.

We applied STRFNet proposed by Vuong et al [22], which uses STRF as a convolution kernel in the first convolutional layer of the convolutional neural network (CNN) to imitate the neural response of primary auditory cortex (A1), on SED. However, STRFNet concentrates on extracting modulation property that it is not sufficient to extract various information within sound. To tackle the limitation, we propose two-branch model named as Two Branch STRFNet (TB-STRFNet). While STRF branch extracts the neuroscience-inspired dynamic modulation information using STRF kernel, baseline branch uses conventional convolution to extract the complementary time-frequency information which would not be captured by STRF branch. In addition, we apply frequency dynamic convolution (FDYConv) on TB-STRFNet to further improve the performance. While STRFNet is inspired by auditory neural response to the sound, FDYConv is inspired by the physical nature of time-frequency sound representation. FDYConv was shown to perform the best when applied on on both branches of TB-STRFNet. Joint application of TB-STRFNet and FDYConv significantly improved

---

This work was supported by the Institute of Civil Military Technology Cooperation funded by the Defense Acquisition Program Administration and Ministry of Trade, Industry and Energy of Korean government under grant No. UM22409RD4, and Korea Research Institute of Ships and Ocean engineering a grant from Endowment Project of “Development of Open Platform Technologies for Smart Maritime Safety and Industries” funded by Ministry of Oceans and Fisheries (PES4880).



(a) (b)

Figure 1. STRF examples with (a) varying scales (Ω) and rates (ω), (b) upward and downward direction.

performance over TB-STRFNet, proving compatibility between two methods as well as the importance of considering domain knowledge.

2. PROPOSED METHODS

2.1. STRF construction

We adopted STRF construction method by Chi et al [20], where STRF design is abstracted considering particular physiological A1 cell characteristic. Since A1 cell response can be predicted with convolution of sound spectrogram and STRF, STRF should reflect the A1 cell response property. There exists observation on physiological data that A1 cell response is effectively elicited by spectro-temporally modulated ripple [25, 26]. Ripple is defined as a spectro-temporally modulated signal which has temporally varying sinusoidal spectrum along log frequency spacing [27]. Given that A1 cell's modulation-reactive property, STRF needs to be constructed to capture the spectro-temporal modulation. STRF design in Chi et al [20] is a function of both spectral and temporal modulation parameters, which are scale (Ω) and rate (ω) respectively. Scale represents neurons' reaction on range of spectral modulation, while rate represents neurons' reaction on range of temporal modulation. Spectrally and temporally variously tuned neurons could be explained by different combinations of scale and rate values.

Fig. 1 (a) represents constructed STRF examples for varying scale and rate settings. X-axis and Y-axis represent time and logarithmic frequency range respectively. STRF is always centered at its center frequency (CF). As 1 CF represents the center frequency, 2 CF and 0.5 CF are double and half of its center frequency respectively. STRF frequency range lies on 2 octaves from 0.5 CF to 2 CF. In fig. 1 (a), scale increases from left column to right column, while rate increases from top to bottom row. Spectral spacing of ripples is narrower in higher scale while it is wider at lower scale. This illustrates that STRF is narrowly tuned to its center frequency at higher scale while it is broadly tuned at lower scale. Temporal spacing of ripple is narrower in higher rate while it is wider in lower rate. This reflects the characteristics of STRF which is more reactive to impulsive stimulus with higher rate while more reactive to prolonged response time with higher rate. Therefore, the scale variation shows that scale reflects the neural frequency tuning property, while the rate variation shows that rate reflects the neural temporal response property.

STRF has upward and downward direction as shown in fig. 1 (b). While upward direction STRF captures increasing spectral component as time passes, downward STRF captures decreasing spectral component as time passes. Both directions of modulation have to be considered to effectively capture the perceptual meaning of the sound. Note that fig.1 (a) is illustrated as downward STRFs just for consistency.

2.2. STRFNet

Constructed STRF has been used as a kernel of the first convolutional layer to tackle several audio-related tasks [22, 23]. We also use this method to verify the effectiveness of STRF on SED. In fig. 2 (a), the architecture of the baseline model used in this work is depicted. Baseline model is composed of six convolution blocks in series followed by two Bi-directional gated recurrent unit (GRU) layers and two fully connected layers. "ConvBlock" in fig. 2 consists of 2D convolutional layer, batch normalization, ReLU activation and 2D maxpool. STRFNet architecture is shown in fig.2 (b), where STRFConv layer is added in front of the baseline model. STRFConv uses 64 different STRFs as convolutional layer kernels, where 32 STRFs are for upward direction modulation and the other 32 STRFs are for downward direction modulation. Instead of directly training the convolution

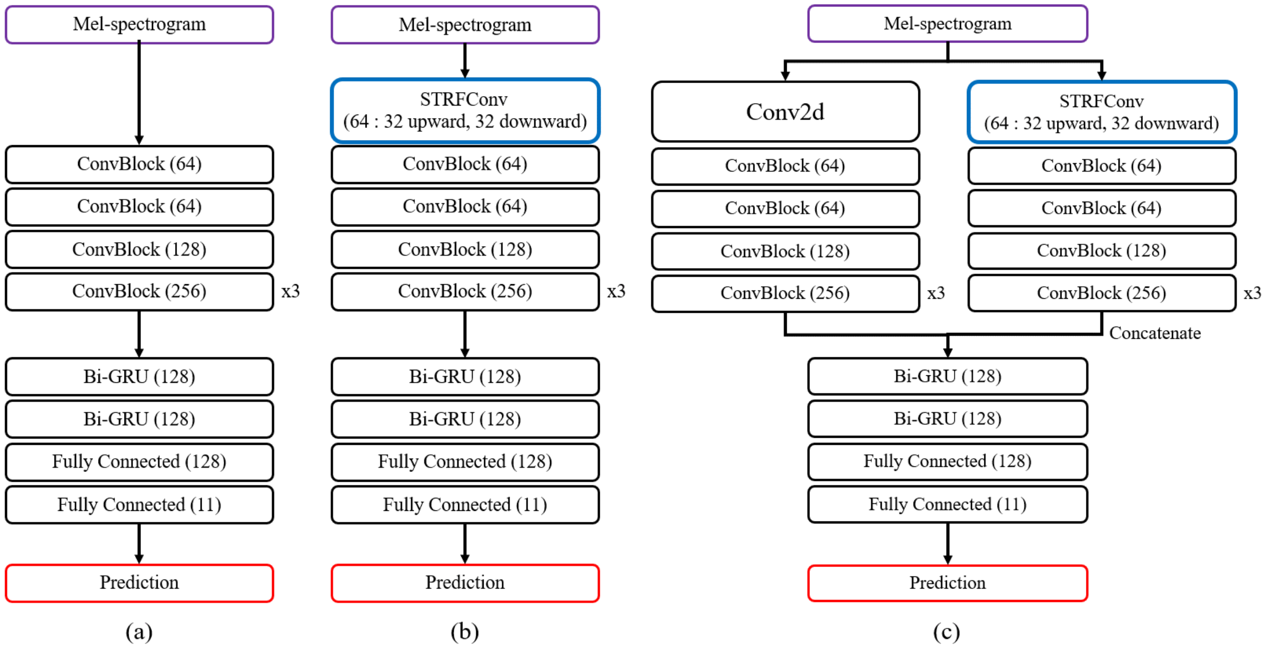


Figure 2: Architectures of (a) baseline model, (b) STRFNet and (c) Two-Branch STRFNet (TB-STRFNet).

kernel as in the conventional convolutional layers, STRFConv trains scales and rates corresponding to the channels of the kernel and then produce kernels using trained scales and rates, thus it is lighter than conventional convolutional layers in terms of the number of parameters. In this work, 32 sets of scales and rates are trained in STRFConv to construct 32 upward and downward sets of STRF.

2.3. TB-STRFNet

STRF kernel only extract spectral and temporal modulation from the sound, while other characteristics of time-frequency patterns within sound would be required for recognition of sound events. In addition, STRF kernel has large kernel size of 50 by 48 and can capture only monotonic modulation form. To compensate such limitations of STRFConv, we propose two-branch model composed of STRF branch and baseline branch, and named as TB-STRFNet. STRF branch is taken from STRFNet while baseline branch is taken from the baseline model with additional 2D convolutional layer before the CNN structure as shown in fig. 2 (c).

Both branches take the identical input which is mel-spectrogram. The STRF branch captures the neuroscience inspired spectro-temporal modulation information from the mel spectrogram. On the other hand, baseline branch consists of a 2D convolutional layer and six convolution blocks. Since STRF branch applies large kernels to extract modulation-related sound information, we expect the baseline branch to extract complementary information using small kernels by focusing on detailed time-frequency patterns. Extracted features from two branches are concatenated to combine separately extracted information. Concatenated feature map would go through the remaining layers with the same procedure as the baseline model.

2.4. STRF with frequency dynamic convolution

To further improve the performance of STRFNet and TB-STRFNet, we experimented on application of frequency dynamic convolution (FDYConv) on those models. FDYConv is proposed to tackle the problem that 2D convolutional layer applies translational equivariance on the frequency dimension while the frequency dimension is shift-variant [9]. This is also related to human auditory system in that it can distinguish frequency-wise translation. Thus FDYConv make sense with the idea of making SED models function similar to human auditory system. To test compatibility of FDYConv with STRFNet, we replaced all convolution layers in STRFNet by FDYConv and named as STRF-FDYNet. In addition, we applied FDYConv on TB-STRFNet to further improve the performance. We replaced convolution layers by FDYConv in baseline branch only, STRF branch only, and both branches of TB-STRFNet and named as TB-STRF-FDYNet1, TB-STRF-FDYNet2, and TB-STRF-FDYNet3, respectively. FDY replaced only convolution layer in convolution blocks for each branch of models, so that first layer of each branch is not FDYConv just as in the original implement of FDYConv [9].

3. EXPERIMENTAL DETAILS

3.1. Implementation Details

MAESTRO Real dataset is composed of 49 audio clips with duration of 3 to 5 minutes and sampling rate of 44.1 kHz [28]. Mel-spectrogram is used as input feature with 8,820 hop length, 17,640 window length and 64 mel-bin. For training, epoch number is 150, batch size is 32, mean-square error for loss function

Table 1: Performance on various SED models.

Model	Params	$F1_{MO}$(%)
DCASE baseline [29]	0.38M	42.91
baseline	2.22M	43.76
TDY-CRNN	7.01M	43.57
FDY-CRNN	7.01M	44.06
STRFNet	2.25M	43.19
STRF-FDYNet	7.24M	**44.33**
TB-baseline	4.08M	44.28
TB-STRFNet	4.08M	44.75
TB-STRF-FDYNet1	9.06M	44.81
TB-STRF-FDYNet2	9.06M	45.16
TB-STRF-FDYNet3	14.05M	**45.55**

and Adam optimizer are used. 5 cross-fold validation setup is used for stable overall evaluation.

3.2. Other SED models for comparison

For comparison, various models are adopted. DCASE baseline model is provided by DCASE Challenge 2023 Task4 subtask B baseline [29]. It has simple model architecture with three CNN layers, one Bi-directional GRU, followed by two fully connected layers. Other than the DCASE baseline model, the models with other methods are based on the baseline model in fig. 2 (a). Temporal dynamic convolution (TDYConv) [11] and frequency dynamic convolution (FDYConv) [9] are dynamic convolution models whose CNN kernel is weighted with time-wise attention and frequency-wise attention respectively. Each temporal and spectral axis-wise attention is extracted from the convolution input. TDY-CRNN and FDY-CRNN are applied in this work to compare the performance with STRF-based models, as they also function similar to human auditory perception and show decent performances. For both TDY-CRNN and FDY-CRNN, dynamic convolutional layer replaced all convolutional layers of the baseline except for the first layer.

3.3. Evaluation Metrics

Macro-average F1 score with optimum threshold ($F1_{MO}$) is used for main evaluation metric of DCASE 2023 Task 4 subtask B [30, 31]. By finding the best threshold which is most fit to certain task, the metric can provide more accurate system evaluation and reduce the need of manual threshold optimization. For comparison of model performance, the performance of each model is averaged by 10 sessions, in that 5 cross-validation procedure is performed for one session.

4. RESULTS AND DISCUSSION

SED performance of various models discussed in this paper are listed Table 1. For performance of single branch models, TDY-CRNN fails to outperform the baseline model. Since the models apply bi-GRU which considers sequential information between time frames, TDY-CRNN did not improve much compared to the baseline. On the other hand, FDY-CRNN which releases translational equivariance of frequency dimension showed significant improvement. STRFNet which captures the spectro-temporal modulation information, performed worse than the baseline. As we expected, large kernel size of STRF missed the detailed time-frequency information and lead to lower performance. However, joint application of STRFConv and FDYConv has further improved the performance of FDY-CRNN. Considering that STRFConv worsen the performance of baseline while enhance the performance of FDY-CRNN, there exists a synergy between STRFNet and FDYConv. The synergy seems to be due joint application of two methods consistent to principles by human auditory system.

TB-STRFNet which aims to capture detailed time-frequency information while extracting spectro-temporal modulation information, outperformed the baseline and the other single branch models. To verify the effect of increased model size, we constructed TB-baseline, which is consist of two baseline branches. While TB-baseline has almost the same number of parameters with TB-STRFNet, TB-STRFNet outperforms TB-baseline. Since TB-baseline outperforms the baseline, increased model size has affected the model performance. However, considering that application of STRFConv on single branch model has worsened performance, positive effect by STRFConv on TB-STRFNet is apparent. STRF effectively extracts additive information from the mel-spectrogram which is helpful to discriminate the event class and its time onset/offset in two branch architecture. In addition, we interpret that TB-STRFNet outperforms TB-baseline because different role of TB-STRFNet branches efficiently extract the various information from the input sound stimulus. Proposed TB-STRFNet is submitted to DCASE Challenge 2023 Task 4b and achieved 4th rank outperforming DCASE baseline by 4.3% [32].

All three models with join application of FDYConv and TB-STRFNet outperformed TB-STRFNet. Since both FDYConv has improved both baseline model and STRFNet, it is effective whether STRFConv is applied or not. In addition, TB-STRF-FDYNet2 applying FDYConv on STRF branch performed better than TB-STRF-FDYNet1 applying FDYConv on the baseline branch. This proves the synergy between STRFConv and FDY-Conv again, as they perform better when applied together than when applied separately. TB-STRF-FDYNet3 performed the best, improving the baseline by 6.2%. This again proves that methods consistent to human auditory systems are effective on SED and using them together even results in great synergy.

5. CONCLUSION

In this work, we applied STRF as a convolutional layer kernel on SED to build SED model functioning closer to human auditory system. While STRFNet performed not as good as the baseline model, TB-STRFNet outperformed the baseline and showed the effect of extracting spectral and temporal modulation information on SED. Furthermore, reflecting frequency-varying perceptual property of auditory system, we applied FDYConv together with STRF. The superior performance of STRF-FDYNet and TB-STRF-FDYNet3 proves that a physiologically consistent deep learning methods enhance SED performance. For future works, we suggest to consider further physiological A1 cell response properties. STRF has dynamic property that STRF is known to be dependent to input sound stimulus [33]. Thus, we may construct the dynamic STRF based model which consider such dynamic STRF property.

6. REFERENCES

[1] A. Mesaros, T. Heittola, T. Virtanen, and M. D. Plumbley, "Sound Event Detection: A tutorial," *IEEE Signal Processing Magazine,* vol. 38, pp. 67-83, September 01, 2021

[2] T. Virtanen, M. D. Plumbley, and D. Ellis, *Computational analysis of sound scenes and events,* Springer, 2018.

[3] J. Ebbers and R. Haeb-Umbach, "Pre-Training And Self-Training For Sound Event Detection In Domestic Environments," DCASE2022 Challenge, Tech. Rep, 2022.

[4] H. Nam, B.Y. Ko, G.T. Lee, S.H. Kim, W.H. Jung, S.M. Choi and Y.H. Park, "Heavily Augmented Sound Event Detection utilizing Weak Predictions," DCASE2022 Challenge, Tech. Rep, 2021.

[5] A.S. Bregman, *Auditory scene analysis: The perceptual organization of sound,* MIT press, pp. 3, 1994.

[6] J. Schnupp, I. Nelken, and A. King, *Auditory neuroscience: Making sense of sound,* MIT press, pp. 51-92, 2011.

[7] D.T. Kemp, "Evidence of mechanical nonlinearity and frequency selective wave amplification in the cochlea," *Archives of oto-rhino-laryngology,* Springer, vol. 224, pp. 37-45, 1979.

[8] J.J. Zwislocki, "Five decades of research on cochlear mechanics." *The Journal of the Acoustical Society of America,* vol. 67, pp. 1679-1685, 1980.

[9] H. Nam, S.H. Kim, and Y.H. Park, "Frequency Dynamic Convolution: Frequency-Adaptive Pattern Recognition for Sound Event Detection,", *INTERSPEECH 2022*. ISCA, 2022.

[10] H. Nam, S.H. Kim, and Y.H. Park, "Filteraugment: An acoustic environmental data augmentation method," *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022.

[11] S.H. Kim, H. Nam, and Y.H. Park. "Temporal dynamic convolutional neural network for text-independent speaker verification and phonemic analysis," *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022.

[12] K. Jasmin, C.F. Lima, and S.K. Scott. "Understanding rostral–caudal auditory cortex contributions to auditory perception," *Nature Reviews Neuroscience*, vol. 20, no. 7, pp. 425-434, 2019.

[13] M. Wang et al. "Single-neuron representation of learned complex sounds in the auditory cortex," *Nature communications*, 2020.

[14] R. Bishop, F. Qureshi, and J. Yan. "Age-related changes in neuronal receptive fields of primary auditory cortex in frequency, amplitude, and temporal domains," *Hearing Research*, 2022.

[15] A. Aertsen and P.I. Johannesma. "Spectro-temporal receptive fields of auditory neurons in the grassfrog: I. Characterization of tonal and natural stimuli," *Biological Cybernetics*, vol. 38, no. 4, pp. 223-234, 1980.

[16] A. Aertsen and P. I. Johannesma. "The spectro-temporal receptive field: a functional characteristic of auditory neurons." *Biological cybernetics*, 1981.

[17] J. J. Eggermont, A. M. H. J. Aertsen, and P. I. M. Johannesma, "Quantitative characterisation procedure for auditory neurons based on the spectro-temporal receptive field," *Hearing Research,* vol. 10, no. 2, pp. 167-190, 1983.

[18] S. V. David, N. Mesgarani, and S. A. Shamma, "Estimating sparse spectro-temporal receptive fields with natural stimuli," *Network: Computation in neural systems,* vol. 18, no. 3, pp. 191-212, 2007.

[19] A. F. Meyer, M. F. Happel, F. W. Ohl, and J. Anemüller, "Estimation of spectro-temporal receptive fields based on linear support vector machine classification," *BMC Neuroscience,* vol. 10, pp. 1-2, 2009.

[20] T. Chi, Y. Gao, M. C. Guyton, P. Ru, and S. Shamma, "Spectro-temporal modulation transfer functions and speech intelligibility," *The Journal of the Acoustical Society of America,* vol. 106, no. 5, pp. 2719-2732, 1999.

[21] A. Edraki, W.Y. Chan, J. Jensen, and D. Fogerty, "Speech intelligibility prediction using spectro-temporal modulation analysis." *IEEE/ACM transactions on audio, speech, and language processing,* pp. 210-225, 2020.

[22] T. Vuong, Y. Xia, and R. Stern. "Learnable Spectro-Temporal Receptive Fields for Robust Voice Type Discrimination." *Interspeech 2020*, ISCA, 2020.

[23] R. Sharma, T. Vuong, M. Lindsey, H. Dhamyal, R. Singh, and B. Raj, "Self-supervision and Learnable STRFs for Age, Emotion, and Country Prediction," p. arXiv:2206.12568.

[24] C.Y. Wang, P.C. Chang, J.J. Ding, T.C. Tai, A. Santoso, Y.T. Liu, J.C. Wang., "Spectral–temporal receptive field-based descriptors and hierarchical cascade deep belief network for guitar playing technique classification," *IEEE Transactions on Cybernetics*, vol. 52, no. 5, pp. 3684-3695, 2020.

[25] N. Kowalski, D.A. Depireux, and S.A. Shamma. "Analysis of dynamic spectra in ferret primary auditory cortex. I. Characteristics of single-unit responses to moving ripple spectra." *Journal of neurophysiology,* pp. 3503-3523, 1996.

[26] R.C. DeCharms, D.T. Blake, and M.M. Merzenich. "Optimizing sound features for cortical neurons." *Science,* pp. 1439-1444, 1998.

[27] S. Shamma "Auditory cortical representation of complex acoustic spectra as inferred from the ripple analysis method." *Network: Computation in Neural Systems*, pp. 439-476, 1996.

[28] I. Martín-Morató and A. Mesaros. "Strong labeling of sound events using crowdsourced weak labels and annotator competence estimation." *IEEE/ACM transactions on audio, speech, and language processing*, pp. 902–914, 2023.

[29] I. Martín-Morató, M. Harju, P. Ahokas, and A. Mesaros, "Strong labeling of sound events using crowdsourced weak labels and annotator competence estimation," *International Conference on Acoustic, Speech and Signal Processing (ICASSP),* IEEE, 2023.

[30] A. Mesaros, T. Heittola, and T. Virtanen. "Metrics for polyphonic sound event detection," *Applied Sciences*, pp. 162, 2016.

[31] J. Ebbers, R. Haeb-Umbach, and R. Serizel. "Threshold Independent Evaluation of Sound Event Detection Scores," *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022.

[32] D. Min, H. Nam, and Y.H. Park "Application of Spectro-Temporal Receptive Field for DCASE 2023 Challenge Task4 B", DCASE2023 Challenge, Tech. Rep, 2023.

[33] J. J. Eggermont, "Context dependence of spectro-temporal receptive fields with implications for neural coding," *Hearing research,* vol. 271, no. 1-2, pp. 123-132, 2011.

CREATING A GOOD TEACHER FOR KNOWLEDGE DISTILLATION IN ACOUSTIC SCENE CLASSIFICATION

Tobias Morocutti[2], *Florian Schmid*[1], *Khaled Koutini*[2], *Gerhard Widmer*[1,2]

[1]Institute of Computational Perception (CP-JKU), [2]LIT Artificial Intelligence Lab,
Johannes Kepler University Linz, Austria
{tobias.morocutti, florian.schmid, khaled.koutini}@jku.at

ABSTRACT

Knowledge Distillation (KD) is a widespread technique for compressing the knowledge of large models into more compact and efficient models. KD has proved to be highly effective in building well-performing low-complexity Acoustic Scene Classification (ASC) systems and was used in all the top-ranked submissions to this task of the annual DCASE challenge in the past three years. There is extensive research available on establishing the KD process, designing efficient student models, and forming well-performing teacher ensembles. However, less research has been conducted on investigating which teacher model attributes are beneficial for low-complexity students. In this work, we try to close this gap by studying the effects on the student's performance when using different teacher network architectures, varying the teacher model size, training them with different device generalization methods, and applying different ensembling strategies. The results show that teacher model sizes, device generalization methods, the ensembling strategy and the ensemble size are key factors for a well-performing student network.

Index Terms— Acoustic Scene Classification, Knowledge Distillation, CP-Mobile, Patchout FaSt Spectrogram Transformer (PaSST), CP-ResNet

1. INTRODUCTION

The objective of Acoustic Scene Classification (ASC) involves labeling an audio clip with a corresponding scene. The DCASE23 challenge's [1] Low-Complexity Acoustic Scene Classification task focuses on utilizing the *TAU Urban Acoustic Scenes 2022 Mobile development dataset (TAU22)* [2]. This dataset comprises one-second audio snippets from ten distinct acoustic scenes. In an attempt to make the models deployable on edge devices, a complexity limit on the models is enforced: models are constrained to have no more than 128,000 parameters and 30 million multiply-accumulate operations (MMACs) for the inference of a 1-second audio snippet. Among other model compression techniques such as Quantization [3] and Pruning [4], Knowledge Distillation (KD) [5–7] proved to be a particularly well-suited technique to improve the performance of a low-complexity model in ASC.

In a standard KD setting, a low-complexity model learns to mimic the teacher by minimizing a weighted sum of hard label loss and distillation loss. While the soft targets are usually obtained by one or multiple possibly complex teacher models, the distillation loss tries to match the student predictions with the computed soft targets based on the Kullback-Leibler divergence.

Jung et al. [8] demonstrate that soft targets in a teacher-student setup benefit the learning process since one-hot labels do not reflect the blurred decision boundaries between different acoustic scenes. Knowledge distillation has also been a very popular method in the DCASE challenge submissions. For example, Kim et al. [9] apply KD using a pretrained teacher. Further, [10] and [11] employ KD to train a low-complexity network on the predictions of a more complex one. Schmid et al. [12] use KD to train a low-complexity CNN on a teacher ensemble consisting of five PaSST [13] models.

To enhance generalization across recording devices, Kim et al. propose a modified version of MixStyle [14] called Freq-MixStyle [12, 15]. This method normalizes each frequency band and denormalizes it with mixed frequency statistics of two different samples.

Another method for improving the device generalization is Device Impulse Response Augmentation [16] which was introduced by Morocutti et al. It convolves audio signals with impulse responses of vintage microphones to increase the recording device variety in the training phase.

In this work, we study the effects of training a low-complexity network on the predictions of a single teacher or a teacher ensemble. We experiment with different network architectures, model sizes and device generalization methods to create the single teacher model that leads a student to perform best on the validation set. Additionally, we analyze the effect of combining teacher models with different network architectures, sizes, or device generalization methods.

2. NETWORK ARCHITECTURES

We experiment with three different teacher networks that were shown [17] to perform well as a teacher for the task of ASC. The architectures consist of two receptive-field regularized [18] convolutional neural networks (CNNs): CP-ResNet [19] and CP-Mobile [17], as well as a Transformer model: Patchout faSt Spectrogram Transformer (PaSST) [13].

2.1. CP-Mobile

CP-Mobile (CPM) [17] is an efficient architecture optimized for ASC. This architecture is designed to be less complex than CP-ResNet by factorizing convolution operations, such as in MobileNets [20, 21] and EfficientNets [22], while maintaining important properties that were shown to be important for ASC tasks, such as the regularized receptive field [18, 19].

In the following experiments, the student model has the CPM architecture with the following configuration: 32 base channels, an expansion rate of 3 and a channels multiplier of 2.3. The details of the CPM architecture are described in [17]. In short, these attributes control the scale of the network: the base channels represent the width of the first few blocks of the network; the channels multiplier determines the expansion in the number of channels as the network gets deeper (i.e. the number of channels in the last convolutional blocks is the number of channels of the previous blocks multiplied by channels multiplier); the expansion rate determines the number of channels in the depthwise convolution. The resulting model consists of almost 128K parameters and 29 million multiply-accumulate operations (MMACs).

We choose CPM as a student model since the architecture is designed for low-complexity ASC and has been shown to outperform CP-ResNet in previous work [17]. In addition, we experiment with using a scaled-up version of CPM as a teacher model for KD. To scale up the network, we increase the width via the base-channels hyperparameter.

2.2. CP-ResNet

CP-ResNet (CPR) [18, 19] is a receptive-field regularized CNN which has been shown to be very successful for ASC in previous editions of the DCASE ASC challenge [1, 2, 23, 24]. Therefore, we also use this network as a teacher model. We use the number of base channels to scale up the network in order to create teacher models with different sizes, similar to the procedure outlined for CPM.

2.3. PaSST

The Patchout faSt Spectrogram Transformer (PaSST) [13] is a complex, self-attention-based model, which is pre-trained on AudioSet [25] and consists of 85M parameters. The pre-trained model can be fine-tuned to achieve state-of-the-art performances on multiple downstream tasks, including ASC [13]. Additionally, PaSST models have proven to be excellent teachers for low-complexity CNNs [12, 26, 27]. Therefore, we also experiment with PaSST as a teacher model.

3. KNOWLEDGE DISTILLATION

We train our student model on the pre-computed predictions of the teacher model or teacher ensemble in addition to the one-hot encoded labels, similar to [27]. Training the student model on the soft labels of the teacher (ensemble) results in the student model learning blurred decision boundaries and establishing important similarity structures between classes. The loss is given in Equation 1 and consists of the hard label loss L_t and distillation loss L_{kd}. The label and distillation loss are weighted using the factor λ. The student and teacher logits are denoted by z_s and z_t, while y stands for the hard labels. τ is a temperature to control the sharpness of the probability distributions created by the softmax activation δ. L_l indicates the Cross-Entropy loss and the Kullback Leibler divergence is used as distillation loss L_{kd}.

$$Loss = \lambda L_l(\delta(z_S), y) + (1-\lambda)\tau^2 L_{kd}(\delta(z_S/\tau), \delta(z_T/\tau)) \quad (1)$$

As suggested in [5], we multiply the distillation loss by τ^2 since the magnitudes of the gradients produced by the soft targets scale as $1/\tau^2$. This ensures that the relative contributions of the hard and soft targets remain roughly unchanged if the temperature used for distillation is modified.

3.1. Experimental Setup

We train the teacher models as well as the student models on the TAU22 [2] dataset with the shifted crops dataset augmentation described in [17]. Regarding Knowledge Distillation, we use the values of 0.02 and 2 for λ and temperature τ, respectively.

For device generalization (DG) we experiment with Freq-MixStyle (FMS) [12, 15] and Device Impulse Response (DIR) augmentation [16] and the combination thereof. FMS is configured by two parameters: α_{fms} determines the shape of the Beta distribution used to randomly draw mixing coefficients, and p_{fms} specifies the probability of whether it is applied to a batch or not. Similar to FMS, DIR is guided by a probability p_{dir} that determines the augmentation strength by specifying the proportion of samples to augment.

The configurations used for FMS and DIR are adapted for each architecture. Results in [16] show that PaSST performs best with $\alpha_{fms} = 0.4$, $p_{fms} = 0.4$ and $p_{dir} = 0.6$ whereas CPR achieves the highest validation accuracy using $\alpha_{fms} = 0.3$, $p_{fms} = 0.8$ and $p_{dir} = 0.4$. While our experiments found that CPM teachers perform well using the same configuration as used for CPR, setting α_{fms}, p_{fms} and p_{dir} to 0.3, 0.4 and 0.6 when training the student network results in higher validation accuracy. More details about our experimental setup are reported in [17].

4. SINGLE MODEL TEACHER

In this section, we compare the performance of different teachers and evaluate the performance of students trained on the predictions of different teacher models using KD. We experiment with using a single CPM, CPR or PaSST model as the teacher and a low-complexity CPM as the student.

4.1. Scaling the Teacher

To investigate the effect of training the student on teachers of different complexity, we scale CPM and CPR by increasing the number of base channels, which modifies the width of the network. We test the effect of scaling the teacher only on CPM and CPR since we use a pre-trained PaSST model.

We trained CPM and CPR models in five different complexity configurations such that their number of parameters is approximately 128K, 450K, 1M, 4M and 8M. Since the number of parameters of CPM and CPR does not scale equally when increasing the base channels, we selected the number of base channels for each size and architecture individually. We used 32, 56, 88, 168 and 232 base channels for CPR and 32, 64, 96, 184 and 264 base channels for CPM.

All different configurations are evaluated over three runs and to ensure that our experiments are independent of each other, we train one student on each of the three teachers.

Additionally, we apply a combination of Freq-MixStyle and Device Impulse Response augmentation to all student as well as all teacher models. From now on, we will refer to the combination of DIR and FMS as DIRFMS.

Table 1 shows that for the teacher, CPM outperforms CPR in each complexity configuration. Additionally, even the smallest variant of CPM achieves a higher validation accuracy than PaSST, which has several orders of magnitude more parameters.

However, the students trained on CPM perform worse than the ones trained on CPR for each size of teacher. Furthermore, the students trained using PaSST as a teacher outperform the best students trained on a CPR variant by only 0.05%. While the teacher with 450K parameters works best for CPR, the variant with 128K parameters makes the best CPM teacher.

In short, the results show that the right scale of a CNN teacher can improve the performance of the students by more than 1%. Furthermore, smaller CNNs can be better teachers, even if the larger

		CPR		**CPM**		**PaSST**	
		T	S	T	S	T	S
Teacher size	**128K**	60.28	63.94	62.66	**63.70**	-	-
	450K	62.05	**64.60**	62.81	62.48	-	-
	1M	62.58	63.99	63.92	62.76	-	-
	4M	62.74	63.51	64.28	62.43	-	-
	8M	**63.28**	63.43	**64.62**	62.83	-	-
	85M	-	-	-	-	**62.20**	**64.65**

Table 1: Validation accuracy of different teacher networks, and a student model trained on these. T and S denote the performance of the teacher and student, respectively. While the teacher networks vary in architecture and size, the student model is always a CPM model with 128k parameters. All results are averages over three independent runs and the last 4 epochs of training.

teachers outperform the smaller ones. Finally, having a different architecture for teacher and student improves the performance of the student.

4.2. Effect of Device Generalization Methods

Table 2 presents the impact of the device generalization (DG) methods DIR, FMS and DIRFMS. For studying the effects of these methods, we use the teacher variations with 128K and 450K parameters for CPM and CPR, respectively, since these teacher models result in the best performing student models, as shown in Section 4.1.

	CPR		**CPM**		**PaSST**	
	T	S	T	S	T	S
Validation Accuracy						
DIRFMS	**62.05**	**64.60**	**62.66**	**63.70**	**62.20**	**64.65**
DIR	57.34	62.47	57.23	61.57	61.64	64.39
FMS	60.99	63.40	61.18	63.66	61.08	64.56
NO AUG	54.13	62.74	53.15	62.47	59.39	63.76
Unseen Accuracy						
DIRFMS	**56.95**	**60.43**	**57.92**	**59.20**	**58.73**	**61.03**
DIR	49.30	56.74	48.62	55.54	57.91	60.90
FMS	54.94	58.91	54.92	58.76	57.57	61.00
NO AUG	44.75	56.70	43.94	56.21	54.08	59.60

Table 2: Validation accuracy of teacher networks trained using different DG methods, and a student model trained on the corresponding teacher predictions. T and S denote the performance of the teacher and student, respectively. The CPM teacher has 128K parameters, the CPR teacher has 450K parameters. While the teacher network varies in architecture and used DG method, the student is always a CPM model with 128k parameters trained with DIRFMS. All results are averages over three independent runs and the last 4 epochs of training.

The results in Table 2 show that FMS, DIR and/or DIRFMS boost both the performance of the teacher models as well as the performance of the student models significantly. The results show that there is a clear effect of these methods on the validation accuracy. Moreover, this effect tends to be even higher on the unseen accuracy. Applying DIRFMS results in the best validation and unseen accuracy, outperforming DIR and FMS. We define *unseen accuracy* as the accuracy on the subset of the validation set that consists of samples of devices not present in the training set. Consistent with the findings in [16], FMS, DIR and DIRFMS have less effect on the performance of PaSST, compared to CPR or CPM.

5. ENSEMBLE TEACHER

Previous work [17] shows that training the student on the predictions of multiple teacher networks is a highly effective method to improve the accuracy of the student in the KD framework. This effect is even more significant when ensembling different architectures or models trained with different device generalization methods. In this section, we will experiment with different ensemble configurations and show their effect on the low-complexity student. We ensemble different models by averaging their logits.

	CPR		**CPM**		**PaSST**
size of teacher	128K	450K	128K	450K	85M
1 teacher	63.94	**64.60**	**63.70**	62.48	**64.65**
3 teacher	**64.53**	64.36	**63.97**	62.77	**64.81**

Table 3: Validation accuracy of student models trained on the predictions of either one or three teacher models which apply both Freq-MixStyle and Device Impulse Response augmentation (**DIRFMS**). The highest accuracy per architecture and per number of teacher is marked bold. For CPR and CPM, the teacher models consist of either 128K or 450K parameters. All results are averages over three independent runs and the last 4 epochs of training.

5.1. Ensembling Teachers with Identical Training Setup

This section presents experiments about ensembling different models that use the same training setup but different seeds. More precisely, we ensemble different models that share the same architecture, complexity and DG methods. The goal is to test if the averaged logits of multiple teacher models are better soft targets for training the student model.

Since the results in Table 2 indicate that DIRFMS has the most positive effect on the students for all teacher architectures, we evaluate the performance of students learning from a teacher ensemble trained with DIRFMS. Additionally, we choose to test the training of the student on the teacher ensembles with two different complexity configurations of the CPR and CPM teachers. Due to the fact that CPR performs best with 450K and CPM with 128K parameters, we select these two complexity levels to evaluate the teacher ensembling on both architectures.

As the results in Table 3 show, the CPR teacher with 128K parameters outperforms the variant with 450K parameters when using an ensemble of three teachers. Further, the variant with 128K parameters also works best for the CPM teacher, outperforming the 450K-parameters variant by 1.2%. When we train the students on the averaged logits of three PaSST models, the validation accuracy of the student increases slightly by 0.16%, compared to using only one PaSST teacher. However, PaSST outperforms the other architectures, with CPM performing worse than CPR.

Teacher Ensemble Variations							
Teacher Architecture							
CPR	✓	✗	✗	✓	✓	✗	✓
CPM	✗	✓	✗	✗	✓	✓	✓
PaSST	✗	✗	✓	✓	✓	✓	✗
Device Generalization Methods							
DIR + FMS	64.25	62.35	64.47	-	-	-	-
DIRFMS + DIR	64.21	63.45	64.63	-	-	-	-
DIRFMS	64.53	**63.97**	64.81	65.19	65.09	**65.15**	64.66
DIRFMS + FMS	**64.74**	63.76	64.89	**65.81**	65.12	64.67	**64.67**
DIRFMS + DIR + FMS	64.10	63.76	**65.16**	65.39	**65.18**	64.85	64.03

Table 4: The accuracy of the student model being trained on a teacher ensemble. The teacher ensembles differ in the combination of architectures and the combination of DG methods. A mark indicates that three models of the corresponding architecture are included in the ensemble. All results are averages over three independent runs and the last 4 epochs of training.

5.2. Ensembling Teachers Trained with Different DG Methods

In this section, we experiment with combining models with the same architecture but trained using different DG methods in order to create a better teacher ensemble. We choose 128K parameters as the teacher complexity for CPR and CPM, since this complexity performs best when combining multiple models, as shown in Table 3. We evaluate the effect of training the student on these teacher ensembles and compare the results with the performance of the students trained using the DIRFMS teacher ensemble described in Section 5.1. All evaluated teacher ensembles contain three models for each included DG method. This implies that the different ensembles stated in the left part of Table 4 contain between 3 and 9 models.

The results in Table 4 indicate that including teachers trained using DIRFMS in the ensemble is essential for every architecture, since the ensembles DIRFMS+FMS, DIRFMS and DIRFMS+DIR+FMS perform best for the CPR, CPM and PaSST architecture, respectively. Including the DIR teacher in the DIRFMS+FMS ensemble only increases the performance of students trained on the predictions of PaSST models. The best-evaluated ensemble of only one architecture is the PaSST DIRFMS+DIR+FMS ensemble, increasing the accuracy by 0.35% compared to the previously best PaSST DIRFMS ensemble.

5.3. Ensembling Teachers with Different Architectures

In this section, we experiment with ensembling different architectures motivated by the assumption that different architectures can learn different features and aspects of the training data and therefore ensembling them would result in a more robust model.

We test each combination of CPR, CPM and PaSST using the combinations of DG methods, which performed best on single architecture ensembles. It is worth noting that the teacher ensemble size depends on the number of used architectures and DG methods. It can therefore range from 6 (2 architectures x 1 DG method x 3 models) to 27 (3 architectures x 3 DG methods x 3 models).

The results in Table 4 clearly show that the teacher ensembles consisting of CPR and PaSST models result in the best-performing students. Adding CPM models to ensembles of CPR and PaSST models worsens the performance of the students for all evaluated DG configurations. More precisely, ensembling CPM and CPR does not lead to performance improvement, and neither does ensembling CPM and PaSST.

Regarding the DG methods, ensembling teacher models trained with DIRFMS and FMS results in the best student performance for the CPR and PaSST combination, creating the best-evaluated ensemble with 65.81% validation accuracy of the student.

6. CONCLUSION

In this work, we show that low-complexity CNNs like the CPM learn more important features from Transformers or relatively small CNNs compared to large CNNs when using Knowledge Distillation. Additionally, we show that applying Device Impulse Response (DIR) augmentation, Freq-Mixstyle (FMS) and especially the combination thereof (DIRFMS) to the teacher models significantly boosts the performance of the teachers and the students. The effect of these DG methods is even more noticeable on the unseen accuracy, compared to the total validation accuracy. Surprisingly, it turns out that the performance of the student does not necessarily improve with the scale of the teacher. For example, ensembling smaller teacher networks can be more beneficial than ensembling bigger ones. Furthermore, we show that the performance of the student improves when the teacher architecture is different than the student architecture. For example, when using PaSST or CPR to train CPM. In contrast, the low-complexity CPM student performs worse when it is trained on any higher complexity variation of the same architecture. Additionally, the predictions of PaSST and CPR complement each other, resulting in better student performance. Finally, using an ensemble of CPR and PaSST trained either using DIRFMS or FMS results in our best student, which has an accuracy of 65.81% with 128K parameters and 32 million MACCS, outperforming the much larger CPR, CPM and PaSST models.

7. ACKNOWLEDGMENT

The LIT AI Lab is supported by the Federal State of Upper Austria. GW's work is supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme, grant agreement No 101019375 (Whither Music?).

8. REFERENCES

[1] I. Martín-Morató, F. Paissan, A. Ancilotto, T. Heittola, A. Mesaros, E. Farella, A. Brutti, and T. Virtanen, "Low-complexity acoustic scene classification in DCASE 2022 challenge," in *DCASE Workshop*, 2022.

[2] T. Heittola, A. Mesaros, and T. Virtanen, "Acoustic scene classification in DCASE 2020 challenge: Generalization across devices and low complexity solutions," in *DCASE Workshop*, 2020.

[3] I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, and Y. Bengio, "Quantized neural networks: Training neural networks with low precision weights and activations," *J. Mach. Learn. Res.*, 2017.

[4] J. Frankle, G. K. Dziugaite, D. Roy, and M. Carbin, "Pruning neural networks at initialization: Why are we missing the mark?" in *ICLR*, 2021.

[5] G. E. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *CoRR*, 2015.

[6] J. Ba and R. Caruana, "Do deep nets really need to be deep?" in *NeurIPS*, 2014.

[7] A. M. Tripathi and O. J. Pandey, "Divide and distill: New outlooks on knowledge distillation for environmental sound classification," *IEEE ACM Trans. Audio Speech Lang. Process.*, 2023.

[8] H. Heo, J. Jung, H. Shim, and H. Yu, "Acoustic scene classification using teacher-student learning with soft-labels," in *Interspeech*. ISCA, 2019.

[9] B. Kim, S. Yang, J. Kim, and S. Chang, "QTI submission to DCASE 2021: Residual normalization for device-imbalanced acoustic scene classification with efficient design," DCASE2021 Challenge, Tech. Rep., 2021.

[10] J.-H. Lee, J.-H. Choi, P. M. Byun, and J.-H. Chang, "Hyu submission for the DCASE 2022: Efficient fine-tuning method using device-aware data-random-drop for device-imbalanced acoustic scene classification," DCASE2022 Challenge, Tech. Rep., 2022.

[11] R. Anastácio, L. Ferreira, F. Mónica, and C. B. Luís, "Ai4edgept submission to DCASE 2022 low complexity acoustic scene classification task1," DCASE2022 Challenge, Tech. Rep., 2022.

[12] F. Schmid, S. Masoudian, K. Koutini, and G. Widmer, "Knowledge distillation from transformers for low-complexity acoustic scene classification," in *DCASE Workshop*, 2022.

[13] K. Koutini, J. Schlüter, H. Eghbal-zadeh, and G. Widmer, "Efficient training of audio transformers with patchout," in *Interspeech*. ISCA, 2022.

[14] J. Fu, Y. Zhong, and F. Yang, "Adversarial domain generalization with mixstyle," in *International Conference on Advanced Robotics and Mechatronics (ICARM)*. IEEE, 2022.

[15] B. Kim, S. Yang, J. Kim, H. Park, J. Lee, and S. Chang, "Domain generalization with relaxed instance frequency-wise normalization for multi-device acoustic scene classification," in *Interspeech*. ISCA, 2022.

[16] T. Morocutti, F. Schmid, K. Koutini, and G. Widmer, "Device-robust acoustic scene classification via impulse response augmentation," in *EUSIPCO*. IEEE, 2023.

[17] F. Schmid, T. Morocutti, S. Masoudian, K. Koutini, and G. Widmer, "CP-JKU submission to DCASE23: Efficient acoustic scene classification with cp-mobile," DCASE2023 Challenge, Tech. Rep., 2023.

[18] K. Koutini, H. Eghbal-zadeh, and G. Widmer, "Receptive field regularization techniques for audio classification and tagging with deep convolutional neural networks," *IEEE ACM Trans. Audio Speech Lang. Process.*, 2021.

[19] K. Koutini, H. Eghbal-zadeh, M. Dorfer, and G. Widmer, "The receptive field as a regularizer in deep convolutional neural networks for acoustic scene classification," in *EUSIPCO*. IEEE, 2019.

[20] M. Sandler, A. G. Howard, M. Zhu, A. Zhmoginov, and L. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *CVPR*. IEEE, 2018.

[21] A. Howard, R. Pang, H. Adam, Q. V. Le, M. Sandler, B. Chen, W. Wang, L. Chen, M. Tan, G. Chu, V. Vasudevan, and Y. Zhu, "Searching for mobilenetv3," in *ICCV*. IEEE, 2019.

[22] M. Tan and Q. V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *ICML*. PMLR, 2019.

[23] K. Koutini, S. Jan, and G. Widmer, "CPJKU Submission to DCASE21: Cross-Device Audio Scene Classification with Wide Sparse Frequency-Damped CNNs," DCASE2021 Challenge, Tech. Rep., 2021.

[24] K. Koutini, F. Henkel, H. Eghbal-zadeh, and G. Widmer, "CP-JKU Submissions to DCASE'20: Low-Complexity Cross-Device Acoustic Scene Classification with RF-Regularized CNNs," DCASE2020 Challenge, Tech. Rep., 2020.

[25] J. F. Gemmeke, D. P. W. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio set: An ontology and human-labeled dataset for audio events," in *ICASSP*. IEEE, 2017.

[26] F. Schmid, S. Masoudian, K. Koutini, and G. Widmer, "CP-JKU submission to DCASE22: Distilling knowledge for low-complexity convolutional neural networks from a patchout audio transformer," DCASE2022 Challenge, Tech. Rep., 2022.

[27] F. Schmid, K. Koutini, and G. Widmer, "Efficient large-scale audio tagging via transformer-to-cnn knowledge distillation," in *ICASSP*. IEEE, 2023.

PRETRAINING REPRESENTATIONS FOR BIOACOUSTIC FEW-SHOT DETECTION USING SUPERVISED CONTRASTIVE LEARNING

Ilyass Moummad[1], *Romain Serizel*[2], *Nicolas Farrugia*[1],

[1] IMT Atlantique, Lab-STICC, UMR CNRS 6285, Brest, France,
{ilyass.moummad, nicolas.farrugia}@imt-atlantique.fr
[2] University of Lorraine, CNRS, Inria, Loria, 5400, Nancy, France,
{romain.serizel}@loria.fr

ABSTRACT

Deep learning has been widely used recently for sound event detection and classification. Its success is linked to the availability of sufficiently large datasets, possibly with corresponding annotations when supervised learning is considered. In bioacoustic applications, most tasks come with few labelled training data, because annotating long recordings is time consuming and costly. Therefore supervised learning is not the best suited approach to solve bioacoustic tasks. The bioacoustic community recasted the problem of sound event detection within the framework of few-shot learning, i.e. training a system with only few labeled examples. The few-shot bioacoustic sound event detection task in the DCASE challenge focuses on detecting events in long audio recordings given only five annotated examples for each class of interest. In this paper, we show that learning a rich feature extractor from scratch can be achieved by leveraging data augmentation using a supervised contrastive learning framework. We highlight the ability of this framework to transfer well for five-shot event detection on previously unseen classes in the training data. We obtain an F-score of 63.46% on the validation set and 42.7% on the test set, ranking second in the DCASE challenge. We provide an ablation study for the critical choices of data augmentation techniques as well as for the learning strategy applied on the training set. Our code is available on Github.[1]

Index Terms— Contrastive learning, representation learning, transfer learning, few-shot learning, bioacoustic sound event detection.

1. INTRODUCTION

Sound Event Detection (SED) is the task of recognizing sound events, including determining their onsets and offsets, as well as recognizing them. SED has many applications in bioacoustics such as monitoring of biodiversity, studying animal behavior and identifying species. Automatic bioacoustic SED provides significant value in our understanding of animal populations and their interactions, as well as individuals and their behaviors. Standard SED systems leverage supervised learning as well as semi-supervised learning (DCASE Challenge Task 4) and have shown strong results in the recent years [1, 2, 3]. Numerous works focused on bird vocalization due to availability of large bird sound datasets [4, 5]. BirdNet [6] is a notable work for bird monitoring, able to identify nearly one thousand bird species. The approach involves training a model in a supervised fashion using a vast dataset comprising over one million labeled bird recordings, using extensive data-preprocessing and data augmentation techniques.

However, such a large scale data collection for training systems is not always feasible in bioacoustics. The challenge lies not only in obtaining annotations but also in acquiring the audio samples themselves (e.g. for rare species or fields that are difficult to reach). As a consequence, bioacoustics SED is considered as a collection of numerous small-data problems, each requiring specialized systems for their individual solutions. Thus, the community of bioacoustics recasted bioacoustic SED as a few-shot learning (FSL) problem [7, 8].

FSL is a machine learning problem where a model has to learn to adapt to new classes of data unseen during training with only few labeled samples. FSL is adapted for many applications where acquisition or annotation is expensive or time consuming. The annual challenge on detection and classification of acoustic scenes and events (DCASE) organized a third edition for the task of few-shot bioacoustic sound event detection. This task focuses on SED in a FSL setting for mammal and bird vocalizations. The goal is to create a system that learns from five exemplar vocalizations (shots) to detect instances of these vocalizations in test audio recording.

Prototypical networks (ProtoNets) [9] were proposed as a baseline to solve the FSL problem of detecting animal sound events in the DCASE challenge [8]. ProtoNets, a meta-learning framework, have been state-of-the-art FSL audio systems in the recent years [10, 11]. The goal of meta-learning training is to develop models that can quickly adapt to new tasks with minimal data by simulating the test scenario within the training process. In Computer Vision, simple transfer learning methods have been shown to outperform meta-learning methods in FSL [12, 13] in several datasets such as MiniImageNet and TieredImagenet, in which case the domain shift between the training data and the few shot generalization is small enough. Here, we propose to test transfer learning to solve FSL problems for the bioacoustic SED [8].

As the generalization capability of the feature extractor is crucial for efficient transfer learning, we propose to train a model using the supervised contrastive learning framework (SCL) [14]. Numerous contrastive learning methods have been proposed in the self-supervised learning (SSL) literature [15, 16, 17], but the fundamental concept of pulling together positive pairs and pushing apart negative pairs remains the same across these approaches. The positive pairs consist of similar samples, while negative pairs consist of dissimilar samples. The selection of these pairs can be achieved through various means, such as data augmentation techniques [15] and/or utilizing class labels as in done in SCL [14]. The representa-

---

This work was co-funded by the AI@IMT program and the company OSO-AI.

[1]: `https://github.com/ilyassmoummad/dcase23_task5_scl`

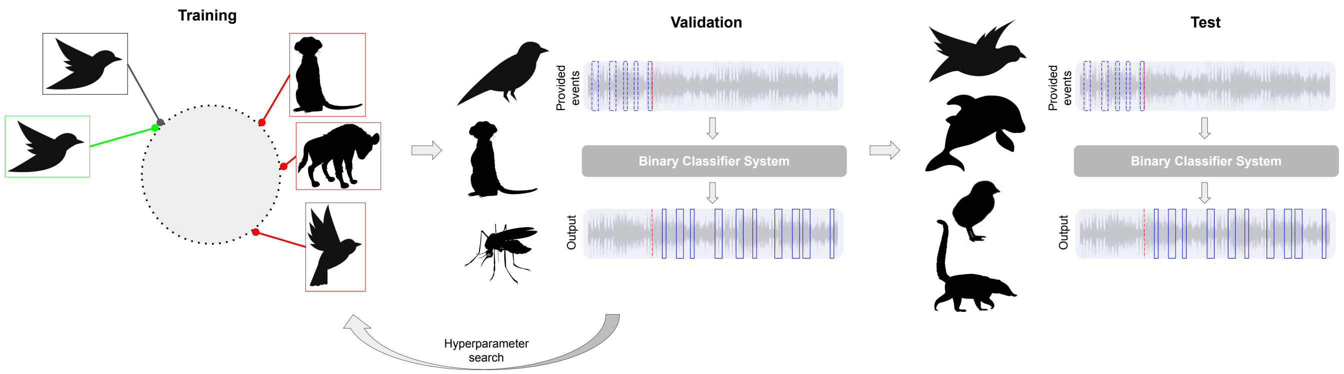


Figure 1: Overview of the proposed framework

tions learned using this framework have shown competitive transfer learning performance with SSL and cross-entropy (CE) learning on a variety of downstream tasks in vision [14]. In audio, the works of Moummad et al. [18] and Nasiri et al. [19] have demonstrated strong generalization capabilities of SCL.

Following the training of the feature extractor using SCL on the training set, the learned model is transferred to the validation set to conduct hyperparameter search. The optimal hyperparameter setting determined from this process is then employed on the test set for evaluation. In summary, our contribution revolves around the proposition of employing supervised contrastive learning to train a feature extractor that can be transferred to new few-shot bioacoustic sound event detection tasks.

2. METHOD

This section provides a comprehensive overview of the methodology employed in this study (Figure 1). Firstly, we present the SCL framework utilized for pre-training a good feature extractor model. Secondly, we describe the data augmentation techniques employed to enhance the diversity and robustness of the learned features. Finally, we detail our transfer learning strategy for adapting the pre-trained model to effectively tackle novel tasks.

2.1. Supervised Contrastive Learning

SCL consists in learning an embedding space in which the samples with the same class labels are close to each other, and the samples with different class labels are far from each other. Formally, a composition of an encoder f and a shallow neural network h called a projector (usually a MLP with one hidden layer) are trained to minimize the distances between representations of samples of the same class while maximizing the distances between representations of samples belonging to different class. After convergence, h is discarded, and the encoder f is used for transfer learning on downstream tasks. The supervised contrastive loss (SCL) is calculated as follows:

$$\mathcal{L}^{SCL} = \sum_{i\in I} \frac{-1}{|P(i)|} \sum_{p\in P(i)} \log \frac{\exp\left(\boldsymbol{z}_i \cdot \boldsymbol{z}_p/\tau\right)}{\sum\limits_{s\in S(i)} \exp\left(\boldsymbol{z}_i \cdot \boldsymbol{z}_s/\tau\right)} \quad (1)$$

where $i \in I = \{1...2N\}$ is the index of an augmented sample within a training batch, containing two views of each original sample. These views are constructed by applying a data augmentation function A twice to the original samples. $\boldsymbol{z}_i = h(f(A(\boldsymbol{x}_i))) \in \mathbb{R}^{D_P}$ where D_P is the projector's dimension. $P(i) = \{p \in I : y_p = y_i\}$ is the set of indices of all positives in the two-views batch distinct from i sharing similar label with i. $|P(i)|$ is its cardinality, $S(i) = \{s \in I : s \neq i\}$, the $\cdot$ symbol denotes the dot product, and $\tau \in \mathbb{R}^{+*}$ is a scalar temperature parameter that controls the penalty strength on hard negative samples.

2.2. Data Augmentation

Data augmentation is crucial for learning a good feature extractor as advocated by the SSL literature [15, 14, 16, **?**]. To this end, we adopt several augmentation modules derived from the audio representation learning domain [17, 20, 21]. The following augmentations are sequentially applied in the prescribed order and are iteratively employed twice on the same data, with the exception of spectrogram mixing, which is exclusively applied to a single view (based on our experimental findings, this configuration demonstrated superior performance). To demonstrate the significance of each augmentation technique, an ablation study is conducted in the subsequent section.

- Spectrogram mixing: we add background sounds using random samples from the same batch. The mixing follows: $\hat{x_1} = \alpha x_1 + (1-\alpha)x_2$. where $\hat{x_1}$ is considered as a view of x_1 and x_2 is a random sample from the batch.
- Frequency shift: we approximate frequency shift by shifting the spectrogram upwards by few bands.
- Random crop: we crop a patch from the spectrogram along the time axis.
- Spectrogram resize: this augmentation is applied after the crop to restore the spectrogram to its original size.
- Power gain: we attenuate the power of the spectrogram by multiplying it with a coefficient sampled uniformly between 0.75 and 1.
- Additive white Gaussian noise: we add a small additive white Gaussian noise to the view

2.3. Transfer Learning

After training the feature extractor, we transfer the model to the validation and test tasks. Each audio file is treated independently as a separate SED problem (as the challenge rules specify). We extract the features of the five positive annotated prototypes (shots) indicating the occurrence of the event of interest. We select intervals preceding the positive events as for the negative prototypes indicating the absence of the event. We train a binary classifier on these two prototypes using cross-entropy loss. The encoder layers can be either frozen or fine-tuned. We use a sliding window along the audio file (starting from the end of the fifth positive shot) to select queries for making predictions. The class activity is determined independently in each query window using the classifier. The onsets and offsets decision is made based on the precise moment when the label for the window transitions from a negative class to a positive class and from a positive class to a negative class, respectively.

3. EXPERIMENTS

3.1. Data

The bioacoustic few-shot sound event detection DCASE task development set consists of a training set and a validation set, for more details we refer the reader to the description of the task in 2022 [8] as these sets did not change from the previous edition.

3.1.1. Training

We train our system on the official training set. We select all the positively annotated segments within each audio file. We compute Mel spectrogram features with a FFT of size 512, a hop length of 128, a number of mels of 128 and a sampling rate of 22.05 kHz. Each positive annotated segment from the training set is chunked into patches of length 200 ms with 100 ms overlap. We apply min-max normalization on each patch.

3.1.2. Validation and test

For each audio file, we extract the first five positively annotated segments. The duration of these segments varies due to the wide range of animals and classes covered by the dataset. Following the approach proposed by Tang et al. [22], we determine the window length based on the mean duration of the events in the file. To compute Mel spectrogram features, we employ identical parameters and normalization technique as those used during the training phase. The shift size equals to half of the window length to predict the class for each query window along the remaining duration of the audio.

3.2. Model

We use a ResNet [23] consisting of three blocks, each comprising three convolutional layers. The feature maps of these convolutions have sizes of 64, 128, and 256, respectively. Following each convolutional layer, we apply batch normalization and a leaky rectified linear unit (ReLU) activation function. Max pooling operations are performed after each block. Specifically, we employ a 2x2 kernel for the first and second blocks, while for the third block, we use a 1x2 kernel. This choice is made to preserve frequency information by avoiding excessive pooling of the frequency bands, as suggested by Hertkorn [24].

To ensure consistent output dimensions despite varying input lengths, we incorporate adaptive max pooling at the end of the network. This pooling operation is configured to yield a desired output size of (8, 1), resulting in a latent vector of size 8 x 256 = 2048. A MLP projector is added, consisted of a hidden layer with a dimension of 2048 and an output layer with a dimension of 512.

3.3. Training details

3.3.1. Data augmentation

The spectrogram mixing coefficient α is sampled from a $\beta(5, 2)$ distribution. The frequency shift size is uniformly sampled between 0 and 10. The crop size (i.e. how much total duration is kept from the original audio) in the Random crop augmentation is uniformly sampled between 60% and 100%. Power gain augmentation is achieved by multiplying the mel spectrogram with a coefficient uniformly sampled between 0.75 and 1. The additive white Gaussian noise is incorporated by adding noise with a mean of zero and a variable standard deviation, which is uniformly chosen between 0 and 0.1.

3.3.2. Training and evaluation

We train our model from scratch on the training set using SCL framework with a temperature τ = 0.06 using SGD optimizer with a batch size of 128, a learning rate of 0.01 with a cosine decay schedule, momentum of 0.9, and a weight decay of 0.0001 for 50 epochs. After training, we discard the MLP projector and transfer the encoder to the validation and test sets by training a linear binary classifier on the pretrained representations. In this phase we used random resized crop along the time axis with a crop size ranging from 90% to 100% of the original size. We submitted four distincts systems to the challenge : freezing all pretrained layers (Frozen), or finetuning the last, two last and all layers (FineTune-1, FineTune-2 and Finetune-3). We optimize our systems using Adam optimizer with a learning rate of 0.01 for 20 epochs for the first system, and 40 epochs with a learning rate of 0.001 for the others. The selection of these hyperparameters is based on evaluation conducted on the validation set.

3.4. Results

The performance of our four systems on the validation set is presented in Table 1. For PB dataset, where events are short (therefore only few patches are available, because we divide longer events into multiple chunks), the first system outperforms the others, indicating that fine-tuning degrades the performance when only few positive patches are present. Conversely, for the HB dataset, where events tend to be longer, the third and fourth systems outperform the others. This indicates that finetuning a greater number of layers is advantageous when more positive patches are present. The second system demonstrates satisfactory performance across all datasets, outperforming the other systems across all datasets with a max F1 score of 63.46%. It is important to note that our results on the validation set exhibit significant variability, primarily attributed to the instability of our proposed cross-entropy adaptation strategy. We acknowledge this limitation and plan to address it in future work.

Table 2 displays the performance scores of our systems on the test sets. Remarkably, the ranking order of these systems on the test set aligns with that observed on the validation set. This consistency further validates the robustness and generalizability of our models across different datasets.

Table 1: Performance of different systems on the validation set; freezing all layers, fine-tuning one, two or all three layers.

System	Precision	Recall	F1-score	HB			ME			PB		
				Pr	Re	F1	Pr	Re	F1	Pr	Re	F1
Frozen	71.41	55.19	62.26	77.14	81.57	79.29	65.45	69.23	67.28	72.64	**36.17**	**48.29**
FineTune-1	**73.93**	**55.59**	**63.46**	**82.95**	82.32	82.63	67.69	84.61	75.21	**72.72**	33.33	45.71
FineTune-2	72.90	55.14	62.79	79.73	89.72	84.43	**74.60**	**90.38**	**81.73**	65.57	31.06	42.19
FineTune-all	67.08	51.58	58.32	81.20	**91.38**	**85.99**	58.75	**90.38**	71.21	65.00	27.65	38.80

*We highlight in bold the best scores for each metric

Table 2: F-score on the test sets of the different submissions

	F-score
Frozen	35.6% (35.3 - 36.0)
FineTune-1	**42.7% (42.2 - 43.1)**
FineTune-2	38.3% (37.9 - 38.7)
FineTune-all	34.4% (33.9 - 34.8)

*with 95% confidence interval

3.5. Ablation study

Table 3 presents our ablation study on data augmentation. Additionally, Table 4 compares pre-training methods : SCL, cross-entropy training (CE), and the self-supervised training method SimCLR [15], which has the same formula as SCL but without positive label pairs. We perform these studies on the validation set using the first system Frozen, where we freeze all layers, as it better captures the impact of the pre-training strategy. We use the same hyperparameter setting described in 3.3 for all experiments except for CE training where we use a learning rate of 0.0001 after thorough exploration. Additionally, we modify the training duration for SimCLR, extending it to 100 epochs. This adjustment is made to account for the longer training requirements typically associated with self-supervised approaches. To ensure reliable results, we trained the model five times on the training set and conducted five evaluations for each run, resulting in a total of twenty-five runs per experiment.

Table 3: Ablation of Data augmentation on the validation set

DA removed	Mean	[Min, Max]
Frozen (with all DAs)	56.47	[49.37, 62.39]
- Spectrogram mixing	56.59	[47.99, 64.65]
- Frequency shift	58.60	[49.73, 66.18]
- Time stretch	55.68	[49.20, 62.83]
- Power gain	56.02	[47.04, 63.01]
- Additive noise	**59.04**	**[52.68, 67.47]**

*Best score is highlighted in bold

The analysis presented in Table 3 indicates that certain data augmentation techniques have a negative impact on the model's performance. Surprisingly, these effects were not evident during the challenge submission due to the limited number of experiments conducted at that time. Notably, the data augmentation setting that yielded the highest score was the setting without the additive white Gaussian noise to the spectrogram. This finding suggests that this particular augmentation strategy was either enforcing an invariance that is not beneficial for the downstream task at hand, or that the task becomes hard given the small size of the training dataset.

We can observe from the results of Table 4 that SCL consistently outperforms both SimCLR and CE frameworks for transfer learning. The superior performance of SCL highlights its efficacy in capturing discriminative features. These findings emphasize the importance of incorporating SCL as a powerful framework for advancing feature representation learning, particularly for enhancing transferability in downstream tasks.

Table 4: Ablation of the pretraining methods on the validation set

Method	Mean	[Min, Max]
CE	51.96	[43.013-57.42]
SimCLR	50.89	[39.28-57.41]
SCL	**56.27**	**[49.37, 62.39]**

*Highest F-score is highlighted in bold

4. DISCUSSION AND PERSPECTIVES

In this study, we have provided a comprehensive description of a simple approach for bioacoustic few-shot sound event detection. We have detailed the methodology behind the systems we developed and submitted for the DCASE 2023 challenge task five. Our approach involves pretraining a feature extractor using supervised contrastive learning and data augmentation on the training set, followed by training binary classifiers on positive and negative prototypes for each audio file in the validation/evaluation sets. We proposed four systems. The first system, which utilized a linear classifier on frozen representations, demonstrated the robustness and transferability of the learned features. When fine-tuning the last layer (the second system) or the last two layers (the third system), the performance is increased. However, our current adaptation strategy, involving training classifiers on available shots, showed performance instability. We also note the gap in performance between the validation and the test sets. HB validation dataset is made of controlled lab recordings, which may make the detection easier, while PB recordings are in the wild with noisy background. Settings of the test set are more close to PB than HB [25]. To address the limitation and instability of our approach, future work will explore more effective adaptation techniques such as meta-learning. Notably, the winning systems in the 2022 and 2023 editions of the DCASE bioacoustic few-shot sound event detection challenge (Tang et al. [22]; Du et al. [26]) employed a frame-level approach, offering a higher time resolution capability compared to our window-level approach. Exploring the frame-level approach, as well as a proposal-based approach [27] for detecting variable length temporal regions of interest, which has not been previously investigated in this task, will be considered for future research. Combining representation learning (meta-learning, self-supervised learning, or supervised learning) is a promising direction for learning useful representation leveraging knowledge from large data, that can transfer well to new tasks.

5. REFERENCES

[1] T. K. Chan and C. S. Chin, "A comprehensive review of polyphonic sound event detection," *IEEE Access*, vol. 8, pp. 103 339–103 373, 2020.

[2] A. Dang, T. H. Vu, and J.-C. Wang, "A survey of deep learning for polyphonic sound event detection," in *2017 International Conference on Orange Technologies (ICOT)*. IEEE, 2017, pp. 75–78.

[3] N. Turpault, R. Serizel, A. P. Shah, and J. Salamon, "Sound event detection in domestic environments with weakly labeled data and soundscape synthesis," in *Workshop on Detection and Classification of Acoustic Scenes and Events*, 2019.

[4] D. Stowell, M. D. Wood, H. Pamuła, Y. Stylianou, and H. Glotin, "Automatic acoustic detection of birds through deep learning: the first bird audio detection challenge," *Methods in Ecology and Evolution*, vol. 10, no. 3, pp. 368–380, 2019.

[5] M. Lasseck, "Acoustic bird detection with deep convolutional neural networks." in *DCASE*, 2018, pp. 143–147.

[6] S. Kahl, C. M. Wood, M. Eibl, and H. Klinck, "Birdnet: A deep learning solution for avian diversity monitoring," *Ecological Informatics*, vol. 61, p. 101236, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1574954121000273

[7] I. Nolasco, S. Singh, V. Morfi, V. Lostanlen, A. Strandburg-Peshkin, E. Vidaña-Vila, L. Gill, H. Pamuła, H. Whitehead, I. Kiskin, *et al.*, "Learning to detect an animal sound from five examples," *arXiv preprint arXiv:2305.13210*, 2023.

[8] I. Nolasco, S. Singh, E. Vidana-Villa, E. Grout, J. Morford, M. Emmerson, F. Jensens, H. Whitehead, I. Kiskin, A. Strandburg-Peshkin, *et al.*, "Few-shot bioacoustic event detection at the dcase 2022 challenge," *arXiv preprint arXiv:2207.07911*, 2022.

[9] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," *Advances in neural information processing systems*, vol. 30, 2017.

[10] C. Heggan, S. Budgett, T. Hospedales, and M. Yaghoobi, "Metaaudio: A few-shot audio classification benchmark," in *Artificial Neural Networks and Machine Learning–ICANN 2022: 31st International Conference on Artificial Neural Networks, Bristol, UK, September 6–9, 2022, Proceedings, Part I*. Springer, 2022, pp. 219–230.

[11] Y. Wang, N. J. Bryan, J. Salamon, M. Cartwright, and J. P. Bello, "Who calls the shots? rethinking few-shot learning for audio," in *2021 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. IEEE, 2021, pp. 36–40.

[12] Y. Tian, Y. Wang, D. Krishnan, J. B. Tenenbaum, and P. Isola, "Rethinking few-shot image classification: a good embedding is all you need?" in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16*. Springer, 2020, pp. 266–282.

[13] Y. Bendou, Y. Hu, R. Lafargue, G. Lioi, B. Pasdeloup, S. Pateux, and V. Gripon, "Easy—ensemble augmented-shot-y-shaped learning: State-of-the-art few-shot classification with simple components," *Journal of Imaging*, vol. 8, no. 7, p. 179, 2022.

[14] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan, "Supervised contrastive learning," *Advances in neural information processing systems*, vol. 33, pp. 18 661–18 673, 2020.

[15] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International conference on machine learning*. PMLR, 2020, pp. 1597–1607.

[16] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin, "Unsupervised learning of visual features by contrasting cluster assignments," *Advances in neural information processing systems*, vol. 33, pp. 9912–9924, 2020.

[17] E. Fonseca, D. Ortego, K. McGuinness, N. E. O'Connor, and X. Serra, "Unsupervised contrastive learning of sound event representations," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 371–375.

[18] I. Moummad and N. Farrugia, "Learning audio features with metadata and contrastive learning," *arXiv preprint arXiv:2210.16192*.

[19] A. Nasiri and J. Hu, "Soundclr: contrastive learning of representations for improved environmental sound classification," *arXiv preprint arXiv:2103.01929*, 2021.

[20] L. Wang and A. v. d. Oord, "Multi-format contrastive learning of audio representations," *arXiv preprint arXiv:2103.06508*, 2021.

[21] D. Niizumi, D. Takeuchi, Y. Ohishi, N. Harada, and K. Kashino, "Byol for audio: Exploring pre-trained general-purpose audio representations," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 31, pp. 137–151, 2022.

[22] J. Tang, Z. Xueyang, T. Gao, D. Liu, X. Fang, J. Pan, Q. Wang, J. Du, K. Xu, and Q. Pan, "Few-shot embedding learning and event filtering for bioacoustic event detection technical report," DCASE2022 Challenge, Tech. Rep., June 2022.

[23] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[24] M. Hertkorn, "Few-shot bioacoustic event detection : Don ' t waste information technical report," DCASE2022 Challenge, Tech. Rep., June 2022.

[25] I. Nolasco, S. Singh, V. Morfi, V. Lostanlen, A. Strandburg-Peshkin, E. Vidaña-Vila, L. Gill, H. Pamuła, H. Whitehead, I. Kiskin, *et al.*, "Learning to detect an animal sound from five examples," *arXiv preprint arXiv:2305.13210*, 2023.

[26] G. Yan, R. Wang, L. Zou, J. Du, Q. Wang, T. Gao, and X. Fang, "Multi-task frame level system for few-shot bioacoustic event detection," DCASE2023 Challenge, Tech. Rep., June 2023.

[27] P. Wolters, C. Daw, B. Hutchinson, and L. Phillips, "Proposal-based few-shot sound event detection for speech and environmental sounds with perceivers," *arXiv preprint arXiv:2107.13616*, 2021.

INCREMENTAL LEARNING OF ACOUSTIC SCENES AND SOUND EVENTS

Manjunath Mulimani, Annamaria Mesaros

Computing Sciences, Tampere University, Tampere, Finland
{manjunath.mulimani, annamaria.mesaros}@tuni.fi

ABSTRACT

In this paper, we propose a method for incremental learning of two distinct tasks over time: acoustic scene classification (ASC) and audio tagging (AT). We use a simple convolutional neural network (CNN) model as an incremental learner to solve the tasks. Generally, incremental learning methods catastrophically forget the previous task when sequentially trained on a new task. To alleviate this problem, we propose independent learning and knowledge distillation (KD) between the timesteps in learning. Experiments are performed on TUT 2016/2017 dataset, containing 4 acoustic scene classes and 25 sound event classes. The proposed incremental learner first solves the ASC task with an accuracy of 94.0%. Next, it learns to solve the AT task with an F1 score of 54.4%. At the same time, its performance on the previous ASC task decreases only by 5.1 percentage points due to the additional learning of the AT task.

***Index Terms*—** Incremental learning, independent learning, knowledge distillation, acoustic scene classification, audio tagging

1. INTRODUCTION

The natural learning system of humans incrementally learns new concepts over time without forgetting the previously learned ones. This process of learning is known as continuous, incremental, or lifelong learning. In contrast, deep learning-based systems have the ability to learn a task efficiently, but fine-tuning the same system with a new task tends to override the previously acquired knowledge. This leads to a phenomenon of deteriorating performance on previously learned tasks known as catastrophic forgetting. Developing a robust system that should not degrade its performance significantly on previous tasks as new tasks are added is currently a pursued problem in many domains.

Most of the studies reported in the literature on incremental learning operate on images, e.g. object detection [1, 2], image classification [3, 4], and semantic segmentation [5, 6]. A few works report on incremental learning of audio such as environmental sound classification (ESC) [7, 8], audio captioning [9], and fake audio detection [10]. However, these methods are restricted to solving an initial base task followed by N incremental tasks of the same problem (e.g. ESC). In addition, these methods were designed to have the same number of classes in the incremental tasks, an assumption that does not hold in practical scenarios. Furthermore, most of these methods use a small portion of data from the previous task during the training of the system on current task data and employ complex postprocessing methods to alleviate catastrophic forgetting [11].

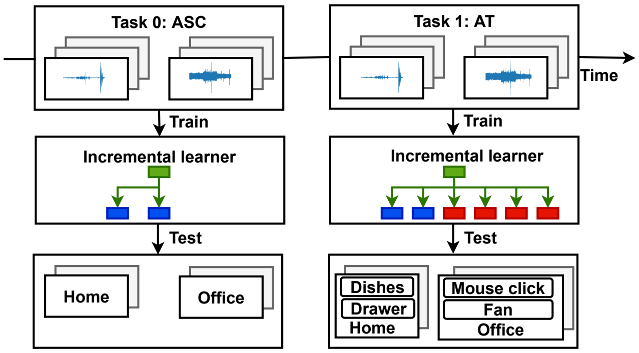


Figure 1: Incremental learning of distinct tasks: acoustic scene classification (ASC) and audio tagging (AT). Our incremental learner learns acoustic scene classes initially (blue units) and sound event classes incrementally (red units). After the learning process of each task, a learner is evaluated on classes of all tasks learned so far.

In this work, we propose an incremental learning method to solve distinct tasks over time: acoustic scene classification (ASC) and audio tagging (AT) (see Fig. 1), to simulate the scenario of new information becoming available at a later time for the same audio material. In this case, the incremental learner first learns the acoustic scenes and then learns more detailed characterization of the acoustic content of the scene, i.e. the sound events active in the given acoustic scene. When learning the sound event classes, the same audio material is used, but the acoustic scene labels are no longer available in the learning process.

The proposed work is based on the class incremental approach [12] and uses a single classifier to learn both acoustic scenes and sound events rather than using a separate classifier for each task, as done in [13]. Using the same classifier makes the problem more challenging, because acoustic scene and sound event classes compete in a single classifier. In comparison [13] employs information on the task identity to get output from a specific classifier at the prediction step. Inspired by different incremental learning methods for image classification [3, 4, 12–14], we propose a simple CNN-based incremental learner to solve distinct tasks over time.

The main contributions of this work are summarized below.

- We design an independent learning (IndL) mechanism that allows a classifier to learn different tasks effectively.
- We combine IndL with Kullback-Leibler (KL) divergence-based distillation loss to learn new sound events while preserving the knowledge of acoustic scene classes.
- We conduct experiments on both two-step ASC-AT and three-step ASC-ASC-AT incremental setups to analyse the behavior of the proposed incremental learner over time.

The rest of the paper is organized as follows. Section 2 introduces the proposed framework for incremental learning of acoustic scenes and sound events tasks. Section 3 presents the different ex-

---

This work was supported in part by Academy of Finland grant 332063 “Teaching machines to listen”. The authors wish to thank CSC-IT Centre of Science Ltd., Finland, for providing computational resources.

periments that compare incremental learning with multi-task learning and separate models for the same tasks. Finally, conclusions and future work are given in Section 4.

2. INCREMENTAL LEARNING

In our incremental learning setting, a set of ASC and AT tasks $\{\tau_0, \tau_1, \ldots, \tau_t\}$ is presented to a learner sequentially in incremental time steps t. The isolated task $\tau_t = \{(\mathbf{x}_i^{\tau_t}, \mathbf{y}_i^{\tau_t}) | 1 \leq i \leq m\}$ presented at time step t is composed of input features $\mathbf{x}_i^{\tau_t}$ and corresponding one-hot (for ASC) or multi-hot (for AT) ground truth label vectors $\mathbf{y}_i^{\tau_t} \in \{0,1\}^{C_t}$. C_t denotes the number of classes in tasks up to and including task τ_t. The two distinct ASC and AT tasks use the same audio clips but none of the tasks share the class labels. Specifically, the learner does not have access to labels of the τ_0 while learning the τ_1. Typically, this class imbalance makes the learner's predictions biased to focus on the classes in the current task and catastrophically forget the classes of the previous task.

In this work, our goal is to build a learner $\mathcal{P}^{\tau_t}$, which can solve all the tasks learned so far. A learner $\mathcal{P}^{\tau_t}$ is a deep network that includes a feature extractor $\mathcal{F}_\theta^{\tau_t}$ (parameterized by weights θ) and a fully connected layer $\mathcal{F}_\phi^{\tau_t}$ (parameterized by weights ϕ) for classification. Output logits of the network on a given input $\mathbf{x}$ are obtained by $\mathbf{o}(\mathbf{x}) = \mathcal{F}_\phi^{\tau_t}(\mathcal{F}_\theta^{\tau_t}(\mathbf{x}))$.

Generally, incremental learners learn to solve one initial base task followed by similar incremental tasks, for example like the setup in [7] where the initial task was learning 30 sound classes, then learning sets of 10 sound classes in incremental time steps for the same classification task. However, in this work, we solve distinct incremental tasks: single-label ASC and multi-label AT. We experiment with two scenarios. In the first scenario, the learner solves ASC in an initial step ($\tau_0 =$ ASC) and AT in the next time step ($\tau_1 =$ AT) as depicted in Fig. 1 (hereafter referred to as ASC-AT task). In the second scenario, the learner solves ASC in an initial step ($\tau_0 =$ ASC), and ASC and AT in subsequent incremental time steps ($\tau_1 =$ ASC and $\tau_2 =$ AT, hereafter referred to as ASC-ASC-AT).

2.1. Incremental ASC-AT learning

In the initial time step $t = 0$, a learner $\mathcal{P}^{\tau_0}$ learns $\mathcal{F}_\phi^{\tau_0}$ to classify the acoustic scene classes of a task τ_0. $\mathcal{P}^{\tau_0}$ is trained using cross-entropy loss $\mathcal{L}^{CE}$ computed using softmax σ over logits $\mathbf{o}$ as:

$$\mathcal{L}^{CE} = -\sum_{k=1}^{C} \mathbf{y}_k^{\tau_0} \cdot \log(\sigma(\mathbf{o}_k)), \tag{1}$$

In the incremental time step $t = 1$, a new learner $\mathcal{P}^{\tau_1}$ is initialized by previous learner $\mathcal{P}^{\tau_0}$. The classifier $\mathcal{F}_\phi^{\tau_1}$ of $\mathcal{P}^{\tau_1}$ is extended to learn sound event classes of a task τ_1 by adding new output units. The output logits of $\mathcal{P}^{\tau_1}$ comprise $\mathbf{o} = \{\mathbf{o}^{old}, \mathbf{o}^{new}\}$. $\mathbf{o}^{old}$ and $\mathbf{o}^{new}$ denote the logits of acoustic scene classes and sound event classes respectively. We propose an independent learning process (IndL) through which the entire $\mathcal{P}^{\tau_1}$ is trained using separate losses for $\mathbf{o}^{old}$ and $\mathbf{o}^{new}$ logits (see Fig. 2).

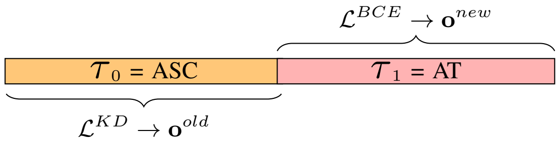

$\mathcal{L}^{BCE} \to \mathbf{o}^{new}$
$\mathcal{T}_0$ = ASC
$\mathcal{T}_1$ = AT
$\mathcal{L}^{KD} \to \mathbf{o}^{old}$

Figure 2: Incremental ASC-AT learning; the different losses are calculated on logits of the previous ASC task and logits of the new AT task separately at the incremental time step $t = 1$.

A binary cross-entropy loss $\mathcal{L}^{BCE}$ is computed using sigmoid σ over logits of the novel acoustic event classes $\mathbf{o}^{new}$ only:

$$\mathcal{L}^{BCE} = -\sum_{k=C_{t-1}+1}^{C_t} \mathbf{y}_k^{\tau_1} \cdot \log(\sigma(\mathbf{o}_k^{new})) + (1 - \mathbf{y}_k^{\tau_1}) \cdot \log(1 - \sigma(\mathbf{o}_k^{new})), \tag{2}$$

where C_{t-1} denotes the number of classes up to and excluding task τ_1. This independent learning of weights ϕ of $\mathcal{F}_\phi^{\tau_1}$ of the novel sound event classes from previous acoustic scene classes is meant to reduce the catastrophic forgetting. To be more specific, IndL allows $\mathcal{P}^{\tau_1}$ to learn from Eq. (2) using only the $\mathbf{o}^{new}$ logits of the sound event classes, while $\mathbf{o}^{old}$ logits of the acoustic scene classes are handled separately. This independence of the two tasks is also evident in the different loss functions per task ($\mathcal{L}^{CE}$ vs $\mathcal{L}^{BCE}$).

A distillation loss $\mathcal{L}^{KD}$ is computed using Kullback-Leibler divergence ($\mathcal{D}_{KL}$) between $\mathbf{o}^{old}$ logits of current $\mathcal{P}^{\tau_1}$ learner and output logits of previous frozen $\hat{\mathcal{P}}^{\tau_0}$ learner:

$$\mathcal{L}^{KD} = \mathcal{D}_{KL}(\hat{\mathbf{v}} || \mathbf{v}), \tag{3}$$

where $\mathbf{v} = \sigma(\frac{\mathbf{o}^{old}}{T})$ denotes the $\mathbf{o}^{old}$ logits of $\mathcal{P}^{\tau_1}$ and $\hat{\mathbf{v}} = \sigma(\frac{\hat{\mathcal{P}}^{\tau_0}(\mathbf{x})}{T})$ denotes the logits of $\hat{\mathcal{P}}^{\tau_0}$. The σ is the softmax and T is the temperature hyperparameter to smooth the $\mathcal{L}^{KD}$. The $\mathcal{L}^{KD}$ acts as a forgetting constraint that penalizes the change concerning the output of the previous learner. Specifically, the learner $\mathcal{P}^{\tau_1}$ preserves the knowledge about the previous ASC task using $\mathcal{L}^{KD}$ and continues to learn the new AT task using $\mathcal{L}^{BCE}$. Therefore, $\mathcal{P}^{\tau_1}$ is trained using combined loss as:

$$\mathcal{L} = \mathcal{L}^{BCE} + \lambda \mathcal{L}^{KD}, \tag{4}$$

where λ denotes the weight of the loss which we adaptively set to $\Omega\sqrt{C_t - C_{t-1}/C_t}$ as per the recommendation of [14]. Ω is a constant. $C_t - C_{t-1}$ denotes the number of new sound event classes.

We use two more techniques reported in the literature for incremental learning of images. One is, that the learning rate (LR) is reduced in incremental time steps, as done in [3]. This was shown to improve the transfer of knowledge from the old to the new learner and mitigate the adverse effect of imbalanced data in incremental time steps. Another is the use of cosine normalization in the classifier $\mathcal{F}_\phi^{\tau_t}$ [14]. It was observed that the magnitudes of the weight and bias of the previous and current classes in $\mathcal{F}_\phi^{\tau_t}$ are significantly different. Cosine normalization restricts the values of input distributions to $[-1, 1]$ and eliminates the bias that arises due to the magnitude difference.

2.2. Incremental ASC-ASC-AT learning

In this case, the learner $\mathcal{P}^{\tau_0}$ learns to solve an additional ASC task using $\mathcal{L}^{CE}$ at the initial time step $t = 0$. In the incremental time

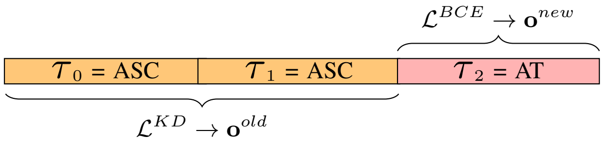

$\mathcal{L}^{BCE} \to \mathbf{o}^{new}$
$\mathcal{T}_0$ = ASC
$\mathcal{T}_1$ = ASC
$\mathcal{T}_2$ = AT
$\mathcal{L}^{KD} \to \mathbf{o}^{old}$

Figure 3: Incremental ASC-ASC-AT learning; the different losses are calculated on old ASC and new AT logits separately at an incremental time step $t = 2$.

step $t = 1$, $\mathcal{P}^{\tau_1}$ continues to learn a new ASC task in the absence of the old ASC task's data. Specifically, the two ASC tasks do not share audio clips nor scene classes. In the incremental time step $t = 1$, $\mathcal{L}^{CE}$ is computed independently using softmax σ over logits of the new acoustic scene classes $\mathbf{o}^{new}$ only, as per Eq. (1) and $\mathcal{L}^{KD}$ is used to hold the knowledge of old acoustic scenes, as per Eq. (3). $\mathcal{P}^{\tau_1}$ is trained using combined loss as:

$$\mathcal{L} = \mathcal{L}^{CE} + \lambda \mathcal{L}^{KD} \tag{5}$$

In the incremental time step $t = 2$, $\mathcal{P}^{\tau_2}$ learns the AT task (using $\mathcal{L}^{BCE}$) by preserving the knowledge of all old acoustic scenes (using $\mathcal{L}^{KD}$) as per the description given in 2.1 and illustrated in Fig. 3. For the AT task, the learner uses the same audio clips as the ASC task at $t = 1$.

3. EVALUATION AND RESULTS

3.1. Datasets

For ASC-AT, we use acoustic scenes and corresponding sound events from TUT 2016/2017 dataset [15, 16]. The dataset contains 192 minutes of audio recordings. Task 0 is composed of four acoustic scenes: home, residential area, city center, and office. Task 1 is composed of 25 sound events: bird singing, brakes squeaking, breathing, etc. Complete details about the data can be found in [15].

For ASC-ASC-AT, we use TUT Acoustic Scenes 2017 [17] and TUT 2016/2017. Task 0 is composed of 11 acoustic scenes: beach, bus, cafe/restaurant, car, forest path, grocery store, library, metro station, park, train, and tram. Tasks 1 and 2 are the ASC and AT from the previous experiment. The learner is trained and tested on official development and evaluation splits of the datasets in each step.

3.2. Implementation details

Input features in each time step are 40-dimensional log mel-band energies obtained from each audio segment in 40 ms frames with 50% overlap. The network architecture of the feature extractor $\mathcal{F}_{\theta}^{\mathcal{T}_t}$ includes three convolutional blocks, each consisting of two 3×3 convolutional layers, with batch-normalization and ReLU nonlinearity applied to each convolutional layer. 2×2 average pooling is applied to each convolutional block, and 20% dropout is applied after each average pooling to avoid overfitting. The number of feature maps of convolutional blocks is set to $\{16, 32, 64\}$. The flattened output of the last convolutional block is considered as the input to the cosine normalized fully-connected layer $\mathcal{F}_{\phi}^{\mathcal{T}_t}$, whose number of output units is equal to the number of classes in each time step.

The learner's network is trained using the SGD optimizer [18] with a momentum of 0.9 and a mini-batch size of 100 for 120 epochs. The initial learning rates for task 0 and incremental task(s) are set to 0.1 and 0.01 respectively. CosineAnnealingLR [18] scheduler is used to update the optimizer in every epoch. Other hyperparameters: T and Ω are empirically set to 2 and 5 respectively.

Method	KD	$t = 0$	$t = 1$	
		Task 0: ASC (Acc)	Task 1: AT (F1)	Task 0: ASC (Acc)
ASC	-	94.0	-	-
AT	-	-	53.0	-
Joint ASC-AT	-	72.0	50.4	-
Incremental ASC-AT	✗	94.0	54.4	84.1 (9.9↓)
Incremental ASC-AT	✓	94.0	54.4	88.9 (5.1↓)

Table 1: Incremental ASC (4 classes) and AT (25 classes) with and without KD, compared to individual and joint learning of tasks with a similar architecture. The value within () denotes the forgetting amount; ↓ indicates that lower is better.

3.3. Baseline systems and evaluation metrics

The performance of the proposed incremental ASC-AT system is compared with the individual ASC, AT, and joint ASC-AT baseline systems. The same network architecture as the incremental ASC-AT system is used in all the baseline systems for fair comparison. Individual ASC and AT systems solve only the ASC or AT task, respectively. A joint ASC-AT system is a multi-task system that is trained for ASC and AT tasks at the same time using cross-entropy loss and binary cross-entropy loss respectively, as proposed in [19]. The performance on ASC and AT tasks is evaluated using accuracy and F1 score (using a threshold of 0.5), respectively.

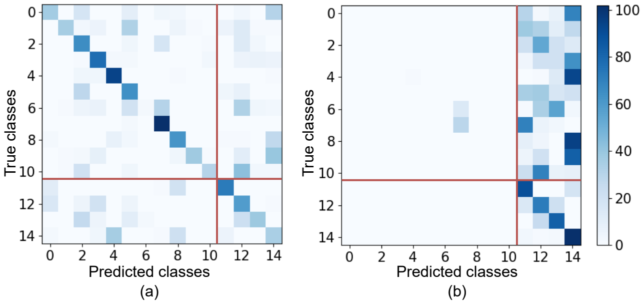
3.4. ASC-AT results

The experimental results provided in Table 1 compare the performance of the proposed incremental ASC-AT system with baseline systems. The learning of ASC and AT tasks at the same time as joint ASC-AT results in an overall performance lower than ASC and AT learned separately in different systems. Particularly, the accuracy of the ASC side in joint ASC-AT is significantly worse. This is also true with existing ASC-sound event detection (SED) multi-task models [16, 19].

On the other hand, the proposed incremental ASC-AT method can solve ASC and AT tasks with an accuracy of 84.1% (for the system without KD) and an F1 score of 54.4% respectively. The performance on the ASC task at the incremental time step is reduced by 9.9 percentage point (p.p.) as compared to the initial time step. Surprisingly, we see a small increase in the F1 score of the AT task as compared to the individual AT system. We hypothesize that this is due to the incremental learner already pre-trained using very relevant acoustic content (but different classes) in the initial time step, which may generate richer feature representations for the AT task. We observe that using KD is more efficient for preserving the knowledge of the previous task. The incremental ASC-AT system with KD outperforms the system without KD, having an average accuracy of 88.9% on the previous ASC task with only 5.1 p.p. reduction in performance.

Ablation setting			$t = 0$	$t = 1$		$t = 2$		
			Task 0:	Task 1: ASC (Acc)		Task 2:	Task 0 and 1: ASC (Acc)	
IndL	KD	LRs	ASC (Acc)	Overall	Task-wise	AT (F1)	Overall	Task-wise
✓	✓	{0.1, 0.1}	65.3	40.0	Task 0: 34.6 (30.7↓) Task 1: 54.9	53.0	38.8	Task 0: 32.9 (32.4↓) Task 1: 54.7 (0.2↓)
✓	✓	{0.01, 0.01}	**68.1**	50.1	Task 0: 54.3 (13.8↓) Task 1: 38.7	53.0	49.1	Task 0: 53.3 (14.8↓) Task 1: 37.3 (1.4↓)
✓	✓	{0.001, 0.001}	49.1	37.2	Task 0: 23.4 (25.7↓) Task 1: 77.0	43.0	34.0	Task 0: 17.7 (31.4↓) Task 1: 75.2 (1.8↓)
✓	✓	{0.01, 0.001}	**68.1**	49.3	Task 0: 57.7 (10.4↓) Task 1: 25.9	46.0	48.3	Task 0: 56.9 (11.2↓) Task 1: 24.1 (1.8↓)
✓	✓	{0.1, 0.01}	65.3	**53.8**	Task 0: **54.6** (10.7↓) Task 1: **53.1**	53.0	**52.2**	Task 0: **53.3** (12↓) Task 1: **49.1** (4↓)
✗	✗	{0.1, 0.01}	65.3	26.4	Task 0: 0.1 (65.2↓) Task 1: 85.0	53.0	23.2	Task 0: 0.1 (65.2↓) Task 1: 80.4 (4.6↓)

Table 2: Incremental ASC-ASC-AT with and without independent learning (IndL) and knowledge distillation (KD) using different LR combinations in initial and incremental steps with Task 0: 11 classes, Task 1: 4 classes, Task 2: 25 classes. For $t = 1$ and $t = 2$ the overall ASC accuracy represents performance over all 15 scene classes, and separate task-wise accuracy is provided (over the 11 classes of Task 0 and over the 4 classes of Task 1).



(a)

(b)

Figure 4: Confusion matrices w/ (a) and w/o (b) KD and IndL for ASC-ASC incremental learning (Task 0 to Task 1 of Table 2). Red lines separate the regions of new and old classes.

3.5. ASC-ASC-AT results

The experimental results for the ASC-ASC-AT scenario are presented in Table 2 and demonstrate the effectiveness of independent learning of acoustic scenes (as $\mathbf{o}^{old}, \mathbf{o}^{new}$) from tasks 0 and 1. Note that ASC-ASC tasks use different audio recordings, in contrast to ASC-AT using the same audio recordings with different labels.

As an ablation study, we experiment with a few combinations of LR, using the same and reduced LR in incremental tasks. Results show that the use of the same LR (0.1, 0.01, or 0.001) in initial and incremental time steps makes the learner either fail to learn new acoustic scene classes effectively (showing high-stability) or significantly forget the old acoustic scene classes (showing high-plasticity). This situation is also known as the stability-plasticity dilemma of a learner between new and old knowledge. Using a LR of 0.1 or 0.01 in all the steps does not affect the performance of AT task; LR 0.001 seems not suitable to solve either task, while 0.01 seems to be best for the initial ASC. Based on the results in Table 2, the combination using a LR of 0.1 and 0.01 provide a best balance between stability and plasticity, with a similar performance on the two ASC tasks at both $t = 1$ and $t = 2$.

Looking at learning with or without IndL, we observe a large difference in forgetting between tasks. A learner without IndL learns from Eq. (1) using all logits $\mathbf{o}$ (combination of $\mathbf{o}^{old}, \mathbf{o}^{new}$) and class labels $\mathbf{y}^{\mathcal{T}_1}$. Because when learning the incremental ASC task 1, the learner does not have access to the data $(\mathbf{x}^{\mathcal{T}_0}, \mathbf{y}^{\mathcal{T}_0})$ of previous ASC task 0, the values of the task 0 targets in $\mathbf{y}^{\mathcal{T}_1}$ are zero. This makes the learner forget the old acoustic scene classes because it sees no examples of them. Hence, the amount of forgetting reached 65.2 p.p. and accuracy dropped to almost zero on the acoustic scene classes of task 0 at time step 1 (last row in Table 2). In contrast, IndL of ASC task 1 with KD achieves an accuracy of 54.6% on the acoustic scene classes of task 0, with only a 10.7 p.p. forgetting. The two cases are illustrated in Fig. 4: without KD and IndL the network mostly predicts new classes (Fig. 4b), while using KD and IndL rebalances the output (Fig. 4a).

It is worth noting that the F1 score of the AT task at $t = 2$ is unaffected and it remains at 53.0%, the same as the individual AT system. This is because the AT task is always learned independently of acoustic scenes due to a different loss function, and takes advantage of a previous model trained on highly relevant acoustic material, even though the class information differs. This suggests that an independent learning mechanism is a suitable approach for learning all the tasks.

4. CONCLUSIONS AND FUTURE WORK

In this paper, we presented incremental ASC-AT and ASC-ASC-AT systems to solve distinct tasks over time. Results show that the performance of the ASC-AT system is close to the individual ASC and AT systems and outperforms joint ASC-AT learning with a similar size architecture. Independent learning of previous and current tasks with knowledge distillation significantly decreased the problem of catastrophic forgetting. In the presented setup, the AT task is always independent of the ASC task; hence, learning these tasks would not much disturb the performances of one another irrespective of their order (whether ASC-AT or AT-ASC). Future work includes more detailed ablation studies of the different choices used to reduce forgetting and to improve overall performance, such as the order, size and type of the incremental tasks, and the use of cosine normalization.

5. REFERENCES

[1] J. Kj, J. Rajasegaran, S. Khan, F. S. Khan, and V. N. Balasubramanian, “Incremental object detection via meta-learning,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 12, pp. 9209–9216, 2021.

[2] J.-M. Perez-Rua, X. Zhu, T. M. Hospedales, and T. Xiang, “Incremental few-shot object detection,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 13 846–13 855.

[3] S. Mittal, S. Galesso, and T. Brox, “Essentials for class incremental learning,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 3513–3522.

[4] H. Ahn, J. Kwak, S. Lim, H. Bang, H. Kim, and T. Moon, “SS-IL: Separated softmax for incremental learning,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 844–853.

[5] L. Yu, X. Liu, and J. Van de Weijer, “Self-training for class-incremental semantic segmentation,” *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–12, 2022.

[6] F. Cermelli, D. Fontanel, A. Tavera, M. Ciccone, and B. Caputo, “Incremental learning in semantic segmentation from image labels,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 4371–4381.

[7] Y. Wang, N. J. Bryan, M. Cartwright, J. P. Bello, and J. Salamon, “Few-shot continual learning for audio classification,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 321–325.

[8] B. Bayram and G. İnce, “An incremental class-learning approach with acoustic novelty detection for acoustic event recognition,” *Sensors*, vol. 21, no. 19, p. 6622, 2021.

[9] J. Berg and K. Drosos, “Continual learning for automated audio captioning using the learning without forgetting approach,” in *Workshop on Detection and Classication of Acoustic Scenes and Events (DCASE)*, 2021, pp. 140–144.

[10] H. Ma, J. Yi, J. Tao, Y. Bai, Z. Tian, and C. Wang, “Continual Learning for Fake Audio Detection,” in *INTERSPEECH*, 2021, pp. 886–890.

[11] K. Joseph, S. Khan, F. S. Khan, R. M. Anwer, and V. N. Balasubramanian, “Energy-based latent aligner for incremental learning,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 7452–7461.

[12] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, “ICaRL: Incremental classifier and representation learning,” in *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, 2017, pp. 2001–2010.

[13] Z. Li and D. Hoiem, “Learning without forgetting,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 12, pp. 2935–2947, 2017.

[14] S. Hou, X. Pan, C. C. Loy, Z. Wang, and D. Lin, “Learning a unified classifier incrementally via rebalancing,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 831–839.

[15] K. Imoto, N. Tonami, Y. Koizumi, M. Yasuda, R. Yamanishi, and Y. Yamashita, “Sound event detection by multitask learning of sound events and scenes with soft scene labels,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 621–625.

[16] D. A. Krause and A. Mesaros, “Binaural signal representations for joint sound event detection and acoustic scene classification,” in *European Signal Processing Conference (EUSIPCO)*, 2022, pp. 399–403.

[17] A. Mesaros, T. Heittola, A. Diment, B. Elizalde, A. Shah, E. Vincent, B. Raj, and T. Virtanen, “DCASE 2017 challenge setup: Tasks, datasets and baseline system,” in *Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE)*, 2017, pp. 85–92.

[18] I. Loshchilov and F. Hutter, “SGDR: Stochastic gradient descent with warm restarts,” in *International Conference on Learning Representations*, 2017. [Online]. Available: https://openreview.net/forum?id=Skq89Scxx

[19] N. Tonami, K. Imoto, M. Niitsuma, R. Yamanishi, and Y. Yamashita, “Joint analysis of acoustic events and scenes based on multitask learning,” in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. IEEE, 2019, pp. 338–342.

FREQUENCY & CHANNEL ATTENTION FOR COMPUTATIONALLY EFFICIENT SOUND EVENT DETECTION

Hyeonuk Nam, Seong-Hu Kim, Deokki Min, Yong-Hwa Park

Korea Advanced Institute of Science and Technology, South Korea
{frednam, seonghu.kim, minducky, yhpark}@kaist.ac.kr

ABSTRACT

We explore on various attention methods on frequency and channel dimensions for sound event detection (SED) in order to enhance performance with minimal increase in computational cost while leveraging domain knowledge to address the frequency dimension of audio data. We have introduced frequency dynamic convolution (FDY conv) in a previous work to release the translational equivariance issue associated with 2D convolution on the frequency dimension of 2D audio data. Although this approach demonstrated state-of-the-art SED performance, it resulted in a model with 150% more trainable parameters. To achieve comparable SED performance with computationally efficient methods for practicality, we explore on lighter alternative attention methods. In addition, we focus on attention methods applied to frequency and channel dimensions. Joint application Squeeze-and-excitation (SE) module and time-frame frequency-wise SE (tfwSE) to apply attention on both frequency and channel dimensions shows comparable performance to SED model with FDY conv with only 2.7% more trainable parameters compared to the baseline model. In addition, we performed class-wise comparison of various attention methods to further discuss various attention methods' characteristics.

Index Terms— sound event detection, computationally efficient, attention, frequency dimension, channel dimension

1. INTRODUCTION

Sound event detection (SED), which aims to recognize a target sound event class and corresponding time localization within a given audio clip, has potential to be applied in various applications such as automation, robotics and monitoring [1, 2, 3]. In order to recognize and locate sound events, we need strong pattern recognition tools. Recent advances in deep learning (DL) methods brought significant progress in SED [2, 3]. While most works directly applied DL methods from other domains to SED without modification, few works adapted DL methods to SED by thoroughly analysing unique characteristics of audio data and sound events.

Frequency dimension has to be carefully considered when applying DL methods on audio-related DL applications. It is shown by previous works that methods considering frequency dimension significantly improved SED performance [4, 5, 6]. SED has been heavily relying on convolutional recurrent neural networks (CRNN) based architectures [2, 3]. 2D convolution in CRNN assumes shift-invariance on both time and frequency dimensions thus enforces translational eqauivariance on both dimensions [4]. However, frequency is a shift-variant dimension where the same pattern sounds different when translated along the frequency dimension. At the same time, frequency exhibits loose shift-invariance within short frequency range thus slight pitch-shift does not harm auditory perception much. Thus frequency dimension is a delicate yet essential component to be considered for audio domain.

In a previous study, we introduced frequency dynamic convolution (FDY conv) to release translational equivariance by 2D convolution on the frequency dimension of 2D audio data to consider its shift-variant characteristic [4]. While FDY conv showed impressive performance on SED, it added 150% more parameters to the model. However, in order to apply SED on various real applications, we might need to implement SED on devices with limited specifications. Thus, there is a need for computationally efficient SED methods which is lighter but as competent as current state-of-the-art models. To address this limitation and improve the practicality of SED models, we explore various lighter attention methods to enhance SED performance more efficiently. We aim to achieve this by addressing the frequency and channel dimensions, since those are two emphasized dimensions in audio domain [4, 6, 7]. Thus we experiment with various attention methods on frequency and channel dimensions. The main contributions of this work are:

1. We explore various alternative attention methods which are computationally efficient for practicality, while considering channel and frequency dimensions to consider unique characteristics of audio domain.
2. Joint application of squeeze-and-excitation (SE) and proposed time-frame frequency-wise SE (tfwSE) to re-weight both channel and frequency dimensions shows comparable performance to state-of-the-art method while only adding model parameters by 2.7%.
3. We discuss the characteristics of various attention methods on SED to provide further insights for practical implementation.

The official implementation code is available on GitHub[1].

2. METHODS

While frequency dynamic convolution (FDY conv) showed state-of-the-art performance and have been widely adopted on SED [6, 8, 9, 10, 11, 12], it adds considerable number of trainable parameters to the networks due to multiple basis kernels [4, 13].

This work was supported by the Institute of Civil Military Technology Cooperation funded by the Defense Acquisition Program Administration and Ministry of Trade, Industry and Energy of Korean government under grant No. UM22409RD4, and Korea Research Institute of Ships and Ocean engineering a grant from Endowment Project of "Development of Open Platform Technologies for Smart Maritime Safety and Industries" funded by Ministry of Oceans and Fisheries(PES4880).

[1] https://github.com/frednam93/lightSED

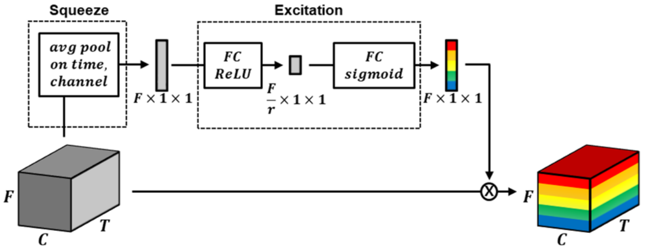


Figure 1: An illustration of frequency-wise Squeeze-Excitation.

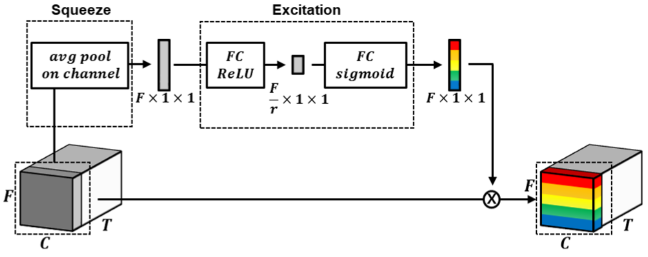


Figure 2: An illustration of time-frame frequency-wise Squeeze-Excitation on one time frame. tfwSE applies this procedure for every time frames.

Thus there remains a need for sufficiently well-performing model with fewer parameters for practical applications. Since FDY conv's strength comes from attention mechanism which selectively focus on important elements of the input, we explore other alternative attention methods to achieve comparable performance.

2.1. Variants of Squeeze-and-Excitation

One alternative computationally efficient attention method widely used is squeeze-and-excitation (SE) [14]. It has been widely applied to various CNN-based models for its light yet powerful performance. SE module is composed of squeeze operation and excitation operation. Squeeze operation averages output of 2D convolution on two dimensions except channel to obtain squeezed intermediate representation. Excitation operation applies two successive fully connected (FC) layers to obtain attention weights representing relative importance of each channel. The channels of convolution output is re-weighted by multiplying the attention weight [14]. When applied to 2D audio data, squeeze operation is applied to the convolution output by:

$$z_c = \frac{1}{F \times T} \sum_{f=1}^{F} \sum_{t=1}^{T} x_{cft} \tag{1}$$

where z_c is intermediate representation after squeeze operation on c-th channel and x_{cft} is the output by preceding 2D convolution with channel index c, frequency index f and time index t. F and T are frequency and time dimension sizes of 2D convolution output. The excitation operation is composed of two FC layers as follows:

$$\mathbf{s} = \sigma(\mathbf{W_2}\delta(\mathbf{W_1 z})) \tag{2}$$

where $\mathbf{s}$ is attention weight, also known as scale, which is multiplied to the output of preceding convolution. $\mathbf{z}$ is the intermediate representation vector. Both span channel dimension of size C. $\mathbf{W_1}$ and $\mathbf{W_2}$ are FC layers, δ refers to ReLU activation and σ refers to sigmoid function.

To apply attention-based re-weighting on frequency dimension, Thienpondt *et al.* proposed frequency-wise Squeeze-Excitation (fwSE) which applies SE on frequency dimension instead [15]. Thus, instead of pooling time and frequency dimensions, fwSE pools channel and time dimensions during squeeze operation as follows:

$$z_f = \frac{1}{C \times T} \sum_{c=1}^{C} \sum_{t=1}^{T} x_{cft} \tag{3}$$

The following excitation operation is the same as (2), just that two FC layers are applied on frequency dimensions instead. Then, obtained attention weight for each frequency bin is multiplied to corresponding frequency components of the preceding convolution output. Fig. 1 illustrates the fwSE mechanism.

Since frequency component varies over time, we propose time-frame fwSE (tfwSE) which applies fwSE on every time frames of input instead of time-averaged input. Thus, tfwSE only pools channel dimension in squeeze operation and then applies excitation operation on every time frames. The squeeze operation on time frame t can be expressed by following equation:

$$z_{ft} = \frac{1}{C} \sum_{c=1}^{C} x_{cft} \tag{4}$$

where z_{ft} is intermediate representation after squeeze operation. Then excitation is applied on frequency dimension on each time frame as follows:

$$\mathbf{s_t} = \sigma(\mathbf{W_2}\delta(\mathbf{W_1 z_t})) \tag{5}$$

where $\mathbf{s_t}$ is scale on time frame t and $\mathbf{z_t}$ is the intermediate representation vector corresponding to time frame t, both spanning channel dimension. Opposed to fwSE by Thienpondt *et al.* which applies frequency-wise attention weights evenly over time-dimension by referring to representative averaged information of the convolution output, proposed tfwSE applies frequency-wise attention weights for each time frame by referring to each individual contents within corresponding time frame [15]. While this could increase computation of excitation operator (fwSE applies excitation on one $\mathbf{z}$ per audio clip, while tfwSE applies excitation on T $\mathbf{z_t}$ per audio clip), it could help generalizing excitation operation on many time frames. The mechanism of tfwSE is illustrated in Fig. 2. This method was previously applied by our submission on detection and classification of acoustic scenes and events (DCASE) 2022 challenge task 3 as well, showing its performance [16]. Similarly, we could apply original SE on each time frame as well. We named it as time-frame SE (tSE). Note that this is not a time-wise version of SE like fwSE, as we do not apply SE by pooling channel and frequency dimensions during squeeze to leave time dimension. Instead, we pool frequency dimension only and apply SE on every time frame in similar way shown in Fig. 2.

2.2. Channel-Frequency Attention Methods

Li *et al.* [7] proposed C2D-Att for speaker verification which applies 2D convolution to obtain attention weights for both channel and frequency dimensions simultaneously. C2D-Att first pools time dimension by averaging, and then apply two consecutive 2D convolution modules to channel and frequency dimensions by introducing additional channel dimension which is increased to 8 and then back to 1. This results in channel-frequency attention weights which are multiplied to channel and frequency dimensions of preceding 2D convolution output. C2D-Att improves the speaker verification performance compared to fwSE by re-weighting channel and frequency dimensions simultaneously [7].

However, considering that channel dimension in CNN is permutable dimension where the convolution module's advantage capturing locality does not matter, it needs further verification if 2D convolution is the best option to apply channel and frequency attention on CNN. While CNN in C2D-Att applies 2D convolution kernel which finds local pattern across frequency and channel dimension, locality matters on frequency dimension only. Therefore, we experiment on joint application of attention on frequency and channel separately, without considering the locality of channel dimension using SE. To apply SE on two dimensions independently, we apply SE and tfwSE in series.

3. EXPERIMENTAL SETUPS

3.1. Model Architecture

The model architecture is based on CRNN model, composed of seven convolution layers followed by two bidirectional gated recurrent unit (GRU) then a FC layer. On the strong predictions, we apply class-wise median filter as post processing. In this work, the model using FDY conv replaced all 2D convolution except the first one. SE and C2D-Att modules are inserted after the activation and before the average pooling within the convolution blocks. They are applied on all convolution layers except the last layer in this work. It is because Hu *et al.* has shown that SE module applies almost constant attention weights at the last layer thus it merely affects the model [14].

3.2. Implementation Details

The overall implementation details follow the previous work [4], which could be referred on the official implementation code of which link is provided in the section 1. The experiments in this work are based on domestic environment sound event detection (DESED) dataset [3]. DESED is composed of synthesized strongly labeled dataset, real weakly labeled dataset and real unlabeled dataset for training and validation. For test, real validation dataset, which is strongly labeled, is used. We do not use any external dataset. We trained each model with single NVIDIA RTX Titan GPU. For the results listed in this paper, the metrics are based on the best score among total 24 models from 12 separate training runs.

DESED is composed of 10 second audio data with 16 kHz sampling rate. We extract mel spectrogram as the input feature for SED model. The settings for mel spectrograms are as follows: 2048 points for number of fft, 256 points for hop length, Hamming window for windowing function, and 128 mel bins. Data augmentation methods applied are frame shift [3], mixup [17], time masking [18] and FilterAugment [5]. Applying heavy data augmentation is crucial for training SED where real strongly labeled data is scarce [19]. As we use three levels of datasets, strongly labeled/weakly labeled/unlabeled dataset, we apply mean teacher to leverage unlabeled dataset [3, 20]. We apply FilterAugment with different random parameters on student and teacher model to train SED model robust against FilterAugment.

Table 1: Performance and computational cost comparison between the baseline, frequency dynamic convolution and various frequency and channel attention methods on DESED real validation dataset.

models	params	time	PSDS1	PSDS2	CB-F1
baseline	4.428M	3h 34m	0.409	0.641	0.520
+FDYconv	11.061M	6h 08m	**0.446**	**0.673**	0.525
+SE	4.537M	3h 49m	**0.435**	**0.654**	0.525
+tSE	4.537M	3h 52m	0.416	0.643	0.526
+fwSE	4.439M	3h 49m	0.411	0.634	0.522
+tfwSE	4.439M	3h 50m	0.415	0.638	0.509
+C2D-Att	4.429M	3h 53m	0.434	**0.659**	0.539
+tfwSE +SE	4.548M	4h 04m	0.437	0.650	0.532
+SE +tfwSE	4.548M	4h 06m	**0.442**	**0.657**	0.526

3.3. Evaluation Metrics

Main evaluation metric employed in this study is the polyphonic sound detection score (PSDS) [21], which considers the intersection between predictions and ground truth to decide if prediction is correct. PSDS also accounts for cross triggers induced by other sound events in the audio. PSDS utilizes area under curve (AUC) - receiver operating characteristic (ROC) curves, enabling comparison of sound event detection (SED) performances without the need for threshold optimization. In DCASE Challenge 2021, 2022 and 2023 Task 4, two variations of PSDS (PSDS1 and PSDS2) are utilized to evaluate SED systems [3]. PSDS1 places emphasis on precise time localization by limiting tolerance for intersection criteria, while PSDS2 prioritizes accurate classification by penalizing cross triggers more. Additionally, we use collar-base F1 score (CB-F1) [22] for class-wise performance comparison, as PSDS cannot be obtained for single sound event. Both PSDS and CB-F1 are ranged between zero and one, and value closer to one indicates better SED performance.

4. RESULTS AND DISCUSSION

4.1. Comparison of Attention Modules

Table 1 shows performance and computational cost of SED models with various frequency-wise and channel-wise attention methods. Computational costs are described by the number of trainable parameters representing model size and training time representing computational efficiency. Note that we aim to achieve computational efficiency as close to the baseline as possible and much less than FDY conv. For comparison, SED model with FDY conv is listed in table 1 as well. Note that the results for FDY-CRNN differ from the results in previous paper due to minor changes in setting. When we compare the performance of SED model with SE variants, we can observe that conventional SE definitely outperforms the baseline. On the other hand, fwSE only slightly outperform the baseline for PSDS1 while their PSDS2 is worse than the baseline. Considering that SE is proposed to re-weight channel dimension and each channel is independent from each other while frequency depends on other frequency bins, re-weighting appears to be more effective on channel dimension than on frequency dimension. In addition, considering the parameter increase in the model, SE has increased model size significantly more thus it involved more computational resource to the model. While SE has increased model size by $\sim$2.5%, fwSE has increased the model size by $\sim$0.25%. Proposed tfwSE is slightly better than fwSE in terms

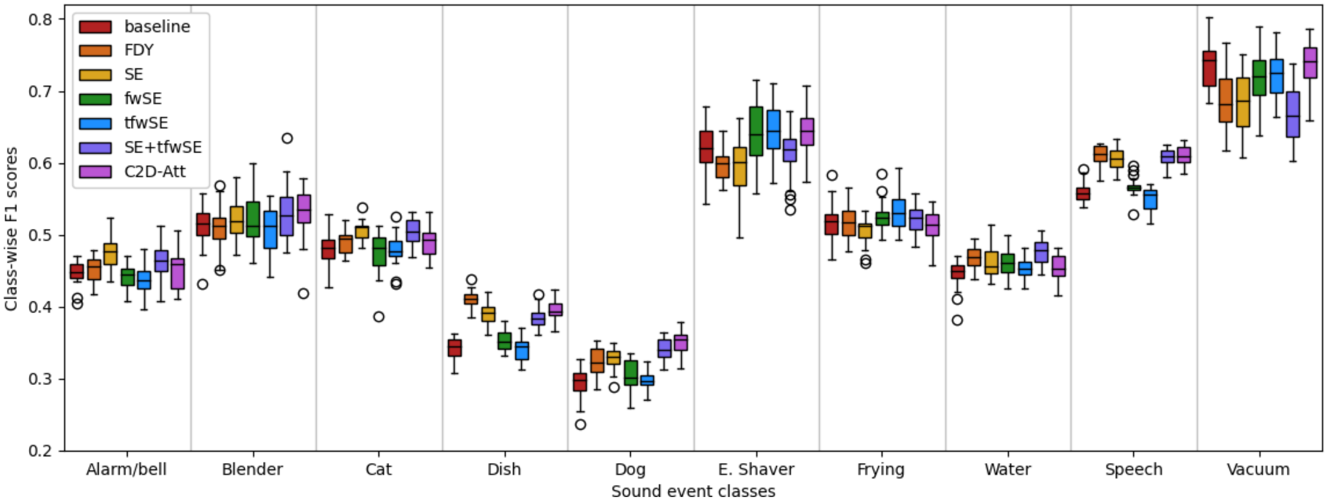


Figure 3: Box-plot of class-wise collar-base F1 scores by multiple models on DESED real validation dataset.

of both PSDS1 and PSDS2. On the other hand, tSE only slightly outperforms the baseline and performs worse than SE. While re-weighting frequency dimension on each time frame has improved frame-wise sound event classification of SED, this effect seems to be not so significant. One explanation to this could be the effect of bi-GRU which processes time-varying information. Likewise, temporal dynamic convolution which applies time-adaptive kernel performed worse than FDY conv on SED [23, 4]. On the other hand, tSE failed to improve SE. Re-weighting a dimension separately on each time frame was not as effective on channel dimension.

Results for methods applying attention simultaneously on channel and frequency dimensions, C2D-Att and joint applications of SE and tfwSE, are also listed in Table 1. C2D-Att shows descent performance comparable to SE, with less parameters compared to SE. In addition, joint application of SE followed by tfwSE shows improvement over SE. While joint applicatoin of SE after tfwSE shows similar performance to SE, we could still conclude that application of attention methods simultaneously on channel and frequency dimensions are effective. Furthermore, the combination of SE and tfwSE achieves comparable results to FDY conv in terms of PSDS1, reaching 99.1% of the PSDS1 by the model with FDY conv. Considering that high PSDS2 scores can be easily achieved using weakSED [19], we could regard that this model performs nearly as well as model with FDY conv. An interesting discovery is that while tfwSE degrades PSDS2 for the baseline model, the joint application of tfwSE after SE enhances PSDS2 compared to the model with SE alone. Moreover, considering that SE + tfwSE outperforms C2D-Att for PSDS1, 2D convolution considering locality of 2-dimensional patterns along frequency-channel dimensions is not as effective as separate consideration of channel and frequency dimensions. However, C2D-Att has advantage over SE + tfwSE in terms of the number of parameters which is increased by very small amount.

4.2. Class-wise Performance Comparison

In Fig. 3, class-wise collar-based F1 scores on multiple models are shown as box-plot. Each box-plot is composed of class-wise F1 scores by 24 models from 12 separate training runs. Consistent to table 1, SE performs better than fwSE and tfwSE on many classes in Fig. 3 as well. SE performed better than fwSE and tfwSE did on alarm/bell ringing, cat, dish, dog and speech while it performed worse on electric shaver, frying and vacuum cleaner. It seems that SE is stronger on transient and non-stationary sound events while it is weaker on quasi-stationary sound events, similar to FDY conv [4]. That is to say, while fwSE and tfwSE re-weight frequency dimension to address frequency dimension, they are stronger on quasi-stationary sound events than on non-stationary sound events. SE + tfwSE shows similar tendency with SE, but slightly better performance in general. Thus SE + tfwSE perform relatively better on non-stationary sound events and relatively worse on quasi-stationary sound events as well. C2D-Att also shows similar tendency with SE but it shows better performance on electric shaver and vacuum cleaner. Note that PSDS is an intersection-based score while the box-plots are based on collar-based score, there are slight discrepancy between table 1 and Fig. 3.

5. CONCLUSION

In conclusion, we experimented on various frequency and channel attention methods to enhance SED performance while minimizing computational cost. The study addressed the challenge of effectively addressing the frequency dimension of audio data by leveraging attention methods. The attention methods demonstrated comparable performance to the previous approach of FDY conv, while reducing the computational cost and improving practicality. In addition, we performed class-wise performance of the attention methods to further analyze the characteristics of SED models with different attention methods. Future research could aim to optimize the proposed attention methods by applying them jointly with FDY conv either to push the performance even more or to find balance between computational cost and the performance.

6. ACKNOWLEDGMENT

We would like to thank Junhyeok Lee from Supertone Inc. for valuable discussions.

7. REFERENCES

[1] T. Virtanen, M. D. Plumbley, and D. Ellis, *Computational Analysis of Sound Scenes and Events*, 1st ed. Springer Publishing Company, Incorporated, 2017, pp. 3–11, 71–77.

[2] E. Çakır, G. Parascandolo, T. Heittola, H. Huttunen, and T. Virtanen, "Convolutional recurrent neural networks for polyphonic sound event detection," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 6, pp. 1291–1303, 2017.

[3] N. Turpault. Dcase2021 task4 baseline. GitHub. Available: https://github.com/DCASE-REPO/DESED_task. [Online]. Available: https://github.com/DCASE-REPO/DESED_task

[4] H. Nam, S.-H. Kim, B.-Y. Ko, and Y.-H. Park, "Frequency Dynamic Convolution: Frequency-Adaptive Pattern Recognition for Sound Event Detection," in *Proc. Interspeech*, 2022.

[5] H. Nam, S.-H. Kim, and Y.-H. Park, "Filteraugment: An acoustic environmental data augmentation method," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022.

[6] S. Xiao, X. Zhang, and P. Zhang, "Multi-dimensional frequency dynamic convolution with confident mean teacher for sound event detection," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023.

[7] J. Li, Y. Tian, and T. Lee, "Convolution-based channel-frequency attention for text-independent speaker verification," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023, pp. 1–5.

[8] K. He, X. Shu, S. Jia, and Y. He, "Semi-supervised sound event detection system for dcase 2022 task 4," DCASE2022 Challenge, Tech. Rep., 2022.

[9] S. Suh and D. Y. Lee, "Data engineering for noisy student model in sound event detection," DCASE2022 Challenge, Tech. Rep., 2022.

[10] S. Xiao, "Pretrained models in sound event detection for dcase 2022 challenge task4," DCASE2022 Challenge, Tech. Rep., 2022.

[11] T. Khandelwal, R. K. Das, A. Koh, and E. S. Chng, "Leveraging audio-tagging assisted sound event detection using weakified strong labels and frequency dynamic convolutions," *arXiv preprint arXiv:2304.12688*, 2023.

[12] L. Xu, L. Wang, S. Bi, H. Liu, and J. Wang, "Semi-supervised sound event detection with pre-trained model," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023.

[13] Y. Chen, X. Dai, M. Liu, D. Chen, L. Yuan, and Z. Liu, "Dynamic convolution: Attention over convolution kernels," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2020.

[14] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[15] J. Thienpondt, B. Desplanques, and K. Demuynck, "Integrating frequency translational invariance in tdnns and frequency positional information in 2d resnets to enhance speaker verification," in *Proc. Interspeech*, 2021, pp. 2302–2306.

[16] B.-Y. Ko, H. Nam, S.-H. Kim, D. Min, S.-D. Choi, and Y.-H. Park, "Data augmentation and squeeze-and-excitation network on multiple dimension for sound event localization and detection in real scenes," DCASE2022 Challenge, Tech. Rep., 2022.

[17] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," in *International Conference on Learning Representations*, 2018.

[18] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition," in *Proc. Interspeech*, 2019, pp. 2613–2617.

[19] H. Nam, B.-Y. Ko, G.-T. Lee, S.-H. Kim, W.-H. Jung, S.-M. Choi, and Y.-H. Park, "Heavily augmented sound event detection utilizing weak predictions," DCASE2021 Challenge, Tech. Rep., 2021.

[20] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," in *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[21] Ç. Bilen, G. Ferroni, F. Tuveri, J. Azcarreta, and S. Krstulović, "A framework for the robust evaluation of sound event detection," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 61–65.

[22] A. Mesaros, T. Heittola, and T. Virtanen, "Metrics for polyphonic sound event detection," *Applied Sciences*, vol. 6, no. 6, 2016.

[23] S.-H. Kim, H. Nam, and Y.-H. Park, "Temporal dynamic convolutional neural network for text-independent speaker verification and phonemetic analysis," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022.

UNSUPERVISED DOMAIN ADAPTATION FOR THE CROSS-DATASET DETECTION OF HUMPBACK WHALE CALLS

Andrea Napoli[*], *Paul R. White*

Institute of Sound and Vibration Research
University of Southampton, UK
{an1g18, P.R.White}@soton.ac.uk

ABSTRACT

Machine learning methods, and deep networks in particular, often underperform on data which lies outside the training distribution. Changes to the data distributions (known as domain shift) are particularly prevalent in bioacoustics, where many external factors can vary between datasets, although the effects of this are often not properly considered. This paper presents a benchmark for out of distribution (OOD) performance based on the detection of humpback whales in underwater acoustic data. Several humpback whale detectors from the literature are implemented as baselines, along with our own detector based on a convolutional neural network (CNN). Then, a set of unsupervised domain adaptation (UDA) algorithms are compared. Results show that UDA can significantly improve OOD performance when few distinct sources of training data are available. However, this is not a substitute for better data, as negative transfer (where the adapted models actually perform worse) is commonly observed. On the other hand, we find that training on a variety of distinct sources of data (at least 6) is sufficient to allow models to generalise OOD, without the need for advanced UDA algorithms. This allows our model to outperform all the baseline detectors we test, despite having 10,000 times fewer parameters and 100,000 times less training data than the next-best model.

Index Terms— Unsupervised domain adaptation, domain shift, passive acoustic monitoring, humpback whale detection

1. INTRODUCTION

Passive acoustic monitoring (PAM) forms a major part of marine mammal conservation. Acoustic surveys are an effective and non-invasive means to further our understanding of species-wise geographic distributions, migration patterns and feeding grounds, monitor ecosystem health, and help to mitigate the impacts of human activity. Automated analysis of survey data can improve our ability to achieve these goals, whilst substantially reducing the manual effort required [1].

An ideal solution to this end would be an off-the-shelf tool which can be easily deployed on any new data and identify all the vocalising species present (and indeed, any other relevant acoustic event). We argue a major obstacle exists to achieving this sort of generalisation ability that particularly affects PAM, but is seldom properly considered. This is the fact that dataset biases [2] in PAM are unusually large compared to other areas of machine learning research (consider, for example, ImageNet [3]: sourced by trawling images from the Internet, this may be a more representative sample of the "set of all possible images" for which it is a surrogate). This increases the likelihood of a mismatch between the data distributions of a model's training set and the data it then encounters when deployed (known as domain shift), violating the i.i.d. assumption and potentially causing significant reductions in performance on new data.

We support the view that shortcut solutions, in which the training distribution contains spurious correlations between classes which do not transfer to new data, are the primary cause of shift-induced performance drops in real-world problems [4]. If these patterns have lower descriptive complexity than the intended solution, models will preferentially use them to "cheat" on a task. This is a significant complication, as the learning bias for simpler solutions is a huge part (but not all) of what makes generalisation possible in the first place (in particular, it helps prevent overfitting). Although a form of data leakage, the introduction of shortcut solutions is oftentimes simply unavoidable when constructing datasets, so we believe these are better thought of as an integral part of the learning problem, rather than mere developer oversight.

Thus, our first aim is to design experiments that create more realistic testing scenarios for PAM algorithms. We can do this by ensuring the training and test sets never contain any domain overlap, to better mimic the distributional shifts which may occur "in the wild" (we call this the *OOD testing* setup).

What exactly constitutes a "domain" in this context we keep deliberately abstract; the primary aim is to confine any covariate which may cause shortcuts to a single domain. For example, in one data source we use [5], separate tapes are often digitised into a single master recording, so these are considered a single domain even if the original tapes were collected in different locations or years. As we are only testing on OOD samples, the fact that some domains have examples collected in different conditions, resulting in shortcuts within a single domain, is inconsequential (we also argue this happens unavoidably anyway).

Our second aim is to identify best practices for maximising OOD performance in these scenarios. Unsupervised domain adaptation (UDA) has previously been used to tackle domain shift across many areas of wildlife monitoring [6], including PAM [7], [8]. For marine mammal PAM, domain shifts have been shown to result in reduced performance [9], and basic supervised finetuning has been used to adapt models to new environments [10]. However, to our knowledge, UDA is unexplored in this context. Thus, in this paper, a range of UDA algorithms from the literature are applied to a test problem of humpback whale detection.

The UDA literature is dominated by the *distribution alignment* approach, which aims to minimise the distance between the

---

[*] Thanks to BAE Systems and the Engineering and Physical Sciences Research Council for funding.

feature distributions of the source and target domains. The crux of this approach is finding how to estimate this distance reliably using only samples from the distributions. Two main approaches exist: kernel methods, which embed the distributions in a reproducing kernel Hilbert space (RKHS) [11]–[15]; and adversarial training, pioneered by [16] and the current basis for practically all state-of-the-art methods. Various extensions to the original "domain adversarial neural network" (DANN) formulation have followed the better-known literature on generative adversarial networks, such as with the introduction of the cycle-consistency loss [17], conditional adversarial training [18] and the Wasserstein objective [19].

As a final note, we call attention to subsequent analyses of existing UDA (and, more broadly, domain generalisation) algorithms in new contexts, on additional, perhaps more realistic, datasets, or averaged across many tasks, which have failed to reproduce or report much-reduced benefits compared to their original publications [6], [8], [20], [21]. Thus, we also consider that testing existing algorithms on new data helps contribute to the bigger picture of how effective or useful these methods actually are.

In summary, in this paper, we compare 8 UDA algorithms on a novel benchmark of OOD humpback whale detection. We also analyse the effect of varying the number of domains used to train the base model.

2. DATA

Humpback whale (*Megaptera novaeangliae*) calls are perhaps the most studied of all marine mammal vocalisations, and also what non-biologists usually mean when they talk about "whale song". The complex nature of the song, its population-level variability, and the fact that humpbacks are found in a wide range of environments all over the world make for an attractive (i.e., challenging) OOD problem. Additionally, the large body of previous work means many acoustic datasets already exist online and there are several well-established baselines to compare our approach to.

We construct a dataset consisting of approximately 100 minutes of audio, labelled as either humpback whale (HW) or non-humpback whale (NHW), from 13 distinct sources. Most of these sources already contain both HW and NHW examples, although some have only a single class; these are paired together so that every domain has examples from both classes, for a total of 9 domains.

Most data was downloaded from freely available sources online: the Watkins Marine Mammal Sound Database (which includes locations in the Caribbean, North Atlantic and Antarctica) [5], the Pacific Islands Passive Acoustic Network [22], the Australian National Mooring Network [23], the Hawaiian Islands Cetacean and Ecosystem Assessment Survey [24] and moby-sound.org; the remaining data was recorded in Madagascar in an in-house collection project [25].

Samples were handpicked to create a diverse, representative, and challenging learning problem, covering a wide range of non-target underwater acoustic events, geographic locations, recording methods and environments. All audio was resampled to 8 kHz, although two domains have original sample rates of 4 and 6 kHz, so do not contain higher-frequency information – we just consider this an additional characteristic of the learning problem to be overcome. Some exemplar spectrograms are shown in Figure 1.

We use the same audio pre-processing pipeline as Allen et al. [26]: mel spectrograms are generated using 100 ms FFT windows with 50% overlap, normalised with per-channel energy normalisation [27], then split into 3.92 s analysis frames with 50% overlap. This results in 3,150 total spectrogram images, measuring 64 by 128 pixels. The number of images is broken down by class and domain in Figure 2.

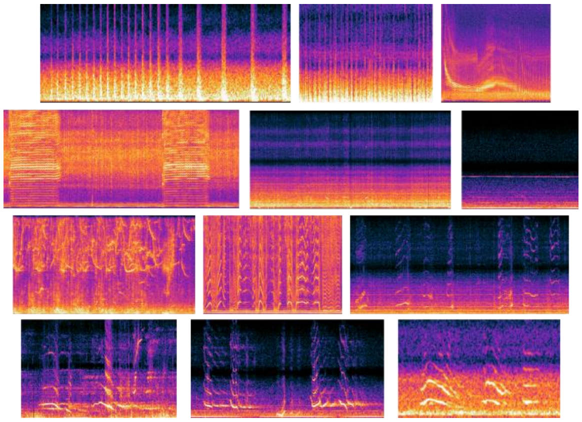


Figure 1: Some exemplar spectrograms of sounds in the dataset (4 kHz bandwidth, time axis scales variable). Top row: sperm whale clicks, pilot whale clicks, seal vocalisations. Second row: minke whale boings, right whale calls in strong vessel noise, electrical interference. Third row: dolphin whistles, dolphin creaks, right whale calls. Bottom row: three humpback whale calls.

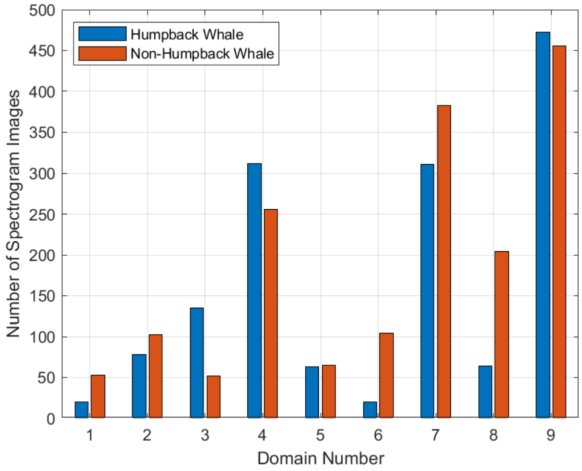


Figure 2: Total number of spectrogram images in the dataset, by class and domain.

Extracting a single value from the literature for what constitutes "acceptable" performance for this task is difficult. Helble et al. [28] state that any automated detector should perform at or above the level of a trained human analyst, although even this benchmark varies greatly based on the call's SNR, the nature of the background noise, as well as the human in question. However, based on values in [28], and without wishing to get too lost in the details, we consider a balanced accuracy of 87% to be the bare minimum required for this task, and anything above 90% to be good.

3. DETECTORS

A simple CNN is designed with 4 convolutional layers and one dense layer. The convolutional layers each have 3 by 3 kernels, (2, 2) stride, 16 filters and RELU activations, with 7,154 trainable parameters total. Batch normalisation was found in testing to deteriorate OOD performance, reproducing findings in the literature [20], so is not used. Training is performed using the Adam optimiser with an initial learning rate of 0.001 and a batch size of 32, for 500 iterations.

In addition to empirical risk minimization (ERM) (that is, the standard training paradigm with no adaptation), 8 UDA algorithms are compared:

- Principal component analysis (PCA)
- Correlation alignment (CORAL) [29]
- Geodesic flow kernel (GFK) [30]
- Transfer component analysis (TCA) [11]
- Joint distribution alignment (JDA) [12]
- Transfer joint matching (TJM) [13]
- Manifold embedded distribution alignment (MEDA) [14]
- Scatter component analysis (SCA) [15]

The CNN is first trained normally on the source domain data. The UDA algorithms are then applied to the activations of the final convolutional layer. Finally, a new dense layer is trained on the transformed source domain features. For the methods based on dimensionality reduction (all but CORAL), the output dimension is set to 8. The whole process is repeated 5 times to reduce the influence of parameter initialisation and provide a measure of the uncertainty for the results.

In addition to the shallow UDA algorithms listed above, various types of deep adversarial UDA [16], [18], [19] were also attempted, but failed to work, and are not included in these results. Other than the notorious difficulties that come with adversarial training (e.g., mode collapse), we also suspect that these methods require larger amounts of data than is available in our application, which may explain why they failed in this case.

3.1. Baselines

We also implement 3 baseline detectors for this task:

1) *Allen et al.* [26], a ResNet-50 [31] architecture (25.6 M parameters) trained on 187,000 hours of data from a single PAM program [22]. The decision threshold is set to the average of all the optimal thresholds stated in the paper (a different threshold is used per site), at 0.13. One domain of our dataset contains data overlap with the training set for this model, so we do not include it when calculating the average test accuracy for this baseline.

2) *YAMNet* [32], a MobileNet-V1 [33] architecture (3.7 M parameters) trained on AudioSet [34], a broad ontology of 527 classes of audio events drawn from YouTube. In particular, we are counting detections of the class “Whale vocalisation”. The training data for this class consisted of around 20 minutes of audio from 129 videos, most of which upon inspection are humpback whales.

3) *Template matching*, via cross-correlation of spectrograms [35]. For each test sample, a 2D correlation is performed with each humpback call training sample and the highest correlation value is taken as the recognition score. The nontarget training samples are unused. The decision threshold is chosen based on tests on a held-out subset of training data, and is set to 0.2.

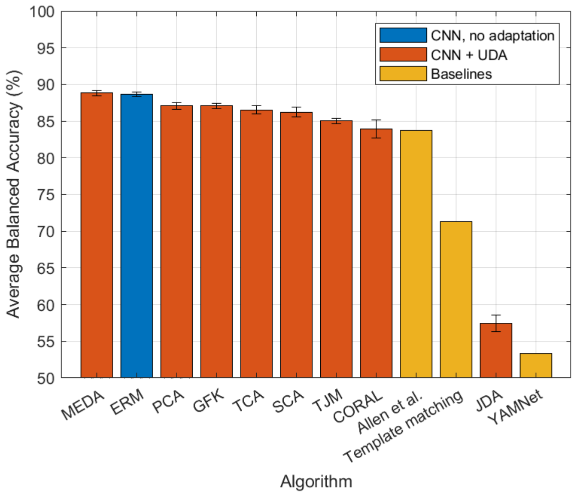


Figure 3: Average balanced accuracy across all domains for each algorithm. Error bars denote standard error in the mean.

4. RESULTS

First, the algorithms and baselines are compared using leave-one-domain-out cross-validation – that is, the model is trained using data from all but one domain at a time. The performance measure used is balanced accuracy, equal to the mean of the true positive rate and true negative rate, and averaged across all domains. The results are shown in Figure 3, where the error bars denote standard error in the mean across the repeats (note, the baselines do not have error bars).

Our tests show that no UDA algorithm exceeds ERM by a significant margin – at most 0.2 percentage points for MEDA. This reproduces recent findings on OOD generalisation from the literature [6], [8], [20], [21] – where the ERM baseline has been described as “frustratingly strong” [20]. It is clear that, in this case, the diversity of the training data makes a far larger difference than the learning algorithm, with our best models significantly outperforming the Allen et al. [26] baseline, despite having 10,000 times fewer parameters, 100,000 times less training data and no pretrained backbone. A total of 4 algorithms, including ERM, exceed the 87% accuracy criterion. Template matching also performs surprisingly well, although this is rather dependent on the domain being tested.

What is perhaps most striking is how often UDA actually reduces performance when it is applied – a phenomenon known as *negative transfer* [36]. Some algorithms completely destroy the model’s predictive power (e.g., JDA) and *every* algorithm underperforms ERM in at least one domain. This behaviour has been observed consistently throughout our work on UDA – not least for the adversarial methods. We suspect that a bias exists in commonly used UDA benchmarks which may, in particular, explain why our application of UDA fails to reproduce the massive improvements on ERM often seen elsewhere. This is that the distribution alignment is often performed on features from a pretrained backbone (usually ResNet-50) which has already “seen” target domain samples. The biased feature distributions then make the alignment task far easier than if no such pretraining is available. Otherwise, the phenomenon of *modal misalignment* (also called false alignment [36], essentially analogous to overfitting) is far

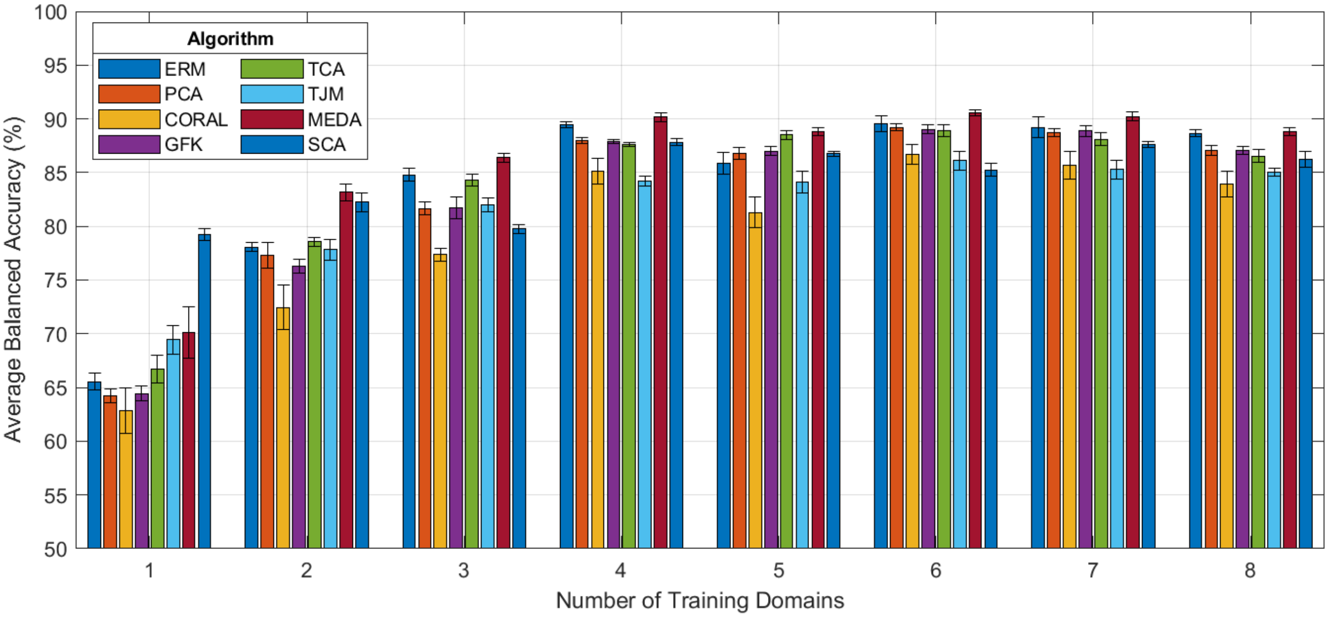


Figure 4: Balanced accuracy by number of domains in the training set, for each algorithm.

more prevalent. Extending the current UDA setup (where all training domains are pooled together and considered a single “source” domain) to multi-source UDA has previously been proposed to alleviate this difficulty [36], and will be investigated in future work.

4.1. How many training domains are needed?

The previous section showed that, given an abundance of training domains, no UDA algorithm significantly outperforms ERM. However, the question arises: when training domains are limited (as can easily happen in PAM, particularly for rare species), can UDA compensate for the lack of diversity in the training data?

In this section, the number of training domains is varied from 1 to 8. The domains that are not used for training are used for validation. This is done across at least 3 cross-validation folds, subject to the training set being large enough (we use a cut-off of at least 500 instances). The average balanced accuracy across all validation folds and domains, along with standard errors, is shown in Figure 4.

It can be seen that UDA is increasingly beneficial as fewer training domains become available. With a single training domain, SCA provides 14 percentage points improvement over ERM, although it is not a complete substitute for better data. Having at least 6 training domains appears to be a necessary and sufficient condition for achieving maximal performance on this dataset: it is the point where the performance of most algorithms no longer increases, as well as the point where UDA no longer significantly improves on ERM.

The fact that OOD accuracy does not clearly increase monotonically with the number of training domains (for example, there is a definite drop for most algorithms in going from 4 to 5 domains) suggests that, as found in [37], the design of the dataset, including the relative abundance of each domain, is an important factor, and naively combining as much data as possible may not be the best strategy. This will be investigated further in future work.

5. CONCLUSION

This paper presented a novel benchmark for OOD generalisation, namely the cross-dataset detection of humpback whales in PAM data. A total of 8 UDA algorithms, applied to a simple CNN detector, were tested on this benchmark, as well as 3 existing baseline detectors. It was shown that large domain shifts exist between data from different PAM projects, resulting in significant underperformance OOD if training data from only one domain is used. However, training on a variety of distinct sources of data (at least 6) is sufficient to allow models to generalise OOD, without the need for advanced algorithms. In cases where limited training domains are available, UDA can be used to recover a large part of the shift-induced performance drop.

Although some algorithms may exceed ERM on average, no algorithm consistently outperforms ERM every time, highlighting the challenges still faced in achieving reliable, trustworthy OOD generalisation. Being able to predict which algorithms will work in a particular domain would be a significant step towards achieving this goal – for example, the best model could then be dispatched automatically using a specialty-aware ensemble [38]. As of yet, no pattern appears to exist, although this will be investigated further in future work.

6. REFERENCES

[1] P. Nguyen, “Development of artificial intelligence methods for marine mammal detection and classification of underwater sounds in a weak supervision (but) Big Data-Expert context,” Doctoral Thesis, Sorbonne University, 2020.

[2] A. Torralba and A. A. Efros, “Unbiased look at dataset bias,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2011.

[3] J. Deng, W. Dong, R. Socher, L.-J. Li, Kai Li, and Li Fei-Fei, “ImageNet: A large-scale hierarchical image database,” in *IEEE Conference on Computer Vision and Pattern Recognition*, Mar. 2010.

[4] M. Arjovsky, L. Bottou, I. Gulrajani, and D. Lopez-Paz, “Invariant Risk Minimization,” *arXiv*, 2019.

[5] L. Sayigh *et al.*, “The Watkins Marine Mammal Sound Database: An online, freely accessible resource,” in *Proceedings of Meetings on Acoustics*, 2016.

[6] S. Sagawa *et al.*, “Extending the WILDS Benchmark for Unsupervised Adaptation,” *ICLR*, 2021.

[7] V. Lostanlen, J. Salamon, A. Farnsworth, S. Kelling, and J. P. Bello, “Robust sound event detection in bioacoustic sensor networks,” *PLoS One*, 2019.

[8] M. Boudiaf, T. Denton, B. van Merriënboer, V. Dumoulin, and E. Triantafillou, “In Search for a Generalizable Method for Source Free Domain Adaptation,” *ICML*, 2023.

[9] O. S. Kirsebom, F. Frazao, Y. Simard, N. Roy, S. Matwin, and S. Giard, “Performance of a deep neural network at detecting North Atlantic right whale upcalls,” *Journal of the Acoustical Society of America*, 2020.

[10] B. Padovese *et al.*, “Adapting deep learning models to new acoustic environments - A case study on the North Atlantic right whale upcall,” *Ecol Inform*, 2023.

[11] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, “Domain adaptation via transfer component analysis,” *IEEE Trans Neural Netw*, 2011.

[12] M. Long, J. Wang, G. Ding, J. Sun, and P. S. Yu, “Transfer feature learning with joint distribution adaptation,” *ICCV*, 2013.

[13] M. Long, J. Wang, G. Ding, J. Sun, and P. S. Yu, “Transfer joint matching for unsupervised domain adaptation,” *ICCV*, 2014.

[14] J. Wang, W. Feng, Y. Chen, M. Huang, H. Yu, and P. S. Yu, “Visual Domain Adaptation with Manifold Embedded Distribution Alignment,” *MM 2018 - Proceedings of the 2018 ACM Multimedia Conference*, pp. 402–410, Jul. 2018.

[15] M. Ghifary, D. Balduzzi, W. B. Kleijn, and M. Zhang, “Scatter Component Analysis: A Unified Framework for Domain Adaptation and Domain Generalization,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 7, pp. 1414–1430, Oct. 2015.

[16] Y. Ganin *et al.*, “Domain-Adversarial Training of Neural Networks,” *JMLR*, 2015.

[17] J. Hoffman *et al.*, “CyCADA: Cycle-Consistent Adversarial Domain Adaptation”, *ICML*. 2017.

[18] M. Long, Z. Cao, J. Wang, and M. I. Jordan, “Conditional Adversarial Domain Adaptation,” *Advances in Neural Information Processing Systems*, May 2017.

[19] J. Shen, Y. Qu, W. Zhang, and Y. Yu, “Wasserstein Distance Guided Representation Learning for Domain Adaptation,” *32nd AAAI Conference on Artificial Intelligence, AAAI 2018*, pp. 4058–4065, Jul. 2017, doi: 10.1609/aaai.v32i1.11784.

[20] I. Gulrajani and D. Lopez-Paz, “In Search of Lost Domain Generalization,” *ICLR*, 2021.

[21] A. Dubey, V. Ramanathan, A. Pentland, and D. Mahajan, “Adaptive Methods for Real-World Domain Generalization,” *CVPR*, 2021, doi: 10.1109/CVPR46437.2021.01411.

[22] NOAA Pacific Islands Fisheries Science Center, “Pacific Islands Passive Acoustic Network (PIPAN) 10kHz Data.” NOAA National Centers for Environmental Information, 2021.

[23] Integrated Marine Observing System, “Australian National Mooring Network,” 2017. https://imos.org.au/facilities/nationalmooringnetwork (accessed Apr. 09, 2023).

[24] NOAA Pacific Islands Fisheries Science Center, “Hawaiian Islands Cetacean and Ecosystem Assessment Survey (HICEAS) towed array data. Edited and annotated for DCLDE 2022,” *NOAA National Centers for Environmental Information*. 2022.

[25] F. Pace, P. White, and O. Adam, “Hidden Markov modeling for humpback whale (Megaptera Novaeanglie) call classification,” *Proceedings of Meetings on Acoustics*, 2012.

[26] A. N. Allen *et al.*, “A Convolutional Neural Network for Automated Detection of Humpback Whale Song in a Diverse, Long-Term Passive Acoustic Dataset,” *Front Mar Sci*, 2021.

[27] Y. Wang, P. Getreuer, T. Hughes, R. F. Lyon, and R. A. Saurous, “Trainable Frontend For Robust and Far-Field Keyword Spotting,” *ICASSP*, 2016.

[28] T. A. Helble, G. R. Ierley, G. L. D’Spain, M. A. Roch, and J. A. Hildebrand, “A generalized power-law detection algorithm for humpback whale vocalizations,” *The Journal of the Acoustical Society of America*, Apr. 2012.

[29] B. Sun, J. Feng, and K. Saenko, “Return of frustratingly easy domain adaptation,” *30th AAAI Conference on Artificial Intelligence, AAAI 2016*, pp. 2058–2065, 2016.

[30] B. Gong, Y. Shi, F. Sha, and K. Grauman, “Geodesic flow kernel for unsupervised domain adaptation,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 2066–2073, 2012.

[31] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” *CVPR*, 2015.

[32] “YAMNet.” https://github.com/tensorflow/models/tree/master/research/audioset/yamnet (accessed Jun. 23, 2022).

[33] A. G. Howard *et al.*, “MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications,” *CoRR*, Apr. 2017, doi: 10.48550/arxiv.1704.04861.

[34] J. F. Gemmeke *et al.*, “Audio Set: An ontology and human-labeled dataset for audio events,” *ICASSP*, 2017.

[35] E. T. Vu *et al.*, “Humpback whale song occurs extensively on feeding grounds in the western North Atlantic Ocean,” *Aquatic Biology*, 2012.

[36] Z. Pei, Z. Cao, M. Long, and J. Wang, “Multi-Adversarial Domain Adaptation,” *AAAI*, Sep. 2018.

[37] T. Nguyen, G. Ilharco, M. Wortsman, S. Oh, and L. Schmidt, “Quality Not Quantity: On the Interaction between Dataset Design and Robustness of CLIP,” *NeurIPS*, Aug. 2022.

[38] Z. Li, K. Ren, X. Jiang, B. Li, H. Zhang, and D. Li, “Domain Generalization using Pretrained Models without Fine-tuning,” *CoRR*, Mar. 2022.

FEW-SHOT BIOACOUSTIC EVENT DETECTION AT THE DCASE 2023 CHALLENGE

I. Nolasco[1], B. Ghani[2], S. Singh[1], E. Vidaña-Vila[3], H. Whitehead[10], E. Grout[4,5], M.G. Emmerson[7], I. Kiskin[9], F. H. Jensen[8], J. Morford[6], A. Strandburg-Peshkin[4,5], L. Gill[11], H. Pamuła[12], V. Lostanlen[13], D. Stowell[2,14]

[1] Centre for Digital Music (C4DM), Queen Mary University of London, London, UK
[2] Naturalis Biodiversity Centre, Leiden, NL
[3] La Salle Campus Barcelona, Ramon Llull University, Barcelona, ES
[4] Dept. of Biology & Centre for the Advanced Study of Collective Behaviour, University of Konstanz, DE
[5] Dept. for the Ecology of Animal Societies, Max Planck Institute of Animal Behavior, DE
[6] The Oxford Navigation group, Dept. of Zoology, Oxford University, Oxford, UK
[7] School of Biological and Behavioural Sciences, Queen Mary University of London, London, UK
[8] Biology Dept, Syracuse University, NY, USA
[9] Institute for People-Centred AI, FHMS, University of Surrey, Surrey, UK
[10] School of Science, Engineering and Environment, University of Salford, Manchester, UK
[11] Landesbund für Vogel- und Naturschutz; Naturkundemuseum Bayern/BIOTOPIA Lab, DE
[12] AGH University of Science and Technology, Kraków, PL
[13] Nantes Université, École Centrale Nantes, CNRS, LS2N, UMR 6004, F-44000 Nantes, FR
[14] Tilburg University, Tilburg, NL

ABSTRACT

Few-shot bioacoustic event detection consists in detecting sound events of specified types, in varying soundscapes, while having access to only a few examples of the class of interest. This task ran as part of the DCASE challenge for the third time this year with an evaluation set expanded to include new animal species, and a new rule: ensemble models were no longer allowed. The 2023 few-shot task received submissions from 6 different teams with F-scores reaching as high as 63% on the evaluation set. Here we describe the task, focusing on describing the elements that differed from previous years. We also take a look back at past editions to describe how the task has evolved. Not only have the F-score results steadily improved (40% to 60% to 63%), but the type of systems proposed have also become more complex. Sound event detection systems are no longer simple variations of the baselines provided: multiple few-shot learning methodologies are still strong contenders for the task.

Index Terms— Few-shot learning, bioacoustics, sound event detection

1. INTRODUCTION

Bioacoustic event detection, the identification of animal vocalizations within specific timeframes, shares many similarities with sound event detection (SED) in varying contexts like urban settings [1] or secured spaces [2, 3]. Nonetheless, bioacoustics poses a unique set of challenges due to the varied recording conditions and diverse animal vocalizations [4]. This makes it an exciting and complex domain within machine learning, with several specialized sub-disciplines focused on different animals. Recent advances in supervised deep convolutional neural networks (CNN) have potential for enhancing feature detection.

However, their supervised nature necessitates extensive, well-categorized acoustic event data and hundreds of annotated examples per class. Gathering this data can be an uphill battle, considering the uneven distribution of species, the labor-intensive nature of audio annotation, and the variable taxonomy based on the use case [5]. The limitations of a supervised sound event detection system become more prominent when extrapolating techniques used in speech to other animal sounds. This complexity arises from the differences in sound duration, interest units, and the context in which the sounds are made. Crucially, understanding the commencement and termination times of animal sounds is vital to community ecology, shedding light on various patterns of communication and influence among species [6]. Unlike speech science with its relatively limited granularity, bioacoustic studies operate at multiple levels, from coarse classification of species to fine distinction of individual call types. Moreover, the diversity in recording equipment used for animal sounds, from far-field to underwater, adds another layer of complexity, transforming bioacoustic event detection into a collection of small-data problems, each requiring specialized systems. This fragmentation, although useful for species classification tasks, impedes the practical application of deep learning in bioacoustics and life sciences more broadly [5].

To address these challenges, this DCASE task proposes a unified approach for bioacoustic event detection across the various subdomains, aiming to mitigate the problems associated with data acquisition, annotation, and the fragmentation in computational bioacoustics. Hence, we compiled a unique ensemble of 14 small-scale datasets, each between 10 minutes and 10 hours long and derived from distinct sources, representing different application contexts. Breaking from the norm of training individual machine learning systems for each dataset, the idea is to develop a single, versatile system capable of identifying sound events across various datasets, with event categories specified at ”query time”. Additionally, dur-

	Dataset	mic type	# audio files	total duration	# labels	# events
Training set	BV: BirdVox-DCASE-10h	fixed	5	10 hours	11	9026
	HT: Hyenas	various	5	5 hours	5	611
	MT: Meerkats	animal mounted	2	70 mins	4	1294
	JD: Jackdaws	mobile	1	10 mins	1	357
	WMW: Western Mediterranean Wetlands Birds	various	161	5 hours	26	2941
Validation set	HB: Humbug mosquitoes	handheld	10	2.38 hours	1	712
	PB: Polish Baltic Sea bird flight calls	fixed	6	3 hours	2	292
	ME: Meerkats	animal mounted	2	20 mins	2	73
Evaluation Set	CHE: Transfer-Exposure-Effects birds	fixed	18	3 hours	3	2550
	DC: BIOTOPIA Dawn Chorus birds	fixed	10	95 mins	3	967
	CT: Coati	handheld	3	48 mins	3	365
	MS: Manx shearwater birds	fixed	4	40 mins	1	1087
	QU: Dolphin quacks	animal mounted	8	74 mins	1	3441
	MGE: Chick calls birds	fixed	3	32 mins	2	1195
	CHE23: Transfer-Exposure-Effects Frogs	fixed	16	40 mins	1	798
	CW: Cow moos	fixed	4	56 mins	1	293

Table 1: Summary of dataset characteristics.

ing an evaluation on an audio file, the system is provided with the initial five instances of the desired sound event. This approach employs a machine learning paradigm known as ”few-shot learning” (FSL) [7, 8], where the aim is to construct precise models using less training data. In this context, FSL is explored using N-way-k-shot classification, where N and k represent the number of classes and the examples per class, respectively. Upon training with the first five occurrences of an event, the system effectively detects subsequent instances of the same event.

Our hypothesis is that bioacoustic event detectors can be trained using available bioacoustic datasets and then generalized to new targets using a few examples at the time of deployment.

2. DATASETS

When the DCASE challenge begins, each task releases its own development set, consisting of a training and validation sets. Participants must use this dataset to develop and validate their systems. As the challenge enters the evaluation phase, the evaluation set is released and participants apply their developed systems and output the predictions which are then used to calculate the final ranking scores. These datasets are organised in subsets that represent different acoustic sources and were gathered here with the specific purpose of broadening the targeted species. A summary of the main characteristics is presented in Table 1. Overall there are 8 sets focusing on bird species, 5 sets of mammal vocalisations (one of which underwater), 1 set of flying insect sounds (HB) and 1 set of amphibian calls (CHE23).

For the Few shot bioacoustic task, the training set is multi-label, since the provided annotations contain more than one class of interest. However, both validation and evaluation sets are single label, meaning that each audio file is annotated only for a single class of interest. While events of other classes are present these are not annotated and should not be predicted by the systems.

Also, the split between training, validation and evaluation sets does not follow the common supervised learning approach, since the classes in the validation and evaluation sets may overlap or not with the classes in the training set. This is designed so that examples of the target classes we want to detect are only provided as the 5 shots.

Given the few-shot setup of this task, each audiofile of the evaluation set is accompanied only with the annotations for the 5 initial events of the class of interest. The datasets used on the 2023 edition of the task remain the same as in previous edition (thoroughly described in [5]), but the evaluation set has been extended with two new subsets of data: Cow moos (CW) and frog croakings (CHE23).

Cow moos (CW): This dataset contains 4 audio files of about 15 minutes each recorded on a Cow's farm in Catalonia, Spain. An ambient microphone connected to a Zoom H5 recorder was hung on the ceiling of a yard with multiple cows. Cow vocalizations were recorded and manually labelled by researchers from La Salle Campus Barcelona and AWEC Advisors S.L. in the framework of the projects CowTalk and CowTalk-Pro.

Transfer-Exposure-Effects Frogs (CHE23): This dataset is part of the same project which originated the CHE dataset, data were collected using unattended acoustic recorders (Songmeter 3) in the Chornobyl Exclusion Zone (CEZ) to capture the Chornobyl soundscape and investigate the longterm effects of the nuclear plower plant accident on the local ecology. The CHE23 dataset consists in 16 audiofiles of varying lengths annotated for frog croaking events, however many other calls of other species are present throughout the recordings. The annotations were produced by Helen Whitehead using Raven Pro 1.6.

Finally, teams are allowed to use certain external datasets and pretrained models. These were selected based on two criteria: They must be open access and classes do not overlap with the evaluation target classes. The complete list of allowed datasets and pretrained models can be accessed in the DCASE challenge task page [11].

3. BASELINES AND EVALUATION METRICS

The benchmarks and evaluation metrics remain identical to those established in the 2022 rendition of the task [9]. The associated code can be procured from the GitHub repository[1].

The few-shot bioacoustic sound event detection task adopts two baselines: 1) Template matching, and 2) Protoyipical networks. Template matching represents a common practice in the bioacoustics domain. The overall approach consists in taking each of the

---

[1] `https://github.com/c4dm/dcase-few-shot-bioacoustic`

Team name	**Best submission**	**Eval set: F-score %**	**Val set F-score %**	**Main characteristics**
Du_NERCSLIP	2	63.78	75.6	Multitask learning; Frame-level embeddings; Voice activity detection
Moummad_IMT	2	42.72	63.46	Contrastive learning learns an Embedding space; fine-tuning encoder on both positive and negative events;
XuQianHu_NUDT_BIT	3	42.5	63.94	prototypical network, Delta MFCC and PCEN; Squeeze Excitation blocks
Gelderblom_SINTEF	2	31.10	36.6	Encoder based on BEATs; prototypical network.
Jung_KT	3	27.12	81.52	Prototypical network trained with a negative-based loss
Wilkinghoff_FKIE	4	16.00	62.64	Embeddings learnt with temporal dimension; template matching with Dynamic warping.
TM_baseline	-	14.89	3.37	Template matching baseline as in [9]
proto_baseline	-	2.92	28.45	Prototypical network baseline as in [9]

Table 2: F-score results per team (best scoring system) on evaluation and validation sets, and summary of system characteristics. Systems are ordered by higher scoring rank on the evaluation set. These results and technical reports for the submitted systems can be found on task 5 results' page [10].

5 initial examples as templates and cross-correlate each template with the remaining audiofile. Events are predicted by selecting time frames where the cross-correlation values surpasses a defined threshold. Prototypical networks [7], on the other hand are trained through episodic learning and employ a 5-way-2-shot classification model in our case. Each prototype represents a coordinate in vector space, calculated as a mean of the coordinates of the 5 samples. Training comprises a *Support set* of 5 labelled samples from each class, while the remaining samples form the *Query set*. A class prototype is computed via an embedding function with learning parameters. Distances are optimised, and the network training creates a general representation where similar sounds are closer. In this way, the future data points are labelled using nearest-neighbour algorithms.

The systems are evaluated based on how well they predict events on the evaluation set. The metric used combines intersection over union and bipartite graph matching algorithms to select the best matches between the predicted events and ground truth events. After the matching phase, we count the number of true positives (TP) , false positives (FP), and false negatives (FN), which are then used to calculate precision, recall and F-score. The systems are ranked using the event based F-score value. The task description and details are provided in [5].

In this year's task, a distinctive modification has been introduced in the evaluation procedure. The use of ensemble models was no longer allowed. The objective behind this rule is to incentivise the development of truly general models, rather than a simple fusion of completely independent models.

4. RESULTS

The third edition of the few-shot bioacoustic event detection task received participation of 6 teams, with a total of 22 submitted systems. The overall F-scores for the best submission per team are presented in Table 2 together with the main characteristics of the respective systems, and the results on each dataset of the evaluation set are presented in Fig. 1.

The winning submission is by the same team that won the previous edition of this task, namely, **Du_NERCSLIP**. The system improves on the last year's submission, [12] by including their frame-level embedding system into a multi-task learning architecture. The new system now includes Target Speaker Voice Activity Detection as one of the branches. This system achieved 63% F-score which is an increase from the best F-score from last year, that was aprox. 60%. Observing Fig. 2, it is possible to observe the improved results on individual datasets for this team compared to the last year's system (columns Du22 and Du23). This shows that the described modifications are responsible for the considerable increase in the overall F-score.

Furthermore, an intriguing observation when looking at the F-scores per dataset in Fig. 1, is that overall systems performed extremely well on the CW dataset, but not the winning submission. Indeed the performance of Du_NERCSLIP's system on CW dataset is similar to the performance on the QU dataset, which is considered to be an extremely difficult case due to its very short events.

Moummad_IMT implemented a system based on Contrastive Learning, a method to learn an embedding space that maximises the distinction between positive events and negative events. During the evaluation stage, the encoder is further fine-tuned on the 5 positive

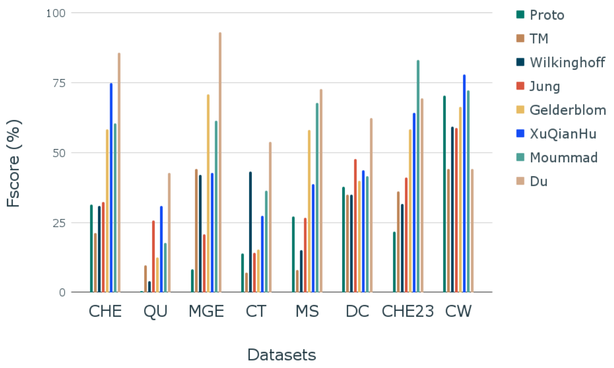


Figure 1: F-Score results of best system for each team and baselines. Systems are ordered from least to highest scoring rank on the evaluation set and indicated by the name of main author.

---

[2] `https://dcase.community/challenge2022/task-few-shot-bioacoustic-event-detection-results`

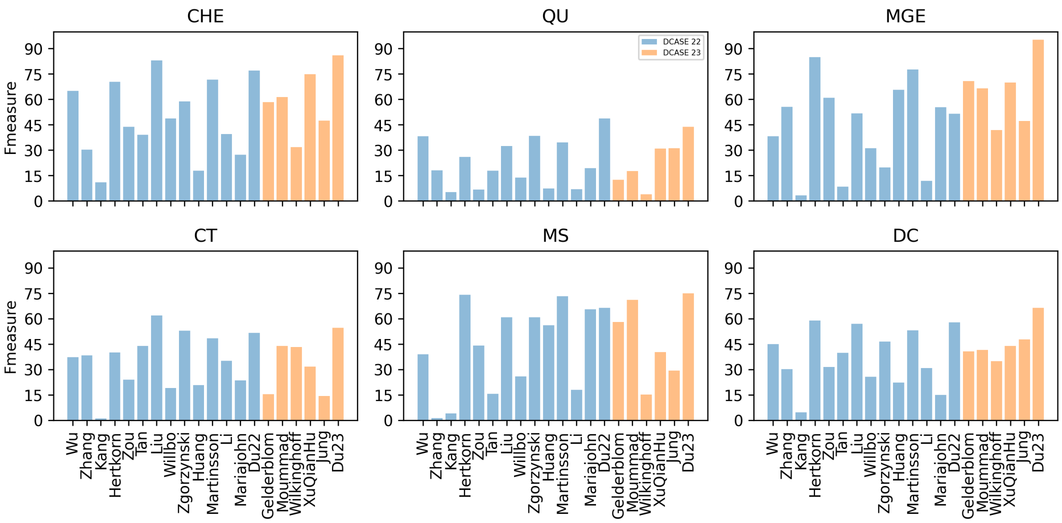


Figure 2: Comparison of the maximum F-score achieved by each team for the datasets used in both 2022[2] and 2023. The results for each year are distinguished by different colors.

examples provided and on selected negative examples for each audiofile. The final predictions are then the result of a simple binary classification in this embedding space.

XuQianHu_NUDT_ BIT largely improved upon the Prototypical Network baseline by including squeeze/excitation (SE) blocks into the encoder part of the network. The idea behind this is to create an adaptive mechanism that assigns different weights to different channels of the feature map, by predicting their importance. The system also adopts the negative sample search mechanism proposed by Liu *et al.* on the previous edition of the task [13], which is designed to improve the learning of the negative prototypes. Also following from Liu *et al* submission, the input features employed are Delta MFCC and PCEN.

Gelderblom_SINTEF followed the Prototypical Network approach, but use the BEATs pretrained model as encoder. BEAT stands for Bidirectional Encoder representation from Audio Transformers released by Microsoft for audio tokenisation and classification. In their submission, the authors explore how useful this model is to represent bioacoustic data and compare the embeddings produced by the pretrained model with the embeddings produced after a few epochs of fine-tuning on the ECS-50 dataset [14].

Jung_KT combines Contrastive Learning and Prototypical Networks. This addresses the problem that the high imbalance between positive samples and negative samples creates in the learning of the prototypes. They propose a novel negative-based prototypical loss function that is used in a fine-tuning stage of the pipeline and drives the system to maximise the positive to negative samples distance and minimise the distance between negative samples.

Wilkinghoff_FKIE adopts template matching and dynamic time warping applied to embeddings trained with temporal resolution. The embedding model is trained to predict both class and temporal position of the sound event.

Observing the results spanning the two-year period (see Fig. 2), it is evident that each dataset presents unique challenges for various algorithms. Notably, the QU dataset consistently proved to be difficult for all participating teams across both years. Certain aspects, which are not fully discussed here, have had an important impact in the performance of these systems. We highlight data augmentation and post processing techniques adapted to the task which have been fundamental to achieve a good score. A comprehensive analysis and discussion of these aspects can be read in [5].

5. CONCLUSION

The 2023 edition of the few-shot bioacoustic event detection task received some very innovative systems that reflect the state-of-the-art in few-shot learning. We especially underscore the introduction of a novel technique, such as contrastive learning, making its initial entry in the history of the task's execution. Contrastive learning in the audio domain has seen increasing success and seems like a promising approach for the few-shot problem. Also of note is the quality of the evaluation set gathered this year. The dataset now extends to 3 different taxonomic groups: mammals, birds and amphibians, which is a good indicator of the variety of challenges faced in the bioacoustics domain. Moving forward we would be interested in analysing how exactly the characteristics of the different datasets impact each system and be able to understand if a single general model is indeed capable of predicting many different classes based on such few examples. The work in [5] started to tackle these questions, and while it is still not clear, the improving results on successive editions of this task indicate that the few-shot setting is a way to go.

6. REFERENCES

[1] A. Mesaros, T. Heittola, A. Eronen, and T. Virtanen, "Acoustic event detection in real life recordings," in *2010 18th European signal processing conference*. IEEE, 2010, pp. 1267–1271.

[2] D. Stowell, D. Giannoulis, E. Benetos, M. Lagrange, and M. D. Plumbley, "Detection and classification of acoustic scenes and events," *IEEE Transactions on Multimedia*, vol. 17, no. 10, pp. 1733–1746, 2015.

[3] N. Turpault, R. Serizel, A. P. Shah, and J. Salamon, "Sound event detection in domestic environments with weakly labeled data and soundscape synthesis," in *Workshop on Detection and Classification of Acoustic Scenes and Events*, 2019.

[4] D. Stowell, "Computational bioacoustic scene analysis," *Computational analysis of sound scenes and events*, pp. 303–333, 2018.

[5] I. Nolasco, S. Singh, V. Morfi, V. Lostanlen, A. Strandburg-Peshkin, E. Vidaña-Vila, L. Gill, H. Pamuła, H. Whitehead, I. Kiskin, *et al.*, "Learning to detect an animal sound from five examples," *arXiv preprint arXiv:2305.13210*, 2023.

[6] D. Stowell, L. Gill, and D. Clayton, "Detailed temporal structure of communication networks in groups of songbirds," *Journal of the Royal Society Interface*, vol. 13, no. 119, p. 20160296, 2016.

[7] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," *Advances in neural information processing systems*, vol. 30, 2017.

[8] Y. Wang, N. J. Bryan, M. Cartwright, J. P. Bello, and J. Salamon, "Few-shot continual learning for audio classification," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 321–325.

[9] I. Nolasco, S. Singh, E. Vidaña-Vila, E. Grout, J. Morford, M. Emmerson, F. Jensen, I. Kiskin, H. Whitehead, A. Strandburg-Peshkin, *et al.*, "Few-shot bioacoustic event detection at the dcase 2022 challenge," in *Detection and Classification of Acoustic Scenes and Events*, 2022.

[10] https://dcase.community/challenge2023/task-few-shot-bioacoustic-event-detection-results, accessed: 10-06-2023.

[11] http://dcase.community/workshop2023/.

[12] J. Tang, Z. Xueyang, T. Gao, D. Liu, X. Fang, J. Pan, Q. Wang, J. Du, K. Xu, and Q. Pan, "Few-shot embedding learning and event filtering for bioacoustic event detection technical report," DCASE2022 Challenge, Tech. Rep., June 2022.

[13] H. Liu, X. Liu, X. Mei, Q. Kong, W. Wang, and M. D. Plumbley, "Surrey system for dcase 2022 task 5 : Few-shot bioacoustic event detection with segment-level metric learning technical report," DCASE2022 Challenge, Tech. Rep., June 2022.

[14] K. J. Piczak, "ESC: Dataset for Environmental Sound Classification," in *Proceedings of the 23rd Annual ACM Conference on Multimedia*. ACM Press, pp. 1015–1018. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2733373.2806390

ADVANCING NATURAL-LANGUAGE BASED AUDIO RETRIEVAL WITH PASST AND LARGE AUDIO-CAPTION DATA SETS

Paul Primus[1], *Khaled Koutini*[1,2], *Gerhard Widmer*[1,2]

[1]Institute of Computational Perception (CP-JKU)
[2]LIT Artificial Intelligence Lab
Johannes Kepler University, Austria

ABSTRACT

This work presents a text-to-audio-retrieval system based on pre-trained text and spectrogram transformers. Our method projects recordings and textual descriptions into a shared audio-caption space in which related examples from different modalities are close. Through a systematic analysis, we examine how each component of the system influences retrieval performance. As a result, we identify two key components that play a crucial role in driving performance: the self-attention-based audio encoder for audio embedding and the utilization of additional human-generated and synthetic data sets during pre-training. We further experimented with augmenting ClothoV2 captions with available keywords to increase their variety; however, this only led to marginal improvements. Our system ranked first in the 2023's DCASE Challenge, and it outperforms the current state of the art on the ClothoV2 benchmark by 5.6 pp. mAP@10.

Index Terms— Natural-Language-Based Audio Retrieval, PaSST, ChatGPT

1. INTRODUCTION

Natural-language-based audio retrieval revolves around ranking audio recordings based on their relatedness to textual descriptions. Current state-of-the-art methods in this domain are based on the dual-encoder approach which converts both recordings and textual descriptions into high-level representations and then aligns them within a shared audio-caption space. The ranking of candidate audios is carried out by measuring their distance to the textual descriptions in the shared embedding space. The dual-encoder setup has been widely adopted in audio retrieval systems [1–4], because it permits fast ranking and the use of pre-trained audio and text embedding models. CNN architectures pre-trained on AudioSet [5] are the most common audio encoders and large transformer models, such as BERT [6] and RoBERTa [7], are the most popular text encoders. Recently, Mei et al. [8] managed to set a new state-of-the-art performance on ClothoV2 [9] by introducing WavCaps [8], a large dataset with synthetic captions.

In this work, we elaborate on our findings in the context of subtask 6b of the 2023 DCASE challenge, which is concerned with natural-language-based audio retrieval. Our method is also based on the dual-encoder method but differs from previous methods in three central aspects: firstly, we use an audio spectrogram transformer called PaSST [10] instead of CNN14 [5] for audio embedding because PaSST outperforms CNN14 on AudioSet [11] and other relevant audio benchmarks [12]. PaSST employs Patchout [10] during training, which increases training speed and memory efficiency while at the same time acting as a regularizer. Swapping CNN14 for PaSST improves the retrieval performance significantly, as we will show in Section 3.3. Secondly, we pre-train our models on AudioCaps [13] and WavCaps [8], two large audio captioning datasets with human-generated and synthetic captions, to alleviate the data scarcity problem. Using these large datasets to pre-train the models results in significantly better retrieval performance, as we will show in Section 3.1. And thirdly, we take advantage of the available metadata associated with each audio recording and augmented the captions during training by generating additional captions using the available keywords and the GPT3.5-turbo API, also known as ChatGPT. This augmentation method successfully reduces overfitting during fine-tuning but only results in a minor performance improvement (see Section 3.5). Our implementation and the keyword-augmented captions are available in our GitHub repository[1].

2. METHOD

Our model uses separate audio and caption embedding networks, denoted as $\phi_a(\cdot)$ and $\phi_c(\cdot)$, respectively, to embed pairs of spectrograms and descriptions $\{(a_i, c_i)\}_{i=1}^{N}$ into a shared D-dimensional space such that representations of matching audio-caption pairs are close. This behavior is achieved by contrastive training, which makes the embeddings of matching audio-caption pairs (a_i, c_i) more similar while pushing the representations of mismatching pairs $(a_i, c_{j;j\neq i})$ apart. The agreement between audio a_i and description c_j is estimated via the normalized dot product in the shared embedding space:

$$C_{ij} = \frac{\phi_{\text{a}}(a_i)^T \cdot \phi_{\text{c}}(c_j)}{\|\phi_{\text{a}}(a_i)\|^2 \, \|\phi_{\text{t}}(c_j)\|^2}$$

The similarity matrix $\mathbf{C} \in \mathbb{R}^{N\times N}$ holds the agreement of matching pairs on the diagonal and the agreement of mismatching pairs off-diagonal. We train the system using the NT-Xent [14] loss, which is defined as the Cross-Entropy (CE) between the ground truth and the posterior over the text queries and the audio recordings; the ground truth is given by the identity matrix $\mathbf{I} \in \mathbb{R}^{N\times N}$:

$$\mathcal{L} = \frac{1}{2\cdot N}\sum_{i=1}^{N} \text{CE}(\mathbf{C}_{i*}, \mathbf{I}_{i*}) + \text{CE}(\mathbf{C}_{*i}, \mathbf{I}_{*i})$$

---

[1] `https://github.com/OptimusPrimus/dcase2023_task6b`

Instruction	GPT3.5-turbo's Response
I will give a description of an audio recording and tags associated with the audio recording. Generate 5 audio captions describing the sound event. Each audio caption should be one sentence with less than 20 words. Use grammatical subject-verb-object sentences. Do not include names of cities, countries, and persons. Do not include the time. Do not include recording device names. Do not write introductions or explanations. Do not use "heard" or "recorded". Start each output sentence with its index. 'A person is making noise by tapping their fingernails on a solid surface.' [dull, metal]	1. The sound of nails tapping on a metal surface is heard. 2. Fingernails are producing a dull sound when tapped. 3. A person is tapping their nails on a solid, metal surface. 4. The noise of someone tapping their fingernails is audible. 5. Nails are rhythmically tapping on a hard metal object.

Table 1: Example query fed to GPT3.5-turbo to augment a ClothoV2 caption (in blue) with the available keywords (in orange; query inspired by WavCaps [8]). The response is a list of rephrased captions, some of which take the keywords into account (highlighted in orange).

2.1. Audio Embedding Models

We choose the Patchout faSt Spectrogram Transformer (PaSST) [10] to convert audio recordings into a compact, high-level vector representation because it achieves state-of-the-art results on multiple audio classification benchmarks [15] while keeping memory and computational complexity low compared to the vanilla audio spectrogram transformers [16]. PaSST uses ImageNet [17] pre-trained parameters from a vision transformer [18, 19] and fine-tunes them on AudioSet [11] for general-purpose audio tagging. The relatively low computational and memory footprint is achieved by dropping patches from the input sequence. This procedure, called Patchout [10], additionally regularizes the model during training. We conducted experiments with PaSST models that take audios of up to ten seconds in length as input and extract overlapping or non-overlapping patches of size 16 × 16 from the input spectrogram. Pre-trained PaSST models are available on GitHub[2]. We additionally experimented with two convolutional neural networks pre-trained on AudioSet, namely CNN10 and CNN14 [5]. These models can handle inputs of arbitrary length, so we directly input up to 30 seconds long audio instead of cutting them into shorter segments. Table 2 gives an overview of all audio embedding models used in our experiments.

	patch stride	patch out	input length (s)	AS mAP	number params
CNN10	-	-	30	38.0	6.3M
CNN14	-	-	30	43.1	81.8M
PaSST-L	10 × 10	4; 50	10	45.9	41.8M
PaSST-N	16 × 16	2; 15	10	46.8	86.2M
PaSST-S	10 × 10	4; 50	10	48.6	86.2M
PaSST-S20	10 × 10	4; 80	20	47.4	86.2M

Table 2: Overview of the audio embedding models compared in our experiments. The first section shows CNNs from [5]. The second section summarizes PaSST variants [10].

2.2. Sentence Embedding Models

We compared five different sentence embedding models: bert-small, bert-base, bert-large, roberta-base, and roberta-large. All models are bi-directional self-attention-based sentence encoders that underwent self-supervised pretraining on the BookCorpus [21] and WikiText datasets [22]. BERT- and RoBERTa-based models differ in the masking strategy used during training: the former was trained using 10 fixed masks for each sentence, while the latter used new, dynamically generated masks in each forward pass. For both models, we selected the output vector that corresponds to the class token as sentence embedding. The parameter counts for bert-small, bert-base, bert-large, roberta-base, and roberta-large are around 29, 110, 345, 123, and 354 million, respectively.

2.3. Shared Audio-Caption Space

The audio and text embeddings generated by the encoders are integrated into a shared audio-caption space by using a simple linear projection that maps the embedding models' output to a size of 1024. Initial experiments suggested that using a non-linear projection (e.g., a multilayer neural network) does not significantly improve performance.

2.4. Datasets

Our final models were trained in two steps on multiple datasets. First, we performed pretraining on ClothoV2, AudioCaps, and WavCaps. The resulting models were then further finetuned on a custom, augmented version of ClothoV2 (called ClothoV2_GPT below), which also takes into account the available meta-data associated with each audio file.

2.4.1. ClothoV2

ClothoV2 [9] contains 10-30 second-long audio recordings and captions that are between 8 and 20 words long. The development set's training, validation, and test split suggested by the organizers contains 3840, 1045, and 1045 recordings, respectively, and each recording is associated with five human-generated captions. The leaderboard evaluation split used for the final system ranking contains 1000 audio recordings and 1000 captions. We used the validation split to monitor the generalization performance and report the performance on the test split in Section 3.

2.4.2. AudioCaps

AudioCaps [13] contains $51,308$ audio recordings taken from AudioSet and one human-written caption for each of them. Each audio

---

[2] `https://github.com/kkoutini/passt_hear21`

	audio embedding	segment length (s)	overlap	text embedding	finetune	GPT-augment	mAP@10	R@1	R@5	R@10
	PaSST-N	10	✗	bert-small	✗	✗	32.98	21.45	48.71	62.05
	PaSST-N	10	✗	bert-base	✗	✗	35.22	23.07	51.48	65.36
	PaSST-N	10	✗	bert-large	✗	✗	35.78	23.75	52.27	65.57
	PaSST-N	10	✗	roberta-base	✗	✗	35.12	23.02	51.89	65.26
	PaSST-N	10	✗	roberta-large	✗	✗	36.65	24.26	53.89	66.87
	CNN10	30	✗	bert-base	✗	✗	23.72	14.18	36.59	49.21
	CNN14	30	✗	bert-base	✗	✗	28.06	17.86	40.82	54.56
	PaSST-L	10	✓	bert-base	✗	✗	33.47	21.67	49.24	63.16
	PaSST-N	10	✗	bert-base	✗	✗	35.22	23.07	51.48	65.36
	PaSST-S	10	✓	bert-base	✗	✗	32.83	20.90	48.82	62.60
	PaSST-N	10	✗	roberta-large	✓	✗	38.00	25.51	55.06	68.56
	PaSST-N	10	✗	roberta-large	✓	✓	38.56	26.07	55.27	69.30
[20]	CNN14	30	-	all-mpnet-base-v2	✗	✗	22.20	13.00	34.30	48.00
[8]	CNN14	30	-	bert-base	✓	✗	32.95	21.41	47.77	62.10

Table 3: Text-to-audio retrieval performance on the ClothoV2 test set for different combinations of language and audio embedding models (sections one and two, respectively). The impact of additional fine-tuning on ClothoV2 and ClothoV2GPT is shown in section three. Section four shows results from the DCASE baseline system [20] and the current state of the art [8] (values based on WavCaps's GitHub repository).

recording has a duration of 10 seconds, and the captions are, on average, 9.8 words long. We concatenated the training, validation, and testing split of AudioCaps into one large dataset and used it for pretraining.

2.4.3. WavCaps

WavCaps [8] is a weakly-labeled audio-caption dataset that contains $403,050$ audio recordings of varying length collected from FreeSound, BBC Sound Effects, SoundBible, and a strongly labeled subset of AudioSet [23]. Each audio file is associated with a synthetic audio caption that was created by instructing the GPT3.5-turbo model to extract relevant sound events from metadata and output a single-sentence description. The generated captions are, on average, 7.8 words long. The authors demonstrated the usefulness of these synthetic captions by successfully using the dataset for audio retrieval, audio captioning, and text-based sound generation.

2.4.4. ClothoV2_GPT

Each audio recording in the ClothoV2 dataset is associated with additional metadata consisting of the file name, a list of keywords, a username, and a web URL. We took advantage of the additional information and used GPT3.5-turbo to augment the human-generated captions with the available keywords. To this end, we instructed the model to take the keywords into account and rephrase the available captions. We generated five new captions for each caption in the training set, resulting in $96,000$ additional captions. Table 1 gives an example query and the resulting augmented captions: GPT3.5-turbo successfully combined the description and the keywords into five varying descriptions with similar content; four of these contain the provided keywords. We will refer to this augmented version of ClothoV2 as ClothoV2_GPT. The generated captions are available in our GitHub repository.

2.5. Preprocessing

To allow batched processing of recordings of varying lengths, we extracted random 30-second snippets from those audio recordings that are longer than 30 seconds and zero-padded shorter recordings to the maximum duration in the current batch. The resulting waveforms were converted to 128-bin log-MEL spectrograms using a 1024-point FFT (32ms) and hop size of 320 (10ms). The spectrograms were centered and whitened with the approximate global mean and standard deviation before feeding them into the audio embedding model. The input sentences were pre-processed by transforming all characters to lowercase and removing punctuation. The resulting strings were tokenized with the WordPiece tokenizer, padded to the maximum sequence length in the current batch, and truncated if they were longer than 32 tokens.

2.6. Training

We pre-trained the models on AudioCaps, WavCaps, and the training set of ClothoV2. Both embedding models were jointly optimized using gradient descent with a batch size of 64. We used the Adam update rule [24] for 16 epochs, with one warmup epoch. Thereafter, the learning rate was reduced from 2×10^{-5} to 10^{-7} using a cosine schedule. The hyperparameters of the optimizer were set to PyTorch's [25] defaults. We further used structured patchout as a regularizer during training with hyperparameters depending on the audio length and patch extraction (see Table 2). Finetuning on ClothoV2_GPT was done in a similar manner as pretraining but only for five epochs with a maximum learning rate of 8×10^{-6}. During the finetuning procedure, a caption was swapped with one of its five GPT-augmented versions with a probability of 0.3.

3. RESULTS

The performance of different audio and sentence embedding models is summarized in Table 3. The combination of PaSST-N and roberta-large, pre-training on all data sets, and fine-tuning with GPT augmentation outperforms the current state of the art [8] based on bert-base and CNN14 by 5.6 pp. mAP@10. In the following sections, we analyze our method in detail to identify each component's impact on the overall performance.

3.1. Pre-training Data Sets

ClothoV2 is relatively small compared to captioning data sets in the image domain; to further enhance the performance, we additionally leveraged AudioCaps and WavCaps. In this section, we investigate the impact of the additional pre-training data sets on the final performance. To this end, we used PaSST-N and bert-base and trained them on different combinations of the three sets. We report the results in terms of mAP@10 on ClothoV2's test set in Table 4. Pre-training on WavCaps or ClothoV2 results in similar performance of around 27 mAP@10; pretraining exclusively on AudioCaps is roughly 6 pp. worse. Surprisingly, adding AudioCaps to WavCaps did not further improve the result; however, adding ClothoV2 to WavCaps or AudioCaps yielded improvements of 6.8 and 9.6 pp., respectively. The best result overall was achieved by combining all three data sets.

AudioCaps	WavCaps	ClothoV2	mAP@10
✓	✗	✗	21.01
✗	✓	✗	27.62
✗	✗	✓	27.28
✓	✓	✗	27.13
✗	✓	✓	34.42
✓	✗	✓	30.64
✓	✓	✓	**35.22**

Table 4: Ablation study on the effect of pre-training data sets.

3.2. Text Embedding Models

We assumed that larger sentence embedding models would lead to better retrieval performance. To test this hypothesis, we experiment with three variants of BERT and two variants of RoBERTa. The results are summarized in the first section of Table 3. Larger BERT sentence encoders indeed performed better (compare bert-small, bert-base, and bert-large), and a similar trend can be observed for RoBERTa (compare roberta-base and roberta-large). The best overall results were achieved by utilizing roberta-large.

3.3. Audio Embedding Models

We likewise assumed that using a self-attention-based architecture would lead to further improvements. To test this assumption, we compared two convolutional architectures (CNN10 and CNN14) to three recent spectrogram tansformers (PaSST-L, PaSST-N, and PaSST-S); section two of Table 3 summarizes the results. Scaling up the audio embedding model from CNN10 to CNN14 yielded an improvement of 4.3 pp. mAP@10. Switching from CNN14 to PaSST-N further improved the mAP@10 by 7.1 pp. PaSST-S, which extracts overlapping spectrogram patches and performs better on AudioSet, surprisingly did not further improve the retrieval performance over PaSST-N. This inconsistency could be due to a suboptimal patchout configuration.

3.4. Audio Context Length

PaSST uses a learnable positional encoding with a fixed length of 10 seconds and consequently cannot handle longer audio segments. To deal with the up to 30-second long audio recordings in ClothoV2, we cut longer waveforms into shorter segments, embedded each segment separately, and averaged the resulting embeddings over time to obtain a single vector representation. To investigate the impact of the segment length, we conducted experiments by splitting the recordings into 2, 5, 10, and 15 seconds long snippets. We used PaSST-S20 for those experiments, an architecture similar to PaSST-S, but with a positional encoding for audios of up to 20 seconds in length. The results are given in Figure 1. While a longer context is advantageous performance-wise, it also comes at the price of higher computational cost (which grows quadratically with the input size). We find that PaSST's default maximum input length of ten seconds strikes a good balance.

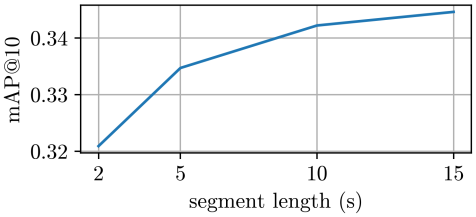


Figure 1: Performance of PaSST-S20 for different audio lengths.

3.5. Fine-Tuning

We further investigated the impact of additional finetuning on the target data sets. To this end, we continued training (as described in Section 2.6) on the training split of ClothoV2 with and without GPT augmentation. The results are given in the third section of Table 3. Finetuning on ClothoV2 without GPT augmentation further improved the pre-trained model by 1.3 pp. mAP@10. When finetuned with GPT-Augmentation, overfitting was slightly reduced, and the model improved by 1.9 pp. mAP@10. A similar advantage for the GPT-augmented data set can be observed for the hidden test set of the 2023's DCASE Challenge.

4. DISCUSSION & CONCLUSION

This work presented a dual-encoder system for automatic audio retrieval, achieving state-of-the-art results on the ColthoV2 benchmark. The results of our experiments attribute the considerable performance gains to two factors: firstly, the additional data sets with human-generated and synthetic captions, and secondly, the audio spectrogram transformer, which scaled better with the additional data compared to convolutional neural networks. Augmentation of the captions with the additional keywords reduced overfitting during finetuning; however, it did not significantly improve retrieval performance. One possible explanation for this is that each recording in the ClothoV2 training set is associated with five different captions, which are likely to contain the most relevant keywords already; adding further captions increases the variety only marginally.

5. ACKNOWLEDGMENT

The LIT AI Lab is financed by the Federal State of Upper Austria. The computational results presented in this work have been partially achieved using the Vienna Scientific Cluster (VSC).

6. REFERENCES

[1] H. Xie, S. Lipping, and T. Virtanen, “Language-based audio retrieval task in DCASE 2022 challenge,” in *Proceedings of the 7th Detection and Classification of Acoustic Scenes and Events 2022 Workshop (DCASE2022)*, Nancy, France, November 2022.

[2] X. Xu, Z. Xie, M. Wu, and K. Yu, “The SJTU system for DCASE2022 challenge task 6: Audio captioning with audio-text retrieval pre-training,” DCASE2022 Challenge, Tech. Rep., 2022.

[3] X. Mei, X. Liu, H. Liu, J. Sun, M. D. Plumbley, and W. Wang, “Language-based audio retrieval with pre-trained models,” DCASE2022 Challenge, Tech. Rep., July 2022.

[4] T. Pellegrini, “IRIT-UPS DCASE 2022 language-based audio retrieval system,” DCASE2022 Challenge, Tech. Rep., July 2022.

[5] Q. Kong, Y. Cao, T. Iqbal, Y. Wang, W. Wang, and M. D. Plumbley, “PANNs: Large-scale pretrained audio neural networks for audio pattern recognition,” *IEEE ACM Trans. Audio Speech Lang. Process.*, 2020.

[6] J. Devlin, M. Chang, K. Lee, and K. Toutanova, “BERT: pre-training of deep bidirectional transformers for language understanding,” in *Proc. of the North American Ch. of the Ass. for Computational Linguistics: Human Language Technologies, NAACL-HLT*, 2019.

[7] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, “Roberta: A robustly optimized BERT pretraining approach,” *CoRR*, vol. abs/1907.11692, 2019.

[8] X. Mei, C. Meng, H. Liu, Q. Kong, T. Ko, C. Zhao, M. D. Plumbley, Y. Zou, and W. Wang, “WavCaps: A chatgpt-assisted weakly-labelled audio captioning dataset for audio-language multimodal research,” *CoRR*, vol. abs/2303.17395, 2023.

[9] K. Drossos, S. Lipping, and T. Virtanen, “Clotho: an Audio Captioning Dataset,” in *Proc. IEEE Int. Conf. Acoustic., Speech and Signal Process., ICASSP*, 2020.

[10] K. Koutini, J. Schlüter, H. Eghbal-zadeh, and G. Widmer, “Efficient training of audio transformers with patchout,” in *23rd Annual Conf. of the Int. Speech Communication Association, Interspeech*, 2022.

[11] J. F. Gemmeke, D. P. W. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, “Audio Set: An ontology and human-labeled dataset for audio events,” in *Proc. IEEE Int. Conf. Acoustic., Speech and Signal Process., ICASSP*, 2017.

[12] J. Turian, J. Shier, H. R. Khan, B. Raj, B. W. Schuller, C. J. Steinmetz, C. Malloy, G. Tzanetakis, G. Velarde, K. McNally, M. Henry, N. Pinto, C. Noufi, C. Clough, D. Herremans, E. Fonseca, J. H. Engel, J. Salamon, P. Esling, P. Manocha, S. Watanabe, Z. Jin, and Y. Bisk, “HEAR: holistic evaluation of audio representations,” in *NeurIPS 2021 Competitions and Demonstrations Track, NeurIPS*, 2021.

[13] C. D. Kim, B. Kim, H. Lee, and G. Kim, “AudioCaps: Generating captions for audios in the wild,” in *Proc. of the North American Ch. of the Ass. for Computational Linguistics: Human Language Technologies, NAACL-HLT*, 2019.

[14] T. Chen, S. Kornblith, M. Norouzi, and G. E. Hinton, “A simple framework for contrastive learning of visual representations,” in *Proc. of the 37nd Int. Conf. on Machine Learning, ICML*, 2020.

[15] K. Koutini, S. Masoudian, F. Schmid, H. Eghbal-zadeh, J. Schlüter, and G. Widmer, “Learning General Audio Representations With Large-Scale Training of Patchout Audio Transformers,” *HEAR: Holistic Evaluation of Audio Representations*, vol. 166, 2022.

[16] Y. Gong, Y. Chung, and J. R. Glass, “AST: audio spectrogram transformer,” in *22nd Annual Conf. of the Int. Speech Communication Association, Interspeech*, 2021.

[17] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and L. Fei-Fei, “ImageNet: A large-scale hierarchical image database,” in *2009 IEEE Computer Society Conf. on Computer Vision and Pattern Recognition CVPR*, 2009.

[18] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, “An image is worth 16x16 words: Transformers for image recognition at scale,” *CoRR*, vol. abs/2010.11929, 2020.

[19] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, “Training data-efficient image transformers & distillation through attention,” in *Proc. of the 38th Int. Conf. on Machine Learning, ICML*, 2021.

[20] “Language-Based Audio Retrieval, Task description,” https://dcase.community/challenge2023/task-language-based-audio-retrieval#description, accessed: 2023-05-21.

[21] Y. Zhu, R. Kiros, R. S. Zemel, R. Salakhutdinov, R. Urtasun, A. Torralba, and S. Fidler, “Aligning books and movies: Towards story-like visual explanations by watching movies and reading books,” in *IEEE Int. Conf. on Computer Vision, ICCV*, 2015.

[22] S. Merity, C. Xiong, J. Bradbury, and R. Socher, “Pointer sentinel mixture models,” in *5th Int. Conf. on Learning Representations, ICLR*, 2017.

[23] S. Hershey, D. P. W. Ellis, E. Fonseca, A. Jansen, C. Liu, R. C. Moore, and M. Plakal, “The benefit of temporally-strong labels in audio event classification,” in *Proc. IEEE Int. Conf. Acoustic., Speech and Signal Process., ICASSP*, 2021.

[24] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in *3rd Int. Conf. on Learning Representations, ICLR*, 2015.

[25] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Z. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, “PyTorch: An imperative style, high-performance deep learning library,” in *Annual Conf. on Neural Information Processing Systems, NEURIPS*, 2019.

FOLEY SOUND SYNTHESIS WITH A CLASS-CONDITIONED LATENT DIFFUSION MODEL

Robin Scheibler, Takuya Hasumi, Yusuke Fujita, Tatsuya Komatsu,
Ryuichi Yamamoto, and Kentaro Tachibana

LINE Corporation, Tokyo, Japan

ABSTRACT

We propose a competitive Foley sound synthesis system based on available components and fine-tuned on a target dataset. We reuse a text-to-audio pre-trained model composed of a latent diffusion model (LDM), trained on AudioCaps, a variational autoencoder (VAE), and a vocoder. We fine-tune the LDM on the development dataset of the DCASE 2023 Task 7 to output a latent representation conditioned on the target class number. The VAE and vocoder are then used to generate the waveform from the latent representation. To improve the quality of the generated samples, we utilize a post-processing filter that selects a subset of generated sounds to match a distribution of target class sounds. In experiments, we found that our system achieved an average Fréchet audio distance (FAD) of 4.744, which is significantly better than 9.702 produced by the baseline system of the DCASE 2023 Challenge Task 7. In addition, we perform ablation studies to evaluate the performance of the system before fine-tuning and the effect of sampling rate on the FAD.

Index Terms— Foley sound synthesis, conditional sound generation, latent diffusion, Fréchet audio distance

1. INTRODUCTION

Foley sound synthesis is the task of generating sound effects added to multimedia content to enhance the perceptual audio experience. Originally developed for cinema and television, it was conducted by skilled artisans using elaborate manual techniques and is named after Jack Donovan Foley, one of their pioneers [1]. Interestingly, Foley sound effects are perceived as more authentic than their real counterparts captured in live recordings [2]. The potential of digital signal processing for sound synthesis was recognized early, as exemplified by the famous Karplus-Strong algorithm [3]. The recent deep learning revolution has brought the realism levels of digital sound synthesis to new levels, in particular for speech [4] and music [5, 6, 7]. A few works have focused specifically on Foley sound synthesis either with video-guidance [8, 9], or without [10, 11, 12, 13]. We can also mention some niche applications like synthesizing cricket sounds as test signals for perceptual experiments [14].

Recently, text-to-audio sound generation based on diffusion models has gained traction. Following a methodology introduced for images [15], AudioLDM [16] has demonstrated impressive conditional sound generation quality. AudioLDM is composed of a latent diffusion model (LDM), a variational autoencoder (VAE), and a neural vocoder. The LDM is conditioned on a text prompt through a Contrastive Language-Audio Pretraining (CLAP) embedding. The latent representation is provided by the VAE which has learned to encode a mel-spectrogram into a compressed latent space. The neural vocoder is based on HiFi-GAN [17], and decodes a waveform from the mel-spectrogram into an uncompressed waveform. Tango [18] has been proposed to enhance the text prompting functionality of AudioLDM using an instruction-tuned large language model (LLM) instead of the CLAP embedding.

Due to this rising interest in general sound generation, a new related task was added to the DCASE 2023 Challenge to stimulate research about this challenging problem. Task 7 [19], aptly called *Foley sound synthesis*, requires participants to build a Foley sound generation model for seven sound classes: dog bark (🐕), footstep (👣), gunshot (🔫), keyboard (💻), moving motor vehicle (🚗), rain (🌧), and sneeze/cough (🤧). The system is then used to produce 100 samples that are first evaluated in terms Fréchet audio distance. In a second stage, a subjective evaluation based on quality, accuracy, and diversity of the samples is conducted. Although text prompting models such AudioLDM or Tango have shown to be effective in fine-grained guidance for audio generation, using them for class conditional generation requires prompt engineering and trial and error. In addition, existing public models have been trained for the generation of 10 s long samples at 16 kHz, while the challenge calls for 4 s samples at 22.05 kHz.

We propose a pragmatic solution to this problem by combining available components to obtain a high quality Foley sound synthesis system. We modify an existing implementation of Tango[1] to enable sound-class-based guidance instead of text prompting. The class-conditioned LDM is trained using the development set of the DCASE 2023 Challenge Task that contains between 600 and 800 sounds of each of the classes. We initialize the model with a pre-trained model of Tango, which was trained with AudioCaps [20] dataset and Flan-T5 [21] LLM. The conditioning part based on Flan-T5 is replaced with a simple linear embedding layer to realize sound-class-based conditioning. Moreover, we propose a post-processing filter that selects a subset of generated samples to match a distribution of the target sound class. The post-processing filter adopts a greedy backward selection strategy that iteratively drops a sample to achieve the minimum Fréchet audio distance (FAD). Our experiments show that our system significantly outperforms the baseline system provided by the task organizers in terms of FAD. Audio samples produced by the system are available online[2].

2. BACKGROUND

Our system, like AudioLDM [16] and Tango [18], is based on the LDM originally proposed for image generation [15]. The LDM operates in the latent-space generated by a VAE pre-trained on mel-spectrograms. The generated mel-spectrograms are inverted into waveforms using the neural vocoder HiFi-GAN [17].

---

[1] https://github.com/declare-lab/tango

[2] http://www.robinscheibler.org/dcase23t7-samples/

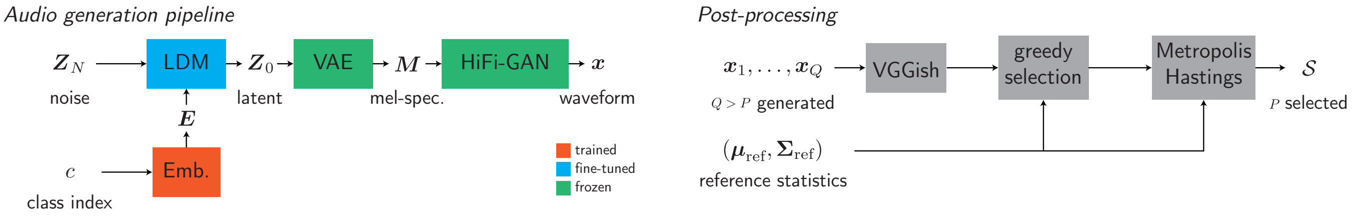


Figure 1: System overview. The audio generation pipeline (left) has three elements. The core is a latent diffusion model (LDM) with class-conditional embeddings (Emb.). We use pre-trained variational-autoencoder and HiFi-GAN vocoder for the reconstruction. The samples produced are then filtered during post-processing (right) by greedy and Metropolis-Hastings optimization.

2.1. Latent Diffusion Models

The LDM transforms a sampled Gaussian noise $\boldsymbol{Z}_N \in \mathbb{R}^{C \times \frac{T}{r} \times \frac{F}{r}}$ into a latent representation $\boldsymbol{Z}_0$ through N reverse diffusion steps with a UNet-based neural network. T is the number of mel-spectrogram frames, F is the number of mel-filter bins, C is the number of channels in latent space, and r is the compression level of VAE. The neural network receives a L-length sequence of d-dimensional embedding vectors $\boldsymbol{E} \in \mathbb{R}^{L \times d}$ transformed from the sound class indices through a linear embedding layer. The conditioner $\boldsymbol{E}$ is fed into the network through the cross-attention mechanism.

Given the latent feature $\boldsymbol{Z}_0$, the corresponding class embedding vector $\boldsymbol{E}$, and random isotropic Gaussian noise $\bar{\boldsymbol{\varepsilon}} \in \mathbb{R}^{C \times \frac{T}{r} \times \frac{F}{r}}$, the neural network is trained to minimize the following loss function $\mathcal{L}$ on the basis of the theory of denoising diffusion probabilistic models [22]:

$$\mathcal{L} = \mathbb{E}_{\bar{\boldsymbol{\varepsilon}}, \boldsymbol{Z}_0, n}\left[\|\bar{\boldsymbol{\varepsilon}} - \boldsymbol{\varepsilon}(\sqrt{\bar{\alpha}_n}\boldsymbol{Z}_0 + \sqrt{1-\bar{\alpha}_n}\bar{\boldsymbol{\varepsilon}}, \boldsymbol{E}, n; \theta)\|_2^2\right], \quad (1)$$

$$\bar{\alpha}_n = \prod_{n'=1}^{n} \alpha_{n'}, \quad (2)$$

$$\alpha_n = 1 - \beta_n, \quad (3)$$

where $\boldsymbol{\varepsilon}(\cdot, \cdot, \cdot; \theta)$ is the neural network that outputs the estimated noise of the same shape as $\boldsymbol{Z}_0$. The variable β_n is the variance of the Gaussian distribution in the forward process.

Classifier-free guidance [23] can be used to boost the fidelity of the sound class. Using this technique, the backward process to obtain $\boldsymbol{Z}_{n-1}$ from $\boldsymbol{Z}_n$ can be written

$$\boldsymbol{Z}_{n-1} = \frac{1}{\sqrt{\alpha_n}}\left(\boldsymbol{Z}_n - \frac{\beta_n}{\sqrt{1-\bar{\alpha}_n}}\tilde{\boldsymbol{\varepsilon}}_n\right) + \sigma_n \tilde{\boldsymbol{\varepsilon}}_n \quad (4)$$

$$\tilde{\boldsymbol{\varepsilon}}_n = w\boldsymbol{\varepsilon}(\boldsymbol{Z}_n, \boldsymbol{E}, n; \theta) + (1-w)\boldsymbol{\varepsilon}(\boldsymbol{Z}_n, \boldsymbol{O}, n; \theta), \quad (5)$$

where σ_n^2 is the variance of the Gaussian distribution in the reverse process, respectively. The symbol $\boldsymbol{O}$ represents the all zero embedding vector used for unconditional inference, and w is a parameter of the guidance scale. Note that, to accelerate the sampling speed at inference time, denoising diffusion implicit models (DDIM) [24] sampling is used.

2.2. Variational autoencoder and neural vocoder

A VAE is used to compress a mel-spectrogram $\boldsymbol{M} \in \mathbb{R}^{T \times F}$ into the latent space parametrized by mean and variance $\mu, \sigma \in \mathbb{R}^{C \times \frac{T}{r} \times \frac{F}{r}}$. The VAE is composed of a stack of CNN-based encoders. In the submitted system pipeline, the latent $\boldsymbol{Z}_0$ produced by the LDM is fed into the decoder of VAE to reconstruct a mel-spectrogram $\boldsymbol{M}$.

To reconstruct a waveform $\boldsymbol{x} \in \mathbb{R}^{T'}$ from a mel-spectrogram $\boldsymbol{M}$ given by the VAE, the generator of HiFi-GAN [17] can be used, where T' is a length of the waveform. The module repeatedly up-samples the mel-spectrograms by a transposed convolution followed by multi-receptive field fusion (MRF). The MRF is composed of residual blocks, where each block processes the inputs by convolutions of multiple kernel sizes and dilations to capture the temporal feature by various receptive fields.

3. PROPOSED SYSTEM OVERVIEW

An overview of our submitted system is depicted in Fig. 1. Our system adopts a similar pipeline with Tango [18], where a latent generator based on LDM, a latent-to-mel decoder using VAE, and a mel-to-wav vocoder are cascaded. Our LDM accepts a sound class index c as a conditioner instead of a text prompt. We use pre-trained VAE and HiFi-GAN models used in AudioLDM [16] to reconstruct a waveform from the latent representation. After the audio generation pipeline, a post-processing filter is employed to drop irrelevant samples to match the distribution of a target sound class. In the following subsections, we describe our implementation of the modules.

3.1. Sound-class-based Conditioning

When training, we initialize our model with a pre-trained checkpoint of Tango. The checkpoint is designed to receive a sequence of embedding vectors $\boldsymbol{E}$ from the Flan-T5 text encoder. We replace the text encoder with a linear embedding layer that projects a sound class index c into a d-dimentional vector. Unlike Tango, we jointly train the conditioner with the main network of LDM. Although the cross-attention mechanism for conditioning accepts a sequence of embedding vectors, which is designed to accept a text sequence, we use a single target class embedding vector as $\boldsymbol{E} \in \mathbb{R}^{1 \times d}$ in this work.

3.2. FAD-oriented Post-processing Filter

The quality of the samples produced by the system, while acceptable, can be improved by over-generating and filtering. For this task, a target sample quality metric is necessary. The FAD metric used in the challenge is an obvious choice. The FAD is computed as follows. First, VGGish [25] embeddings of both the reference and generated samples are computed. The embeddings are computed for segments of 16,000 samples with half-overlap. This produces 10 embedding vectors per 4 s of generated audio. We note that the challenge abuses the metric a little bit since the VGGish model was trained on 16 kHz data, while the challenge uses 22.05 kHz. The mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$ of the embedding vectors of both reference and generated audio are computed and their Fréchet distance [26] is

$$\mathrm{FAD}(\boldsymbol{\mu}_r, \boldsymbol{\Sigma}_r, \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g) = \|\boldsymbol{\mu}_r - \boldsymbol{\mu}_g\|^2 + \mathrm{tr}\left(\boldsymbol{\Sigma}_r + \boldsymbol{\Sigma}_g - 2(\boldsymbol{\Sigma}_r\boldsymbol{\Sigma}_g)^{1/2}\right). \quad (6)$$

To obtain P samples, we first generate Q samples, with $Q > P$. Then, we first reduce the number of samples by greedy selection. We start with the set of all Q samples, denoted $\mathcal{S} = \{1, \ldots, Q\}$. At each iteration, we remove sample k whose absence decreases the FAD most, i.e.,

$$k = \arg\min_{\ell \in \mathcal{S}} \mathrm{FAD}(\boldsymbol{\mu}_r, \boldsymbol{\Sigma}_r, \bar{\boldsymbol{\mu}}_g^{(\mathcal{S}-\ell)}, \bar{\boldsymbol{\Sigma}}_g^{(\mathcal{S}-\ell)}), \quad (7)$$

where $\bar{\boldsymbol{\mu}}_g^{(\mathcal{S}-\ell)}$ and $\bar{\boldsymbol{\Sigma}}_g^{(\mathcal{S}-\ell)}$ are the mean and covariance matrix, respectively, after removing the ℓth sample. Then, we update $\mathcal{S} \leftarrow \mathcal{S} - \{k\}$, where ”$-$” here is the set difference operator. We repeat until the size of $\mathcal{S}$ is P, or no sample can be removed without the FAD increasing.

If we still have more than P samples, we apply the Metropolis-Hastings (MH) algorithm [27] to find a good sub-set of P elements. We initialize the algorithm by uniformly sampling at random 100 subsets of P samples and picking the one with lowest FAD. At each iteration of the algorithm, we randomly swap two samples. We first pick at random one of the current P samples. Then, we pick one of the discarded samples with probability inversely proportional to the embedding distance to the first sample. We swap the two samples and evaluate the FAD. If it decreases, we accept the change. If it increases, we only accept the change with a small probability that decreases over time with a linear schedule. Otherwise, we reject the change. The subset with the lowest FAD over all iterations is returned by the algorithm.

We note that such a filtering system allows to achieve an arbitrarily small FAD, at the cost of generating an increasingly large number of samples. In our final system where $P = 100$, we set $Q = 200$ to strike a balance between FAD performance and generation time.

4. EXPERIMENTS

4.1. Effect of Sampling Rate on FAD

Our generative model operates at 16 kHz and thus requires upsampling to 22.05 kHz to match the dataset. This means that the frequency band from 8 kHz to 11.025 kHz will be empty. We test the effect of this by downsampling the development dataset to 16 kHz and then back up to 22.05 kHz. Table 1 shows the FAD of the development dataset, i.e., computed with test set as reference[3], before and after this operation. The FAD increases by less than 1, which seems acceptable for our purpose. The effect varies by class, and surprisingly the FAD decreases for the *rain* class.

Table 1: Fréchet audio distance (FAD) of the development dataset under several selection procedure and sampling frequencies. The column *full* is the full development set. *100 random* indicates that we chose at random a 100 samples from each class. The result at different stages of the post-filtering a under *optimization*. The latter is further divided into the result of the greedy optimization, which may have more than 100 sample, the random initialization of MH (+*rand.*), and the final MH stage.

class	full	100 random		optimized (22 kHz)		
	22 kHz	22 kHz	16 kHz	greedy†	+rand.	+MH
🐕	1.144	1.883	2.954	0.609	0.802	0.793
👣	2.072	2.388	3.846	0.715	0.862	0.837
🔫	2.606	3.036	4.368	0.667	0.899	0.887
💻	2.772	3.210	3.067	0.441	0.460	0.460
🚗	4.324	5.426	7.950	1.210	1.281	1.281
🌧	3.007	3.624	3.423	0.783	0.902	0.902
🤧	0.400	0.768	1.367	0.188	0.268	0.265
avg.	2.332	2.904	3.854	0.660	0.782	0.775

† More than 100 samples

4.2. FAD-based Post-filtering

We evaluate the post-filtering described in Section 3.2 on the development dataset of the DCASE 2023 Task 7. We evaluate the FAD at different stages of the post-filtering pipeline and show the results in Table 1, under the *optmized* column. We see that the greedy stage is very effective and reduces the FAD to 23.5 % of the random selection. However, there may still be more than 100 samples at this stage. Selecting 100 samples out the ones selected by the greedy approach increases slightly the FAD by about 0.22 points. The effect of the MH algorithm is not always effective, but can further reduce the FAD by up to 0.03 points in the best case.

4.3. Foley Sound Synthesis

4.3.1. Models and Hyperparameters

HiFi-GAN and VAE: We used pre-trained checkpoints of HiFi-GAN and VAE used in [16]. The HiFi-GAN model was trained with AudioSet [28]. All the training data were segmented or padded into 10 seconds and resampled to 16 kHz, i.e., $T' = 160,000$. Each audio sample was transformed into a 64-dim Mel-spectrogram ($F = 64$) with a window length of 1024, and a hop length of 160. The number of frames T was 1024 by padding 24 frames to avoid further padding with downsampling operations in VAE and LDM. The VAE model was trained with AudioSet [28], AudioCaps [20], Freesound [4], and BBCSFX [5]. The compression level r was 4, and the number of channels C was 8.

Baselines: We use two baselines. 1) The challenge baseline systems [19], itself based on [12]. It uses a VQ-VAE for compression to latent space and HiFi-GAN for signal reconstruction. For latent generation, it uses an auto-regressive model based on PixelSNAIL [29]. This model operates at 22.05 kHz. 2) We also evaluate the direct use of Tango [18] for the task. We condition the generation on text prompts designed for each of the target classes — 🐕: “A dog barking”, 👣: “Footsteps”, 🔫: “Gun shot”, 💻: “Typing

---

[3] The mean vector and covariance matrix of the VGGish embeddings of the test set were provided by the task organizers for the purpose of computing the FAD with respect to the hidden test set.

[4] https://freesound.org/

[5] https://sound-effects.bbcrewind.co.uk

Table 2: Fréchet audio distance (FAD) with baseline and our systems. 'raw' indicates the system without the FAD filter, i.e., the first 100 samples from the audio generation pipeline were used. 'filtered' indicates our submitted system with the FAD filter.

class	baseline	Tango	ours	
	[19]	[18]	raw	filtered
🐕	13.411	6.031	5.835	3.816
👣	8.109	11.889	11.209	8.227
🔫	7.951	8.082	5.790	3.427
💻	5.230	11.014	3.698	2.758
🚗	16.108	14.636	11.440	6.837
🌧️	13.337	8.550	7.031	5.399
🤧	3.770	9.450	3.658	2.741
avg.	9.702	9.450	6.952	4.744

a keyboard", 🚗: "motor vehicle moving", 🌧️: "Heavy rain", and 🤧: "woman sneezing." Since Tango produces 10 s long samples at 16 kHz, we truncate them to 4 s and upsample to 22.05 kHz.

Proposed System: We initialized our LDM using a checkpoint of Tango [6]. The model used the conditioning vector dimension $d = 1024$. The initial checkpoint was trained with AudioCaps [20]. For fine-tuning, we used the DCASE2023 Task7 development set. Since the audio data were sampled at 22.05 kHz and segmented in four seconds, we resampled them to 16 kHz and padded them into 10 seconds. We set $N = 1000$ forward diffusion steps for fine-tuning. Our LDM was fine-tuned with the AdamW optimizer with an initial learning rate of 3e-5 and a linear decay learning rate scheduler. We fine-tuned the model for 100k training iterations, with an effective batch size of 42 using seven A100 GPUs. In the inference phase, we used DDIM [24] for 100 sampling steps and a classifier-free guidance scale of $w = 3$. As our model produces a 10-second audio segment at a 16 kHz sampling rate, we extracted the first four-second segment and resampled it to 22.05 kHz to fit the challenge rule.

Post-processing: For each sound class, we generated $Q = 200$ samples with the aforementioned audio generation pipeline. Then the FAD filter is applied to reduce the number of samples to $P = 100$.

4.3.2. Results

Table 2 shows FAD of the baselines and our system with respect to the evaluation set. First, we observe that the Tango model, which was trained on a much larger dataset, but did not include the Task 7 development set, performs slightly better than the baseline. This shows the effectiveness of a large, diverse dataset. It is likely that further prompt engineering would improve the result. The improvement seems to come from large reductions of FAD on the dog bark and rain classes. We note that both the baseline and Tango could be improved by the post-filtering, which we did not try.

Our fine-tuned model conditioned on class embeddings performs better accross the board, with the exception of footstep, where the baseline is better. However, informal listening tests revealed the samples to be of good quality regardless. We also point out that the keyboard class nearly saturates the lower bound FAD of 100 random samples of the development set at 16 kHz (see Table 1). We also see that the moving motor vehicle class is fairly difficult, as suggested by the FAD of this class in the development set.

With the FAD filter, the FADs were significantly reduced regardless of the sound classes. Specifically, we achieve a reduction of 32 % of the average FAD compared to the raw outputs. The results demonstrate that the proposed audio generation pipeline can generate class-specific audio samples with sufficient diversity, and that the proposed FAD filter can select a subset of them with the statistics of the target sound class.

5. CONCLUSION

We proposed a system based on class-conditioned latent diffusion model for the DCASE2023 Task7: Foley sound synthesis. We efficiently make use of existing models that we adapt to the task and pair with some signal processing for adaptation and output filtering. We fine-tune pre-trained model for text-to-audio generation on the development set of Task 7, and modifies its conditioning mechanism to use class embedding vectors. Our post-filtering system uses greedy and global optimization to select a set of samples matching the statistics of the target evaluation set and decrease the FAD. We found this technique very successful. However, we also noticed during preliminary experiments that the audio quality of samples selected this way did not always match the FAD score obtained. This highlights the importance of generating good samples in the first stage of the system. Overall, our submission system achieved significantly better FAD scores than both the challenge baseline and an out-of-the-box text-to-audio model.

An important aspect that is yet to be understood is how to leverage unlabelled data for pre-training, since training sets for sound effects may be small depending on the target class.

6. REFERENCES

[1] S. Pauletto, "Foley performance and sonic implicit interactions," in *The Body in Sound, Music and Performance: Studies in Audio and Sonic Arts*, L. O Keefe and I. Nogueira, Eds. Abingdon, Oxon: Routledge, 2023.

[2] L. M. Heller and L. Wolf, "When hybrid sound effects are better than real recordings," *Proc. Meet. Acoust.*, vol. 46, no. 1, p. 050002, May 2022.

[3] K. Karplus and A. Strong, "Digital synthesis of plucked-string and drum timbres," *Computer Music Journal*, vol. 7, pp. 43–55, 1983.

[4] X. Tan, T. Qin, F. Soong, and T.-Y. Liu, "A survey on neural speech synthesis," July 2021, arXiv:2106.15561 [cs, eess].

[5] C. Hernandez-Olivan and J. R. Beltrán, "Music Composition with Deep Learning: A Review," in *Advances in Speech and Music Technology: Computational Aspects and Applications*, ser. Signals and Communication Technology, A. Biswas, E. Wennekes, A. Wieczorkowska, and R. H. Laskar, Eds. Cham: Springer International Publishing, 2023, pp. 25–50.

[6] A. Agostinelli, T. I. Denk, Z. Borsos, *et al.*, "MusicLM: Generating music from text," Jan. 2023, arXiv:2301.11325 [cs, eess].

[7] M. Pasini and J. Schlüter, "Musika! Fast infinite waveform music generation," in *Proc. ISMIR*, Bengaluru, IN, Dec. 2022.

---

[6] https://huggingface.co/declare-lab/tango

[8] S. Andreu and M. Villanueva Aylagas, “Neural Synthesis of Sound Effects Using Flow-Based Deep Generative Models,” *Proc. AAAI Conf. Artif. Intell. Interact. Digit. Entertain.*, vol. 18, no. 1, pp. 2–9, Oct. 2022.

[9] S. Ghose and J. J. Prevost, “AutoFoley: Artificial synthesis of synchronized sound tracks for silent videos with deep learning,” *IEEE Trans. Multimedia*, vol. 23, pp. 1895–1907, June 2021.

[10] A. Barahona-Rıos and S. Pauletto, “Synthesising knocking sound effects using conditional WaveGAN,” in *Proc. 17th Sound and Music Computing Conference*, Torino, IT, June 2020.

[11] S. Li, L. Zhang, C. Dong, *et al.*, “FastFoley: Non-autoregressive Foley Sound Generation Based on Visual Semantics,” in *Man-Machine Speech Communication*, ser. Communications in Computer and Information Science, L. Zhenhua, G. Jianqing, Y. Kai, and J. Jia, Eds. Singapore: Springer Nature, 2023, pp. 252–263.

[12] X. Liu, T. Iqbal, J. Zhao, *et al.*, “Conditional sound generation using neural discrete time-frequency representation learning,” in *Proc. MLSP*, Oct. 2021, pp. 1–6.

[13] S. Pascual, G. Bhattacharya, C. Yeh, J. Pons, and J. Serrà, “Full-band general audio synthesis with score-based diffusion,” in *Proc. ICASSP*, Rhodes, GR, June 2023, pp. 1–5.

[14] M. Oliveira, V. Almeida, J. Silva, and A. Ferreira, “Analysis and re-synthesis of natural cricket sounds assessing the perceptual relevance of idiosyncratic parameters,” in *Proc. ICASSP*, Rhodes, GR, June 2023, pp. 1–5.

[15] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, “High-resolution image synthesis with latent diffusion models,” in *Proc. IEEE ICCV*, 2022, pp. 10 684–10 695.

[16] H. Liu, Z. Chen, Y. Yuan, *et al.*, “AudioLDM: Text-to-audio generation with latent diffusion models,” *arXiv preprint arXiv:2301.12503*, 2023.

[17] J. Kong, J. Kim, and J. Bae, “HiFi-GAN: Generative adversarial networks for efficient and high fidelity speech synthesis,” in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 17 022–17 033.

[18] D. Ghosal, N. Majumder, A. Mehrish, and S. Poria, “Text-to-audio generation using instruction tuned LLM and latent diffusion model,” *arXiv preprint arXiv:2304.13731*, 2023.

[19] K. Choi, J. Im, L. Heller, *et al.*, “Foley sound synthesis at the DCASE 2023 challenge,” *arXiv preprint arXiv:2304.12521*, 2023.

[20] C. D. Kim, B. Kim, H. Lee, and G. Kim, “AudioCaps: Generating captions for audios in the wild,” in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Minneapolis, Minnesota, June 2019, pp. 119–132.

[21] H. W. Chung, L. Hou, S. Longpre, *et al.*, “Scaling instruction-finetuned language models,” *arXiv preprint arXiv:2210.11416*, 2022.

[22] J. Ho, A. Jain, and P. Abbeel, “Denoising diffusion probabilistic models,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 6840–6851, 2020.

[23] J. Ho and T. Salimans, “Classifier-free diffusion guidance,” *arXiv preprint arXiv:2207.12598*, 2022.

[24] J. Song, C. Meng, and S. Ermon, “Denoising diffusion implicit models,” in *Proc. ICML*, 2021. [Online]. Available: https://openreview.net/forum?id=St1giarCHLP

[25] S. Hershey, S. Chaudhuri, D. P. W. Ellis, *et al.*, “CNN architectures for large-scale audio classification,” in *Proc. ICASSP*, New Orleans, LA, USA, Mar. 2017, pp. 131–135.

[26] D. C. Dowson and B. V. Landau, “The Fréchet distance between multivariate normal distributions,” *Journal of Multivariate Analysis*, vol. 12, no. 3, pp. 450–455, Sept. 1982.

[27] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, “Equation of state calculations by fast computing machines,” *The Journal of Chemical Physics*, vol. 21, no. 6, pp. 1087–1092, June 1953.

[28] J. F. Gemmeke, D. P. Ellis, D. Freedman, *et al.*, “Audio Set: An ontology and human-labeled dataset for audio events,” in *Proc. ICASSP*. New Orleans, LA, USA: IEEE, 2017, pp. 776–780.

[29] X. I. Chen, N. Mishra, M. Rohaninejad, and P. Abbeel, “PixelSNAIL: An improved autoregressive generative model,” in *Proc. ICML*. PMLR, July 2018, pp. 864–872.

DISTILLING THE KNOWLEDGE OF TRANSFORMERS AND CNNS WITH CP-MOBILE

Florian Schmid[1], *Tobias Morocutti*[2], *Shahed Masoudian*[1], *Khaled Koutini*[2], *Gerhard Widmer*[1,2]

[1]Institute of Computational Perception (CP-JKU), [2]LIT Artificial Intelligence Lab,
Johannes Kepler University Linz, Austria
{florian.schmid, tobias.morocutti, shahed.masoudian}@jku.at

ABSTRACT

Designing lightweight models that require limited computational resources and can operate on edge devices is a major trajectory in deep learning research. In the context of Acoustic Scene Classification (ASC), the DCASE community hosts an annual challenge on low-complexity ASC, contributing to the research on Knowledge Distillation (KD), Model Pruning, Quantization and efficient neural network design. In this work, we propose a system that contributes to the latter by introducing CP-Mobile, a lightweight CNN architecture constructed of residual inverted bottleneck blocks and Global Response Normalization. Furthermore, we improve Knowledge Distillation by showing that ensembling CNNs and Audio Spectrogram Transformers form strong teacher ensembles. Our proposed system improves the results on the *TAU Urban Acoustic Scenes 2022 Mobile development dataset* by around 5 percentage points in accuracy compared to the top-ranked submission for Task 1 of the DCASE 22 challenge and achieves the top rank in the DCASE 23 challenge[1].

***Index Terms*—** CP-Mobile, Receptive Field Regularization, Patchout FaSt Spectrogram Transformer (PaSST), CP-ResNet, Knowledge Distillation, Device Impulse Response augmentation, Freq-MixStyle

1. INTRODUCTION

The task of Acoustic Scene Classification (ASC) is to assign a scene label to an audio clip. The *Low-Complexity Acoustic Scene Classification* task of the DCASE 23 challenge [1] is based on the *TAU Urban Acoustic Scenes 2022 Mobile development dataset (TAU22)* [2], consisting of 1-second audio clips, each belonging to one of 10 different acoustic scenes. Audio clips are recorded by three real devices and six simulated devices, including three simulated devices that are not included in the train split, making device generalization an important and challenging task. The challenge further introduces limits on the model size (128 kB) and the computational complexity in terms of multiply-accumulate operations (30 million MACs). Systems are ranked according to class-wise averaged accuracy, consumed MACs for the inference of a 1-second audio clip, and the model size, encouraging participants to design models with a good performance-complexity trade-off.

ASC Architectures: Convolutional Neural Networks (CNNs) are well-established models to tackle low-complexity ASC and dominated the leaderboard in previous editions of the challenge [1–3]. Common practice is to regularize the receptive field of CNNs [4, 5], which has been shown to improve the generalization performance. Particularly successful implementations of receptive-field regularized CNNs (RFR-CNNs) include BC-ResNet [6, 7] and CP-ResNet [8–10]. Recently, Audio Spectrogram Transformers achieved competitive results on multiple downstream tasks in the audio domain, including the Patchout FaSt Spectrogram Transformer (PaSST) [11] achieving state-of-the-art results on the *TAU Urban Acoustic Scenes 2020 Mobile development dataset (TAU20)* [2].

Efficient Model Design: A substantial amount of prior work exists on making conventional CNNs more efficient by factorizing convolution operations. In this regard, MobileNets [12, 13] and EfficientNets [14], introduced in the vision domain, have been successfully adapted to the audio domain [15, 16]. MobileNets and EfficientNets are based on inverted bottleneck blocks and inspire CP-Mobile, introduced in Section 3.

Model Compression Techniques: Besides designing efficient architectures, model compression techniques such as Parameter Pruning [17, 18], Quantization [19, 20] and Knowledge Distillation (KD) [21, 22] are popular to reduce a system's complexity further. Quantization to 8-bit precision was forced by the DCASE 22 challenge [1] rules, Parameter Pruning has been successfully applied to ASC systems [6, 9, 23], and KD has been the most successful technique in previous editions of the challenge with the top 3 teams using KD in the DCASE 22 and 23 challenges [1].

Device Generalization Methods: Many different approaches have been applied to counter the distribution shift caused by the unseen devices at test time. In this regard, Domain Adaptation has been used to force device-invariant representations extracted by the model [8, 24]. Other approaches tried to train device translators [6], change the sampling frequency of devices [7], or remove device-specific information by normalization [25]. An augmentation technique called Freq-MixStyle (FMS) [25, 26] lead to the best performance on unseen devices in the DCASE 22 challenge, which recently has been paired with device impulse response (DIR) augmentation to boost the performance further [27].

In this work, we propose a new ASC system, outperforming the top-ranked system in the DCASE 22 challenge by 5% accuracy on the TAU22 development dataset and achieving the top rank in the DCASE 23 challenge. The main contribution of our ASC system is twofold: 1) we achieve a new state-of-the-art teacher model performance by ensembling Audio Spectrogram Transformers and CNNs trained with different FMS and DIR settings, and 2) we introduce CP-Mobile, an efficient, factorized CNN that can distill the knowledge of the large teacher ensemble under low-complexity limits. We introduce the teacher ensemble in Section 2, CP-Mobile in Section 3 and connect them in the KD setup described in Section 4. The results are presented in Section 5, including a detailed ablation study assessing the components of our system.

---

[1] Source Code: `https://github.com/fschmid56/cpjku_dcase23`

2. TEACHER ENSEMBLE: PASST AND CP-RESNET

Audio spectrogram transformer models such as PaSST [11] are purely self-attention-based models, making them excellent at capturing the global context of an audio clip. PaSST has been shown to be a good teacher model for low-complexity CNNs [10,16,26]. CP-ResNet (CPR) [4], however, is a RFR-CNN that gradually builds local features covering a spatially restricted size before applying a global pooling operation.

Experiments in [26] and [16] show that high-performing ensembles can be achieved by ensembling PaSST models trained with varying FMS [25,26] and model configurations. To further increase the diversity of predictions in the ensemble, we experiment with including models trained with DIR augmentation [27] and CPR models. We follow the model configurations and training protocols used in [27] and use a CPR with 128 base channels, resulting in a model with approximately 4M parameters. We finetune the AudioSet [28] pre-trained PaSST, consisting of 85M parameters, on the TAU22 dataset, using a structured patchout of 6 on the frequency dimension. In addition to the training protocol of [27], we augment TAU22 by adding shifted crops of the reassembled 10-second audio clips, as proposed in [29]. PaSST and CPR models are trained in 4 different configurations: 1) using no device generalization method, 2) using DIR, 3) using FMS and 4) using the combination of DIR and FMS. Hyperparameters for DIR and FMS are chosen according to [27] and set to $\alpha = 0.4$, $p_{FMS} = 0.4$ and $p_{DIR} = 0.6$ for PaSST and to $\alpha = 0.4$, $p_{FMS} = 0.8$ and $p_{DIR} = 0.4$ for CPR. The results for the teacher ensembles are presented in Section 5.1.

3. STUDENT MODEL: CP-MOBILE

In this section, we introduce CP-Mobile (CPM), a novel efficient architecture for ASC. The goal is to maintain beneficial properties from CPR [4,5], such as the regularized receptive field, while reducing the complexity and factorizing convolution operations, such as in MobileNets [12,13] or EfficientNets [14]. Given that the teacher ensemble consists of multiple millions of parameters, an important point is to increase the student model's capacity to be able to distill as much knowledge as possible from the teacher ensemble to the student, even in a low-complexity setting.

Input	Operator	Stride
256 x 64 x 1	Conv2D@3x3, BN, ReLU	2 x 2
128 x 32 x BC/4	Conv2D@3x3, BN, ReLU	2 x 2
64 x 16 x BC	CPM Block S	1 x 1
64 x 16 x BC	CPM Block D	2 x 2
32 x 8 x BC	CPM Block S	1 x 1
32 x 8 x BC	CPM Block T	2 x 1
16 x 8 x BC*CM	CPM Block S	1 x 1
16 x 8 x BC*CM	CPM Block T	1 x 1
16 x 8 x BC*CM²	Conv2D@1x1, BN	
16 x 8 x CLS	Avg. Pool	

Table 1: CP-Mobile Architecture: *Input*: frequency bands x time frames x channels; *Conv2D@KxK*: Conv2D with kernel size KxK; *BC*: Base Channels; *CM*: Channels Multiplier; *CPM Block S/D/T*: Standard/Downsampling/Transition; *CLS*: Number of Classes

First, we factorize all 3x3 convolution operations in CPR into a pointwise expansion convolution, a depthwise convolution and a pointwise projection convolution and obtain residual inverted bottleneck blocks (referred to as *CPM blocks* in the following). We replace the max pool operations with strided convolutions to downsample the spatial dimensions. All shortcut paths that require an additional pointwise convolution are removed and the strided input convolution is split into two separate strided convolutions to reduce the computational burden when operating on the high-dimensional input spectrograms. We experiment with Relaxed Instance Frequency-wise Normalization [25], SubSpectral Normalization [30] and Global Response Normalization (GRN) [31] integrated into different positions in the CPM blocks. While substantial improvements for multiple normalization and position combinations can be achieved, using GRN after adding the shortcut and before the final ReLU activation leads to the highest performance gain.

Table 3 shows the architecture of CPM. CPM's complexity scales in four dimensions: number of blocks (depth), number of base channels (BC), network width modified using the channels multiplier (CM) and expansion rate of inverted bottlenecks (EXP). The depth of the network and the strides determine the receptive field of the model. The overall spatial downsampling factor and the position of the strided convolutions are inspired by the max pooling layer positions in the low-complexity CP-ResNet in [10]. Experimenting with CPM models of varying depths, we find that using 6 CPM blocks creates a suitable receptive field size.

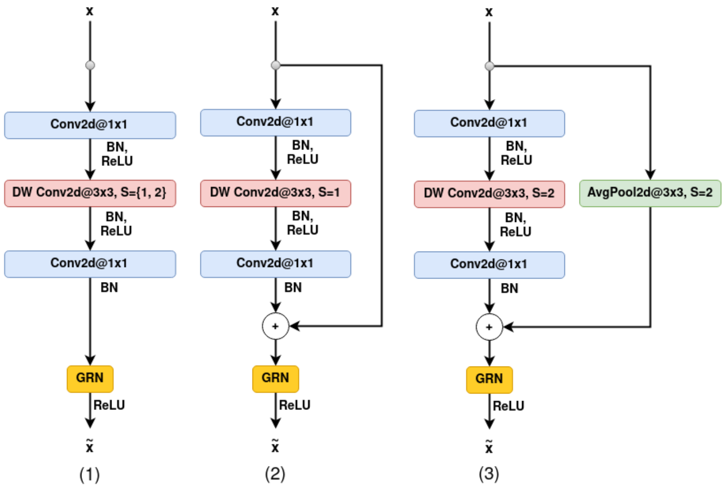


Figure 1: CPM blocks: (1) Transition Block (input channels $\neq$ output channels), (2) Standard Block, (3) Spatial Downsampling Block (S denotes stride)

Figure 1 depicts the structure of a CPM block consisting of two pointwise and a depthwise convolution. The depthwise convolution operates on the expanded channel representation, which has the size of the number of block input channels times the scaling factor EXP. We differentiate between Transition, Standard and Spatial Downsampling blocks (CPM blocks T, S, D). CPM block T increases the channel dimension, uses no residual connection and can be used with a strided depthwise convolution. CPM blocks S and D have matching input and output channel dimensions and use a residual connection. CPM block D uses average pooling with a kernel size of 3 and a stride of 2 on the shortcut path to match the spatial dimensions of the block output. GRN [31] is applied before the final ReLU activation. GRN calculates a normalization value $\mathcal{N}_i$ for each channel, where $||X_i||$ is the L2-norm of channel i:

$$\mathcal{N}_i = \frac{||X_i||}{\sum_c^C ||X_c||/C} \tag{1}$$

The normalization values $\mathcal{N}_i$ are used to calibrate the channel responses, including two trainable parameters γ and β and a residual connection: $\hat{X}_i = \gamma * \mathcal{N}_i * X_i + \beta + X_i$. GRN was introduced in [31] to increase the feature diversity across channels. The main consideration for using GRN in CPM is to avoid feature redundancies in models with restricted capacity.

4. KNOWLEDGE DISTILLATION SETUP

Similar to [16], CPM is trained on the one-hot encoded labels and the pre-computed predictions of the teacher ensemble described in Section 5.1. Compared to the hard labels, the teacher soft labels describe blurred decision boundaries and establish important similarity structures between classes. The loss, consisting of a combination of hard label loss L_l and distillation loss L_{kd}, is given in Equation 2. λ is a weight that trades off label and distillation loss, z_S and z_T are student and teacher logits, y denotes the hard labels and τ is a temperature to control the sharpness of the probability distributions created by the softmax activation δ. L_l is the Cross-Entropy loss and Kullback-Leibler divergence is used as distillation loss L_{kd}.

$$Loss = \lambda L_l(\delta(z_S), y) + (1-\lambda)\tau^2 L_{kd}(\delta(z_S/\tau), \delta(z_T/\tau)) \tag{2}$$

4.1. Experimental Setup

Preprocessing: For training the student models, the raw audio is downsampled to 32 kHz and Mel spectrograms with 256 frequency bins are computed. Short-Time Fourier Transformation is applied with a window size of 96 ms and a hop size of 16 ms. Increasing the window size from 64 to 96 ms and applying a 4096-point FFT leads to a slight improvement compared to [10], as shown in Table 4 (*large FFT window*).

Training: CPM student models are trained for 75 epochs on the TAU22 dataset with the shifted crops dataset augmentation described in [29]. We use a batch size of 256, Adam optimizer [32] and a learning rate scheduler that increases the learning rate to its peak value until epoch 7 and linearly decreases it from epoch 25 to 67 to 0.5% of the peak value. The peak learning rate varies for models of different sizes and is listed in Table 3. For device generalization, we use FMS [25, 26] and DIR augmentation [27] and set the hyperparameters $\alpha = 0.4$, $p_{FMS} = 0.4$ and $p_{DIR} = 0.6$. For KD [21], setting $\tau = 2$ and using a high weight on the distillation loss with $\lambda = 0.02$ turned out beneficial.

5. RESULTS

Below, we give the results of the teacher ensembles, analyze the performance of CPM models scaled to different complexity levels and offer a detailed ablation study of our system's main components.

5.1. Teacher Ensemble Results

Table 2 lists CPR and PaSST models trained with different DIR and FMS configurations and the accuracies achieved by individual models and the respective ensembles. Rows starting with *Configs* specify the combination of PaSST and CPR models or models trained with different FMS and DIR settings. The models in the *Configs* ensembles are chosen randomly from the pool of available models, such that each config, indicated by the superscript, is equally represented. All ensembles are created by averaging the logits of the individual models and # specifies the number of models in the ensemble.

Besides the known fact [27] that device generalization via FMS and DIR improves the accuracy substantially compared to the baselines (^{1} and ^{5}), two important observations can be made:

1) Training with different device generalization methods leads to models with varying device expertise, increasing the ensemble's diversity. Therefore, ensembles consisting of models trained with different settings for FMS and DIR outperform ensembles consisting of models trained with the same configuration. This effect is more dominant for CPR, where the setting *Configs:* ^{2,3,4} improves by 0.74% accuracy over the *CPR*^{4} configuration, even though the individual models that make up the ensemble score on average 1.24% lower in accuracy compared to the *CPR*^{4} setting.

2) The ensemble's diversity is further extended to different views on the data. CPR focuses on building local features while PaSST models the global context. Independent of the device generalization method, ensembling PaSST and CPR leads to a substantial performance improvement with the ensembles *Configs:* ^{1,5} and *Configs:* ^{4,8} outperforming the individual models that make up the ensemble by around 5% accuracy.

Model Config				Ensemble	
Model	**FMS**	**DIR**	**Acc.**	**#**	**Acc.**
CPR^1	✗	✗	$56.40_{\pm 0.18}$	3	57.47
CPR^2	✗	✓	$58.96_{\pm 0.21}$	3	60.06
CPR^3	✓	✗	$62.27_{\pm 0.22}$	3	63.22
CPR^4	✓	✓	$62.56_{\pm 0.33}$	3	63.74
Configs: ^{2,3,4}			$61.32_{\pm 1.67}$	3	64.48
PaSST^5	✗	✗	$59.48_{\pm 0.64}$	3	60.99
PaSST^6	✗	✓	$61.55_{\pm 0.05}$	3	62.51
PaSST^7	✓	✗	$61.08_{\pm 0.38}$	3	62.06
PaSST^8	✓	✓	$62.19_{\pm 0.15}$	3	63.28
Configs: ^{6,7,8}			$61.82_{\pm 0.40}$	3	63.37
Configs: ^{1,5}			57.48	2	62.52
Configs: ^{4,8}			62.40	2	67.30
Configs: ^{2,3,4,6,7,8}			$61.49_{\pm 1.30}$	12	**68.16**

Table 2: Results of the teacher models CPR and PaSST and the respective ensembles on TAU22 [2]. The *Model Config* section indicates the configuration and the average accuracy and standard deviation of individual models. The *Ensemble* section lists the ensemble size (#) and the accuracy achieved by the ensemble.

For building the final teacher ensemble, we exploit both observations. *Configs:* ^{2,3,4,6,7,8} is constructed of 6 CPR and 6 PaSST models, each including 2 models using DIR, 2 using FMS and 2 using DIR and FMS. Constructing even larger ensembles does not improve the accuracy considerably. This final ensemble achieves an accuracy of 68.16%, leading to an improvement of approximately 5.3% accuracy compared to the PaSST-only teacher ensemble used in the top-ranked submission for the DCASE 22 challenge (62.82%) [10]. We generate the predictions for the TAU22 development set and the added shifted crops [29], average the logits of the

12 models and reuse them to train our CPM students.

5.2. Student Models at Different Scales

Table 3 shows CPM models with different model scaling hyperparameters **BC**, **CM** and **EXP**, as introduced in Section 3. We find that the number of base channels **BC** should be adapted to the required complexity level, e.g., models below 10k parameters achieve the best performance with **BC=8**, while models with **BC=32** work best for models above 50k parameters. While small accuracy improvements can be achieved when scaling up **CM** and **EXP**, the performance quickly saturates for values larger than 2. To achieve the best performance, the learning rate needs to be increased for smaller models.

All accuracies presented in Table 3 are based on models quantized to 8-bit precision. The Quantization Aware Training [20] applied to CPM models is detailed in [29]. Our smallest model outperforms the DCASE baseline system [1] by almost 10% accuracy while requiring only around 12% of the model size and 5% of MACs. The largest CPM model presented achieves an accuracy of 63.21%, improving the accuracy by around 4% compared to the top-ranked system [10] of the DCASE 22 challenge [1] while being more than two times smaller in terms of model size and requiring around 50% of the number of MACs.

Model	**BC**,**CM**,**EXP**,**LR**	**Size (B)**	**MMACs**	**Acc.**
CPM	8,2.1,1.7,0.003	5,722	1.58	52.61$_{\pm 1.25}$
CPM	16,1.5,1.75,0.003	12,310	4.35	58.42$_{\pm 0.51}$
CPM	24,1.5,1.9,0.002	30,106	9.64	61.77$_{\pm 0.54}$
CPM	32,1.7,1.9,0.001	54,182	16.80	63.21$_{\pm 0.44}$
DCASE BL. [1]		46,512	29.23	42.9$_{\pm 0.77}$

Table 3: **BC**, **CM** and **EXP** are model scaling hyperparameters introduced in Section 3 and **LR** denotes the learning rate. **Model Size** is given in Bytes after quantization and **MMACs** specifies million multiply-accumulate operations required for the inference of a 1-second audio clip. The presented accuracies are reported in terms of the mean and standard deviation of 3 independent runs.

5.3. Ablation Study

Table 4 presents an ablation study of our system using a CPM with scaling factors BC=32, CM=2.3 and EXP=3, resulting in a model with 127k parameters and 29 million MACs. Removing one component at a time, the results reveal that KD, and even more, the new CPM architecture, are the dominating performance factors. In the following, the results are analyzed in detail.

CPM: The setting "- CP-Mobile" indicates the use of the low-complexity CP-ResNet used in the top-ranked submission for DCASE 22 [10] integrated into our setup. CPM outperforms CPR by 4.54% in accuracy while the two models are of comparable complexity, demonstrating the increased model capacity of CPM to distill knowledge from the teacher ensemble. GRN is an integral part of the CPM blocks, improving accuracy by 1.53% and the residual connections are also an important factor accounting for an increase of 1% in accuracy.

KD: KD is an important component of our system, increasing the accuracy by 3.41%. However, using no KD, CPM still performs only 0.31% worse in accuracy than the best single teacher model (CPR^{4}) while having only 3.2% of its parameters, underlining the efficiency of CPM. Excluding the PaSST or CPR models from the teacher ensemble leads to a drop in accuracy of 0.81% and 1.22%, respectively, showing that the student benefits from the performance gain of ensembling Transformers and CNNs but can not fully exploit the large improvement of the teacher ensemble.

Device Generalization: The results underline that the combination of DIR and FMS to tackle device generalization works best and using no device generalization method leads to a severe performance drop (-1.87% accuracy). In particular, the ability to generalize to unseen devices suffers with the unseen accuracy dropping by 4.18% in terms of accuracy when neither DIR, nor FMS is used.

Augmentation and Preprocessing: Using a larger FFT window size compared to the setup used in [10] and applying the shifted crop dataset augmentation introduced in [29] improves the system's performance slightly.

System	**Accuracy**	**Acc. Diff**	**Unseen Acc.**
Our Proposed System	**65.66**$_{\pm 0.88}$	Ref. Val.	61.68$_{\pm 1.15}$
- CP-Mobile	61.12$_{\pm 0.44}$	-4.54	57.45$_{\pm 0.63}$
- GRN	64.13$_{\pm 0.58}$	-1.53	60.51$_{\pm 0.88}$
- Residual Connections	64.65$_{\pm 0.23}$	-1.01	61.07$_{\pm 0.38}$
- KD	62.25$_{\pm 0.41}$	-3.41	56.72$_{\pm 0.23}$
- PaSST teachers	64.85$_{\pm 0.21}$	-0.81	60.70$_{\pm 0.51}$
- CP-ResNet teachers	64.44$_{\pm 0.37}$	-1.22	61.19$_{\pm 0.68}$
- DIR	64.74$_{\pm 0.33}$	-0.92	59.99$_{\pm 0.23}$
- FMS	65.15$_{\pm 0.36}$	-0.51	60.05$_{\pm 0.59}$
- DIR, FMS	63.79$_{\pm 0.39}$	-1.87	57.50$_{\pm 0.64}$
- large FFT window	65.29$_{\pm 0.04}$	-0.37	61.68$_{\pm 0.34}$
- Shifted Crops	65.28$_{\pm 0.11}$	-0.38	61.73$_{\pm 0.07}$

Table 4: Ablation Study of our proposed setup using CPM (127k params, 29 million MACs) and removing one component at a time. **Acc. Diff.** specifies the difference in accuracy compared to the full system and **Unseen Acc.** is the accuracy on devices unseen during training. All accuracies are reported in terms of the mean and standard deviation of 3 independent runs.

6. CONCLUSION

In this work, we propose a system that advances the state of the art in low-complexity Acoustic Scene Classification with two main contributions: Firstly, we improve Knowledge Distillation by forming teacher ensembles consisting of CNNs and Transformers trained with Freq-MixStyle and Device Impulse Response augmentation. Secondly, we introduce an efficient CNN architecture, CP-Mobile, with residual inverted bottleneck blocks and Global Response Normalization. CP-Mobile can be scaled down to a size of 5.7 kB while still beating the DCASE baseline system by almost 10 % in accuracy. Finally, we assess the importance of our system's components in a detailed ablation study and confirm the high impact of CP-Mobile and Knowledge Distillation. The proposed system outperforms the top-ranked approach for the DCASE 22 challenge by more than 5% in terms of accuracy on the TAU22 development set.

7. ACKNOWLEDGMENT

The LIT AI Lab is supported by the Federal State of Upper Austria. GW's work is supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme, grant agreement No 101019375 (Whither Music?).

8. REFERENCES

[1] I. Martín-Morató, F. Paissan, A. Ancilotto, T. Heittola, A. Mesaros, E. Farella, A. Brutti, and T. Virtanen, "Low-complexity acoustic scene classification in dcase 2022 challenge," in *DCASE Workshop*, 2022.

[2] T. Heittola, A. Mesaros, and T. Virtanen, "Acoustic scene classification in dcase 2020 challenge: Generalization across devices and low complexity solutions," in *DCASE Workshop*, 2020.

[3] I. Martin, T. Heittola, A. Mesaros, and T. Virtanen, "Low-complexity acoustic scene classification for multi-device audio: Analysis of dcase 2021 challenge systems," in *DCASE Workshop*, 2021.

[4] K. Koutini, H. Eghbal-zadeh, and G. Widmer, "Receptive field regularization techniques for audio classification and tagging with deep convolutional neural networks," *IEEE ACM Trans. Audio Speech Lang. Process.*, 2021.

[5] K. Koutini, H. Eghbal-zadeh, M. Dorfer, and G. Widmer, "The receptive field as a regularizer in deep convolutional neural networks for acoustic scene classification," in *EUSIPCO*. IEEE, 2019.

[6] B. Kim, S. Yang, J. Kim, and S. Chang, "QTI Submission to DCASE 2021: Residual Normalization for Device-Imbalanced Acoustic Scene Classification with Efficient Design," DCASE2021 Challenge, Tech. Rep., 2021.

[7] J.-H. Lee, J.-H. Choi, P. M. Byun, and J.-H. Chang, "Hyu submission for the DCASE 2022: Efficient fine-tuning method using device-aware data-random-drop for device-imbalanced acoustic scene classification," DCASE2022 Challenge, Tech. Rep., 2022.

[8] K. Koutini, F. Henkel, H. Eghbal-zadeh, and G. Widmer, "CP-JKU Submissions to DCASE'20: Low-Complexity Cross-Device Acoustic Scene Classification with RF-Regularized CNNs," DCASE2020 Challenge, Tech. Rep., 2020.

[9] K. Koutini, S. Jan, and G. Widmer, "CPJKU Submission to DCASE21: Cross-Device Audio Scene Classification with Wide Sparse Frequency-Damped CNNs," DCASE2021 Challenge, Tech. Rep., 2021.

[10] F. Schmid, S. Masoudian, K. Koutini, and G. Widmer, "CP-JKU submission to dcase22: Distilling knowledge for low-complexity convolutional neural networks from a patchout audio transformer," DCASE2022 Challenge, Tech. Rep., 2022.

[11] K. Koutini, J. Schlüter, H. Eghbal-zadeh, and G. Widmer, "Efficient training of audio transformers with patchout," in *Interspeech*. ISCA, 2022.

[12] M. Sandler, A. G. Howard, M. Zhu, A. Zhmoginov, and L. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *CVPR*. IEEE, 2018.

[13] A. Howard, R. Pang, H. Adam, Q. V. Le, M. Sandler, B. Chen, W. Wang, L. Chen, M. Tan, G. Chu, V. Vasudevan, and Y. Zhu, "Searching for mobilenetv3," in *ICCV*. IEEE, 2019.

[14] M. Tan and Q. V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *ICML*. PMLR, 2019.

[15] Y. Gong, Y. Chung, and J. R. Glass, "PSLA: improving audio tagging with pretraining, sampling, labeling, and aggregation," *IEEE ACM Trans. Audio Speech Lang. Process.*, 2021.

[16] F. Schmid, K. Koutini, and G. Widmer, "Efficient large-scale audio tagging via transformer-to-cnn knowledge distillation," in *ICASSP*. IEEE, 2023.

[17] J. Frankle, G. K. Dziugaite, D. Roy, and M. Carbin, "Pruning neural networks at initialization: Why are we missing the mark?" in *ICLR*, 2021.

[18] C. Liu, Z. Zhang, and D. Wang, "Pruning deep neural networks by optimal brain damage," in *INTERSPEECH*. ISCA, 2014.

[19] I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, and Y. Bengio, "Quantized neural networks: Training neural networks with low precision weights and activations," *J. Mach. Learn. Res.*, 2017.

[20] B. Jacob, S. Kligys, B. Chen, M. Zhu, M. Tang, A. G. Howard, H. Adam, and D. Kalenichenko, "Quantization and training of neural networks for efficient integer-arithmetic-only inference," in *CVPR*. IEEE, 2018.

[21] G. E. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *CoRR*, 2015.

[22] J. Ba and R. Caruana, "Do deep nets really need to be deep?" in *NeurIPS*, 2014.

[23] C.-H. H. Yang, H. Hu, S. M. Siniscalchi, Q. Wang, W. Yuyang, X. Xia, Y. Zhao, Y. Wu, Y. Wang, J. Du, and C.-H. Lee, "A lottery ticket hypothesis framework for low-complexity device-robust neural acoustic scene classification," DCASE2021 Challenge, Tech. Rep., 2021.

[24] P. Primus, H. Eghbal-zadeh, D. Eitelsebner, K. Koutini, A. Arzt, and G. Widmer, "Exploiting parallel audio recordings to enforce device invariance in cnn-based acoustic scene classification," in *DCASE Workshop*, 2019.

[25] B. Kim, S. Yang, J. Kim, H. Park, J. Lee, and S. Chang, "Domain generalization with relaxed instance frequency-wise normalization for multi-device acoustic scene classification," in *Interspeech*. ISCA, 2022.

[26] F. Schmid, S. Masoudian, K. Koutini, and G. Widmer, "Knowledge distillation from transformers for low-complexity acoustic scene classification," in *DCASE Workshop*, 2022.

[27] T. Morocutti, F. Schmid, K. Koutini, and G. Widmer, "Device-robust acoustic scene classification via impulse response augmentation," in *Submitted to EUSIPCO*, 2023.

[28] J. F. Gemmeke, D. P. W. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio set: An ontology and human-labeled dataset for audio events," in *ICASSP*. IEEE, 2017.

[29] F. Schmid, T. Morocutti, S. Masoudian, K. Koutini, and G. Widmer, "CP-JKU submission to dcase23: Efficient acoustic scene classification with cp-mobile," DCASE2023 Challenge, Tech. Rep., 2023.

[30] S. Chang, H. Park, J. Cho, H. Park, S. Yun, and K. Hwang, "Subspectral normalization for neural audio data processing," in *ICASSP*. IEEE, 2021.

[31] S. Woo, S. Debnath, R. Hu, X. Chen, Z. Liu, I. S. Kweon, and S. Xie, "Convnext V2: co-designing and scaling convnets with masked autoencoders," *CoRR*, 2023.

[32] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR*, 2015.

DEVICE GENERALIZATION WITH INVERSE CONTRASTIVE LOSS AND IMPULSE RESPONSE AUGMENTATION

Lorenz P. Schmidt, Nils Peters

Friedrich-Alexander-Universität Erlangen-Nürnberg, Erlangen, Germany
International Audio Laboratories, Erlangen, Germany
{lopa.schmidt, nils.peters}@fau.de

ABSTRACT

Acoustic Scene Classification poses a significant challenge in the DCASE Task 1 TAU22 dataset with a sample length of only a single second. The best performing model in the 2023 challenge achieves an accuracy of 62.7% with a gap to unseen devices of approximately 10%. In this study, we propose a novel approach using Inverse Contrastive Loss to ensure a device class invariant latent representation and a better generalization to unseen devices. We evaluate the interaction of this contrastive learning approach with impulse response augmentation and show the effectiveness for suppressing device related information in the encoder structure. Results indicates that both, contrastive learning and impulse response augmentation, improves generalization to unseen devices. Further the impulse response dataset should have a balanced frequency response to work effectively. Combining contrastive learning and impulse response augmentation yields embeddings with least device related information, but does not improve scene classification accuracy when compared to augmentation alone.

Index Terms— acoustic scene classification, contrastive learning, device impulse response, augmentation, passt, transformer

1. INTRODUCTION

Acoustic Scene Detection plays a vital role in various applications, such as hearing aids [1], smart homes [2], hands-free telephony and biological signal analysis [3]. The objective is to classify an acoustic scene into several, pre-defined, classes, enabling the application of algorithms under varying conditions. For example, the noise suppression and beamforming in hearing aids uses different approaches for a closed room and open-space [1]. With significant progress in recent years, especially with the introduction of data-based models, the DCASE Challenge [4] Task 1 for Acoustic Scene Classification (ASC) attracts a great number of contributions. Recently the focus shifted to resource-aware methods with complexity constraints [4].

One difficulty of low-complexity inference is generalization to unseen devices. The TAU Urban Acoustic Scenes 2022 Mobile dataset [5] poses a considerable challenge with an inference length of only a single second. Furthermore the data is heavily imbalanced towards one recording device. The discrepancy between recording devices arises due to variations in microphone characteristics, frequency responses and other device-specific (possible non-linear) factors that influences the signal captured. The TAU22 dataset contains audio recorded with three real devices (A: Soundman OKM II Classic/Studio A3, B: Samsung Galaxy S7, C: GoPro Hero5 Session) and three simulated devices (S1-S3). Further three unseen devices (S4-S6) are artificially generated for the testing dataset, emphasizing the importance of generalization to unseen devices. The dataset is heavily biased towards device A with 62.5% of all samples, while the remaining 8 devices contain only 37.5% of data.

We evaluate the interaction of contrastive learning and device impulse response augmentation. For our challenge submission [6] we used contrastive learning to improve device generalization. We show that combining both suppresses device related information in the model embedding better, than just using each method individually. We estimate the influence on classification performance for a state-of-the-art Transformer model with the TAU22 dataset.

By applying Inverse Contrastive Learning (ICL) [7] to the problem we encourage the model to learn device invariant representation. We use two device impulse response (DIR) datasets for augmentation. The first contains recordings of 66 vintage microphone impulse responses [8]. The second dataset is generated from 25 professional microphones recorded at different angles and distances [9], amounting in a total of 8138 DIRs.

We first discuss in Section 2 related work when dealing with heavily imbalanced datasets. This includes resampling methods, invariance learning and data augmentation. In our method part, Section 3 and Section 4, we discuss how contrastive learning helps to improve generalization ability and the differences between both device impulse response datasets. In the final Section 5 we explain our experiments, the outcomes and discuss implications for training.

2. RELATED WORK

2.1. Data Imbalance Resampling

The problem of device generalization is part of the broader issue of imbalanced training. A common countermeasure is oversampling of under-representated groups [10] to duplicate minority class samples. On the other hand, undersampling of over-represented removes considered samples during training epochs [11]. These techniques create therefore a more balanced training set and prevent the model to be biased for the majority class. The DCASE ASC dataset is characterised by a stark bias towards a single device (more than 60% of total data, where 16% would be uniformly distributed). Undersampling would prolong training time for the model until the whole dataset is seen at least once.

Another possibility (for avoiding duplicating excessively when oversampling) is synthetic minority oversampling, where new samples are generated by interpolating between existing minority classes [12]. Freq-MixStyle [13] is an instance of this approach by

---

The International Audio Laboratories Erlangen are a joint institution of the Friedrich-Alexander-Universität Erlangen-Nürnberg and Fraunhofer IIS.

mixing frequency statistics of spectrograms, which shows a good generalization performance. Also Adaptive Synthetic Sampling [14] weights likelihood of samples by difficulty and generates minority classes that are harder to learn more often.

2.2. Invariance Learning

The concept of learning invariant representations is closely related to data imbalance in the sense that both address biases in datasets. While data imbalance focuses on unequal distributions in samples, learning invariant representations aims to extract features that are robust to variations introduced by factors in the data. In both cases the aim is reducing the impact of biases and promote generalization.

One example of invariance learning is the Generative Adversarial Network (GAN). An adversarial discriminator infers the device class during training and promotes learning invariant embeddings. The generator and discriminator are trained in tandem, where the generator creates realistic looking audio samples [15].

Other invariance learning methods extend contrastive learning to self-supervised settings. As reported in [16], an online model tries to predict the representation of a target model with an augmented view. This makes representation invariant in view differences and can be applied without any labels of data.

2.3. Data Augmentation

Data augmentation helps to manage imbalanced datasets by increasing the diversity and quantity of samples, therefore improving generalization ability of the model and reducing risk of overfitting. Common examples are SpecAugment [17] introducing random freq/temporal masking and warping, pitch shifting [18], time stretching and noise injection [19] into the data samples.

A simple, yet effective, method for generalizing to new devices is impulse response augmentation. In our case we convolve our training data with measured or simulated device impulse response to create a more diverse and realistic dataset. For ASC the generated data makes the training more robust and resembles a more realistic inference environment. To model non-linear effects, dynamic range compression [20] can simulate the microphone characteristics.

3. CONTRASTIVE LEARNING

The goal of contrastive learning is to find a latent representation where positive pairs are grouped together, while negative pairs are separated. Originally introduced for supervised learning [21], it recently finds extensions to unsupervised and self-supervised settings [22, 23] and application to audio [24]. In our dataset, we have device classes available making supervised methods possible.

In our approach, positive samples are selected from different device classes, and should exhibit a greater similarity on average compared to negative samples. Negative samples are from the same device class and the training process should maximize their dissimilarity. This method is used in ICL to find more mode-collapse robust latent representations, compared to approaches using Kullback-Leibler divergence or Maximum-Mean Discrepancy [7]. ICL utilizes a loss function defined as follows

$$\mathcal{L}_{\text{ICL}} = \mathop{\mathbb{E}}_{\substack{(z,c)\sim p(z,c)\\(\hat{z},\hat{c})\sim p(\hat{z},\hat{c})}} [\mathbb{1}(c=\hat{c})f(z,\hat{z}) + \mathbb{1}(c\neq\hat{c})g(z,\hat{z})] \tag{1}$$

$$\mathcal{L} = \mathcal{L}_{\text{CE}} + \lambda_{\text{ICL}}\mathcal{L}_{\text{ICL}} \tag{2}$$

with asymmetric penalties for positive and negative samples

$$g(z,\hat{z}) = d_Z^2(z,\hat{z}) \tag{3}$$

$$f(z,\hat{z}) = \exp((\alpha - d_Z(z,\hat{z}))/\beta), \tag{4}$$

where the threshold α and barrier strength β defines the extent to which latent similarity for the same device classes are penalized. The linear combination with the default cross-entropy loss term $\mathcal{L}_{\text{CE}}$ is controlled by λ_{ICL}. The distance $d_Z(\cdot,\cdot)$ is the ℓ_2 norm for all our experiments. We discuss the selection of barrier parameters α, β in our experiments described in Sec. 5.2.

The objective of ICL is to make training invariant to unwanted variables in the dataset. It can be used to address biases and confounding effects related to demographical variables [25, 26], for example age, gender, income etc. This helps mitigate biases and ensures correct model inference without unwanted side-effects.

In our case, we employ a Transformer model [27] as the encoder structure to project a spectrogram into a lower-dimensional embedding. The encoder is expected to learn meaningful and robust representations that can be utilized for the downstream task of acoustic scene classification. Data augmentation plays a crucial role in training a good generalizing encoder. Furthermore, it can also improve the impact of contrastive learning in two ways, as shown in Sec. 5.4. First, augmentation leads to more device classes, which gives the contrastive learning more positive and negative samples for training. Second, the augmented device classes share some device characteristics with neighbouring classes. This makes the negative sampling more difficult, forcing the model to use a variety of device specific traits in the data. In our case the device of an acoustic scene sample is altered with an additional DIR.

4. DEVICE IMPULSE RESPONSES

In this section, we provide a description of two different datasets of microphone impulse responses that are used for augmenting the ASC TAU22 training set. Their characteristics are quite different.

The first dataset contains recordings of 66 vintage microphones produced by the MicIR project [8]. They are recorded in a booth with the swept-sine method. The source is placed in approximately 20-30cm distance from the microphone. Due to different room reflections, the recordings should not be considered as free-field. As seen in Fig. 1 the vintage DIRs have a frequency dependent variability and pronounced low-pass behaviour for frequencies above 10 kHz. Between the 1 kHz and 10 kHz region the data follows a narrow band in 0.1 to 0.9 quantiles with 20% of data in a wider 20dB variation.

The second dataset contains DIRs of 25 microphones for multiple angles and distances, and is henceforth called Multi DIRs. Incident angles are varied from 0° to 355° in steps of 5° and at source-to-microphone distances of 0.5m, 1.25m and 5m. The microphone is rotated with a computer-assisted turntable. The microphone characteristics include omnidirectional, cardioid, supercardioid and bidirectional polar patterns. The set is quite varied, due to different microphone transduction types (condenser, moving-coil, ribbon), single/dual and small/large diaphragms, and end/side address designs.

The distribution of frequency responses (see Fig. 1 for Multi DIRs) shows a more frequency independent variability of responses when compared to the Vintage DIRs. Further a smaller dip for frequencies above 10 kHz distorts the training data distribution less

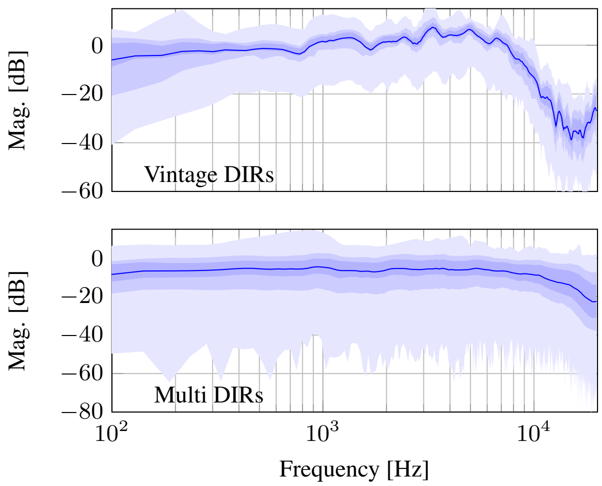


Figure 1: Microphone Frequency Responses for 100% (▨), 80% (▨), 50% (▨) of all data and mean (—) values.

and avoids an inference mismatch. Finally the 0.1-0.9 quantile is wider and exposes more variations to the model during training.

When looking at a specific example (the Røde NT2-A cardioid microphone) over variations in incident angle, a rich pattern can be seen in Figure 2. It follows the characteristics of a cardioid microphone polar pattern, with a deep notch at 180°. This shows a large variability for even a single microphone characteristic.

5. EXPERIMENTS

We evaluate the tandem setting of contrastive learning and DIR augmentation with the TAU22 [5] dataset split to 139,970 samples for training, 29,680 samples for validation and 29,680 samples for testing. They are recorded at a sampling rate of 44.1 kHz in 12 different European cities and 10 acoustic scenes. We first describe how we set-up our model for all our training sessions. Then the specifics of ICL and DIR augmentation are explained and their effects on device invariance and scene classification discussed.

5.1. PaSST Model

We extract Mel-scaled spectrograms with 128 bands from audio subsampled to 32 kHz sampling rate. Individual windows have a length of 800 samples and an overlap of 320 samples. We apply a logarithmic transformation to normalize the feature distribution.

We use the Patchout faSt Spectrogram Transformer (PaSST) [27, 28] as our encoder structure and a single feed-forward layer for classification. The transformer has a patch size of 16, depth of 12 and 12 heads. Furthermore, the embedding dimension is 768, where the classifier projects the final embedding to 10 scene classes. To speed up training, we apply patchout along frequency axis with a rate of 6 patches similar to the PaSST model [27].

To avoid overfitting and improve generalization, the dataset is augmented in the following ways. We merge recordings to 10s snippets and extract randomly sliding windows of 1s during training. We also apply independent frequency masking for 48 bins and do time masking for 24 windows and use randomized frequency cutoff of up to 500Hz. With this we follow the training approach described in PaSST [27], but we do not use mixstyle augmentation to compare augmentation and contrastive learning properly. We apply an Adam optimizer with the same settings as in our challenge submission [6]. An initial learning rate of 0.00042 is gradually reduced on plateau with a patience of 10 epochs and factor of 0.5. We operate the optimizer at $\beta = (0.957, 0.9514)$ and $\epsilon = 0.038$. Each configuration is trained three times for 250 epochs with a batch-size of 64 and we use the best performance in our results.

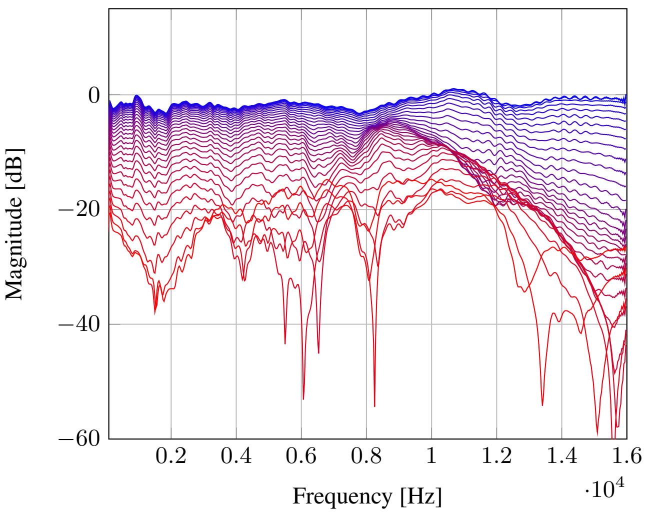


Figure 2: Frequency Responses of the Røde NT2-A cardioid microphone for angles 0° (—) to 180° (—) recorded at a distance of 50cm.

5.2. Inverse Contrastive Loss

We apply a inverse contrastive loss during training to make classification invariant to device characteristics. The augmented term penalizes latent distances of same device classes. This implies a tradeoff when choosing the hyperparameters for training. The exponential term (see Equation 4) acts as a barrier function for a shifted threshold α, where the strength is controlled by β (with indicator function in limit $\beta \to 0$). We choose $\beta = 1$ for all our experiments and perform grid search for suitable hyper-parameters which results are shown in Table 1. Even though the variables are not independent, we grid-searched them separately. We first fix $\lambda_{\text{ICL}} = 10$ to observe the effect of thresholding on the performance. It exhibits a slight decrease in performance when increasing to $\alpha = 0.2$, while a more drastical degradation when increasing further. Therefore we conclude with this value for the remaining of our experiments. The loss weight λ_{ICL} does not have such a drastic effect on the performance, but increasing too much decreases performance by 2% accuracy. We choose $\lambda_{\text{ICL}} = 0.5$ as a conservative measure. The accuracy improved on the validation dataset compared to $\lambda_{\text{ICL}} = 0.1$, indicating a positive effect of ICL.

5.3. Impulse Response Augmentation

For DIR augmentation, we use two dataset sources [8, 9]. We resample both to 32kHz sampling rate. Further, we window the Multi DIRs dataset [9] to 1024 samples with a Kaiser window ($\beta = 2$).

With this, we train an IR generator, similar to the FAST-RIR [29] diffuse room impulse generator. The model is conditioned on

α	LogLoss	Acc. [%]	λ_{ICL}	LogLoss	Acc. [%]
0.0	1.265	51.72	0.1	1.139	56.87
0.2	**1.270**	51.28	0.5	**1.100**	58.32
0.4	1.313	50.34	1.0	1.113	58.52
0.8	1.331	49.02	3.0	1.139	56.87
1.5	2.166	21.60	6.0	1.183	56.30

Table 1: Results for different α and λ_{ICL} values. We fixed $\lambda_{\text{ICL}} = 10.0$ for the α search and $\alpha = 0.2$ for the λ_{ICL} search. Based on the results we choose $\alpha = 0.2$, $\lambda_{\text{ICL}} = 0.5$ for further experiments.

Method	Accuracy [%]
PaSST	82.04
+ ICL	65.65
+ Multi DIRs	41.08
+ ICL + Multi DIRs	**17.53**

Table 2: Device classification accuracy results for the embedding of a PaSST model with different generalization methods (see Sec 5.4). Lower accuracy indicates better invariance to device class.

the microphone characteristics (1) directivity (2) transducer (3) diaphragm properties and angle/distance in cartesian coordinates in total of 12 variables. We train the generator in the same GAN framework as the original method with a final MSE of 0.00527. Unfortunately the approximately 8000 samples of Multi DIR are not sufficient for training a microphone impulse response model. We see good generalization for varying incident angles, but not for source distance and new synthetic device classes. Applying the generator to our ASC model gives only a best log-loss of 1.56 and we drop it therefore for our next comparison.

5.4. Device Related Latent Information

As an additional study we measure the device related information during training. We create a separate device classifier with the same capacity as the acoustic scene classifier and train it with the default Adam optimizer until convergence. Since the device class is imbalanced, we use a balanced cross-entropy term as our loss measure.

The results in Table 2 are evaluated on the validation set for the 6 devices of the training set. Because the device occurrence is balanced, random guess is set at $1/6$.

The use of contrastive learning does not lower device accuracy as much as impulse response augmentation. A possible explanation for this is that we can use augmentation aggressively, while use of contrastive learning has a negative effect on training (see Table 1). Further augmentation adds variability to the dataset and does not necessarily inhibit the primary task.

Interestingly, combining augmentation and contrastive learning reduces device accuracy further to the points of random guess. This indicates, that the latent space for acoustic scene classification does not have device-related information. When looking at the final results in Table 3, on the other hand, the results are still biased towards the more common devices. To illustrate, see that the acoustic scene classifier benefits from a robust latent representation. Even though we have minimized device related information the encoder still generalizes the spectrogram for the majority class better. To mitigate this effect we would have to resample to even class distribution, for example with synthetic augmentation (see related work in Sec. 2). Another possibility is that the device classify is too shallow to model the benefiting factors for the scene classification, even though they have the same capacity.

5.5. Acoustic Scene Classification Results

As the final experiment we train the PaSST model with the illustrated four different settings for device generalization. We see a large gap of 0.3 log-loss between real devices and simulated/unseen devices in Table 3 for the vanilla PaSST model - with device A best performing of 1.012 log-loss.

The vintage DIR augmentation improves the performance for unseen devices, but degrades that of real devices. This gives a worse overall performance. The multi angle DIR dataset on the other hand improves performance for all three device families, with the largest improvement in unseen devices of approximately 0.1 log-loss. When applying contrastive learning we see a similar effect, but not as pronounced as the impulse response augmentation. Further the performance for real devices suffer slightly.

Finally combining impulse response augmentation with contrastive learning improves performance slightly, compared to contrastive learning alone. On the other hand, it does not improve performance when comparing to Multi DIRs augmentation alone.

6. CONCLUSION

To summarize, contrastive learning makes latent space invariant to device classes and improves generalization. To that effect, impulse response augmentation works better, but best device invariance is achieved by combining both methods. The Multi DIRs shows a greater variability and less bias for frequency responses and works better for data augmentation when compared to the Vintage DIR. In the final ASC experiment, contrastive learning improves log-loss, but is outperformed by applying proper data augmentation alone. Nevertheless, contrastive learning can be advantageous compared to domain specific augmentation, especially when the training is only affected by data imbalance and not by unseen classes or no effective augmentation technique is available.

Method	Real Devices				Simulated Devices				Unseen Devices				Overall
	A	B	C	Avg.	S1	S2	S3	Avg.	S4	S5	S6	Avg.	
PaSST	1.012	1.266	1.070	1.116	1.371	1.492	1.326	1.396	1.401	1.36	1.509	1.423	1.181
+ Vintage DIRs	1.082	1.360	1.212	1.218	1.462	1.449	1.361	1.424	1.343	1.289	1.557	1.396	1.212
+ Multi DIRs	**0.979**	1.221	1.090	1.097	**1.347**	1.427	**1.318**	**1.364**	**1.277**	**1.302**	**1.425**	**1.334**	**1.139**
+ ICL	1.021	1.297	**1.074**	1.131	1.375	1.465	1.325	1.388	1.284	1.364	1.467	1.372	1.167
+ ICL + Multi DIRs	1.030	**1.190**	1.139	**1.096**	1.372	**1.412**	**1.318**	1.367	1.302	1.327	1.474	1.368	1.156

Table 3: Log-loss validation performance of the proposed methods on the TAU Urban Acoustic Scenes 2022 Mobile dataset [5] with provided split. The PaSST model is trained for three different seeds and best performance is picked. The validation results are grouped into real devices (A, B, C), simulated devices (S1, S2, S3) and unseen devices (S4, S5, S6) and averaged values given to compare device families.

7. REFERENCES

[1] V. Vivek, S. Vidhya, and P. Madhanmohan, “Acoustic scene classification in hearing aid using deep learning,” in *2020 International Conference on Communication and Signal Processing (ICCSP)*, 2020, pp. 0695–0699.

[2] S. Krstulović, “Audio event recognition in the smart home,” *Computational Analysis of Sound Scenes and Events*, 2018.

[3] A. I. Humayun, S. Ghaffarzadegan, M. I. Ansari, Z. Feng, and T. Hasan, “Towards domain invariant heart sound abnormality detection using learnable filterbanks,” *IEEE Journal of Biomedical and Health Informatics*, vol. 24, no. 8, pp. 2189–2198, aug 2020. [Online]. Available: https://doi.org/10.1109%2Fjbhi.2020.2970252

[4] I. Martín-Morató *et al.*, “Low-complexity acoustic scene classification in dcase 2022 challenge,” 2022. [Online]. Available: https://arxiv.org/abs/2206.03835

[5] T. Heittola, A. Mesaros, and T. Virtanen, “Acoustic scene classification in dcase 2020 challenge: generalization across devices and low complexity solutions,” in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2020 Workshop (DCASE2020)*, 2020, pp. 56–60. [Online]. Available: https://arxiv.org/abs/2005.14623

[6] L. Schmidt, B. Kiliç, and N. Peters, “Submission to DCASE 2023 task 1: Device invariant training with structured filter pruning for low complexity acoustic scene classification,” DCASE2023 Challenge, Tech. Rep., May 2023.

[7] A. K. Akash, V. S. Lokhande, S. N. Ravi, and V. Singh, “Learning invariant representations using inverse contrastive loss,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 8, 2021, pp. 6582–6591.

[8] Xaudia, “Microphone impulse response project,” accessed 2023-06-14. [Online]. Available: http://micirp.blogspot.com/

[9] J. C. Franco Hernández, B. Bacila, T. Brookes, and E. De Sena, “A multi-angle, multi-distance dataset of microphone impulse responses,” *J. Audio Eng. Soc*, vol. 70, no. 10, pp. 882–893, 2022. [Online]. Available: http://www.aes.org/e-lib/browse.cfm?elib=22014

[10] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, “SMOTE: Synthetic minority over-sampling technique,” *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, jun 2002. [Online]. Available: https://doi.org/10.1613%2Fjair.953

[11] G. E. A. P. A. Batista, R. C. Prati, and M. C. Monard, “A study of the behavior of several methods for balancing machine learning training data,” *SIGKDD Explor. Newsl.*, vol. 6, no. 1, p. 20–29, jun 2004. [Online]. Available: https://doi.org/10.1145/1007730.1007735

[12] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, “Smote: synthetic minority over-sampling technique,” *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[13] B. Kim, S. Yang, J. Kim, H. Park, J. Lee, and S. Chang, “Domain generalization with relaxed instance frequency-wise normalization for multi-device acoustic scene classification,” *arXiv preprint arXiv:2206.12513*, 2022.

[14] H. He, Y. Bai, E. A. Garcia, and S. Li, “Adasyn: Adaptive synthetic sampling approach for imbalanced learning,” in *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, 2008, pp. 1322–1328.

[15] Y. Cui, M. Jia, T.-Y. Lin, Y. Song, and S. Belongie, “Class-balanced loss based on effective number of samples,” 2019.

[16] J.-B. Grill *et al.*, “Bootstrap your own latent: A new approach to self-supervised learning,” 2020.

[17] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, “SpecAugment: A simple data augmentation method for automatic speech recognition,” in *Interspeech 2019*. ISCA, sep 2019. [Online]. Available: https://doi.org/10.21437%2Finterspeech.2019-2680

[18] H. Su, H. Zhang, X. Zhang, and G. Gao, “Convolutional neural network for robust pitch determination,” in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 579–583.

[19] S. Pascual, A. Bonafonte, and J. Serra, “Segan: Speech enhancement generative adversarial network,” *arXiv preprint arXiv:1703.09452*, 2017.

[20] H. Hu *et al.*, “Device-robust acoustic scene classification based on two-stage categorization and data augmentation,” DCASE2020 Challenge, Tech. Rep., June 2020.

[21] S. Chopra, R. Hadsell, and Y. LeCun, “Learning a similarity metric discriminatively, with application to face verification,” in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05)*, vol. 1, 2005, pp. 539–546 vol. 1.

[22] T. Gao, X. Yao, and D. Chen, “Simcse: Simple contrastive learning of sentence embeddings,” 2022.

[23] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan, “Supervised contrastive learning,” 2021.

[24] A. Saeed, D. Grangier, and N. Zeghidour, “Contrastive learning of general-purpose audio representations,” in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 3875–3879.

[25] V. S. Lokhande, R. Chakraborty, S. N. Ravi, and V. Singh, “Equivariance allows handling multiple nuisance variables when analyzing pooled neuroimaging datasets,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 10 432–10 441.

[26] Z. Cao, H. Yu, H. Yang, and A. Sano, “Pirl: Participant-invariant representation learning for healthcare,” 2022.

[27] K. Koutini, J. Schlüter, H. Eghbal-zadeh, and G. Widmer, “Efficient training of audio transformers with patchout,” in *Interspeech 2022*. ISCA, sep 2022. [Online]. Available: https://doi.org/10.21437%2Finterspeech.2022-227

[28] A. Dosovitskiy *et al.*, “An image is worth 16x16 words: Transformers for image recognition at scale,” *arXiv preprint arXiv:2010.11929*, 2020.

[29] A. Ratnarajah, S.-X. Zhang, M. Yu, Z. Tang, D. Manocha, and D. Yu, “Fast-rir: Fast neural diffuse room impulse response generator,” 2022.

MULTI-LABEL OPEN-SET AUDIO CLASSIFICATION

Sripathi Sridhar[*], *Mark Cartwright*

Sound Interaction and Computing (SInC) Lab, New Jersey Institute of Technology
{ss645, mark.cartwright}@njit.edu

ABSTRACT

Current audio classification models have small class vocabularies relative to the large number of sound event classes of interest in the real world. Thus, they provide a limited view of the world that may miss important yet unexpected or unknown sound events. To address this issue, open-set audio classification techniques have been developed to detect sound events from unknown classes. Although these methods have been applied to a multi-class context in audio, such as sound scene classification, they have yet to be investigated for polyphonic audio in which sound events overlap, requiring the use of multi-label models. In this study, we establish the problem of multi-label open-set audio classification by creating a dataset with varying unknown class distributions and evaluating baseline approaches built upon existing techniques.

Index Terms— Open-set, multi-label, audio classification, dataset

1. INTRODUCTION

Audio classification (AC), the machine listening task of identifying sound events in an audio recording, has typically been studied as two task variants, i.e. multi-class AC, where the input recordings are expected to contain only one event, and multi-label AC, where the input recordings may contain multiple overlapping sound events. Real-world audio recordings in typical urban, domestic or environmental settings often contain multiple sound sources of anthrophony, biophony, and geophony, and thus, are better modeled as a multi-label AC task.

Multi-label AC is a common machine listening task that has been applied to various scenarios such as urban sound data [1], everyday environments [2], and music [3]. Much of this work however assumes a small fixed class vocabulary, a closed-set task, which does not reflect real-world scenarios. Everyday sound scenes consist of sources drawn from hundreds if not thousands of classes depending on the class granularity of interest, and people are constantly exposed to novel classes, e.g., those from new or uncommon technology and animal vocalizations. To the "ears" of these models, unknown sound classes simply do not exist or — possibly worse — are confused with known classes. This limited class vocabulary size can be attributed to the cost and difficulty of annotating large-scale audio datasets. However, the result of this barrier is a limited view of the acoustic world by AC models that may miss important yet unexpected or unknown sound events, hindering machine listening's transformative potential.

One solution to this problem is to build models with a dynamic vocabulary that can be updated in a lightweight manner without having to retrain the model from scratch. An example of this approach is few-shot classification [4], which is often formulated within a meta-learning framework where a model can learn a new class from a small 'support set' of examples [5]. Prior work has applied this to tasks such as instrument recognition [6], multi-label audio classification [7], and multi-label drum transcription [8]. However, this method still requires the user or researcher to supply a support set for unseen or novel classes [7], and thus, such supervised approaches are only useful if you know what you are hoping to find and have examples of it. In many situations — e.g., urban noise monitoring, audio accessibility, bioacoustic monitoring — it is the rare events and unexpected events that are arguably the most important to detect, i.e., the machine listening equivalent of a "black swan event" [9]. To this end, we focus on detecting the presence of unknown classes in addition to known classes, referred to as open-set modeling.

Open-set modeling has seen research interest in the image domain for several years [10, 11], but it has only more recently gained interest in the audio domain and been applied to tasks such as domestic sound classification [12], acoustic scene classification [13, 14], and the related yet distinct task of anomalous sound detection [15]. However, all of these tasks are binary or multi-class AC — to the best of our knowledge, open-set modeling has not been applied to multi-label AC.

As in [12, 10], we define *known known* (KK) classes as known (i.e., in-vocabularly) classes seen during training and inference, *known unknown* (KU) classes as unknown (i.e., out-of-vocabulary) classes seen during training and inference, and *unknown unknown* (UU) as unknown classes seen only at inference. A fourth category, unknown known (UK) classes, are classes in which only semantic or metadata information is available in the absence of discrete labels — this category is not considered in this work. We collectively refer to KU and UU as unknown classes, and KK as known classes.

We define multi-label open-set AC (MLOS) as the task of assigning between 0 and $|\mathrm{KK}| + 1$ class labels to an audio recording, where $|\mathrm{KK}|$ is the cardinality of the set of known classes and $+1$ refers to the label indicating the presence of an unknown sound class. Thus, an MLOS model needs to both estimate which known classes are present as well as decide whether at least one unknown class is present. This is in contrast to multi-class open-set AC models which assign only 1 of $|\mathrm{KK}| + 1$ class labels to an audio recording.

In this paper, we (1) establish the problem of MLOS, (2) introduce a new dataset with varying unknown class distributions to investigate this problem, and (3) evaluate baseline approaches comprised of combinations of existing machine listening techniques.

2. DATASET

Prior open-set AC datasets are either multi-class [12] or focused on binary anomalous sound detection [16]. In order to establish the

---

[*] This work is partially supported by NSF award #1955357

MLOS task, we are interested in exploring the effects of polyphony, and levels of "openness" while working with a large class vocabulary. While few-shot datasets like FSD-MIX-CLIPS [7] meet the polyphony criteria, they do not have varying levels of "openness" nor dataset variants where different classes are assigned to the KK, KU and UU categories. As in [17], we define "openness" as

$$O^* = 1 - \sqrt{(2 \times C_{tr}) / (C_{tr} + C_{te})}, \quad (1)$$

where $C_{tr} = |\text{KK} \cup \text{KU}|$ is the number of classes seen during training and $C_{te} = |\text{KK} \cup \text{KU} \cup \text{UU}|$ the number of classes seen during testing. Thus, for larger C_{tr}, we assign lower values of openness.

To this end, we develop a new dataset of synthetic soundscapes using open-set criteria. As in FSD-MIX-CLIPS, we use a subset of FSD50K where each clip has a single 'present and predominant' label, i.e., the labeled sound event is the only type of sound present with the exception of mild background noise [18]. This gives us 7600 source events from 89 classes, each between 0.5s and 4s in duration. We use only the leaf node labels according to the Audioset ontology [19]. Hereafter we refer to this subset of FSD50K as the *source dataset*.

First, we split the classes into 5 subsets of 18 classes each (except for one subset with 17 classes), and from these subsets, we create 10 variations of class assignments into KK, KU, and UU as shown in Table1 — 5 with a low degree of openness and 5 with a high degree of openness, i.e. no KU classes. The openness coefficients are $O^* = 0.05$ or 0.06 for low openness ($C_{tr} = 72$ or 71) and $O^* = 0.13$ or 0.14 for high openness ($C_{tr} = 54$ or 53). For each class assignment variation i, we generate an intermediate dataset called 'Open-Set Soundscape-i' (OSS-i), consisting of 10s 44.1kHz synthetic soundscapes using Scaper [20] — 200k training, 30k validation, and 30k test with no source overlap between splits. The training and validation sets are synthesized from only the known class subsets, e.g. in dataset variant 1, from L1-L4 in the low openness case and H1-H3 in the high openness case (Var. 1 in Table 1). In both openness cases, the test set is synthesized using all the subsets. Additionally, we also create a small tuning validation set using all the subsets for hyperparameter tuning, ensuring no example overlap with the test set.

In each OSS-i, we maintain the class distribution of the source dataset as closely as possible while enforcing a minimum of 200 examples per class. Each soundscape has one to four overlapping source sound events in the foreground, which we place between 0 to 9s in the soundscape. We augment each source with pitch shifting (-2 to +2 semitones) and time stretching (by a factor of 0.8 to 1.2). We use uniform random sampling for all augmentations during generation.

For each OSS-i dataset variant, we generate a dataset of 1s clips by centering a window on each event in the 10s soundscape and labeling a class as present if it overlaps with this window. This yields 10 datasets (5 high, 5 low openness) with ∼500k clips each.

We refer to this as the Open-Set Tagging (OST) dataset and use it to train and evaluate our models. Both OSS and OST datasets are publicly available [1].

Openness	Low					High				
Subset	L1	L2	L3	L4	L5	H1	H2	H3	H4	H5
Var. 1	KK	KK	KK	KU	UU	KK	KK	KK	UU	UU
Var. 2	UU	KK	KK	KK	KU	UU	KK	KK	KK	UU
Var. 3	KU	UU	KK	KK	KK	UU	UU	KK	KK	KK
Var. 4	KK	KU	UU	KK	KK	KK	UU	UU	KK	KK
Var. 5	KK	KK	KU	UU	KK	KK	KK	UU	UU	KK

Table 1: Class splits for high and low openness dataset variations

3. MODELS

In this study, for the sake of brevity we focus on the high openness MLOS task, as it is the more challenging scenario. Therefore in the following we use D_k to denote the set of known classes seen during training, and D_u for the set of unknown classes seen only during inference.

In this section, we present five baseline models, two of which use oracle sources as a way of further exploring the limitations of these approaches.

3.1. Multi-label

Given a multi-label input example x, the classifier C generates a logit vector $\mathbf{v} = C(x) \in \mathbb{R}^N$, where $N := |D_k|$ i.e. KK classes present during training. To estimate whether the input contains a class in D_k, we take the indices above a threshold λ, i.e. $\{j : v_j > \lambda; j \in [0, N-1]\}$.

Our baseline approach to the MLOS task is to run inference using a standard multi-label classifier. Then, to predict the unknown class we use the open-set decision criteria discussed later in this section.

The classifier consists of two stages. The first stage is a frozen OpenL3 encoder pre-trained on the environmental subset of Audioset [21], which has shown competitive performance across a variety of audio and music classification tasks in the NeurIPS HEAR 2021 challenge [22]. The encoder input is a 256 frequency bin log-melspectrogram input, with output embeddings of dimension 6144.

The second stage is a multi-layer perceptron (MLP) with five dense layers. Each layer consists of 1024 units and ReLU activation. The number of output units depends on the number of classes in the dataset variant, i.e. $|D_k|$ classes. This system is depicted in Figure 1.

The multi-label classifier output has sigmoid activations and is trained using binary cross-entropy loss. Instead of using a threshold in our experiments, we used an overly-optimistic oracle strategy, picking the m sources with the highest logits, where m is the polyphony from the ground-truth data. We use the checkpoint with the best validation loss for evaluation.

3.2. Combinatorial multi-class

In order to isolate the effect of multi-label training, we include a 'combinatorial multi-class' model. Here we map each unique label combination in the OST training set to a class ID, effectively creating a multi-class model training setup. While OST has around 8000 unique class combinations, we note that this approach would lead to a 'combinatorial explosion' and may be infeasible as the number of classes and unique combinations increase.

Apart from a categorical cross-entropy loss function and different number of output layer units, we use the same architecture and training setup as described in Section 3.1.

3.3. Source estimates multi-class PIT

Since prior work on open-set AC has been in the multi-class setup, we include a model with a universal source separation front-end

---

[1] 10.5281/zenodo.7241704

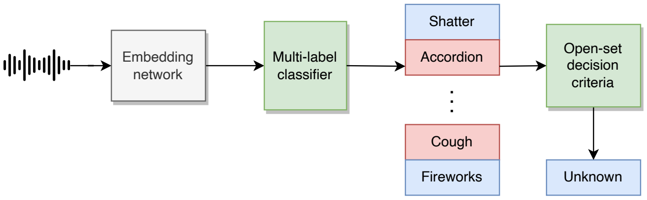


Figure 1: Multi-label model consisting of a pre-trained frozen OpenL3 embedding network and a MLP classifier.

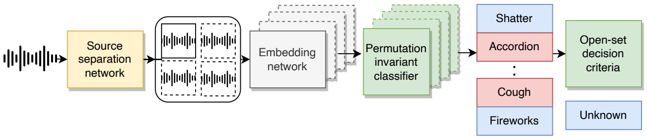

Source separation network
Embedding network
Permutation invariant classifier
Shatter
Accordion
:
Cough
Fireworks
Open-set decision criteria
Unknown

Figure 2: Source estimates multi-class PIT model consisting of a separation network and multi-class classifier trained using a permutation invariant loss. The separation network is trained separately using MixIT, then its weights are frozen as the classifier is trained.

module to convert the MLOS task to a set of multi-class open-set classification tasks and leverage existing approaches for these subtasks. Related prior work successfully used such a universal source separation pre-processing step to improve classification precision in multi-label closed-set birdsong classification [23]. A separation model generates source estimates for a multi-class classifier that generates predictions. We hypothesize that the separation model will also improve performance on the MLOS task, particularly on unseen known class combinations (which could be misclassified by an open-set model as *unknown* if estimated as a whole) and for clips with high polyphony. However, this approach does come with the risk of error propagation from the separation model to the classifier caused by poor source estimates.

Given an input example x, the separation model S generates eight source estimates s_i. Using m of these eight source estimates as input, the multi-class classifier C generates logits $\mathbf{v}_i = C(s_i) \in \mathbb{R}^N$ for each source s_i, where again $N := |D_k|$, and we use the class with the max logit as our known class prediction for that source, i.e. $\text{argmax}(\mathbf{v}_i)$.

The separation network is a TDCN++ model trained on unlabeled polyphonic mixtures using mixture invariant training (MixIT) [24]. As the authors of [23, 24] note the importance of training MixIT on the target domain for quality source estimates, we train from scratch on data from all variants of the OST dataset for 1M steps and use the checkpoint with the best validation performance. Estimating the number of actual sources from the 8 fixed outputs is a challenging task and a potential failure point. In this paper, we opt for an overly optimistic scenario and use an oracle pruning strategy for testing. We pick the m source estimates with the highest energy, where m is the number of ground-truth sources. We follow this protocol both during training and inference. An existing risk of this approach is that the chosen source estimate may only contain background for input examples with low SNR. Additionally, this protocol may be sub-optimal if the model over-separates, especially in examples with low polyphony.

The multi-class classifier has the same architecture as in Section 3.1 , and is trained using a permutation invariant cross-entropy loss [24]. Since the label assignment is only available at the clip level, we generate a prediction for each source estimate and compute the total loss for $m!$ label-source combinations. The best match that minimizes the total loss is used to update the model weights. We use the suffix permutation invariant training (PIT) to denote that a model is trained this way. The model is depicted in Figure 2.

3.4. Oracle sources multi-class PIT

In order to understand the effects of error propagation due to the separation network, we train a model with a perfect separation model, i.e. with the oracle sources. These oracle sources when re-combined yield the OST clips used to train the multi-label classifier model. We use the same model and training setup as in Section 3.3.

3.5. Oracle sources multi-class model

A key limitation of PIT is that it does not guarantee accurate source-label matching during training. In order to further isolate the effect that PIT may have on performance, we evaluate a reference multi-class model with the same architecture trained with oracle sources using standard cross-entropy loss. Given our modeling choices, this serves as an expected upper bound in terms of performance, as it is a true multi-class model.

3.6. Open-set decision criteria

We evaluate two simple open-set decision criteria that have been used previously in multi-class open-set studies. Here, we use these techniques both in the multi-class and multi-label configurations, however, the latter would suffer from false positives in scenarios with no activity or background noise events.

The first approach is softmax thresholding, where the maximum softmax probability (MSP) is compared against a threshold δ [25]– where a model predicts *unknown* if it is below and *known* otherwise. Let $\hat{\mathbf{y}}$ be the classifier output for models without separation, e.g. $\hat{\mathbf{y}} = \text{sigmoid}(\mathbf{v})$, and $\hat{y}_o \in \{0, 1\}$ the open-set prediction, with 0 and 1 denoting a known and unknown class prediction respectively, then

$$\hat{y}_o = \Big\{ 1 \text{ if } \max(\hat{\mathbf{y}}) < \delta;\ \text{else } 0 \tag{2}$$

For PIT models and the oracle sources multi-class model, we predict unknown if any of the *m* source estimates contain an unknown class:

$$\hat{y}_o = \Big\{ 1 \text{ if } \max(\hat{\mathbf{y}}_i) < \delta,\ \text{for } i \in [0, m-1];\ \text{else } 0 \tag{3}$$

where $\hat{\mathbf{y}}_i$ is the classifier output for a source estimate.

The second approach is Openmax [26], which aims to correct 'overconfident' model predictions when the example is less likely to belong to the training distribution of the predicted class. Openmax re-weights the logit vector by penalizing the top α ranked logits using models of the training distribution tail for each class. The class-specific models are parameterized by the Weibull distribution tail size τ and logit rank limit α. It also computes an unknown class probability p_u based on the degree of recalibration needed, which is then appended to the updated classifier output. We refer the reader to [26] for further details.

For models without separation, we compute the updated classifier output $\hat{\mathbf{y}}_w$ using the re-weighted logit vector $\mathbf{v}_w$, e.g. $\hat{\mathbf{y}}_w = \text{sigmoid}(\mathbf{v}_w)$. Then, similar to Equation 2–

$$\hat{y}_o = \Big\{ 1 \text{ if } \max(\hat{\mathbf{y}}_w) < \delta \text{ or} \max(\hat{\mathbf{y}}_w) = p_u;\ \text{else } 0 \tag{4}$$

	Accuracy (SD)	
	MSP	Openmax
Multi-label	57.4 (2.9)	–
Source estimates PIT	54.3 (3.0)	–
Oracle sources PIT	**59.7** (4.7)	**61.3** (3.0)
Combinatorial multi-class	59.1 (1.5)	–
Oracle sources multi-class	**61.1** (3.8)	**61.2** (3.8)

Table 2: Unknown detection results using maximum softmax probability thresholding (MSP) and openmax. All results are accuracy averaged over the five dataset variants, with standard deviations in parentheses.

For models with separation we apply this re-weighting and thresholding protocol to $\mathbf{v}_i$, the source estimate logit vectors.

We tune δ, τ, and α on the tuning validation set using Optuna, a Python package for efficient hyperparameter optimization [27], and use hyperparameters from the trial that maximizes unknown detection accuracy.

4. EVALUATION

We evaluate the models separately on closed-set classification and unknown detection. For the former, we evaluate the model only on examples without unknown classes. For the latter, we evaluate the models on all examples at the clip level for a binary classification task. We present the unknown detection results in Table 2 and closed set classification results in Table 3.

From Table 2, we note that the multi-label model is worse than the oracle sources multi-class model. In this dataset, every example has at least one source, however, in scenarios where no event may be present we expect this gap to be larger, as the multi-label model may generate more false positives during silence or background noise events.

Combinatorial multi-class is only slightly worse than oracle sources multi-class. While this is an interesting finding, there are two key limitations. This model does not scale well as the number of classes increases, leading to the 'combinatorial explosion' issue [28]. Furthermore, this dataset follows the imbalanced source dataset distribution making certain known classes more likely than others, meaning that the model does not encounter new class combinations in the test set, leading to an optimistic view of its unknown detection accuracy. We expect this model to perform poorly in scenarios with unseen combinations of known classes, potentially generating false positives.

Oracle sources PIT does better than the multi-label model by about 4%, which suggests that a perfect universal source separator could improve performance on this task. However, the gap is smaller than expected, potentially due to false positives caused by overconfident model predictions [26]. We see some evidence of this in Table 2 where Openmax accuracy for the oracle sources PIT model is better than its MSP accuracy, suggesting that this model is falsely overconfident for examples containing unknown class events.

We also note that oracle sources multi-class is better than oracle sources PIT by about 2%– since they are both trained on the same data, the difference must be due to PIT.

Finally, source estimates PIT is not as good as the oracle sources PIT model, and in fact, performs worse than the multi-label model. This indicates that more research may be needed for universal source separation models to be useful in this task. Some prior results suggest that training the classifier together on the input mixture and source estimates may improve closed-set classification [23], but it remains to be seen whether this translates to unknown detection where the model needs to separate out unknown class events as well.

	Micro F1	**Macro F1**	**mAP**
Multi-label	0.449 (0.01)	0.349 (0.02)	0.400 (0.02)
Source Estimates PIT	0.407 (0.01)	0.332 (0.01)	0.347 (0.01)
Oracle Sources PIT	**0.511** (0.02)	**0.461** (0.04)	**0.501** (0.04)
Oracle sources multi-class	**0.581** (0.01)	**0.541** (0.01)	**0.590** (0.01)

Table 3: Closed-set classification results on 53 or 54 classes, depending on the dataset variant. All metrics are averaged over the five dataset variants, with standard deviations in parentheses.

We notice similar trends in closed-set classification (Table 3) as in unknown detection MSP accuracy. The multi-label model as well as the oracle sources PIT model perform significantly worse than the oracle sources multi-class model, which is in line with the expectation of multi-label classification being a more challenging task. Oracle sources PIT does better than the multi-label model, which suggests that a perfect source separation model would be useful. Lastly, the overall modest performance of the oracle sources multi-class model on both closed- and open-set tasks suggests that better audio representations are also needed to improve performance.

5. DISCUSSION AND CONCLUSION

In this work, we introduced the multi-label open-set audio classification (MLOS) task and developed a synthetic dataset with varying unknown class distributions. We then presented several baseline models using combinations of existing machine listening techniques and evaluated their performance on known class and unknown class metrics.

We show that MLOS is a challenging task that existing approaches alone cannot adequately solve. In our study, we find that a perfect source separation model may be useful for MLOS, but further research is needed for universal source separation models to provide similar improvements in open-set classification.

While we see some interesting results, some other questions were raised, such as how unseen known class combinations might affect unknown class detection, particularly for the multi-label and combinatorial multi-class models. We plan to evaluate this by varying vocabulary and dataset size to control the ratio of seen and unseen known class combinations in the test set.

Moreover, we consider here a simplistic data scenario where there is always at least one sound present. We plan to investigate how the inclusion of background event classes would affect some of the models discussed here, such as the multi-label and source estimates multi-class PIT.

By sharing the dataset and these baseline results, we hope to invite further interest from the community to this under-explored area of research.

6. ACKNOWLEDGMENT

We would like to thank Simran Kaur for helping to generate the initial version of the dataset.

7. REFERENCES

[1] M. Cartwright, A. E. M. Mendez, J. Cramer, V. Lostanlen, G. Dove, H.-H. Wu, J. Salamon, O. Nov, and J. P. Bello, "SONYC Urban Sound Tagging (SONYC-UST): A Multilabel dataset from an urban acoustic sensor network," *Proc. of the Detection and Classification of Acoustic Scenes and Events 2019 Workshop (DCASE2019).*

[2] E. Cakir, T. Heittola, H. Huttunen, and T. Virtanen, "Polyphonic sound event detection using multi label deep neural networks," in *2015 International Joint Conference on Neural Networks (IJCNN)*, July 2015, pp. 1–7, iSSN: 2161-4407.

[3] S. Gururani, C. Summers, and A. Lerch, "Instrument activity detection in polyphonic music using deep neural networks." in *ISMIR*, 2018, pp. 569–576.

[4] M. Fink, "Object classification from a single example utilizing class relevance metrics," *Advances in neural information processing systems*, vol. 17, 2004.

[5] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," *Advances in neural information processing systems*, vol. 30, 2017.

[6] H. Flores Garcia, A. Aguilar, E. Manilow, and B. Pardo, "Leveraging hierarchical structures for few-shot musical instrument recognition," in *Proc. of the 22nd Int. Society for Music Information Retrieval Conference*, 2021.

[7] Y. Wang, N. J. Bryan, M. Cartwright, J. Pablo Bello, and J. Salamon, "Few-Shot Continual Learning for Audio Classification," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).*

[8] Y. Wang, J. Salamon, M. Cartwright, N. J. Bryan, and J. P. Bello, "Few-Shot Drum Transcription in Polyphonic Music," *International Society for Music Information Retrieval Conference*, 2020.

[9] N. N. Taleb, *The black swan: The impact of the highly improbable*. Random house, 2007, vol. 2.

[10] A. Bendale and T. Boult, "Towards Open World Recognition," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA.

[11] H. Zhang, A. Li, J. Guo, and Y. Guo, "Hybrid models for open set recognition," in *European Conference on Computer Vision*. Springer, 2020, pp. 102–117.

[12] J. Naranjo-Alcazar, S. Perez-Castanos, P. Zuccarello, F. Antonacci, and M. Cobos, "Open Set Audio Classification Using Autoencoders Trained on Few Data," *Sensors*, vol. 20, no. 13, p. 3741, July 2020.

[13] Z. Kwiatkowska, B. Kalinowski, M. Kośmider, and K. Rykaczewski, "Deep learning based open set acoustic scene classification," *Proc. Interspeech 2020.*

[14] A. Mesaros, T. Heittola, and T. Virtanen, "Acoustic scene classification in dcase 2019 challenge: Closed and open set classification and data mismatch setups," *Proc. of the Detection and Classification of Acoustic Scenes and Events 2019 Workshop (DCASE)*, 2019.

[15] Y. Kawaguchi, K. Imoto, Y. Koizumi, N. Harada, D. Niizumi, K. Dohi, R. Tanabe, H. Purohit, and T. Endo, "Description and discussion on DCASE 2021 challenge task 2: Unsupervised anomalous sound detection for machine condition monitoring under domain shifted conditions," *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2021 Workshop (DCASE)*, 2021.

[16] Y. Koizumi, S. Saito, H. Uematsu, Y. Kawachi, and N. Harada, "Unsupervised Detection of Anomalous Sound Based on Deep Learning and the Neyman–Pearson Lemma," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*.

[17] C. Geng, S.-J. Huang, and S. Chen, "Recent Advances in Open Set Recognition: A Survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2020.

[18] E. Fonseca, X. Favory, J. Pons, F. Font, and X. Serra, "FSD50K: an open dataset of human-labeled sound events," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 829–852, 2022.

[19] J. F. Gemmeke, D. P. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio set: An ontology and human-labeled dataset for audio events," in *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2017, pp. 776–780.

[20] J. Salamon, D. MacConnell, M. Cartwright, P. Li, and J. P. Bello, "Scaper: A library for soundscape synthesis and augmentation," in *2017 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA).*

[21] J. Cramer, H.-H. Wu, J. Salamon, and J. P. Bello, "Look, Listen, and Learn More: Design Choices for Deep Audio Embeddings," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).*

[22] J. Turian, J. Shier, H. R. Khan, B. Raj, B. W. Schuller, C. J. Steinmetz, C. Malloy, G. Tzanetakis, G. Velarde, K. McNally, *et al.*, "Hear: Holistic evaluation of audio representations," in *NeurIPS 2021 Competitions and Demonstrations Track*. PMLR, 2022, pp. 125–145.

[23] T. Denton, S. Wisdom, and J. R. Hershey, "Improving bird classification with unsupervised sound separation," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* IEEE.

[24] S. Wisdom, E. Tzinis, H. Erdogan, R. Weiss, K. Wilson, and J. Hershey, "Unsupervised sound separation using mixture invariant training," *Advances in Neural Information Processing Systems*, vol. 33, pp. 3846–3857, 2020.

[25] B. Dubuisson and M. Masson, "A statistical decision rule with incomplete knowledge about classes," *Pattern recognition*, vol. 26, no. 1, pp. 155–165, 1993.

[26] A. Bendale and T. E. Boult, "Towards Open Set Deep Networks," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* Las Vegas, NV, USA: IEEE, June 2016, pp. 1563–1572.

[27] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019.

[28] H. Phan, T. N. T. Nguyen, P. Koch, and A. Mertins, "Polyphonic audio event detection: multi-label or multi-class multi-task classification problem?" in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* IEEE, 2022, pp. 8877–8881.

SPECTRAL TRANSCODER : USING PRETRAINED URBAN SOUND CLASSIFIERS ON UNDERSAMPLED SPECTRAL REPRESENTATIONS

Modan Tailleur[1], *Mathieu Lagrange*[1], *Pierre Aumond*[2], *Vincent Tourre*[3]

[1] Nantes Université, École Centrale Nantes,
CNRS, LS2N, UMR 6004, F-44000 Nantes, France, {modan.tailleur,mathieu.lagrange}@ls2n.fr
[2] Univ Gustave Eiffel, CEREMA, UMRAE, F-44344 Bouguenais, France, pierre.aumond@univ-eiffel.fr
[3] Nantes Université, École Centrale Nantes, CNRS, AAU, UMR 1563, F-44000 Nantes, France,
vincent.tourre@ec-nantes.fr

ABSTRACT

Slow or fast third-octave bands representations (with a frame resp. every 1-s and 125-ms) have been a de facto standard for urban acoustics, used for example in long-term monitoring applications. It has the advantages of requiring few storage capabilities and of preserving privacy. As most audio classification algorithms take Mel spectral representations with very fast time weighting (ex. 10-ms) as input, very few studies have tackled classification tasks using other kinds of spectral representations of audio such as slow or fast third-octave spectra.

In this paper, we present a convolutional neural network architecture for transcoding fast third-octave spectrograms into Mel spectrograms, so that it could be used as input for robust pre-trained models such as YAMNet or PANN models. Compared to training a model that would take fast third-octave spectrograms as input, this approach is more effective and requires less training effort. Even if a fast third-octave spectrogram is less precise both on time and frequency dimensions, experiments show that the proposed method still allows for classification accuracy of 62.4% on UrbanSound8k and 0.44 macro AUPRC on SONYC-UST.

Index Terms— Convolutional Neural Network (CNN), Generative algorithm, third-octave spectrogram, Mel spectrogram, Urban soundscape

1. INTRODUCTION

In recent years, various sound source classification models have gained recognition for their robustness. Among them, YAMNet [1] and PANNs [2] pre-trained models have emerged as powerful models capable of predicting the presence of more than 500 sound sources, thanks to their training on the extensive Audioset database [3]. These models are widely recognized as among the most effective sound source classification models available and use Mel spectral representations with a frame every 10-ms as input .

IEC 61672-1 [4] standardizes the measurement of fast (125-ms) and slow (1-s) third-octave spectral representations, which have been used in several noise monitoring applications [5, 6, 7, 8, 9, 10]. Fast third-octave spectrograms offer several advantages over Mel spectrograms for long-term monitoring applications. First, they make recordings unintelligible and thus preserve privacy, as demonstrated by Gontier et al. [11]. Moreover, they are more lightweight, with a bit rate approximately 138 times lower than that of 16bits, 32kHz, mono waveform recordings and about 30 times lower than that of Mel recordings (see table 1 for precise references).

Gontier et al. [12] addressed multi-label classification tasks in urban environments using a Convolutional Neural Network (CNN) directly trained on third-octave spectrograms. While their model showed good performance on the Cense Lorient dataset [8], it lacks robustness on other third-octave recorded datasets. This limitation arises partly from training the model on highly homogeneous datasets. Pre-trained models such as YAMNet and PANNs, on the other hand, have shown robustness in a variety of sound source classification tasks. Unfortunately, these models are trained on Mel spectrograms with 10-ms frames, and can only consider the corresponding Mel representation as input.

To enable the direct use of those pre-trained models with other types of spectrograms such as fast third-octave ones, we present in this paper a transcoding method that converts fast third-octave spectrograms into Mel spectrograms. This transcoding operation is done using a CNN module learned with a teacher-student approach that leverages the pre-trained models' outputs to reconstruct Mel spectrograms. While this study focuses on a specific fast third-octave representation, we believe that the proposed method can be adapted to any kind of spectral representation. Section 2 reviews prior work on the transcoding task. Sections 3 and 4 outline our model architecture and training method. In section 5, we evaluate the performances of the transcoder. Generated audio and open source code are available online. [1]

2. RELATED WORK

To the best of our knowledge, no work is available for the task at hand in audio processing specifically. In computer vision, several methods have been proposed to address the task of converting one set of features to another set of features (feature translation) [13, 14]. A pseudo-inverse can be employed to retrieve a Mel spectrogram from a fast third-octave spectrogram and temporal information can be interpolated. This would result in a blurred Mel spectrogram, which could be seen as analogous to a noisy image in a denoising paradigm. Auto-encoding methods [15], adversarial methods [16], and diffusion methods [17] have been used in super-resolution and denoising tasks.

In contrast to previous works, our goal is to obtain generated Mel spectrograms that can achieve similar output class distributions as the original Mel spectrogram when processed by the pre-trained model used for training the transcoder.

---

[1] Companion website: `https://github.com/modantailleur/paperSpectralTranscoder`

3. METHODS

3.1. Spectral representations

In this study, we selected the Lorient Cense project fast third-octave calculation method [8] which involves computing 29 third-octave bands within the frequency range of 20Hz to 12,5kHz, using a rectangular 125-ms temporal window.

It is worth noting that YAMNet and PANNs models require Mel spectrograms as input, but the spectrograms used by these classifiers differ slightly from each other (as shown in Table 1). Therefore, we present two different transcoders in the subsequent sections to match the input requirements of each pre-trained model.

spectral representation	Mel		Third-Octave
origin	PANN	YAMNet	Cense Lorient
sample rate	32kHz	16kHz	32kHz
window size	1024 (32ms)	400 (25ms)	4096 (128ms)
fft size	1024 (32ms)	512 (32ms)	4096 (128ms)
hop size	320 (10ms)	160 (10ms)	4000 (125ms)
window	hann	hann	rectangular
frequency bins	64	64	29
min frequency	50Hz	125Hz	20Hz
max frequency	14kHz	7,5kHz	12,5kHz
mel normalisation	slaney	-	-
mel formula	slaney	htk	-
log offset	1.0	0.001	19.95
bit rate	103kb/s	100kb/s	3,71kb/s

Table 1: Differences between PANN (ResNet38) and YAMNet Mel spectral spectrograms, and Cense third-octave spectrograms

3.2. Model

The proposed CNN transcoder model, consists of two parts: a PINV transcoder and a Convolutional Neural Network (CNN) (see Figure 2). The PINV transcoder presented in figure 1 first reconstructs the full-band spectrogram from the third-octave spectrogram using a pseudo-inverse method. Then, it performs time-axis interpolation to match the time dimension of the target Mel spectrogram. Finally, the log Mel filterbank is applied to the full-band spectrogram, resulting in a roughly predicted Mel spectrogram. This PINV transcoder conveniently matches the target Mel spectrogram dimensions, and is adaptable to various undersampled spectral data.

The CNN part then refines the Mel spectrogram by adding residual information to it (see figure 2). The CNN architecture, which is identical to the one used by Lagrange et al. [18], is fully convolutional and has several layers, each employing rectified linear units (ReLU) activations. In the following sections, we refer to our transcoder, which is trained on pre-trained models' output logits, as CNN trained on logits (or CNN-logits).

3.3. Teacher-student approach

We take a teacher-student approach to train our CNN model in order to generate a Mel spectrogram by taking into account the output of YAMNet or PANNs pre-trained classifiers (see figure 3). We selected the ResNet38 PANN model, which has 73,783,247 parameters, as it is the most performing model to date that uses Mel spectrograms as input [2]. We also consider the well-established YAMNet classifier which has a lower number of parameters: 3,740,425. PANNs and YamNet parameters are not updated during the CNN transcoder training, reducing computational complexity and ensuring broader applicability to pre-trained classifiers using similar Mel spectrogram inputs.

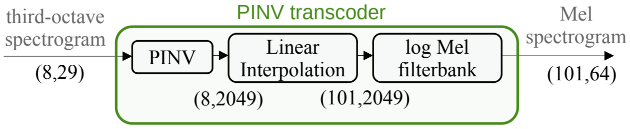


Figure 1: PINV transcoder architecture, to recover a 1s sample PANN Mel spectrogram from a 1s sample fast third-octave spectrogram

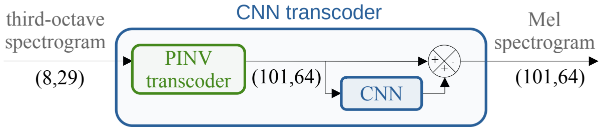


Figure 2: CNN transcoder architecture, to recover a 1s sample PANN Mel spectrogram from a 1s sample fast third-octave spectrogram

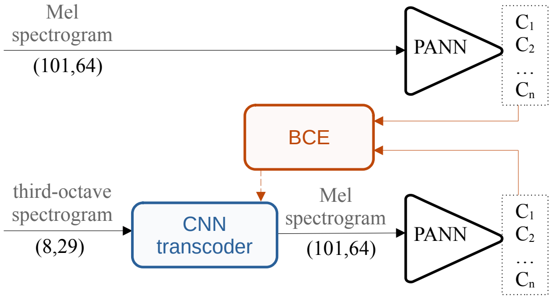


Figure 3: PANN CNN transcoder trained with a teacher-student approach (CNN-logits) using the Binary Cross-Entropy (BCE) loss function

4. EXPERIMENTAL PROTOCOL

4.1. Data

The dataset used for training and evaluating our models is the TAU Urban Acoustic Scenes 2020 Mobile dataset [19]. This dataset consists of 10-second audio clips from 10 different acoustic scenes, namely airport, indoor shopping mall, metro station, pedestrian street, public square, street with a medium level of traffic, traveling by tram, traveling by bus, traveling by an underground metro, and urban park. The dataset includes recordings from multiple devices that overlap in the given development subset. As the evaluation dataset has not been released yet, we use only the development subset for training and evaluating our models. To ensure non-overlapping data, we use only data from device A, which provides 29h20 of audio. We randomly split the development subset into training (75%), validation (12.5%), and evaluation (12.5%) sets. All audio files are normalized based on the maximum absolute value.

4.2. Baselines

In this study, we compare the performance of the CNN-logits transcoder with the performance of a reference PINV transcoder (as shown in Figure 1), which does not require any learning.

In addition, we explore an alternative training method that is solely based on the Mean Squared Error (MSE) loss between the generated Mel spectrogram and the ground truth spectrogram, without relying on a teacher-student approach. This transcoder will be referred to as CNN trained on mels (or CNN-mels) in the subsequent sections.

To further evaluate the performance of our proposed teacher-student approach, we compare it with other teacher-student methods that are not explicitly designed for transcoding fast third-octave spectrograms into Mel spectrograms (see figure 4). Specifically, we retrain the PANN and YAMNet models, as well as efficient nets (efficient net b_0 with 4,682,059 parameters and efficient net b_7 with 65,135,455 parameters) [20], using pseudo-inverted Mel-spectrograms as input with the method illustrated in Figure 1. In the subsequent sections, we will refer to these retrained models as PANN-1/3-oct, YAMNet-1/3-oct, Effnet-b0, and Effnet-b7.

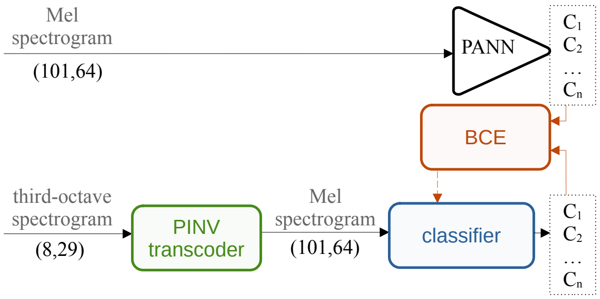


Figure 4: Classifier trained with a teacher-student approach, to match PANN outputs

4.3. Training procedure

Both types of training methods, i.e., with and without teacher-student approach, employ Adam optimizer [21] during optimization.

For the CNN architecture, we have conducted experiments with varying kernel sizes, numbers of layers, dilations, numbers of channels, and learning rates. Only the models with parameters leading to the best loss are presented in the subsequent sections.

All the models are trained for 200 epochs, with a batch size of 64, leading to 2,472,000 iterations. We checked empirically that convergence is reached for all models.

4.4. Metrics

To assess the performance of the proposed methods, we introduce the Prediction to Prediction accuracy on First Class (PtoPa-FC) metric, and calculate it on our evaluation subset of the TAU Urban Acoustic Scenes 2020 Mobile dataset. This metric measures the accuracy of the pre-trained model that uses transcoded Mel spectrograms as input in predicting the same first class as that of the pre-trained models that use ground truth Mel spectrograms as input. However, it should be noted that this metric only provides information regarding the accuracy of the first predicted class. Therefore, we also analyze the KL-divergence between the distribution of the two predictions vectors. All predictions are based on 10-second audio excerpts.

To further evaluate the effectiveness of our models, we subject them to testing on two additional annotated datasets: SONYC-UST [22] and UrbanSound8K [23]. However, the output classes of the pre-trained PANN and YAMNet models do not correspond exactly to the target classes of these datasets. To address this issue, we propose to augment the pre-trained models with two additional fully connected layers that have an intermediate size of 100. These layers are trained on the training subset of SONYC-UST and evaluated on the test subset, and we employ cross-validation for UrbanSound8K as recommended by the authors. The objective of the fully connected layers is to aggregate the 527 (or 521) input classes of the pre-trained models into the 8 (or 10) target classes of UrbanSound8K or SONYC-UST datasets, respectively. Importantly, we only train the additional fully connected layers, and the pre-trained models are not re-trained during this process. We apply a threshold of 0.5 for the multi-label task of SONYC-UST, and we consider the class with the highest output value as present for UrbanSound8K multi-class classification task. We found that our proposed method outperforms a manual aggregation method similar to the one proposed by [24], which gave poorer results on both datasets using our models.

5. RESULTS

Table 2 summarizes the performance of the methods on the TAU Urban Acoustic Scenes 2020 Mobile dataset. The parameter tuning procedure mentioned in section 4.3 identify a CNN model with a kernel size of 5, no dilation, 64 channels, and 5 layers, trained with a learning rate of 10^{-3}. This model contains 192,961 parameters, which represents 0.26% of PANN's and 5.2% of YamNet's total number of parameters. Our CNN-logits model outperforms the baseline models for PANN, achieving a PtoPa-FC of 89.3% and a lower KL-divergence than the baselines. When YAMNet is used as the target classifier, our CNN-logits model achieves a higher PtoPa-FC than the other models. Notably, the KL-divergence of our model is higher than that of the YAMNet-1/3-oct model. This suggests that while its predicted first class is closer to that of YAMNet, the overall distribution of predictions across all classes is further away from the ones of the pre-trained model.

The classification results of PANN and YAMNet models on the SONYC-UST and UrbanSound8k datasets using both original and transcoded Mel spectrograms as input are shown in Table 3. The state-of-the-art macro-AUPRC for a model that is fully trained on the SONYC-UST dataset is reported between 0.49 and 0.65 [22, 25]. In contrast, the best accuracy achieved on the UrbanSound8k dataset is 90% [26]. Despite not being specifically trained on these datasets, the PANN model using ground truth Mel spectrograms as input still achieves fairly good results, albeit not outperforming state-of-the-art models. PANN models that use transcoded Mel spectrograms as input have a 18.6% decrease in accuracy compared to when a ground truth Mel spectrogram is used. This is promising, as fast third-octave spectrograms contain much less information both on frequency and time dimensions. In contrast, using the transcoder for YAMNet resulted in a much more significant drop in accuracy.

The CNN-logits method performed less effectively when used with YAMNet. YAMNet is smaller and less accurate than the ResNet38 PANN model, as evidenced by its lower performance

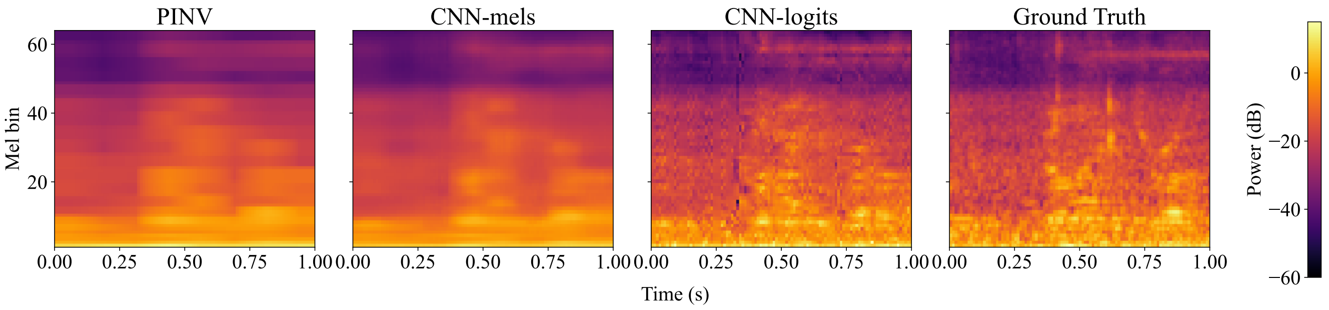


Figure 5: Mel spectrograms of a 1s file from the evaluation dataset, using different transcoding methods.

classifier	teacher-student	model	MSE (mels) ↓	KL divergence (logits) ↓	PtoPa-FC % ↑	training time
PANN	No	PINV	12.43	0.014438	0.4	-
		CNN-mels	**10.14**	0.013945	0.5	-
	Yes	CNN-logits	18.56	**0.000734**	**89.3**	14456s
		PANN 1/3 oct	-	0.000792	83.7	31702s
		Effnet-b0	-	0.008994	69.4	**11157**s
		Effnet-b7	-	0.006117	76.8	39653s
YAMNet	No	PINV	3.23	0.032255	0.8	-
		CNN-mels	**0.2**	0.013405	0.1	-
	Yes	CNN-logits	1.39	0.001863	**85.1**	5073s
		YAMNet 1/3 oct	-	**0.000919**	83.3	**4667s**
		Effnet-b0	-	0.005072	75.1	11153s
		Effnet-b7	-	0.003189	79.3	39701s

Table 2: Performance of the different models on TAU Urban Acoustic Scenes 2020 Mobile evaluation subset using pre-trained models predictions

input spectrogram	Mel		transcoded Mel	
classifier	PANN	YAMNet	PANN	YAMNet
accuracy on UrbanSound8k	**81.0** %	75.5 %	**62.4** %	42.7 %
mAUPRC on Sonyc-UST	**.52**	.48	**.44**	.34

Table 3: Performance of the different models on UrbanSound8k and Sonyc-UST. Bold values indicate the best scores achieved by classifiers using either ground truth Mel spectrograms as input (left), or Mel spectrograms transcoded from third-octave spectrograms.

on the multi-label and multi-class classification tasks in the UrbanSound8k and SONYC-UST datasets. Consequently, the coarser output logits of YAMNet compared to PANN suggest that the feature vector of size 521 produced by YAMNet may not be as relevant for spectrogram reconstruction.

The CNN-logits method produces spectrograms that are more realistic and less blurry than those obtained using the CNN-mels and PINV baselines (as shown in Figure 5). This can be attributed to the fact that by minimizing the MSE between the two spectrograms, the algorithm tends to produce results that are closer to the ground truth in terms of average pixel-to-pixel distance but leads to globally blurry results. Conversely, by training on a set of 527 (or 521) high-level features, the neural network has more degrees of freedom and is not constrained to be as close to the ground truth spectrogram. As shown in Table 2, this is reflected in the lower MSE for the CNN-mels model than for the CNN-logits model.

6. CONCLUSION

In this study, we proposed a teacher-student approach to learning a transcoder whose task is to transform any spectral representation into a Mel spectrogram, for being used as input of pre-trained classifiers such as PANN and YAMNet models. This technique demonstrates a relatively high accuracy of 62.4% and macro AUPRC of 0.44 on UrbanSound8k and SONYC-UST, respectively, despite the limitations of a third-octave spectrogram in terms of temporal and frequency resolution.

However, one limitation of this method is that a new transcoder must be trained for each Mel spectral representation, in order to adapt to its different possible parameters (number of Mel bins, hop size, sample rate, etc...). To address this limitation, future research could explore reconstructing the audio entirely from a fast or slow third-octave spectral representation, which would allow the usage of any pre-trained classifier, including the state-of-the-art PANN model Wavegram-Logmel-CNN, which utilizes information on both time-domain waveforms and log Mel spectrograms.

Very interestingly, our experiments show empirically that predicted Mels using a loss built on logits do not only allow effective prediction but also results in Mels that have far better time / frequency structure.

7. ACKNOWLEDGMENT

This work was performed using HPC resources from GENCI-IDRIS (Grant 20XX-AD011013544).

8. REFERENCES

[1] Tensorflow, "Sound classification with yamnet," 2020, last access on 09/05/2023. [Online]. Available: https://github.com/tensorflow/models/tree/master/research/audioset/yamnet/

[2] Q. Kong, Y. Cao, T. Iqbal, Y. Wang, W. Wang, and M. D. Plumbley, "Panns: Large-scale pretrained audio neural networks for audio pattern recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 2880–2894, 2020, publisher: IEEE.

[3] J. F. Gemmeke, D. P. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio set: An ontology and human-labeled dataset for audio events," in *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2017, pp. 776–780.

[4] IEC, "61672-1: 2013 Electroacoustics—Sound Level Meters—Part 1: Specifications," 2013.

[5] P. Aumond, A. Can, B. De Coensel, D. Botteldooren, C. Ribeiro, and C. Lavandier, "Modeling soundscape pleasantness using perceptual assessments and acoustic measurements along paths in urban context," *Acta Acustica united with Acustica*, vol. 103, no. 3, pp. 430–443, 2017, publisher: S. Hirzel Verlag.

[6] A. J. Torija, D. P. Ruiz, and A. F. Ramos-Ridao, "Application of a methodology for categorizing and differentiating urban soundscapes using acoustical descriptors and semantic-differential attributes," *The Journal of the Acoustical Society of America*, vol. 134, no. 1, pp. 791–802, 2013, publisher: Acoustical Society of America.

[7] M. Nilsson, D. Botteldooren, and B. De Coensel, "Acoustic indicators of soundscape quality and noise annoyance in outdoor urban areas," in *Proceedings of the 19th International Congress on Acoustics*, 2007.

[8] A. Can, J. Picaut, J. Ardouin, P. Crepeaux, E. Bocher, D. Ecotiere, and M. Lagrange, "CENSE Project: general overview," in *Euronoise 2021: European Congress on Noise Control Engineering*, 2021.

[9] F. Mietlicki, C. Mietlicki, and M. Sineau, "An Innovative Approach for long term environmental noise measurement: RUMEUR Network in the Paris Region," in *Proceedings of the EuroNoise*, 2015.

[10] J. C. Farrés, "Barcelona noise monitoring network," in *Proceedings of the EuroNoise*, 2015, pp. 218–220.

[11] F. Gontier, M. Lagrange, P. Aumond, A. Can, and C. Lavandier, "An efficient audio coding scheme for quantitative and qualitative large scale acoustic monitoring using the sensor grid approach," *Sensors*, vol. 17, no. 12, p. 2758, 2017, publisher: MDPI.

[12] F. Gontier, C. Lavandier, P. Aumond, M. Lagrange, and J.-F. Petiot, "Estimation of the perceived time of presence of sources in urban acoustic environments using deep learning techniques," *Acta Acustica united with Acustica*, vol. 105, no. 6, pp. 1053–1066, 2019, publisher: S. Hirzel Verlag.

[13] J. Hu, R. Ji, H. Liu, S. Zhang, C. Deng, and Q. Tian, "Towards Visual Feature Translation," in *Proc. IEEE Conf. Comput. Vis. Pat- tern Recognit.*, 2019, pp. 2999–3008.

[14] W. Kuang, Y.-L. Chan, S.-H. Tsang, and W.-C. Siu, "Fast HEVC to SCC transcoder by early CU partitioning termination and decision tree-based flexible mode decision for intra-frame coding," *IEEE Access*, vol. 7, pp. 8773–8788, 2019, publisher: IEEE.

[15] K. Zeng, J. Yu, R. Wang, C. Li, and D. Tao, "Coupled deep autoencoder for single image super-resolution," *IEEE transactions on cybernetics*, vol. 47, no. 1, pp. 27–37, 2015, publisher: IEEE.

[16] X. Wang, K. Yu, S. Wu, J. Gu, Y. Liu, C. Dong, Y. Qiao, and C. Change Loy, "Esrgan: Enhanced super-resolution generative adversarial networks," in *Proceedings of the European conference on computer vision (ECCV) workshops*, 2018, pp. 0–0.

[17] C. Saharia, J. Ho, W. Chan, T. Salimans, D. J. Fleet, and M. Norouzi, "Image super-resolution via iterative refinement," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022, publisher: IEEE.

[18] M. Lagrange and F. Gontier, "Bandwidth extension of musical audio signals with no side information using dilated convolutional neural networks," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 801–805.

[19] A. Mesaros, T. Heittola, and T. Virtanen, "A multi-device dataset for urban acoustic scene classification," in *Proc. Workshop Detection Classification Acoust. Scenes Events*, 2018.

[20] M. Tan and Q. Le, "Efficientnetv2: Smaller models and faster training," in *International conference on machine learning*. PMLR, 2021, pp. 10 096–10 106.

[21] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[22] M. Cartwright, A. E. M. Mendez, J. Cramer, V. Lostanlen, G. Dove, H.-H. Wu, J. Salamon, O. Nov, and J. Bello, "SONYC urban sound tagging (SONYC-UST): A multilabel dataset from an urban acoustic sensor network," in *Proc. Workshop Detection Classification Acoust. Scenes Events*, 2019, pp. 35–39.

[23] J. Salamon, C. Jacoby, and J. P. Bello, "A dataset and taxonomy for urban sound research," in *Proceedings of the 22nd ACM international conference on Multimedia*, 2014, pp. 1041–1044.

[24] F. Gontier, V. Lostanlen, M. Lagrange, N. Fortin, C. Lavandier, and J.-F. Petiot, "Polyphonic training set synthesis improves self-supervised urban sound classification," *The Journal of the Acoustical Society of America*, vol. 149, no. 6, pp. 4309–4326, 2021, publisher: Acoustical Society of America.

[25] A. Arnault and N. Riche, "CRNNs for Urban Sound Tagging with spatiotemporal context," *DCASE2020 Challenge, Tech. Rep.*, 2020.

[26] A. Gazneli, G. Zimerman, T. Ridnik, G. Sharir, and A. Noy, "End-to-end audio strikes back: Boosting augmentations towards an efficient audio classification network," *arXiv preprint arXiv:2204.11479*, 2022.

AUDIO DIFFERENCE CAPTIONING UTILIZING SIMILARITY-DISCREPANCY DISENTANGLEMENT

Daiki Takeuchi, Yasunori Ohishi, Daisuke Niizumi, Noboru Harada, and Kunio Kashino

NTT Corporation, Japan

ABSTRACT

We proposed *Audio Difference Captioning* (ADC) as a new extension task of audio captioning for describing the semantic differences between input pairs of similar but slightly different audio clips. The ADC solves the problem that conventional audio captioning sometimes generates similar captions for similar audio clips, failing to describe the difference in content. We also propose a cross-attention-concentrated transformer encoder to extract differences by comparing a pair of audio clips and a similarity-discrepancy disentanglement to emphasize the difference in the latent space. To evaluate the proposed methods, we built an AudioDiffCaps dataset consisting of pairs of similar but slightly different audio clips with human-annotated descriptions of their differences. The experiment with the AudioDiffCaps dataset showed that the proposed methods solve the ADC task effectively and improve the attention weights to extract the difference by visualizing them in the transformer encoder.

Index Terms: audio difference captioning, contrastive learning, crossmodal representation learning, deep neural network

1. INTRODUCTION

Audio captioning is used to generate the caption for an audio clip [1–10]. Unlike labels for scenes and events [11–15], captions describe the content of the audio clip in detail. However, conventional audio captioning systems often produce similar captions for similar audio clips, making it challenging to discern their differences solely based on the generated captions. For instance, suppose two audio clips of heavy rain are input into a conventional captioning system. The system will generate a caption describing the content of each, like "*It is raining very hard without any break*" and "*Rain falls at a constant and heavy rate*"[1] as illustrated in Fig. 1(a). The difference, such as which rain sound is louder, is difficult to understand from the generated captions in this case.

To address this problem, we propose *Audio Difference Captioning* (ADC) as a new extension task of audio captioning. ADC takes two audio clips as input and outputs text explaining the difference between two inputs as shown in Fig. 1. We make the ADC clearly describe the difference between the two audio clips, such as "Make the rain louder," which describes what and how to modify one audio clip to the other in the instruction form, even for audio clips with similar texts. Potential real-world applications include machine condition and healthcare monitoring using sound by captioning anomalies that differ from usual sounds.

The ADC task has two major challenges: different content detection and detection sensitivity. Since the difference between a pair of audio clips can be classes of contained events or an attribute, such as loudness, the ADC needs to detect what difference to describe. When the difference lies in an attribute, the ADC needs to be sensitive enough to detect the magnitude of the attribute, such as rain is hard or moderately shown in the example in Fig. 1.

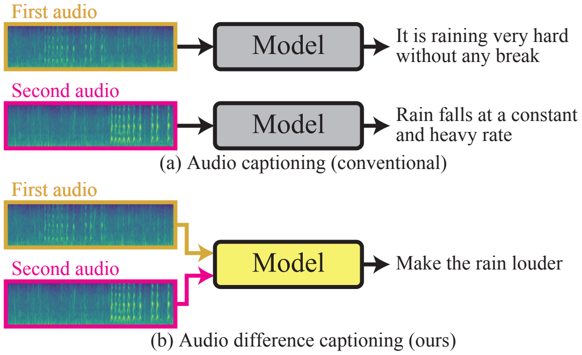

First audio
Model
It is raining very hard without any break
Second audio
Model
Rain falls at a constant and heavy rate
(a) Audio captioning (conventional)
First audio
Second audio
Model
Make the rain louder
(b) Audio difference captioning (ours)

Figure 1: Conceptual diagram of conventional audio captioning and audio difference captioning. Audio difference captioning describes the difference between pair audio clips, while conventional audio captioning describes the contents of each.

To handle these challenges, the ADC should extract features of difference based on the cross-reference of two audio clips. These features should carry enough information to differentiate critical attributes such as loudness. A typical choice of a feature extractor could be pre-trained models to classify labels [16–18]. However, these models learn to discriminate sound event classes, learning what is common while ignoring subtle differences such as raining hard or quietly unless the class definition covers that.

To meet the requirements of the ADC mentioned above, we propose (I) a cross-attention-concentrated (CAC) transformer encoder and (II) a similarity-discrepancy disentanglement (SDD). The CAC transformer encoder utilizes the masked multi-head attention layer, which only considers the cross-attention of two audio clips to extract features of difference efficiently. The SDD emphasizes the difference feature in the latent space using contrastive learning based on the assumption that two similar audio clips consist of similar and discrepant parts.

We demonstrate the effectiveness of our proposals using a newly built dataset, AudioDiffCaps, consisting of two similar but slightly different audio clips synthesized from existing environmental sound datasets [11, 15] and human-annotated difference descriptions. Experiments show that the CAC transformer encoder improves the evaluation metric scores by making the attention focus only on cross-references. The SDD also improves the scores by emphasizing the differences between audio clips in the latent space. Our contributions are proposals of (i) the ADC task, (ii) the CAC transformer encoder and SDD for solving ADC, (iii) the AudioD-

---

[1] These captions were taken from the Clotho dataset [2]

iffCaps dataset, and (iv) demonstrating the effectiveness of these proposals.

2. AUDIO DIFFERENCE CAPTIONING

We propose ADC, a task for generating texts to describe the difference between two audio clips. ADC estimates a word sequence $\boldsymbol{w}$ from the two audio clips $\boldsymbol{x}$ and $\boldsymbol{y}$.

The general framework to solve ADC includes three main functions: audio embedding, audio difference encoding, and text decoding. Audio embedding calculates two audio embedding vectors from two audio clips, respectively. Audio difference encoding captures the difference between two audio embedding vectors. Text decoding generates a description of the differences from captured differences. Audio embedding and audio difference encoding require approaches specific to ADC. In particular, difference encoding is the function unique to audio difference captioning. This function requires a model structure to capture the subtle differences between two audio clips, unlike conventional audio captioning that captures the content of a single audio clip. Moreover, the sensitivity to the subtle difference between two similar audio clips is also necessary for audio embedding. The pre-trained audio embedding models widely used for conventional environmental sound analysis tasks are often trained for classification tasks and are suitable for identifying predefined labels. Consequently, the outputs of these pre-trained audio embedding models are not sensitive to the subtle differences between audio clips with the same label. Therefore, learning to emphasize the differences between similar audio clips in the latent space is necessary when applying pre-trained audio embedding models to the ADC.

3. PROPOSED METHOD

Based on the above discussion, we propose the ADC system illustrated in Fig. 2. Our system consists of an audio feature extractor (red), difference encoder (blue), text decoder (green), and similarity-discrepancy disentanglement (purple).

3.1. Audio feature extractor

The audio feature extractor uses a pre-trained audio embedding model to calculate audio embedding vectors. Two audio clips $\boldsymbol{x}$ and $\boldsymbol{y}$ are the input, and the audio embedding vectors corresponding to the clips $\boldsymbol{X} \in \mathbb{R}^{H \times T_x}$ and $\boldsymbol{Y} \in \mathbb{R}^{H \times T_y}$ are the output, where H is the size of hidden dimension, T_x is the time length of $\boldsymbol{X}$, and T_y is the time length of $\boldsymbol{Y}$

3.2. Difference encoder

The difference encoder extracts information about the differences between the two audio clips from audio embedding vectors $\boldsymbol{X}$ and $\boldsymbol{Y}$. To extract difference information efficiently, we utilize a cross-attention-concentrated (CAC) transformer encoder as the main function of the difference encoder. The CAC transformer encoder utilizes the masked multi-head attention layer, allowing only mutual cross-attention between two audio clips by the attention mask illustrated in the upper right of Fig. 2.

The detailed procedure is as follows. First, special tokens that indicate the order of the audio clips $\mathcal{X} \in \mathbb{R}^{H \times 1}$ and $\mathcal{Y} \in \mathbb{R}^{H \times 1}$ are concatenated at the beginning of $\boldsymbol{X}$ and $\boldsymbol{Y}$, respectively. Next, these two sequences are concatenated to make the input of the difference encoder $\boldsymbol{Z}$ like $\boldsymbol{Z} = [\mathcal{X}, \boldsymbol{X}, \mathcal{Y}, \boldsymbol{Y}]$. Then, positional encoding $\mathscr{P}$ is applied to $\boldsymbol{Z}$. Finally, $\mathscr{P}(\boldsymbol{Z})$ is input to CAC transformer encoder to obtain the output $\hat{\boldsymbol{Z}} = [\hat{\mathcal{X}}, \hat{\boldsymbol{X}}, \hat{\mathcal{Y}}, \hat{\boldsymbol{Y}}]$.

3.3. Text decoder

The transformer decoder is utilized as a text decoder like as [5]. The text decoder calculates word probability from the output of the difference encoder $\hat{\boldsymbol{Z}}$.

3.4. Similarity-discrepancy disentanglement

The similarity-discrepancy disentanglement (SDD) loss function is an auxiliary loss function aimed at obtaining a difference-emphasized audio representation. When there is an explainable difference between two audio clips, these clips consist of similar and discrepant parts. To introduce this hypothesis, we design contrastive learning to bring similar parts closer and keep discrepant parts. We propose two types of implementations that apply SDD to the input of the difference encoder $\boldsymbol{Z}$ or the output of it $\hat{\boldsymbol{Z}}$, as shown in Fig. 2, and call the former and latter implementations early and late disentanglement, respectively.

We explain the procedure in the case of early disentanglement. Note that the case of late disentanglement only replaces $\boldsymbol{Z}$ with $\hat{\boldsymbol{Z}}$. First, $\boldsymbol{Z}$ is split along the hidden dimension and assigned to similar and discrepant parts like in the upper left illustration of Fig. 2. If $\boldsymbol{Z} \in \mathbb{R}^{H \times (T_x+T_y+2)}$, $\boldsymbol{Z}$ is split into similar part $\boldsymbol{Z}_{\mathrm{S}}$ and discrepant part $\boldsymbol{Z}_{\mathrm{D}}$ like

$$\boldsymbol{Z}_{\mathrm{S}} = [\mathcal{X}_{\mathrm{S}}, \boldsymbol{X}_{\mathrm{S}}, \mathcal{Y}_{\mathrm{S}}, \boldsymbol{Y}_{\mathrm{S}}] \in \mathbb{R}^{(H/2) \times (T_x+T_y+2)}, \tag{1}$$

$$\boldsymbol{Z}_{\mathrm{D}} = [\mathcal{X}_{\mathrm{D}}, \boldsymbol{X}_{\mathrm{D}}, \mathcal{Y}_{\mathrm{D}}, \boldsymbol{Y}_{\mathrm{D}}] \in \mathbb{R}^{(H/2) \times (T_x+T_y+2)}. \tag{2}$$

Then, the SDD is performed by $\mathcal{L}_{\mathrm{SDD}} = \mathcal{L}_{\mathrm{S}} + \mathcal{L}_{\mathrm{D}}$, where

$$\mathcal{L}_{\mathrm{S}} = \mathrm{SymInfoNCE}(\Phi([\mathcal{X}_{\mathrm{S}}, \boldsymbol{X}_{\mathrm{S}}]), \Phi([\mathcal{Y}_{\mathrm{S}}, \boldsymbol{Y}_{\mathrm{S}}])), \tag{3}$$

$$\mathcal{L}_{\mathrm{D}} = \mathrm{PairCosSim}(\Psi([\mathcal{X}_{\mathrm{D}}, \boldsymbol{X}_{\mathrm{D}}]), \Psi([\mathcal{Y}_{\mathrm{D}}, \boldsymbol{Y}_{\mathrm{D}}])), \tag{4}$$

SymInfoNCE is the symmetric version of the InfoNCE loss used in [19], PairCosSim is the cosine similarity for each correct data pair, Φ and Ψ are embedding networks consisting of the bidirectional-LSTM and average pooling, and $\mathcal{L}_{\mathrm{SDD}}$ is the final value of the SDD loss function. That is, the SDD loss function views $[\mathcal{X}_{\mathrm{S}}, \boldsymbol{X}_{\mathrm{S}}]$ and $[\mathcal{Y}_{\mathrm{S}}, \boldsymbol{Y}_{\mathrm{S}}]$ as similar parts and brings them closer by using $\mathcal{L}_{\mathrm{S}}$ and views $[\mathcal{X}_{\mathrm{D}}, \boldsymbol{X}_{\mathrm{D}}]$ and $[\mathcal{Y}_{\mathrm{D}}, \boldsymbol{Y}_{\mathrm{D}}]$ as discrepant parts and keeps them apart by $\mathcal{L}_{\mathrm{D}}$.

The entire loss function $\mathcal{L}$ is the weighted sum of cross-entropy loss for word prediction $\mathcal{L}_{\mathrm{CE}}$ and the SDD: $\mathcal{L} = \mathcal{L}_{\mathrm{CE}} + \lambda \mathcal{L}_{\mathrm{SDD}}$, where λ is a weighting parameter.

4. EXPERIMENT

Experiments were conducted to evaluate the proposed CAC transformer encoder and SDD loss function. We constructed the AudioDiffCaps dataset consisting of pairs of similar but slightly different audio clips and a human-annotated description of their differences for the experiments.

4.1. AudioDiffCaps dataset

The constructed AudioDiffCaps dataset consists of (i) pairs of similar but slightly different audio clips and (ii) human-annotated descriptions of their differences.

The pairs of audio clips were artificially synthesized by mixing foreground event sounds with background sounds taken from existing environmental sound datasets (FSD50K [15] and ESC-50 [11])

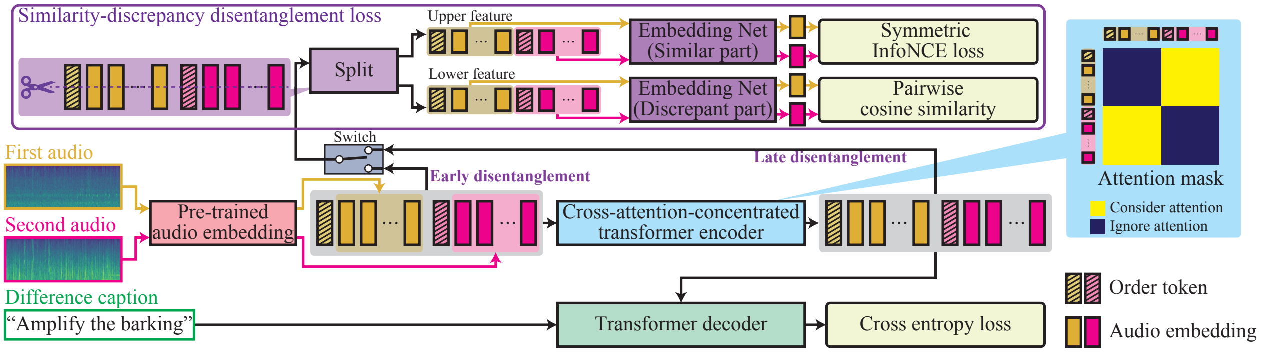


Figure 2: Model architecture of our proposed method. The cross-attention-concentrated transformer encoder uses an attention mask illustrated in the upper left. The similarity-discrepancy disentanglement is conducted by symmetric InfoNCE loss and pairwise cosine similarity. The input to them is either the input or output of the cross-attention-concentrated transformer encoder.

using the Scaper library for soundscape synthesis and augmentation [20]. We used the same mixing procedure as our previous work [21]. Data labeled *rain* or *car_passing_by* in FSD50K was used as background, and six foreground event classes were taken from ESC-50 (i.e., data labeled *dog*, *chirping_bird*, *thunder*, *footsteps*, *car_horn*, and *church_bells*). Each created audio clip was 10 seconds long. The maximum number of events in one audio clip was two, with 0-100% overlap (no overlap-range control applied). Each foreground event class had 32 or 8 instances in the development or evaluation set, respectively. Similar to previous work, we focused on the three types of difference: increase/decrease of background sounds, increase/decrease of sound events, and addition/removal of sound events. The development and evaluation sets contained 5996 and 1720 audio clip pairs, respectively. (That is, development and evaluation sets contained 11992 and 3440 audio clips.)

The human-annotated descriptions were written as instruction forms explaining "what and how" to change the first audio clip to create the second audio clip. In the preliminary study, we found that declarative sentences, in some cases, tend to use ordinal numbers such as "*First sound* is louder than *second sound*". Since these cases do not express what the actual difference is, the AudioDiffCaps dataset uses instruction forms with a fixed direction of change from the first audio clip to the second one, e.g., "Make the rain louder" [2] . A wider variety of descriptions explaining the same concept, such as declarative sentences, should be included in future works. The presentation order of the pair to the annotator was randomly selected. Annotators were five naïve workers remotely supervised by an experienced annotator. Each pair of audio clips in the development set had between 1 and 5 descriptions (a total of 28,892) while each pair in the evaluation set had exactly five descriptions assigned to it (a total of 8600).

4.2. Experimental conditions

We used 10% of the development set for validation. The optimizer was Adam [22]. The number of epochs was 100. We used the BLEU-1, BLEU-4, METEOR, ROUGE-L, CIDEr [23], SPICE [24], and SPIDEr [25] as evaluation metrics. They were also used for conventional audio captioning [26].

We used BYOL-A [27], a pre-trained audio embedding model, as the audio feature extractor in our ADC implementation, and we fine-tuned the BYOL-A throughout experiments. The transformer encoder and decoder used the official implementation of PyTorch. The number of layers was 1. The hidden size was 768. The number of heads was 4. The activation was RELU. The dimension of the feedforward layer was 512. The dropout rate was 0.1. For the attention mask of the transformer encoder, we compared two types; one with the proposed cross-attention mask and the other without a mask. The text decoder used the teacher forcing algorithm during training and the beam search algorithm [28, 29] during inference. The value of λ was empirically set to 0, 0.5, 1.0, or 2.0.

4.3. Results

The results of evaluation metrics are shown in Table 1, where bold font indicates the highest score, "Mask" and "Disent." indicate the attention mask utilized in the transformer encoder and input of SDD loss function, respectively. When the CAC transformer encoder was evaluated by comparing the two lines above, the proposed method had superior or equivalent scores to the conventional method in all evaluation metrics. There was no significant difference in the evaluation metrics related to the degree of matching with single-word references, such as BLEU-1. One likely reason is that the scores above a certain level can be obtained by outputting words in arbitrary sentences, such as "a" and "the" in these metrics. In contrast, the scores of BLEU-4, ROUGE-L, CIDEr, and SPIDEr, affected by the accuracy of consecutive words, were improved using the proposed cross-attention mask. Therefore, the proposed cross-attention mask was thought to make the feature extraction of differences more efficient and simplify the training of the text decoder. As a result, phrase-level accuracy was improved.

The effect of SDD was verified from the results of the second to eighth lines. The results in (a) and (b) were the conventional transformer without cross attention mask or SDD loss and the CAC transformer without SDD loss ($\lambda = 0$) Ones from (c) to (h) were the result when using early/late disentanglement. Since the scores of BLEU-4, ROUGE-L, CIDEr, and SPIDEr improved under all conditions comparing (b) and others, the SDD loss function was effective for the audio difference captioning task. The improvement in the case of late disentanglement (f), (g), and (h) was remarkable, and the results obtained the best scores in all evaluation metrics with late disentanglement. In other words, it was essential to use the information to be compared to decompose the similar part and the different parts in the feature amount space. That corresponds to the difference determined depending on the comparison target.

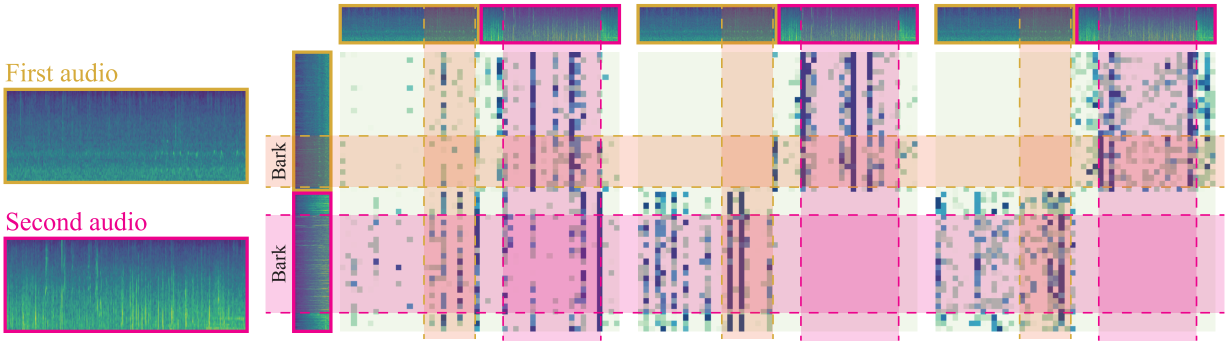
Fig. 3 shows one of the evaluation data and estimated caption and attention weight of the transformer encoder from each system. The leftmost colomn is the Mel-spectrogram of the two input audio

---

[2] The dataset is available at `https://github.com/nttcslab/audio-diff-caps`.

Table 1: Results of evaluation metrics

ID	System	Mask	Disent.	BLEU-1	BLEU-4	METEOR	ROUGE-L	CIDEr	SPICE	SPIDEr
(a)	Baseline	N/A	N/A	67.1	31.7	24.3	56.9	82.7	19.5	51.1
(b)	CAC transformer	Cross	N/A	67.0	33.4	25.2	59.5	90.2	19.5	54.9
	CAC transformer									
(c)	w/ Early SDD ($\lambda = 0.5$)	Cross	Early	67.0	33.7	25.3	59.6	91.8	19.4	55.6
(d)	w/ Early SDD ($\lambda = 1.0$)	Cross	Early	66.8	32.2	25.3	59.3	91.7	19.5	55.6
(e)	w/ Early SDD ($\lambda = 2.0$)	Cross	Early	66.9	33.5	25.3	59.6	92.8	18.7	55.8
(f)	w/ Late SDD ($\lambda = 0.5$)	Cross	Late	**70.3**	39.2	**26.4**	**61.6**	**97.6**	21.3	59.4
(g)	w/ Late SDD ($\lambda = 1.0$)	Cross	Late	69.9	38.3	26.3	61.5	96.3	21.2	58.7
(h)	w/ Late SDD ($\lambda = 2.0$)	Cross	Late	69.9	**39.5**	26.3	61.3	97.1	**22.6**	**59.9**



Reference caption:
increase the pitch and volume of the dog barks

Output caption:
increase the volume of and thunder the bird the the

(a) Baseline

Output caption:
amplify the rain and the and thunder thunder the the

(b) Cross-attention-concentrated transformer

Output caption:
amplify the sound of the dog

(h) Cross-attention-concentrated transformer w/ Late disentanglement(λ=2.0)

Figure 3: Examples of output caption and attention weights. The leftmost row was the Mel-spectrogram of two audio clips and one reference caption. The three on the right were the attention weights of the transformer encoder and the output caption.

clips and one of the reference captions. The three on the right are the attention weight of the transformer encoder and output caption, where the attention weight shows the average of multiple heads. The audio clips on the left and above the weights correspond to the input and memory of the transformer, respectively. The area colored pink and yellow on the weights corresponds to the dog barking. Since there was a difference in the loudness of the dog barking between the two clips, the attention was expected to focus on areas where pink and yellow overlap to extract the difference.

First, in (a), since the attention weight was not constrained, it was also distributed widely to areas other than the above compared with the other two. On the other hand, the attention weights of (b) and (h) concentrated on areas where pink and yellow overlap since the attention of the same input and memory was unavailable. Comparing (b) and (h), while the attention of the part containing the barking of the dog in the memory was large at any time-frame in (b), more attention was paid to the pink and yellow overlapping areas where both input and the memory contain the barking of the dog in (h). Since the late disentanglement required that similar and discrepant parts be retained in the output of the transformer encoder calculated using these attention weights, it was thought that the late disentanglement induced attention to be paid to the part where there was a difference when comparing the two sounds instead of paying attention to the parts that are likely to exist the difference compared with the distribution of training data, such as a dog barking.

5. CONCLUSION

We proposed *Audio Difference Captioning* (ADC) as a new extension task of audio captioning for describing the semantic differences between similar but slightly different audio clips. The ADC solves the problem that conventional audio captioning sometimes generates similar captions for similar but slightly different audio clips, failing to describe the difference in content. We also propose a cross-attention-concentrated transformer encoder to extract differences by comparing a pair of audio clips and a similarity-discrepancy disentanglement to emphasize the difference feature in the latent space. To evaluate the proposed methods, we newly built an AudioDiffCaps dataset consisting of pairs of similar but slightly different audio clips and a human-annotated description of their differences. We experimentally showed that since the attention weights of the cross-attention-concentrated transformer encoder are restricted only to the mutual direction of the two inputs, the differences can be efficiently extracted. Thus, the proposed method solved the ADC task effectively and improved the evaluation metric scores.

Future work includes utilizing a pre-trained generative language model such as BART [30] and applying a wider variety of audio events and types of differences.

6. ACKNOWLEDGMENTS

BAOBAB Inc. supported the annotation for the dataset.

7. REFERENCES

[1] C. D. Kim, B. Kim, H. Lee, and G. Kim, "AudioCaps: Generating captions for audios in the wild," in *Proc. Conf. N. Am. Chapter Assoc. Comput. Linguist.*, 2019, pp. 119–132.

[2] K. Drossos, S. Adavanne, and T. Virtanen, "Clotho: An audio captioning dataset," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, 2019, pp. 736–740.

[3] D. Takeuchi, Y. Koizumi, Y. Ohishi, N. Harada, and K. Kashino, "Effects of word-frequency based pre- and post-processings for audio captioning," in *Proc. Detect. Classif. Acoust, Scenes Events Workshop (DCASE)*, November 2020.

[4] X. Xu, H. Dinkel, M. Wu, and K. Yu, "A crnn-gru based reinforcement learning approach to audio captioning." in *Proc. Detect. Classif. Acoust. Scenes Events (DCASE) Workshop*, 2020, pp. 225–229.

[5] X. Mei, X. Liu, Q. Huang, M. D. Plumbley, and W. Wang, "Audio captioning transformer," in *Proc. Detect. Classif. Acoust. Scenes Events (DCASE) Workshop*.

[6] F. Gontier, R. Serizel, and C. Cerisara, "Automated audio captioning by fine-tuning bart with audioset tags," in *Proc. Detect. Classif. Acoust. Scenes Events (DCASE) Workshop*, 2021.

[7] Y. Koizumi, Y. Ohishi, D. Niizumi, D. Takeuchi, and M. Yasuda, "Audio captioning using pre-trained large-scale language model guided by audio-based similar caption retrieval," *arXiv preprint arXiv:2012.07331*, 2020.

[8] X. Xu, M. Wu, and K. Yu, "Diversity-controllable and accurate audio captioning based on neural condition," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, 2022, pp. 971–975.

[9] X. Mei, X. Liu, J. Sun, M. D. Plumbley, and W. Wang, "Diverse audio captioning via adversarial training," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, 2022, pp. 8882–8886.

[10] X. Liu, X. Mei, Q. Huang, J. Sun, J. Zhao, H. Liu, M. D. Plumbley, V. Kilic, and W. Wang, "Leveraging pre-trained bert for audio captioning," in *Proc. 24th Eur. Signal Process. Conf. (EUSIPCO)*, 2022, pp. 1145–1149.

[11] K. J. Piczak, "ESC: Dataset for Environmental Sound Classification," in *Proc. 23rd Annual ACM Conf. Multimedia*, pp. 1015–1018.

[12] D. Barchiesi, D. Giannoulis, D. Stowell, and M. D. Plumbley, "Acoustic scene classification: Classifying environments from the sounds they produce," *IEEE Signal Process. Mag.*, vol. 32, no. 3, pp. 16–34, 2015.

[13] A. Mesaros, T. Heittola, and T. Virtanen, "TUT database for acoustic scene classification and sound event detection," in *Proc. 24th Eur. Signal Process. Conf. (EUSIPCO)*. IEEE, 2016, pp. 1128–1132.

[14] J. F. Gemmeke, D. P. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio set: An ontology and human-labeled dataset for audio events," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*. IEEE, 2017, pp. 776–780.

[15] E. Fonseca, X. Favory, J. Pons, F. Font, and X. Serra, "Fsd50k: an open dataset of human-labeled sound events," *arXiv preprint arXiv:2010.00475*, 2020.

[16] S. Hershey, S. Chaudhuri, D. P. W. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold, M. Slaney, R. J. Weiss, and K. Wilson, "CNN architectures for large-scale audio classification," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*. IEEE, 2017, pp. 131–135.

[17] Q. Kong, Y. Cao, T. Iqbal, Y. Wang, W. Wang, and M. D. Plumbley, "PANNs: Large-scale pretrained audio neural networks for audio pattern recognition," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 28, pp. 2880–2894, 2020.

[18] Y. Gong, Y.-A. Chung, and J. Glass, "AST: Audio spectrogram transformer," in *Proc. Interspeech*, 2021, pp. 571–575.

[19] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning transferable visual models from natural language supervision," *arXiv preprint arXiv:2103.00020*, 2021.

[20] J. Salamon, D. MacConnell, M. Cartwright, P. Li, and J. P. Bello, "Scaper: A library for soundscape synthesis and augmentation," in *Proc. IEEE Workshop Appl. Signal Process. Audio Acoust. (WASPAA)*. IEEE, 2017, pp. 344–348.

[21] D. Takeuchi, Y. Ohishi, D. Niizumi, N. Harada, and K. Kashino, "Introducing auxiliary text query-modifier to content-based audio retrieval," in *Proc. Interspeech*, 2022.

[22] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. in Int. Conf. Learn. Represent. (ICLR)*, 2014.

[23] R. Vedantam, C. Lawrence Zitnick, and D. Parikh, "CIDEr: Consensus-based image description evaluation," in *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2015, pp. 4566–4575.

[24] P. Anderson, B. Fernando, M. Johnson, and S. Gould, "SPICE: Semantic propositional image caption evaluation," in *Eur. Conf. Comput. Vis. (ECCV)*, 2016, pp. 382–398.

[25] S. Liu, Z. Zhu, N. Ye, S. Guadarrama, and K. Murphy, "Improved image captioning via policy gradient optimization of spider," in *IEEE Int. Conf. Comput.Vis. (ICCV)*, 2017, pp. 873–881.

[26] DCASE2022 Challenge Task 6: Automated Audio Captioning and Language-Based Audio Retrieval, https://dcase.community/challenge2022/task-automatic-audio-captioning-and-language-based-audio-retrieval.

[27] D. Niizimi, D. Takeuchi, Y. Ohishi., N. Harada, and K. Kashino, "BYOL for audio: Self-supervised learning for general-purpose audio representation," in *Proc. Int. Jt. Conf. Neural Netw. (IJCNN)*. IEEE, 2021.

[28] P. Koehn, *Statistical machine translation*. Cambridge University Press, 2009.

[29] ——, "Pharaoh: a beam search decoder for phrase-based statistical machine translation," in *Mach. Transl.: Real Users Res. Proc. 6th Conf. Assoc. Mach. Transl. Am. (AMTA-2004)*, vol. 3265, 2004.

[30] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," in *Proc. 58th Annu. Meet. Assoc. Comput. Linguist.*, 2020, pp. 7871–7880.

CROSS-DIMENSIONAL INTERACTION WITH INVERTED RESIDUAL TRIPLET ATTENTION FOR LOW-COMPLEXITY SOUND EVENT DETECTION

Tanmay Khandelwal and Rohan Kumar Das

Fortemedia Singapore, Singapore
f20170106p@alumni.bits-pilani.ac.in, rohankd@fortemedia.com

ABSTRACT

Attention mechanisms have been widely used in a variety of sound event detection (SED) tasks, owing to their ability to build interdependencies among channels or spatial locations. The existing state-of-the-art (SOTA) architectures and attention modules incorporated in SED have a high computational cost in terms of the number of parameters. To address this issue, we propose a lightweight module utilizing triplet attention on an inverted residual network (IRN) referred to as an inverted residual triplet attention module (IRTAM) for replacing the standard 2D convolutional neural network. The IRTAM captures cross-dimensional interdependencies using the rotation operation followed by residual transformations with a three-branch structure embedded in IRN. On DCASE 2022 Task 4 validation set, the proposed lightweight module improves the performance of the baseline by 34.1% in terms of polyphonic sound event detection score and achieves SOTA results with only 27.6% parameters of the baseline.

Index Terms: sound event detection, low-complexity, triplet attention, inverted residual network

1. INTRODUCTION

Sounds help in better understanding our surroundings and in detecting environmental changes. The ability to recognize and classify sound events in our surroundings is inherent in the human body. The sound event detection (SED) systems *automate* this process to detect the sound events to mark their corresponding onset and offset. It has important practical applications as well as theoretical significance and has been applied to audio surveillance in environments such as smart-homes, cities, and monitoring biodiversity.

Real-world audio recordings frequently contain numerous overlapping sound occurrences. Recent advances in predicting and recognizing these overlapping events have shifted from traditional methods like Gaussian mixture models [1], hidden Markov models [2], and support vector machines [3] to advanced deep learning techniques. The recent success of convolutional recurrent neural networks (CRNNs) [4] and transformer [5] structures have achieved state-of-the-art (SOTA) results in the field of SED. These modern, cutting-edge structures demand high computing resources that are beyond the capacity of many embedded and mobile applications. Therefore, reducing the number of parameters in a SED model allows the method to be fit for systems with limited resources while also decreasing the training time.

Most of the previously built systems [6, 7] proposed the use of *depthwise separable convolutions* and showed the system's effectiveness with reduced parameters. Another way to target an effective *low-complexity* SED system is to use *attention mechanisms* [8]. In human perception, attention refers to the process of selectively concentrating on parts of the given information while ignoring the rest. This mechanism aids in the refinement of perceived information while preserving its context. In the case of deep learning systems with a basic building block as the 2D convolutional layer, filters capture local spatial patterns along all input channels and generate feature maps jointly encoding the time-frequency and channel information.

Several works have been aimed at capturing either *spatial* or *channel* attention, done by building dependencies among channels or weighted spatial masks for *spatial* attention. One such promising approach is a component called the squeeze and excitation (SE) [9] block, which can be seamlessly integrated into the convolutional neural network (CNN). This SE block removes the spatial dependency by using global average pooling to learn a channel-specific descriptor, which is then used to rescale the input feature map to highlight only useful channels. The SE block was succeeded by the convolutional block attention module (CBAM) [10], which emphasized the importance of providing robust representative attention by combining *spatial* and *channel* attention. This method of combining *spatial* attention and *channel* attention improved the performance compared to the SE block. However, most attention modules add substantial computational overhead, and stacking these complex modules usually ignores the interdependence between *spatial* dimensions and *channel* dimension of the input feature.

In this work, we devote to incorporating *cross-dimensional* interaction while computing attention weights to provide rich feature representations for *low-complexity* SED systems by a novel inverted residual *triplet attention* module (IRTAM) that uses a three-branch structure, where each branch is responsible for aggregating *cross-dimensional* interactive features. We summarize the major contributions of this work as follows:

- Inspired from *MobileNetV2* [11], we propose to incorporate an inverted residual network (IRN) with a linear bottleneck to replace the standard 2D convolution block. The IRN makes the SED model suitable to be deployed for *real-time* applications on low computational devices.
- We propose to introduce a *triplet attention* [12] module into the IRN at a negligible computational overhead to effectively learn *cross-dimensional* interaction. The attention module is made up of *three branches*, each of which is responsible for capturing the *cross-dimensional* interaction between the input's *spatial* dimensions and *channel* dimension.

We consider the two-stage system developed by [13–15] for the detection and classification of acoustic scenes and events (DCASE) 2022 Task 4 participation for the studies in this work. We also used *data augmentation* and *adaptive post-processing* techniques to increase the robustness of the developed system.

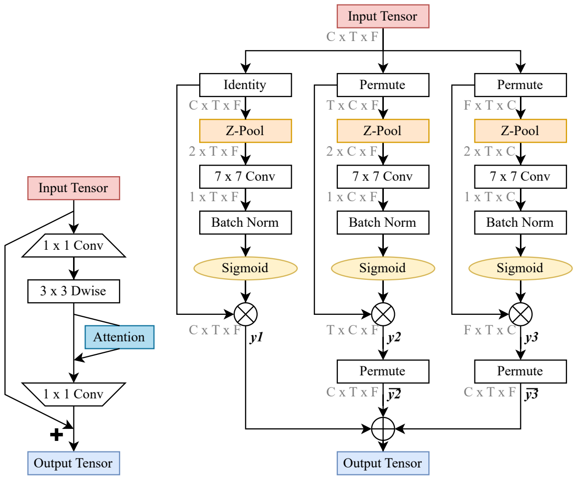
2. SOUND EVENT DETECTION SYSTEM

2.1. Baseline

The baseline [16] architecture, adopted from the DCASE Task 4 Challenge 2022, is a CRNN that combines a CNN and a recurrent neural network (RNN). The CNN block is composed of 7 layers, each with 16, 32, 64, 128, 128, 128, and 128 filters. The kernel size for each convolutional layer is 3×3 and each layer is followed by a Gaussian error linear unit activation and batch normalization. For frequency and temporal pooling, the average pooling layer is employed, and its sizes are [[2, 2], [2, 2], [1, 2], [1, 2], [1, 2], [1, 2], [1, 2]], respectively. The RNN block is made up of two layers of 128 bidirectional gated recurrent units (Bi-GRUs), resulting in a total of 1.1M parameters. After the RNN block comes the attention pooling layer, which is the product of multiplying a linear layer with softmax activations and a linear layer with sigmoid activations. The baseline employs the *mean-teacher* (MT) model, which updates the teacher model's weights using an exponential moving average from the student model.

2.2. Inverted residual network (IRN)

Taking inspiration from *MobileNetV2* [11], we propose to incorporate IRN to replace the standard 2D convolutions, as depicted in Figure 1 (a). The proposed replacement has a distinct property that allows the network expressiveness (encoded by expansion layers) to be separated from its capacity (encoded by bottleneck inputs). Further, it allows lightweight model implementation for low-computational embedded systems. The block uses *depthwise separable convolutions* to replace the fully convolutional operations with a factorized version to split the standard convolution into two separate layers. The block performs three separate convolutions. First, a *pointwise convolution* is used to expand the low-dimensional input feature map to a higher-dimensional space. Followed by a *depthwise convolution*, achieving spatial filtering. Finally, the spatially filtered feature map is projected back to a low-dimensional subspace using another pointwise convolution. Figure 1 (a) shows the residual link between low-dimensional feature maps.



(a) (b)

Figure 1: The proposed architectures (a) IRN with attention module (b) triplet attention module for SED.

2.3. Triplet attention

We propose to incorporate an effective channel attention module, namely, *triplet attention* [12], into the IRN. The *cross-dimensional* interaction is captured by this parameter-free attention mechanism, which may be integrated into other standard networks. In traditional ways of computing channel attention, there is a significant loss of spatial information as the input tensor is spatially decomposed into one pixel per channel by employing global average pooling. This leads to a loss of interdependence between the *channel* dimension and *spatial* dimension on these single pixels. Further, as the *channel* attention and *spatial* attention in [9] are computed independently of each other, the relationship between the two is not considered. To address this issue, we propose capturing *cross-dimensional* interaction with no dimensionality reduction by adding *triplet attention* to the IRN for SED applications.

The *triplet attention* is composed of *three parallel branches*, built to capture dependencies between the (C, F), (C, T), and (F, T) of the input feature, where C, F, T represent the channel, frequency, time feature maps, respectively. Two of the branches capture the cross-dimension interaction between the channel dimension C and either the spatial dimension F or T. The last, final branch resembles CBAM, which is used to build *spatial attention*. For each branch, the input is permuted as shown in Figure 1 (b), and then it is passed through *Z-pool*. The *Z-pool* layer reduces the zeroth dimension to two by concatenating average pooling and max pooling across that dimension. This helps to retain a rich representation while shrinking the depth, resulting in less computational requirement. The operation of *Z-pool* is as follows:

$$Z\text{-}pool(x) = [MaxPool_{0d}(x), AvgPool_{0d}(x)] \tag{1}$$

where *0d* is the 0th-dimension along which the operation is applied and *x* is the input tensor. The resultant from the *Z-pool* is passed through a standard convolutional layer of kernel size 7×7, followed by batch normalization. The attention weights are generated by passing the tensor through a sigmoid function and are applied to the input tensor for the respective branch. The resulting output is then rotated back to its original state to retain the original input shape. The results of all *three branches (y1, y2, y3)* are aggregated with straightforward *averaging* as given below:

$$y = \frac{1}{3}(y1 + \bar{y}2 + \bar{y}3) \tag{2}$$

where $\bar{y}2$ and $\bar{y}3$ represents the 90° clockwise permutation to retain the original input shape of (C × T × F).

2.4. Proposed architecture

We employed the IRN described in Section 2.3 to replace the standard 2D CNNs, which results in a smaller amount of parameters. The *triplet attention* module was plugged in after the *depthwise separable convolution* in the IRN, as shown in Figure 1 (a). This newly generated module is referred to as IRTAM, which enables the acquisition of more blended *cross-dimensional* feature information. The updated architecture has the same number of layers, but the size of the feature map in each module is reduced to 16, 32, 64, 64, 64, 64, and 64, respectively. The updated architecture consists of 2 layers of Bi-GRU with 64 hidden units, resulting in a total of 304k parameters for the entire model compared to the 1.1M parameters in the baseline. In summary, the updated architecture with the proposed replacement has 27.6% of the parameters of the baseline.

Table 1: Summary of DCASE 2022 Task 4 development set.

Clips	Description
10,000	Synthetic strongly labeled data
3,470	Real strongly labeled data (external set)
1,578	Real weakly labeled data
14,412	In-domain unlabeled data
1,168	Real strongly labeled validation data
2,500	Synthetic strongly labeled validation data

3. EXPERIMENTAL SETUP

3.1. Dataset

In our experiments, we used the DCASE 2022 Task 4 dataset, which is identical to the DCASE 2023 Task 4A dataset and consists of 10-second audio clips extracted from AudioSet or constructed using isolated sound events to simulate a domestic environment. The split for the development training set is reported in Table 1. Additionally, the public evaluation ("YouTube" evaluation) collection consists of 692 YouTube clips.

3.2. Pre-processing

The audio clips are first re-sampled at 16 kHz to a mono channel. They are then segmented using a window size of 2048 samples with a hop length of 256 samples. The spectrograms of segmented waveforms are extracted using the short-time Fourier transform. Then, log-mel spectrograms are created by using mel-filters in the frequency domain of 0 to 8 kHz, followed by a logarithmic operation. Silence is used to pad the clips that are less than 10-seconds long.

3.3. Two-stage system for SED

We incorporate the two-stage system developed by [13–15] for DCASE 2022 Task 4 participation, depicted in Figure 2. In this system, Stage-1 focuses on audio-tagging (AT), whereas Stage-2 improves SED by using the reliable *pseudo-labels* generated by Stage-1. To extract the embeddings in Stage-1, we used a CNN-14-based pre-trained audio neural network [17] as the feature extractor. The embeddings extracted are fed into the Bi-GRU, which has 2 layers with 1024 hidden units. Stage-1 is trained using a strongly labeled set converted into weak predictions referred to as a *weakified* set, a weakly labeled set, and an unlabeled set with 64 mel-bins, to improve AT performance, as shown in Figure 2. Additionally, the AT system (Stage-1) predicted unlabeled set and employed those as *pseudo-weak* labels in Stage-2 training with 128 mel-bins. In Stage-2, we used the proposed lightweight architecture with 304k parameters described in Section 2.4. It is trained on a *pseudo-weakly* labeled set in addition to the strongly labeled and the weakly labeled set in a supervised manner. In training, the weak and *pseudo-weak* sets were merged. Both strongly and weakly labeled samples were assigned a weight of 1 using the baseline system's loss functions.

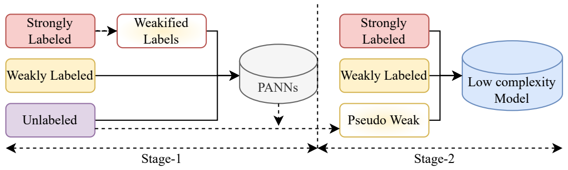


Figure 2: Two-stage system, with Stage-1 focusing on AT and Stage-2 focusing on SED.

3.4. Training process

For all experiments, the batch size is 48 (1/4 strong set, 1/4 weak set, 1/2 unlabeled set). We employed the Adam optimizer with a maximum learning rate of 0.001 and a learning rate ramp-up over the first 50 epochs of the optimization process. A total of 100 epochs are used to train Stage-1, and 200 epochs are used to train Stage-2. The weak training data was used to generate a 90% training set and a 10% validation set. Then the validation is performed on the 10% held-out weak subset and on the strongly labeled synthetic validation set. The system was built with PyTorch Lightning and trained on an NVIDIA Quadro RTX 5000 GPU.

3.5. Additional methods

We used several *data augmentation* techniques to artificially generate more data and improve the model's robustness during the training in both stages. We employed time-masking [18], frame-shifting, mixup [16], and Gaussian noise addition in Stage-1 and time-masking, frame-shifting, mixup and frequency-masking [18] in Stage-2. We also adopted *adaptive post-processing* [19] in all the experiments, where the median filter window sizes are different for each event category, calculated *heuristically* based on the varying length of each event in real life. Furthermore, for each class, we used *probability value correction* [20], in which we multiplied the probability generated by the model by a magnification factor to correct the probability to a maximum value of 1. For inference *temperature tuning* as in [21] a temperature factor of 2.1 is employed. In our final developed system, we also used the *external set* released by DCASE 2022 Task 4 organizers during training, with each stage employing MT and *interpolation consistency training* (ICT) [22] to utilize the unlabeled training data.

3.6. Evaluation metric

In our studies, we used polyphonic sound event detection scores (PSDS) [23] introduced in the DCASE 2022 Task 4 as a performance metric to evaluate the systems. The PSDS is more resistant to labeling subjectivity, allowing for the interpretation of both the ground truth and the detection of temporal structure. It computes a *single* PSDS using polyphonic receiver operating characteristic curves, allowing for comparison regardless of the operating point. Furthermore, it can be customized for a variety of applications, ensuring that the desired user experience is achieved. As a result, it *overcomes* the limitations of traditional event F-scores based on collars. We compute the PSDS in our studies using *two* different scenarios that emphasize different system properties. Scenario-1 requires the system to respond quickly to event detection, focusing on the temporal localization of the sound event. Scenario-2, on the other hand, focuses on preventing class confusion rather than reaction time. The greater the values for PSDS1 and PSDS2, the better for both scenarios. Notably, the PSDS metric employed here adheres to the DCASE 2022 Task 4 protocol and differs from the threshold-independent PSDS used in DCASE 2023 Task 4A.

4. RESULTS AND ANALYSIS

4.1. Proposed IRTAM

We consider the two-stage framework described in the previous section for our studies with *low-complexity* SED systems.

Table 2: Performance comparison showing the importance of the proposed method on DCASE 2022 Task 4 validation set.

System	PSDS1	PSDS2	#Parameters
Baseline	0.351	0.552	1.1M
IRN	0.343	0.519	301k
IRN + SE	0.359	0.521	442k
IRN + CA	0.419	0.694	333k
IRN + Triplet Attention (IRTAM)	**0.440**	**0.708**	304k
+ data augmentation	0.446	0.702	304k
+ external set	0.457	0.712	304k
+ ICT	0.471	0.710	304k
+ median filtering	0.480	0.727	304k
+ probability correction	**0.483**	**0.728**	304k

First, in Stage-2 used for inference, we replace the standard 2D CNNs in the baseline with the proposed IRN described in Section 2.3, resulting in a reduction of parameters from 1.1M in the baseline to 301k. From Table 2, we observe a minor degradation in the performance with a decrease in PSDS1 from 0.351 to 0.343 and in PSDS2 from 0.552 to 0.519 owing to the reduction in the number of parameters. Following our proposed design, we next incorporate an attention module in the IRN after the *depthwise separable convolution* layer to assist the model in learning the *frequency-dependent* patterns and feature *interdependencies* between channels and time-frequency locations.

We are also interested in comparing the performance of the proposed IRTAM (IRN + *triplet attention*) with widely popular SE attention and another recent method, namely, coordinate attention (CA) [24] incorporated in IRN. From Table 2, we observe that the SED performance increases with the introduction of both SE and CA modules. However, on comparing their performance to our proposed IRTAM, we find that IRN with *triplet attention* (IRTAM) performs better than both the other attention modules considered. It is also observed that the increase in the number of parameters for IRTAM is very negligible compared to that with SE and CA. Thus, these studies show the effectiveness of the proposed *low-complexity* IRTAM module, specifically due to the introduction of *triplet attention*, for capturing *cross-dimensional* interaction in SED models. Further, we show the contribution of each additional method discussed in Section 3.5 to apply on the proposed developed system to achieve the final PSDS1 of 0.483 and PSDS2 of 0.728 on the validation set, giving a 34.1% increase compared to the baseline in terms of both *PSDS metrics*.

4.2. Ablation study on triplet attention branches

With the use of a *three-branch* structure, we verify that it is important to capture the *cross-dimensional* interaction between (T, F), (T, C), and (C, F). In Table 3, we compare the results when each branch is turned on, represented by the combination given in each row, to analyze the influence of the branches in the *triplet attention* module. As can be seen, the findings corroborate our understanding that individual and pair branch interaction is inferior to the performance of *triplet attention*, which involves all *three branches* being active.

4.3. System comparison

To further assess the efficacy of the proposed module, the system is also compared with the top-ranked *single* (without ensemble) systems submitted to DCASE 2022 Task 4. In Table 4, the scores for the other systems are directly taken from their cited technical reports released in the challenge. The proposed *low-complexity* system surpasses systems with large parameters and gets close to the top-ranked system, which has 15.4M parameters while having just 304k parameters. We also note that the proposed attention module is network-independent and can be employed in any model to replace standard convolutions with the IRTAM block. Furthermore, on the public evaluation set, the final system with the proposed IRTAM achieved a PSDS1 of 0.488 and a PSDS2 of 0.720, in contrast to the baseline system having a PSDS1 of 0.387 and a PSDS2 of 0.592.

Table 3: Ablation study to show the gain of each branch in the triplet attention on DCASE 2022 Task 4 validation set, where (x,y) is the interplay between dimensions x and y to compute attention weights and aggregated average.

Branch Interaction	PSDS1	PSDS2	#Parameters
(F,T)	0.420	0.643	304k
(C,T)	0.410	0.614	304k
(C,F)	0.424	0.657	304k
((F,T), (C,T))	0.480	0.723	304k
((F,T), (C,F))	0.468	0.716	304k
((C,T), (C,F))	0.459	0.730	304k
((F,T), (C,T), (C,F))	**0.483**	**0.728**	304k

Table 4: Comparison with top-ranked single systems (without ensemble) from DCASE Task 4 2022 on the validation set.

System	PSDS1	PSDS2	#Parameters
Ebbers-UPB-task4 [25]	0.505	0.807	15.4M
Proposed	**0.483**	**0.728**	**304k**
Zhang-UCAS-task4 [26]	0.459	0.672	11M
Kim-GIST-task4 [27]	0.455	0.670	1M
Dinkel-XiaoRice-task4 [28]	0.425	0.644	37M

5. CONCLUSION

In this work, we proposed an inverted residual network with *triplet attention* as a module referred to as IRTAM to replace the standard 2D convolutional neural networks for SED applications. The proposed *low-complexity* attention module was designed to capture *cross-dimensional* interaction with minimal computational overhead. To show the effectiveness of the developed lightweight architecture employing IRTAM, we considered the DCASE 2022 Task 4 dataset for the studies. Our findings demonstrated the efficacy of incorporating *cross-dimensional* interaction in SED applications by improving the baseline by 34.1% and significantly outperforming some other attention modules in both aspects of the *PSDS metric*. Furthermore, our ablation study validated the relevance of capturing *cross-dimensional* interaction using a *three-branch* structure and showed overall effectiveness by achieving comparable results to systems with a large number of parameters. It is also worth noting that the proposed system contains only 27.6% of the baseline parameters, making the model suitable for *low-complexity* SED applications. We intend to extend the proposed IRTAM to larger model sizes in the future.

6. REFERENCES

[1] L. Vuegen, B. V. D. Broeck, P. Karsmakers, J. F. Gemmeke, B. Vanrumste, and H. V. hamme, "An MFCC-GMM approach for event detection and classification," *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2013.

[2] X. Zhuang, X. Zhou, T. S. Huang, and M. Hasegawa-Johnson, "Feature analysis and selection for acoustic event detection," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 17–20, 2008.

[3] L. Lu, F. Ge, Q. Zhao, and Y. Yan, "A SVM-based audio event detection system," *International Conference on Electrical and Control Engineering (ICECE)*, pp. 292–295, 2010.

[4] E. Cakir, G. Parascandolo, T. Heittola, H. Huttunen, and T. Virtanen, "Convolutional recurrent neural networks for polyphonic sound event detection," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 6, pp. 1291–1303, 2017.

[5] K. Chen, X. Du, B. Zhu, Z. Ma, T. Berg-Kirkpatrick, and S. Dubnov, "HTS-AT: A hierarchical token-semantic audio transformer for sound classification and detection," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 646–650, 2022.

[6] K. Drossos, S. I. Mimilakis, S. Gharib, Y. Li, and T. Virtanen, "Sound event detection with depthwise separable and dilated convolutions," *International Joint Conference on Neural Networks (IJCNN)*, pp. 1–7, 2020.

[7] T. Khandelwal, R. K. Das, and E. S. Chng, "Is your baby fine at home? baby cry sound detection in domestic environments," *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pp. 275–280, 2022.

[8] H. Nam, S.-H. Kim, B.-Y. Ko, and Y.-H. Park, "Frequency dynamic convolution: Frequency-adaptive pattern recognition for sound event detection," *Interspeech*, pp. 2763–2767, 2022.

[9] W. Xia and K. Koishida, "Sound event detection in multichannel audio using convolutional time-frequency-channel squeeze and excitation," *Interspeech*, pp. 3629–3633, 2019.

[10] S. Woo, J. Park, J.-Y. Lee, and I.-S. Kweon, "CBAM: Convolutional block attention module," *European Conference on Computer Vision (ECCV)*, pp. 3–19, 2018.

[11] M. Sandler, A. G. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4510–4520, 2018.

[12] D. Misra, T. Nalamada, A. U. Arasanipalai, and Q. Hou, "Rotate to attend: Convolutional triplet attention module," *IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 3138–3147, 2021.

[13] T. Khandelwal, R. K. Das, A. Koh, and E. S. Chng, "FMSG-NTU submission for DCASE 2022 Task 4 on sound event detection in domestic environments," *Detection and Classification of Acoustic Scenes and Events (DCASE) Challenge*, 2022.

[14] ——, "Leveraging audio-tagging assisted sound event detection using weakified strong labels and frequency dynamic convolutions," *IEEE Statistical Signal Processing Workshop*, 2023.

[15] T. Khandelwal and R. K. Das, "Dynamic thresholding on fixmatch with weak and strong data augmentations for sound event detection," *International Symposium on Chinese Spoken Language Processing (ISCSLP)*, pp. 428–432, 2022.

[16] L. Delphin-Poulat and C. Plapous, "Mean teacher with data augmentation for DCASE 2019 Task 4 technical report," *Detection and Classification of Acoustic Scenes and Events (DCASE) Challenge*, 2019.

[17] Q. Kong, Y. Cao, T. Iqbal, Y. Wang, W. Wang, and M. D. Plumbley, "PANNs: Large-scale pretrained audio neural networks for audio pattern recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 2880–2894, 2020.

[18] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "SpecAugment: A simple data augmentation method for automatic speech recognition," *Interspeech*, pp. 2613–2617, 2019.

[19] K. Miyazaki, T. Komatsu, T. Hayashi, S. Watanabe, T. Toda, and K. Takeda, "Convolution-augmented transformer for semi-supervised sound event detection," *Detection and Classification of Acoustic Scenes and Events (DCASE) Challenge*, 2020.

[20] S. Mizobuchi, H. Ohashi, A. Izumi, and N. Kodama, "Mizobuchi PCO team's submission for DCASE 2022 Task 4 sound event detection using external resources," *Detection and Classification of Acoustic Scenes and Events (DCASE) Challenge*, 2022.

[21] X. Zheng, H. Chen, and Y. Song, "Zheng USTC team's submission for DCASE 2021 Task 4 - semi-supervised sound event detection," *Detection and Classification of Acoustic Scenes and Events (DCASE) Challenge*, 2021.

[22] V. Verma, A. Lamb, J. Kannala, Y. Bengio, and D. Lopez-Paz, "Interpolation consistency training for semi-supervised learning," *International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 3635–3641, 2019.

[23] C. Bilen, G. Ferroni, F. Tuveri, J. Azcarreta, and S. Krstulovic, "A framework for the robust evaluation of sound event detection," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 61–65, 2020.

[24] Q. Hou, D. Zhou, and J. Feng, "Coordinate attention for efficient mobile network design," *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13 708–13 717, 2021.

[25] J. Ebbers and R. Haeb-Umbach, "Pre-training and self-training for sound event detection in domestic environments," *Detection and Classification of Acoustic Scenes and Events (DCASE) Challenge*, 2022.

[26] S. Xiao, "Pretrained models in sound event detection for DCASE 2022 challenge Task 4," *Detection and Classification of Acoustic Scenes and Events (DCASE) Challenge*, 2022.

[27] J. W. Kim, G. W. Lee, H. K. Kim, Y. S. Seo, and I. H. Song, "Semi-supervised learning-based sound event detection using frequency-channel-wise selective kernel for DCASE challenge 2022 Task 4," *Detection and Classification of Acoustic Scenes and Events (DCASE) Challenge*, 2022.

[28] H. Dinkel, Z. Yan, Y. Wang, M. Song, J. Zhang, and W. Wang, "A large multi-modal ensemble for sound event detection," *Detection and Classification of Acoustic Scenes and Events (DCASE) Challenge*, 2022.

EXPLORING MULTI-TASK LEARNING WITH WEIGHTED SOFT LABEL LOSS FOR SOUND EVENT DETECTION WITH SOFT LABELS

Tanmay Khandelwal and Rohan Kumar Das

Fortemedia Singapore, Singapore
f20170106p@alumni.bits-pilani.ac.in, rohankd@fortemedia.com

ABSTRACT

The learning of sound events often depends on data that is manually labeled by human annotators. In this study, we explore the use of soft labels for sound event detection (SED), which takes into account the uncertainty and variability in human annotations. To address the challenges posed by uncertain or noisy labels, we propose a weighted soft label (WSL) loss function. This loss function effectively emphasizes reliable annotations while mitigating the influence of less confident or noisy labels. Additionally, we introduce auxiliary tasks into a multi-task learning (MTL) framework, which helps to leverage the shared information between the tasks and improves the overall performance of the model. Furthermore, we explore the usage of pretrained models and various front-end feature extraction methods. Experimental results on the MAESTRO-Real dataset introduced in the DCASE 2023 Task 4B demonstrate a significant improvement of 14.9% in the macro-average F1 score with optimum threshold per class compared to the challenge baseline model on the validation set, highlighting the effectiveness of our proposed system.

Index Terms— sound event detection, soft labels, multi-task learning, acoustic scenes, weighted loss

1. INTRODUCTION

The primary aim of sound event detection (SED) is to autonomously identify and extract significant information from audio recordings, enabling the detection of specific events or activities. SED holds immense potential to augment diverse domains, leading to enhanced safety, convenience, and efficiency. It already plays a critical role in a wide range of applications, including surveillance systems [1, 2], acoustic monitoring [3], smart-homes [4–6], and human-computer interaction.

The lack of labeled training data presents a notable challenge in SED. The process of collecting and annotating extensive audio datasets with labeled sound events is time-consuming and demanding. The scarcity of annotated data impedes the training of accurate models and limits their performance. As a result, researchers are exploring alternative techniques such as employing soft-labeling training methods and transfer learning to mitigate this issue. Soft labels provide a representation of the degree of presence or confidence for specific sound events in each audio segment or frame, in contrast to hard labels that assign binary labels (e.g., 0 or 1). By incorporating confidence scores, soft labels effectively capture the uncertainty and variability associated with sound events, facilitating more nuanced analysis and decision-making processes.

In addition to soft-label generation, researchers are also investigating transfer learning as a means to enhance SED. Transfer learning enables the utilization of knowledge acquired from pretrained models on different but related tasks. Instead of training a model from scratch on a specific SED task, transfer learning allows the model to benefit from the learned representations and features of a pretrained model. Previous works have shown the effectiveness of using the features from pretrained models like pretrained audio neural networks (PANNs) [7–9], audio spectrogram transformers (ASTs) [10], and bidirectional encoder representation from audio transformers (BEATs) [11], trained on a large dataset. The models are fine-tuned using their learned features, customized to the specific SED task at hand, leading to improved performance.

Sound events occurring in nature are typically intricately linked with acoustic scenes. An acoustic scene encompasses the auditory environment in which sound events occur, reflecting the distinctive combination of various sound sources, background noise, and spatial characteristics. Understanding and analyzing sound events within their corresponding acoustic scenes play a pivotal role in SED and related applications. For instance, in the acoustic scene “cafe” the sound events “coffee machine” and “cutlery and dishes” are likely to occur, whereas the sound events “bird singing” and “wind blowing” occur infrequently. On the basis of these previous methods, [12] has proposed methods of SED that take into account acoustic scene information in an unsupervised manner. [13, 14] have proposed scene classification methods considering sound events using Bayesian generative models. Similarly, the methods proposed in [15, 16] focus on the joint analysis of acoustic scenes and sound events using neural network models based on multi-task learning (MTL). Such MTL-based methods leverage the existing knowledge and reduce the need for manual labeling, thus effectively addressing the challenge of data scarcity.

The detection and classification of acoustic scenes and events (DCASE) 2023 edition has recently introduced a new subtask, 4B [17], which aims to explore the potential benefits of incorporating soft labels in improving performance. In our study, we extend the idea of integrating soft labels into the training procedure of SED models. Our investigation specifically revolves around the utilization of soft labels using this newly released dataset in the DCASE 2023 Task 4B dataset [18]. This dataset was specifically designed for exploring the estimation of strong labels through crowdsourcing. It consists of 49 real-life audio files captured from 5 distinct acoustic scenes, accompanied by their corresponding annotation outcomes. To effectively leverage the soft-level probabilities provided in the dataset, we propose a novel weighted soft label (WSL) loss function that mitigates the impact of less confident or noisy labels. Moreover, we delve into the integration of two auxiliary tasks within an MTL framework to enhance the effectiveness of the SED model. To further improve the model’s capabilities, we also explore the utilization of pretrained models and different front-end methods for feature extraction.

Table 1: Categorization of acoustic events into different acoustic scenes for the MAESTRO-Real dataset.

Acoustic event Acoustic scene	Bird singing	Car	People talking	Footsteps	Children voices	Wind blowing	Brakes squeaking	Large vehicle	Cutlery and dishes	Furniture and dragging	Coffee machine	Metro approaching	Metro leaving	Door opens/closes	Announcement	Shopping cart	Cash register beeping
Cafe/Restaurant				✓	✓			✓	✓	✓	✓						
City center		✓	✓	✓	✓		✓	✓									
Grocery store			✓	✓	✓										✓	✓	✓
Metro station			✓	✓	✓							✓	✓	✓			
Residential area	✓	✓	✓	✓	✓	✓											

2. PROPOSED METHODS

2.1. Multi-task learning (MTL) framework

Conventionally, acoustic scenes and sound events have been treated as separate entities in most methods. However, in reality, acoustic scenes play a crucial role in shaping the perception and interpretation of sound events by providing a contextual backdrop. Recognizing the significance of this relationship, we aim to leverage it to gain valuable insights that can enhance SED methods. In this study, we leverage the five acoustic scenes available in the DCASE 2023 Task 4B dataset, and provide a summary of the sound events that take place in these acoustic scenes, as shown in Table 1.

From the table, it is evident that certain sound events, such as “shopping cart” occur exclusively in a specific acoustic scene and are not present in any other acoustic scene. Similarly, the sound event “bird singing” is only observed in residential areas and not in any other acoustic scene. Additionally, we notice that some events, like “footsteps” and “children voices” are common across multiple acoustic scenes. As a result, we propose an additional task of classifying the acoustic environment associated with a sound event as either indoor (I) or outdoor (O). This classification helps to differentiate the surroundings in such sound events. We present two additional tasks related to acoustic scenes: (1) categorizing the acoustic scene for each frame where a sound event takes place, termed acoustic scene classification (ASC), and (2) determining whether each frame’s sound event occurs indoors or outdoors, known as acoustic environment classification (AEC). As depicted in Table 2 the acoustic scenes associated with the sound events are separated into five different classes. Additionally, we determined whether the acoustic scenes were indoors (I) or outdoors (O) based on their respective environments. To enhance the performance of the SED model, we integrate the information from these two auxiliary tasks into the primary SED branch.

Table 2: Classification of the 5 acoustic scenes into different scene labels and environment labels.

Acoustic scene	Scene label	Environment	Environment label
Cafe/Restaurant	A	indoor	I
City center	B	outdoor	O
Grocery store	C	indoor	I
Metro station	D	indoor	I
Residential area	E	outdoor	O

In order to capture low-level features that can benefit all three tasks, we design the network to share certain common layers. These shared layers facilitate the extraction of features that are relevant to all tasks. Previous studies in [19] have demonstrated that leveraging knowledge from easier tasks can improve the performance of harder tasks. In our case, we consider SED as the most challenging task, followed by ASC, and finally AEC. Therefore, we anticipate that the two auxiliary tasks will contribute to improving the SED performance. To conduct the joint training with these two tasks, we use a combined loss function L_{MTL}, which is the weighted loss function. It can be expressed mathematically as

$$L_{MTL} = \alpha \times L_{SED} + \beta \times L_{ASC} + \gamma \times L_{AEC} \tag{1}$$

where α, β, and γ are the trade-off factors that regulate the weighted loss. By adopting an MTL framework with joint training, we benefit from the fact that once the MTL-based model is trained, the auxiliary branches can be removed from the model architecture. During inference, only the single SED branch is utilized, ensuring that the number of parameters remains the same as that of a single SED branch.

2.2. Weighted soft label (WSL) loss

The DCASE 2023 Task 4B baseline uses mean-square error (MSE) loss, to teach the system to predict outputs as close as possible to the provided soft activity indicators instead of binary as described below:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C}(y_{ij} - p_{ij})^2 \tag{2}$$

where N represents the total number of samples, C is the number of classes, y_{ij} is the ground truth soft label for sample i and class j, and p_{ij} is the predicted value for sample i and class j. We extend this loss function to incorporate weights derived from the probabilities assigned to the soft labels by the annotator. Our proposed weighted soft label (WSL) loss function assigns varying importance to each prediction based on its associated probability, as described below:

$$WSL = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C} y_{ij} \cdot (y_{ij} - p_{ij})^2 \tag{3}$$

where y_{ij} also acts as the weight assigned to the soft label for sample i and class j. Higher weight is given to predictions with higher probabilities, indicating a higher level of confidence in those predictions. This weighted approach allows the model to focus more on accurately predicting instances with higher probabilities while considering the uncertainty associated with softer labels. As a result, the model can learn to optimize its performance by prioritizing predictions based on their probability-weighted importance, leading to improved accuracy and robustness.

3. ARCHITECTURE

3.1. Baseline

The baseline system [17] for DCASE 2023 Task 4B adopts the convolutional recurrent neural network (CRNN) architecture with a linear output layer. The convolutional neural network (CNN) component of the model consists of three layers, each featuring 128 filters. A kernel size of 3 × 3 is applied to each convolutional layer, followed by the activation function rectified linear unit and batch normalization [20]. Frequency and temporal pooling are performed using a max pooling layer with sizes of [[1, 5], [1, 2], [1, 2]], respectively. To mitigate overfitting, a dropout rate of 0.2 is applied after each layer. This is followed by the recurrent neural network (RNN) block, consisting of a single layer of 32 bidirectional gated recurrent units (Bi-GRUs) [21].

3.2. Proposed architecture

In this study, we incorporate large-scale PANNs [7] into our approach due to resource limitations. The PANNs have been pretrained on the extensive Audioset dataset, which consists of 5000 hours of audio spanning 527 sound classes. By leveraging the preexisting knowledge encoded in these pretrained models, we aim to replace the CNN component of the baseline model with PANNs, thereby benefiting from their learned representations and features. The PANNs architecture comprises 6 convolutional blocks, with each block consisting of 2 convolutional layers using a 3 × 3 kernel size. In our study, we investigate the extraction of embeddings after each convolutional block within the PANNs model. These embeddings are subsequently inputted into a single-layer Bi-GRU containing 256 hidden units. The complete CRNN model, encompassing both the PANNs and Bi-GRU components, is unfrozen and trained throughout the experimentation process. Additionally, to include the two supplementary tasks outlined in Section 2.1, we uphold the CNN component as the shared element across all tasks. Furthermore, we integrate distinct Bi-GRUs and output layers for each distinct task, as depicted in Figure 1, each sized at (200×17), (200×5), and (200×2) for the SED, ASC, and AEC branches, respectively.

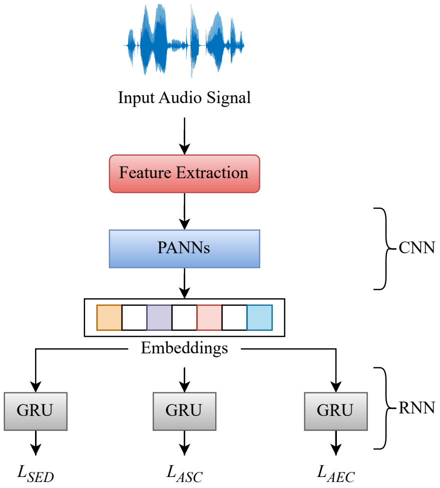


Figure 1: The proposed MTL framework with PANNs along with three parallel Bi-GRUs for different tasks.

4. EXPERIMENTAL SETUP

4.1. Dataset

This study utilizes the multi-annotator estimated strong (MAESTRO)-Real dataset [18] released for the DCASE 2023 Task 4B. The dataset comprises 49 real-life audio files captured from 5 distinct acoustic scenes and includes corresponding annotation outcomes. The total duration of the dataset amounts to 189 minutes and 52 seconds. The audio files are a subset of the TUT Acoustic Scenes 2016 dataset and encompass five acoustic scenes: cafe/restaurant, city center, grocery store, metro station, and residential area. Each scene consists of 6 classes, with some classes being common across all scenes, resulting in a total of 17 classes as presented in Table 1. The dataset consists of the following components: (1) audio recordings comprising the 49 real-life recordings, each ranging from 3 to 5 minutes in length, and (2) soft labels representing estimated strong labels with a time resolution of 1s obtained through crowdsourced data, with values ranging between 0 and 1 indicating the certainty of the annotators. The soft labels follow a format that includes the start time, end time, textual label, and a corresponding value indicating the soft label for each event class within the given segment. For example: “2 3 car 0.9”, “2 3 footsteps 0.7”, and so on.

4.2. Feature extraction and training

For the baseline system [17], a batch size of 32 is employed, and the input features are mel-band energies extracted using a hop length of 200 ms and 64 mel filter banks. Additionally, we explored different front-end feature extraction techniques such as mel-frequency cepstral coefficient (MFCC), linear frequency cepstral coefficient (LFCC), and constant-Q transform (CQT) to replace the log-mel spectrogram. The DCASE 2023 Task 4B dataset is organized according to a 5-fold cross-validation setup, where around 70% of the data per class is allocated for training, and the remaining portion is dedicated to testing. To optimize the training process, we employ the Adam [22] optimizer, with an initial learning rate of 0.001. The training process is executed over a total of 150 epochs, utilizing the computational power of the Nvidia RTX A4000.

4.3. Evaluation

In this study, we utilize the macro-average segment-F1 score $(F1_{MO})$ under the optimum threshold [23] as our primary evaluation metric. It is calculated over 1s segments, following the same approach as the DCASE 2023 Task 4B challenge. The $F1_{MO}$ score considers the best F1 score per class achieved with a class-specific threshold. Additionally, we report the micro-average F1 score $(F1_m)$, micro-average error rate (ER_m), and macro-average F1 score $(F1_M)$ calculated over 1s segments using a decision threshold of 0.5 applied to the system output.

5. EXPERIMENTAL RESULTS

In this part, we present outcomes of the proposed methods, including ablation studies on the DCASE 2023 Task 4B validation set.

5.1. Architecture with feature extraction

We first present the outcome in Table 3 obtained for the baseline as reported by the organizers of the DCASE 2023 Task 4B. The baseline incorporates the log-mel spectrogram with the configuration specified in Section 4.2. Subsequently, we substitute the baseline architecture with the proposed architecture described in Section 3.2, which utilizes PANNs. When using PANNs, we extract the embeddings after the 6th block. Our observations show that employing PANNs with log-mel spectrogram alone enhances the $F1_{MO}$ score from 42.8 to 45.4 as represented in Table 3. The following analysis compares various commonly employed feature extraction methods discussed in Section 4.2. Our findings reveal that the MFCC-based feature extraction method outperforms the log-mel spectrogram utilized in the baseline, as well as the LFCC and CQT front-ends. It improves the $F1_{MO}$ score for the 6-blocks-based PANNs to 46.5 from 45.4. Having determined MFCC as the chosen feature extraction method, we proceed to explore the layer from which we extract the embeddings. We decrease it from the 6th Block to the 3rd Block and conduct experiments accordingly. Through this analysis, we discover that extracting embeddings after the 4th Block yields the most significant improvement in the $F1_{MO}$ score, increasing it from 46.5 to 48.2.

5.2. WSL loss function

Once we determine that the highest score is achieved by extracting embeddings after the 4th Block, we introduce the WSL loss function, as outlined in Section 5.2. The loss function prioritizes the learning of well-defined patterns while minimizing the influence of ambiguous or noisy instances. Consequently, this enhancement leads to an improvement in the $F1_{MO}$ score, increasing it from 48.2 to 48.9.

5.3. MTL framework

To enhance our system, we introduce the MTL framework comprising two auxiliary branches in addition to the primary SED branch. In an ablation study, we compare the performance of the proposed system (PANNs+WSL) by incorporating different MTL branches. Initially, we integrate only the ASC branch with the SED branch with (α=0.85, β=0.15, and γ=0). After tuning the weights in the loss function, this configuration achieves the highest $F1_{MO}$ score of 49.2. Next, we replace the ASC branch with the AEC branch (α=0.85, β=0, and γ=0.15), which results in a $F1_{MO}$ score of 49.0. Finally, we introduce all three branches, including the SED, ASC, and AEC branches, with tuned hyperparameters (α=0.85, β=0.1, and γ=0.05). This configuration yields the best overall score of 49.3, demonstrating the effectiveness of the MTL framework and the impact of each auxiliary branch.

Table 3: Comparison of performance, showing the impact of architectural changes and variations in feature extraction methods.

System	Blocks	Feature	ER_m	$F1_m$	$F1_M$	$F1_{MO}$
Baseline	-	Log-mel	0.487	70.34	35.83	42.8
PANNs	6 Blocks	Log-mel	0.442	72.64	36.97	45.4
PANNs	6 Blocks	CQT	0.493	67.53	31.84	42.0
PANNs	6 Blocks	LFCC	0.447	71.5	31.75	46.0
PANNs	6 Blocks	MFCC	0.415	74.18	34.33	**46.5**
PANNs	5 Blocks	MFCC	0.410	75.1	37.21	48.0
PANNs	4 Blocks	MFCC	0.408	76.74	39.42	**48.2**
PANNs	3 Blocks	MFCC	0.470	73.5	39.35	46.2

Table 4: Illustration of performance improvement following the implementation of the WSL loss function.

System	Feature	ER_m	$F1_m$	$F1_M$	$F1_{MO}$
Baseline	Log-mel	0.487	70.34	35.83	42.8
PANNs (4 Blocks) + WSL	MFCC	0.416	75.61	38.60	**48.9**

Table 5: Ablation study for analyzing the contribution of each branch.

System	MTL	ER_m	$F1_m$	$F1_M$	$F1_{MO}$
Baseline	-	0.487	70.34	35.83	42.8
PANNs (4 Blocks) + WSL	SED + ASC	0.416	76.33	39.65	49.2
PANNs (4 Blocks) + WSL	SED + AEC	0.412	76.29	40.85	49.0
PANNs (4 Blocks) + WSL	SED + ASC + AEC	0.406	76.61	39.87	**49.3**

5.4. System comparison

Our experiments come to a close as we present the results of comparing our system with other high-performing submissions for DCASE 2023 Task 4B. Table 6 displays the reported performances of the baseline system as well as other systems, sorted based on the $F1_{MO}$ score. We observe that our system achieves a performance comparable to other systems while demonstrating an improvement of 14.9% over the baseline system. Additionally, it is worth noting that our system outperforms the 3rd system [24] in all metrics besides the $F1_{MO}$ score.

Table 6: Performance comparison of our proposed system with other submissions in DCASE 2023 Task 4B.

System	ER_m	$F1_m$	$F1_M$	$F1_{MO}$
Xu-SJTU-task4b-3 [25]	0.246	86.13	57.91	69.85
Bai-JLESS-task4b-4 [26]	0.360	78.63	42.45	56.16
Liu-SRCN-task4b-2 [24]	0.430	72.90	28.80	49.70
PANNs (4 Blocks) + WSL + MTL **(Ours)**	0.406	76.61	39.87	**49.30**
Nhan-VNUHCMUS-task4b-1 [27]	0.450	72.43	37.32	46.71
Min-KAIST-task4b-1 [28]	0.445	72.78	36.12	45.81
Cai-NCUT-task4b-1 [29]	0.439	74.84	39.57	43.50
Baseline [17]	0.487	70.34	35.83	42.8

6. CONCLUSION

In this study, we present our methods for sound event detection using soft labels introduced in DCASE 2023 Task 4B. We propose several novel approaches and demonstrate their effectiveness through our findings. Firstly, we suggest using PANNs embeddings and modifying the feature extraction process. Secondly, we propose a weighted soft label (WSL) loss function. Lastly, we incorporate an MTL framework with auxiliary branches for ASC and AEC tasks, enhancing the performance of the primary SED task through joint training. In the future, we intend to explore making task weights adaptive rather than relying on hyperparameter tuning.

7. REFERENCES

[1] A. Harma, M. McKinney, and J. Skowronek, "Automatic surveillance of the acoustic activity in our living environment," *IEEE International Conference on Multimedia and Expo*, pp. 634–637, 2005.

[2] M. Crocco, M. Cristani, A. Trucco, and V. Murino, "Audio surveillance: A systematic review," *ACM Computing Surveys (CSUR)*, pp. 1–46, 2016.

[3] T. Khandelwal, R. K. Das, and E. S. Chng, "Is your baby fine at home? Baby cry sound detection in domestic environments," *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pp. 275–280, 2022.

[4] J. P. Bello, C. Silva, O. Nov, R. L. Dubois, A. Arora, J. Salamon, C. Mydlarz, and H. Doraiswamy, "SONYC: A system for monitoring, analyzing, and mitigating urban noise pollution," *Communications of the ACM*, pp. 68–77, 2019.

[5] J. P. Bello, C. Mydlarz, and J. Salamon, "Sound analysis in smart cities," *Springer International Publishing*, pp. 373–397, 2018.

[6] C. Debes, A. Merentitis, S. Sukhanov, M. Niessen, N. Frangiadakis, and A. Bauer, "Monitoring activities of daily living in smart homes: Understanding human behavior," *IEEE Signal Processing Magazine*, pp. 81–94, 2016.

[7] Q. Kong, Y. Cao, T. Iqbal, Y. Wang, W. Wang, and M. D. Plumbley, "PANNs: Large-scale pretrained audio neural networks for audio pattern recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, pp. 2880–2894, 2020.

[8] T. Khandelwal, R. K. Das, A. Koh, and E. S. Chng, "Leveraging audio-tagging assisted sound event detection using weakified strong labels and frequency dynamic convolutions," *IEEE Statistical Signal Processing Workshop*, 2023.

[9] ——, "FMSG-NTU submission for DCASE 2022 Task 4 on sound event detection in domestic environments," *Detection and Classification of Acoustic Scenes and Events (DCASE) Challenge*, 2022.

[10] Y. Gong, Y.-A. Chung, and J. Glass, "AST: Audio spectrogram transformer," *Interspeech*, pp. 571–575, 2021.

[11] S. Chen, Y. Wu, C. Wang, S. Liu, D. Tompkins, and F. Wei, "BEATs: Audio pre-training with acoustic tokenizers," 2022. [Online]. Available: https://arxiv.org/abs/2212.09058

[12] A. Mesaros, T. Heittola, and A. Klapuri, "Latent semantic analysis in sound event detection," *European Signal Processing Conference*, pp. 1307–1311, 2011.

[13] K. Imoto and S. Shimauchi, "Acoustic scene analysis based on hierarchical generative model of acoustic event sequence," *IEICE Transactions on Information and Systems*, pp. 2539–2549, 2016.

[14] K. Imoto and N. Ono, "Acoustic topic model for scene analysis with intermittently missing observations," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, pp. 367–382, 2019.

[15] H. Bear, I. Nolasco, and E. Benetos, "Towards joint sound scene and polyphonic sound event recognition," *Interspeech*, pp. 4594–4598, 2019.

[16] N. Tonami, K. Imoto, M. Niitsuma, R. Yamanishi, and Y. Yamashita, "Joint analysis of acoustic events and scenes based on multi-task learning," *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pp. 338–342, 2019.

[17] I. Martín-Morató, M. Harju, P. Ahokas, and A. Mesaros, "Strong labeling of sound events using crowdsourced weak labels and annotator competence estimation," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023.

[18] I. Martín-Morató and A. Mesaros, "Strong labeling of sound events using crowdsourced weak labels and annotator competence estimation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, pp. 902–914, 2023.

[19] T. Khandelwal and R. K. Das, "A multi-task learning framework for sound event detection using high-level acoustic characteristics of sounds," *Interspeech*, 2023.

[20] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating deep network training by reducing internal covariate shift," *International Conference on Machine Learning (ICML)*, pp. 448–456, 2015.

[21] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder–decoder for statistical machine translation," *Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1724–1734, 2014.

[22] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *International Conference on Learning Representations (ICLR)*, 2015.

[23] J. Ebbers, R. Haeb-Umbach, and R. Serizel, "Threshold-independent evaluation of sound event detection scores," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1021–1025, 2022.

[24] Y. Jin, M. Chen, J. Shao, Y. Liu, B. Peng, and J. Chen, "DCASE 2023 challenge Task 4 technical report," *Detection and Classification of Acoustic Scenes and Events (DCASE) Challenge*, 2023.

[25] X. Xuenan, M. Ziyang, Y. Fei, Y. Guanrou, W. Mengyue, and C. Xie, "Sound event detection by aggregating pretrained embeddings from different layers," *Detection and Classification of Acoustic Scenes and Events (DCASE) Challenge*, 2023.

[26] H. Yin, J. Bai, S. Huang, and J. Chen, "How information on soft labels and hard labels mutually benefits sound event detection tasks," *Detection and Classification of Acoustic Scenes and Events (DCASE) Challenge*, 2023.

[27] T.-D. Nhan, B. Param, and Y. Zhang, "Sound event detection with soft labels using self-attention mechanisms for global scene feature extraction," *Detection and Classification of Acoustic Scenes and Events (DCASE) Challenge*, 2023.

[28] D. Min, H. Nam, and P. Yong-Hwa, "Application of spectro-temporal receptive field for DCASE 2023 challenge Task 4B," *Detection and Classification of Acoustic Scenes and Events (DCASE) Challenge*, 2023.

[29] H. Zhang, L. Zuo, J. Chen, X. Cai, and M. Wu, "Sound event detection based on soft label," *Detection and Classification of Acoustic Scenes and Events (DCASE) Challenge*, 2023.

EVENT CLASSIFICATION WITH CLASS-LEVEL GATED UNIT USING LARGE-SCALE PRETRAINED MODEL FOR OPTICAL FIBER SENSING

Noriyuki Tonami[1], *Sakiko Mishima*[1], *Reishi Kondo*[1], *Keisuke Imoto*[2], *Tomoyuki Hino*[1]

[1]NEC Corporation, Japan, [2]Doshisha University, Japan

ABSTRACT

Optical fiber sensing is a technology in which sounds, vibrations, and temperature are detected using an optical fiber; especially the sounds/vibrations-aware sensing is called distributed acoustic sensing (DAS). DAS has the potential to capture various types of sounds and/or vibrations in wide areas, e.g., the ground, the sea, and a city area, in our everyday life. To precisely recognize the various types of events, e.g., whale calls, car horns, and wind, by DAS, therefore two problems. First, there is little publicly available data and few pretrained models for the various types of events. Second, the signal-to-noise ratio (SNR) of DAS data is lower than that of other sensor data, such as microphone data, because of optical noise and low sensitivity of DAS. To tackle the lack of DAS data, we first demonstrate a DAS simulation method where DAS observations are simulated by exploiting a microphone simulation. We then propose a method of event classification for DAS utilizing a pretrained audio recognition model, where none of the DAS data are used for training. Moreover, we advocate a class-level gated unit with the pretrained model to overcome the poor classification performance caused by the low SNR of the DAS data. In the proposed method, class probabilities, which are the output of the pretrained model, are employed for controlling priors of DAS, such as events of interest or optical noise. Directly controlling the class probabilities, which are non-black-box values, as priors enables us to utilize not only a pretrained model but also powerful human knowledge. To verify the performance of the proposed method, we conduct event classification, where we simulate observed signals by DAS with the ESC-50 dataset. Experimental results show that the accuracy of the proposed method is improved by 36.75 percentage points compared with that of conventional methods.

Index Terms**—** Optical fiber sensing, distributed acoustic sensing, deep neural network, sound event classification

1. INTRODUCTION

Optical fiber sensing is the detection of sounds and/or vibrations using an optical fiber [1, 2], which is known as distributed acoustic sensing (DAS) or phase-sensitive optical time domain reflectometry (ϕ-OTDR). Optical fiber sensing, including DAS, is superior to other methods in terms of the scalability of the sensing area, the electromagnetic resistance, and the usability of existing optical fibers. Because of these features, DAS is widely used for various applications, especially detecting vibrations, such as whale call detection [3], structural health monitoring [4], seismic activity monitoring [5], border monitoring [6], and pole localization [7]. Owen *et al.* [6] introduced the DAS-based system for distinguishing people, vehicles, and flying objects. Waagaard *et al.* [8] proposed large-scale sensing over 171 km using DAS. Ip *et al.* [1, 2] discussed the use of telecom cables for sensing sounds and/or vibrations.

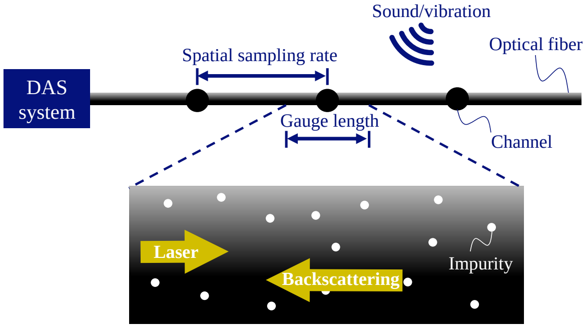


Figure 1: Overview of DAS system

Recently, deep neural networks (DNNs) have been in the spotlight in the field of DAS [7, 9, 10, 11]. DNN-based methods require a large amount of training data to achieve the high performance of DAS. To alleviate this limitation, Zhao *et al.* [10] proposed a data augmentation of DAS and its application to seismic data. Generative adversarial network (GAN)-based methods [11] have been studied to produce training data and are designed for seismic applications.

DAS has two problems in the precise recognition of various types of events, such as whale calls, dog barking, and footsteps. First, there is little publicly available data and few pretrained models for analyzing various types of event classes in the field of DAS. On the other hand, in communities where acoustic signal processing and statistical methods have been studied for various types of sounds [12], various types of dataset or pretrained model [13, 14, 15] are available. Second, the signal-to-noise ratio (SNR) is lower in DAS compared with methods using other sensors, such as microphones. The lower SNR of DAS is caused mainly by optical noise and the low sensitivity of DAS.

To address the lack of DAS data, we first demonstrate a DAS simulation method where a simulator of microphones is utilized for simulating DAS observations. We then propose an event classification method of DAS using a pretrained audio recognition model trained by microphone data. Moreover, we introduce a class-level gated unit with the pretrained model to tackle the problem of the low SNR of DAS data. In the proposed method, posteriors of the pretrained model are employed to control prior information, i.e., events of interest or optical noise, which can be directly manipulated by humans in the inference stage. The proposed gated unit that directly controls the probabilities of event classes, which are non-black-box values, enables us to utilize not only the pretrained model but also human knowledge.

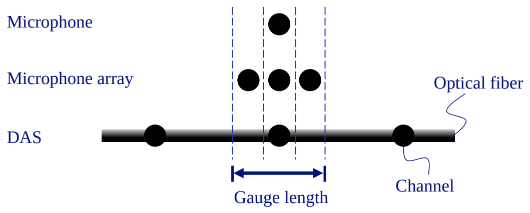


Figure 2: Comparison of DAS and microphone

2. DISTRIBUTED ACOUSTIC SENSING

2.1. Principle of DAS

In DAS, a sensor captures the phase change of a Rayleigh backscattering light wave. Figure 1 shows an overview of the DAS system. The backscattering is triggered by a coherent laser that collides with impurities in the optical fiber. The phase change of the backscattering is proportional to the optical fiber stretching [16], that is, sounds or vibrations that propagate through the optical fiber.

In DAS, the stretching of the optical fiber is measured over the gauge length (GL) L. The total phase change $\Delta\phi$ in L [17] is

$$\Delta\phi = \int_{-\frac{L}{2}}^{\frac{L}{2}} \epsilon(x)\, dx, \tag{1}$$

where $\epsilon(x)$ indicates the strain, that is, the observed sounds and/or vibrations along the optical fiber, at position x of the optical fiber. The point is that the observed signals depend on L. In general, larger L suppresses optical noise, although it distorts the observed signal of sounds or vibrations. An optical fiber sensor with GL of L is also interpreted as a linear sensor array in Eq. 1 where the directivity of the angle of a source signal and its distortion [18]. As can be seen in Fig. 1, arbitrary multiple sensing points can be set along the optical fiber in accordance with a predefined spatial sampling rate.

2.2. Difference between DAS and acoustical microphones

There are two main differences between DAS and microphones: optical noise and GL. The differences make the SNR of DAS data lower compared with that of microphone data. In the first difference, optical noise [19, 20], shot noise [2] is dominant because of the randomness of photons. The second difference is the idea of the gauge. A larger GL distorts signals observed by DAS. On the other hand, a smaller GL, i.e., $\lim_{L\to 0}\Delta\phi$, approximates a point sensor such as a microphone. As can be seen in Fig. 2, DAS data within the GL is thus regarded as a microphone array where channels are densely distributed. Note that the effects of the GL are evident in a single channel of DAS data, unlike a microphone array.

3. PROPOSED METHOD

In this section, we first introduce a simulation method of DAS observation to address the lack of DAS evaluation data for analyzing various types of events. Second, the event classification method for DAS utilizing the pretrained audio recognition model and the class-level gated unit are proposed.

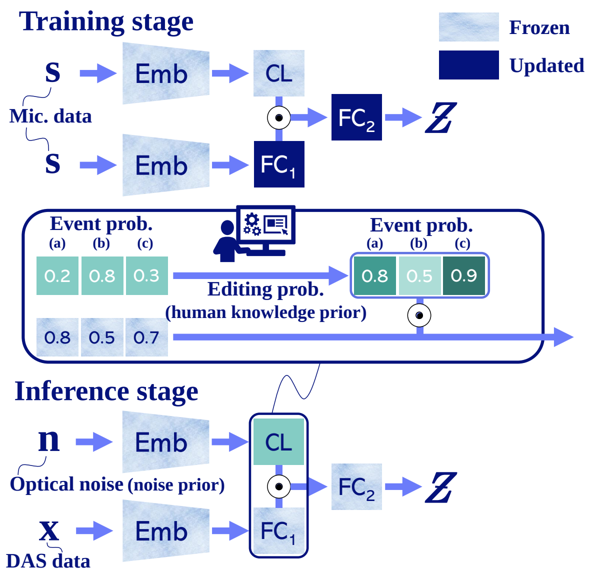


Figure 3: Overview of proposed method

3.1. DAS simulation utilizing microphone simulation

To simulate DAS, we employ multiple simulated microphones, i.e., a microphone array. This is because the sound propagation and the GL are easily implemented using tools of a microphone array simulation, e.g., pyroomacoustics [21] where distance attenuation, reverberation, or directivity is easily simulated. We simply take an average of signals observed by microphones of M channels.

$$\tilde{\mathbf{s}}_{m'} = \frac{1}{M}\sum_{m=0}^{M-1}\mathbf{s}_m. \tag{2}$$

Here, $\tilde{\mathbf{s}}_{m'} \in \mathbb{R}^T$ indicates a DAS signal of the m'-th channel with T temporal frames. $\mathbf{s}_m \in \mathbb{R}^T$ denotes a signal captured by a microphone m of the array. The average of the multiple channels corresponds to L in Eq. 1. $\tilde{\mathbf{s}}_{m'}$ is distorted by the GL, i.e., no-delay-and-sum operation. As an example referring to Fig. 2, there is a single channel data value of DAS for each observed data of three microphones ($M = 3$).

The signal detected by DAS is reportedly expressed as [22]

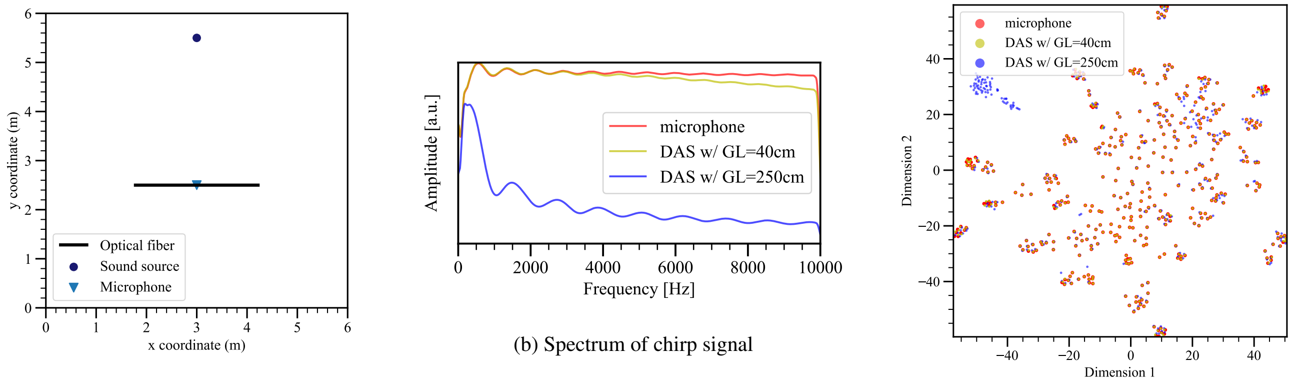
$$\mathbf{x} = \tilde{\mathbf{s}}_{m'} + \mathbf{n}, \tag{3}$$

where $\mathbf{n} \in \mathbb{R}^T$ represents the noise signal. $\mathbf{x} \in \mathbb{R}^T$ is the noisy signal that is corrupted by the GL and the noise signal. When $\mathbf{n}$ is the shot noise of optical noise, it follows the Gaussian distribution $\mathcal{N}(\mu, \sigma)$.

3.2. Event classification of DAS with pretrained audio recognition model and class-level gated unit

To precisely classify various types of events under a lower SNR condition of DAS data, we propose the event classification method of DAS with the class-level gated unit utilizing the pretrained audio recognition model.

[Training stage] In the proposed method, only the data and pretrained model trained with the microphone data are used for the training to tackle the lack of DAS data. Figure 3 shows an overview



(a) Coordinates of simulation

(b) Spectrum of chirp signal

(c) Visualization of embedding by pretrained model

Figure 4: (a) Coordinates of simulation and (b), (c) results of preliminary experiments

of the proposed method. Given a pretrained audio recognition model of the microphone, its deep feature extractor $\mathsf{Emb} : \mathbb{R}^T \to \mathbb{R}^D$ and the classification layer $\mathsf{CL} : \mathbb{R}^D \to [0,1]^E$ are defined. The posterior for event classes is then

$$\mathcal{P} = \mathsf{CL}\big(\mathsf{Emb}(\mathcal{A})\big)\ |_{\mathcal{A}=\mathbf{s}}, \tag{4}$$

where T, D, and E represent the temporal length of an audio clip, the number of dimensions of embedding, and the number of event classes of the pretrained model, respectively. $\mathcal{A}$ denotes an arbitrary signal. When the model is trained with the microphone data, $\mathbf{s}_m \mid \exists m$ is used as $\mathcal{A}$. Given two fully connected layers (FCs) $\mathsf{FC}_1 : \mathbb{R}^D \to \mathbb{R}^E$ and $\mathsf{FC}_2 : \mathbb{R}^E \to \mathbb{R}^{E'}$, the class-level gated unit is

$$\mathcal{Z} = \mathsf{FC}_2\Big(\mathsf{FC}_1\big(\mathsf{Emb}(\mathcal{B})\big) \odot \mathcal{P}\Big)\ |_{\mathcal{B}=\mathbf{s}}\ . \tag{5}$$

$\odot$ and E' indicate elementwise multiplication and the number of target event classes of DAS, respectively. $\mathcal{B}$ denotes an arbitrary signal. Equation 5 is similar to that of a gated linear unit (GLU) [23]. In our method, gating is conducted on the probabilities of event classes of the pretrained audio recognition model, which are non-black-box values.

[Inference stage] In inference stages, DAS data $\mathbf{x}$ are used as $\mathcal{A}$ and $\mathcal{B}$ in Eqs. 4 and 5, respectively. $\mathcal{P}$ can then be edited by human interaction. Directly controlling the interpretable values, i.e., the probabilities of event classes, enables us to utilize human knowledge as an additional prior. Human knowledge can enhance the classification performance with low SNR of DAS data since humans have rich knowledge compared with pretrained audio recognition models. For example, values of event classes of $\mathcal{P}$ unrelated to a recording situation and/or events of interest can be directly masked with zero or a small value:

$$\mathcal{P} = (p_0, \ldots, p_e, \ldots, p_{E-1}), \tag{6}$$

where $p_e \in [0,1]$ is the value of event e of the predefined event classes for the pretrained model and can be directly set by a human. Secondarily, $\mathcal{A}\ |_{\mathcal{A}=\mathbf{n}}$ in Eq. 4 can be used for mitigating the optical noise in the level of the class of the pretrained model in the inference stages:

$$\mathcal{P} = \max\Big(\mathsf{FC}_1\big(\mathsf{Emb}(\mathcal{B})\big) - \mathsf{CL}\big(\mathsf{Emb}(\mathcal{A})\big),\ \mathbf{0}\Big)\Big|_{\mathcal{B}=\mathbf{x},\ \mathcal{A}=\mathbf{n}}, \tag{7}$$

where $\max(\mathbf{a}, \mathbf{b})$ is a function that returns the larger value element of vectors $\mathbf{a}$ and $\mathbf{b}$ in an elementwise manner. $\mathbf{0}$ represents the E dimensional vector where all elements are zero. In Eq. 7, only denoised probabilities of event classes are expected to be passed through. In the inference stages, the softmax function is applied to $\mathcal{Z}$ for obtaining the maximum value of the posteriors of the event classes.

4. EXPERIMENT

4.1. Experimental conditions

[Simulation procedure] To simulate DAS observations, we followed the procedure described in Sec. 3.1. We first simulated a linear microphone array and a sound source, as shown in Fig. 4a, using the pyroomacoustics toolbox [21]. By using pyroomacoustics, we can easily simulate the sound propagation and the idea of the GL. The microphone array consists of 250 channels at intervals of 1 cm. The observed signals of the channels are then averaged using Eq. 2. M was set to 40 (L =40cm) or 250 (L =250cm) in our experiment. Here, the center of the gauge with $M = 40$ matches those with $M = 250$. We finally obtained single channel data of DAS from the signals observed by M microphones using Eqs. 2 and 3. The signal of the sound source was omnidirectionally propagated. In our experiments, we did not simulate any reverberations or revetments of the optical fiber. Moreover, for the shot noise of optical noise, we use Gaussian noise $\sim \mathcal{N}(0,1)$ with variable SNRs.

[Dataset, classification model, and acoustic feature] We used the ESC-50 dataset [13] to evaluate the performance of our methods. ESC-50 comprises 4,000 5-second audio clips with 50 event classes. For the classification model, we used pretrained CNN14 in PANNs [15]. PANNs were trained using AudioSet [14] where event classes are organized in a hierarchy, i.e., ontology. In our experiment, Emb and CL are those of CNN14 with the frozen parameters, where D and E are set to 2,048 and 527, respectively. E' is set to 50,

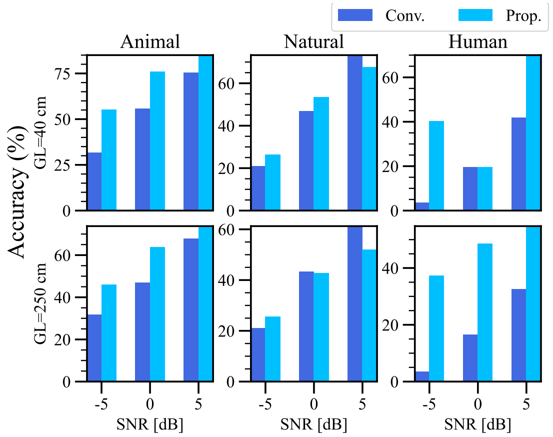


Figure 5: Accuracy (%) of event classification for each target major category with directly controlling gate

which is the number of classes in ESC-50. All clips are downsampled to 32kHz in our experiment. As acoustic features, we used 64-dimensional log-mel energies calculated for every 1,024 sample window and 320 sample hop sizes.

[Training and evaluation] We conducted the 5-fold cross-validation with the ESC-50 dataset. In all experiments, we trained the models using $(M/2)$th-channel signals of the microphone array before all of the channels were averaged. Note that the $(M/2)$th channel is located at the center of the microphone array. This is because we aim to reproduce the situation of the lack of DAS data. In the inference stages, we used only the DAS simulation data, which were averaged with Eqs. 2 and 3. In the training stages, the parameters of only FC_1 and FC_2 were updated by backpropagation using Adam [24]. In all FCs, except for FC_2, ReLU activation was used.

4.2. Experimental results

[Comparison between DAS and microphone data] We first confirm the amplitude spectrum of a chirp signal in the DAS and microphone simulations. Figure 4b shows the 50-to-10,000Hz spectrum. As shown in the figure, GL=40cm does not considerably distort the signal compared with GL=250cm. Furthermore, in Fig. 4c, the embedding vectors of DAS and microphone data obtained by Emb of PANNs [15] are visualized by t-SNE [25]. As can be seen in the figure, there is little difference between the distributions of the microphone and DAS data of GL=40cm. Even when GL is set to 250cm, most of the embeddings of DAS are overlapped with those of the microphone. The results prove that microphone data are similar to DAS data except for optical noise, obtained using large-scale pretrained audio recognition models.

[Event classification with controlling gate] In this experiment, we directly control $\mathcal{P}$ to evaluate the performance of targeting an event of interest. To conduct this experiment, we utilize the overlap between the major categories of ESC-50 and the ontology of AudioSet. In the major categories of ESC-50, we focus on "Animal," "Natural," and "Human," which correspond to "Animal," "Natural sounds," and "Human sounds" of the AudioSet ontology. When events in a major category of ESC-50 are targeted, p_e in Eq. 6 is set to 1; otherwise, 0. For example, when event classes of the major category "Human" of ESC-50 are targeted, p_e in Eq. 6 corresponding to "Human sounds" of the AudioSet ontology is set to 1; otherwise, 0. Figure 5 indicates the results of event classification with Eq. 6 in terms of each major category. "Conv." indicates CNN14 [15] fine-tuned with ESC-50 of microphone data, where the last two FCs were trained, as described in [15]. "Prop." represents the proposed method with directly controlled $\mathcal{P}$. The results show that the proposed method outperformed the conventional method in terms of classification accuracy. In particular, our method improved the accuracy of "Human" by 36.75 percentage points compared with that of the conventional method under the condition of SNR$= -5$dB and GL $= 40$cm. On the other hand, the "Natural" class is misclassified when using the proposed method under some conditions. This is because the "Natural" class, e.g., wind or rain, possibly co-occurs with other classes. In other words, the proposed method with Eq. 6 might discard the information of co-occurrence among event classes.

Table 1: Accuracy (%) of event classification with denoising optical noise

	SNR [dB]					
	-5		0		5	
	GL [cm]					
	40	250	40	250	40	250
Conv.	18.90	15.65	39.00	32.00	59.00	48.45
w/ spectral subtract. [26]	23.60	24.25	**54.65**	37.45	**69.65**	39.50
w/ Wiener filter [27]	21.35	16.65	34.15	23.90	47.35	32.20
Prop. w/ Eq. 7	**32.45**	**27.55**	51.95	**40.05**	64.50	**51.50**

[Event classification with denoising] In this experiment, we evaluated the denoising performance for event classification of DAS data with optical noise signals. Note that none of the DAS data were used for training models to simulate the lack of DAS data. We thus employed non-machine-learning-based denoising methods for the comparison. "Prop." represents the proposed method where $\mathcal{P}$ was produced by Eq. 7 with the optical noise $\mathbf{n}$. Table 1 shows the results of event classification with denoising optical noise. The results reveal that the classification performance is improved when using the proposed method compared with the conventional methods. In particular, the more degraded signals, i.e., lower SNR and/or larger GL, receive greater benefits from the proposed method with Eq. 7. This is because the proposed method does not further distort the signals distorted by the GL, unlike the conventional methods. Moreover, the proposed denoising method can utilize the statistical information of the pretrained model, unlike the conventional methods.

5. CONCLUSION

In this paper, we proposed the event classification of DAS data utilizing the pretrained audio recognition model with the class-level gated unit for accurately classifying various types of events under low SNR conditions without DAS training data. In the proposed method, the class-level outputs of the pretrained model, which are non-black-box values, are employed for controlling priors of DAS data, that is, the optical noise and/or events of interest. This enables us to exploit not only the statistical information of the pretrained model but also human knowledge. To evaluate the performance of the proposed method, we conducted event classification where signals observed by DAS were simulated with the ESC-50 dataset. Experimental results show that the accuracy of event classification by the proposed method is improved by 36.75 percentage points compared with that of the conventional methods.

6. REFERENCES

[1] E. Ip, Y. Huang, M. Huang, M. Salemi, Y. Li, T. Wang, Y. Aono, G. Wellbrock, and T. Xia, "Distributed fiber sensor network using telecom cables as sensing media: Applications," *Proc. Optical Fiber Communications Conference and Exhibition* (*OFC*), pp. 1–3, 2021.

[2] E. Ip, J. Fang, Y. Li, Q. Wang, M. Huang, M. Salemi, and Y. Huang, "Distributed fiber sensor network using telecom cables as sensing media: technology advancements and applications," *Journal of Optical Communications and Networking* (*JOCN*), vol. 14, no. 1, pp. 61–68, 2022.

[3] L. Bouffaut, K. Taweesintananon, H. J. Kriesell, R. A. Rorstadbotnen, J. R. Potter, M. Landro, S. E. Johansen, J. K. Brenne, A. Haukanes, O. Schjelderup, and F. Storvik, "Eavesdropping at the speed of light: Distributed acoustic sensing of baleen whales in the arctic," *Frontiers in Marine Science*, vol. 9, pp. 1–13, 2022.

[4] H. Peter, X. James, Z. Shenghan, D. Matthew, L. Linqing, S. Kenichi, P. Carlo, Z. Christian, M. Demetrio, F. Fabio, L. Francisco, and M. Chris, "Dynamic structural health monitoring of a model wind turbine tower using distributed acoustic sensing (DAS)," *Journal of Civil Structural Health Monitoring*, vol. 11, pp. 833–849, 2021.

[5] T. Parker, S. Shatalin, and M. Farhadiroushan, "Distributed acoustic sensing a new tool for seismic applications," *First Break*, vol. 32, no. 2, pp. 61–69, 2014.

[6] A. Owen, G. Duckworth, and J. Worsley, "Optasense: Fibre optic distributed acoustic sensing for border monitoring," *Proc. european Intelligence and Security Informatics Conference*, pp. 362–364, 2012.

[7] Y. Lu, Y. Tian, S. Han, E. Cosatto, S. Ozharar, and Y. Ding, "Automatic fine-grained localization of utility pole landmarks on distributed acoustic sensing traces based on bilinear resnets," *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing* (*ICASSP*), pp. 4675–4679, 2021.

[8] O. Waagaard, E. Rønnekleiv, A. Haukanes, F. Stabo-Eeg, D. Thingbø, S. Forbord, S. Aasen, and J. Brenne, "Real-time low noise distributed acoustic sensing in 171km low loss fiber," *OSA Continuum*, vol. 4, no. 2, pp. 688–701, 2021.

[9] L. Shiloh, A. Eyal, and R. Giryes, "Deep learning approach for processing fiber-optic das seismic data," *Proc. International Conference on Optical Fiber Sensors*, pp. 1–4, 2018.

[10] Y. Zhao, Y. Li, and N. Wu, "Data augmentation and its application in distributed acoustic sensing data denoising," *Geophysical Journal International*, vol. 228, no. 1, pp. 119–133, 2021.

[11] A. Venketeswaran, N. Lalam, J. Wuenschell, P. Ohodnicki, M. Badar, K. Chen, P. Lu, Y. Duan, B. Chorpeningand, and M. Buric, "Recent advances in machine learning for fiber optic sensor applications," *Advanced Intelligent System*, vol. 4, no. 1, pp. 1–24, 2022.

[12] https://dcase.community/.

[13] J. Piczak, "ESC: Dataset for environmental sound classification," *Proc. the 23rd Annual ACM Conference on Multimedia* (*ACMM*), pp. 1015–1018, 2015.

[14] J. Gemmeke, D. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. Moore, M. Plakal, and M. Ritter, "Audio set: An ontology and human-labeled dataset for audio events," *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing* (*ICASSP*), pp. 776–780, 2017.

[15] Q. Kong, Y. Cao, T. Iqbal, Y. Wang, W. Wang, and M. Plumbley, "PANNs: Large-scale pretrained audio neural networks for audio pattern recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing* (*TASLP*), vol. 28, pp. 2880–2894, 2020.

[16] Z. Zhong, F. Wang, M. Zong, Y. Zhang, and X. Zhang, "Dynamic measurement based on the linear characteristic of phase change in ϕ-otdr," *IEEE Photonics Technology Letters*, vol. 31, no. 14, pp. 1191–1194, 2019.

[17] T. Dean, T. Cuny, and A. Hartog, "The effect of gauge length on axially incident p-waves measured using fibre optic distributed vibration sensing," *Geophysical Prospecting*, vol. 65, pp. 184–193, 2016.

[18] W. Li, Y. Chen, Y. Liang, Y. Lu, and Z. Meng, "Directivity dependence of a distributed fiber optic hydrophone on array structure," *Sensors*, vol. 22, no. 6297, pp. 1–12, 2022.

[19] J. Zhou, Z. Pan, Q. Ye, H. Cai, R. Qu, and Z. Fang, "Characteristics and explanations of interference fading of a ϕ -otdr with a multi-frequency source," *Journal of Lightwave Technology*, vol. 31, no. 17, pp. 2947–2954, 2013.

[20] A. Hartog, O. Kotov, and L. Liokumovich, "The optics of distributed vibration sensing," *Proc. Second EAGE Workshop on Permanent Reservoir Monitoring 2013–Current and Future Trends*, pp. 1–5, 2013.

[21] R. Scheibler, E. Bezzam, and I. Dokmanić, "Pyroomacoustics: A python package for audio room simulations and array processing algorithms," *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing* (*ICASSP*), pp. 351–355, 2018.

[22] L. Jiajing, W. Zhaoyong, L. Bin, W. Xiao, L. Luchuan, Y. Qing, Q. Ronghui, and C. Haiwen, "Distributed acoustic sensing for 2d and 3d acoustic source localization," *Optics Letters*, vol. 44, no. 7, pp. 1690–1693, 2019.

[23] Y. Dauphin, A. Fan, M. Auli, and D. Grangier, "Language modeling with gated convolutional networks," *Proc. International conference on machine learning* (*ICML*), pp. 933–941, 2017.

[24] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *Proc. International Conference on Learning Representations* (*ICLR*), 2015.

[25] L. Maaten and G. Hinton, "Visualizing data using t-SNE," *Journal of machine learning research*, vol. 9, no. 11, pp. 1–27, 2008.

[26] M. Berouti, R. Schwartz, and J. Makhoul, "Enhancement of speech corrupted by acoustic noise," *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing* (*ICASSP*), pp. 208–211, 1979.

[27] J. Lim and A. Oppenheim, "All-pole modeling of degraded speech," *IEEE/ACM Transactions on Audio, Speech, and Language Processing* (*TASLP*), vol. 26, no. 3, pp. 197–210, 1978.

AUDIO-CHANGE CAPTIONING TO EXPLAIN MACHINE-SOUND ANOMALIES

Shunsuke Tsubaki[1] *Yohei Kawaguchi*[2], *Tomoya Nishida*[2], *Keisuke Imoto*[1]
Yuki Okamoto[3], *Kota Dohi*[2], *Takashi Endo*[2]

[1]Doshisha University, Faculty of Science and Engineering, Kyoto, Japan
[2]Hitachi, Ltd., Japan, [3]Graduate School of Information Science and Engineering,
Ritsumeikan University, Kyoto, Japan

ABSTRACT

This paper defines the new problem of "audio-change captioning," which describes what has changed between two audio samples. Conventional audio-captioning methods cannot be used to explain such change, and conventional image-change-captioning methods cannot explain the differences in audio samples. To address these issues, we propose a neural-network model for generating sentences that explain how a machine's normal and anomalous sounds changed in relation to each other. We also created a dataset called MIMII-Change by annotating pairs of normal and anomalous samples extracted from MIMII-DG for each type of sound in machine-operation sounds. The experimental results indicate that our model with spatial attention architecture is effective for stationary sounds because it is able to determine changes in global features, while our model with Transformer Encoder architecture is effective for periodic and sudden sounds because it is able to determine temporal dependencies.

Index Terms— Automated audio captioning, Natural language generation, Deep learning

1. INTRODUCTION

Automated audio captioning (AAC) [1] is one of the tasks that has received particular attention in the field of environmental sound analysis (ESA). The purpose of AAC is to automatically generate textual descriptions (captions) of an audio signal. By representing an audio signal with captions, the relationship between acoustic events and acoustic scenes in the audio signal and their respective states can be described. AAC is expected to have practical applications in a variety of areas, such as assisting the hearing-impaired to understand environmental sounds and analyzing sound in video-based security surveillance systems. It can also be used for other fields such as multimedia retrieval [2, 3]. The framework commonly used in AAC is the sequence-sequence encoder-decoder [4], and like many natural-language-processing tasks, Transformer [5] is the predominant model in AAC [6, 7, 8]. Several studies were conducted to improve the performance of caption generation by providing additional information beyond the encoded audio-embedding information to the text decoder [6, 9]. The utility of such semantic guidance has been explored in image and video captioning, achieving better performance [10, 11].

While the purpose of AAC is to describe a single sound, in real-world problem solving, it may be useful to compare two acoustic signals and describe the changes between them. The anomalous sound detection (ASD) [12] system for machine-operation sounds, only informs about the presence of anomalies without specifying what has changed and how. As a result, experts need to verify the detection results and perform additional tasks to determine if repairs are necessary and which components should be repaired. To simplify this process and reduce the workload for experts, we propose representing the differences between normal and anomalous sounds using linguistic information. This approach allows for an efficient analysis of anomalous machine operation sounds, enabling experts to identify the specific changes and alleviate their burden.

Hence, we define the task of describing the change between two audio signals as audio-change captioning, address the task of explaining anomalous sounds in machines, and introduce the task description and learning scheme. It should be noted that in this study, the objective is not to classify anomalous sounds as in traditional ASD, but rather to focus on expressing how they are anomalous.

Change captioning has already been studied in the image domain. It is used to describe what has changed between two image scenes (before/after) using natural language. Jhamtani and Berg-Kirkpatrick [13] used a pixel-difference-based approach to identify regions of change between before and after images. Because images are assumed aligned and that there is always a change between the two images, this approach cannot distinguish relevant changes from distractors, which is data disguised as change such as viewpoint changes. Therefore, to make it more useful for users, Park et al. [14] created a model that distinguishes between distractors, such as viewpoint change or lighting change, and semantically significant changes such as object movement or change. The model was made robust to distractors by using a dual-attention mechanism to identify regions of change between images. Thus, while change captioning has been studied in the image domain and various methods have been proposed, a pixel-difference-based approach, such as Jhamtani and Berg-Kirkpatrick's [13], is not considered effective for the audio domain, which is time-series information. This study is the first attempt at automated audio-change captioning.

We propose a neural-network model for generating change captions from two sounds. The aim is to generate a textual caption of the changes between the audio files and that is as close as possible to the change caption given by a human for the same audio file. As the suitable model architecture differs due to the sound-occurrence interval or section, we divided sound types into three categories in accordance with sound occurrence and used different architectures for our model. We used Transformer Encoder, which is effective in many AAC tasks, and spatial attention, which is also considered effective [14], as model architectures. For stationary sound changes, we employed spatial attention, while for periodic and non-periodic sounds, we employed Transformer Encoder. In addition to the metrics used in Detection and Classification of Acoustic Scenes and Events (DCASE) [15], i.e., BLEU [16],

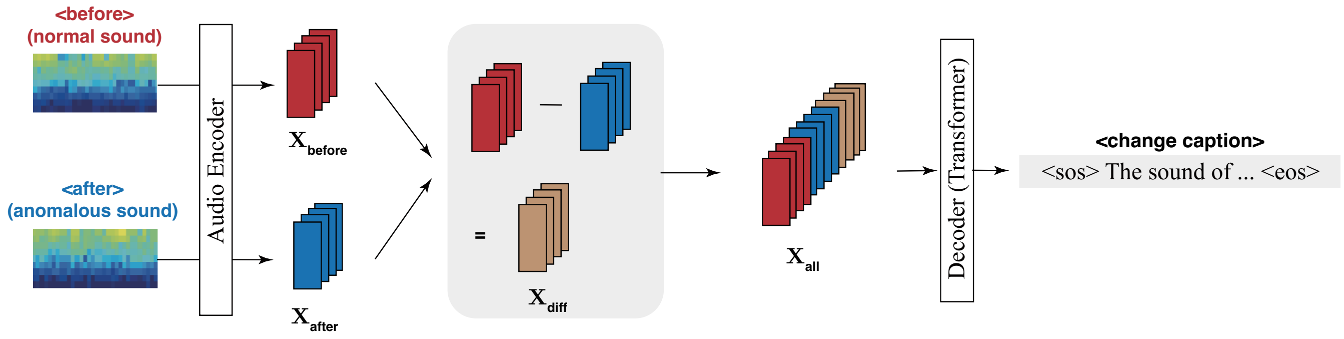


Figure 1: Our proposed audio-change captioning model to explain machine-sound anomalies

METEOR [17], CIDEr [18], and SPIDEr [19], we used Sentence-BERT [20, 21], which is used to evaluate sentence-meaning agreement. Since no suitable dataset for audio-change captioning existed, we created the malfunctioning industrial machine investigation and inspection (MIMII)-Change dataset, which is based on the malfunctioning industrial machine investigation and inspection for domain generalization (MIMII-DG) [22] which was created for anomalous sound detection (ASD) that contains both normal and anomalous sounds of five types of machine-operation sounds (bearing, fan, gearbox, slider, valve). MIMII-Change consists of pairs of normal and anomalous sounds, and each pair is annotated with the changes between these sounds.

2. TASK DESCRIPTION

We now introduce the proposed neural-network model for generating change captions from two sounds. Like many models for explanatory-sentence-generation tasks, it uses a network architecture that encodes variable-length inputs into a fixed-dimension vector and uses this representation to "decode" them into the desired output sentence. Thus, we propose to directly maximize the probability of a correct description given two sounds using the following formulation:

$$\theta^* = \underset{\theta}{\operatorname{argmax}} \sum_{(A_{\text{before}}, A_{\text{after}}, W)} \log p(W \mid A_{\text{before}}, A_{\text{after}}; \theta), \tag{1}$$

where θ are the parameters of our model, A_{before} is an audio before changing, A_{after} is an audio after changing, and W is its correct description. Sentences are generally generated autoregressively from the left (i.e. first word) to the right (i.e. final word). That is, at time step t, the decoder predicts the posterior probability on the vocabulary given the encoded acoustic feature, the start token w_0, and the previously generated words w_1 to w_{n-1}. Thus, $p(W \mid A_{\text{before}}, A_{\text{after}})$ can be formulated as

$$\begin{aligned} &\log p(W \mid A_{\text{before}}, A_{\text{after}}) = \\ &\qquad \sum_{n=0}^{N} \log p(w_n \mid A_{\text{before}}, A_{\text{after}}, w_0, ..., w_{n-1}), \end{aligned} \tag{2}$$

where N is the length of sentence. Note that θ has been removed for convenience. The description-generation process ends when a stop token is generated or the maximum number of generation steps is reached.

3. PROPOSED MODEL

3.1. Training scheme

To analyze the content of a sound clip, it is important to obtain a valid feature representation of the sound clip. We first extract the spectrogram then obtain embedding vectors $X_{\text{all}} \in \mathbb{R}^{(T*2) \times D}$ by using the encoder. This procedure can be formulated as

$$X_{\text{before}}, X_{\text{after}} = \mathcal{E}(A_{\text{before}}, A_{\text{after}}), \tag{3}$$

where ($A_{\text{before}} \in \mathbb{R}^{T \times F}$, $A_{\text{after}} \in \mathbb{R}^{T \times F}$) are the log mel-spectrograms of "before" and "after" sounds, $X_{\text{before}} \in \mathbb{R}^{T \times D}$ and $X_{\text{after}} \in \mathbb{R}^{T \times D}$ are embedding vectors extracted by encoders $\mathcal{E}$, T is the number of time frames, F is the number of mel bins, and D is the dimension of the latent embedding.

We then subtract X_{before} from X_{after} to capture semantic differences in the embedding space. The resulting vector X_{diff} is concatenated with X_{before}. This procedure can be formulated as

$$X_{\text{diff}} = X_{\text{after}} - X_{\text{before}} \tag{4}$$

$$X_{\text{all}} = [X_{\text{before}} : X_{\text{after}} : X_{\text{diff}}], \tag{5}$$

where [:] indicates concatenation.

We used Transformer Encoder [5] and spatial attention [14, 23] as the audio encoders. Spatial attention consists of a two-layer convolutional neural network (CNN) and creates spatial-attention maps $a_{\text{before}}, a_{\text{after}} \in \mathbb{R}^{T \times F}$ a from A_{before}, A_{after}. Thus, spatial attention can localize the change areas between A_{before} and A_{after} and is valid for image-change captioning [14]. For more information on this model architecture, see Park et al.'s study [14].

The decoder predicts the entire caption using X_{all}. The <bos> and <eos> tokens are added before and after the original caption to indicate the beginning and end of the sentence, respectively. The decoder operates in a step-by-step auto-regressive decoding scheme: at the first time step, <bos> is sent to the decoder, then at each time step n, the decoder takes the output word w_{n-1} of the last time step and generates word w_n as the input word in the next time step until <eos>. Finally, the decoder generates a sentence $S = \{w_1, ..., w_N\}$, where w_n is a word and N is the number of words in the sentence. The entire model is trained end-to-end by cross entropy loss. We use a standard transformer [5] as a decoder, which consists of multi-head self-attention on the caption sequence and multi-head encoder-decoder attention on the extracted feature sequence. An overview of our proposed model is given in 1

3.2. Category division

Machine-operating sounds include a variety of sounds, such as regularly occurring and suddenly occurring sounds, and the suitable model architecture differs depending on the type of sound. Regarding changes in regularly occurring sounds, it is considered important to determine the changes in the global characteristics between the two sounds, whereas it is considered important to determine the temporal dependencies between the two sounds for changes in suddenly occurring sounds. Therefore, as proposed method, we divided the sound types into three categories in accordance with the interval of sound occurrence and interval between sound occurrences, and MIMII-Change was created so that a single pair of normal and anomalous sounds had three captions (see Section 4 for more details).

4. MIMII-CHANGE DATASET

Since there is no appropriate dataset to study audio-change captioning, we created MIMII-Change. All sounds are single channel, 10 s in duration, and down-sampled to 16 kHz. We utilized a test dataset from MIMII-DG, consisting of five types of machine sounds (each type consisting of 300 normal sounds and 300 anomalous sounds), and created pairs by assigning one anomalous sound to one normal sound. This resulted in a total of 1,500 pairs (300 pairs × 5 machine types).

Three annotators compared the normal and anomalous sounds of each pair and annotated the changes. The annotators were instructed to always use onomatopoeia when describing sound changes. This is because onomatopoeia, which is a character sequence for phonetically imitating a sound, are effective for describing diverse environmental sound features [24, 25]. Onomatopoeia can be used to describe detailed changes, such as changes in the pitch of a machine's operating sound. The annotator also created three captions for a pair in accordance with the three categories of "stationary sound changes," "periodic sound changes" and "non-periodic sound changes." This is because it is difficult to express all changes in a single sentence, and from a model-learning perspective, it is undesirable for sentences to be redundant. Each of the three categories is defined as follows: "stationary sound": a single sound that occurs continuously for more than about 5 s, "periodic sound": a sound that repeats (including intervals) for more than 5 s, and "non-periodic sound": a sound that occurs multiple times but has no periodicity or appears and disappears suddenly. To improve learning efficiency, annotators provided captions according to templates. Templates mean that, for example, a change in pitch is always described as, "The pitch of ... became higher/lower."

The 1.500 pairs were divided into two 75 and 25% segments, which we call development and evaluation, respectively. All words in the captions must be included in the development split, and there should be no words that are only included in the evaluation split. This prevents the presence of unused words in training (i.e. words that only appear in development) and unknown words in evaluation (i.e. words that do not appear in development). We also split the data so that the word-occurrence frequency in development is always greater than that in evaluation. The number of data items, words, and onomatopoeia after splitting of each category are as listed in Table 1.

Table 1: Number of words of each category

	onomatopoeia/other words	Total
stationary	146 / 68	214
periodic	756 / 107	863
non-periodic	1,155 / 105	1,260

Table 2: Experimental conditions

Optimizer	Adam [26]
Training epoch	100
Batch size	16
GPU	GeForce RTX 3060

5. EXPERIMENTS

5.1. Evaluation metrics

To evaluate audio-change captioning, we used the conventional rule-based evaluation metrics BLEU [16], METEOR [17], CIDEr [18], SPICE [27], and SPIDEr [19]. Most conventional rule-based metrics focus on n-gram or sub-sequence-based matching between candidate and reference captions. CIDEr and SPICE, proposed for image captioning, show better correlation with human judgment in the captioning task. However, they cannot evaluate the semantic similarity between sentences, and they have not yet been able to resemble human evaluation [21]. To address this issue, we used the model-based evaluation metric Sentence-BERT [20, 21]. Sentence-BERT can be used to obtain a fixed-length sentence-embedding vector for input captions. The sentence embeddings are then used to calculate similarities between candidate and reference captions by calculating their cosine similarities. We also used the phoneme error rate (PER) [28] to evaluate onomatopoeia correspondence. Since each onomatopoeia is tokenized, it is not possible to match onomatopoeia with similar constituent phonemes. For example, "gagaga" and "gaga" would be evaluated as completely different onomatopoeia. To address this issue, onomatopoeia were broken down into phonemes according to a previous study [29], and similarity was calculated between onomatopoeia in terms of the PER. The PER is the "edit distance" between two phoneme sequences, normalized by the length of target phonemes, and expressed using Eq. 6. Since the number of onomatopoeia appearing in different sentences may differ, we used the mean phoneme error rate (MPER). The MPER is the average of PER of all combinations of onomatopoeia in a sentence and expressed using Eq. 7, where N is the number of phonemes in a reference caption, M is the number of phonemes in candidate caption, R_n is the n-th onomatopoeia of a no reference caption, and C_m is the m-th onomatopoeia of a candidate caption.

$$\mathrm{L}(R_n, C_m) = \frac{\text{Repalcement Err.} + \text{Insertion Err.} + \text{Deletion Err.}}{\text{Number of Target Phonemes}} \tag{6}$$

$$\text{MPER} = \frac{\sum_{n=1}^{N}\sum_{m=1}^{M} \mathrm{L}(R_n, C_m)}{N * M} \tag{7}$$

Since the PER is calculated for all combinations of onomatopoeia, it is not possible to evaluate onomatopoeia order correspondence,

Table 3: Experimental results

model_type (#model_parameters)	BLEU_3	BLEU_4	METEOR	CIDEr	SPICE	SPIDEr	Sentence-BERT	MPER
Stationary								
TraEnc. (10.8M)	0.616	0.542	0.427	0.969	0.340	0.655	0.793	0.281
SpaAttn. (0.9M)	**0.669**	**0.601**	**0.441**	**1.086**	0.365	**0.726**	**0.796**	0.266
PANNs+TraEnc. (82.6M)	0.659	0.583	0.436	0.933	**0.381**	0.657	0.791	**0.251**
Periodic								
TraEnc. (10.8M)	**0.464**	**0.387**	0.390	**0.946**	**0.255**	**0.601**	0.725	**0.338**
SpaAttn. (0.9M)	0.426	0.354	**0.402**	0.881	0.249	0.565	**0.727**	0.380
PANNs+TraEnc. (82.6M)	0.383	0.306	0.369	0.729	0.213	0.471	0.689	0.362
Non-periodic								
TraEnc. (10.8M)	**0.413**	**0.339**	**0.427**	**1.864**	**0.373**	**1.118**	**0.728**	0.327
SpaAttn. (0.9M)	0.328	0.269	0.411	1.441	0.304	0.873	0.678	**0.321**
PANNs+TraEnc. (82.6M)	0.346	0.284	0.392	1.434	0.331	0.882	0.682	0.365

so the MPER is used only as a metric to measure onomatopoeia agreement in sentences. For example, the PER value of the candidate sentence "A changed to B" and the candidate sentence "B changed to A" would be the same with respect to the reference caption "Changed from A to B." Here, A and B are onomatopoeia.

5.2. Experimental setup

We used the 64-dimensional log mel-band energy as an acoustic feature, which is extracted on the basis of a 64-ms frame length with a 32-ms shift size. Other conditions are listed in Table 2. As this paper presents the first methodology for audio-change captioning, there are no previous results to compare the presented ones. For that reason, several model architectures are compared to investigate their effectiveness.

Transformer Encoder Transformer encoders can determine the temporal dependencies of each input sequence. Therefore, it is considered effective for periodic and non-periodic sound with short sound onset intervals. In this experiment, Transformer Encoder with three layers and four multi-head attention was used.

Spatial attention Spatial attention [23] is an architecture based on convolutional neural networks and it generates a spatial-attention map by using the inter-spatial relationship of features. Spatial attention differs from channel attention in that it focuses on where information is located and has been shown to be effective in locating points of change [14]. In this experiment, spatial attention consisting of a two-layer CNN was used. The spatial attention architecture is able to determine global features, which may be effective for stationary sound.

Acoustic feature extraction with pretrained audio neural networks (PANNs) The effectiveness of transfer learning of pre-trained models has been shown in many audio-related tasks. To confirm the effectiveness of pre-trained models, we used PANNs [30], a pre-trained model for acoustic recognition, as a feature extractor. Specifically, we used a pre-trained 14-layer CNN (CNN14). Acoustic features are extracted from the spectrogram by using PANNs, the outputs X_{before} and X_{after} is subtracted, and X_{all}, calculated in the same manner as Eq. 5, is passed through an encoder.

5.3. Results

Table 3 lists the evaluation results for each version in each of the three categories. All versions used Transformer Decoder as decoder and had different encoders. TraEnc. denotes Transformer Encoder, SpaAttn. denotes spatial attention.

Transformer Encoder vs. spatial attention As shown in Table 3, Spatial attention performed best for "stationary sound changes." As shown in Table 1, the number of words for "stationary sound changes" was 214, which is much smaller than the other categories. For steady sound changes, it is considered important to capture the change in the global features between two sounds. Therefore, spatial attention, which has a relatively easy task difficulty and consists of a two-layer CNN, was more effective. Transformer Encoder was more effective for "periodic sound changes" and "non-periodic sound changes" because the vocabulary was large and it is considered important to capture the temporal dependency between the two sounds.

Validity of PANNs as feature extractor In all three categories, there was no performance improvement due to feature extraction with PANNs. This may be due to the fact that PANNs is trained by solving audio tagging, so features are lost in MIMII-Change in which all sounds are classified as machine-operation sounds.

Our experiments showed that different model architectures were suitable for different categories of sounds with distinct characteristics. Specifically, we found that using spatial attention was effective for the "stationary sound changes," while using Transformer Encoder was effective for the "periodic sound changes" and "non-periodic sound changes."

6. CONCLUSION

We defined a new problem, "audio-change captioning," which describes what has changed between two audio samples and proposed a neural-network model for generating sentences that explain how a machine's normal and anomalous sounds changed in relation to each other. We also created the MIMII-Change dataset that is based on MIMII-DG, annotated each type of sound, and investigated the characteristics of audio-change captioning. Our experiments showed that different categories of sounds with distinct characteristics required different model architectures for optimal performance. By utilizing models tailored to each category of sound, we were able to achieve high accuracy by leveraging the specific features of the sound.

7. REFERENCES

[1] K. Drossos, S. Adavanne, and T. Virtanen, “Automated audio captioning with recurrent neural networks,” in *Proc. of IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, Oct. 2017.

[2] A. S. Koepke, A.-M. Oncescu, J. Henriques, Z. Akata, and S. Albanie, “Audio retrieval with natural language queries: A benchmark study,” *IEEE Transactions on Multimedia*, pp. 1–1, 2022.

[3] X. Mei, X. Liu, J. Sun, M. D. Plumbley, and W. Wang, “On metric learning for audio-text cross-modal retrieval,” *arXiv preprint arXiv:2203.15537*, 2022.

[4] X. Mei, X. Liu, M. D. Plumbley, and W. Wang, “Automated audio captioning: an overview of recent progress and new challenges,” *EURASIP Journal on Audio, Speech, and Music Processing*, no. 1, Oct 2022.

[5] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” *Proc. of Advances in Neural Information Processing Systems (NIPS)*, vol. 30, 2017.

[6] Y. Koizumi, R. Masumura, K. Nishida, M. Yasuda, and S. Saito, “A Transformer-Based Audio Captioning Model with Keyword Estimation,” in *Proc. Interspeech*, 2020, pp. 1977–1981.

[7] Z. Chen, D. Zhang, J. Wang, and F. Deng, “Audio captioning with meshed-memory transformer,” DCASE2021 Challenge, Tech. Rep., 2021.

[8] Y. Koizumi, Y. Ohishi, D. Niizumi, D. Takeuchi, and M. Yasuda, “Audio captioning using pre-trained large-scale language model guided by audio-based similar caption retrieval,” 2020.

[9] Z. Ye, H. Wang, D. Yang, and Y. Zou, “Improving the performance of automated audio captioning via integrating the acoustic and semantic information,” in *DCASE*, 2021.

[10] Q. You, H. Jin, Z. Wang, C. Fang, and J. Luo, “Image captioning with semantic attention,” in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 4651–4659.

[11] J. Yuan, C. Tian, X. Zhang, Y. Ding, and W. Wei, “Video captioning with semantic guiding,” in *IEEE Fourth International Conference on Multimedia Big Data (BigMM)*, 2018, pp. 1–5.

[12] Y. Kawaguchi, K. Imoto, Y. Koizumi, N. Harada, D. Niizumi, K. Dohi, R. Tanabe, H. Purohit, and T. Endo, “Description and discussion on DCASE 2021 challenge task 2: Unsupervised anomalous sound detection for machine condition monitoring under domain shifted conditions,” 2021.

[13] H. Jhamtani and T. Berg-Kirkpatrick, “Learning to describe differences between pairs of similar images,” in *EMNLP*. Association for Computational Linguistics, Oct.-Nov. 2018, pp. 4024–4034.

[14] D. H. Park, T. Darrell, and A. Rohrbach, “Robust change captioning,” in *Proc. of IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 4624–4633.

[15] https://dcase.community/.

[16] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, “BLEU: a method for automatic evaluation of machine translation,” in *Proc. of the 40th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, July 2002, pp. 311–318.

[17] A. Lavie and A. Agarwal, “METEOR: An automatic metric for MT evaluation with high levels of correlation with human judgments,” in *Proc. of the Second Workshop on Statistical Machine Translation*. Association for Computational Linguistics, June 2007, pp. 228–231.

[18] R. Vedantam, C. L. Zitnick, and D. Parikh, “CIDEr: Consensus-based image description evaluation,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2015, pp. 4566–4575.

[19] S. Liu, Z. Zhu, N. Ye, S. Guadarrama, and K. Murphy, “Improved image captioning via policy gradient optimization of spider,” in *Proc. of IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 873–881.

[20] N. Reimers and I. Gurevych, “Sentence-BERT: Sentence embeddings using Siamese BERT-networks,” in *EMNLP-IJCNLP*. Association for Computational Linguistics, Nov. 2019, pp. 3982–3992.

[21] Z. Zhou, Z. Zhang, X. Xu, Z. Xie, M. Wu, and K. Q. Zhu, “Can audio captions be evaluated with image caption metrics?” in *ICASSP*, 2022, pp. 981–985.

[22] K. Dohi, T. Nishida, H. Purohit, R. Tanabe, T. Endo, M. Yamamoto, Y. Nikaido, and Y. Kawaguchi, “MIMII DG: Sound dataset for malfunctioning industrial machine investigation and inspection for domain generalization task,” *In arXiv e-prints: 2205.13879*, 2022.

[23] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, “CBAM: Convolutional block attention module,” in *Proc. of European conference on computer vision (ECCV)*, 2018, pp. 3–19.

[24] Y. Okamoto, S. Horiguchi, M. Yamamoto, K. Imoto, and Y. Kawaguchi, “Environmental sound extraction using onomatopoeic words,” in *ICASSP*, 2022, pp. 221–225.

[25] Y. Okamoto, K. Imoto, S. Takamichi, R. Yamanishi, T. Fukumori, and Y. Yamashita, “Onoma-to-wave: Environmental sound synthesis from onomatopoeic words,” *APSIPA Transactions on Signal and Information Processing*, vol. 11, no. 1, 2022.

[26] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *CoRR*, vol. abs/1412.6980, 2014.

[27] P. Anderson, B. Fernando, M. Johnson, and S. Gould, “SPICE: Semantic propositional image caption evaluation,” in *Proc. of European Conference on Computer Vision (ECCV)*. Springer, 2016, pp. 382–398.

[28] S. Ikawa and K. Kashino, “Generating sound words from audio signals of acoustic events with sequence-to-sequence model,” in *ICASSP*, 2018, pp. 346–350.

[29] https://github.com/KeisukeImoto/RWCPSSD_Onomatopoeia/blob/master/katakana2accphrase.csv.

[30] Q. Kong, Y. Cao, T. Iqbal, Y. Wang, W. Wang, and M. D. Plumbley, “PANNs: Large-scale pretrained audio neural networks for audio pattern recognition,” vol. 28. IEEE, 2020, pp. 2880–2894.

AUTOMATIC DETECTION OF COW VOCALIZATIONS USING CONVOLUTIONAL NEURAL NETWORKS

Ester Vidaña-Vila[1], Jordi Malé[1], Marc Freixes[1], Mireia Solís-Cifré[1], Miquel Jiménez[1], Cristian Larrondo[2,3], Raúl Guevara[2], Joana Miranda[2], Leticia Duboc[1], Eva Mainau[2], Pol Llonch[2], Rosa Ma Alsina-Pagès[1]*

[1] HER – Human-Environment Research, La Salle – Ramon Llull University, Barcelona, ES
[2] AWEC Advisors S.L., Edifici Eureka, Parc de Recerca de la Universitat Autònoma de Barcelona
[3]Center for Applied Research in Veterinary and Agronomic Sciences, Faculty of Veterinary Medicine and Agronomy, Universidad de Las Américas, Chile
* Corresponding author: ester.vidana@salle.url.edu

ABSTRACT

The well-being of animals holds significant importance in our society. Apart from the ethical concerns, recent studies have highlighted the correlation of animal growth, reproductive potential, and overall productivity with animal welfare. In this context, the vocalizations of cows have emerged as a valuable indicator of their well-being for veterinary researchers, but gathering and labelling the vocalizations for their in-depth study is time-consuming and labour-intensive. For this reason, in this work, we present an acoustic event detection algorithm that has been trained and validated with different setups using acoustic data collected from two different farms. The experimental set-up consists of a Convolutional Neural Network followed by a post-processing stage for the detection of vocalizations, so veterinary researchers can easily analyze them. The experimental evaluation assesses the importance of selecting the convenient post-processing and overlapping acoustic window for finding new vocalizations. Furthermore, the study evaluates the significance of using data collected specifically from the same farm for acoustic event detection, as opposed to employing data from a different farm. Results show that by merging training data from different farms, including the farm that is being evaluated, an F1 score of 57.40% and a recall of 74.05% can be achieved.

Index Terms— Acoustic event detection, Cow vocalization, Deep learning, Bioacoustics, Cow monitoring

1. INTRODUCTION

Animal welfare has gained significant importance in our society, both for its ethical consideration and because it can affect animal growth, reproductive potential, and overall productivity [1]. For this reason, society is demanding welfare-monitoring methodologies that do not affect the physical integrity of the animals [2]. Among various animals, cows have gained particular attention from veterinary researchers due to the potential insights that can be gained from monitoring and interpreting their vocalizations (thus, avoiding animal manipulation). This vocal information is key, as it can provide details about the animals' conditions, such as pain, stress and hunger, among others [3, 4].

In order to respond to this need, recent contributions in the field have focused on developing algorithms for both automatically detecting and classifying the vocalizations of cows [3, 5] and analysing them for welfare monitoring [6, 7]. These automatic techniques can help farmers, veterinarians and researchers to gain valuable insights into the conditions and well-being of cows. However, most of these studies were conducted on single farms, which limits the ability to evaluate the performance of the algorithm in various environments and farm setups.

The work presented in this paper tackles this problem; that is, it seeks to develop an algorithm that can detect cow vocalizations in multiple farm environments. This research has been carried out under the umbrella of the project *"CowTalkPro: Desarrollo de un Sensor de Sonido en vacas para evaluar la salud y el bienestar animal"* (in English: Development of a Sound Sensor in cows to assess animal health and welfare.). Its interdisciplinary team is composed of engineers from La Salle Campus Barcelona (Ramon Llull University) and veterinarians and researchers from AWEC Advisors S.L..

The CowtalkPro project aims to create a single sensor that can be deployed in multiple farms—not only one—for real-time monitoring of the welfare of cows. More specifically, this project is concerned with three particular periods in the cows' lives:

First, during the initial weeks of life, monitoring calves can help support their health and, consequently, their wellbeing. If many coughs are heard within a short period, it might indicate that there are sick calves in the yard. For veterinarians and farmers, early detection of respiratory illness in calves is crucial to avoid spreading virus and because late treatment of such conditions could affect the production of that cow in its adult life.

Second, during the dry-offs, which are transitional phases between milk production and their dry phase before the get inseminated, cows may vocalize because they are experimenting pain or discomfort. Detecting these feelings can help cows' welfare by indicating the need to apply pain mitigation actions.

Finally, monitoring cows vocalizations during calving may inform whether the cow needs the assistance of a farmer.

Therefore, the resulting sensor can benefit farmers, veterinarians, and veterinary researchers interested in the assessment and monitoring of cows welfare.

Prior to the development of the sensor, it is important to determine which vocalizations give important insight to determine animal well-being. This normally requires the collection and interpretation of a significant number of cow vocalizations by the veterinary researchers, which is a complex and time-consuming task. In order to support this work, we have developed an automatic detector of vocalizations over audio recordings. The algorithm takes an audio file recorded on a farm and detects two types of sounds: vocalizations and coughs.

At the current stage of the project, the algorithm, described in this paper, focuses on detecting vocalizations for the veterinarian researchers to analyse. More specifically, an acoustic event detection algorithm has been trained and validated using acoustic data collected from two distinct farms, with the aim of improving its adaptability and reliability in monitoring cow vocalizations in real-world scenarios. The presented algorithm utilizes a Convolutional Neural Network (CNN) as the primary detection model, which is then followed by a post-processing stage to refine the results.

The experimental evaluation of our approach encompasses two key aspects: on one hand, we investigate the significance of selecting the appropriate post-processing techniques and overlapping acoustic window for effectively detecting vocalizations. These parameters play a crucial role in uncovering new vocalizations that might otherwise go unnoticed. On the other hand, we explore the implications of using farm-specific data for acoustic event detection, as opposed to employing data from a different farm. This analysis allows us to assess the impact of dataset heterogeneity on the algorithm's performance.

The paper is organised as follows. Section 2 presents the experimental evaluation pipeline. Next, Section 3 details the obtained results. Finally, the conclusions and future work are presented in Section 4.

2. EXPERIMENTAL EVALUATION

This section provides an overview of the experimental evaluation pipeline, which includes the following components: data collection campaigns conducted in two farms, the utilization of a CNN-based algorithm for automated vocalization detection, post-processing techniques employed to determine the onset and offset of each vocalization, and the utilization of data from multiple farms to assess the algorithm's generalization capabilities.

2.1. Data collection

For the experimental evaluation, audio files recorded in two different farms have been used. The first farm is located in Girona (Spain), and the second farm is located in Valencia (Spain). In both cases, a similar recording setup was used. That is, a mains powered audio recorder Zoom H5 [8] placed inside of a box, and connected to an omnidirectional microphone via a long XLR wire (about 30 m). The microphone hung on the ceiling of the cows' yard. An example of set-up is shown in Figure 1. Two microphones were placed on each farm. In Valencia, both microphones were in a big yard for calves, with a separation of about 50 m between them. In Girona, one microphone was over a calves yard and the other one covered dairy cows at the dry-off period.

The hardware set-up was installed in the farm collecting continuous data for about one year. Due to the limitation of the SD card that can be placed on the Zoom recorder, which cannot hold more than 32 GB, and using a sample rate of 44,100 Hz, each week, we have recorded for about four days and a half. After that, the SD card had to be manually replaced.

A small proportion of this audio data has been manually labelled and used for the experiments. Specifically, for this work, the following audio files of 15 min each have been used:

- **Girona:** 40 audio files from cows and 79 audio files from calves.
- **Valencia:** 80 audio files from calves.

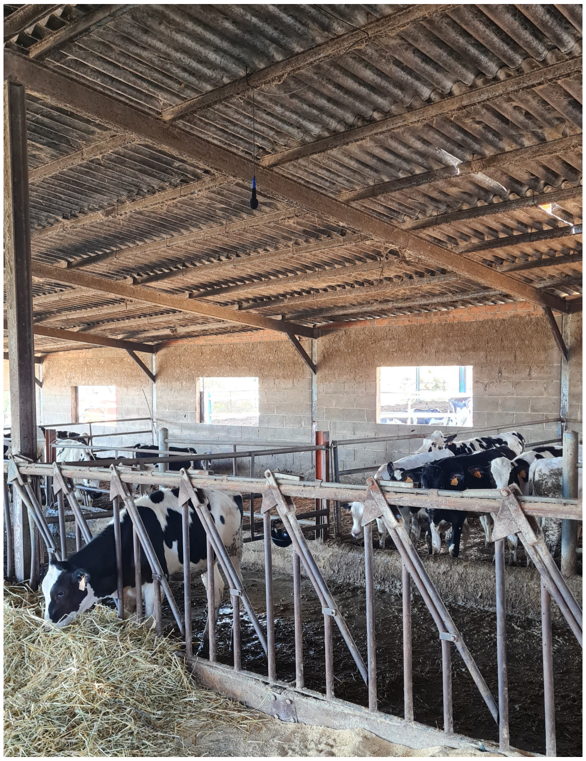


Figure 1: Installation of a microphone over the calves yard in Girona.

This makes a total of 199 files, which represent almost 50 hours of labelled acoustic data. The annotation process was carried out by two different annotators under the supervision of veterinary experts using the Audacity software. The annotation taxonomy had two different categories: vocalizations and coughs.

The test set was built with 20 audio files from Valencia, as this farm has many more calves than Girona — and therefore more vocalizations per audio file.

The remaining audio files were chosen to be used as Training set with different splits, to evaluate whether using data from different farms improves or impairs the metrics of the vocalizations detection model explained in the following subsection.

Set	Farm	Vocalizations	Coughs
Train	Girona	2 289	1 107
Train	Valencia	3 107	1 579
Total train	Both	5 396	2 686
Test	Valencia	1 756	129

Table 1: Amount of labels found in every dataset.

As it can be observed in Table 1, the test set contains 129 cough instances and 1,756 vocalizations. This class imbalance is due to the nature of the audio files, as cows tend to vocalize more than cough, especially when they are not sick.

2.2. Automatic detection of vocalizations

The model used to automatically detect the cow vocalizations is a MobileNet [9] architecture. This model was chosen because its light architecture could be applied in the future to real-time detection of vocalisations on farms using low-cost devices (e.g., Raspberry Pi [10]), as tested by a subset of the authors of this paper in other domains [11], which is the final goal of the CowTalkPro project.

In all experiments, the MobileNet was trained for 15 epochs, using early stopping to obtain the best model (lowest validation loss) out of the 15. As inputs of the CNN, spectrograms were used. In line with previous studies [11], a window size of 1 second was selected to sample the audio file for training.

The CNN was configured as a multilabel classifier, as there might be more than one acoustic event present in a 1-second fragment (e.g., one cow is vocalizing while another cow is coughing). The two possible outputs of the model are vocalizations or coughs.

At the inference stage, the CNN was concatenated with a post-processing algorithm, which is in charge of delimiting the starting and ending point of every vocalization (on-set and off-set times). To achieve this, at inference time, and contrarily to the training stage (in which the audio files were split in windows of 1 second without overlap), the audios were split in overlapping windows.

2.2.1. Post-processing technique

The selection of the overlap time plays a decisive role for an accurate detection of vocalizations. For this reason, we present the classification results for three different overlapping times: 0.1 seconds, 0.25 seconds and 0.5 seconds. Figure 2 illustrates the different overlapping times.

For this experiment, all the data except for the one selected as test set was used for training, meaning that it incorporated data from both farms.

The metrics were calculated using the *"sed_eval" - Evaluation toolbox for Sound Event Detection* [12]. More specifically, segment-based metrics were used, with a configuration of a t_collar of 0.9 and percentage_of_length of 0.1. The first parameter is a tolerance with respect to the ground truth event duration, and the second one is the percentage of the length within which the estimated offset has to be in order to be considered a valid estimation.

2.2.2. Using data from different farms for training

After the previous experiment, and once a convenient post-processing overlapping time was selected, another set of experiments was carried out. In this case, the aim of the experiment was to quantify how the training data affected the results.

Three training sets were configured, each one used for a different experiment:

1. **Experiment 1**: Using the complete dataset of Girona (cows and calves) and the 60 audio files from Valencia that were not used as test set.
2. **Experiment 2**: Using only the dataset from Girona (cows and calves). Therefore, in this experiment, the training set consists of data recorded in a different farm than the one used for testing.
3. **Experiment 3**: Using only the dataset from Valencia. This means that the data used for training comes from the same farm as the data used for testing.

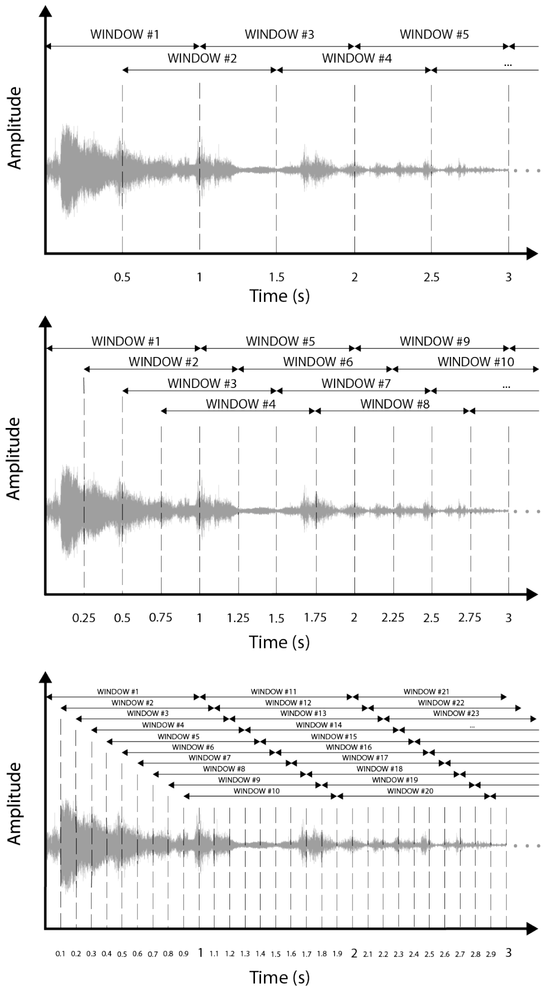


Figure 2: Three different overlaps at inference time to detect vocalizations. On top, an overlap of 0.5 seconds, in the middle, an overlap of 0.25 seconds, and in the bottom, an overlap of 0.1 seconds.

The motivation behind doing these three experiments was to evaluate whether incorporating data recorded in the same farm improve the scores of the classifier.

3. RESULTS

3.1. Post-processing technique

Table 2 shows the results of the experiment regarding the overlapping times. As it can be observed, selecting different overlapping has a huge impact on results. While the F1-score is more or less maintained (achieving its highest value with an overlap of 0.25 s), the Precision and Recall vary substantially. The biggest overlap (0.5 s) results in higher Precision and lower Recall, while the smallest overlap (0.1 s) results in lower Precision and higher Recall.

Having a more precise system means that the number of false positive events is lower. Therefore, the presented results show that a wider overlap filters more false positive events.

Analogously, having a system with a higher Recall suggests that

Overlap	F1-score	Precision	Recall
0.1 s	57.4%	46.87%	**74.05%**
0.25 s	**61.7%**	63.24%	60.24%
0.5 s	53.68%	**77.74%**	40.99%

Table 2: Precision, Recall and F1-score obtained by varying the overlapping window for the vocalizations detection.

there are fewer false negative events (i.e., that fewer vocalizations are missed). A smaller overlap, even if less precise, decrements the number of vocalizations that are mistakenly confused by noise.

As the aim of the presented algorithm is to detect vocalisations that can be further analysed by AWEC veterinary experts, the smallest overlap (0.1 s) was selected to detect all possible vocalisations, even if some of them are false positives that need to be manually removed. Therefore, for the following experiments, the post-processing stage was carried out with the overlap of 0.1 s.

3.2. Using data from different farms for training

Farm training data	F1-score	Precision	Recall
Both Farms	57.4%	46.87%	**74.05%**
Girona	50.58%	41.51%	64.72%
Valencia	**59.25%**	**54.16%**	65.39%

Table 3: F1-score, Precision and Recall of the three experiments.

Three different set-ups were evaluated, using 20 audio files recorded in Valencia as test set. As it can be seen in Table 3, the best F-score (59.25%) is obtained in the experiment that contains only audio files from Valencia. However, the highest Recall (74.05%) was obtained when using audio files from both farms for training.

Nevertheless, the results obtained when using data from Girona only are not very different from those in which Valencia audios are used.

This leads to the conclusion that using audio data from the same farm that is being evaluated is desirable, but not completely necessary to have moderately good results (note that there is a difference of 8,67% of F1-score only between the best and the worst system).

4. CONCLUSIONS

This paper addresses the problem of automatically detecting the vocalizations of cows for further analysis by veterinary researchers, as these vocalizations can be an indicator of their welfare.

The developed algorithm uses a lightweight deep learning architecture that can run over a low-cost platform. Two experiments have been conducted, using data collected from two different dairy farms (Girona and Valencia) and manually labelling it.

The first experiment aimed at determining the optimal overlap time for vocalization detection. It was observed that the chosen overlap time correlated with the Precision and Recall metrics of the system. The system with the highest Recall was achieved when using the smallest overlapping time, resulting in more overlapped windows.

The second experiment focused on assessing the model's ability to generalize and classify vocalizations from different farms. Moderately improved results were observed when utilizing training data collected from the farm under monitoring. In fact, the best result (F-score of 59.25%) was obtained when using data solely from one farm (the same one used for both training and testing). However, the performance improvement was only 8.67% compared to the worst result, which involved training with data from one farm and testing on data from the other farm. These findings suggest that vocalisation detection generalisation is possible, even when operating in a farm without previously recorded samples.

In future research, we plan to incorporate data from additional farms to validate the conclusions drawn in this study in diverse environmental settings. In terms of the CowTalkPro project, once the veterinary researchers have analyzed the automatic vocalizations detected by the algorithm in multiple environments and the acoustic sensors are deployed in the farms, it will be necessary to study how can the real-time system assist both veterinary researchers and farmers to improve the welfare of cows.

5. ACKNOWLEDGMENT

This research is being supported by the project SNEO-20211301 "Desarrollo de un sensor de sonido en vacas para evaluar su salud y el bienestar animal", a NEOTEC funding awarded to AWEC Advisors together with HER-La Salle research team. The authors would also like to thank the Departament de Recerca i Universitats (Generalitat de Catalunya) under Grant Ref. 2021-SGR-01396 for the funding of Human-Environment Research (HER) research group.

6. REFERENCES

[1] P. Llonch, E. Mainau, I. R. Ipharraguerre, F. Bargo, G. Tedó, M. Blanch, and X. Manteca, "Chicken or the egg: the reciprocal association between feeding behavior and animal welfare and their impact on productivity in dairy cows," *Frontiers in veterinary science*, vol. 5, p. 305, 2018.

[2] B. Gołebiewska, M. Gebska, and J. Stefańczyk, "Animal welfare as one of the criterion determining polish consumers'decisions regarding their purchase of meat," *Acta Scientiarum Polonorum. Oeconomia*, vol. 17, no. 3, pp. 13–21, 2018.

[3] D.-H. Jung, N. Y. Kim, S. H. Moon, C. Jhin, H.-J. Kim, J.-S. Yang, H. S. Kim, T. S. Lee, J. Y. Lee, and S. H. Park, "Deep learning-based cattle vocal classification model and real-time livestock monitoring system with noise filtering," *Animals*, vol. 11, no. 2, 2021. [Online]. Available: https://www.mdpi.com/2076-2615/11/2/357

[4] V. Exadaktylos, M. Silva, and D. Berckmans, "Chapter automatic identification and interpretation of animal sounds, application to livestock production optimisation," 2014.

[5] S. Ntalampiras, A. Pezzuolo, S. Mattiello, M. Battini, and M. Brščić, "Automatic detection of cow/calf vocalizations in free-stall barn," in *2020 43rd International Conference on Telecommunications and Signal Processing (TSP)*, 2020, pp. 41–45.

[6] G. Meen, M. Schellekens, M. Slegers, N. Leenders, E. van Erp-van der Kooij, and L. Noldus, "Sound analysis in dairy cattle vocalisation as a potential welfare monitor," *Computers and Electronics in Agriculture*, vol. 118, pp. 111–115, 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0168169915002549

[7] G. Özmen, İ. A. Ozkan, I. Seref, S. Tasdemir, Ç. Mustafa, and E. Arslan, "Sound analysis to recognize cattle vocalization in a semi-open barn," *Gazi Mühendislik Bilimleri Dergisi*, vol. 8, no. 1, pp. 158–167, 2022.

[8] *H5 Handy Recorder - Operation Manual*, Zoom Corporation, 2014.

[9] D. Sinha and M. El-Sharkawy, "Thin mobilenet: An enhanced mobilenet architecture," in *2019 IEEE 10th annual ubiquitous computing, electronics & mobile communication conference (UEMCON)*. IEEE, 2019, pp. 0280–0285.

[10] R. P. Foundation, "Raspberry pi," https://www.raspberrypi.com (accessed on 12 Jun 2023).

[11] E. Vidaña-Vila, J. Navarro, D. Stowell, and R. M. Alsina-Pagès, "Multilabel acoustic event classification using real-world urban data and physical redundancy of sensors," *Sensors*, vol. 21, no. 22, p. 7470, 2021.

[12] A. Mesaros, T. Heittola, and T. Virtanen, "Metrics for polyphonic sound event detection," *Applied Sciences*, vol. 6, no. 6, 2016. [Online]. Available: https://www.mdpi.com/2076-3417/6/6/162

LOW-COMPLEXITY ACOUSTIC SCENE CLASSIFICATION USING DEEP MUTUAL LEARNING AND KNOWLEDGE DISTILLATION FINE-TUNING

Shilong Weng[1], *Liu Yang** [1], *Binghong Xu*[1], *Xing Li*[2]

[1]School of Computer Science and Cyber Engineering, Guangzhou University, Guangzhou, China
[2]vivo Mobile Commun co Ltd, China
2112106223@e.gzhu.edu.cn, yangliupu@gmail.com, 2112006235@e.gzhu.edu.cn, li.xing2@vivo.com

ABSTRACT

In this paper, a novel model training framework constituted by deep mutual learning (DML) and knowledge distillation (KD) fine-tuning is proposed for low-complexity acoustic scene classification (ASC). The model training phase consists of two stages. In the first stage, a ResNet38 teacher model pre-trained on AudioSet and three low-complexity BC-Res2Net student models with different widths and depths are involved in DML to enhance the teacher model performance, and attain a well-initialized student model. In the second stage, we utilize KD fine-tuning to teach this student model to learn from the high-performing teacher model while maintaining the predictive performance of the teacher model. Experimental results on *TAU Urban Acoustic Scenes 2022 Mobile development dataset* demonstrate the effectiveness of the proposed framework as well as its superiority over using KD alone under the same configurations.

Index Terms— Acoustic scene classification, deep mutual learning, knowledge distillation fine-tuning, ResNet38, BC-Res2Net

1. INTRODUCTION

Low-complexity acoustic scene classification (ASC) aims to classify a given recording into a predefined acoustic scene category by a well-designed system. It has received increasing interest because it enables deployment of classification systems on a wide range of edge devices with limited computational capacity and memory resources. This paper focus on the low-complexity ASC task in the Detection and Classification of Acoustic Scenes and Events (DCASE) 2023 challenge [1]. The low-complexity and generalization requirements of this task are characterized by three key points:

P1. Audios were recorded by a variety of devices in different cities, and synthetic data for several mobile devices was also generated based on the recorded audio.

P2. The memory for model parameters must be capped at 128K, regardless of the parameter type utilized.

P3. The computational consumption for a single inference must be limited to 30 million multiply-accumulate operations (MMACs).

For P1, augmentation schemes are frequently employed to enhance the generalization capacity of the system on recordings from unseen devices [2, 3, 4]. For P2 and P3, most low-complexity ASC approaches are based on model compression techniques and can be assorted into four classes, including feature selection [5, 6], pruning [7, 8, 9], designing efficient network architectures [10, 11, 12] and knowledge distillation (KD) [13, 14, 15]. KD has been widely utilized to derive efficient and lightweight student models by training them to emulate large and high-performing teacher models. Motivated by the concept of KD, Zhang et al. [16] presented a model training strategy called deep mutual learning (DML), in which multiple student models could learn collaboratively and teach each other throughout the training process, aiding in discovering a wider and more robust minima that generalized better to test data. The DML strategy has been applied in various fields [17, 18] and proven to be useful.

In this paper, we propose an effective model training framework that consists of DML and KD fine-tuning, in which DML plays a vital role in preparing both the teacher model and student model for the following KD fine-tuning. As opposed to solely using student models in [16], the proposed framework incorporates one pre-trained teacher model and three student models of same type but with different widths and depths during the DML stage.

The remainder of this paper is organized as follows. Section 2 describes the methodology for preprocessing and augmenting data prior to input into models. In Section 3, the proposed model training framework that consists of DML and KD fine-tuning is presented to obtain a low-complexity ASC system. Section 4 describes the experimental setup and presents the experimental results. Finally, Section 5 concludes this study.

2. DATA PREPROCESSING AND AUGMENTATION

2.1. Data preprocessing

The dataset utilized for this task is the *TAU Urban Acoustic Scenes 2022 Mobile development dataset* [19]. It is derived from the *TAU Urban Acoustic Scenes 2020 Mobile development dataset* by cropping the original 10-second audio files into 1-second clips, and the sampling rate was 44.1 kHz. We borrowed the CP-JKU scheme from [20] and reassembled all the training audio into 10-second segments according to the segment identifiers. Then the audio was downsampled to 32 kHz.

2.2. Microphone Impulse Response and Augmentation

To enhance the diversity of training data and promote the generalization capability of the ASC model to various recording devices, we simulate "new" recording devices by randomly convolving the reassembled 10-second audio signals with the freely available microphone impulse responses (IRs) from the Microphone Impulse Response Project (MicIRP) library [21] as suggested in [3]. Totally

---

This work was partially funded by the National Natural Science Foundation of China under grant 61801133. *Corresponding author.

68 IRs of vintage microphones are utilized, which means synthetic audio data that recorded by 68 “new” devices is included in the training data. The probability of the audio in training dataset being convolved with IRs is set as 0.5, in order to ensure both the original audio and the simulated audio are fed into the model during training.

Then each 10-second recording was randomly cropped into a 1-second snippet and fed to the model in a single epoch. That is to say, only one-tenth of the available data can be seen by the model, which can increase the diversity of the training data to a certain extent as well.

Furthermore, two kinds of data augmentation techniques are applied to the training data sequentially. The first one includes time shifting and time-frequency masking operations. We randomly shift an audio clip by a time interval shorter than 1 second forward. To extract temporal and spectral features from the audio data, we apply short-time Fourier transform (STFT) to the shifted audio using a Hanning window of size 2048 and a hop size of 1024 samples for student models, and a Hanning window of size 800 and a hop size of 320 samples for teacher model. Then mel filter banks are applied with 256 mel bins for both student and teacher models, followed by a logarithmic operation to obtain the log mel spectrograms of the audio. Finally, we apply the time-frequency masking to the log mel spectrograms, and the maximum size of each masking band is set as 8 for the time domain and 40 for the frequency domain, respectively. The application probability of both time shifting and time-frequency masking is 0.7. The second kind of data augmentation techniques includes mixup [22] and mixstyle [23]. The weight parameters of both mixup and mixstyle are chosen as $\alpha = 0.3$, and their application probabilities are 0.7 and 0.6, respectively.

3. MODEL TRAINING FRAMEWORK USING DML AND KD FINE-TUNING

A novel framework that combines DML with KD fine-tuning is proposed for model training. As shown in Fig. 1(a), three low-complexity student models and a pre-trained teacher model are employed in DML. The goal of DML is to further improve the performance of teacher model and attain a well-initialized student model. Then we utilize KD fine-tuning to transfer the knowledge of the high-performing teacher model to the low-complexity student model.

3.1. Deep Mutual Learning

DML trains two or more networks which are denoted as $\mathrm{Model} = \{\mathrm{model}_1, \cdots, \mathrm{model}_N\}$ simultaneously. In the proposed framework, the number of networks $N = 4$. At each iteration, every network learns from the other networks. Fig. 1 (b) illustrates the schematic diagram of DML. Note that for convenience, Fig. 1(b) only displays how model_1, which is denoted as Init_BC-Res2Net, learns from other models. For the nth model, denoting its logit on the mth category as z_m^n, then its predicted soft probability on the mth category can be calculated by comparing z_m^n with the other logits [13],

$$\hat{y}_m^n = \frac{\exp\left(z_m^n / T_{\mathrm{dml}}\right)}{\sum_{j=1}^{M} \exp\left(z_j^n / T_{\mathrm{dml}}\right)},\ m = 1, \cdots, M, \tag{1}$$

where M is the total number of categories, and T_{dml} is a temperature utilized to control the degree of smoothing of the soft probability. When $T_{\mathrm{dml}} = 1$, (1) degenerates into softmax operation.

The output probability distribution of the nth model can be written as

$$\hat{Y}_{T_{\mathrm{dml}}}^n = \left[\hat{y}_1^n, \cdots, \hat{y}_M^n\right],\ n = 1, \cdots, N, \tag{2}$$

and is passed to the other networks as a soft label. The soft label loss of the nth model is computed as

$$L_{\mathrm{soft}}^n = \frac{1}{N-1} \sum_{\substack{1 \le l \le N \\ l \ne n}} \mathrm{KL}\left(\hat{Y}_{T_{\mathrm{dml}}}^n \middle|\middle| \hat{Y}_{T_{\mathrm{dml}}}^l\right). \tag{3}$$

The hard label loss of the nth model is obtained by cross-entropy. Finally, the total loss of the nth model in the DML process is the weighted sum of its hard label loss and soft label loss, i.e.,

$$L_{\mathrm{dml}}^n = L_{\mathrm{label}}^n + \lambda_{\mathrm{dml}} L_{\mathrm{soft}}^n, \tag{4}$$

where λ_{dml} is the weight of the soft label loss.

Note that DML does not require additional knowledge source and it extracts knowledge directly through interactions among networks. It can effectively improve the performances of all networks involved in learning. More importantly, the interactions among the output soft labels of the models enable DML to avoid overfitting and enhance the robustness of all the models. After the DML process, a high-performing teacher model and a properly initialized student model are obtained for the following KD fine-tuning.

3.2. Student Model

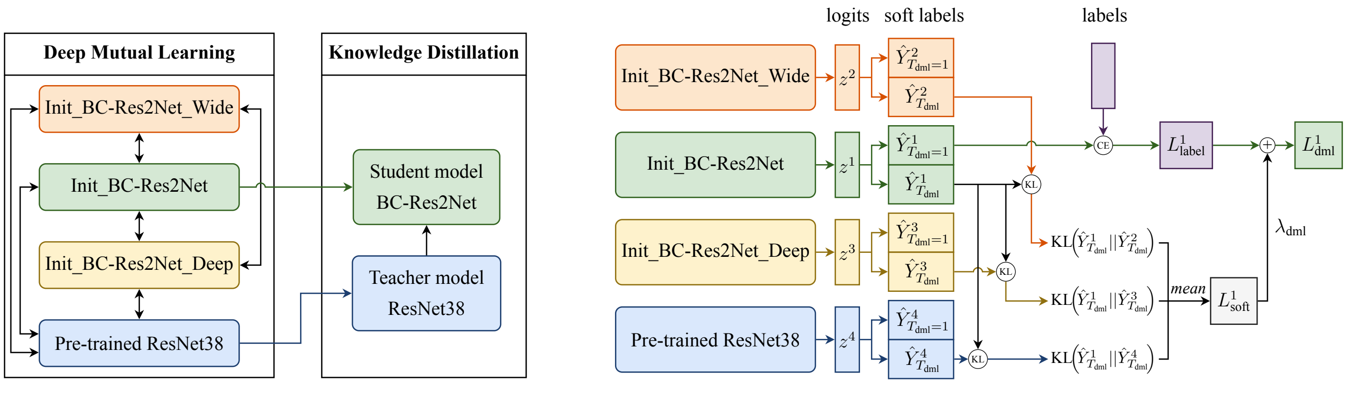
The student model employed in the proposed model training framework is based on the Broadcast Residual Network (BC-ResNet) [24]. BC-ResNet was a deep neural network developed for efficient keyword detection, and it utilized both residual learning and broadcast mechanism. In the student model employed in the proposed framework, the ResNet part in BC-ResNet is replaced by Res2Net [25], and the new model is referred to as BC-Res2Net [26]. By adding small blocks of residuals to the original residual cell structure, Res2Net can extract features within different receptive fields and in multiple scales at a lower computational cost. In addition, a simple but effective module called Residual Normalization (ResNorm) is added to BC-Res2Net to reduce the system reliance on various devices [11].

Three student models with different widths and depths are utilized in DML, including a BC-Res2Net with the number of channels $C = 24$, a wider BC-Res2Net with $C = 80$, which is denoted as BC-Res2Net_wide, and a deeper BC-Res2Net named BC-Res2Net_deep, in which $C = 24$, and the number of BC-Res2Block and ResNorm within each module is doubled. The purpose of adding BC-Res2Net_wide and BC-Res2Net_deep to DML is to allow BC-Res2Net to learn specific information contained in deeper and wider networks, thereby compensating for its limitations of width and depth.

Denoting the number of Mel bins, and the number of time steps as F and T, Table 1 shows the overall architecture of the employed BC-Res2Net and the size of the output feature map in each block.

3.3. Teacher Model

We use ResNet38 trained by Kong et al. [27] on AudioSet [28] as the teacher model. ResNet38 is a deep audio neural network trained with 1.9 million audio clips and an ontology of 527 sound classes. Residual networks help ResNet38 to alleviate the vanishing


Deep Mutual Learning
Init_BC-Res2Net_Wide
Init_BC-Res2Net
Init_BC-Res2Net_Deep
Pre-trained ResNet38
Knowledge Distillation
Student model
BC-Res2Net
Teacher model
ResNet38
logits
soft labels
labels
z^2
z^1
z^3
z^4
$\hat{Y}^2_{T_{\mathrm{dml}}=1}$
$\hat{Y}^2_{T_{\mathrm{dml}}}$
$\hat{Y}^1_{T_{\mathrm{dml}}=1}$
$\hat{Y}^1_{T_{\mathrm{dml}}}$
$\hat{Y}^3_{T_{\mathrm{dml}}=1}$
$\hat{Y}^3_{T_{\mathrm{dml}}}$
$\hat{Y}^4_{T_{\mathrm{dml}}=1}$
$\hat{Y}^4_{T_{\mathrm{dml}}}$
CE
KL
L^1_{label}
L^1_{dml}
λ_{dml}
$\mathrm{KL}(\hat{Y}^1_{T_{\mathrm{dml}}}||\hat{Y}^2_{T_{\mathrm{dml}}})$
$\mathrm{KL}(\hat{Y}^1_{T_{\mathrm{dml}}}||\hat{Y}^3_{T_{\mathrm{dml}}})$
$\mathrm{KL}(\hat{Y}^1_{T_{\mathrm{dml}}}||\hat{Y}^4_{T_{\mathrm{dml}}})$
mean
L^1_{soft}

(a)

(b)

Figure 1: Diagram of the proposed model training framework. (a) DML prepares both student model and teacher model for the following KD fine-tuning. (b) Three BC-Res2Net student models and a pre-trained ResNet38 teacher model are involved in DML. For convenience, only the process by which Init_BC-Res2Net learns from other models is displayed. ⓒⓔ denotes the computation of cross-entropy.

Table 1: Architecture of BC-Res2Net as a student model.

Block	Output Size
input	$(1, F, T)$
ResNorm Conv2D (5×5)	$(2C, F/2, T/2)$
BC-Res2Block × 1 ResNorm, MaxPool(2,2)	$(C, F/4, T/4)$
BC-Res2Block × 1 ResNorm, MaxPool(2,2)	$(1.5C, F/8, T/8)$
BC-Res2Block × 3 ResNorm	$(2C, F/8, T/8)$
BC-Res2Block × 3 ResNorm	$(2.5C, F/8, T/8)$
Conv2D (5×5, Group = 2.5C) Conv2D (1×1), Mean	$(4C, 1, 1)$
Conv2D (1×1)	(10)

gradient problem that commonly encountered in training very deep networks. The large number of sound classes can provide a comprehensive representation of unique sounds. Therefore, ResNet38 has demonstrated high accuracy rates in real-world sound classification tasks.

3.4. Knowledge Distillation Fine-tuning

KD has been widely used in various fields as a model compression tool. When training a student model, the probability distributions of the teacher model's predictions on the input audio samples, which are also known as soft labels, are utilized as an additional target. Therefore, KD allows the student model to imitate the output of the teacher model as much as possible, leading to improved generalization capacity and increased fitting speed of the student model.

For KD fine-tuning, we utilize the DML trained ResNet38 and BC-Res2Net as the teacher model and the initialized BC-Res2Net student model, respectively. Soft labels and soft label loss are calculated in a similar way to DML as expressed in (2) and (3). Denoting the soft label loss of the student model in KD fine-tuning by L_{dist}, the total loss in KD can be calculated by

$$L_{\mathrm{kd}} = L_{\mathrm{label}} + \lambda_{\mathrm{kd}} L_{\mathrm{dist}}, \tag{5}$$

where λ_{kd} is the weight of the soft label loss.

4. EXPERIMENTAL SETUP AND RESULTS

4.1. Training Setup

The learning rate during the experiments is fixed at 1e-4 for individual training of both the student and teacher models in the process of DML and KD fine-tuning[1]. Adam optimizer is utilized, and our experimental results indicate that the type of optimizer does not have a significant impact on the outcomes.

For student model, the scale size in Res2Net is set as 4. During DML and KD fine-tuning, the temperatures T_{dml} and T_{kd} are both set at a medium value 3 to generate soft labels, ensuring that the labels are smooth while not too much information is lost at the same time. For weight of the soft label loss, $\lambda_{\mathrm{dml}} = 1$ and $\lambda_{\mathrm{kd}} = 50$. This is due to the fact that in DML, each student model is trained from scratch, and the purpose of DML is to promote the performance of the pre-trained teacher model and obtain a well-initialized student model. Therefore, we do not want a model to put great influence on another. However, in the KD fine-tuning, we hope the student model to learn as much as possible from the representations of the high-performing teacher model.

4.2. Results

The performances of our student model BC-Res2Net and teacher model ResNet38 are evaluated on the test set provided by DCASE 2023 challenge and illustrated in Table 2. During the experiments, we followed the official data partitioning principle [29].

It can be seen from Table 2 that mixstyle outperforms mixup for both student model and teacher model, which means mixstyle is more competent to enhance device generalization. What is more, using the combination of DML and KD fine-tuning produces superior results compared to using DML alone. Clearly, the training

---

[1] Source code is available at https://github.com/wsdragon2010/GZHU_DCASE2023_TASK1

Table 2: Accuracy and log loss performances of student model and teacher model on test set under different configurations. BC-Res2Net has a width of C = 24. "Conv_IR" indicates whether the input audio is convolved with IRs in a certain probability. "Real", "Seen" and "Unseen" represent real devices, seen simulated devices and unseen simulated devices, respectively.

Model	**Configuration**					**Acc. (%)**				**Log Loss**
	Conv_IR	Mixup	Mixstyle	DML	KD	Real	Seen	Unseen	Overall	
student model BC-Res2Net	✗	✗	✗	✗	✗	63.71	48.17	36.90	49.59	2.586
	✓	✗	✗	✗	✗	**66.99**	55.73	45.45	56.05	1.581
	✗	✓	✗	✗	✗	62.91	52.09	42.68	52.55	1.558
	✗	✗	✓	✗	✗	65.40	55.34	47.05	55.93	1.289
	✓	✗	✓	✗	✗	64.34	56.90	52.21	57.81	1.202
	✓	✗	✓	✓	✗	61.04	57.34	**56.09**	58.16	1.158
	✓	✗	✓	✓	✓	65.67	**60.50**	54.61	**60.26**	**1.131**
teacher model ResNet38	✗	✗	✗	✗	-	70.30	52.28	44.61	55.73	3.442
	✓	✗	✗	✗	-	74.07	61.14	58.24	64.48	1.645
	✗	✓	✗	✗	-	72.76	54.04	48.63	58.47	1.307
	✗	✗	✓	✗	-	74.15	61.04	56.81	64.00	1.544
	✓	✗	✓	✗	-	74.59	67.96	64.09	68.88	1.138
	✓	✗	✓	✓	-	**76.09**	**71.10**	**69.97**	**72.39**	**0.836**

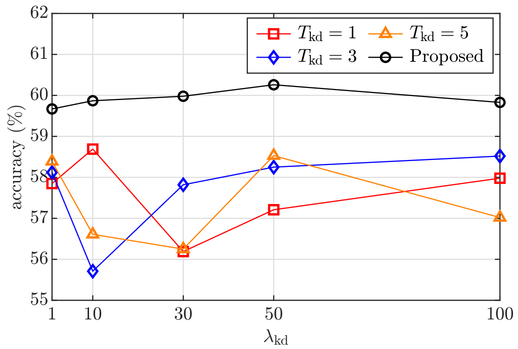


Figure 2: Accuracy of using KD alone with various λ_{kd} and T_{kd}, comparing to the performance of the proposed framework.

Table 3: Performance comparison of various student models and teacher models.

Model	**Params**	**MMACs**	**Acc. (%)**
RFR-CNN [14], 2022	127,046	29.06	59.76
BC-Res2Net (Ours)	**76,906**	**23.97**	**60.26**
PaSST-Ensemble [15], 2023	-	-	63.63
PaSST & CP-ResNet Ensemble [15], 2023	-	-	68.31
ResNet38 (Ours)	73,804,121	9,179.52	**72.39**

framework that includes convolution with IRs, mixstyle, DML and KD fine-tuning performs the best.

Fig. 2 illustrates the accuracy performance of the student model by using KD alone with various values of weight parameter λ_{kd} at different temperatures T_{kd}. By comparing with the performance of the proposed framework while $T_{\text{kd}} = 3$, $T_{\text{dml}} = 3$ and $\lambda_{\text{dml}} = 1$, it can be observed from Fig. 2 that regardless of the parameter tuning, using KD alone can not pass the performance beyond our proposed framework. Besides, experiments reveal that DML enables the student model to converge more quickly with improved performance during KD fine-tuning. Conversely, using KD alone tends to result in unstable student performance and makes the model sensitive to the weight parameter λ_{kd}, as shown in Fig. 2. This highlights the necessity of DML. To summarize, the combination of DML and KD fine-tuning provides a fast and effective way to improve the performance of low-complexity ASC system.

To demonstrate the effectiveness of the proposed model training framework, we compare our student model BC-Res2Net with the student model denoted as "RFR-CNN" that employed in [14], and compare our teacher model ResNet38 with the teacher models referred to as "PaSST-Ensemble" and "PaSST & CP-ResNet Ensemble" in [15]. Note that "PaSST-Ensemble" is the fusion of 6 different PaSST models, and "PaSST & CP-ResNet Ensemble" uses the fusion results of 6 different PaSST models and 6 different CP-ResNet models, while we utilize only one teacher model, i.e., ResNet38. As shown in Table 3, our student model outperforms "RFR-CNN" by approximately 0.5% while having less parameters and MMACs. Moreover, our teacher model exhibits an absolute advantage over the two teacher models in [15] by almost 4% in terms of overall classification accuracy.

5. CONCLUSION

In this paper, we tackle with the low-complexity ASC task in DCASE 2023 challenge. We present a novel model training framework that consists of DML and KD fine-tuning. DML helps both teacher model and student model prepare for the following KD fine-tuning, then KD is used to compress the knowledge of a high-performing ResNet38 teacher model into a low-complexity BC-Res2Net student model in an optimal manner. Experimental results demonstrate that DML plays a critical role in enhancing the final performance of the proposed low-complexity ASC system. Next, we aim to apply the proposed training framework to newer and stronger models in an attempt to achieve even better performance.

6. REFERENCES

[1] I. Martín-Morató, F. Paissan, A. Ancilotto, T. Heittola, A. Mesaros, E. Farella, A. Brutti, and T. Virtanen, "Low-complexity acoustic scene classification in DCASE 2022 challenge," 2022.

[2] J. Han, M. Matuszewski, O. Sikorski, H. Sung, and H. Cho, "Randmasking augment: a simple and randomized data augmentation for acoustic scene classification," in *ICASSP*, 2023, pp. 1–5.

[3] S. Sonowal and A. Tamse, "Novel augmentation schemes for device robust acoustic scene classification," in *Interspeech*, 2022, pp. 4182–4186.

[4] H. Hu, C.-H. H. Yang, X. Xia, X. Bai, X. Tang, Y. Wang, S. Niu, L. Chai, J. Li, H. Zhu, F. Bao, Y. Zhao, S. M. Siniscalchi, Y. Wang, J. Du, and C.-H. Lee, "A two-stage approach to device-robust acoustic scene classification," in *ICASSP*, 2021, pp. 845–849.

[5] L. P. Schmidt, B. Kiliç, and N. Peters, "Feature selection using alternating direction method of multiplier for low-complexity acoustic scene classification," in *DCASE 2022 Workshop*, 2022.

[6] C. Paseddula and S. V. Gangashetty, "Acoustic scene classification using single frequency filtering cepstral coefficients and DNN," in *IJCNN*, 2020, pp. 1–6.

[7] S. Tofigh, M. O. Ahmad, and M. Swamy, "A low-complexity modified thinet algorithm for pruning convolutional neural networks," *IEEE Signal Process. Lett.*, vol. 29, pp. 1012–1016, 2022.

[8] J. Kim, K. Yoo, and N. Kwak, "Position-based scaled gradient for model quantization and pruning," in *NIPS*, 2020, pp. 20 415–20 426.

[9] J. Frankle, G. K. Dziugaite, D. M. Roy, and M. Carbin, "Pruning neural networks at initialization: Why are we missing the mark?" in *ICLR*, 2021.

[10] M. Aswathy and K. Suresh, "RQNet: residual quaternion CNN for performance enhancement in low complexity and device robust acoustic scene classification," *IEEE Trans. Multimedia*, pp. 1–13, 2023.

[11] B. Kim, S. Yang, J. Kim, and S. Chang, "Domain generalization on efficient acoustic scene classification using residual normalization," in *DCASE 2021 Workshop*, 2021.

[12] A. Singh and M. D. Plumbley, "Low-complexity CNNs for acoustic scene classification," in *DCASE 2022 Workshop*, 2022.

[13] G. E. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *ArXiv*, vol. abs/1503.02531, 2015.

[14] F. Schmid, S. Masoudian, K. Koutini, and G. Widmer, "Knowledge distillation from transformers for low-complexity acoustic scene classification," in *DCASE 2022 Workshop*, 2022.

[15] F. Schmid, T. Morocutti, S. Masoudian, K. Koutini, and G. Widmer, "CP-JKU submission to DCASE23: efficient acoustic scene classification with CP-Mobile," DCASE 2023 challenge, Tech. Rep., 2023.

[16] Y. Zhang, T. Xiang, T. M. Hospedales, and H. Lu, "Deep mutual learning," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4320–4328.

[17] F. Xudong, G. Xiaofeng, K. Ping, L. Xianglong, and Z. Yalou, "Pedestrian detection and tracking with deep mutual learning," in *ICCWAMTIP*, 2021, pp. 217–220.

[18] R. Masumura, M. Ihori, A. Takashima, T. Tanaka, and T. Ashihara, "End-to-end automatic speech recognition with deep mutual learning," in *APSIPA ASC*, 2020, pp. 632–637.

[19] T. Heittola, A. Mesaros, and T. Virtanen, "Acoustic scene classification in DCASE 2020 challenge: generalization across devices and low complexity solutions," in *DCASE 2020 workshop*, 2020, pp. 56–60.

[20] F. Schmid, S. Masoudian, K. Koutini, and G. Widmer, "CP-JKU submission to DCASE22: distilling knowledge for low-complexity convolutional neural networks from a patchout audio transformer," DCASE 2022 challenge, Tech. Rep., 2022.

[21] "Microphone impulse response project." [Online]. Available: http://micirp.blogspot.com/

[22] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "Mixup: beyond empirical risk minimization," in *ICLR*, 2018, pp. 1–13.

[23] K. Zhou, Y. Yang, Y. Qiao, and T. Xiang, "Domain generalization with mixstyle," in *ICLR*, 2021.

[24] B. Kim, S. Chang, J. Lee, and D. Sung, "Broadcasted residual learning for efficient keyword spotting," in *Interspeech*, 2021.

[25] S.-H. Gao, M.-M. Cheng, K. Zhao, X.-Y. Zhang, M.-H. Yang, and P. Torr, "Res2Net: a new multi-scale backbone architecture," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 2, pp. 652–662, 2021.

[26] J.-H. Lee, J.-H. Choi, P. M. Byun, and J.-H. Chang, "Multi-scale architecture and device-aware data-random-drop based fine-tuning method for acoustic scene classification," in *DCASE 2022 Workshop*, 2022.

[27] Q. Kong, M. Yin Cao, T. Iqbal, Y. Wang, W. Wang, and M. D. Plumbley, "PANNs: large-scale pretrained audio neural networks for audio pattern recognition," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 28, pp. 2880–2894, 2020.

[28] J. F. Gemmeke, D. P. W. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio set: an ontology and human-labeled dataset for audio events," in *ICASSP*, 2017, pp. 776–780.

[29] A. Mesaros, T. Heittola, and T. Virtanen, "A multi-device dataset for urban acoustic scene classification," in *DCASE 2018 workshop*, 2018, pp. 9–13.

TWO VS. FOUR-CHANNEL SOUND EVENT LOCALIZATION AND DETECTION

Julia Wilkins[1], *Magdalena Fuentes*[1], *Luca Bondi*[2],
Shabnam Ghaffarzadegan[2], *Ali Abavisani*[2], *Juan Pablo Bello* [1]

[1] New York University, New York, NY, USA,
[2] Bosch Research, Pittsburgh, PA, USA

jw3596@nyu.edu

https://github.com/juliawilkins/SELD-2v4-DCASE23/

ABSTRACT

Sound event localization and detection (SELD) systems estimate both the direction-of-arrival (DOA) and class of sound sources over time. In the DCASE 2022 SELD Challenge (Task 3), models are designed to operate in a 4-channel setting. While beneficial to further the development of SELD systems using a multichannel recording setup such as first-order Ambisonics (FOA), most consumer electronics devices rarely are able to record using more than two channels. For this reason, in this work we investigate the performance of the DCASE 2022 SELD baseline model using three audio input representations: FOA, binaural, and stereo. We perform a novel comparative analysis illustrating the effect of these audio input representations on SELD performance. Crucially, we show that binaural and stereo (i.e. 2-channel) audio-based SELD models are still able to localize and detect sound sources *laterally* quite well, despite overall performance degrading as less audio information is provided. Further, we segment our analysis by scenes containing varying degrees of sound source polyphony to better understand the effect of audio input representation on localization and detection performance as scene conditions become increasingly complex.

Index Terms— sound event localization and detection, sound source localization, spatial audio, explainability

1. INTRODUCTION

Sound Event Localization and Detection (SELD) is the process of estimating the direction-of-arrival (DOA) and class of sound events over time, given an input audio signal. SELD systems can translate well to a variety of real-world applications, including navigation for autonomous systems and assistive robotic devices. SELD methods are rooted in traditional signal processing techniques for multichannel audio processing, such as Steered Response Power [1] and acoustic intensity vectors [2]. For human-inspired audio recordings (e.g. binaural recordings), interaural time difference (ITD) and interaural level difference (ILD) are commonly used to characterize the direction of arrival of sounds [3]. However, these cues alone have shown limitations in terms of localization accuracy in real-world scenes that are particularly noisy, reverberant, or polyphonic [4–6]. Deep learning approaches were recently popularized to address these challenges in the context of SELD tasks [7–11]; most systems still utilize signal processing-based features like generalized cross correlation (GCC) and Mel spectrograms but benefit from automatic feature learning to improve robustness in difficult scene conditions [7, 11–13]. For example, in [14], authors use a CRNN architecture with magnitude and phase spectrograms from multichannel audio to show accurate DOA estimation and multiple sound source detection in reverberant conditions.

In the DCASE 2022 SELD challenge (Task 3), models were evaluated using real multichannel sound recordings. Participants had access to real recordings for development and could also use additional synthetic or real data for training. The challenge operates in a multichannel setting, utilizing two formats of 4-channel recordings: first-order Ambisonics (FOA) and a tetrahedral mic array. We are interested in exploring the capabilities of current SELD systems using more commonly found 2-channel microphone setups, namely binaural and stereo, as typical consumer electronics devices lack such complex 4-channel configurations.

There is little prior research quantifying the effect of using various audio input representations (i.e. 2 vs. 4-channel audio) for SELD tasks in deep learning-based systems. In the psychoacoustics community, this effect is well-studied; it is known that there is a general loss in spatial understanding between 4-channel audio configurations (e.g. Ambisonics) vs. 2-channel configurations (e.g. binaural or stereo). [15,16]. Humans can localize lateral sound sources well in binaural and stereo settings, but front-back confusion may increase without sufficient spatial information [3, 17, 18]. Further, perceiving the elevation of sound sources when listening to stereo audio in particular has been shown to be very difficult, largely due to the lack of interaural cues present in this recording configuration unlike that of a binaural setup [16]. However, these phenomena are underexplored in the context of deep learning-based systems for SELD. In [19], authors compared sound event detection performance using synthetic FOA, binaural, and monaural audio data in a CRNN-based system. Our approach differs significantly in that we provide a quantitative analysis of localization *and* detection performance, we use a FOA dataset of real recordings in addition to synthetic and decode these recordings to binaural, and lastly we include the stereo audio configuration as a point of comparison as this is common in consumer electronics devices today.

In this work we present a novel comparative analysis of the DCASE 2022 SELD baseline model across FOA, binaural, and stereo audio input representations. To the best of our knowledge, this is the first work quantifying the effect of these audio configurations on both localization and detection performance in a deep-learning based SELD system. We show that lateral sound source localization remains fairly accurate in the 2-channel settings despite an overall degradation in SELD performance, and provide an analysis of performance in scenes of varying levels of polyphonic sound source complexity.

2. PROBLEM FORMULATION

In this manuscript, we examine the problem of Sound Event Localization and Detection (SELD) under different audio input representations: first-order Ambisonics (FOA), binaural, and stereo recordings. In this context, *detection* refers to determining the number of active sound sources per class over time, while *localization* aims at identifying the azimuth and elevation angle for each of the active sources over time. While Ambisonics recordings provide state-of-the-art performance in SELD [20], in practical applications we hypothesize that binaural and stereo recordings are more accessible.

We rely on the most popular framework used by participants in the DCASE 2022 Challenge Task 3. A multichannel audio recording is fed as input to a Convolutional Neural Network (CNN), whose output is a 4-dimensional matrix arranged according to the Multi-Activity Coupled Cartesian DOA (ACCDOA) format [21]. For a given class, time instant, and sound source index, the model arrives at a three-dimensional vector (x, y, z) whose orientation represents the direction of arrival of the sound, and whose intensity is directly proportional to the likelihood of a sound of that class being present at a given time.

First-order Ambisonics (FOA): FOA is a 4-channel, 3D audio recording format. In FOA, each channel corresponds to a spherical harmonic component representing a change in sound pressure in a specific direction [22]. The channels *W, Y, Z, X* map to the omnidirectional, left-right, vertical, and front-back directions of sound pressure change, respectively.

Binaural: The binaural recording technique aims to capture 3D audio in just two channels, ideally simulating the experience of a human experimncing auditory cues. Binaural audio is typically recorded using two microphones placed in the ears of a dummy head (e.g. Neumann KU100), or synthesized using the head-related transfer functions (HRTFs) of such a dummy head [23]. Binaural recordings deliver immersive spatial sounds containing amplitude, time and timbral differences of two channels vs. traditional stereo recordings where only amplitude and time differences are available.

Stereo: In stereo recordings, two microphones are used to capture the left and right audio channels independently. This differs from binaural recordings; in the binaural configuration the goal is to simulate a human's listening experience. Critically, in a stereo setup, elevation differentiation cannot be perceived; binaural recordings contain the filtering effect of the head, ear pinna, and torso and this is not present in a stereo recording configuration [16].

3. EXPERIMENTAL SETUP

3.1. Datasets

Following the setup of the DCASE 2022 Task 3 challenge, we rely on the STARSS22 dataset [24], together with a synthetic mixture (SYNMIX) for baseline training[1] provided by the organizers of the challenge. The STARSS22 dataset is comprised of 121 recordings of various lengths of real sound scenes across 13 sound event classes, with around 5 hours of audio recordings in 4-channel FOA format and an interpolated tetrahedral microphone array. At the time of this work, the evaluation set was not yet released, so we use the "development" partition of train and test, consisting of 67 and 54 recordings, respectively. The dataset contains instances with up to 5 simultaneous sound sources, and up to 4 simultaneous sources of the same class, though 2-source polyphony is much more frequent.

Due to the small size of the STARSS22 dataset, a base set of synthetic data was also provided to participants (SYNMIX). This data is synthesized using audio samples from FSD50k [25] convolved with Spatial Room Impulse Responses from the TAU-Nigens Spatial Sound Events 2020 [26] and 2021 [27]. The dataset contains 1200, 1-minute synthesized FOA recordings across classes mapped to the classes present in STARSS22, and maximum polyphony of 2 sources.

Both datasets are annotated at 100ms resolution with labels of sound source class, azimuth, and elevation as well as additional flags for overlapping sound events. The azimuth angles $\phi \in [-180^\circ, 180^\circ]$, and elevation $\theta \in [-90^\circ, 90^\circ]$, with 0° at front. Note that azimuth angles increase counterclockwise.

3.2. Input representations

To fairly compare the three multichannel audio representations, we look at the problem of sound localization on the horizontal plane only by removing the elevation component, thus fixing elevation to 0° in the ground truth. We train and test separately for each input representation using the same acoustic scenes, simply replacing the original FOA audio representation with binaural or stereo audio, as per following procedures.

FOA → Binaural: To decode the original FOA audio from the STARRS22 and synthetic datasets to binaural, we used the BinauralDecoder plug-in from the IEM Plug-In Suite[2]. This decoder uses pre-processed Neumann KU100 dummy head HRTFs via the magnitude least-squares (MagLS) method proposed in [28]. We apply this binaural decoding to all FOA audio used in training and testing, yielding 2-channel binaural audio for our experiments [3].

FOA → Stereo: To convert our FOA audio to stereo, we used a very simple translation: $\mathit{left} = W + Y$ and $\mathit{right} = W - Y$, following [29]. Note that W is the omnidirectional signal and Y is the first-order horizontal (left-right) component. An increase in air pressure from left causes an increase in values of Y and an increase in pressure from the right causes a decrease in values of Y. Because of this, the simple translation above allows us to move easily from FOA to left and right channels yielding 2-channel stereo audio.

3.3. Baseline model

The model used for our analysis is the DCASE 2022 Task 3 Baseline model[4]. The architecture is similar to the CRNN-based model initially proposed in [7], with extensions to accommodate simultaneous sources of the same class in the Multi-ACCDOA format [21]. The input to the model is the multichannel audio, segmented into 5-second chunks, yielding a sequence of 50 x 0.1 second frames. In the FOA configuration, Mel spectrogram features are used to capture frequency information and intensity vectors provide spatial information. In the binaural and stereo settings, we modify the model slightly to use Mel spectrograms and GCC features. GCC features are commonly used in 2-channel localization settings to capture Time Difference of Arrival (TDOA) information between two microphones. Audio is resampled to 24kHz, and 64 Mel coefficients are computed from an STFT on windows of 1024 samples with a hop size of 480 samples. The model has 604.5K trainable parameters. Models are trained for a multi-output regression task, with a mean-squared-error loss, for 200 epochs using 1 RTX 8000 GPU, in batches of 64 samples with a learning rate of 10^{-3}. The model checkpoint with the lowest validation loss is selected.

3.4. Data augmentation via Audio Channel Swapping (ACS)

An initial exploration of the STARSS and SYNMIX datasets revealed that the distribution of azimuth angles across sound sources

---

[1] https://zenodo.org/record/6406873#.Y_-SBuzMK2o.

[2] https://plugins.iem.at/docs/plugindescriptions/#binauraldecoder.

[3] https://github.com/juliawilkins/ambisonics2binaural_simple.

[4] https://github.com/sharathadavanne/seld-dcase2022.

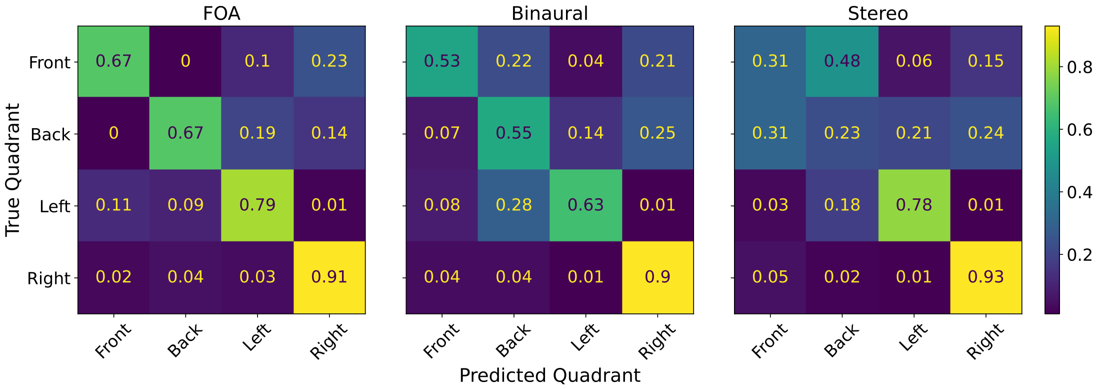


Figure 1: Normalized confusion matrices showing true vs. predicted quadrant of sources across audio configurations. The FOA model performs near-perfect at distinguishing front and back sources, while front and back sources are commonly confused in binaural and stereo settings. Quadrants of size 90° are defined based on the azimuth angle of a sound source: Front $\in [-45°, 45°]$, Left $\in [45°, 135°]$, Back $\in [135°, \pm180°] \cup [\pm180°, -135°]$, Right $\in [-135°, -45°]$

was largely imbalanced, with far more sound sources in the front and right regions than in the left and back. Following [30], we hypothesize that localization performance on the real test dataset could be improved by balancing this distribution. To do so, we use a data augmentation technique known as Audio Channel Swapping (ACS) [31]. We perform 3 transformations involving azimuth to simulate the rotation of sound sources by 90°, 180°, and 270°. We performed different permutations of swapping and negating the X and Y of FOA channels directly. This simple augmentation strategy not only quadruples our overall dataset size but more importantly gives us a uniform distribution of azimuth angles. We show that this augmentation has a significant impact on localization performance in Table 1. Please refer to [31] for more details on ACS.

3.5. Evaluation metrics

We use the joint localization and detection metrics as defined by the DCASE 2022 Task 3 SELD Challenge in our proceeding analysis. The F-Score and error rate (ER) capture location-dependent detection. True Positives (TP) and False Positives (FP) are considered with a tolerance 20° in the direction of arrival. Class-dependent localization error (LE) and localization recall (LR) measure localization performance without considering the spatial threshold. See [32] for more details on SELD metrics.

4. RESULTS

4.1. A baseline model for FOA input

Prior to evaluating the impact of different input representations, we first assess the performance of a baseline model trained and evaluated on FOA input using varied training data configurations. The STARSS22 and SYNMIX dataset are both quite imbalanced in terms of distribution of sound source across azimuth angles. As described in Section 3.4, we use Audio Channel Swapping (ACS) to mitigate this problem and balance the distribution at train time.

Table 1 reports results for 5 training data configurations: **A**: training and evaluating only in azimuth using STARSS22 dataset; **B**: adding SYNMIX dataset to A's training; **C**: adding ACS augmentation to B's training, **B**$^{+E}$: training and evaluating B in both azimuth and elevation; **C**$^{+E}$: training and evaluating C in both azimuth and elevation. Note that B^{+E} and C^{+E} help us to understand the impact of removing elevation in the overall metrics. By comparing C^{+E} and C, we see how removing elevation improves all metrics, as one could imagine given less degree of freedom in the predictions. Moreover, we see an improvement in the joint localization and detection metrics across the board with the addition of the augmented data. Hence, we use **C** as our reference configuration to assess the impact of the input representation in proceeding sections.

Conf.	SELD ↓	ER↓	F ↑	LE ↓	LR ↑
A	0.65	0.73	15.3%	53.7°	27%
B	0.47	0.62	34.5%	22.5°	51%
C	0.42	0.56	43.3%	16.9°	54.1%
B^{+E}	0.53	0.70	27.3%	26.1°	47.5%
C^{+E}	0.48	0.62	33%	22.7°	51%

Table 1: Results with **FOA** input across different configurations; A: STARSS22; B: A + SYNMIX; C: B with ACS; B^{+E} and C^{+E}: B and C are trained and evaluated using both azimuth and elevation. Results are reported on the STARSS22 DCASE dev-test set. ↓ indicates metrics that are better when value is lower, ↑ viceversa.

4.2. Comparing audio input representations

Table 2 reports results when changing input representation, moving from the highly-privileged FOA representation, to binaural, and stereo audio. Our experiments show that as one moves from FOA to binaural and stereo, overall SELD model performance degrades. While this is to be expected because binaural and stereo audio are not designed to capture full spatial audio, this is the first quantification of deep learning-based SELD performance across these audio input representations on real multichannel recordings lays the groundwork for our deeper proceeding analysis.

4.3. Localization error by sound source quadrant

We are also interested in dissecting localization performance to understand where key success and failure points occur in terms of sound source position and polyphonic scene conditions.

In Figure 1, we show a set of confusion matrices illustrating the distribution of true quadrants of sound sources vs. predicted quadrants across audio input representations. We segment the 90°

Input	SELD ↓	ER ↓	F ↑	LE ↓	LR↑
FOA	0.42	0.56	43.3%	16.9°	54.1%
Binaural	0.50	0.67	33.9%	30.1°	49.2%
Stereo	0.60	0.76	21.7%	42.9°	38.8%

Table 2: Results for models trained using STARSS22 + SYNMIX using ACS, with different audio input representations. Results are reported on the STARSS22 DCASE development-test set. ↓ indicates metrics that are better when value is lower, ↑ viceversa.

quadrants as follows, based on azimuth angle: Front $\in [-45°, 45°]$, Left $\in [45°, 135°]$, Back $\in [135°, \pm180°] \cup [\pm180°, -135°]$, Right $\in [-135°, -45°]$. Notably, using the FOA representation, the model has near-perfect performance in terms of distinguishing front vs. back sources. In the binaural setting, we see an increase in front-back confusion, and in the stereo setting this error is glaring as 48% of sources in the front are predicted in the back quadrant. In fact, this is a well-studied topic in psychoacoustics related to the cone of confusion phenomenon, which occurs when a sound source is equidistant to both the left and right ears [33–35]. Thus, it is difficult for the listener to distinguish whether a sound source is in front or behind them. It is likely that our binaural model is affected by this as well. Across audio input representations, the accuracy of source detection in the left and right quadrants is fairly consistent, showing reliability in terms of lateral sound source detection given 2- or 4-channel audio input.

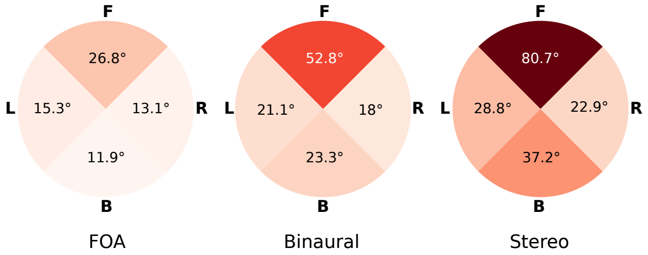


Figure 2: Average localization error across audio representations, based on ground truth sound source quadrant position. Results are normalized by number of instances of sound sources per quadrant.

In Figure 2, we analyze average localization error (LE) based on the quadrant of the ground truth sound sources. In the FOA setting, the difference of LE between the left, right, and back quadrants is quite small, however the error for sources in the front is nearly double that of the other quadrants. In the binaural setting, LE increases in the front and back quadrants, approximately doubling that of the FOA setting, though this increase is much less notable in the lateral (left-right) regions. Further, in the stereo context, we find similar trends but with overall poorer performance. The front and back LE are over three times that of the FOA model, with less significant degradation in the performance of the left and right quadrants. Here, we crucially observe that despite the binaural and stereo models struggling to localize sources in the front quadrant in particular compared to the FOA system, these 2-channel models are still able to localize sources laterally quite well.

4.4. SELD performance in polyphonic conditions

The DCASE SELD challenge is unique in that the test dataset contains real audio recordings with multiple overlapping sound sources. Hence, investigating SELD model performance in complex polyphonic conditions can help us better understand how these systems handle more complex scene conditions that are closer to reality. In Figure 3, we analyze localization recall (LR) of the FOA, binaural, and stereo models in the presence of 1, 2, 3, and 4 simultaneous sources (this encapsulates both simultaneous sources of the same or different classes). Note that approximately 56% of frames contain 1 source, 31% contain 2, 10% contain 3, and 3% contain 4 or more simultaneous sources, so we normalize by source count accordingly in Figure 3. We show that LR steadily decreases in all audio configurations as the number of polyphonic sound sources increases in Figure 3. The model struggles to detect the correct number of sources as the scene conditions become increasingly complex, though proportionally the decrease in recall is relatively similar across audio contexts as polyphony increases.

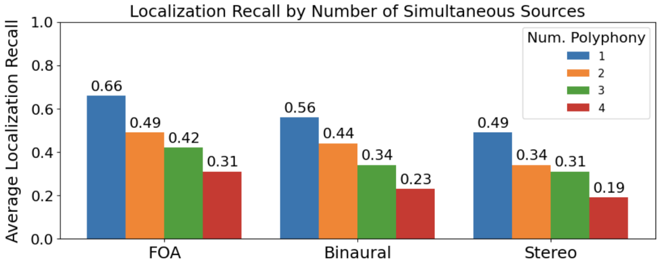


Figure 3: Localization recall in multiple audio representations, segmented by number of simultaneous sources in the test data and normalized by number of sources satisfying each condition.

We also analyze localization error (LE) across polyphonic conditions. Here we find that while on average LE increases as we use less-informative audio representations (i.e. stereo), it is not a fully monotonically increasing trend across polyphonic conditions. In the FOA setting, the LE is similar regardless of level of polyphony. In the binaural and stereo settings, there is a much larger spread of LE across conditions, however not in a monotonically increasing manner, e.g. in the stereo setting the average LE is 31.3° in the occurrence of 3 overlapping sources vs. 46.1° for 2 sources. We hypothesize that there are many interacting effects contributing to this, including but not limited to class imbalance in different polyphonic conditions, simultaneous sources of the same class, and the nature of the LE metric as it does not take false negatives into account.

5. CONCLUSION

This work presents a novel comparative analysis of the DCASE 2022 SELD baseline model across first-order Ambisonics, binaural, and stereo audio input representations. We show quantitatively that while localization and detection performance decreases given less informative audio representations, binaural and stereo-based SELD models are still able to localize lateral sound sources relatively well. These findings could be highly informative in the development of applications such as an audio-visual navigation system equipped with a stereo microphone configuration and a camera; if we are confident in lateral source localization based on auditory cues, we can lean more on visual cues for sources directly in front of the camera. Future work in this space could entail an investigation into the effect of sound source class or of overlapping sources of the same class on localization performance across polyphonic conditions and audio input representations.

6. REFERENCES

[1] J. H. DiBiase, *A high-accuracy, low-latency technique for talker localization in reverberant environments using microphone arrays*. Brown University, 2000.

[2] L. Perotin, R. Serizel, E. Vincent, and A. Guérin, "Crnn-based joint azimuth and elevation localization with the ambisonics intensity vector," in *2018 16th International Workshop on Acoustic Signal Enhancement (IWAENC)*. IEEE, 2018, pp. 241–245.

[3] R. Stern, D. Wang, and G. Brown, "Binaural sound localization-chapter 5 from computational auditory scene analysis," 2006.

[4] C. Giguère and S. M. Abel, "Sound localization: Effects of reverberation time, speaker array, stimulus frequency, and stimulus rise/decay," *The Journal of the Acoustical Society of America*, vol. 94, no. 2, pp. 769–776, 1993.

[5] S. Hafezi, A. H. Moore, and P. A. Naylor, "Augmented intensity vectors for direction of arrival estimation in the spherical harmonic domain," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 10, pp. 1956–1968, 2017.

[6] C. Evers, A. H. Moore, and P. A. Naylor, "Multiple source localisation in the spherical harmonic domain," in *2014 14th International Workshop on Acoustic Signal Enhancement (IWAENC)*. IEEE, 2014, pp. 258–262.

[7] S. Adavanne, A. Politis, J. Nikunen, and T. Virtanen, "Sound event localization and detection of overlapping sources using convolutional recurrent neural networks," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 1, pp. 34–48, March 2018. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8567942

[8] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu, *et al.*, "Conformer: Convolution-augmented transformer for speech recognition," *arXiv preprint arXiv:2005.08100*, 2020.

[9] F. Zhao, R. Li, and D. Pan, "Deep learning for binaural sound source localization with low signal-to-noise ratio," *Journal of Physics: Conference Series*, vol. 1828, p. 012017, 02 2021.

[10] C. Gan, H. Zhao, P. Chen, D. Cox, and A. Torralba, "Self-supervised moving vehicle tracking with stereo sound," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[11] P.-A. Grumiaux, S. Kiti'c, L. Girin, and A. Gu'erin, "A survey of sound source localization with deep learning methods," *The Journal of the Acoustical Society of America*, vol. 152 1, p. 107, 2021.

[12] C. Knapp and G. Carter, "The generalized correlation method for estimation of time delay," *IEEE transactions on acoustics, speech, and signal processing*, vol. 24, no. 4, pp. 320–327, 1976.

[13] W. He, P. Motlicek, and J.-M. Odobez, "Deep neural networks for multiple speaker detection and localization," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 74–79.

[14] S. Adavanne, A. Politis, and T. Virtanen, "Direction of arrival estimation for multiple sound sources using convolutional recurrent neural network," in *2018 26th European Signal Processing Conference (EUSIPCO)*. IEEE, 2018, pp. 1462–1466.

[15] F. L. Wightman and D. J. Kistler, "Headphone simulation of free-field listening. ii: Psychophysical validation," *The Journal of the Acoustical Society of America*, vol. 85, no. 2, pp. 868–878, 1989.

[16] J. Blauert, *Spatial Hearing: The Psychophysics of Human Sound Localization*. MIT Press, 1997.

[17] T. Rudzki, I. Gomez-Lanzaco, J. Stubbs, J. Skoglund, D. T. Murphy, and G. Kearney, "Auditory localization in low-bitrate compressed ambisonic scenes," *Applied Sciences*, vol. 9, no. 13, p. 2618, 2019.

[18] L. Thresh, C. Armstrong, and G. Kearney, "A direct comparison of localization performance when using first, third, and fifth ambisonics order for real loudspeaker and virtual loudspeaker rendering," in *Audio Engineering Society Convention 143*. Audio Engineering Society, 2017.

[19] S. Adavanne, A. Politis, and T. Virtanen, "Multichannel sound event detection using 3d convolutional neural networks for learning inter-channel features," in *2018 international joint conference on neural networks (IJCNN)*. IEEE, 2018, pp. 1–7.

[20] N. Poschadel, S. Preihs, and J. Peissig, "Multi-source direction of arrival estimation of noisy speech using convolutional recurrent neural networks with higher-order ambisonics signals," in *2021 29th European Signal Processing Conference (EUSIPCO)*, 2021, pp. 1015–1019.

[21] K. Shimada, Y. Koyama, S. Takahashi, N. Takahashi, E. Tsunoo, and Y. Mitsufuji, "Multi-accdoa: Localizing and detecting overlapping sounds from the same class with auxiliary duplicating permutation invariant training," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Singapore, Singapore, May 2022.

[22] M. A. Gerzon, "Periphony: With-height sound reproduction," *Journal of the audio engineering society*, vol. 21, no. 1, pp. 2–10, 1973.

[23] I. Engel, D. F. Goodman, and L. Picinali, "Assessing hrtf preprocessing methods for ambisonics rendering through perceptual models," *Acta Acustica*, vol. 6, p. 4, 2022.

[24] A. Politis, K. Shimada, P. Sudarsanam, S. Adavanne, D. Krause, Y. Koyama, N. Takahashi, S. Takahashi, Y. Mitsufuji, and T. Virtanen, "Starss22: A dataset of spatial recordings of real scenes with spatiotemporal annotations of sound events," 2022. [Online]. Available: https://arxiv.org/abs/2206.01948

[25] E. Fonseca, X. Favory, J. Pons, F. Font, and X. Serra, "Fsd50k: an open dataset of human-labeled sound events," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 829–852, 2021.

[26] A. Politis, S. Adavanne, and T. Virtanen, "A dataset of reverberant spatial sound scenes with moving sources for sound event localization and detection," 2020. [Online]. Available: https://arxiv.org/abs/2006.01919

[27] A. Politis, S. Adavanne, D. Krause, A. Deleforge, P. Srivastava, and T. Virtanen, "A dataset of dynamic reverberant sound scenes with directional interferers for sound event localization and detection," 2021. [Online]. Available: https://arxiv.org/abs/2106.06999

[28] C. Schörkhuber, M. Zaunschirm, and R. Höldrich, "Binaural rendering of ambisonic signals via magnitude least squares," in *Proceedings of the DAGA*, vol. 44, 2018, pp. 339–342.

[29] F. Zotter and M. Frank, *XY, MS, and First-Order Ambisonics*. Cham: Springer International Publishing, 2019, pp. 1–22. [Online]. Available: https://doi.org/10.1007/978-3-030-17207-7_1

[30] Q. Wang, L. Chai, H. Wu, Z. Nian, S. Niu, S. Zheng, Y. Wang, L. Sun, Y. Fang, J. Pan, J. Du, and C.-H. Lee, "The nerc-slip system for sound event localization and detection of dcase2022 challenge," DCASE2022 Challenge, Tech. Rep., June 2022.

[31] Q. Wang, J. Du, H.-X. Wu, J. Pan, F. Ma, and C.-H. Lee, "A four-stage data augmentation approach to resnet-conformer based acoustic modeling for sound event localization and detection," 2021. [Online]. Available: https://arxiv.org/abs/2101.02919

[32] A. Politis, A. Mesaros, S. Adavanne, T. Heittola, and T. Virtanen, "Overview and evaluation of sound event localization and detection in dcase 2019," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 684–698, 2020. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9306885

[33] L. Rayleigh, "Xii. on our perception of sound direction," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 13, no. 74, pp. 214–232, 1907.

[34] O. Balan, A. Moldoveanu, and F. Moldoveanu, "A systematic review of the methods and experiments aimed to reduce front-back confusions in the free-field and virtual auditory environments." *RoCHI*, pp. 24–29, 2018.

[35] B. G. Shinn-Cunningham, S. Santarelli, and N. Kopco, "Tori of confusion: Binaural localization cues for sources within reach of a listener," *The Journal of the Acoustical Society of America*, vol. 107, no. 3, pp. 1627–1636, 2000.

PLDISET: PROBABILISTIC LOCALIZATION AND DETECTION OF INDEPENDENT SOUND EVENTS WITH TRANSFORMERS

Peipei Wu[1], *Jinzheng Zhao*[1], *Yaru Chen*[1], *Berghi Davide*[1], *Yi Yuan*[1], *Chenfei Zhu*[2],
Yin Cao[3], *Yang Liu*[4], *Philip J.B. Jackson*[1], *Mark D. Plumbley*[1], *Wenwu Wang*[1]

[1] University of Surrey, Guildford, UK,
[2] Daqian Information, Wuhan, China,
[3] Xi'an Jiaotong-Liverpool University, Suzhou, China,
[4] Meta, Seattle, USA,

ABSTRACT

Sound Event Localization and Detection (SELD) is a task that involves detecting different types of sound events along with their temporal and spatial information, specifically, detecting the classes of events and estimating their corresponding direction of arrivals at each frame. In practice, real-world sound scenes might be complex as they may contain multiple overlapping events. For instance, in DCASE challenges task 3, each clip may involve simultaneous occurrences of up to five events. To handle multiple overlapping sound events, current methods prefer multiple output branches to estimate each event, which increases the size of the models. Therefore, current methods are often difficult to be deployed on the edge of sensor networks. In this paper, we propose a method called Probabilistic Localization and Detection of Independent Sound Events with Transformers (PLDISET), which estimates numerous events by using one output branch. The method has three stages. First, we introduce the track generation module to obtain various tracks from extracted features. Then, these tracks are fed into two transformers for sound event detection (SED) and localization, respectively. Finally, one output system, including a linear Gaussian system and regression network, is used to estimate each track. We give the evaluation results of our model on DCASE 2023 Task 3 development dataset.

***Index Terms*—** Sound Event Localization and Detection, Transformer, Linear Gaussian System

1. INTRODUCTION

Currently, applications in various fields, such as robotics and surveillance, rely on Sound Event Localization and Detection (SELD) technology. Therefore, conducting in-depth research on this topic is crucial. Since 2019, DCASE has been hosting relevant challenges that have significantly improved SELD systems [1, 2].

The first notable method in SELD is SELDNet [3]. However, it is limited in dealing with multiple overlapping events from the same class with different locations. To address this issue, EINv2 introduced a new track-wise output format [4]. Since then, Permutation-Invariant Training (PIT) has been utilized in SELD [5], which forms part of the baseline system used in DCASE 2023 Task 3. However, EINv2 still requires multiple output branches to estimate the corresponding track, which increases the model's size. Especially if the number of overlapping events is higher than the number of output branches, EINv2 cannot predict all events simultaneously. In other words, some events might be ignored.

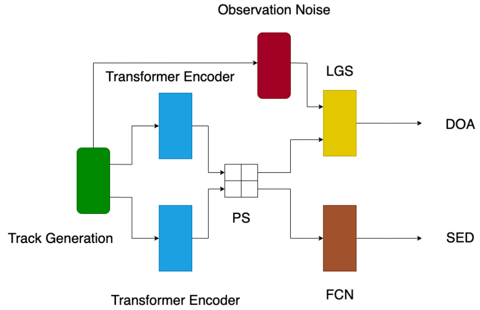


Figure 1: The new output branch for SELD. PS, LGS, and FCN denote parameter-sharing, linear Gaussian systems, and fully-connected networks, respectively.

Building upon previous work, this paper presents a novel three-stage solution for SELD. First, in contrast to EINv2, we generate different tracks from the extracted features prior to the attention module. Second, we employ a transformer instead of a simple convolutional recurrent neural network (CRNN) in SELD. Third, we introduce a linear Gaussian system to predict the Direction of Arrival (DOA) from each track rather than relying on regression networks. It is worth noting that in EINv2, the number of output branches is double the number of tracks, as each track requires separate output networks for DOA and SED predictions, respectively. If the number of tracks is large, this can pose challenges for EINv2, whereas our proposed model handles this efficiently.

In the following Section 2, we review the related work which we used in our proposed method. Section 3 introduces the proposed method in detail. Section 4 showcases the experimental results along with their corresponding analysis. The last section concludes our contribution and future work.

2. RELATED WORKS

2.1. Trackwise output format

This format type is first introduced in [4]. It can be defined as:

$$\boldsymbol{Y}_{\text{Trackwise}} = \{(y_{\text{SED}}, y_{\text{DOA}} | (y_{\text{SED}} \in \mathbb{1}_{\mathbf{S}}^{M\times K}, y_{\text{DOA}} \in \mathbb{R}^{M\times 3})\} \quad (1)$$

where y_{SED} and y_{DOA} are predictions for SED and DOA, respectively, $\mathbb{1}$ denotes one-hot encoding, M is the number of tracks, K

is the number of classes, $\mathbf{S}$ is the set of sound event classes, and $\mathbb{R}^{M\times 3}$ represents spatial information by Cartesian coordinates.

However, this format type can lead to a track permutation problem. In most cases, $M \ll K$ indicates that not all classes of sound events happen in each frame. In other words, events are not consistently predicted in fixed tracks. As a result, in the training process, tracks do not know which ground truths are corresponded to themselves correctly. To address this issue, permutation-invariant training is employed as a solution.

2.2. Permutation-Invariant training

Permutation-invariant Training was first introduced for speaker separation in [5]. Let t denote the frame index. Given a frame-level permutation set $\mathbf{P}(t)$, which consists of all possible prediction-label pairs, ground truth labels are assigned based on the possible combinations within this set of pairs. The lowest loss is then used for backpropagation. The PIT loss can be defined as follows:

$$\mathcal{L}^{\mathrm{PIT}} = \min_{\alpha\in\mathbf{P}(t)} \sum_{M} \{\ell_{\alpha}^{\mathrm{SED}}(t) + \ell_{\alpha}^{\mathrm{DOA}}(t)\} \tag{2}$$

where α is one of the possible prediction-label pair, $\ell_{\alpha}^{\mathrm{SED}}(t)$ and $\ell_{\alpha}^{\mathrm{DOA}}(t)$ are SED and DOA loss, respectively.

2.3. Linear-Gaussian system

The linear Gaussian system represents a linear relationship between variables, where the observed variables are corrupted by Gaussian noise. This modeling approach has been widely utilized in various tasks, including detection or tracking tasks. A simple linear Gaussian system can be described by the following equation:

$$y = \mathbf{H}x + \omega \tag{3}$$

where y represents the observed state, x represents the latent state (which is hidden), $\mathbf{H}$ is the observation matrix, and ω represents the observation noise. A more complex version of the linear Gaussian system can refer to the Bayesian filters, involving parameter optimization, such as Kalman Filter [6].

3. THE PROPOSED METHOD

In this section, we will discuss the proposed method in detail. Firstly, we introduce parameter-sharing (PS) technology to enable multi-task learning. Then, we discuss the network in three stages: Feature Extraction, Transformer, and Tracks Estimation. At last, we will give a summary of the proposed method's structure.

3.1. Parameter-Sharing

Due to SELD involving both sound event detection and corresponding localization, this task is considered a complex multi-task rather than a single task. Therefore, joint SELD learning can benefit from multi-task learning (MTL) [7]. Considering that SED and DOA predictions have different noise patterns, a good representation F can average the noise patterns from both sides. Additionally, certain features R in F may be easily obtained from one side (SED or DOA) but difficult from the other side. MTL can aid in obtaining a good representation F.

Parameter-sharing (PS) is a classical MTL method, including soft PS and hard PS [8]. The comparison between soft PS, hard PS, and no PS can be seen in [4]. Thanks to their work, in this paper, we select soft PS directly. The cross-stitch is used for soft PS. Let D_c, D_t, and D_f denote the dimensions of feature maps, time steps, and frequency, respectively. The learnable parameters are denoted as $\delta_{i,j} \in \mathbb{R}^{D_c}$. From the original feature maps $(x^{\mathrm{SED}}, x^{\mathrm{DOA}})$, the new feature map updated by cross-stitch is given as:

$$[\hat{x}^{\mathrm{SED}}, \hat{x}^{\mathrm{DOA}}]^{\mathrm{T}} = \mathbf{\Delta}[(x^{\mathrm{SED}}, x^{\mathrm{DOA}})]^{\mathrm{T}} \tag{4}$$

where $\hat{x}^{\mathrm{SED}}, \hat{x}^{\mathrm{DOA}} \in \mathbb{R}^{D_c\times D_t\times D_f}$ is the new feature map, $\mathbf{\Delta}$ is a matrix with the dimension of 2×2 consisting the learnable parameters, and T means transpose operation.

3.2. Feature Extraction

The first stage, Feature Extraction, includes a CNN-based feature extractor, the track, and the observation noise generation module. The primary objective of this stage is to obtain feature embedding and observation noise.

SELDnet introduces a three-layer CNN-based feature extractor, but its simple structure is considered less sensitive to small-sized features. Moreover, SELDnet didn't provide extractors for SED and DOA branches separately. As a result, it might ignore some specific features R in F, as discussed earlier. Therefore, this simple extractor is not ideal for joint SELD learning. We adopted the extractor from EINv2 [4] directly. Same we provide different inputs for SED and DOA extractors. Only the DOA extractor will be applied observation noise generation module.

Afterward, we generate M tracks from feature embeddings, where M is a fixed input value. Therefore, we design a fully-connected network (FCN) to implement. First, two embeddings are flattened. Then, a linear layer is designed to increase the dimension M times. Last, we reshape the embedding and obtain M tracks. Also, the cross-stitch method is applied to the FCN.

Considering that the linear Gaussian system (LGS) is only applied to the direction of arrival (DOA) branch, we solely adopt the observation noise module for the DOA's feature map. The observation noise module consists of a linear layer to convert the feature map into the observation state noise dimension (2-D or 3-D, depending on the requirements).

3.3. Transformer

The Transformer was first proposed in [9], and we adopted it for handling temporal information. We design separate Transformers for SED and DOA, similar to the previous stage. Considering Transformer requires input with positional information. Thus, we apply a fixed absolute positional encoding on each track as follows:

$$P_{t,2i} = 0.1\sin\left(t/10^{8i/D_c}\right), \quad P_{t,2i+1} = 0.1\sin\left(t/10^{8i/D_c}\right), \tag{5}$$

where t represents the time step and i denotes the feature map index. Then, the positional encoded features will be fed into the Transformer's encoder. Each encoder layer contains 8 multi-head self-attention structures, and the input embedding dim is 512. Between each encoder layer, soft PS is applied to balance the gap between SED and DOA's representations. The entire Transformer consists of two encoder layers.

3.4. Tracks Estimation

The last stage, Tracks Estimation, aims to estimate SED and DOA in each track. In EINv2, each track has two FCNs to estimate SED and DOA. If there is more than one track, EINv2 needs to add more FCNs to cover the additional tracks. For instance, if there are three tracks, EINv2 will need 6 FCNs to cover all estimations. Different from it, we design the re-useable estimation block to cover inputs from different tracks to estimate the SED and the DOA of each track.

For SED estimation, we employ a regression method to obtain. The transformer's output is fed into FCN and activated by the sigmoid function. As for the DOA estimation, we adopt the linear Gaussian system (LGS) to calculate the posterior estimation. The calculation process is as follows:

$$\mathbf{I} = \mathbf{H}\mathbf{E}\mathbf{H}^{\mathrm{T}} + \mathbf{N}_{\mathrm{o}} \tag{6}$$

Here, $\mathbf{I}$ represents the innovation covariance matrix, $\mathbf{H}$ is the observation matrix as defined in Equation 3, $\mathbf{E}$ denotes the identity matrix, and $\mathbf{N}_{\mathrm{o}}$ is the output from the observation noise module. The observation noise is obtained by passing the observation embeddings (with a dimension of 512) through a linear layer. This projection maps the observation embeddings to the state embedding, which has a dimension of 3. The posterior covariance matrix $\mathbf{C}_{\mathrm{p}}$ is then obtained as:

$$\mathbf{C}_{\mathrm{p}} = (\mathbf{E}^{-1} + \mathbf{I})^{-1} \tag{7}$$

where $[\cdot]^{-1}$ denotes the inverse operation. Next, the residual matrix $\mathbf{R}$ is calculated as:

$$\mathbf{R} = \mathbf{H}(\mathbf{x} - \mathbf{B}_{\mathrm{o}}) \tag{8}$$

where $\mathbf{x}$ represents the state embedding transferred from observation embedding, and $\mathbf{B}_{\mathrm{o}}$ is the bias in the observation model. Finally, the DOA estimation, also known as the posterior mean matrix, is obtained as follows:

$$\hat{\mathbf{x}}^{\mathrm{DOA}} = \mathbf{C}_{\mathrm{p}}\mathbf{E}^{-1} + \mathbf{R}. \tag{9}$$

3.5. PLDISET and loss function

In the previous section, we discussed Permutation Invariant Training (PIT) but did not provide detailed information about the loss functions for sound event detection (SED) and direction of arrival (DOA). In this subsection, we will explain the loss functions and provide an overview of the PLDISET method.

We select Binary Cross Entropy (BCE) as the loss function for the SED task, which is a classification task. It measures the cross-entropy between the predictions and the labels for SED. For the DOA task, the evaluation is based on the distance between the estimations and the ground truths. Since Cartesian coordinates are introduced, we can use the mean squared error between two points in Cartesian coordinates as the loss function for DOA.

To train the SELD model and optimize its performance in both SED and DOA tasks, these loss functions are used. The overall loss is computed by summing the individual losses for SED and DOA with appropriate weights.

The overview of the PLDISET is depicted in Figure 2. For the sound event detection (SED) task, we use log mel spectrogram as the input feature. In the case of the direction of arrival (DOA) task, both log mel spectrogram and intensity vector map are selected as the input features.

4. EXPERIMENT AND EVALUATION

4.1. Dataset and data augmentation

The DCASE 2023 development dataset consists of multichannel recordings of sound scenes captured in different rooms and environments. The dataset includes temporal and spatial annotations for prominent events belonging to a set of target classes. The total duration of the dataset is 7.5 hours. However, due to the limited size of the dataset, it is insufficient to train a competitive deep-learning-based model. To overcome this limitation, we utilized the simulation generator script provided by the DCASE 2022 challenge to generate an additional 30 hours of recordings. The generated dataset includes two versions: a noiseless version and a noisy version.

4.2. Metrics

We use the DCASE challenge's metrics to evaluate our method. The evaluation metrics used in this challenge are based on true positives (TP) and false positives (FP), taking into account not only correct or wrong detections but also the proximity to a distance threshold T° (angular threshold in our case) from the reference. For this challenge, the threshold is set to $T = 20^{\circ}$. The details can be seen in [10, 11, 12].

4.3. Hyper-Parameters

We apply the Fast Fourier Transform (FFT) on the recordings using a 1024-point Hann window with a hop size of 600 points. To extract the log-mel spectrogram from the FFT result, we select 256 mel bands. Next, we segment the audio clips into chunks of a fixed length of 4 seconds without overlapping. The intensity vector map is obtained as well.

For model training, we utilize the AdamW optimizer for optimization. The initial learning rate is set to 0.0005 for the first 80 epochs and is then reduced to 0.00005 for the subsequent 10 epochs. During the finetuning of the model, the scheduler strategy changes to use a learning rate of 0.0005 for the first 10 warm-up epochs. Afterward, the learning rate is multiplied by 0.1 every 10 epochs. The weighted term for the Permutation Invariant Training (PIT) loss is selected as 0.5 for both the sound event detection (SED) and direction of arrival (DOA) losses.

4.4. Baseline system

We evaluate our proposed method by comparing it to the baseline system (SELDNet) provided by the DCASE challenge, which has been widely used as a benchmark [3, 13, 14, 15]. The baseline system extends the original SELDNet [3] by introducing multi-head self-attention blocks, using the Multi-ACCDOA output format, and employing SALSA-lite features to handle multiple overlapping sound events. Furthermore, we add EINv2 for comparison as well.

4.5. Evaluation

We compare the proposed method with the baseline and EINv2 in three steps. First, we trained all three algorithms on the noiseless dataset using respective default settings. Table 1 shows their performances. On the SED task, PLDISET and EINv2 achieved similar performance and much better than the baseline. As for the DOA

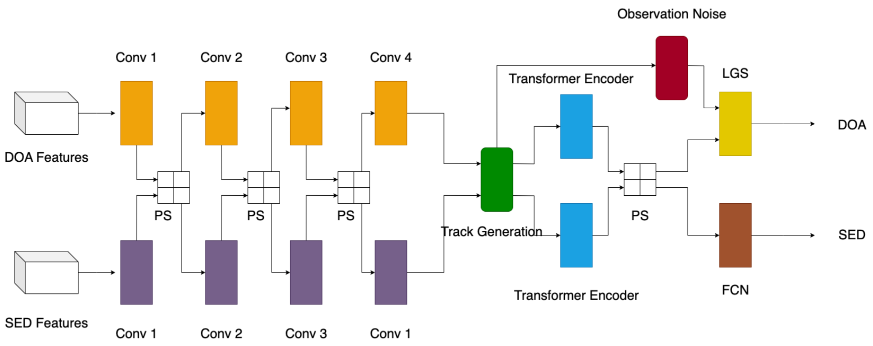


Figure 2: Network Architecture of the PLDISET.

task, PLDISET is slightly lower than the baseline, with a minor gap.

Table 1: Official metrics on the noiseless generated dataset

Methods	ER_{20}	F_{20}	LE_{CD}	LR_{CD}
baseline	0.52	49.2	18.8	58.9
EINv2	0.36	55.7	**11.3**	**79.8**
PLDISET	**0.35**	**56.1**	19.1	58.1

Afterward, we finetuned the models from the first step on the noisy datasets. The evaluation results on the test dataset are provided in Table 2. EINv2 performed best on both tasks. The proposed method achieved similar results on the SED task and was not far from the baseline on the DOA task.

Table 2: Official metrics on the noisy generated datasets

Methods	ER_{20}	F_{20}	LE_{CD}	LR_{CD}
baseline	0.55	48.9	20.0	49.9
EINv2	**0.38**	**52.5**	**13.1**	**75.2**
PLDISET	**0.38**	52.1	21.5	47.1

In the last step, we evaluated those methods on the development dataset of the DCASE Challenge 2023. We finetuned models from previous steps on the training part. Table 3 demonstrates the results on the evaluation set. The proposed method and EINv2 outperform well on the SED task with an error rate of around 0.39. The performance of PLDISET on the DOA task is close to the baseline.

Table 3: Official metrics on the DCASE development dataset

Methods	ER_{20}	F_{20}	LE_{CD}	LR_{CD}
baseline	0.57	48.7	22.0	47.7
EINv2	**0.38**	**53.3**	**14.5**	**72.4**
PLDISET	0.39	52.6	23.6	47.4

The proposed method shows its advantages on the SED task in the three comparisons, with the lowest error rate of 0.35 and the highest of 0.39. Considering that some datasets consist of real-world recordings that are more challenging than the simulated data, the proposed method shows its excellent capability in handling the SED tasks under different complex scenarios. As for the DOA task, unlike other works, we adopt a probabilistic method for localization instead of a regression-based approach. However, the PLDISET method shows a gap in the DOA task compared to EINv2 and the baseline. The possible reason for the disadvantage is the LGS may result in lower accuracy in the DOA estimations due to inaccurate prior information or an inappropriate model.

Compared to other works, one of the distinguishing features of PLDISET is its ability to estimate all tracks using a single output branch. For most methods, they require assigning output modules for each track. But PLDISET can reuse the output module for each track. The experimental results demonstrate that PLDISET performs well in SED tasks, showing its strong ability to accurately detect and classify sound events without multiple regression networks. Although the localization ability may not be as refined as in some other works, it still achieves satisfactory results. Overall, PLDISET balances sound event detection and localization tasks well. Considering that the parameters of the LGS can be updated and constrained by certain rules, there are potential research prospects in further exploring and refining this aspect. By improving the prior information and refining the model, it may be possible to enhance the accuracy of DOA estimations in the PLDISET method. Besides that, PLDISET shows its prospects of extending into a tracking version. In tracking problems, different numbers of targets appear in each frame which is quite common. Currently, PLDISET reuses the single output branch to cover all tracks, which can be improved to handle different tracks input. In addition, temporal information can be considered in the tracking problem. Therefore, some historical information, such as the Kalman Filter decreasing the error by regression in the transaction, can be used to adjust the LGS to improve tracking accuracy.

5. CONCLUSION AND FUTURE WORK

In this study, we introduced a novel network called PLDISET for SELD. We design the new output branch to estimate all tracks rather than create several branches for each track. The proposed method is evaluated on three datasets by comparing the baseline and EINv2 to show its advantages and potential. The source code and improving work based on the proposed method for sound event tracking will be released in the future.

6. REFERENCES

[1] W.-G. Choi and J.-H. Chang, "Confidence regularized entropy for polyphonic sound event detection," in *Proceedings of the 7th Detection and Classification of Acoustic Scenes and Events 2022 Workshop (DCASE2022)*, Nancy, France, November 2022.

[2] J. Hu, Y. Cao, M. Wu, Q. Kong, F. Yang, M. D. Plumbley, and J. Yang, "Sound event localization and detection for real spatial sound scenes: Event-independent network and data augmentation chains," in *Proceedings of the 7th Detection and Classification of Acoustic Scenes and Events 2022 Workshop (DCASE2022)*, Nancy, France, November 2022.

[3] S. Adavanne, A. Politis, J. Nikunen, and T. Virtanen, "Sound event localization and detection of overlapping sources using convolutional recurrent neural networks," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 1, pp. 34–48, 2019.

[4] Y. Cao, T. Iqbal, Q. Kong, F. An, W. Wang, and M. D. Plumbley, "An Improved Event-Independent Network for Polyphonic Sound Event Localization and Detection," *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 00, pp. 885–889, 2021.

[5] D. Yu, M. Kolbæk, Z.-H. Tan, and J. Jensen, "Permutation invariant training of deep models for speaker-independent multi-talker speech separation," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 241–245.

[6] G. Bishop, G. Welch, *et al.*, "An introduction to the kalman filter," *Proc of SIGGRAPH, Course*, vol. 8, no. 27599-23175, p. 41, 2001.

[7] P. Vafaeikia, K. Namdar, and F. Khalvati, "A brief review of deep multi-task learning and auxiliary task learning," *arXiv*, 2020.

[8] D. S. Sachan and G. Neubig, "Parameter sharing methods for multilingual self-attentional translation models," *arXiv*, 2018.

[9] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *arXiv*, 2017.

[10] A. Politis, A. Mesaros, S. Adavanne, T. Heittola, and T. Virtanen, "Overview and evaluation of sound event localization and detection in dcase 2019," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 684–698, 2020. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9306885

[11] A. Mesaros, S. Adavanne, A. Politis, T. Heittola, and T. Virtanen, "Joint measurement of localization and detection of sound events," in *2019 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2019, pp. 333–337.

[12] A. Politis, A. Mesaros, S. Adavanne, T. Heittola, and T. Virtanen, "Overview and evaluation of sound event localization and detection in dcase 2019," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 684–698, 2021.

[13] K. Shimada, Y. Koyama, S. Takahashi, N. Takahashi, E. Tsunoo, and Y. Mitsufuji, "Multi-accdoa: Localizing and detecting overlapping sounds from the same class with auxiliary duplicating permutation invariant training," *arXiv*, 2022.

[14] T. N. T. Nguyen, D. L. Jones, K. N. Watcharasupat, H. Phan, and W.-S. Gan, "SALSA-lite: A fast and effective feature for polyphonic sound event localization and detection with microphone arrays," in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, may 2022. [Online]. Available: https://doi.org/10.1109%2Ficassp43922.2022.9746132

[15] P. Sudarsanam, A. Politis, and K. Drossos, "Assessment of self-attention on learned features for sound event localization and detection," 2021.

CROWDSOURCING AND EVALUATING TEXT-BASED AUDIO RETRIEVAL RELEVANCES

Huang Xie, Khazar Khorrami, Okko Räsänen, Tuomas Virtanen

Unit of Computing Sciences, Tampere University, Finland

ABSTRACT

This paper explores grading text-based audio retrieval relevances with crowdsourcing assessments. Given a free-form text (e.g., a caption) as a query, crowdworkers are asked to grade audio clips using numeric scores (between 0 and 100) to indicate their judgements of how much the sound content of an audio clip matches the text, where 0 indicates no content match at all and 100 indicates perfect content match. We integrate the crowdsourced relevances into training and evaluating text-based audio retrieval systems, and evaluate the effect of using them together with binary relevances arise from audio captioning. Conventionally, these binary relevances are defined by captioning-based audio-caption pairs, where being positive indicates that the caption describes the paired audio, and being negative applies to all other pairs. Experimental results indicate that there is no clear benefit from incorporating crowdsourced relevances alongside binary relevances when the crowdsourced relevances are binarized for contrastive learning. Conversely, the results suggest that using only binary relevances defined by captioning-based audio-caption pairs is sufficient for contrastive learning.

Index Terms— Text-based audio retrieval, non-binary relevance, crowdsourcing assessment

1. INTRODUCTION

Text-based audio retrieval, or text-to-audio retrieval, refers to searching for audio clips with free text queries, which has great potential in real-world applications, such as search engines and multimedia databases. Early works [1, 2] have mainly focused on methods of retrieving audio with carefully curated phrases (e.g., audio tags, onomatopoeic words). With the availability of large audio-caption datasets (e.g., Clotho [3] and AudioCaps [4]) in recent years, increasing attention has been drawn to developing methods for audio retrieval using free-form text [5].

Most of the literature tackles text-to-audio retrieval with cross-modal learning methods. Oncescu *et al.* [6] first established benchmarks in this topic with an adapted text-to-video retrieval model. With the recent success of large-scale pretrained audio models (e.g., PANNs [7]) and language models (e.g., BERT [8]), pretrained models are widely used for text-to-audio retrieval and fine-tuned on task-specific data to learn joint representations of audio and text [5]. Besides, several works [9, 10] explored training strategies for text-to-audio retrieval models. For example, Mei *et al.* [9] evaluated several cross-modal learning objectives (e.g., InfoNCE loss [11]) in the context of text-to-audio retrieval.

In aforementioned works [6, 9, 10], relevances of audio to be retrieved are usually assumed to be binary, i.e., either relevant or irrelevant, given a text query. For example, an audio clip is relevant to a caption if and only if the caption thoroughly describes its sound content (i.e., their content matches perfectly). Practically, due to the lack of annotated non-binary relevances in existing datasets, binary relevances defined by captioning-based audio-caption pairs are adopted for system training and evaluation. Binary relevances are positive for audio-caption pairs where the caption describes the paired audio, and negative for all other pairs. This allows producing large quantities of positive and negative examples for contrastive learning in state-of-the-art systems [5].

As a caption might partially describe the sound content of an audio clip, we explore grading audio-text relevances with non-binary numerical scores. Specifically, we crowdsource audio-text relevances graded on a scale of 0 to 100, where 0 indicates completely irrelevant (i.e., no content match at all) and 100 indicates completely relevant (i.e., perfect content match). The main contributions of this work are: 1) we crowdsource non-binary audio-text relevances for environmental audio and its existing captions; 2) we integrate the crowdsourced relevances into training and evaluating text-to-audio retrieval systems, and evaluate the effect of using them alongside binary relevances defined by captioning-based audio-caption pairs; 3) we release all the data and the crowdsourcing instructions to the research community to allow others to explore non-binary relevances [12].

2. CROWDSOURCING ASSESSMENTS

This section presents the proposed method for crowdsourcing audio-text relevances.

2.1. Crowdsourcing Pipeline

We crowdsource audio-text relevances on Amazon Mechanical Turk (MTurk). Here we first introduce the terms used in this work. A *human intelligence task* (HIT) represents a single task that a crowdworker can work on. An *assignment* is a copy of a HIT that is assigned to a crowdworker. A worker *answer* is the submitted task result when a crowdworker completes an assignment.

Fig. 1 presents an overview of the crowdsourcing pipeline. Audio clips and captions for relevance assessments are combined to form HITs, each of which consists of five audio clips and one caption. Every HIT is assigned to multiple MTurk workers. Raw answers containing graded audio-text relevances are collected and aggregated after workers complete their assignments.

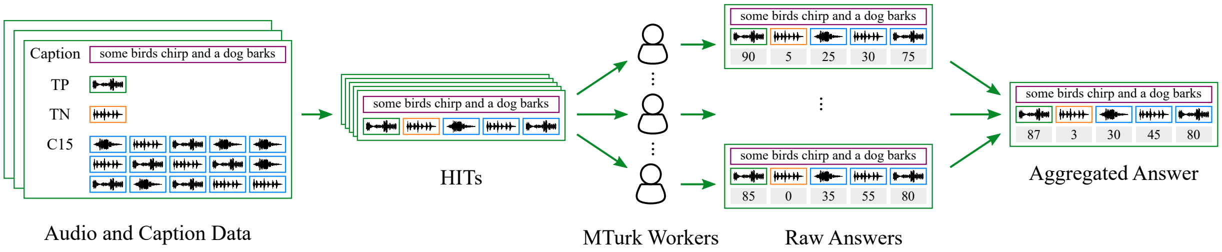

Caption
some birds chirp and a dog barks
TP
TN
C15
Audio and Caption Data
some birds chirp and a dog barks
HITs
MTurk Workers
some birds chirp and a dog barks
90 5 25 30 75
some birds chirp and a dog barks
85 0 35 55 80
Raw Answers
some birds chirp and a dog barks
87 3 30 45 80
Aggregated Answer

Figure 1: An overview of the pipeline for crowdsourcing relevance assessments.

Split	#Captions	#Audio		
		#TPs	#TNs	#C15s
development	200	200	200	3000
validation	200	200	200	3000
evaluation	200	200	200	3000

Table 1: Statistics of captions and audio clips selected for crowdsourcing relevance assessments.

2.2. Audio and Caption Data

We select a subset of captions and audio clips from each split in Clotho [3], with each subset containing 200 captions and 17 audio clips for each caption. Clotho crowdsources captions for each audio clip, and captions scored high by other workers results to the final captions [3]. The audio clips selected for each caption consists of one true positive clip (TP, being completely relevant to the caption), one true negative clip (TN, being completely irrelevant to the caption), and 15 relevance-unknown candidates (C15). Specifically, we select audio clips corresponding to the captions in Clotho as TPs and obtain TNs using low audio-caption similarity scores estimated by the baseline system in DCASE 2023 Challenge Task 6B[1] followed by human verification. For C15s, we choose the top five clips with high audio-caption similarity scores, together with another 10 randomly selected clips to include audio clips having diverse relevances. Table 1 summarizes the captions and audio clips for crowdsourcing relevance assessments.

2.3. Crowdsourcing Task Setting

For each caption, five HITs are created, each of which is assigned to several crowdworkers. Each HIT contains five audio clips, which are provided for crowdworkers to assess their individual relevance to the caption. The C15s of a caption are split into five batches of three C15s, with one batch per HIT. The TP and TN clips are used for quality check after crowdsourcing.

In each assignment, crowdworkers are asked to assign numeric scores (between 0 and 100) to indicate their judgements of how much the sound content in each audio clip matches the given caption in that assignment. Inspired by [13], we grade audio-text relevances on a scale of 0 to 100, where 0 indicates completely irrelevant (i.e., no content match at all) and 100 indicates completely relevant (i.e., perfect content match). An initial value of 0 is set as the default relevance score for each audio clip. Crowdworkers are required to listen to each audio clip entirely.

2.4. Quality Check

To collect high-quality answers, a quality check is conducted before and after crowdsourcing. Worker requirements are set up for selecting workers with high-quality work on MTurk (e.g., workers with a HIT approve rate greater than 98%). Besides, crowdworkers should pass a predefined qualification test by correctly answering questions about identifying the audio clip described by a given caption from three candidates before they can accept our HITs (i.e., receiving assignments).

With the fact that TPs are clearly more relevant than TNs within the same assignment, the former should receive higher scores than the latter. Let s_{tp} be the graded relevance of a TP, and s_{tn} be the one of the TN within the same assignment. Consistency verification on s_{tp} and s_{tn} is applied to check and select worker answers at the worker level.

Let $D = \{S_i \mid 1 \leq i \leq N\}$ denote the set of N answers submitted by a worker, where $S_i = \left\{s^i_{tp}, s^i_{tn}, s^i_{c1}, s^i_{c2}, s^i_{c3}\right\}$ represents the i-th answer from the worker, and s_{c1}, s_{c2}, s_{c3} are the scores of the three C15 clips within the same assignment. As mentioned above, a C15 can be either completely relevant or irrelevant, or even partially relevant to a given caption. For every $S_i \in D$, we measure two random variables for the worker: X, which represents the difference of s_{tp} and s_{tn}, and Y, which denotes the difference of every pair of s_{c1}, s_{c2}, s_{c3}. Intuitively, s_{tp} should be higher than s_{tn} by more than what is expected by chance when sampling from Y. For consistency verification on s_{tp} and s_{tn}, we therefore require that X and Y should satisfy:

$$E(X) = E(s_{tp} - s_{tn}) \geq E(Y) + \sigma(Y), \tag{1}$$

where E represents the expected value and σ represents the standard deviation across D (i.e., all answers from the worker). If the inequality is not satisfied, D will be discarded completely.

3. AUDIO-TEXT RELEVANCE SCORES

This section analyzes the crowdsourced relevances.

---

[1] https://dcase.community/challenge2023/task-language-based-audio-retrieval.

Split	#HITs	#Workers	#Answers
development	1000	109	6651
validation	1000	113	5064
evaluation	1000	118	6489

Table 2: Statistics of crowdsourced data.

3.1. Crowdsourced Raw Scores

Table 2 summarizes information about the crowdsourced data after filtering the data based on the quality check. For each HIT, answers were collected from at least five distinct crowdworkers. In total, 18204 answers were crowdsourced from 340 MTurk workers.

Fig. 2 presents the distribution of raw relevance scores of TP, TN, and C15 clips. For TPs, approximately 60% of relevance scores have a value of 100. For TNs, about 90% of relevance scores are zeros, and over 98% of these scores are less than 20. It indicates that most crowdworkers can appropriately assess the relevances of TPs and TNs to a given caption. For C15s, over 10% of relevance scores have a value of 100, which indicates that some C15s are highly relevant to a given caption. We notice that around 20% of relevance scores of TPs are zeros, which shows the necessity of further processing on the crowdsourced raw scores.

3.2. Aggregated Scores

The raw scores from different workers regarding the relevance of an audio clip to a text query are aggregated by discarding a maximum and a minimum score and then averaging the remaining to produce a statistic that is robust to outliers. Fig. 3 presents the distribution of aggregated relevance scores of TP, TN, and C15 clips. After aggregating, the distribution of relevance scores becomes more balanced, with fewer instances of extreme or polarized judgements (e.g., scores of 0 and 100). Particularly, TPs exhibit a broader spectrum of relevances (e.g., having scores spanning from 30 to 100) compared to other clips. Over 99% of TPs have a score above 10, while about 99% of TNs have a score below 10. Around 30% of C15s have a score above 10, and roughly 10% have a score above the mean score of TPs (i.e., $s > 72$).

4. EXPERIMENTS

This section reports experimental results of using the crowdsourced relevances for text-to-audio retrieval. Due to the lack of established methods for using non-binary relevances for training and evaluation, we binarize the crowdsourced relevances.

4.1. Audio-Caption Pairs

Similar to previous studies [5], we tackle text-to-audio retrieval with contrastive learning. To obtain positive and negative examples for contrastive learning, we binarize the crowdsourced relevances (see 3.2) using the mean score of TPs as an arbitrary threshold. Specifically, we obtain positive audio-caption pairs by combining: 1) a caption with its high-graded C15 clips, which have a score above the threshold; 2) the TP clip of a caption with the captions corresponding to its high-graded C15 clips in Clotho; 3) the siblings (i.e., captions describing the same TP clip in Clotho) of a caption with its high-graded C15 clips. All other audio-caption combinations are treated as negative pairs. The resulting positive and negative pairs are referred to as content-matching pairs with “Binarized Crowdsourced Relevances” (BiCrRel).

Split	BiCrRel	BiRel	BiCrRel+BiRel
development	3890	2370	6260
validation	2560	1580	4140
evaluation	2440	1390	3830

Table 3: Number of positive audio-caption pairs in BiCrRel, BiRel, and their combination (“BiCrRel+BiRel”).

As a baseline, we created a subset of Clotho by selecting those audio-caption pairs of which the audio or the caption were part of BiCrRel, i.e., using the captioning-based clip-specific audio-caption pairs from Clotho [3]. The selected pairs are referred to as captioning-based pairs with “Binary Relevances” (BiRel), which include the same audio clips and captions as in BiCrRel. Table 3 summarizes information about BiCrRel, BiRel, and their combination (“BiCrRel+BiRel”). The development / validation / evaluation splits are used for training / validation / evaluation, respectively.

4.2. Retrieval System

We experiment with the retrieval system proposed as the baseline in DCASE 2023 Challenge Task 6B[2], where a pretrained CNN14 [7] is employed as the audio encoder and the Sentence-BERT (i.e., “all-mpnet-base-v2”) [14] is used as the text encoder. This system is trained by optimizing the InfoNCE loss [11] such that embeddings of the paired audio and text are pulled together while those of the unpaired are pushed far away.

Audio Encoder. The CNN14 [7], which is pretrained on AudioSet [15], is employed as the audio encoder, with its last linear layer discarded. An extra linear layer is added on the top to generate 300-dimensional audio embeddings. The audio encoder is fine-tuned during training.

Text Encoder. The Sentence-BERT [14], which is derived from BERT [8] for the purpose of generating robust sentence embeddings, is used as the text encoder. An extra linear layer is also added on the top to generate 300-dimensional text embeddings. The Sentence-BERT is frozen during training.

InfoNCE Loss. The InfoNCE loss [11] is a symmetric cross-entropy loss, taking the form of

$$\mathcal{L} = -\frac{1}{M}\sum_{i=1}^{M}[\log\frac{\exp(z_{ii}/\tau)}{\sum_{j=1}^{M}\exp(z_{ij}/\tau)} + \log\frac{\exp(z_{ii}/\tau)}{\sum_{j=1}^{M}\exp(z_{ji}/\tau)}], \tag{2}$$

---

[2] https://dcase.community/challenge2023/task-language-based-audio-retrieval.

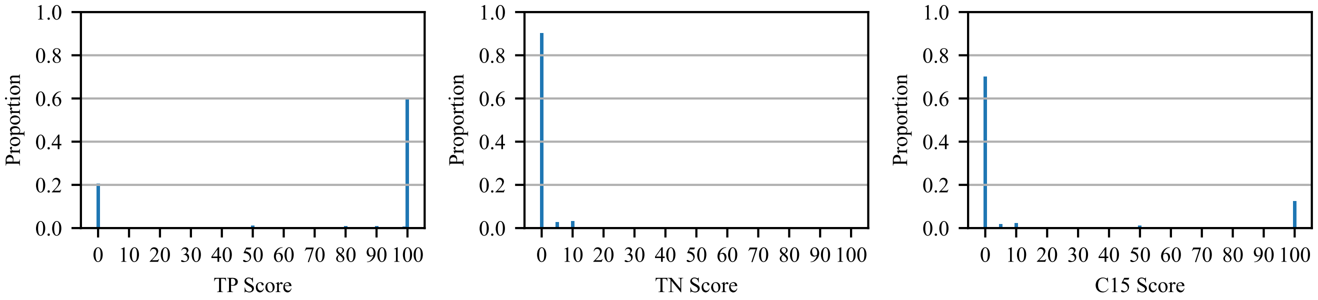


Figure 2: Distribution of raw relevance scores of TP, TN, and C15 clips.

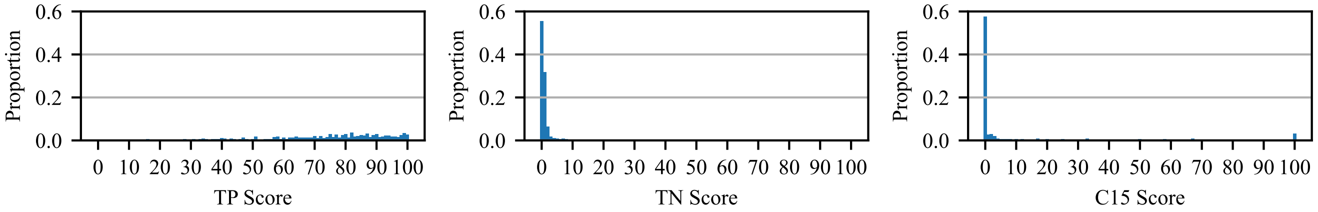


Figure 3: Distribution of aggregated relevance scores of TP, TN, and C15 clips.

where τ represents the temperature hyper-parameter, M denotes the number of audio-text pairs, and z_{ij} represents the cosine similarity of the i-th audio embedding and the j-th text embedding. It has been widely used to train cross-modal retrieval systems [5].

Training Setup. The retrieval system is trained with mini-batches consisting of 32 audio-text pairs from a development split. An Adam optimizer with an initial learning rate of 0.001 is adopted to optimize training. Learning rate is reduced by a factor of ten once the validation loss does not improve for five epochs. Training is terminated by early stopping with a patience of ten epochs.

4.3. Evaluation Metrics

Retrieval performance is measured with recall at 10 (R@10) on different evaluation splits. The R@10 is defined as the proportion of relevant items among the top 10 results to all the relevant items contained in the data and is averaged over queries [5]. The more relevant items are within top 10 results, the higher R@10 it is.

4.4. Results

Table 4 shows that training the system on BiRel leads to high R@10 (e.g., 0.566 on BiRel), whereas training the system on BiCrRel yields low R@10 (e.g., 0.478 on BiRel) and on BiCrRel+BiRel results in intermediate R@10 (e.g., 0.509 on BiRel). We conclude that the crowdsourced relevances do not improve the performance from captioning-based audio-caption pairs when they are reduced to binary relevances. A possible explanation is that captions in Clotho are crowdsourced specifically to describe an exact audio clip, while the crowdsourced relevances are graded based on their matching content with a given caption (i.e., different underlying purposes and criteria for generating captions and assessing relevances). Besides, with the fact that each caption in BiRel is annotated with one relevant audio clip while a caption in BiCrRel can have several relevant audio clips, it makes text-to-audio retrieval on BiCrRel more difficult and leads to a decrease in R@10.

Training Data	**Evaluation Data**		
	BiCrRel	**BiRel**	**BiCrRel+BiRel**
BiCrRel	0.357	0.478	0.407
BiRel	0.412	0.566	0.479
BiCrRel+BiRel	0.363	0.509	0.426

Table 4: Evaluation R@10 of text-to-audio retrieval with the retrieval system trained on different pairs.

5. CONCLUSIONS

We explore grading audio-text relevance for text-based audio retrieval via crowdsourcing assessments. We crowdsource audio-text relevances graded on a scale of 0 to 100, where 0 indicates completely irrelevant and 100 indicates completely relevant. We integrate crowdsourced relevances into training and evaluating text-to-audio retrieval systems, and evaluate the effect of using them alongside binary relevances defined by captioning-based audio-caption pairs. Experimental results show that the crowdsourced relevances do not positively contribute to the performance when they are reduced to binary relevances, and using only binary relevances defined by captioning-based audio-caption pairs is sufficient for contrastive learning.

6. ACKNOWLEDGMENT

The research leading to these results has received funding from Emil Aaltonen foundation funded project “Using language to interpret unstructured data” and Academy of Finland grant no. 314602.

7. REFERENCES

[1] M. Slaney, "Semantic-audio retrieval," in *Proc. Int. Conf. Acoustic., Speech and Signal Process. (ICASSP)*, 2002, pp. IV–4108–IV–4111.

[2] S. Ikawa and K. Kashino, "Acoustic event search with an onomatopoeic query: Measuring distance between onomatopoeic words and sounds," in *Proc. Detect. Classif. Acoust. Scenes Events Work. (DCASE)*, 2018, pp. 59–63.

[3] K. Drossos, S. Lipping, and T. Virtanen, "Clotho: an audio captioning dataset," in *Proc. Int. Conf. Acoustic., Speech and Signal Process. (ICASSP)*, 2020, pp. 736–740.

[4] C. Kim, B. Kim, H. Lee, and G. Kim, "Audiocaps: Generating captions for audios in the wild," in *Proc. Conf. North Am. Chapter Assoc. Comput. Linguist. Hum. Lang. Technol. (HLT-NAACL)*, 2019, pp. 119–132.

[5] H. Xie, S. Lipping, and T. Virtanen, "Language-based audio retrieval task in dcase 2022 challenge," in *Proc. Detect. Classif. Acoust. Scenes Events Work. (DCASE)*, 2022, pp. 216–220.

[6] A. Oncescu, A. Koepke, J. Henriques, Z. Akata, and S. Albanie, "Audio retrieval with natural language queries," in *Proc. Annu. Conf. Int. Speech Commun. Assoc. (INTERSPEECH)*, 2021, pp. 2411–2415.

[7] Q. Kong, Y. Cao, T. Iqbal, Y. Wang, W. Wang, and M. Plumbley, "Panns: Large-scale pretrained audio neural networks for audio pattern recognition," *IEEE/ACM Trans. Audio Speech Lang. Process.*, pp. 2880–2894, 2020.

[8] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North Am. Chapter Assoc. Comput. Linguist. Hum. Lang. Technol. (NAACL-HLT)*, 2019, pp. 4171–4186.

[9] X. Mei, X. Liu, J. Sun, M. Plumbley, and W. Wang, "On metric learning for audio-text cross-modal retrieval," in *Proc. Annu. Conf. Int. Speech Commun. Assoc. (INTERSPEECH)*, 2022, pp. 4142–4146.

[10] H. Xie, O. Räsänen, and T. Virtanen, "On negative sampling for contrastive audio-text retrieval," in *Proc. Int. Conf. Acoustic., Speech and Signal Process. (ICASSP)*, 2023, pp. 1–5.

[11] A. Van den Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," 2018. [Online]. Available: https://arxiv.org/abs/1807.03748

[12] TAU Audio-Text Graded Relevances 2023 Dataset, https://github.com/xieh97/retrieval-relevance-crowdsourcing.

[13] K. Roitero, E. Maddalena, S. Mizzaro, and F. Scholer, "On the effect of relevance scales in crowdsourcing relevance assessments for information retrieval evaluation," *Inf. Process. Manag.*, p. 102688, 2021.

[14] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using Siamese BERT-networks," in *Proc. Empherical Methods Nat. Lang. Process. (EMNLP)*, 2019, pp. 3982–3992.

[15] J. F. Gemmeke, D. P. W. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio set: An ontology and human-labeled dataset for audio events," in *Proc. IEEE Int. Conf. Acoustic., Speech and Signal Process. (ICASSP)*, 2017, pp. 776–780.

TEXT-DRIVEN FOLEY SOUND GENERATION WITH LATENT DIFFUSION MODEL

Yi Yuan, Haohe Liu, Xubo Liu, Xiyuan Kang, Peipei Wu, Mark D.Plumbley, Wenwu Wang

Centre for Vision Speech and Signal Processing, University of Surrey, United Kingdom

ABSTRACT

Foley sound generation aims to synthesise the background sound for multimedia content. Previous models usually employ a large development set with labels as input (e.g., single numbers or one-hot vector). In this work, we propose a diffusion model based system for Foley sound generation with text conditions. To alleviate the data scarcity issue, our model is initially pre-trained with large-scale datasets and fine-tuned to this task via transfer learning using the contrastive language-audio pretraining (CLAP) technique. We have observed that the feature embedding extracted by the text encoder can significantly affect the performance of the generation model. Hence, we introduce a trainable layer after the encoder to improve the text embedding produced by the encoder. In addition, we further refine the generated waveform by generating multiple candidate audio clips simultaneously and selecting the best one, which is determined in terms of the similarity score between the embedding of the candidate clips and the embedding of the target text label. Using the proposed method, our system ranks 1st among the systems submitted to DCASE Challenge 2023 Task 7. The results of the ablation studies illustrate that the proposed techniques significantly improve sound generation performance. The codes for implementing the proposed system are available at `https://github.com/yyua8222/Dcase2023_task7`.

Index Terms— Sound generation, Diffusion model, Transfer learning, Language model

1. INTRODUCTION

The development of deep learning models has recently achieved remarkable breakthroughs in the field of sound generation [1, 2, 3, 4]. Among various application domains of sound, Foley sounds, the mimic of background sound, play a crucial role in enhancing the perceived acoustic properties of movies, music, videos and other multimedia content [5]. The development of an automatic Foley sound synthesis system holds immense potential in simplifying traditional sound generation, which often involves intensive labour work on sound recording and mixing.

Currently, many sound generation models [1, 6, 7] adopt an encoder-decoder architecture, showing remarkable performance. Liu et al. [7] utilize a convolutional neural network (CNN) encoder, a variational autoencoder (VAE) decoder and a generative adversarial network (GAN) vocoder. The encoder embeds the input feature (e.g., label) into latent variables and the decoder transforms this intermediate information into mel-spectrogram which is then converted to a waveform by the vocoder. Diffsound [6] utilizes text as input and obtains the semantic features by using a contrastive language image pre-training (CLIP) model [8]. AudioGen [4] further improves the performance by using a pre-trained Transfer Text-to-Text Transformer (T5) [9] to obtain text embedding, which is then used to generate the waveform directly without using a vocoder.

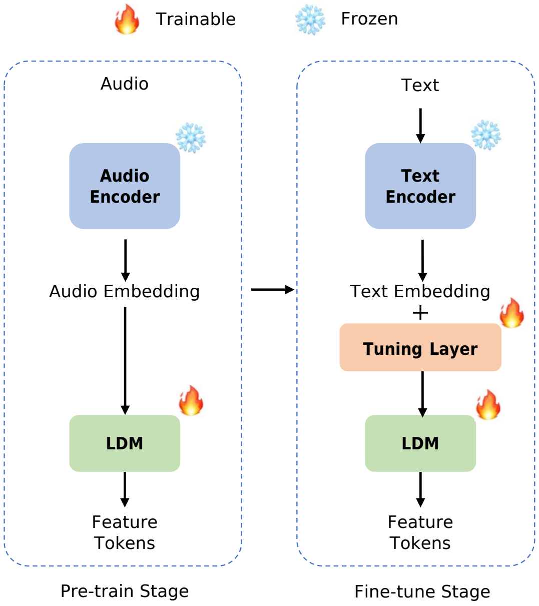


Figure 1: The training process of the LDM model. Audio Embedding are applied for pre-train the LDM on large dataset, while text embedding and an extra tuning layer are applied for fine-tune the LDM on target dataset

This paper proposes a latent diffusion model (LDM) based method for Foley sound generation. Our model follows the structure of AudioLDM [1], an audio generation model that comprises a diffusion model based encoder, a VAE based module for learning audio prior, and a HiFi-GAN vocoder for waveform generation. Due to the lack of training data for the sound generation task, we follow the idea of pre-training [10, 11], by initially training all three models on large-scale datasets such as AudioSet [12], AudioCaps [13] and Freesound[1], and then transferring them onto the target development set. For inputs, the category labels are initially wrapped into relevant texts (e.g., turning the label “Keyboard” into text “Someone using keyboard”) before they are passed into the contrastive language-audio pre-training (CLAP) [14] for generating the text embeddings. To learn the most suitable semantic features of each sound, an embedding tuning layer is then added to text embedding for finding the optimal embedding during the fine-tuning stage. As shown in Figure 1, we first use audio embeddings to pre-train the LDM model. Then a tuning layer is introduced into the system, which is updated via transfer learning along with the LDM module. For outputs, the cosine-similarity score obtained in terms of the outputs and target labels is used to select the best-related sounds from a pool of candi-

---

[1] https://freesound.org

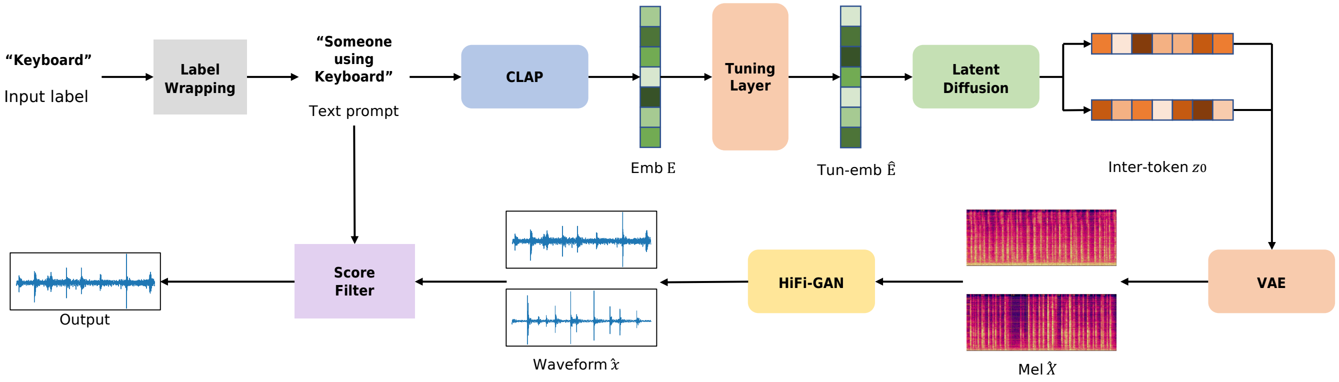


Figure 2: The overview of the system for both fine-tune stage and inference stage

date sound clips, which can improve the overall quality of the final result. Through experiments and ablation studies, we observe that the proposed techniques in our system can significantly improve the model performance in both the relevance of generated sound and the stability of the overall quality. Our system achieves a Fréchet audio distance (FAD) score of 4.52 on the DCASE task 7 validation set, significantly better than the baseline model with a FAD of 9.7.

The remaining sections of this paper are organised as follows. Section 2 describes the overview of the proposed system. The methodology of the network is explained in Section 3. Section 4 introduces the experimental setup. Results are shown in Section 5. Section 6 summarizes this work and draws the conclusion.

2. SYSTEM OVERVIEW

Our proposed system is based on the widely used structure on sound generation, which consists of an encoder, a generator, a decoder and a vocoder. The system adopts the same structure as AudioLDM [1], which used the CLAP [14] as the encoder and a latent diffusion model as the generator.

As a cascade model, the decoder and vocoder are trained separately and then built into the overall system with the trained parameters frozen when training the LDM model based generator. Instead of directly using labels as the input, we employ a wrapping strategy to generate text descriptions for each label as the initial mechanism to enhance the semantic information of the input. For example, we turn the label "Keyboard" into texts "Someone using keyboard". Then, we introduce an embedding tuning layer after the encoder in order to produce a more suitable embedding for each sound.

During the generating stage, with the text input, the system extracts the text embedding using the CLAP model, and the LDM model then generates the intermediate representation of the sound feature, using the text embedding as a condition. Subsequently, the mel-spectrogram can be decoded from the tokens by the VAE decoder, which is then transformed into waveform by the GAN vocoder. This system is then further improved with several techniques:

- Transfer learning is introduced to boost the performance by pre-training the model on larger datasets.
- A tuning layer is applied during the fine-tuning stage to find the optimal embedding.
- Similarity score between the embedding of the generated output and the target embedding is applied to select the best match results among a group of waveform clips generated by the system.

Detailed explanations of these methods are provided in following section. The overall structure of the system is shown in Fig. 2.

3. PROPOSED METHOD

3.1. System structure

3.1.1. CLAP based encoder

We use the CLAP model to obtain the embedding of the input. CLAP consists of a text encoder f_{text} that turns a text description y into text embedding $\boldsymbol{E}^y$ and an audio encoder f_{audio} that computes an audio embedding $\boldsymbol{E}^x$ from audio samples x. The two encoders are trained with cross-entropy loss, resulting in an aligned latent space with the same dimension D_e for both audio and text embedding. Since most large audio datasets (e.g., AudioSet) only provide audio-label pairs, we leverage the cross-modal information provided by two encoders. Specifically, the system is pre-trained on larger datasets with audio embedding and fine-tuned with text embedding on the task development set. During the fine-tuning process, the text embedding $\boldsymbol{E}^y$ is passed through a trainable linear layer to find the optimal embedding feature for each class of sound. Details of this mechanism are presented in Section 3.2

3.1.2. LDM based generator

Our system uses an LDM[15] to generate the intermediate latent tokens, with the feature embedding ($\boldsymbol{E}^y$ or $\boldsymbol{E}^x$) as the condition. These tokens are then used by the VAE decoder to generate the mel-spectrogram. During training, the LDM involves two processes: 1) A forward process where the latent vector $\boldsymbol{z}_0$ is gradually turned into a standard Gaussian distribution $\boldsymbol{z}_N$ in N steps, with noise $\boldsymbol{\epsilon}$ added in each step. 2) A reverse process for the model to predict the transition probabilities $\boldsymbol{\epsilon}_\theta$ of each step n, for reconstructing the data $\boldsymbol{z}_0$ by removing the noise $\boldsymbol{z}_N$. The model is trained with a re-weighted objective [16] as:

$$L_n(\theta) = E_{\boldsymbol{z}_0,\boldsymbol{\epsilon},n}||\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\boldsymbol{z}_n, n, \boldsymbol{E})||_2^2 \quad (1)$$

where $\boldsymbol{\epsilon}_\theta$ is the Gaussian distribution predicted by LDM with current state $\boldsymbol{z}_n$, current step n, and current condition $\boldsymbol{E}$. During sampling, the model first generates random Gaussian noise as z_N, and then applies the denoising process by predicting the reverse transition probability and taking the $\boldsymbol{E}$ from CLAP as the condition.

3.1.3. VAE decoder & HiFi-GAN vocoder

We utilize a combination of a VAE decoder and a HiFi-GAN vocoder to transform latent feature tokens into waveforms. Our approach involves training a VAE [17] to decode the latent feature tokens into mel-spectrograms, and a HiFi-GAN [18] to generate the corresponding waveforms. To achieve this, we initially convert the waveforms into mel-spectrograms using the Short-Time Fourier Transformation (STFT). The VAE is trained to compress the mel-spectrograms, $\boldsymbol{X}$, into a latent space vector z_0, then reconstruct the mel-spectrograms $\hat{\boldsymbol{X}}$ from the compressed representation. In parallel, we employ a HiFi-GAN to convert the mel-spectrograms $\hat{\boldsymbol{X}}$ into the corresponding waveform representations, denoted as $\hat{x}$.

3.2. Practical issues

Transfer learning To deal with the issue of data scarcity, our system takes advantage of a pre-trained model [10] by initially training all three models on extensive audio datasets, followed by fine-tuning them on our development dataset. Specifically, the LDM model undergoes its initial training phase using large-scale datasets with audio embeddings as inputs, while the model is then trained on the development dataset utilizing text embeddings.

Embedding tuning To first initialize the text embedding with more semantic features, we apply some hand-picked text by extending the label with some adjunct word (e.g., dogbark into a dog bark). We then apply a tuning strategy to determine the optimal embedding of each sound class. To implement this, we introduce a linear layer *L(x)* with trainable parameters to fine-tune the text embedding before passing it to the LDM model. To guide this trainable layer with only minor updates on the embedding, the parameters are initialized with an identity matrix as weight, along with a Gaussian noise as bias l_b. Hence, the initial *L(x)* serves as an adding function of input *x* and bias l_b at the beginning of the training process. Then, the system learns to update the parameter of both weight and bias for optimal embedding during training. The embedding updated by this linear layer is also used as the target embedding for the score-selecting function discussed in the following section.

Score-based selection To improve the overall generation quality and robustness, a scoring mechanism is applied to determine the best matches among sampling results. Leveraging the fact that CLAP provides embeddings in the same latent space for audio and text, we utilize the cosine similarity between the output audio and the target text. By comparing the FAD score of different groups of output clips with different score-selecting thresholds, specific thresholds are established for each class, allowing the system to only selects the results surpassing these thresholds.

4. EXPERIMENTAL SETUP

4.1. Dataset

DCASE2023-T7 consists of a training set and an official evaluation set with seven different classes of fully labelled urban sounds. Each class has around 600 to 800 4-second sound clips in the training set and exactly 100 clips in the evaluation set. We randomly partitioned the training dataset into two subsets, with a ratio of $9:1$ for training and validation purposes, while the evaluation set was exclusively used during the evaluation phase.

AudioSet is a large-scale dataset for audio research, which consists a wide range of sounds. In detail, Audioset provides around 2.1 million 10-second audio with 527 classes of labels. Our system uses AudioSet during the pre-training stage.

Freesound is a similar audio dataset with labels but with a non-fixed length, ranging from one second to several minutes. To unify the output length, all the sounds in Freesound are padded into a 10-second-long clip to match the data in Audioset.

By combining AudioSet and Freesound, we collected around 2.2M sounds in 22.05Khz for pre-training the LDM, VAE and GAN models. By using the audio-embedding and mel-spectrogram as input conditions, we only utilize the audio features to pre-train the models, while label features are then used during fine-tuning stage with the official training dataset.

4.2. Evaluation metrics

We apply the FAD [19] score as main evaluation metric. In detail, FAD calculates the Fréchet distance F between a group of target sound audio clip t and a group of generated sound audio clip r, formed:

$$F = ||\mu_r - \mu_t||^2 + tr(\Sigma_r + \Sigma_t - 2\sqrt{\Sigma_r\Sigma_t}) \quad (2)$$

where μ and Σ are the mean and covariance of Gaussian of the embedding vector from each group extracted by VGGish [20].

4.3. Parameter setting

Both the decoder and vocoder are trained separately, then they are integrated into the overall system with parameters fixed when training the LDM model. Initially, all three models are pre-trained using AudioSet and Freesound from scratch and then fine-tuned with the development set.

For the mel-spectrogram of 22.05kHz sounds, we set the window length as 1024 samples, the hop size as 256 and the number of mel-filterbank as 80. The VAE is trained with a compression level of 4, which encodes the mel-spectrogram into a latent vector of 20 in the frequency dimension and 86 in the time dimension. The length of the audio embedding $\boldsymbol{E}^x$ and text embedding $\boldsymbol{E}^y$ from the CLAP encoder in Section 3.1 is 512. All the models are optimized with Adam optimizer under an initial learning rate of 3.0×10^{-5}, with 3 epochs on the pre-training dataset and up to 1000 epochs on the training set. We test the model (LDM-S) performance by generating 100 clips per class and calculate the FAD score as the evaluation metric for every $100,000$ step.

To investigate the influence of the input embedding features, we employ a diverse set of labels and texts as embeddings for training the model. In the case of embedding tuning, we begin by selecting a specific set of text for providing the initial text embedding. Then, we train the tuning layer along with the LDM model, guiding the embedding towards the optimal value.

To further investigate the potential of the model on sound generation, we trained a larger LDM with a bigger CLAP model (LDM-L) with the same training configurations. To balance the computing complexity and the output quality, we trained this model on 16kHz sounds and upsample it to 22.05kHz before output the results. For the 16kHz mel-spectrogram, the hop size is decreased to 160 with a mel-bin dimension of 64. The results of both models are shown in the following section.

System	Dog Bark	Footstep	Gun Shot	Keyboard	Moving Motor Vehicle	Rain	Sneeze Cough
Baseline [7]	13.41	8.11	7.95	5.23	16.11	13.34	3.77
LDM-S	4.41	7.44	7.46	3.13	16.97	12.62	3.02
LDM-S+Pre	4.17	6.86	7.25	3.15	15.68	12.95	2.85
LDM-S+Pre+Text	3.84	5.66	6.66	3.48	14.35	12.62	2.12
LDM-S+Pre+Text+Filter	3.53	5.04	5.65	2.80	15.29	9.76	**1.92**
LDM-S+Pre+Text+Filter+Tuned	**3.36**	**4.77**	**5.19**	**2.69**	14.83	10.00	1.98
LDM-L+Pre+Text+Filter+Tuned	6.04	5.05	6.44	3.07	**11.08**	**4.74**	2.93

Table 1: The best results of each system on the DCASE2023-T7 evaluation set. LDM-S: model trained from scratch. Pre: model with pre-training on large datasets. Text: using label-related text as input. Filter: applying the score-selecting function. Tuned: model with a fine-tuned embedding. The score selection for motor sound is used with the text embedding of " A moving motor ".

5. RESULTS AND ANALYSIS

The performance of our system on DCASE2023-T7 validation set is reported in Table 1. Most of our models outperform the baseline [7] by a large margin in terms of FAD. The results obtained from different sizes of LDM highlight distinct strengths: LDM-S is better at generating clear and distinct sounds like dog barks, footsteps, and gunshots, whereas the larger model (LDM-L) demonstrates superior performance in handling complex sounds such as motor sounds and rain sounds.

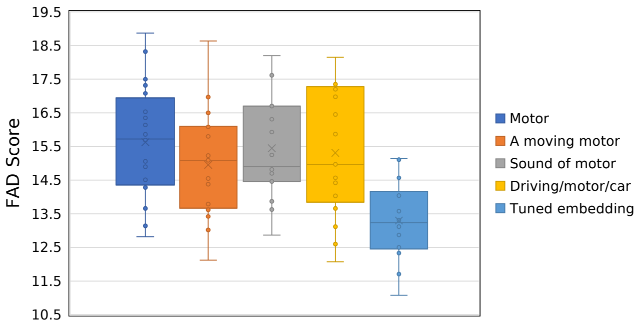


Figure 3: Results of motor with different training embedding

Ablation studies are also conducted to investigate the effects of each proposed technique. The experimental results in Table 1 demonstrate that transfer learning generally improves the system performance in most cases with respect to the evaluation metrics. Applying the embedding tuning strategy enables the system to optimize the embedding value for each class, which further improves the performance. To validate the effectiveness of this embedding tuning mechanism, we conduct several experiments with models trained with different frozen text embeddings. All the models are sampled under the same configuration for up to 20 times and the results of motor sound are presented in the box chart of Fig. 3. It can be observed that training without embedding tuning may yield results with varying quality, ranging from the best FAD of around 12.5 to the largest score of up to 18. On the contrary, generating a well-trained embedding value can contribute to more stable results. This might be because the updated embedding during training can provide more semantic information for both LDM denoising and the waveform tuning process.

From Table 1, the utilization of the similarity score function significantly enhances the overall performance, leading to improved output quality in most scenarios. However, despite the improvements observed in the majority of classes, we noticed that the generation quality of motor sounds did not exhibit a significant decrease in FAD (best achieved 11.08). By operating several subjective evaluations (human evaluation), we find out that this might be because most motor sounds consist of noise-like sounds and sound events with distinct differences (e.g., driving sounds and engine sounds), making it challenging for CLAP to identify and extract a single embedding that aligns perfectly with all the target clips. To address this issue and improve the correlation of the score function, we introduced a multi-target-selection approach to replace the single embedding score-selection. Specifically, we collected a set of audio embeddings that demonstrated top feature correlation with the training dataset and randomly selected an audio embedding for the score-selection during each iteration. As the result presents in Table 2, our system with multiple audio-embedding filters achieves a notable FAD score of 8.88 for motor sounds.

Embedding	Moving Motor Vehicle
Label	16.97
Motor	13.14
A moving motor	12.12
Sound of motor	12.87
Driving/motor/car	12.07
Tuned embedding	11.08
Audio embedding	**8.88**

Table 2: The best results on LDM-L with FAD on motor sounds between different score-selection. Embeddings indicate the text/label value for training and similarity calculation.

6. CONCLUSION

This paper proposes a framework for small-domain Foley sound generation. Our system leverages a diffusion-based model and applied several methods to enhance performance. On the input feature, our experiment shows that the input embedding can significantly affect the overall quality. To alleviate this distinct gap between label and sound alignment, we proposed a trainable embedding for tuning the embedding value. Our result indicates that an improved embedding can further improve the quality and stability of the model. For output, a score-selection strategy is utilized to select the best clip along with CLAP score similarity. The experimental result shows that our system can significantly improve over the baseline network by a large margin. In the future, we will explore more efficient and end-to-end methods for audio feature extraction and fine-tuning.

7. ACKNOWLEDGMENT

This research was partly supported by a research scholarship from the China Scholarship Council (CSC) No.202208060240, funded from British Broadcasting Corporation Research and Development (BBC R&D), Engineering and Physical Sciences Research Council (EPSRC) Grant EP/T019751/1 “AI for Sound”, and a PhD scholarship from the Centre for Vision, Speech and Signal Processing (CVSSP), University of Surrey. For the purpose of open access, the authors have applied a Creative Commons Attribution (CC BY) license to any Author Accepted Manuscript version arising.

8. REFERENCES

[1] H. Liu, Z. Chen, Y. Yuan, X. Mei, X. Liu, D. Mandic, W. Wang, and M. D. Plumbley, “AudioLDM: Text-to-Audio generation with latent diffusion models,” in *International Conference on Machine Learning*, 2023.

[2] Z. Borsos, R. Marinier, D. Vincent, E. Kharitonov, O. Pietquin, M. Sharifi, D. Roblek, O. Teboul, D. Grangier, M. Tagliasacchi, *et al.*, “Audiolm: a language modeling approach to audio generation,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2023.

[3] R. Huang, J. Huang, D. Yang, Y. Ren, L. Liu, M. Li, Z. Ye, J. Liu, X. Yin, and Z. Zhao, “Make-An-Audio: Text-To-Audio generation with prompt-enhanced diffusion models,” *arXiv preprint arXiv:2301.12661*, 2023.

[4] F. Kreuk, G. Synnaeve, A. Polyak, U. Singer, A. Défossez, J. Copet, D. Parikh, Y. Taigman, and Y. Adi, “AudioGen: textually guided audio generation,” in *International Conference on Learning Representations*, 2023.

[5] K. Choi, J. Im, L. Heller, B. McFee, K. Imoto, Y. Okamoto, M. Lagrange, and S. Takamichi, “Foley sound synthesis at the dcase 2023 challenge,” *In arXiv e-prints: 2304.12521*, 2023.

[6] D. Yang, J. Yu, H. Wang, W. Wang, C. Weng, Y. Zou, and D. Yu, “Diffsound: Discrete Diffusion Model for Text-to-sound Generation,” *arXiv preprint arXiv:2207.09983*, 2022.

[7] X. Liu, T. Iqbal, J. Zhao, Q. Huang, M. Plumbley, and W. Wang, “Conditional sound generation using neural discrete time-frequency representation learning,” *IEEE International Workshop on Machine Learning for Signal Processing*, pp. 1–6, 2021.

[8] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, *et al.*, “Learning transferable visual models from natural language supervision,” in *International Conference on Machine Learning*, 2021, pp. 8748–8763.

[9] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, “Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer,” *arXiv preprint arXiv:1910.10683*, 2019.

[10] Y. Yuan, H. Liu, J. Liang, X. Liu, M. D. Plumbley, and W. Wang, “Leveraging pre-trained audioldm for sound generation: A benchmark study,” *arXiv preprint arXiv:2303.03857*, 2023.

[11] D. Hendrycks, K. Lee, and M. Mazeika, “Using pre-training can improve model robustness and uncertainty,” in *International Conference on Machine Learning*, vol. 97, 2019, pp. 2712–2721.

[12] J. F. Gemmeke, D. P. W. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, “AudioSet: An ontology and human-labeled dataset for audio events,” in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2017, pp. 776–780.

[13] C. D. Kim, B. Kim, H. Lee, and G. Kim, “AudioCaps: Generating captions for audios in the wild,” in *NAACL-HLT*, 2019.

[14] Y. Wu, K. Chen, T. Zhang, Y. Hui, T. Berg-Kirkpatrick, and S. Dubnov, “Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation,” in *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*, 2023.

[15] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, “High-resolution image synthesis with latent diffusion models,” in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, 2022, pp. 10 674–10 685.

[16] J. Ho, A. Jain, and P. Abbeel, “Denoising diffusion probabilistic models,” in *Neural Information Processing Systems*, 2020.

[17] V. Iashin and E. Rahtu, “Taming visually guided sound generation,” in *British Machine Vision Conference (BMVC)*, 2021.

[18] J. Kong, J. Kim, and J. Bae, “HiFi-GAN: generative adversarial networks for efficient and high fidelity speech synthesis,” in *Proceedings of the 34th International Conference on Neural Information Processing Systems*, 2020, pp. 17 022–17 033.

[19] K. Kilgour, M. Zuluaga, D. Roblek, and M. Sharifi, “Fréchet Audio Distance: A Metric for Evaluating Music Enhancement Algorithms,” *arXiv preprint arXiv:1812.08466*, 2018.

[20] K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition,” *arXiv preprint arXiv:1409.1556*, 2014.