

VLADIMIR IASHIN

Multi-modal Video Content Understanding

VLADIMIR IASHIN

Multi-modal Video Content Understanding

ACADEMIC DISSERTATION

To be presented, with the permission of
the Faculty of Information Technology and Communication Sciences
of Tampere University,
for public discussion in the TB109
of the Tietotalo, Korkeakoulunkatu 1, Tampere,
on 12 May 2023, at 1:00 pm.

ACADEMIC DISSERTATION

Tampere University, Faculty of Information Technology and Communication
Sciences
Finland

*Responsible
supervisor
and Custos*

Professor
Esa Rahtu
Tampere University
Finland

Pre-examiners

Assistant Professor
Andrew Owens
University of Michigan
United States

Associate Professor
Romain Serizel
Université de Lorraine
France

Opponent

Assistant Professor
Yuki Asano
University of Amsterdam
Netherlands

The originality of this thesis has been checked using the Turnitin OriginalityCheck service.

Copyright ©2023 author

Cover design: Roihu Inc.

ISBN 978-952-03-2871-9 (print)

ISBN 978-952-03-2872-6 (pdf)

ISSN 2489-9860 (print)

ISSN 2490-0028 (pdf)

<http://urn.fi/URN:ISBN:978-952-03-2872-6>



Carbon dioxide emissions from printing Tampere University dissertations have been compensated.

PunaMusta Oy – Yliopistopaino
Joensuu 2023

ACKNOWLEDGEMENTS

I would like to express my gratitude to my scientific supervisor, Professor Esa Rahtu, for his support and guidance throughout my doctorate studies. He played a fundamental role in my pursuit of the degree and I feel truly fortunate to work with him. I am grateful for his time, insightful suggestions, and out-of-the-box thinking. He could always find a bright side in any circumstance. Thank you for giving me the freedom and resources to explore my ideas and helping me to follow my ambitions.

Over the course of my studies, I was honoured to collaborate with remarkable academics from the Visual Geometry Group (VGG) at the University of Oxford. Specifically, I feel exceptionally lucky to have had a chance to work with Professor Andrew Zisserman who is not only astonishingly wise but also modest, kind, and empathetic. There is so much to learn from you. Also, I would like to thank Professor Weidi Xie (now at Shanghai Jiao Tong University) who was always willing to help me with my technical questions and share his expertise. Thank you for saving me from mistakes I would have made otherwise and for being direct and honest.

My studies were generously funded by the Academy of Finland grants 327910 and 324346 which allowed me to have a decent living as a PhD student. In addition, I would like to acknowledge the computation resources for my experiments that were kindly provided by CSC – IT Center for Science, Finland. I remember occupying 80 x NVidia V100 GPUs at once for several days. As well as, I thank beautiful Finland where all this was possible, and its modest, respectful, and polite people.

During my studies, I had a chance to meet several very special people whom I can proudly call my friends. I have spent hundreds of fun hours with Soumya Tripathy talking about all kinds of things, *e. g.* IDEs, mechanical watches, specialty coffee, and PhD-student life. Also, I was also fortunate to meet Shanshan Wang who was proven to be a loyal, supportive, and deeply caring friend. I am glad to know Shushik Avagyan, a person with whom I happen to have a similar cultural background. I thank Lingyu Zhu for inspiring me to ask great questions and leave

nothing on the table. I wish we could spend more time together. I am happy to know Nicklas Fianda who is not only incredibly smart and curious but also a delightful human being. Without all of you, my days would be less bright.

Last, but certainly not least, I thank my family. In particular, I am infinitely grateful to my grandmother, Yevgeniya Gashnurova, and mother, Svetlana Iashina, for their unconditional love and support. I know it was not easy for you. I thank them for giving me the freedom for making my own life-changing decisions. This thesis would not be possible without you. Finally, I would like to highlight the importance of Anna Iashina in my life who became my spouse during my doctorate studies. Among other things, I thank her for teaching me how to be a better human to others. She liked me back when I did not have anything for the way I am (or perhaps because I knew the matrix form solution to linear regression by heart). I am so lucky to have you by my side.

Tampere, April 2023

Vladimir Iashin

ABSTRACT

Video is an important format of information. Humans use videos for a variety of purposes such as entertainment, education, communication, information sharing, and capturing memories. To this date, humankind accumulated a colossal amount of video material online which is freely available. Manual processing at this scale is simply impossible. To this end, many research efforts have been dedicated to the automatic processing of video content.

At the same time, human perception of the world is multi-modal. A human uses multiple senses to understand the environment and objects, and their interactions. When watching a video, we perceive the content via both audio and visual modalities, and removing one of these modalities results in less immersive experience. Similarly, if information in both modalities does not correspond, it may create a sense of dissonance. Therefore, joint modelling of multiple modalities (such as audio, visual, and text) within one model is an active research area.

In the last decade, the fields of automatic video understanding and multi-modal modelling have seen exceptional progress due to the ubiquitous success of deep learning models and, more recently, transformer-based architectures in particular. Our work draws on these advances and pushes the state-of-the-art of multi-modal video understanding forward.

Applications of automatic multi-modal video processing are broad and exciting! For instance, the content-based textual description of a video (video captioning) may allow a visually- or auditory-impaired person to understand the content and, thus, engage in brighter social interactions. However, prior work in video content description relies on the visual input alone, missing vital information only available in the audio stream.

To this end, we proposed two novel multi-modal transformer models that encode audio and visual interactions simultaneously. More specifically, first, we introduced a late-fusion multi-modal transformer that is highly modular and allows the processing

of an arbitrary set of modalities. Second, an efficient bi-modal transformer was presented to encode audio-visual cues starting from the lower network layers allowing more rich audio-visual features and stronger performance as a result.

Another application is the automatic visually-guided sound generation that might help professional sound (foley) designers who spend hours searching a database for relevant audio for a movie scene. Previous approaches for automatic conditional audio generation support only one class (*e.g.* “dog barking”), while real-life applications may require generation for hundreds of data classes and one would need to train one model for every data class which can be infeasible.

To bridge this gap, we introduced a novel two-stage model that, first, efficiently encodes audio as a set of codebook vectors (*i.e.* trains to make “building blocks”) and, then, learns to sample these audio vectors given visual inputs to make a relevant audio track for this visual input. Moreover, we studied the automatic evaluation of the conditional audio generation model and proposed metrics that measure both quality and relevance of the generated samples.

Finally, as video editing is becoming more common among non-professionals due to the increased popularity of such services as YouTube, automatic assistance during video editing grows in demand, *e.g.* off-sync detection between audio and visual tracks. Prior work in audio-visual synchronization was devoted to solving the task on lip-syncing datasets with “dense” signals, such as interviews and presentations. In such videos, synchronization cues occur “densely” across time, and it is enough to process just a few tens of a second to synchronize the tracks. In contrast, open-domain videos mostly have only “sparse” cues that occur just once in a seconds-long video clip (*e.g.* “chopping wood”).

To address this, we: a) proposed a novel dataset with “sparse” sounds; b) designed a model which can efficiently encode seconds-long audio-visual tracks in a small set of “learnable selectors” that is, then, used for synchronization. In addition, we explored the temporal artefacts that common audio and video compression algorithms leave in data streams. To prevent a model from learning to rely on these artefacts, we introduced a list of recommendations on how to mitigate them.

This thesis provides the details of the proposed methodologies as well as a comprehensive overview of advances in relevant fields of multi-modal video understanding. In addition, we provide a discussion of potential research directions that can bring significant contributions to the field.

CONTENTS

1	Introduction	15
1.1	Scope	16
1.2	Summary of articles	17
1.3	Outline	18
2	Background	19
2.1	Earlier work in multi-modal video understanding	21
2.1.1	Multi-modal machine learning	21
2.1.2	Video understanding	22
2.2	Multi-modal video content understanding with deep learning	25
2.2.1	Video captioning	25
2.2.2	Video paragraph captioning	28
2.2.3	Visual question answering	28
2.2.4	Text to video retrieval	30
2.2.5	Video moment retrieval	31
2.2.6	Text-guided video generation	34
2.2.7	Multi-modal action recognition	36
2.2.8	Multi-modal video foundation models and applications	37
2.3	Transformer architecture	39
3	Multi-modal Dense Video Captioning	47
3.1	Related work	48
3.2	Multi-modal transformer for dense video captioning (MDVC)	49
3.3	Better use of audio-visual cues with bi-modal transformer (BMT)	52
3.4	Experiments and results	57
3.5	Discussion	60

4	Visually-guided Sound Generation for Open-domain Videos.	63
4.1	Related work	64
4.2	Codebook-based conditional sampling (Spectrogram VQGAN). . .	65
4.3	Automatic metrics for spectrogram-based audio generation	71
4.4	Experiments and results	72
4.5	Discussion	77
5	Audio-visual Synchronization with Sparse Signals	79
5.1	Related work	80
5.2	Audio-visual synchronization with sparse signals (SparseSync). . .	81
5.3	Preventing temporal artefact leakage	84
5.4	Experiments and results	86
5.5	Discussion	89
6	Conclusion	91
	References	93
	Publication I	139
	Publication II	151
	Publication III	169
	Publication IV	187

List of Figures

2.1	Transformer architecture	41
3.1	Multi-modal Dense Video Captioning model (MDVC)	50
3.2	Bi-modal Transformer (BMT)	53
3.3	Bi-modal Multi-headed Event Proposal Generator.	56
4.1	Training pipeline of the visually-guided autoregressive codebook-based conditional sampler	66
4.2	The pipeline of generating new audio that is relevant to visual cues . . .	70

4.3	Qualitative results of spectrogram reconstruction with Spectrogram VQGAN (Stage I): examples from VGGSound	73
4.4	Qualitative results of spectrogram reconstruction with Spectrogram VQGAN (Stage I): examples from VAS	73
4.5	Qualitative comparison of our approach to the state-of-the-art baseline model performing visually-guided audio generation	74
5.1	Audio-visual synchronization model: SparseSync	81

List of Tables

3.1	Dense video captioning results of the proposed MDVC and BMT compared to prior work.	59
3.2	Event proposal generation results of the bi-modal multi-headed event proposal generator	60
3.3	The effect of other modalities on performance on the dense video captioning task	61
4.1	Reconstruction performance of Spectrogram VQGAN (Stage I) in a quantitative study.	73
4.2	Comparison of visually-guided sound generation models	75
4.3	The effect of adding PatchGAN and perceptual losses during training of Spectrogram VQGAN	75
5.1	Training to detect temporal artefacts	84
5.2	Results of comparison between the state-of-the-art baseline and SparseSync (Ours) on the audio-visual synchronization task.	88
5.3	Performance of SparseSync per data class of VGGSound-Sparse.	88
5.4	Ablation study SparseSync	88

ABBREVIATIONS

AAC	Advanced Audio Codec
ASR	Automatic Speech Recognition
BERT	Bi-directional Encoder Representations from Transformers [1]
BMT	Bi-modal Transformer
BoV	Bag-of-Visual-Words
CLIP	Contrastive Language-Image Pretraining [2]
CNN	Convolutional Neural Network
CPU and GPU	Central and Graphics Processing Unit
DETR	Detection with Transformer [3]
FC	Fully Connected layer
FFN	Feed-Forward Position-wise Network (in Transformer) [4]
FID	Frèchet Inception Distance
GAN	Generative Adversarial Networks [5]
GCN	Graph Convolutional Network
GloVe	Global Vector [6]
GPT	Generative Pretrained Transformer (OpenAI) [7-9]
GRU	Gated Recurrent Unit [10]
H.264	MPEG-4 Part 10
HMM	Hidden Markov Models
I3D	Two-stream Inflated 3D CNN [11]
KL-divergence	Kullback-Leibler divergence

LDM	Latent Diffusion Model [12]
LNorm	Layer Normalization [13]
LPIPS and LPAPS	Learned Perceptual Image (Audio) Patch Similarity
LRS3	Lip Reading Sentences [14]
LSTM	Long Short-Term Memory
MDVC	Multi-modal Dense Video Captioning model
MFCC	Mel-Frequency Cepstral Coefficients
MHA	Multi-Headed Attention (in Transformer) [4]
MKL	Melception-based KL-divergence
MLP	Multilayer Perception
NAS	Neural Architecture Search
NLP	Natural Language Processing
POS	Part-of-Speech
ReLU	Rectified Linear Unit
RGB	Red, Green, and Blue channels
RL	Reinforcement Learning
RNN	Recurrent Neural Network
S3D	Spatiotemporal 3D CNN [15]
SIFT	Scale-Invariant Feature Transform
SVM	Support Vector Machine
VAE	Variational Auto-Encoder
VAS	An audio-visual dataset introduced in [16]
VGGish	VGG-Net-based CNN pre-trained on AudioSet [17]
VLAD	Locally Aggregated Descriptor [18]
VQA and VideoQA	Visual Question Answering and Video Question Answering
VQVAE	Vector-Quantized Variational Auto-Encoder [19]
YOLOv3	Object detector “You Only Look Once” ver. 3 [20]

ORIGINAL PUBLICATIONS

- Publication I V. Iashin and E. Rahtu, “Multi-modal dense video captioning,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, IEEE, 2020.
- Publication II V. Iashin and E. Rahtu, “A better use of audio-visual cues: Dense video captioning with bi-modal transformer,” in *31st British Machine Vision Conference 2020*, BMVA Press, 2020.
- Publication III V. Iashin and E. Rahtu, “Taming visually guided sound generation,” in *32nd British Machine Vision Conference 2021*, BMVA Press, 2021.
- Publication IV V. Iashin, W. Xie, E. Rahtu, and A. Zisserman, “Sparse in space and time: Audio-visual synchronisation with trainable selectors,” in *33rd British Machine Vision Conference 2022*, BMVA Press, 2022.

Author's contribution

In Publications I–IV, the author made instrumental contributions to a literature review, design of the models, implementation and execution of experiments, and paper presentation, as well as prepared the codebase for public release. The design of experiments and ideas were processed together with the co-authors. As well as, all authors participated in paper writing. Esa Rahtu provided high-level supervision in Publications I–III and in Publication IV together with A. Zisserman. In Publication IV, A. Zisserman and W. Xie posed the research problem for investigation.

1 INTRODUCTION

Nowadays, video data is omnipresent online. Television is being replaced with streaming services such as YouTube and Netflix. In addition, people generate countless hours of video content and share it publicly. Such a large amount of video content is impossible for a human to analyze. Therefore, this problem opens many opportunities for data-driven approaches dedicated to the automatic extraction of important information from a video or generating new content.

Human perception of the world is multi-modal. When a person decides how good is an apple, they tend to consider its taste, smell, visual appearance, and how firm it is [21]. In addition, entertainment has also become multi-modal, *e. g.* gaming consoles that are relying on audio and visual senses, today also have haptic motors in controllers. Similarly, we use multiple senses when another person tells us something, *i. e.* we not only listen but also follow the facial features and gestures that help us understand what is being told. Thus, in order to design a better model for video understanding, the input should be multi-modal.

Besides the biological plausibility of a multi-modal model, the motivation comes from a practical perspective. Given more hints or cues about the task, it is easier to answer it correctly. An additional modality provides with more information about the same event or an object. Conveniently, sometimes collecting an extra modality for a dataset does not bring additional costs, *e. g.* a collection of videos often naturally contains an audio track beside the RGB track. Therefore, it is beneficial to use more modalities when a practitioner targets the model's performance.

Apart from building better video representations with multiple modalities, certain applications require an ability to translate the content from one modality to another. Among others, a routine job of foley designers who currently spend hours creating or searching a database for relevant sounds to a movie scene. Along with, a visually impaired who may benefit from a textual description of a video online. Accordingly, multi-modal video understanding is a wide and important research area.

Building a model to solve multi-modal video understanding tasks is associated with addressing a broad range of challenges that are common among these and many other video understanding tasks. In particular, one needs to decide on how multiple modalities interact or are fused within a model given their distinct machine representation, *i. e.* audio is a two-dimensional signal, RGB stream is four-dimensional, and text has a single dimension. Most importantly, the video stream also includes the time dimension which should be efficiently encoded and processed. Moreover, video processing is often associated with a substantial computation burden and the presence of other modalities aggravates this issue even more.

This thesis introduces a variety of multi-modal architectures suited for video understanding tasks. Inspired by the wide success of the transformer architecture in natural language processing due to its ability to model long sequences, the transformer forms the basis of the proposed approaches. We tailor and generalize the architecture to address the challenges of multi-modal video understanding tasks. With these developments, we push the state-of-the-art in multi-modal video understanding.

1.1 Scope

This thesis aims to outline and expand the field of multi-modal video understanding. First of all, *modality* is defined as a type of data. In this thesis, the primary focus is on vision (or RGB sequence), audio, and text modalities. Thus, a *multi-modal model* is a system that learns using data from multiple modalities. Secondly, *video understanding* is a general term that groups together a variety of tasks that a computer is expected to solve with video data. The terms *video understanding* and *video content understanding* will be used interchangeably throughout the thesis. Note that a *cross-modal* task is considered to be a special case of multi-modal tasks. To sum up, the content of this thesis is related to the topics at the intersection of multi-modal data-intensive models and video understanding tasks.

In this work, we extend the advances in multi-modal video understanding by making contributions in the fields of dense video captioning, video-guided audio generation, and audio-visual synchronization. Meanwhile, the rest of the related topics in multi-modal video content understanding tasks are comprehensively outlined in the background chapter. More specifically, we aim to address the following research questions in this thesis.

1. What is the effect of other modalities on a performance of a dense video captioning model? (see Publications I and II)
2. Could autoregressive visually-guided sampling of latent codebook codes make open-domain audio generation possible? (see Publication III)
3. How to evaluate the performance of a conditional spectrogram-based generative model? (see Publication III)
4. Could trainable query vectors effectively encode the input feature sequences for audio-visual synchronization? (see Publication IV)
5. Do common RGB and audio compression algorithms leave temporal artefacts and how to detect them? (see Publication IV)

1.2 Summary of articles

In this thesis, we present our contributions to three major applications of multi-modal video understanding. Specifically, dense video captioning (Publications I and II), visually-guided audio generation (Publication III), and audio-visual synchronization (Publication IV).

Given a video, dense video captioning requires a model to, first, generate a set of temporal event boundaries and, then, make a textual description of each event. Although a human uses audio and visual senses to perceive video content, the majority of prior approaches in this area relied on visual modality alone. To this end, we explored the use of multiple modalities in Publication I. In particular, we contributed a novel transformer-based architecture which allows employing an arbitrary number of input modalities (MDVC), *i. e.* audio, visual, and speech in a form of a text, and achieved the state-of-the-art results on a popular benchmark.

In Publication II, we make further progress by addressing the shortcomings of MDVC and improve its performance. Specifically, despite being modular, the MDVC architecture is redundant. It lacks low-level multi-modal interactions and relies on visual-only event proposal generation. To address this, we generalize the transformer architecture for bi-modal inputs (audio and vision), and design a novel multi-modal event proposal generator. Although the bi-modal transformer (BMT) operates on only two modalities and has less parameters, it outperforms MDVC.

Video-guided audio generation was explored in Publication III. Most of the prior works focused on generating audio for narrow domains of videos, while open-domain

approaches required training one model per data class. To this end, we introduce a two-stage approach that factorizes the task into two sub-tasks. First, a set of spectrogram “building blocks” (codebook) is trained as a part of an autoencoder (Spectrogram VQGAN). Second, a transformer is trained to sample these “blocks” autoregressively while being prompted with video cues. In addition, novel fidelity and relevance metrics for the automatic evaluation of conditional spectrogram-based audio synthesis has been designed. The presented approach allows for building a model which supports the generation of a large variety of classes, has fewer parameters, and is more than 1000 times faster than the prior state-of-the-art model.

An audio-visual synchronization model is required to predict if the audio and visual tracks are out-of-sync temporally and how to fix it. In Publication IV, we worked with videos that have “sparse” synchronization signals. For instance, a video of a talking face has a “dense” synchronization signal while a video with a dog that barks only once during a 10-second clip is an example of a “sparse” synchronization signal. In the latter case, a model needs to process the whole video clip to catch the synchronization signal. This is challenging for today’s sequence-modelling approaches (a transformer) as the length of the audio-visual input tends to be quite long. To this end, we proposed a model, called SparseSync, that learns a small set of query vectors. These vectors effectively encode useful synchronization cues. This significantly reduces the input length of the synchronization transformer. The proposed approach outperforms the previous state-of-the-art by a large margin. In this work, we also introduced a novel dataset with videos that have sparse synchronization cues.

1.3 Outline

Chapter 2 contains background for multi-modal video understanding research. Chapter 3 summarizes the contributions to dense video captioning that were outlined in Publications I and II. The contributions of Publication III to the video-guided audio generation field are presented in Chapter 4. Next, the findings in audio-visual synchronization from Publication IV are in Chapter 5. Finally, Chapter 6 concludes.

2 BACKGROUND

The interest in multi-modal architectures and their applications for video understanding has emerged several decades ago. Earlier attempts were dedicated to audio-visual speech recognition, video retrieval, human interaction during group meetings, and audio-visual emotion recognition. The model training was done on top of pre-extracted features that were, in turn, domain-specific and hand-crafted. Clearly, there was a potential for end-to-end approaches that would make feature extraction more efficient, and deep learning methods quickly became natural candidates for this.

The success of deep neural architectures in data modelling is often attributed to the successful application of convolutional neural networks (CNNs) to the ImageNet image classification challenge [22] in 2012 [23] and onwards [24, 25]. Inspired by the success of CNNs in image classification, the research has quickly spread to other sub-areas of computer vision, such as action recognition [26], object detection [27], and optical flow estimation [28] and many other tasks. Shortly after, other fields have begun a transition to deep learning architectures, including language [29] and audio processing [17]. Research in multi-modal learning was also influenced by deep learning. The transition to deep learning led to the improvement of model performance across the field and allowed tackling more challenging applications in video understanding, such as video captioning, cross-modal generation, and audio-visual synchronization to name a few.

The input data in many tasks of multi-modal video understanding form a sequence, such as the RGB stream, audio waveform, or a piece of text. Therefore, the field was highly influenced by the advances in natural language processing (NLP). In recent years, one of the most prominent advances in NLP, and deep learning in general, was Transformer which was introduced for language translation by Vaswani *et al.* [4] in 2017. Considering that the input encoding in a transformer is non-recurrent, it allowed for learning long-term dependencies, which were somewhat impaired with a recurrent neural net (RNN) due to the sequential encoding of data

in a hidden state.

The first application of the transformer beyond the NLP field was demonstrated for video captioning by [30] in 2018. More specifically, the visual track in a video may be considered a sequence of frame-wise features that can be quite long coupled with a strong performance on the translation task, the transformer was a promising candidate for video captioning (video-to-text translation). This work in video captioning, as well as the adaptation of the transformer’s encoder in language representation learning (BERT) by [1] in 2019, inspired remarkable advances in learning joint visual-linguistic representations that were useful for vision and language downstream tasks [31–36]. By 2020, the transformer became a popular architecture for sequential data.

In early 2021, the transformer was successfully applied to several multi-modal image understanding tasks. Specifically, text-based object detection and segmentation (MDETR) [37] which was inspired by the very first adaptation of a transformer for an image processing problem displayed for object detection and segmentation (DETR) in [3]. This was followed by the text-conditioned image generation (CogView) [38] that was motivated by the success of conditional image generation based on depth, a low-res image, or segmentation mask (VQGAN [39]) as well as text-conditioned image generation in DALL-E [40]. Although the transformer played a significant role in the final performance of these models, the visual features were still extracted by a CNN which meant that the transformer operated on feature representations rather than on raw input data.

At that time, the end-to-end transformer-only backbones started to appear in the literature and attract a lot of attention. Even though the Vision Transformer (ViT) [41], was not the first deep convolution-free image recognition model (see Zhao *et al.* [42]), it was undeniably a milestone work and remains to be the most popular development to this end. ViT has been adapted to many uni-modal settings of multimedia processing, such as action recognition [43], audio classification [44], and point cloud processing [45]. Shortly after, the transformer was generalized to tackle a large variety of uni-modal tasks using raw input data including the above as well as optical flow and playing StarCraft II (Perceiver [46, 47]). ViT was also used in the multi-modal setting such as audio-visual action recognition (MBT) [48], speech recognition (Whisper) [49], and, very notably, for building multi-modal *foundation*

*models*¹ image-text (CLIP and Flamingo) [2, 51], audio-text [52], and other input modalities (PolyViT and Florence) [53, 54].

In this chapter, we present a background of the research areas that inspired this work. The chapter consists of three sections. It starts with an outline of earlier works in multi-modal machine learning and video content understanding (Section 2.1). Next, Section 2.2 provides a deep dive into the methodology of a variety of research areas in multi-modal video understanding that was brought by deep learning. More specifically, the advances in video captioning (2.2.1) and paragraph video captioning (2.2.2), visual question answering (2.2.3), video retrieval (2.2.4), and video moment retrieval (2.2.5) are introduced to provide a conceptual foundation for dense video captioning which is extensively explored later in this thesis. Similarly, advances in textually-guided video generation (2.2.6) are presented to contextualize the contributions of this thesis to visually-guided audio generation. After, the efforts in multi-modal action recognition (2.2.7) are summarized to motivate the design of the two-stream audio-visual synchronization architecture that is presented later. The section concludes with current arts in building multi-modal foundation models (2.2.8) which flooded the deep learning literature and, perhaps, would become standard practice in near future. Finally, Section 2.3 presents Transformer architecture in detail as it forms the basis for the methodology that this thesis contributes.

2.1 Earlier work in multi-modal video understanding

2.1.1 Multi-modal machine learning

Although a deep neural architecture *de facto* became the standard approach for tackling multiple modalities, the interest towards the multi-modal approach sparked long before the deep learning era (2012–) and was emerging in several areas [55].

First, inspired by the McGurk effect [56]², the use of audio and visual modalities has been shown to benefit a speech recognition model [57] (a neural network!) in 1989 and [58–61] (a hidden Markov model, an HMM). Second, as the internet and

¹The term “foundation models” was coined in [50] to refer to a model that was pre-trained (often via self-supervision) with large-scale data and fine-tuned for downstream tasks.

²The auditory illusion that occurs when a person sees an incongruent visual speech signal. For instance, if lips visually pronounce *fa-fa-fa* but the original audio sounds as *da-da-da*, a person will perceive *fa* instead of *da*. Similarly, as lips move to produce *ba*, the original audio (*da*) will be mistakingly perceived as *ba*.

digital technologies were growing in popularity in the early 2000s, multi-modal the research was also dedicated to video content retrieval, indexing [62], and summarization [63, 64]. In particular, audio-visual rule-based models [65–67], along with a vector-quantized codebook [68, 69], or with additional text modality and linear models [70], or, later, HMMs [63, 64, 71–73]. Third, in the 2000s, audio-visual signals were of particular interest in the human interaction community due to the popularity of the annual Multi-modal Interaction Workshop [74], and audio-visual datasets such as AMI Meeting Corpus [75]. A common approach was the support-vector machine (SVM) on hand-crafted features [76–78] and HMMs [79, 80].

Fourth, coupled with the advances in multi-modal human interaction and a growing interest towards the Paul Ekman’s theory of universal emotions [81, 82], the research has been conducted in the area of multi-modal sentiment recognition. Many promising emotion recognition datasets were introduced such as [83–86], including the emotional conversation dataset SEMAINE [87], and an annual audio-visual emotion recognition challenge [88]. Similar to other machine learning models at that time (the 2010s), the approaches relied on SVM [88–92], conditional random fields (CRFs) [91], HMMs [86, 93] and long short-term memory (LSTM [94]) [92] (see more details in [95]).

Fifth, another direction of multi-modal research was dedicated to visual-language applications such as image captioning [96, 97]. In particular, image captioning was tackled as a *retrieval* task, *i. e.* by computing similarities between sentence and visual features [98–100]; or as *generation*, *i. e.* by modelling objects, their attributes, and spatial relationship between them in CRFs followed by an HMM language model that fills in a template [101, 102] or picking the best caption among all proposals using Google N-gram frequencies [103], or filtering the proposals with a set of manually-crafted constraints in the Integer Linear Programming setting [104], or making a syntactic tree [105, 106].

2.1.2 Video understanding

The field of video understanding has flourished recently due to the advances that deep learning methodologies brought to the picture. The new data-driven approaches allowed reaching decent performance on previously unimaginable tasks such as video captioning, question answering, and generation, to name a few. In contrast, earlier works were dedicated to much more modest applications. One of the most explored

areas was action recognition and the tasks from the TRECVID workshop³ that supplied many large high-quality datasets covering multiple research areas including shot boundary detection, audio-visual learning, and video retrieval among others [55]. This section covers milestone ideas in early video content understanding.

The requirements for an action recognition model were outlined in the late 1970s by Marr and Nishihara [107]. In particular, the representation should: a) be easy to compute; b) support a large number of classes; c) be unique from any point of view; and d) be similar, but not the same, between two objects of the same class. However, the earliest attempts at action representation were as modest as representing a walking person on a video with a 3D model of connected cylinders corresponding to the person's parts (WALKER) [108]. In 1994, following the idea of a 3D model of the body, Rohr [109] tackled pedestrian recognition with a Kalman filter.

Nonetheless, building volumetric action models from videos was a tedious and expensive procedure. For this reason, the following research was mostly focused on extracting action representations instead [110]. These models can be classified into two types based on the level of representation: *holistic* (object shape, its movements and structure) and *local* (descriptor features from points of interest).

The earliest work that employed holistic representation was done by Polana and Nelson [111] who classified actions based on periodicity in optical flow using Fourier transform. Later works were strongly influenced by Bobick and Davis [112] who proposed to represent motion depicted with multiple frames in a single image as a changing silhouette, which was extended to volumes later in [113, 114]. However, with these representations, it was difficult to model variations in object appearance, point of view, and changing details within object silhouettes (*e.g.* clapping) [110, 115, 116].

Since the middle of the 2000s, video representations were mostly built at the local level. Earlier works relied on the local descriptors extracted from cuboids that were outlined by an interest point detectors [115, 117] and tracking trajectories [116, 118]. Before training a classifier (*e.g.* an SVM), the extracted local features might have varying sizes per video and, thus, can be aggregated via, for example, a bag-of-visual-words (BoV) aka. a codebook [115, 119, 120] or sparse coding [121, 122]. A comprehensive analysis of prior work on human behaviour on videos can be found in the survey by Borges *et al.* [123].

³trecvid.nist.gov/index.html

Others explored shot boundary detection on a video, *i. e.* a sequence of frames produced by a single camera. Earlier works in automatic shot boundary detections relied on colour histograms, edge change ratio, edge contrast, and standard deviation of pixel intensities [124]. Multiple methods followed a classification approach for boundary detection: rule-based [125] and statistical [126] methods. The shot boundary detection field benefited strongly from the annual TRECVID workshop. In particular, finite state automata (FSA) was used by Zheng *et al.* [127] to solve the task. Yuan *et al.* [128] relied on the graph partition model and an SVM classifier. The information-based approaches were also employed, *e. g.* by modeling entropy of RGB pixels in [129] and also using Speeded Up Robust Features (SURF) in [130]. Others tackled the problem with methods based on linear algebra, *e. g.* QR [131] and eigenvalue [132] decomposition methods. The Singular Value Decomposition (SVD) was used by Lu and Shi [133] to reduce the dimension of features to speed up boundary detection. A survey of many other approaches, including deep learning techniques, one may find in Abdhussain *et al.* [134].

Text-to-video retrieval is another important area of video understanding research. The problem was initially approached by simplifying the problem into text-to-text search which can be achieved by reusing the corresponding subtitles, which might reflect the video content for some applications [135]. However, using raw text to query videos was a challenging task and, instead, many works focused on retrieving videos given a set of “concepts”. The concepts included, but were not limited to, colour, texture, shape, local descriptors, and temporal features. Smith *et al.* [136] tackled image-to-video retrieval using these features extracted from an image to match them to those of keyframes from videos. While Snoek *et al.* [137] proposed to query videos directly with certain concepts. More information on concept-based video retrieval can be found in [138].

Another research direction in video content understanding was video captioning. Earliest works in describing video content with text are dated back to the early 2000s and “fill-in-templates” was a common approach back then. In general, this approach could be described as follows. First, a visual object or action detector extracts information about a subject, a verb/action, and an object. Second, a model attempts to fill in pre-defined templates [139–144]. Sometimes, the resulting caption was filtered according to linguistic data from a database database [143] or to the similarity of the proposed caption to the training samples [144].

2.2 Multi-modal video content understanding with deep learning

Prior to the deep learning era, modality information was encoded with manually-crafted features that were specific for each domain. For instance, textual features were commonly computed with variants of a bag of words (BoW) such as term frequency-inverse document frequency (or tf-idf). A similar technique was adapted for calculating features from an image using the scale-invariant feature transform (SIFT) [55, 145–147], or a bag of “visual” words. In the case of audio, common features were Mel-frequency cepstral coefficients (MFCCs), energy and spectral characteristics, and zero crossing rate [148]. These feature extraction algorithms were replaced with a trained feature extractor that encodes information into dense vector representations which allows stronger performance in solving downstream tasks. This section focuses on applications of deep learning to multi-modal video understanding.

2.2.1 Video captioning

Video captioning is a natural extension of image captioning. It benchmarks the model’s “understanding” of a video from its ability of it to generate a text description (a caption⁴) of video content. Overall, the problem can be viewed as a sequence-to-sequence task (“video-to-text”) and, therefore, the proposed approaches draw on the successful attempts in sequence-to-sequence learning and architectural elements from the fields of activity recognition and natural language processing.

Pioneering works Prior to the “deep learning era” video captioning models were mostly rule-based [139–144]. Nowadays, video captioning architectures follow the encoder-decoder design which was proposed in Rohrbach *et al.* [149] and Donahue *et al.* [150] inspired by machine translation. Early attempts to apply deep learning methods to video captioning involved encoding video frames with a 2D convnet, followed by aggregation via temporal average pooling, which is then used as an input to an LSTM that decodes a caption word-by-word (see Venugopalan *et al.* [151]). Since averaging temporal representations could potentially wipe out important temporal structures, Yao *et al.* [152] suggested to use a weighted average of 3D convnet features where the weights are determined via an attention mechanism. The idea of attention was adapted by Song *et al.* [153] who suggested adjusting it to ignore

⁴Not to be confused with “closed captioning” (CC) which is often referred to as subtitles.

non-visual words to make generated captions more relevant to visual cues. In the follow-up work, Venugopalan *et al.* [154] suggested to use a shared LSTM for both temporal encoding of RGB/flow frames and generating a caption. Xu *et al.* [155] relied on VLAD [18] to obtain better representation of the spatial features.

Semantic “tags” Meanwhile, other works explored the potential to bridge visual and linguistic modality by learning or extracting semantic “tags”. In particular, Rohrbach *et al.* [156] explored the possibility of pre-training 2D CNN feature extractor to classify actions (verbs), places, and objects and reuse the output of such feature extractor to generate a caption. A more explicit connection between visual and linguistic concepts was made via semantic tags or object labels extracted from image and action “visual words” in numerous of works [157–161]. Others suggested using part-of-speech (POS) tags learned from captions to condition the decoding LSTM [162] or a mixture model [163]. While Zhang *et al.* [164] simplified caption generation by retrieving linguistic hints from a text database.

Finer spatial features Several works were dedicated to improving visual representation with region-of-interest features. Specifically, Li *et al.* [165] and Yan *et al.* [166] used region and frame level features aggregated with an attention module. Inspired by image detection, Yang *et al.* [167] used region-based (local) features along with the whole-frame (global) while Ma *et al.* [168] used attention modules to fuse local and global features across time. Similarly, local and global features were used in Wu *et al.* [169] who suggested extracting local features of a CNN following a trajectory which has promising visualization properties. Object interactions were also modelled with a graph in other works [170–172].

Extra memory blocks To utilize the limited LSTM memory more efficiently, Pan *et al.* [173] suggested having a two-staged LSTM: one for sub-clips (local), another to summarize the representation of each sub-clip (global). Baraldi *et al.* [174] suggested detecting a segment for the corresponding caption and encoding (“remembering”) only the content of this segment. While Wang *et al.* [175] and Pei *et al.* [176] suggested having an explicitly designed shared visual-linguistic memory block.

Better training objectives Other works focused on designing a better objective to improve model performance. In particular, Pan *et al.* [177] used both 2D and 3D

convnet features (mean pooled) to prime an LSTM and improved coherence of captioning and training speed with coherence and relevance losses. Wang *et al.* [178] added a reconstruction loss (representation-caption-representation) to improve the training dynamics of the captioning model. Liu *et al.* [179] employed a reconstruction loss that reconstructs visual input features in an autoencoder, and a ranking loss ensures similarity between extracted features and the caption. Elements of Reinforcement Learning (RL) were also employed for video captioning. [180–182] adapted an RL objective to directly optimize the captioning quality metric (CIDEr [183]). An interesting application of RL was suggested by Chen *et al.* [182] who proposed using an additional reward for the “diversity” of selected frames as a majority of the frames are redundant (6–8 frames per video was found to be enough).

Other modalities Other approaches employed multi-modal features. Ramanishka *et al.* [184] (2016) generalized the encoder-decoder architecture presented in [151] (see above) to tackle multiple modalities such as audio, action recognition label as well as motion (3D CNN) and object recognition (2D CNN). With a slight architectural variation, this idea was explored in Jin *et al.* [185] (2016) who also included subtitles. Chen *et al.* [186] proposed to learn audio-visual topics via textual supervision to improve captioning performance. Audio was employed by Xu *et al.* [187], Wang *et al.* [188], and Hori *et al.* [189] who used attention to fuse audio and visual information, while Hao *et al.* [190] also used audio modality and relied on shared weights of LSTM to fuse multi-modal input data.

Attention and transformers Recently, architectural elements of the transformer [4] started to appear in video captioning. Chen *et al.* [191] was one of the first works to use the encoder-decoder transformer for video captioning. Following this work, Pan *et al.* [171] used the transformer specifically for caption decoding Zheng *et al.* [192] used transformer to encode local object-level features and connect them to syntax-guided queries (as “object queries” in DETR [3]⁵). If previous video features were extracted with a pre-trained and fixed feature extractor, Lin *et al.* [193] suggested relying on a trainable transformer to extract features and solve the issue of long input sequences via learning a sparse attention mask. More recent works were dedicated to large-scale transformer pre-training on video-text datasets which is, then, fine-tuned for video captioning, besides many other relevant applications (see page 37).

⁵Interestingly, DETR [3] was published half a year after Zheng *et al.* [192].

Datasets To train a video captioning model, one needs a collection of video-text pairs. Specifically, MSVD [194], MSR-VTT [195], and, more recently, VATEX [196] were among the popular ones, but many other datasets were used as well [197–199].

2.2.2 Video paragraph captioning

Describing a video, that is tens of seconds long, with a single sentence might not be enough because most of the videos contain multiple distinct events. To this end, the video captioning task branched out to captioning with a paragraph that consists of multiple coherent sentences [200]. This idea was followed in Yu *et al.* [201] who approached it with two RNN models: a “local” which operates words and keeps the sentence state and “global” that works with sentences and keeps the paragraph state. Gella *et al.* [202] used the last-step hidden state of an LSTM from the previous caption to prime another LSTM for the current caption in order to preserve coherence within a paragraph. RL objectives were used to a penalty on sentence and paragraph levels in Xiong *et al.* [203] who also suggested using temporal annotation to improve coherence and concise as in dense video captioning (see p. 47). Similarly, Song *et al.* [204] used a sentence-level penalty in the RL loss. To combat the redundancy in generated captions, Park *et al.* [205] introduced a set of discriminators specific for certain tasks, *e. g.* relevance, diversity, and coherence.

Transformers were also used for paragraph video captioning. In particular, object interactions have been modelled with self-attention and, then, used by a language decoder by Zhou *et al.* [206]. Lei *et al.* [207] and Song *et al.* [204] enhanced a transformer with a memory module that helps to encode previously generated caption words within a paragraph. Meanwhile, DETR-like queries [3] were used to pick useful visual cues via cross-attention modules by Wang *et al.* [208].

2.2.3 Visual question answering

Visual Question Answering (VQA) is another important area of multi-modal research. Answering a question given an image was naturally extended to videos (or VideoQA) meaning that earlier deep learning VideoQA methods were heavily influenced by advances in VQA. Thus, we start with a brief outline of works in VQA.

VQA datasets The goal of a visual question-answering model is to pick (multi-choice) or generate a correct answer given an image-question pair. The interest in

this area has been sparked by the emergence of several datasets, such as COCO-QA [209], a larger-scale VQA v. 1.0/2.0 [210, 211], VizWiz-VQA [212], abstract reasoning datasets, such as CLEVR [213] and the artificial-scenes part of the VQA dataset [210].

Common VQA approaches The earlier deep neural approaches to VQA involved extracting a text embedding from a question with an RNN and an image embedding with a CNN or a region-based Bottom-Up and Top-Down Attention (BUTD) [214]. The extracted text and visual features were often fused with a bilinear pooling [215] in [216, 217], a variant of the Tucker decomposition [218] (MUTAN) in [219–221], a dynamic module network [222] in [223, 224], a graph neural network [221, 225, 226], or simply with feature multiplication [227–230]. Since text processing is an essential part of a VQA system, transformers started to appear early for this task and brought significant improvement to the results [32, 231–233].

Common VideoQA approaches VideoQA was pioneered by Tapaswi *et al.* [234]. If an image for VQA was encoded with a variant of a 2D CNN, for VideoQA one should also take care of the time dimension. To this end, it was common to apply a 3D CNN or an RNN on frame-wise (RGB and optical flow) features that were extracted with a 2D CNN. To fuse encoded video and text modalities, approaches relied on attention [235–242], a relation model or feature multiplication [243] and concatenation [244, 245]. A few works employed a concept of memory to account for longer-term dependencies in input sequences [238, 239], while graph neural networks were used for fine-grained modelling of the scene [242, 244, 246–251].

Transformers in VideoQA Similar to video captioning and other research areas, transformers brought a substantial gain in performance here as well [246, 250, 252–257] partially due to the large-scale pre-training as discussed later on p. 37.

VideoQA datasets Nearly every publication in VideoQA was accompanied by a new dataset [234–237, 243, 253, 258–266]. Recently, audio-visual datasets [252, 267, 268] and approaches [252, 267–269] started to appear in the literature. Yet, only a few became commonly used to benchmark proposed models. Specifically, TGIF-QA [235], MSVD-QA, and MSRVTT-QA [236] which are based on video captioning datasets TGIF [270], MSVD [194], and MSRVTT [195]. However, these

datasets were automatically annotated using the available captions, which appears to be the most crucial issue in this field.

2.2.4 Text to video retrieval

The abundance of video data online inspired many attempts for text-based retrieval. On a high level, a text-to-video retrieval model should be efficient in embedding a query (text) and database objects (videos) into a common space where relevant text-video pairs are closer to each other than irrelevant pairs. This section covers milestone works in video-text retrieval with deep learning methods.

Textual representation With the emergence of large-scale datasets and deep learning, the representations of the text and videos became neural and trainable. Specifically, the text query was often vectorized as word2vec [271] or GloVe [6] and aggregated with an RNN in [272–277], by NetVLAD [278] in [279], by a Gaussian mixture model [280], or by mean pooled word2vec vectors [276, 281], semantic graph and attention [277], skip thought vectors [282] in [283], directly in a GRU [284, 285], and, most recently, by BERT [1, 286, 287] in [288–290] and other transformer architectures [291, 292].

Visual representation The video representation was formed frame-wise with an image recognition backbone (2D CNN) [272–276, 283, 285, 286], often coupled with an action recognition model (3D CNN) [277, 279–281, 284, 288, 291, 292] or a spatial-temporal graph [290]. In earlier works representation was aggregated by a temporal pyramid scheme [272], average/max pooling [273, 274, 276, 279–281, 283, 284, 287, 290, 291], RNN [274–276, 285], and attention [275]. More recently, transformers became a popular choice for this end [286, 288, 289, 292, 293].

Training objectives The models are trained with max-margin ranking loss [273–277, 279, 280, 283–285, 288–291]. However, other objectives were also used such as the mean squared error loss [281], Euclidean distance [272], a variant of InfoNCE loss [294] in [286, 287, 295, 296], or a combination of max-margin, cluster [297], and cycle consistency losses [298] in [292]. Sometimes, video captioning loss is used to facilitate training [295, 296].

Inference metrics During inference, the similarity between the video and text embeddings are usually computed with a certain similarity metric, *e. g.* cosine similarity [276, 280, 281, 284–287, 289–291, 295, 296], Euclidean distance [272, 273, 283], an MLP [274, 275], and dot-product [279, 288].

Datasets The most common datasets for training text-to-video retrieval models are MSVD [194], MSR-VTT [195], LSMDC [299], DiDeMo [273], ActivityNet Captions [198], and VATEX [196], yet many others are available as well [197, 199, 270, 300, 301]. Nonetheless, recent developments in large-scale video-linguistic pre-training are strongly influenced by HowTo100M [302] due to its size (see page 37).

Recent developments A few notable ideas have recently appeared in the literature. For instance, other descriptors (aka. “experts”) are being involved to improve performance, *e. g.* audio, motion (inc. optical flow), speech-to-text, optical character recognition (OCR), and face tracks [275, 279, 288, 289, 291, 293]. Others [277, 289, 295] argued against feature aggregation and propose to switch from global (video to sentence) to local (a chunk of words to chunks of video features) alignment. Alternatively, the problem of video-to-text retrieval can be re-formulated from “sentence to clip” to “paragraph to combination of clips (a video)” retrieval [286, 292, 297]. The idea of a dual encoder that embeds video and text features into a common metric space was challenged in [303, 304] who suggested shifting towards multi-modal fusion with a transformer attention which was expected to be more capable. Recently, transformer-only approaches started to appear in the literature [295, 296, 305]. Also, a new chapter of multi-modal retrieval was opened with the availability of large-scale pre-training (such as HowTo100M [302]) and transformer which is discussed on page 37.

2.2.5 Video moment retrieval

The task of video moment retrieval requires a model to localize a specific segment within a video that is described by a text query. This task is also known as “natural language video localization” (NLVL). The problem of video moment retrieval bridges video retrieval and action detection. Specifically, for the video retrieval task, one would index a database of videos but, for video *moment* retrieval, the search is conducted within one video and the model is expected to localize the event described

by a sentence or paragraph query. Similarly, if for the action detection, the query is discrete label space, for a video moment retrieval it is an arbitrary piece of text.

Fixed proposals Initial attempts in video moment retrieval were relying on query-independent temporal proposals, which allowed utilizing advances in video retrieval. In particular, [273, 306–312] suggested splitting a video into a set of equal segments and running the search on them. However, this requires checking each segment regardless of its relevance to the query. As well as, the resulting temporal boundaries are inflexible.

Anchor-based proposal generation Many attempts have been made to make proposals more flexible. One direction of such efforts is “anchor”-based methods. For instance, Chen *et al.* [311] proposed to model frame-word mapping via an LSTM and multi-scale proposal candidates. Xu *et al.* [313] enhanced an action proposal model (R-C3D [314]) with an attention mechanism that uses text hints from the text query and a captioning loss was used. Chen *et al.* [312] to model cross-modal interactions with attention and gating modules. A fine-grained approach to modelling query semantics was suggested by Yuan *et al.* [315]. Wang *et al.* [316, 317] explored the idea of both coarse- and fine-grained cross-modal interaction and Gao *et al.* [318] focused on inference speed. Zhang *et al.* [319] used a graph convolutional network (GCN) [320] to iteratively adjust candidate event representations. In the meantime, Ge *et al.* [321], showed a way of accounting for activity concepts in visual and language cues to improve proposal generation. Mithun *et al.* [322] tackled the task in a weakly supervised way which might be beneficial as the human annotation of clip-query pairs is tedious. Instead of considering each clip-query pair individually, Zhang *et al.* [323] outlined a way of accounting for the relationship between multiple queries that a video might have. Zeng *et al.* [324] used a graph to model object-word interactions. Despite being simple and straightforward, anchor-based methods still reach strong performance [325].

Towards finer proposal boundaries Another class of approaches regress the temporal segment boundaries instead of using predefined proposals. This approach was pioneered by Yuan *et al.* [326] who used cross-modal attention followed by an MLP to determine the segment boundaries. Chen *et al.* [327], similar to [321], used joint learning of activity and word concepts. Ghosh *et al.* [328] used LSTM outputs to

predict the segment boundaries. Zhang *et al.* [329] proposed to formalize the task as a span-based question-answering problem. Reinforcement learning (RL) setting was used in multiple works to iteratively refine the temporal segment boundaries [330–332] and spatial regions [333]. Using annotations more “densely” was explored by Lu *et al.* [334] who added a binary classification head for each frame whether it belongs to the ground truth segment. Similarly, Zeng *et al.* [335] made a model to predict the distance to the start/end points from each frame. Also, fine-grained multi-modal interactions have been explored in many works. For instance, Mun *et al.* [336] considered the text query as a set of multiple semantic phrases, while Chen and Jiang [337] built a bi-partite graph between visual objects and words. Chen *et al.* [338] suggested using audio and motion that were ignored before, as well as a way to model inter-modality interactions. Yang *et al.* [339] and Nan *et al.* [340] drew attention to spurious correlations of the datasets and suggested using causal interventions. Zhao *et al.* [341] used cascading to refine boundary prediction at different scales of visual-textual features.

Multi-modal fusion and prediction The query and visual representations were often fused by concatenation, dot product, or addition that was followed by either an MLP [306, 315, 318, 321, 327, 330, 331, 335, 336], attention [309, 315, 322, 338, 340–344], CNN [323], or a graph [324]. While Liu *et al.* [307, 312, 329, 334] relied on attention mechanisms between two modalities, as well as a cross-modal gating mechanism coupled with a GRU [312] and [311, 313, 328] fused visual and text modalities in an LSTM.

Transformers Similar to contemporary works in other areas, elements of the transformer architecture started to appear in the literature. For instance, Zhang *et al.* [343] used the transformer encoder for modelling long-term dependencies in a video along with a syntactic GCN for the text. Lin *et al.* [345] used two transformers for making proposal candidates, re-rank them based on how much information was lost after masking query words. Liu *et al.* [346] suggested capturing query representation on word-, phrase-, and sentence levels and to model visual-linguistic interactions in a graph with transformer attention. [347–350] employed transformer’s cross-modal attention to fuse visual-linguistic features for finer cross-modal integration while Wang *et al.* [344] relied on the transformer encoder to model the temporal interactions of multi-modal features. In contrast, Zhang *et al.* [351] modeled visual-textual

interactions jointly in a concatenated sequence passed to a transformer encoder (as in VideoBERT [31]). Similarly, the transformer was used in Liu *et al.* [352] who also adapted linguistic dependency parsing [353] for visual data. Wang *et al.* [354] used a transformer to fuse visual-linguistic features and additionally suggested avoiding penalising semantically similar moments within a batch during contrastive training. While DETR-like approach [3] was used by Lei *et al.* [301].

Textual representation The processing of a text query, by and large, resembles the one that is used in video retrieval. For instance, Skip-thought vectors [306, 309, 310, 321, 327, 330, 331], word2vec [306, 313, 324], GloVe [273, 306, 308, 309, 311, 312, 315, 317, 319, 321, 323, 326–329, 334, 335, 337, 338, 340–344, 346–350, 352, 354] vectors often encoded with an RNN or mean pooled. In recent work, text encoding is done mainly with variants of a transformer [351].

Visual representation Similarly, the visual representation was encoded with contemporary (often pre-trained for image or action recognition) feature extractors. In particular, 2D [273, 308, 309, 311, 318, 322, 323, 327, 331, 335, 338, 341, 342, 349, 350] and 3D [273, 306, 308–313, 315, 317–319, 321–324, 326, 328–330, 334–336, 338, 340–344, 346–349, 351, 352, 354] CNNs.

Datasets The use of the datasets is remarkably consistent among the works. Specifically, the most common datasets are video captioning datasets with temporal annotations for events. In particular, TACoS [200], Charades-STA [306], ActivityNet Captions [198], and, but less often, DiDeMo [273].

2.2.6 Text-guided video generation

One of the applications that were barely imaginable before the deep learning era was video generation. Many works in video generation focused on predicting future frames given a few ground truth frames (a prime) [355–369] while others explored unconditional video generation [358, 361, 368–374]. In an attempt to guide the generation, recent research has been dedicated to text-conditioned video generation.

Pioneering works Earlier works in text-conditional video generation focused on simple scenes, *i. e.* moving digits (MNIST) [355], side views of a walking person, top-view cooking videos (subsets of [194, 375]), and scenes from a popular cartoon [376].

In particular, Mittal *et al.* [377] generated a frame sequence using a text-conditioned LSTM which acts as a decoder part of a variational autoencoder (VAE) during inference. Li *et al.* [378] suggested combining a conditional VAE, which augments the latent vector with RNN text features, and generative adversarial networks (GAN) [5] that generates a set of images given a text-based motion-augmented input vector. Pan *et al.* [379] also employed a combination of an RNN and a GAN, yet with three discriminators to criticize relevance to the input text as well as the temporal and spatial coherence of the generated video. Whereas, Liu *et al.* [380] reduced the “VAE blurriness” and GAN’s training instability with an LSTM that takes in linguistic features that were optimized during training to match the visual features extracted from a real video coupled with a cycle-consistency loss [381]. An alternative approach was suggested by Gupta *et al.* [376] who generated scenes of a popular cartoon (“Flintstones”), using a template-based approach: first, the model breaks down the input text to compose a scene-character template, then, it retrieves the character and scene background that have the closest embeddings from a database. Although the results were promising, generating open-domain videos remained to be a challenge.

Approaches with conditional autoregressive sampler Recent works have been inspired by the striking results of two concurrent works: VQGAN [39] and DALL-E [40]. Both are based on training in a two-staged model which intuitively can be described as “first, make a set of (visual) building blocks; second, learn to arrange these blocks given a (textual) prompt”. More specifically, first, a vector-quantized variational autoencoder (VQVAE) [19] is trained to encode an image to a quantized representation consisting of codes from a learnable codebook and decode this representation back to the original image as closely as possible; and, second, an autoregressive model (*e. g.* a transformer [4]) is trained to sample the quantized representation from the codebook given a text token sequence. During inference, the sampled quantized representation can be reconstructed into an image using the decoder part of the pre-trained autoencoder (first stage). One could reuse the image-based autoencoder codebook (first stage) and train the autoregressive sampler (second stage) to generate multiple frames consecutively. Examples of this approach are GODIVA and, later, NÜWA (Wu *et al.* [382, 383]). As well as, CogVideo (Hong *et al.* [384]), which relies on a “frozen” CogView-2 (Ding *et al.* [385]), also used text-to-image model with a progressive resolution that improves the generation speed, resolution, and the quality of samples compared to earlier works. Nonetheless, this image-based ap-

proach leads to poor temporal modelling. Phenaki (Villegas *et al.* [386]) addresses it by keeping a track of the first and a few most recent frames. The 3D version of VQGAN was proposed by Yan *et al.* [368] for video generation was later adapted and improved by Ge *et al.* [387] for textually-guided video generation.

Diffusion models Even more recent advances were sparked by impressive results of diffusion models in text-to-image generation, *e. g.* ADM [388], GLIDE [389], and DALL-E 2 [390]. A diffusion image generation model can be described as follows. Since, an image (step 0) can be transformed into Gaussian noise (step T) after adding a small amount of noise at every step $t \in [0, T]$, a model (*e. g.* U-Net [391]) can be trained to reverse (denoise) a t^{th} step given a pair of images at steps t and $t - 1$ as well as the original image as a reference. During sampling, the model inputs noise and applies T consecutive denoising steps. To “guide” sampling, *e. g.* with a data class or text, the denoising step is conditioned on gradients from an independently trained noisy-image classifier [388] (classifier guided diffusion) or from CLIP [7]. To get rid of the additional pre-trained model, one may train a conditional diffusion model by randomly replacing the condition embeddings with zeros (classifier-free guidance [392]). Rombach *et al.* [12] suggested to “diffuse” in the latent space instead of the pixel space to reduce the cost of training (aka. latent diffusion model (LDM) or Stable Diffusion). The image-based diffusion model with classifier-free guidance was adapted for the video domain by Ho *et al.* [358] (VDM). While, Singer *et al.* [393] (“Make-a-Video”) and Ho *et al.* [394] (“Imagen Video”) pre-trained a small-resolution text-image diffusion model and used a sequence of independently trained interpolation and spatio-temporal super-resolution blocks.

Datasets Textually-guided video generation requires a large dataset with video-text pairs. Since the field is still heavily under development (most of these works has been released in 2022), the selection of datasets among works is diverse. HowTo100M [302], MSR-VTT [195], TGIF [235], VATEX [196], WebVid [395], MUGEN [396] were among those, yet it is also common to rely on internal private datasets [394].

2.2.7 Multi-modal action recognition

Activity recognition is considered to be the main research area in video content understanding. A model that was pre-trained for action recognition often forms the

basis for other video understanding problems that require strong video representations. Multi-modal research in activity recognition was mostly dedicated to employing auditory cues from an audio track that is naturally present in a video file.

In earlier works, the visual track was encoded with a variant of a 2D or 3D CNN, which formed the main contribution of a paper. In turn, the 1D audio waveform is processed with Short-Term Fourier Transform (STFT) that extracts a spectrogram (image-like 2D representation), which is sometimes followed by a log-Mel-scale transform. The spectrogram is encoded by a 2D CNN that resembles an image recognition model. It is also common to employ optical flow frames or a skeleton (pose) to improve performance. The modalities are often fused with concatenation [397–404], but some other exotic fusion strategies exist, *e. g.* lateral connections (addition) along the visual network [405], adaptive fusion of sub-network outputs based on a data class [406, 407], attention [408, 409], transformer [410–413], or neural architecture search (NAS) [414]. Recently, transformer-only multi-modal backbones started to appear in the literature [48, 415].

Interestingly, even though a stronger performance is expected, adding other modalities (*e. g.* audio) to a model leads to the decline in model performance [404, 405]. This phenomenon was attributed to the overfitting that occurs at different rates for visual and audio sub-networks. To mitigate this, re-weighting of losses based on overfitting dynamics is proposed in [404] and modality-dropout in [405].

Although most of the activity recognition datasets were mainly designed for visual recognition, these are still being used for multi-modal training. Specifically, HMDB [416], UCF101 [417], Kinetics [418], and EPIC-KITCHENS [419, 420].

2.2.8 Multi-modal video foundation models and applications

Conventional wisdom has it that a deep learning model benefits from large-scale pre-training when it comes to its performance and generalization capabilities. When a practitioner attempts an image classification problem, a common practice is to use a model that was pre-trained on a large-scale general-purpose dataset, such as ImageNet [22], and fine-tune it on the target, often small-scaled, dataset. Following this practice, one might expect not only stronger performance but also faster convergence, compared to training the same model from scratch.

Unlike image classification, other supervised learning tasks might require more effort during annotation, such as video question answering. For this reason, a high-

quality large-scale dataset might not be available. Thus, a large body of recent work in video understanding has been dedicated to multi-modal (often visual-text) pre-training on the datasets that are noisy but easy to collect at a large scale (*e. g.* HowTo100M [302]). It turns out that pre-training a model to solve a learning problem that is only useful for the sake of pre-training, *e. g.* to contrast between positive and negative video-text pairs, allows learning useful representation for desired (down-stream) tasks. Hence, these are called “foundation models” [50].

Besides substantial gains in performance after fine-tuning on down-stream tasks, foundation models have strong zero-shot capabilities⁶ on tasks with text labels. More specifically, a model that was pre-trained contrastively on video-text pairs can be used to compare an embedding of a video clip to the embedding of each class label of an “unseen” dataset. This can be achieved by simply re-formatting class labels as captions by filling a template like “a video clip of data_class”. Since a model was pre-trained to score video-text correspondence, designing this straightforward “communication device” to the model allows applying it to “unseen” datasets making it an exciting step towards building a general-purpose model.

Pre-training modalities and applications When it comes to the video-text foundation models, the pre-training is often done with two modalities: vision (an RGB stream) and text [31, 35, 421–439]. Notably, the possibility of reusing paired image-text data during pre-training has been explored in a variety of works [425, 427, 428, 430, 432, 433, 440], while others also employed the audio modality [435, 441–444]. Once the model is pre-trained on this data, it has the potential to be applied with and without fine-tuning on a many down-stream tasks such as text-video retrieval [35, 421–427, 429, 431–435, 437–439, 441, 442, 445], action recognition [31, 423, 427–429, 434, 435, 438, 441, 442, 444], video captioning [31, 422, 424, 425, 427, 431, 438], action segmentation [426, 431, 434, 445], video question answering [35, 421, 422, 427, 431–433, 436, 437, 439, 443, 445], action localization [434, 445], visual common sense reasoning [443], and action anticipation [443, 444]. An attempt to unify the evaluation procedure of foundation models on multiple down-stream tasks, Li *et al.* [446] (VALUE) suggested relying on a suite of 11 down-stream video understanding datasets across 3 tasks (text-to-video retrieval, video question answering, and video captioning).

⁶Zero-shot performance is referred to here in a broader sense (as in CLIP [2]), *i. e.* performance of a model on “unseen” datasets/tasks instead of performance on “unseen” classes of a dataset.

Architectures The idea of large-scale pre-training and fine-tuning on other tasks is not novel. However, the multi-task evaluation setting certainly gained popularity after the success of transformers, such as BERT [1], GPT [7–9], and CLIP [2]. Although earlier models were 2D/3D CNNs [35, 421, 435], most of the foundation models are transformers nowadays [31, 35, 422–425, 427, 428, 431–433, 436, 437, 439, 442–445].

Datasets Research in large-scale video-text pre-training was strongly influenced by How-To-100M [302], a dataset of 100M+ “how-to” YouTube videos paired with subtitles extracted by an automatic speech recognition system (ASR). Recently, other suitable datasets started to appear in the literature, *i. e.* WebVid [395], YT-Temporal-180M [436], as well as internal databases [31].

2.3 Transformer architecture

Transformer architecture [4] is one of the most prominent ideas in the past few years. Its non-autoregressive nature and attention mechanism allow long-term dependence modelling and parallelisation capabilities. First, it took by storm natural language processing (NLP) community as a German-English translation model [4], context-aware text representation model (BERT) [1], and language model (GPT) [7–9]. Shortly after, it started to appear in the computer vision community, *e. g.* image classification (SAN, ViT, Swin) [41, 42, 447], object detection (DETR) [3], and image generation (TransGAN) [448] as well as action recognition (TimeSformer, ViViT, MBT) [43, 48, 449]. The transformer has been applied in other areas, such as audio classification (AST) [44] and reinforcement learning (Decision Transformer) [450].

This section defines the vanilla transformer architecture as outlined by Vaswani *et al.* [4]. Although encoder-only versions are more common nowadays (BERT, ViT, GPT), defining the transformer in its original form for language translation is beneficial for several reasons. First, the architecture has been remarkably consistent and was adapted with minimal changes for other tasks making it a general framework while the rest are special cases of it. Second, the concept of “softening the dictionary look-up” with queries-keys-values attention is more intuitive for language translation. Third, the cross-modal potential of a transformer is easier to appreciate.

Tokenization for text The process of tokenization allows us to build an “interface” between raw data and the model. The transformer was designed to tackle sequence-to-sequence modelling and language translation in particular. Therefore, text tokenization is defined next. For the sake of simplicity, the following explanation relies on word-level tokenization⁷. A *token* is a piece of input data. For instance, “A cat is riding a bike.” can be tokenized as “a”, “cat”, ..., “bike”, “.” word tokens. Once the tokenization has been applied across the whole dataset, a vocabulary is built as the token-to-index mapping, meaning that each word is replaced with a unique integer. Now, these tokens can be embedded into d -dimensional space with a fully connected layer. This process and the output vector are called *embedding*. For the t^{th} token of the sentence that has i_t index in the vocabulary (vocab), a token embedding (x_t^{emb}) is calculated as a dot-product between by a matrix with trainable parameters $W \in \mathbb{R}^{|\text{vocab}| \times d}$ and a one-hot vector with identity at the index i_t :

$$x_t^{\text{emb}} = \text{OneHot}(i_t, |\text{vocab}|)W \quad (2.1)$$

It is equivalent to picking the t^{th} row of W and that the matrix W holds embeddings for each word from the vocabulary. Thus, in practice, the operation in Equation (2.1) is implemented as an index look-up to W as it is more computationally efficient than the dot-product. Ultimately, the transformer inputs a sequence of token embeddings extracted from each word in a sentence.

Architecture overview As shown in Figure 2.1, the encoder-decoder Transformer has three major blocks: encoder, decoder, and generator. For the German-to-English translation, the encoder makes a representation of a sentence in German:

$$\mathbf{z} = \text{Encoder}(\mathbf{x}), \quad (2.2)$$

where $\mathbf{x} = (x_1, x_2, \dots, x_{|\mathbf{x}|}) \in \mathbb{R}^{|\mathbf{x}| \times d}$ is a input sequence of token embeddings with dimension d from German words and $\mathbf{z} = (z_1, z_2, \dots, z_{|\mathbf{x}|}) \in \mathbb{R}^{|\mathbf{x}| \times d}$ is the encoder output which has the same length. The decoder uses the outputs of the encoder \mathbf{z} as well as embeddings of previously generated tokens (up to t) of the target English

⁷However, other types of text tokenization are available, e. g. as characters, and word pieces [451, 452] (as in BERT or GPT) that are commonly used today and serve as a controllable middle ground between word- and character-level tokenization.

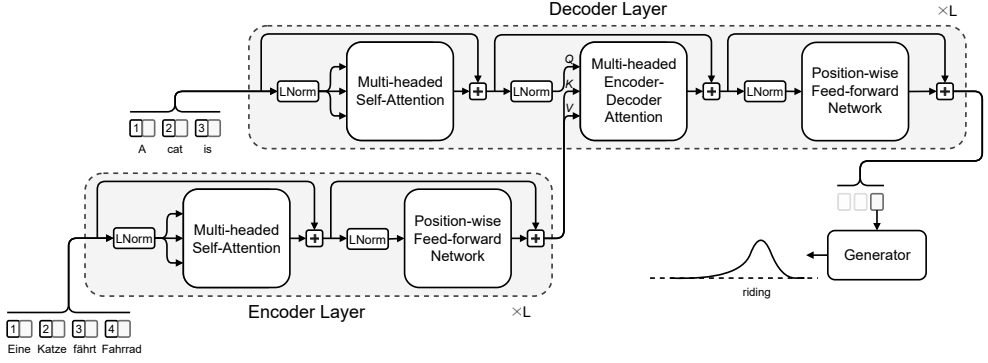


Figure 2.1 Transformer architecture. An example of translating “A cat is riding a bike” from German to English is shown. The model consists of three blocks: an Encoder, Decoder, and Generator. Both the encoder and decoder have L layers. The input token embeddings are summed with positional embeddings (1, 2, ...). Each sub-layer of encoder and decoder layers has a layer normalization (“LNorm”) and residual connection around it.

sentence $\mathbf{y}_{\leq t} = (y_1, y_2, \dots, y_t) \in \mathbb{R}^{t \times d}$ to output $\mathbf{g}_{\leq t} = (g_1, g_2, \dots, g_t) \in \mathbb{R}^{t \times d}$:

$$\mathbf{g}_{\leq t} = \text{Decoder}(\mathbf{y}_{\leq t}, \mathbf{z}). \quad (2.3)$$

This output of the decoder (g_t) is used by the generator to produce the probability distribution for the next token ($t+1$) from the pre-defined English vocabulary:

$$p_{t+1} = \text{Generator}(g_t). \quad (2.4)$$

Dot-product attention The concept of *scaled dot-product attention* is defined with *queries* (Q), *keys* (K), and *values* (V) abstractions:

$$\text{Attention}(Q, K, V) = \text{Softmax}_{\text{row}} \left(\frac{QK^T}{\sqrt{d}} \right) V, \quad (2.5)$$

where Q, K, V are sequences of d -dimensional vectors and the Softmax is applied row-wise. Notice, K and V must be of the same length and the output of the attention operation has the same size as Q . Intuitively, a *value* vector at position i is re-weighted with weights determined by the softmaxed dot-product between the *query* vector at position i and each row of *keys*. The scaling by $1/\sqrt{d}$ is applied to keep the softmax gradients within the non-zero region [4]. The attention mechanism allows accessing information at a specific position from every position of the input sequence.

Whereas, an RNN cell updates its hidden state at each position which leads to poor temporal modelling as information from distant positions might vanish.

Multi-headed attention Let's assume that the *multi-headed attention* has H (e. g. 8 or 16) heads. Having multiple heads allows the block to learn multiple representation sub-spaces of a smaller size in parallel by splitting the input dimensions (d_q) into H chunks of size $d_{in} = d_q/H$ [4]. The multi-headed attention (MHA) is defined as

$$\text{head}_b(q, k, v) = \text{Attention}(qW_b^q, kW_b^k, vW_b^v) \quad \text{for all } b \in [1, H] \quad (2.6)$$

$$\text{MHA}(q, k, v) = [\text{head}_1(q, k, v), \dots, \text{head}_H(q, k, v)] W^{\text{out}}, \quad (2.7)$$

where q , k , and v are sequences of d_q , d_k and d_v -dimensional vectors, $W_b^* \in \mathbb{R}^{d_* \times d_{in}}$ and $W^{\text{out}} \in \mathbb{R}^{d_{in} \cdot H \times d_q}$ are trainable parameters, and $[\]$ is a concatenation operator across the first dimension and Note that the size of the $\text{MHA}(q, k, v)$ output corresponds to the size of q . For simplicity, it is often assumed that $d_q = d_k = d_v = d$.

Position-wise feed-forward network The feed-forward network (FFN) follows a multi-headed attention layer in each encoder and decoder layer. It is sometimes called *position-wise fully-connected network*. The architecture of the network is as simple as a 2-layer MLP that is applied at each position of the input sequence, *i. e.* the weights are shared across all positions:

$$\text{FFN}(x) = \text{ReLU}(xW_1 + b_1)W_2 + b_2, \quad (2.8)$$

where x is a d -dimensional input vector, ReLU is a rectified linear unit non-linearity, and $W_1 \in \mathbb{R}^{d \times 4d}$, $W_2 \in \mathbb{R}^{4d \times d}$ and biases b_1, b_2 are weights to be trained. Although the motivation behind FFN was not discussed in the original paper, yet it is evident that the FFN adds trainable parameters and, thus, increases model capacity.

Encoder The encoder is a stack of L (e. g. 6) encoder layers. Given an input sequence of token embeddings, it makes a representation of it which the decoder uses as context later. The output from an encoder layer is used as the input to the next layer: Each encoder layer l consists of two sub-layers: *multi-headed self-attention* (MHA) and *position-wise feed-forward network* (FFN). Inputs to each sub-layer are normalized with layer normalization (LNorm) [13] and have a residual connection [453]:

$$\bar{\mathbf{z}}^l = \text{LNorm}(\mathbf{z}^l) \quad (2.9)$$

$$\mathbf{r}^l = \mathbf{z}^l + \text{MHA}(\bar{\mathbf{z}}^l, \bar{\mathbf{z}}^l, \bar{\mathbf{z}}^l) \quad (2.10)$$

$$\bar{\mathbf{r}}^l = \text{LNorm}(\mathbf{r}^l) \quad (2.11)$$

$$\mathbf{z}^{l+1} = \mathbf{r}^l + \text{FFN}(\bar{\mathbf{r}}^l), \quad (2.12)$$

where $\mathbf{z}^l \in \mathbb{R}^{|\mathbf{x}| \times d}$ is the input sequence to the layer l as well as $\mathbf{z}^0 = \mathbf{x}$ and $\mathbf{z}^L = \mathbf{z}$. Since all inputs to MHA are the same, it is called *self*-attention. The self-attention allows an element at each output position to build complex relationships with tokens at any position of the input sequence, including the element in its position.

Decoder Similar to the encoder, the decoder is a stack of L decoder layers. It inputs embeddings of the previously generated tokens ($\mathbf{g}_{\leq t}^0 = \mathbf{y}_{\leq t}$) and the output from the last encoder layer (\mathbf{z}). A decoder layer consists of three sub-layers: multi-headed self-attention, *multi-headed encoder-decoder attention* (MHA), and position-wise fully-connected network (FFN). Similar to each encoder layer, layer normalization and the residual connection is used in each sub-layer:

$$\bar{\mathbf{g}}_{\leq t}^l = \text{LNorm}(\mathbf{g}_{\leq t}^l) \quad (2.13)$$

$$\mathbf{b}_{\leq t}^l = \mathbf{g}_{\leq t}^l + \text{MHA}(\bar{\mathbf{g}}_{\leq t'}^l, \bar{\mathbf{g}}_{\leq t'}^l, \bar{\mathbf{g}}_{\leq t}^l) \quad (2.14)$$

$$\bar{\mathbf{b}}_{\leq t}^l = \text{LNorm}(\mathbf{b}_{\leq t}^l) \quad (2.15)$$

$$\mathbf{u}_{\leq t}^l = \mathbf{b}_{\leq t}^l + \text{MHA}(\bar{\mathbf{b}}_{\leq t'}^l, \mathbf{z}, \mathbf{z}) \quad (2.16)$$

$$\bar{\mathbf{u}}_{\leq t}^l = \text{LNorm}(\mathbf{u}_{\leq t}^l) \quad (2.17)$$

$$\mathbf{g}_{\leq t}^{l+1} = \mathbf{u}_{\leq t}^l + \text{FFN}(\bar{\mathbf{u}}_{\leq t}^l), \quad (2.18)$$

where $\mathbf{g}_{\leq t}^l \in \mathbb{R}^{t \times d}$ is an input sequence to the layer l and $\mathbf{z} \in \mathbb{R}^{|\mathbf{x}| \times d}$ are the encoder outputs. Note that the sequence of the encoded German tokens \mathbf{z} is at the places of *keys* (K) and *values* (V) in the Equation (2.16), and the embeddings of the English tokens at that sub-layer serve as *queries* (Q). It allows each element of the English embedding sequence ($\bar{\mathbf{b}}_{\leq t}^l$) to have access to every position of the encoded German sentence \mathbf{z} . This query-context (English-German) routing mechanism has a strong potential for cross-modal research.

Generator The output from the last decoder layer ($\mathbf{g}_{\leq t}^L = \mathbf{g}_{\leq t}$) is used in the *generator* that models the probability distribution of the next word of the English translation across the pre-defined vocabulary. Typically, it is defined as a fully connected layer with softmax applied on the last element of the decoder output (g_t):

$$p_{t+1} = \text{Softmax}(g_t W^G + b^G), \quad (2.19)$$

where $W^G \in \mathbb{R}^{d \times |\text{vocab}|}$ and biases $b^G \in \mathbb{R}^{|\text{vocab}|}$ are trainable parameters. The next token can be sampled from a multinomial distribution with the weights p_{t+1} .

Where to start and when to stop? The role of special tokens During inference, the transformer does not have previously generated tokens. A common technique to this end is to have a special token prepended to each target sequence during training with a *starting* token (<START>). It is added to the vocabulary along with other tokens during the tokenization process. Therefore, during inference, to predict the first word of the target sentence, the transformer inputs the embeddings from the source sequence (German) in the encoder and the embedding of the start token from the target sequence in the decoder. Similarly, the *ending* token (<END>) is appended to the end of the target sequence and added to the vocabulary. This allows the model to learn to “signal” when to stop sampling.

Positional encoding The transformer does not have a “sense” of order in an input sequence. In other words, it is *position-invariant*, *i. e.* if the input sequence is be randomly permuted the output will be intact⁸. For this reason, *positional encoding* is added to the input sequence. Originally (Vaswani *et al.*), the positional encoding was defined as alternating cosine and sine functions with varying frequencies:

$$PE(t, i) = \begin{cases} \sin\left(\frac{t}{10000^{2i/d}}\right) & \text{if } i \text{ is even} \\ \cos\left(\frac{t}{10000^{2i/d}}\right) & \text{if } i \text{ is odd,} \end{cases} \quad (2.20)$$

where t is an index of the token in the input sequence and capped with the longest sequence in the training set and $i \in [1, d]$ is an index from latent dimension. In prac-

⁸It is easy to see when inspecting the attention mechanism (Equation (2.5)). Attention is defined as a set of dot products between sequences. The dot-product is a sum of element-wise multiplications, in which each pair of elements can be swapped with another pair (equivalent to switching the position of the input sequence) without changing the result of the dot-product. Hence, the positional invariance.

tice, positional encoding is represented as a 2D matrix with d columns pre-calculated beforehand. Nowadays, however, the sine/cosine positional encoding is rarely used and was replaced with trainable parameters.

Objective If the transformer is trained to predict the next word, *cross-entropy loss* is used. *Label smoothing* [454] is also commonly used to avoid the model being overly “confident” in its predictions because the ground truth may be noisy or, in the case of text modelling, words may have synonyms. Also, during training, the sequence of “previously generated” tokens is replaced with ground truth tokens, which is also known as *teacher forcing* to avoid error accumulation. To combat overfitting, the *dropout* [455] is used across the architecture.

Masking Another important detail that one should take care of during training is token *masking*. Notice that the forward pass computes predictions for the next token for each position of the input sequence. Therefore, one may plug in the whole ground truth sentence, run a forward pass just once and compare the outputs to the ground truth. However, since the self-attention mechanism allows the decoder to attend to all positions, including next positions, the model might cheat by simply taking the ground truth from the $t+1$ position when making a prediction at position t . To avoid this, masking with $-\infty$ is applied to the QK^T values such that the position that are higher than t have a zero weight and cannot route their information after applying the softmax in Equation (2.5). The masking is applied by replacing the values of QK^T that are above and to the right of the diagonal elements with $-\infty$.

Making batches Similar to other networks, during training inputs to a transformer are batched together. In general, the inputs might have different lengths (sentences have a different number of words). To this end, *padding* tokens (<PAD>) are used to extend shorter sequences to match the longest sequence within the batch. The padding tokens should be masked out as described above.

Tokenization for other types of data So far, the concept of token embeddings was shown in the text data. Recently, the transformer was adapted for many other tasks, including computer vision. Considering the structured nature of the text, tokenization is rather straightforward and can be done on the word, character, or word-chunk levels. When it comes to image input, more insides are needed. The

most well-known approach is to split an image into 16×16 px patches and embed them into a d -dimensional representation using a fully connected layer as suggested in ViT [41]. However, it was not the first attempt to “tokenize” visual data. In particular, the features extracted from each video frame were used as tokens for video captioning by Zhou *et al.* [30]. Also, a 2D feature map that was pre-extracted with a 2D CNN can be used as in DETR [3]. Another approach to visual tokenization is a VQVAE which encodes and, then, quantizes the representation with tokens/embeddings from the discrete codebook (VQGAN, DALL-E) [19, 39]. These approaches might also be straight-forwardly generalized into the 3D video data, to 1D audio waveforms (Jukebox [456]), and 2D audio spectrograms (AST [44]). In contrast to all previous visual and audio tokenization methods, Perceiver (Jaegle *et al.* [46, 47]) treats a raw data value (of a pixel or a waveform) as a token.

Known limitations Although the transformer has a theoretical capability of modelling within-sequence interaction of arbitrary-long sequences, it quickly becomes infeasible in practice when the input sequence grows in size. Notice that if the input length is t , the shape of QK^T in Equation (2.5) is $t \times t$ which means the (GPU) memory footprint grows quadratically with respect to the input length. However, this issue might be temporal as better hardware is being actively developed. Another problem with the transformer architecture is the weak performance on small-scale datasets. It was experimentally shown that the transformer start to “shine” when it is trained on large-scale datasets and benefits more from larger pre-training compared to more traditional approaches that still outperform the transformer on smaller-scale datasets [41]. Therefore, the transformer architecture might not completely replace the old technologies (*e. g.* CNNs), at least in the current state.

3 MULTI-MODAL DENSE VIDEO CAPTIONING

Video captioning requires a video understanding model to produce a textual description of the video content (see relevant work on page 25). By default, the description consists of a single sentence which might not be enough for a seconds-long video clip. One way of addressing this problem would be to use multiple sentences that convey a coherent description which is called *paragraph* video captioning (see prior work on page 28). Another way of approaching it is to assume that a video consists of multiple events, *e. g.* a cooking video, in which a cook starts by making dough, then make the filling, and finally, put it in the oven. Therefore, a natural approach would be to, first, detect events temporally and, then, make a textual description for each detected event. This task is called *dense* video captioning.

Prior works in dense video captioning focused mainly on the visual modality alone and ignore potentially crucial cues from an audio track. For example, knocking on the door from the opposite side might be invisible on the visual track but processing the audio track might facilitate a better understanding of this scene. Besides audio, speech might also provide useful cues for a captioning model because sometimes a narrator reacts to a scene in a certain way or partially verbalizes the content of the events. At the same time, the transformer architecture, which was originally designed for translation, fits well into the video captioning framework as one may formulate the problem as a translation from video to text.

This chapter introduces two novel ways of how the transformer architecture could be generalized into the multi-modal setting to tackle the dense video captioning task more efficiently. In particular, Section 3.1 presents the related work on dense video captioning. In Section 3.2 a new multi-modal framework for dense video captioning is introduced which is followed by our newer development of a novel generalized transformer for bi-modal input in Section 3.3. Finally, the discussion regarding potential directions for future research is presented in Section 3.5.

3.1 Related work

Inspired by the dense *image* captioning [457], Krishna *et al.* [198] suggested exploring captioning videos in a similar way. In particular, they used Deep Action Proposals (DAPs) network [458] to predict the temporal event boundaries, and an LSTM to caption the video clip that was outlined by an event boundary. Besides the proposed architecture, the ActivityNet Captions dataset was released to the public which inspired many follow-up works in dense video captioning. In most of these works the design of an event proposal generation module was inspired by the advances in action proposal generation. Specifically, Wang *et al.* [459] made use of wider context in the Single-stream Temporal Action (SST) proposal generation network [460] and a dynamic gating mechanism to control the influence of the context. Meanwhile, Zhou *et al.* [30] put forward the transformer architecture with a variant of ProcNets [199] that generates event proposals. Duan *et al.* [461] proposed an approach to addressing the problem of expensive annotation that dense video captioning requires and studied the idea of weakly-supervision with a cycle-consistency loss [381] given a set of captions with temporal annotation.

The loss values have only a weak correlation with the resulting captioning metrics. For this reason, others explored the optimization of these non-differentiable metrics directly, with reinforcement learning (RL) objectives as it was initially outlined for image captioning [462]. This idea was adapted for dense video captioning by Li *et al.* [463] who also suggested relying on a variant of the Single Shot Detector (SSD) [464] for proposal generation. In the meantime, Xiong *et al.* [203] explored sentence and paragraph-level rewards to improve coherence in video captioning with an LSTM model and used Structured Segment Networks [465] as an event proposal module. The idea of multi-level rewards to improve story-telling was studied by Mun *et al.* [466] who combined the SST proposal generation network (as in [459]) with a Pointer Network [467] to filter generated proposals.

Only a few works have explored the use of other modalities for dense video captioning. In particular, Rahman *et al.* [468] generalized the weakly-supervised approach of Duan *et al.* (as outlined above) by adding the audio modality and relied on the Tucker decomposition to fuse audio and visual. Although weak supervision brings its benefits, the performance was substantially lower compared to supervised models. While Shi *et al.* [469] tackled captioning of cooking videos with additional

speech transcripts. The transcripts were encoded with BERT and fused with visual representations in an LSTM module, followed by another LSTM network that generates a caption. Although the proposed method reached strong performance, it was applied to the instructional (cooking) videos where transcription has a strong correlation with the captions (see the HowTo100M dataset). In this work, we propose two new approaches that address these issues.

3.2 Multi-modal transformer for dense video captioning (MDVC)

The framework consists of two major parts: the event localization module and the multi-modal dense video captioning module as outlined in Figure 3.1. We refer to the proposed framework as MDVC. Specifically, we will rely on *Bidirectional Single-stream Temporal Action proposal network* (Bi-SST) [459] to generate event proposals. The Bi-SST operates on 3D convolutional network features that are, then, encoded by a recurrent neural network. The outputs of the recurrent net are used to predict the most appropriate proposal *anchor* along with a confidence score. Once proposals have been selected for a video, each of them is captioned with a transformer-based model. We rely on multiple modalities to extract important information from video content: audio, speech in a form of subtitles, and vision. The outputs of individual feature transformers are concatenated and passed to the generator that predicts the next caption word. The captioning is done autoregressively, *i. e.* word-by-word.

Event localization module

The goal of the event localization module is to make a set of potential temporal regions for *important* events on a video. Here we rely on the Bi-directional Single Stream Temporal action proposal net (Bi-SST) [459] considering its strong performance. The input to Bi-SST is a sequence of features extracted by a 3D convnet features (C3D [470]). The features are extracted from non-overlapping stacks of 16 RGB frames with a stride of 64 frames. Principal Component Analysis (PCA) is used to reduce the dimensionality of each feature vector from 4096 to 500.

These extracted features are passed to a bi-directional LSTM [94] which performs two passes: forward and backward (reversed in time). During the forward pass, the LSTM holds the information about the current visual feature and all visual features from previous states (past). Using this accumulated information, the net-

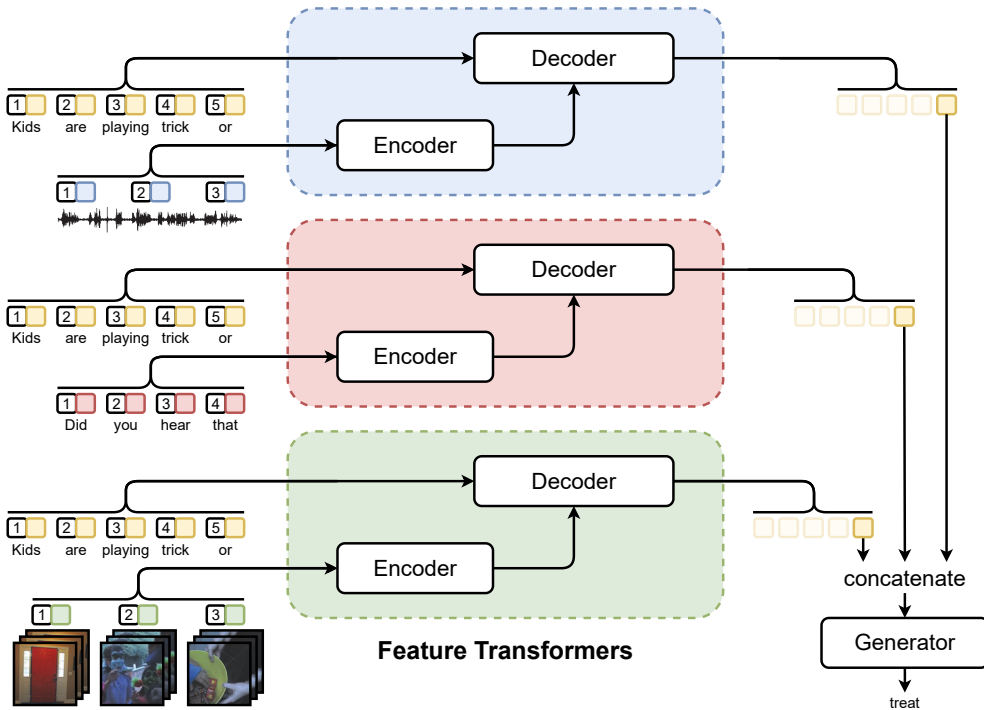


Figure 3.1 Overview of the Multi-modal Dense Video Captioning model (MDVC). It takes in features from audio (blue), speech (red), and visual (green) modalities that correspond to a certain temporal proposal. Each sequence of features is processed in a corresponding feature transformer as well as the previously generated caption words (yellow). The output of the transformers is concatenated and used in the generator to predict the next caption word.

work makes a prediction (confidence) of how relevant to each of the 128 pre-defined event anchors¹ at each temporal position of a feature t . During the forward pass, an LSTM encodes the input visual features step-by-step from the beginning of the video till the end. Since the model encoded only *past* information ($< t$) by a step t , the temporal ends of the anchor proposals are made to end at t . Similarly, during the backward pass, the LSTM iterates from the end of the video towards its beginning and holds a hidden state accumulating *future* cues ($> t$). Therefore, a second set of anchor-confidence pairs is formed but now all starting from step t since no earlier information ($< t$) has been processed. The final set of proposals, *i. e.* triplets (start, end, confidence), is picked according to the confidence score and manually selected threshold.

¹Similar the procedure for object detection [464, 471], the event anchors are pre-calculated with the K-Means algorithm which clusters ground truth temporal boundaries.

Captioning module

The task of the dense video captioning module is to produce a textual description of the clip content that is bounded by an event proposal. Prior works mainly rely on visual-only modality to perform captioning. We argue that other modalities such as audio and speech transcripts may provide useful cues to a model which improves captioning performance. Considering this hypothesis and the previous success of the transformer architecture, we propose to tackle captioning with a stack of transformer-based encode-decoder architectures that jointly learn to extract relevant information from multi-modal feature sequences and use it to predict the next caption word, see Figure 3.1 for details.

In this work, we rely on multi-modal input features that are extracted as follows. *VGGish* [17] is used to extract a 128- d audio features from approximately 1 second of audio. The speech transcripts (subtitles) have been obtained using automatic speech recognition (ASR) tool available on YouTube API². The subtitle words are embedded into 512- d space with a trainable embedding as described in Section 2.3 on page 40. The visual frames are processed along with optical flow frames in a *Two-stream Inflated 3D convnet* (I3D) [11] which produces 1024- d feature vectors for approximately 1 second of the original visual stream.

Once the sequences of features are obtained for each modality, they are passed to the corresponding feature transformer along with the token embeddings from previously generated captioning words. The architecture of the feature transformer is similar to the vanilla encoder-decoder transformer architecture which was outlined in Section 2.3 except for the generator. The latent dimension of the feature transformer, including the dimension of the caption token embedding table resembles the dimensionality of the input features, *i. e.* 128 for audio, 512 for speech, and 1024 for visual modalities. To generate the next caption word, the generator uses the last element from each sequence and concatenates these elements along the latent dimension, and passes it to the fully-connected layer followed by a softmax which will output the probability distribution over the training vocabulary ($\approx 10k$ words). The captioning model is trained in a similar way to the translating transformer with a cross-entropy loss with label smoothing as described in Section 2.3.

²<https://developers.google.com/youtube/v3/docs/captions>

3.3 Better use of audio-visual cues with bi-modal transformer (BMT)

Although MDVC (Section 3.2) yields strong results compared to prior works, it has a few drawbacks. In particular, despite being straightforward, some parts of the captioning architecture are redundant, *e. g.* each feature transformer has to learn an individual text embedding module in the decoder. Another issue that could potentially impair the model performance is the lack of interaction between multiple modalities. In fact, MDVC relies on the late fusion to make a decision on the next caption word. Even though this could be beneficial for the sake of modularity, *e. g.* by reusing the feature transformers separately, fusing the features at the very end might prevent the model from learning low-level multi-modal interactions. Finally, the proposal generation hinges on visual information alone, yet other modalities might provide useful cues for the start and end of the events.

Therefore, we present a novel approach for dense video captioning that addresses these issues. We call it BMT which stands for “Bi-modal Transformer”. The proposed model outperforms MDVC by a substantial margin while being smaller in size and using only audio and visual modalities. In addition, the multi-modal features are fused early which allows the model to effectively encode multi-modal information and use it not only for captioning but also for proposal generation.

Framework overview

The *Bi-modal Transformer* (BMT) is a general-purpose architecture for sequential bi-modal inputs. We rely on BMT to perform both event proposal generation and captioning. More specifically, the BMT is an encoder-decoder transformer which, during captioning, inputs a (trimmed) sequence of audio features (extracted by *VG-Gish* [17] in our case) and a (trimmed) sequence of visual features (*I3D* [11]) as it is shown on Figure 3.2.

The sequences of audio and visual features are passed through a stack of L bi-modal encoder layers. Compared to the original transformer encoder, it has a novel block which we refer to as *Bi-modal Multi-headed Attention*. This block allows modelling interactions between modalities early on. The outputs of the bi-modal encoder are passed to the bi-modal attention blocks in bi-modal decoder layers. These outputs are used as the context (*keys* and *values*) for the encoded sequence of caption word embeddings (with *GloVe* [6]) from the previously generated caption words. The

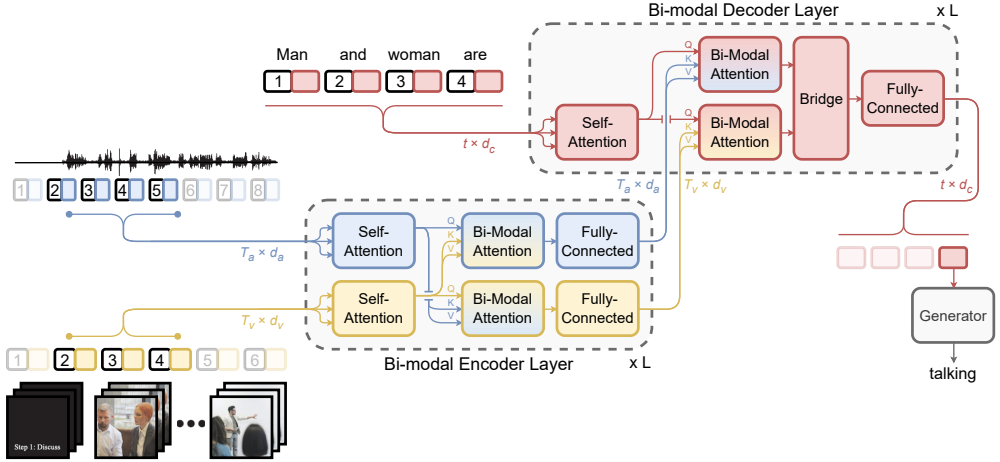


Figure 3.2 Overview of Bi-modal Transformer (BMT). It inputs features from audio (blue) and visual (yellow) modalities that correspond to a certain temporal proposal, the rest of the streams is discarded. The L -layer bi-modal encoder (bottom) inputs audio and visual features and outputs bi-modal representations. The encoder representations are used as a context in the L -layer bi-modal decoder (top) that also takes in the previously generated caption words (red). The output of the bi-modal decoder is passed to the generator which predicts the next caption word. Layer normalization layers and residual connections are omitted for clarity.

outputs of the bi-modal attention blocks inside of the decoder layer are fused in the *Bridge* module. The last bi-modal decoder layer outputs representations that are used in *Generator* which produces the next caption word.

As shown in Figure 3.3, to generate proposals that were used to trim the input features, we rely on the bi-modal encoder and the novel *multi-headed proposal generation module* that relies on the multi-modal output of the encoder. The proposal generator uses multiple heads with unique temporal perceptive fields to predict the offsets to a pre-defined set of event anchors inspired by the success of *YOLOv3* [20] in object detection. The predicted proposals are finally sorted for confidence.

The proposed model is trained in two stages. First, we train a captioning module on ground truth event proposals. Second, we reuse the pre-trained blocks of the captioning model to initialize parts of the proposal generation model which is trained next. We found that doing this in the reversed order as described is beneficial for the final performance. Therefore, in this section, we begin with the captioning module and continue with the event localization module.

Captioning module

As shown in Figure 3.2, the *bi-modal encoder* inputs sequences of audio (A) and visual (V) features that were trimmed according to proposal boundaries and outputs vision-guided audio features A_v and audio-guided visual features V_a . The *bi-modal decoder* inputs these features along with the previously generated caption words (c_1, c_2, \dots, c_t) and makes representation that is, finally, used in the *generator* to produce the next caption word (c_{t+1}). We omit the proposal index from our notation for clarity.

Bi-modal encoder The encoder consists of L layers and inputs the trimmed audio ($A \in \mathbb{R}^{T_a \times d_a}$) and visual ($V \in \mathbb{R}^{T_v \times d_v}$) features. Compared to the original transformer’s encoder layer [4], the bi-modal encoder has not two but three sub-layers: multi-headed self-attention, *multi-headed bi-modal attention* (new), and position-wise feed-forward network. The self-attention block and feed-forward network are similar to those of the vanilla transformer as defined in Section 2.3. Compared to the self-attention, the multi-head bi-modal attention for 2D inputs x, y is defined as

$$\text{SelfAtt}(x) = \text{MultiHeadAtt}(x, x, x), \quad (3.1)$$

$$\text{BiModalAtt}(x, y) = \text{MultiHeadAtt}(x, y, y). \quad (3.2)$$

Then, given $A_{\text{fc}}^0 = A$ and $V_{\text{fc}}^0 = V$, the l^{th} bi-modal encoder layer is defined by

$$A_{\text{self}}^l = \text{SelfAtt}(A_{\text{fc}}^{l-1}), \quad V_{\text{self}}^l = \text{SelfAtt}(V_{\text{fc}}^{l-1}), \quad (3.3)$$

$$A_{\text{bm}}^l = \text{BiModalAtt}(A_{\text{self}}^l, V_{\text{self}}^l), \quad V_{\text{bm}}^l = \text{BiModalAtt}(V_{\text{self}}^l, A_{\text{self}}^l), \quad (3.4)$$

$$A_{\text{fc}}^l = \text{FeedForwardNet}(A_{\text{bm}}^l), \quad V_{\text{fc}}^l = \text{FeedForwardNet}(V_{\text{bm}}^l), \quad (3.5)$$

All blocks have unique trainable parameters, *i. e.* not shared. The last bi-modal encoder layer outputs vision-guided audio features $A_{\text{fc}}^L = A_v$ and audio-guided visual features $V_{\text{fc}}^L = V_a$. Both sets of features are passed to the bi-modal decoder.

Bi-modal decoder The decoder consists of L layers and inputs token embedding from previously generated caption words $C_t = (c_1, c_2, \dots, c_t) \in \mathbb{R}^{t \times d_c}$ along with vision-guided audio features $A_v \in \mathbb{R}^{T_a \times d_a}$ and audio-guided visual features $V_a \in \mathbb{R}^{T_v \times d_v}$. Compared to the vanilla transformer which has three sub-layers in a decoder layer, the bi-modal decoder consists of four sub-layers: multi-headed self-attention,

bi-modal multi-headed attention (new), *bridge connection* (new), and position-wise feed-forward net. Except for the bridge connection sub-layer, other sub-layers are defined in the bi-modal encoder layer. The bridge connection is defined as follows:

$$\text{Bridge}(x, y) = \text{ReLU}([x, y]W^b + b), \quad (3.6)$$

where x, y can be arbitrary 2D-inputs ($\in \mathbb{R}^{t \times d_c}$), $[\cdot, \cdot]$ is the concatenation operation across the second dimension, and $W^b \in \mathbb{R}^{2d_c \times d_c}$ and b are trainable weights. Given caption embeddings $S_{\text{fc}}^0 = C_t$ and outputs from the bi-modal encoder V_a, A_v , the l^{th} bi-modal decoder layer is defined as:

$$S_{\text{self}}^l = \text{SelfAtt}(S_{\text{fc}}^{l-1}), \quad (3.7)$$

$$S_{\mathcal{A}}^l = \text{BiModalAtt}(S_{\text{self}}^l, A_v), \quad S_{\mathcal{V}}^l = \text{BiModalAtt}(S_{\text{self}}^l, V_a), \quad (3.8)$$

$$S_{\text{bm}}^l = \text{Bridge}(S_{\mathcal{A}}^l, S_{\mathcal{V}}^l), \quad (3.9)$$

$$S_{\text{fc}}^l = \text{FeedForwardNet}(S_{\text{bm}}^l). \quad (3.10)$$

The outputs of the last layer of the bi-modal decoder (S_{fc}^L) are used in the generator that predicts the next caption word.

Generator The generator inputs caption features S_{fc}^L from the bi-modal decoder. The architecture of the generator is similar to the original transformer. In particular, is a fully-connected layer followed by the softmax which outputs the probabilities for the next caption word from the vocabulary.

Event proposal generation module

The goal of the event proposal generator is to make a set of temporal proposals that could potentially outline events in a video. In this work, we introduce a novel proposal generator which contains two sets of proposal heads that make predictions for the two bi-modal encoder’s output streams as it is shown in Figure 3.3. Each proposal head makes offset predictions for each 1D *temporal anchor* as it is done in object detection but in 2D. The head makes such predictions at every position. Finally, the predictions from every position, anchor, head, and modality are gathered and sorted by confidence.

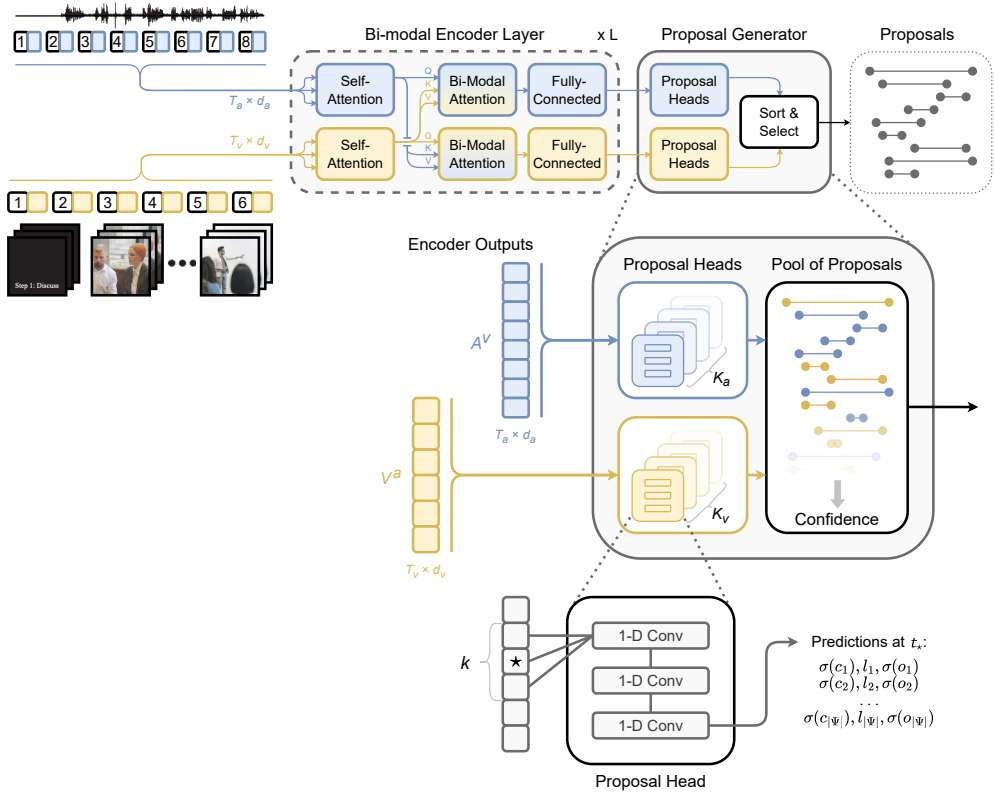


Figure 3.3 Overview of the Bi-modal Multi-headed Event Proposal Generator. It takes in features from audio (blue) and visual (yellow) modalities from a full video. The L -layer bi-modal encoder (top) inputs audio and visual features and outputs bi-modal representations. This two-stream output is used in the two stacks of proposal generation heads (middle): K_v for the visual and K_a for the audio stream. Each proposal head (bottom) has a unique temporal receptive field k . A head makes predictions for each of $|\Psi|$ anchors at every position (shown at the position t_*).

Proposal generation head As shown in Figure 3.3 (bottom), a proposal head inputs a sequence of features that come from the bi-modal encoder ($A_v \in \mathbb{R}^{T_a \times d_a}$ and $V_a \in \mathbb{R}^{T_v \times d_v}$). The head makes predictions at each position on the interval $[1, T_{a/v}]$ and for each temporal anchor. The design of the head is partially inspired by the detection layer of the *You Look Only Once v3* (YOLOv3) object detector [20] and consist of three 1D convolutional layers. The first layer has the kernel size of k features which is unique to each proposal head and varies greatly to accommodate the diversity of event durations across the dataset. The second and third conv layers have a kernel size of 1. The ReLU activation and dropout are used for each conv

layer. For each temporal anchor, a head outputs a triplet (c, l, o) . These values can be transformed into the centre of the temporal bounding box, its temporal length, and prediction confidence as follows:

$$\text{center} = p + \sigma(c); \quad \text{length} = \text{anchor} \cdot \exp(l); \quad \text{confidence} = \sigma(o), \quad (3.11)$$

where σ is a sigmoid function. We omit the anchor index for clarity.

Bi-modal multi-headed proposal generator The proposal generator makes predictions from two sets of proposal heads. We use K_a for audio and K_v for visual streams. Therefore, in total, the proposal generator outputs $T_a \cdot K_a \cdot |\Psi_a| + T_v \cdot K_v \cdot |\Psi_v|$ prediction triplets. During inference, the proposal list is sorted by the confidence score.

Making temporal proposals and picking kernel sizes The sets of anchors for audio and visual streams are picked with a K-Means algorithm which clusters ground truth temporal event annotations. The numbers of clusters for audio and visual streams are picked to balance the number of predicted proposals per stream, such that $T_a \cdot |\Psi_a| = T_v \cdot |\Psi_v|$, assuming that the numbers of proposal heads per stream are equal ($K_a = K_v$). The kernel size values (k 's) for proposal heads are also determined by K-Means clustering to maximize the chance of having a higher overlap between the kernel's perceptive field and a potential event duration. For implementation details, a reader is referred to Publication II.

3.4 Experiments and results

In this section, we present the experimentation setting and results for MDVC (Section 3.2) and BMT (Section 3.3) compared to the prior state-of-the-art approaches.

Dataset ActivityNet Captions [198] is used as a dataset for experimentations. This dataset consists of ≈ 20 k YouTube videos with human-generated dense temporal annotations and corresponding captions. Videos are 2 minutes long on average and annotated with approximately 4 localized captions that have around 14 words. We follow the "official" split of the dataset, *i. e.* 2:1:1 for training, validation, and testing. The validation videos are annotated twice. For MDVC, the speech transcripts are obtained from the YouTube ASR system and we found that around a third of all

videos have at least one speech segment. At the time of development (early 2019), we could obtain approximately 90 % of the $\approx 20k$ videos from YouTube, the other 10 % of videos are no longer available to the public.

Metrics To put the performance of our model into the appropriate context, we follow the same evaluation metrics as in prior works. Specifically, we use BLEU@3,4 [472] and METEOR [473] machine translation metrics. The METEOR is used as the main metric since it is considered to correlate the most with human judgement. The dense video captioning performance is evaluated as follows. If a generated proposal overlaps with the ground truth more than a threshold value of *temporal Intersection-over-Union* (tIoU), the captioning metric is calculated and, otherwise, zero is recorded. The metric for a video is calculated by averaging metrics for each tIoU threshold in the list ([0.3, 0.5, 0.7, 0.9]). Then, metrics are calculated across all videos in the dataset and averaged. Since the validation dataset has two sets of annotations, we take an average of the two metrics. The top 100 proposals are picked according to confidence as the preferred candidates. For more details on the evaluation procedure, a reader is referred to Publications I and II.

Results: dense video captioning In this chapter, we introduced two novel frameworks for dense video captioning called MDVC and BMT. In Table 3.1, we show the comparison to prior work across two settings: captioning with ground truth (GT) and generated (learned) proposals. As it is shown in Table 3.1, MDVC and BMT outperform all prior non-RL methods on METEOR when captioning learned proposals³. Moreover, BMT reaches the highest BLEU@3,4 scores among all methods in the learned proposal setting. When it comes to the performance on ground truth (GT) proposals, our models perform strongly and outperform most of the methods except for [30] on BLEU@3,4 while being on par in terms of METEOR. We also highlight that our models were trained on 90 % of videos that were available to prior works because the other 10 % are no longer available on YouTube. When comparing MDVC against its descendant BMT, we observed several benefits of BMT. In particular, although BMT has a smaller capacity (59M vs 179M parameters) and does

³The evaluation of a video captioning model is challenging and METEOR (or BLEU) is only a proxy for how good a caption is. Therefore, direct optimization of METEOR using a reinforcement learning (RL) objective might not necessarily result in a better caption. For instance, the method proposed by Li *et al.* [463] noticeably boosts METEOR when an RL objective is used but other metrics remain intact.

	With RL	GT Proposals			Learned Proposals		
		B@3↑	B@4↑	M↑	B@3↑	B@4↑	M↑
Li <i>et al.</i> [463]	yes	4.55	1.62	10.33	2.27	0.73	6.93
Xiong <i>et al.</i> [203]	yes	–	–	–	2.84	1.24	7.08
Mun <i>et al.</i> [466]	yes	4.41	1.28	13.07	2.94	0.93	8.82
Krishna <i>et al.</i> [198]	no	4.09	1.60	8.88	1.90	0.71	5.69
Li <i>et al.</i> [463]	no	4.51	1.71	9.31	2.05	0.74	6.14
Zhou <i>et al.</i> [30]	no	5.76	2.71	11.16	2.91	1.44	6.91
Wang <i>et al.</i> [459]	no	–	–	10.89	2.27	1.13	6.10
Mun <i>et al.</i> [466]	no	–	–	–	–	–	6.92
Rahman <i>et al.</i> [468]*	no	3.04	1.46	7.23	1.85	0.90	4.93
MDVC (Ours)*	no	4.52	1.98	11.07	2.53	1.01	7.46
BMT (Ours)*	no	4.63	1.99	10.90	3.84	1.88	8.44

Table 3.1 Dense video captioning results of the proposed MDVC and BMT compared to prior work across two settings: captioning ground truth (GT) and predicted proposals (learned). The performance is reported on both validation sets of ActivityNet Captions. Metrics are BLEU@3,4 (B@3,4) and METEOR (M). Additionally, we report the methods that rely on a reinforcement learning (with RL) objective which boosts METEOR. (*) — smaller training dataset due to the missing videos (see § Dataset for details).

not rely on speech modality, it performs on par when ground truth proposals are captioned. Also, the newly introduced audio-visual proposal generator allows for significantly stronger performance in the setting with generated proposals compared to MDVC. We invite a reader to inspect qualitative results in Publications I and II.

Results: event proposal generation Table 3.2 shows the results of the comparison between the novel bi-modal multi-headed event proposal module and prior work methods. Since MDVC relies on the proposal generator of Wang *et al.* [459], we omit it from the table. According to the results, our method significantly outperforms other approaches, despite being trained on fewer videos.

Ablation: effect of other modalities In this experiment (Table 3.3), we show the importance of multi-modal cues for dense video captioning by comparing the performance of ablated models that rely on a subset of selected modalities. Accord-

	F1↑
Xiong <i>et al.</i> [203]	33.01
Wang <i>et al.</i> [459]	50.40
Zhou <i>et al.</i> [30]	53.31
Mun <i>et al.</i> [466]	56.56
BMT (Ours)*	60.27

Table 3.2 Event proposal generation results of the bi-modal multi-headed event proposal generator module compared with the performance of proposal generators in prior work in dense video captioning. The results are reported on validation sets of ActivityNet-Captions. The metric is the F1-score which is the harmonic mean of the precision and recall. (*) — smaller training dataset due to the missing videos (see § Dataset for details).

ing to the results, adding more modalities benefits a dense video captioning model. Notably, audio-only models perform significantly worse compared to vision-only models. Since MDVC relies on a visual-only proposal generation module, we cannot evaluate the influence of the multi-modal setup on the learned proposal settings. However, the positive effect of using audio-visual cues is clearly visible for BMT.

3.5 Discussion

Related work: new developments Publications I and II were published in 2020 and the field of dense video captioning has expanded in multiple directions. In particular, Wang *et al.* [474] explored the semantic and temporal relationships between event proposals to improve the coherence of captioning further. While Suin and Rajagopalan [475] focused on reducing the computation cost of a dense video captioning system. The idea of not relying on temporal annotations (weak supervision) when training a dense video captioning model ([461, 468]) was further developed by Chen *et al.* [476] who suggested a more efficient method of the captioner-localizer interaction. In the meantime, Deng *et al.* [477] also proposed to switch from the conventional “localize-then-describe” into making a paragraph of sentences (draft) for the whole video and, then, ground temporally each sentence, followed by a refinement module that relies on an RL objective. Wang *et al.* [208] and Choi *et al.* [478] explored the query-based approach inspired by an object detector DETR [3] which allowed training event localization and captioning in a single-stage model.

Future research: better datasets It appears that most of the prior arts focused on designing a novel architecture, yet it is not the bottleneck place at the moment. Although the field of dense video captioning has experienced many great developments

Input Modality	Ground Truth Proposals		Learned Proposals	
	BLEU@4↑	METEOR↑	BLEU@4↑	METEOR↑
MDVC				
Audio	1.13	8.79	–	–
Visual	1.77	10.58	1.07	7.31
Audio + Visual	1.90	10.83	–	–
Audio + Visual + Speech	1.98	11.09	–	–
BMT				
Audio	1.14	8.81	1.15	6.98
Visual	1.66	10.29	1.30	7.47
Audio + Visual	1.99	10.90	1.88	8.44

Table 3.3 The effect of other modalities on performance on the dense video captioning task. The results are reported on ActivityNet Captions validation sets and across two proposed models.

across the years due to the novel architectural elements, there are other directions for future research to be explored.

First and foremost is the absence of a large-scale video dataset with temporal and textual annotations which one could use to train (or fine-tune) a model and exploit multi-modal interactions. Previous approaches rely on either ActivityNet Captions [198], which is an open-domain dataset but small in size (10k training videos), or YouCookII [199], which is a narrow-domain smaller-scale cooking video dataset (1.4k training videos). For instructional videos (including cooking videos), one could explore the potential of the scale of the HowTo100M dataset [302] and the ways of combating the temporal and semantic noisiness of the annotations.

One of the interesting directions to solve the problem of expensive data annotation was weak supervision [461, 468, 476], *i. e.* training a dense video captioning system on a video captioning dataset without temporal annotations. Nonetheless, it appears that the future of (dense) video captioning, as well as many other tasks in video content understanding, is to become one of the tasks of large-scale pre-training of a foundation model as discussed in Section 2.2.8. Therefore, it might be more beneficial to build a large-scale high-quality multi-modal video dataset for

general-purpose pre-training rather than investing efforts into developing a downstream task-specific video dataset to push the field forward.

Future research: better evaluation Currently, the evaluation of dense video captioning, and video captioning in general, is far from perfect. Although METEOR and BLEU@K are the best candidates at this stage, they have a weak correlation with human judgement. This could be another application of a foundation model that was pre-trained contrastively on video-text pairs. Similar to the CLIP-based image captioning metric, one could rely on the video-text similarity of such a model to develop more reliable metrics. Another benefit of making a better suite of evaluation metrics is associated with the availability of the RL-based approaches [203, 463, 466, 477] which might significantly improve captioning performance.

4 VISUALLY-GUIDED SOUND GENERATION FOR OPEN-DOMAIN VIDEOS

Generation of relevant audio for a video clip requires a video understanding model to relate visual and audio cues. Solving the visually-guided sound generation could potentially help the sound (foley¹) designers to create suitable sounds for the visual scene which often requires a substantial amount of time spent searching relevant databases of sounds or manual work.

Prior arts in visually-guided audio generation focused primarily on generating sounds for specific domains, such as musical instruments or a handful of classes. Yet, the generated samples were short, of low quality, and required several GPU minutes to sample a second of the target audio. Moreover, one needed to train one model per class, which quickly becomes infeasible in an attempt to cover “in the wild” videos as it might require training dozens or even hundreds of such models. In this work, we addressed these issues by relying on a two-stage training approach. In particular, during the first stage, the model effectively compresses the training audio dataset into a set of representative vectors (a spectrogram codebook). These vectors can be sampled given a video cue during the second stage. In addition, we introduced a novel suite of evaluation metrics for conditional spectrogram generation.

In this chapter, we present a novel framework for a visually-guided sound generation that supports many classes in a single model. Related work is outlined next. In Section 4.2, the design of the architecture and the training process is described. The experimentation setup as well as the results are presented in Section 4.4. Section 4.5 provides a discussion regarding the follow-up works and future research directions.

¹Sound effects in filmmaking that are manually created.

4.1 Related work

Generation of instrument music with visual cues Most of the prior arts in visually-guided sound generation focused on the generation of instrumental music. The work of Owens *et al.* [479] sparked the interest towards visually indicated audio by presenting a novel dataset of hitting and scratching sounds emitted by a drumstick (“Greatest Hits”) as well as a CNN followed by an LSTM as a generative model. Chen *et al.* [480] explored the generation of single instrument audio given an image and vice-versa. To achieve this, Chen *et al.* relied on two generative adversarial nets (GANs) as well as the URMP dataset [481]. To improve the performance of the GAN-based approach, Hao *et al.* [482] used the cross-modal cycle-consistency [483] while Tan *et al.* [484] embellished it with a transformer’s self-attention. Prediction of Midi given a top view on hands playing the piano was investigated by Su *et al.* [485]. Kurmi *et al.* [486] explored the joint generation of short audio and visual tracks simultaneously. Although these works show promising results in generating instrument music, the generated samples are quite short (~ 1 second), span a narrow domain and represent staged scenarios.

In contrast, we propose a model that generates 10-second audio clips and supports the generation of relevant sounds to open-domain videos.

Generation of open-domain sounds based on visual cues Although visually-guided audio generation is a relatively new field, a few interesting approaches have been developed over the last few years. In particular, Chen *et al.* [487] relied on a small subset of AudioSet [488] and suggested learning a residual to a class-representative spectrogram (average). To improve the quality (fidelity) and relevance of generated audio, Zhou *et al.* [483] focused on training one model per data class using a hierarchical RNN to generate a waveform. In addition to the novel architecture, Zhou *et al.* introduced a new 10-class dataset (VEGAS) which is based on AudioSet. While Chen *et al.* [16] explored the influence of *invisible* background audio on training and suggested using bottlenecked ground truth audio representation as well as visual input to improve training dynamics of the LSTM-GAN-based sound generation model. Besides the new approach for training, they introduced an improved version of the VEGAS dataset, called VAS. Although the results were promising, this setting quickly becomes impractical once a dataset contains tens or, even, hundreds of data classes because one is required to train one model for each class. Another issue is the

sampling speed which takes minutes to generate one second of the audio.

In this work, we introduce a novel approach for the visually-guided sound generation that supports many data classes in a single model, and the generation requires less time than it takes to play the generated audio on a single GPU. Furthermore, our model significantly outperforms the state-of-the-art in terms of sample quality while being on par in terms of relevance to the visual input.

Metrics for automatic evaluation of audio generation Evaluation of machine-generated content is a challenging task. Inspired by the image generation metrics that are based on the dataset distribution, a variant of FID [489] was adapted to evaluate a music enhancement model in [490] and a text-to-speech model in [491]. The proposed methodology, however, operates on small windows of one second and might miss long-term coherence. A metric measuring similarity of two speech segments was introduced in a form of a perceptual loss by Manocha *et al.* [492] who suggested to collect human opinions on the similarity and train a classifier on such data. The main drawback of this approach is the significant budget requirement that one may need to overcome to collect a training set of human judgements for large-scale open-domain datasets.

In this work, we designed a suite of metrics to evaluate not only the quality (fidelity) but also the relevance of open-domain long (10-second) samples.

4.2 Codebook-based conditional sampling (Spectrogram VQGAN)

Our goal in this work is to design and train a model which is capable of sampling high-fidelity audio given visual cues. Moreover, the model is expected to be capable of supporting open-domain visual inputs and producing seconds-long samples in a matter of seconds. Instead of generating audio samples on the spectrogram-pixel level or waveform values, we suggest “factorizing” the problem into two sub-problems that can intuitively be described as follows. First, a set of “building spectrogram blocks” is trained as an autoencoder with a codebook. Second, an autoregressive model is trained to sample these building blocks given a visual prompt (a condition). In this section, we define both stages in detail and outline the behaviour of the model during inference. The overview of the training for both stages is illustrated in Figure 4.1.

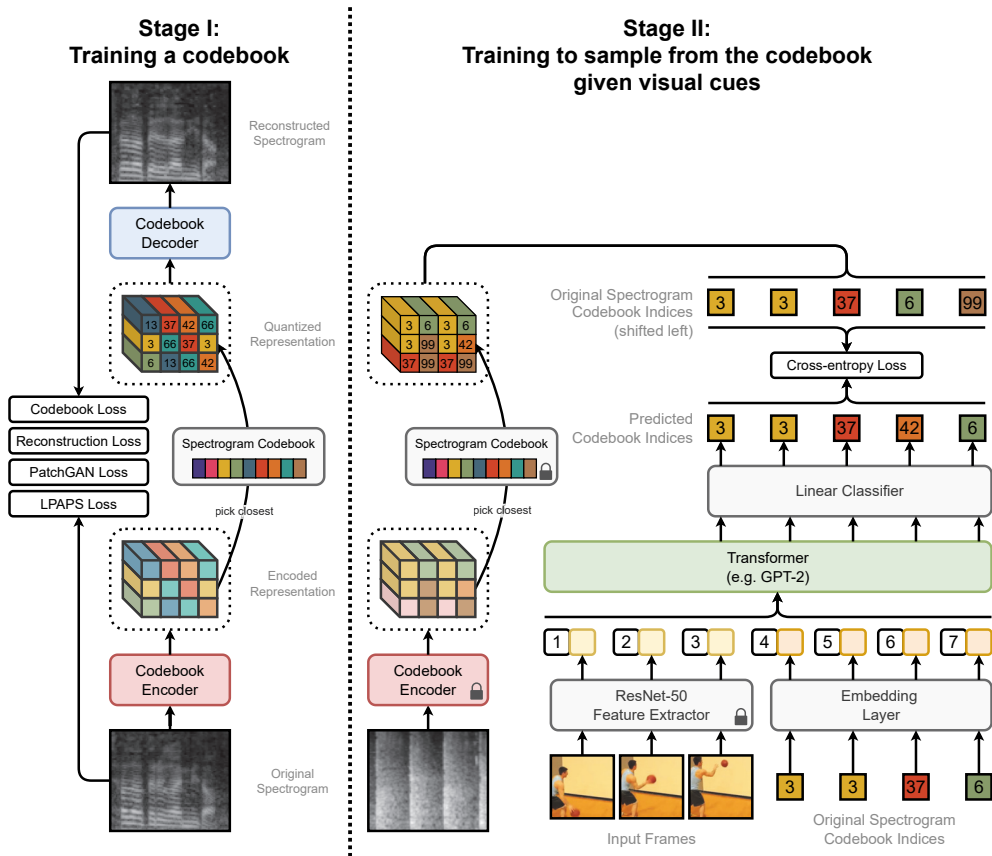


Figure 4.1 Training pipeline of the visually-guided autoregressive codebook-based conditional sampler. During **Stage I**, an autoencoder with a discrete bottleneck is trained on spectrograms. In particular, an input log mel-spectrogram is encoded into a small-scale representation by a codebook encoder (e.g. a CNN). Next, this representation is quantized by picking the closest codebook entry for each element. Then, the quantized representation is decoded via the codebook decoder to reconstruct the spectrogram. The autoencoder is trained to ensure the similarity between the input and output spectrograms. During **Stage II**, a sampler (e.g. transformer) is trained to sample codebook codes in an autoregressive manner, given a sequence of video frames and previously generated codebook indices. The ground truth for the training of the transformer is provided by the codebook indices of the quantized representation of the corresponding ground truth audio spectrogram.

Stage I: Training a spectrogram codebook

The overview of Stage I is outlined in Figure 4.1 (left). Considering that the transformer is the state-of-the-art architecture for autoregressive modelling, we rely on it to perform sampling during Stage II. However, considering the quadratic complexity of the transformer with respect to the input sequence length, sampling raw spectrogram pixels or waveform values is impractical. Therefore, drawing on success of VQGAN [39] in high-resolution image synthesis, we encode an input spectrogram into a small-scale vector-quantized representation which allows a transformer to model long-term dependencies during Stage II and keeps the memory footprint manageable.

Spectrogram Vector-Quantized Variational Autoencoder (Spectrogram VQVAE)

The goal of Spectrogram VQVAE is to minimize the reconstruction error between an input spectrogram and its reconstruction from a small-scale quantized representation. For efficiency, we operate on log mel-spectrograms. Therefore, given a ~ 10 -second frequency-time spectrogram $x \in \mathbb{R}^{F \times T}$ (e. g. 80×848) the codebook encoder E produces a small resolution representation $\hat{z} = E(x) \in \mathbb{R}^{F' \times T' \times n_z}$ (e. g. $5 \times 53 \times 1024$). Then, each element of \hat{z} is replaced with the closest element from the codebook $\mathcal{Z} = \{z_k\}_{k=1}^K \in \mathbb{R}^{K \times n_z}$ with K codes (e. g. $K = 1024$). This quantization step \mathbf{q} is defined as

$$\mathbf{q}(\hat{z}) := \left(\arg \min_{z_k \in \mathcal{Z}} \|\hat{z}_{ft} - z_k\| \quad \text{for all } (f, t) \text{ in } (F' \times T') \right), \quad (4.1)$$

yielding the quantized representation $z_{\mathbf{q}} = \mathbf{q}(\hat{z}) \in \mathbb{R}^{F' \times T' \times n_z}$. Finally, $z_{\mathbf{q}}$ is decoded by the codebook decoder D into a reconstructed spectrogram: $\hat{x} = D(z_{\mathbf{q}})$. Notice that the quantization step is not differentiable. To work around it, a straight-through estimator is employed. The training objective of VQVAE is defined as follows

$$\mathcal{L}_{\text{reconstruction}} = \|x - \hat{x}\| \quad (4.2)$$

$$\mathcal{L}_{\text{codebook}} = \|\hat{z} - \text{stop}[z_{\mathbf{q}}]\|_2^2 + \beta \|\text{stop}[\hat{z}] - z_{\mathbf{q}}\|_2^2 \quad (4.3)$$

$$\mathcal{L}_{\text{VQVAE}} = \mathcal{L}_{\text{reconstruction}} + \mathcal{L}_{\text{codebook}} \quad (4.4)$$

where *stop* is an operation that does not affect the forward pass but stops the gradient from propagating during the backward pass and β is a hyper-parameter (e. g. 0.25).

Spectrogram VQGAN and spectrogram-based perceptual loss One of the limitations of a VQVAE is the large bottleneck size ($F' \times T'$) which makes the generation of longer sequences difficult for a transformer. As it was shown in [39], reducing the bottleneck resolution, *e. g.* to 1/16, yields blurred reconstructions. To reduce the resolution of the bottleneck, [39] suggested adding two losses to VQVAE, which are adversarial (PatchGAN) [493] and perceptual (LPIPS) [494] losses. Since LPIPS was originally designed for image generation and relies on a pre-trained ImageNet classifier (VGG-16 [24]), we had to adapt it for spectrogram inputs. First, we explored available models for spectrogram-based classification in the literature. We found that the closest architecture to VGG-16 for spectrogram classification is VGGish [17] (equivalent to the capacity of VGG-9). However, the short time span (less than 1 second) and, thus, poor temporal modelling might prevent distinguishing fake and real spectrograms, which is essential for our purpose as our model generates 10-second samples. To this end, we pre-train a variant of VGG-16 on a large-scale open-domain dataset (VGGSound [495]). We call this network VGGish-*ish* and use it as a backbone for the perceptual loss (LPAPS²). With these two losses, the total training loss for Spectrogram VQGAN is defined by:

$$\mathcal{L}_{\text{PatchGAN}} = \log D(x) + \log (1 - D(\hat{x})) \quad (4.5)$$

$$\mathcal{L}_{\text{LPAPS}} = \sum_s \frac{1}{F^s T^s} \|\hat{x}^s - x^s\|_2^2 \quad (4.6)$$

$$\mathcal{L}_{\text{SpecVQGAN}} = \mathcal{L}_{\text{VQVAE}} + \mathcal{L}_{\text{PatchGAN}} + \mathcal{L}_{\text{LPAPS}}, \quad (4.7)$$

where D is the discriminator network that is applied to patches of spectrograms instead of whole spectrograms, as it is done for images in [493], and $\hat{x}^s, x^s \in \mathbb{R}^{F^s \times T^s \times C^s}$ are fake and real feature maps obtained at the s^{th} scale of VGGish-*ish*.

Stage II: Conditional autoregressive spectrogram sampler

Once the Stage I model was trained to reliably reconstruct an input spectrogram, we could proceed with Stage II. The goal behind Stage II is to train a model to sample indices to the codebook given a set of encoded visual cues. Provided that the codebook decoder could reliably reconstruct a spectrogram from a quantized representation, it is now possible to model spectrogram synthesis on the representation level instead

²Unlike LPIPS [494], we do not fine-tune it on human perceptual similarity judgements.

of raw spectrogram pixels (or waveform values), which also allows us to rely on a transformer as the input sequence is significantly shorter now. The ground truth for training is formed by the codebook encoder which provides a quantized representation of the spectrogram that corresponds to the stream of visual cues that prime the sampling. The training pipeline for Stage II is depicted in Figure 4.1 (right).

Architecture Inspired by success in autoregressive image generation [39, 496], our sampler is a typical encoder-only transformer that is similar to GPT-2 [8] ($L = 24$ layers, $H = 12$ heads, $d = 1024$ hidden units, ~ 310 M trainable parameters). The transformer inputs sequences of embedded tokens from video frames and codebook indices. The visual tokens are extracted by a frame-wise feature extractor (e.g. ResNet [453]) and form a sequence $\mathcal{F} = \{f_i\}_{i=1}^N \in \mathbb{R}^{N \times d_r}$ which is embedded into the d -dimensional space by applying a linear projection layer. The codebook indices are embedded in the same way as the text tokens (see page 40) forming a sequence of token embeddings $s_{\leq t} = (s_1, s_2, \dots, s_t)$ where $t = F' \cdot T' - 1$ during training. The transformer inputs the sequence of frame-wise features (\mathcal{F}) concatenated with the embedded codebook indices ($s_{\leq t}$). Similar to the training procedure for a transformer (see page 45), during training, ground truth codebook tokens are used as previously generated tokens (aka. teacher forcing) and masking is applied to prevent the attention mechanism from peeking at the next tokens. Also, we highlight the importance of following the column-major order of unflattening the 2D sequence of ground truth tokens, as shown in Figure 4.1 (right), instead of the row-major (or raster) order.

Inference: Generating new audio given visual cues

Once the Spectrogram VQGAN and the codebook-based sampler are trained, one may generate new audio that is relevant to the content of an RGB stream. As it is shown in Figure 4.2, the model samples indices to the codebook in an autoregressive manner given a sequence of visual features and embeddings of previously generated codebook indices. The process begins with only the visual sequence as the input that is used to predict the first index. Then, the first predicted index is appended to the input sequence which is now used to predict the second index and so on. Once the desired length is achieved, the visual tokens are discarded and each of the sampled codebook indices is replaced with a codebook entry making the quantized representation which can be decoded by the codebook decoder to obtain a novel

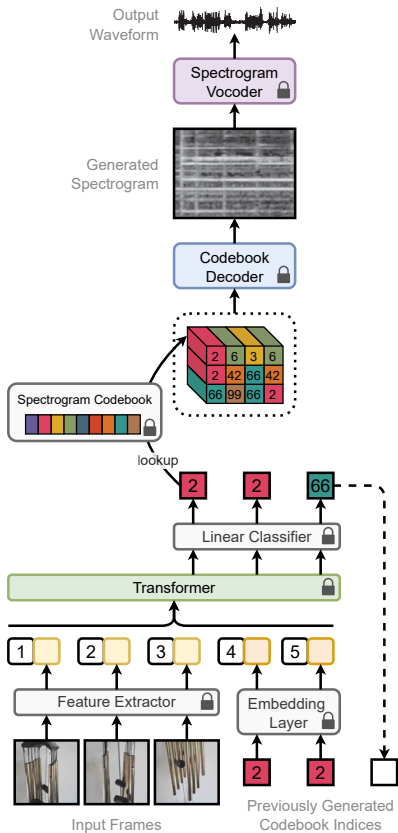


Figure 4.2 The pipeline of generating new audio that is relevant to the input visual cues. The transformer (Stage II) samples codebook indices in an autoregressive manner. To predict the first codebook index, it inputs only visual features. Then, the embedding of the predicted index is appended to the input sequence and passed to the transformer which predicts the second codebook index. This procedure is repeated until the desired length is reached. Next, each predicted index is used to look up the codebook to obtain a quantized representation. The representation is decoded into a spectrogram via the codebook decoder (Stage I) and transformed into a waveform by a spectrogram vocoder.

spectrogram. The generated spectrogram is vocoded to the corresponding waveform.

Spectrogram vocoder The goal of the spectrogram vocoder is to transform a spectrogram into a waveform. We considered a few candidates for the job. In particular, the Griffin-Lim algorithm [497], WaveNet [498], and a GAN-based network. Griffin-Lim is fast to compute on a CPU but reconstruction results are poor. The reconstruction with WaveNet produces high-quality results but it is prohibitively slow, even on a GPU. As a trade-off, in this work, we rely on MelGAN [499] which we pre-trained on a large-scale open-domain audio dataset (VGGSound [495]) from scratch. Although MelGAN was originally designed for speech audio, we found that it performs well even when trained on open-domain videos while taking only a fraction of a second to vocode a 10-second spectrogram on a GPU.

4.3 Automatic metrics for spectrogram-based audio generation

Evaluating generated content is a challenging task. A common way of performing an evaluation is through a human study. However, there are limitations to this approach. First, considering the fast-paced experimentation cycle of a deep learning project, conducting a human study might become the bottleneck. Second, the results of such a study are hard to reproduce which makes it difficult to compare baselines. Thus, every follow-up work in a research area should run an additional comparison with all prior works in the same setting to fairly compare the methods. Third, it is an expensive procedure and might constitute an unbreachable barrier for newcomers. Therefore, in this work, we concentrate on designing automatic metrics for the evaluation of generated audio spectrograms. We aim to evaluate both perceptual quality (fidelity) and relevance to the visual cues.

Fidelity To design metrics for fidelity, we draw on the advances in the evaluation of image generation that rely on a pre-trained classifier. Specifically, *Inception Score* (IS) [500] assumes that meaningful samples are expected to have low entropy in conditional label distribution (classifier’s predictions) produced by *InceptionV3* [454] that was pre-trained on ImageNet. Nowadays, the IS metric was mostly replaced by *Frèchet Inception Distance* (FID) [489]. Unlike IS, FID uses ground truth data as a reference and measures the difference between InceptionV3’s pre-classification layer features of real and fake samples. Inspired by these efforts, in this work, we employ FID to measure the fidelity of the generated spectrogram features. To this end, we pre-trained a version of Inception for audio spectrogram classification on a large-scale open-domain audio dataset (VGGSound). We refer to this network as *Melception*.

Relevance We measure relevance per video as the distance between the distributions produced by the audio spectrogram classifier (*Melception*) for fake and ground truth spectrograms. The ground truth spectrogram is obtained from the audio track which corresponds to the input visual cues that are used to generate the fake spectrogram. The distance between the distributions is computed as a KL-divergence. The final metric is obtained by taking an average of the individual metrics for each video on the whole dataset. We call this metric *Melception-based KL-divergence* (MKL).

4.4 Experiments and results

Datasets In this work, we employ two datasets. First, a small-scale human-curated dataset called *VAS* [16]. It contains around 12.5k videos from 8 classes such as *drum*, *baby*, *sneeze*, *fireworks*, *hammer* etc. The videos are 6.7 seconds long on average. The train/validation/test splits are made following the prior work [16] for a fair comparison. Second, a large-scale automatically annotated open-domain dataset called *VGGSound* [495]. The dataset contains more than 200k videos from YouTube from 309 data classes. Since some videos were no longer available, we could obtain only ~ 190 k videos. We rely on the original train/test split but additionally hold out a validation subset of the training set for development such that the class frequency matches the test set. This split is used during the training of all models, *i. e.* Stages I and II, VGGish-ish, MelGAN, and Melception. As far as we are aware, *VGGSound* has never been used for sound generation. See more details in Publication III.

Metrics As it was outlined in Section 4.3, we employ Melception-based FID for evaluation of the quality (fidelity) of the generated audio spectrograms. While Melception-based KL-divergence (MKL) is used to measure the relevance of the samples to the visual input. We average the per-video MKL across the whole evaluation dataset.

Results: spectrogram reconstruction (Stage I) Considering that multiple parts from Spectrogram VQGAN are going to be used during training of the visually-guided sampler and during inference, it is essential to ensure that the autoencoder is well-trained and reaches strong reconstruction performance. Note, the performance of the Stage II model will be upper-bounded by the performance of the Stage I model.

Based on the quantitative results in Table 4.1, we make several observations. First, Spectrogram VQGAN reaches near-ideal performance in terms of fidelity (FID) and relevance when it is trained and evaluated on the *VGGSound* dataset. Second, the reconstruction quality of the *VAS*-pretrained model is weaker than the *VGGSound*-pretrained codebook when evaluated on the *VAS* dataset. This is the consequence of a) *VAS* having less (10x) training data; b) the *VAS* codebook is smaller³ ($K = 256$ vs 1024 codes); c) the diversity of classes in *VGGSound* covers the diversity of *VAS*.

Similar conclusions can be drawn from the qualitative results shown in Figures 4.3 and 4.4. Specifically, the model that was trained on *VGGSound* produces recon-

³In experiments with a larger codebook, the model failed to use the available capacity and collapsed.

Trained on	Evaluated on	FID↓	$\overline{\text{MKL}}\downarrow$
VGGSound	VGGSound	1.0	0.8
VGGSound	VAS	3.2	0.7
VAS	VAS	6.0	1.0

Table 4.1 Reconstruction performance of Spectrogram VQGAN (Stage I) in a quantitative study on VAS and VGGSound datasets. Metrics are Melception-based FID and MKL metrics.

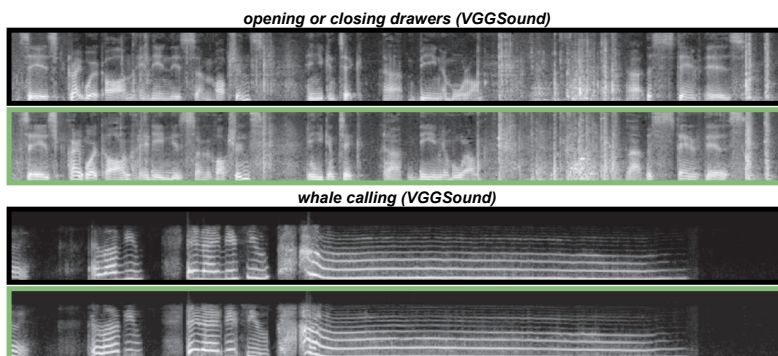


Figure 4.3 Qualitative results of spectrogram reconstruction with Spectrogram VQGAN (Stage I): examples from VGGSound. Spectrograms: ground truth (**top**) and reconstructed with the Spectrogram VQGAN pre-trained on VGGSound (**bottom**).

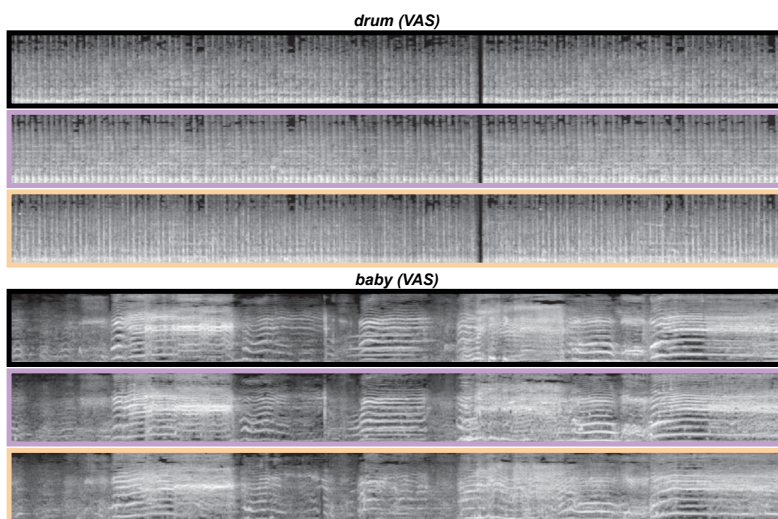


Figure 4.4 Qualitative results of spectrogram reconstruction with Spectrogram VQGAN (Stage I): examples from VAS. Spectrograms: ground truth (**top**), reconstructed with the Spectrogram VQGAN pre-trained on VGGSound (**middle**) or pre-trained on VAS (**bottom**).

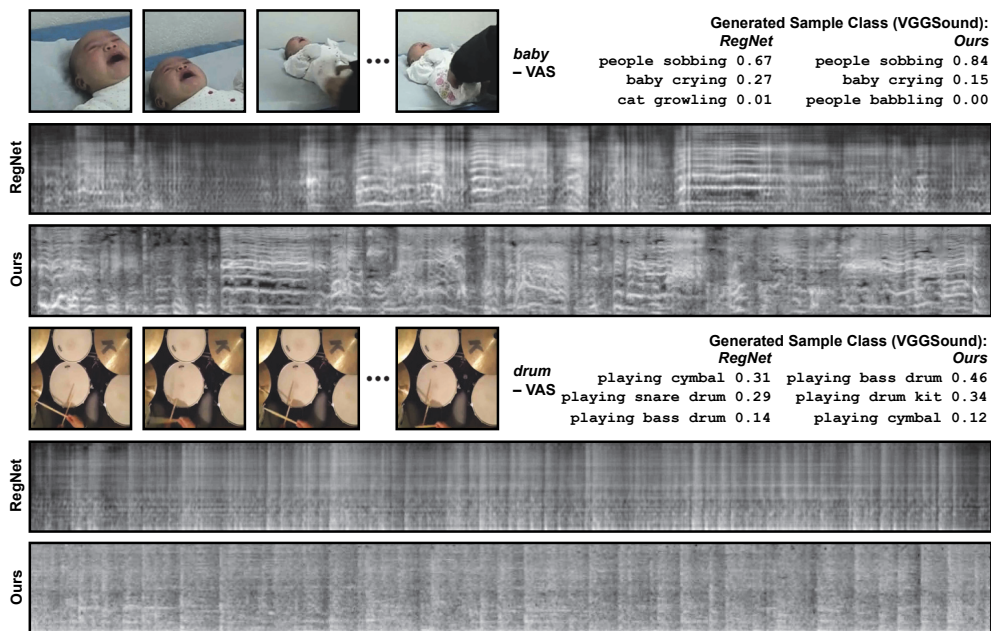


Figure 4.5 Qualitative comparison of our approach to the state-of-the-art baseline model [16] performing visually-guided audio generation on the VAS dataset.

structions that are close to the ground truth. In addition, the reconstruction of the VGGSound-pre-trained model generalizes well on the VAS dataset, *i. e.* the reconstructions are close to the input, without fine-tuning on VAS. Similar to the quantitative results, the reconstructions obtained with the VAS-pre-trained codebook tend to be of a lower quality than those that are produced by the model with the VGGSound-pre-trained codebook.

Results: visually-guided audio generation In quantitative experiments, we compare the performance of the proposed model on two datasets across multiple settings.

First, to put the metric values into context, we additionally report the performance of a model that was trained without visual cues. It acts as a baseline for relevance (MKL). The results in Table 4.2 show the importance of visual cues in generating relevant sounds which is visible when comparing MKL values to the setting without visual features on VAS and VGGSound datasets, yet the gap is smaller in the later case due to the larger number of classes in VGGSound.

Second, for a fair comparison to the baseline (discussed later), we also include the results with BN-Inception frame-wise features (RGB + optical flow) besides the

Evaluation dataset: VAS (8 classes)

Model	Codebook	Input features	Params↓	FID↓	$\overline{\text{MKL}}\downarrow$	$\odot\downarrow$
Ours	VGGSound	—	379M	33.7	9.6	8
Ours	VAS	—	377M	28.7	9.2	8
Ours	VGGSound	ResNet-50	379M	20.8	6.2	12
Ours	VAS	ResNet-50	377M	22.6	5.8	12
Ours	VGGSound	BN-Inception	379M	20.5	6.0	12
Ours	VAS	BN-Inception	377M	25.4	5.9	12
RegNet [16]	—	BN-Inception	$8 \times 105\text{M}$	78.8	5.7	1500
Ours	VGGSound	BN-Incept. + cls	379M	20.2	5.7	12
Ours	VAS	BN-Incept. + cls	377M	24.9	5.5	12

Evaluation dataset: VGGSound (309 classes)

Model	Codebook	Input features	Params↓	FID↓	$\overline{\text{MKL}}\downarrow$	$\odot\downarrow$
Ours	VGGSound	—	379M	13.5	9.7	8
Ours	VGGSound	ResNet-50	379M	10.5	6.9	12
Ours	VGGSound	BN-Inception	379M	9.6	6.8	12

Table 4.2 Comparison of visually-guided sound generation models across two datasets: VAS (top) and VGGSound (bottom). The evaluation performance of the proposed codebook code sampler (Stage II) is shown with VAS- and VGGSound-pretrained codebooks and two sets of features: ResNet-50 (RGB) and BN-Inception (RGB + optical flow). The performance of the model without visual cues is also reported. The number of trainable parameters (Params) is reported in millions. Fidelity and relevance are measured with Melception-based FID and MKL (averaged across the dataset) metrics. We also compare the sampling speed in seconds (\odot). Notice that the baseline architecture was trained for each class separately, while our model supports all 8 VAS classes at once.

$\mathcal{L}_{\text{VQVAE}}$	$\mathcal{L}_{\text{PatchGAN}}$	$\mathcal{L}_{\text{LPAPS}}$	FID↓	$\overline{\text{MKL}}\downarrow$
✓			130.4	9.6
✓	✓		1.4	1.1
✓	✓	✓	1.0	0.8

Table 4.3 The effect of adding PatchGAN and perceptual losses during training of Spectrogram VQGAN. The results are measured with two Melception-based metrics FID and MKL and reported on the test set of VGGSound.

ResNet-50 (RGB only). For this experiment, we use 212 frames extracted at 21.5 fps from 10-second videos. We observe that, despite the absence of optical flow cues, ResNet-50 features are on par with BN-Inception features in fidelity and relevance on both datasets. In addition, the benefits of using ResNet-50 instead of BN-Inception include the availability of open-source implementations and ease of use.

Third, we compare the results using different codebooks as the building blocks for the sampler. Interestingly, the model that samples from the VGGSound-pre-trained codebook reaches strong performance on the VAS dataset which suggests that the resulting codebook codes that are used for sampling appear to be general enough to maintain comparable performance when applied to another dataset.

Fourth, we additionally append the class token to the input sequence of visual features to provide an explicit signal to hint to the model which class of the generated sample is expected. This is done to compare to the baseline approach (RegNet [16]) which trains one model per data class. Although this is still a more challenging setting compared to the baseline, our model significantly outperforms it in terms of fidelity (FID) while being on par in terms of relevance (MKL). Moreover, generation takes more than 100x less time to generate a 10-second video and the total number of parameters is substantially smaller considering that our model supports all classes while the baseline requires training one model per class.

The qualitative comparison to the baseline (RegNet [16]) is depicted in Figure 4.5. Although it might be difficult to convey and appreciate the generated sample in the paper format, yet the difference is quite drastic even by visual inspection of the spectrograms. Specifically, the samples produced by the baseline are noticeably more blurred and noisy. To provide more evidence, we include the audio classification results of the Melception network that was trained on ground truth audio spectrograms on the VGGSound dataset. To conclude, our model generates less blurred and less noisy (higher fidelity) spectrograms which are also relevant to the video data class.

Ablation: effect of additional loss terms for reconstruction In Table 4.5, we show the importance of including adversarial and perceptual losses into the training objective of Spectrogram VQGAN during Stage I (see page 67). This conclusion is consistent with [39]. For more ablation experiments, qualitative results, and a variety of applications of the proposed model a reader is referred to the Publication III (especially to the supplementary material).

4.5 Discussion

Related work: new developments Publication III was published in late 2021 and a few exciting developments appeared in the literature ever since. In particular, Ghose and Prevost [501] focused on building a GAN-based model that generates visually aligned (synchronized) audio samples given the visual features. Hayes *et al.* [396] also used the two-stage approach (3D VQVAE and a transformer) to generate audio given video or text and introduced a new large-scale dataset collected from a game engine. Choi *et al.* [502] outlined the design of a foley sound synthesis challenge which might guide the future efforts towards building a better visually-guided sound generation model and more unified evaluation techniques. In the meantime, a similar codebook-based sampling approach was explored by Sheffer and Adi [503] who relied on VQVAE-2 [504] and CLIP [2] visual features. Cui *et al.* [505] suggested guiding the acoustic characteristics of the generated audio given a reference audio sample along with video frames.

Future research: better datasets In this work, we relied on the small-scale VAS and large-scale VGGSound datasets. Although the audio-visual correspondence is strong in VAS, it is a rather small dataset with as few as 8 data classes and those are not covered by samples well, *e. g.* there are only about 300 videos for some classes. In the case of VGGSound, it is substantially larger but the audio-visual correspondence is quite poor due to the automatic nature of annotations. In fact, the content of the audio track does correspond to the data class in most of the videos that we observed, yet the visual track often does not depict the content of the audio. Therefore, the new bigger dataset with stronger audio-visual correspondence would certainly not only improve the results of visually-guided sound generation but also positively influence the video understanding community. One of the developments towards this direction is the ACAV100M dataset [506], which aims to extend the VGGSound dataset (by 100+ times). Unfortunately, most of the videos in this dataset contain speech (85+ %) and it significantly limits its application to general-purpose tasks.

Future research: audio-visual alignment While working on Spectrogram VQ-GAN, we noticed that the generated audio samples are rarely synchronized temporally with visual content. We attribute it to the quality of the VGGSound dataset which is relatively noisy. More specifically, as we found in Publication IV, audio-

visual synchronization is possible for a small portion of the videos due to the lack of strong correspondence between the streams. Besides, VGGSound includes classes which are difficult to synchronize, *e. g. helicopter, cat purring, bee buzzing*, etc. Therefore, we, once again, encourage the development of a large-scale open-domain audiovisual video dataset. Another reason behind the lack of synchronization is the video feature extractor. In particular, we rely on pre-trained (and frozen during training) ResNet-50 which does not encode any temporal interaction between frames. Thus, experiments with 3D convnets that are fine-tuned during training should be performed in future works to facilitate development in this direction.

Future research: an even better spectrogram codebook In this work, we managed to train a strong autoencoder called Spectrogram VQGAN on the VGGSound dataset. Nonetheless, we noticed that the difference between the original and reconstruction of speech and music is noticeable when comparing both side-by-side. Therefore, we believe that one could try to train an even better codebook, *e. g.* on the large-scale the LAION Audio dataset [507], which is a combination of higher-quality audio data compared to the YouTube-based VGGSound dataset. Notice that, in order to train a codebook, only a large-scale open-domain *audio* dataset is required.

Future research: latent diffusion model instead of autoregressive sampler We rely on the two-stage approach for spectrogram generation. First, the codebook is trained by an autoencoder. Second, the transformer is trained to autoregressively sample codebook codes to produce the bottleneck representation of a new spectrogram. As it was shown for image generation in [12], condition-guided latent diffusion model could replace the transformer in the second stage. This should allow to train a better conditional spectrogram generation model and drastically improve the results.

5 AUDIO-VISUAL SYNCHRONIZATION WITH SPARSE SIGNALS

Given audio and visual video tracks, that are potentially out of sync, an audio-visual synchronization model is expected to predict the temporal offset between the tracks. Solutions to the audio-visual synchronization task could expand the functionality of video editing software, *i. e.* by notifying an editor if the tracks are out of sync.

We differentiate “sparse” and “dense” synchronization signals. For instance, a video of a talking person (*e. g.* a presentation) has “dense” synchronization signals in the time since at each second of a video it is possible to tell if audio and visual tracks are in-sync. Whereas, a video of a dog that barks once in a ten-second video clip exhibits a “sparse” synchronization signal, and a model needs to process the whole video to synchronize it which makes it a challenge for recent deep-learning models. Besides the sparseness in *time*, we also specify the sparseness in space. For example, the cropped video of a talking person is “dense” in space (and time).

Previous works in audio-visual video synchronization focused primarily on videos with “dense” synchronization cues such as recordings of cropped talking faces. However, open-domain videos can have synchronization cues that are sparse in both time and space. To bridge this gap, we explore the synchronization of videos with sparse signals and introduce a novel multi-modal transformer-based architecture that allows us to effectively process longer videos. Moreover, we introduce a new video dataset with “in the wild” videos with sparse synchronization signals, called VGGSound-Sparse. In addition, we found that common video compression algorithms leak temporal artefacts that a synchronization model could use to learn a shortcut.

This chapter begins with the related work section (5.1) which is followed by the description of the proposed network in Section 5.2. Next, Section 5.3 presents the ways of detecting and preventing temporal artefacts in the video data. Then, the experimentation pipeline and results are presented in Section 5.4. While Section 5.5 discusses future research directions.

5.1 Related work

Audio-visual synchronization of face tracks Earlier work in audio-visual synchronization focused on the synchronization of face tracks rather than open-domain videos. Prior to the deep learning era, methods relied on hand-crafted features as in Hershey and Movellan [508] and in Slaney and Covell [509]. Deep learning certainly moved the progress in audio-visual synchronization forward as well. In particular, Chung and Zisserman [510] focused on the synchronization of lip movements and a two-stream CNN architecture (SyncNet) that was trained contrastively by pooling apart embeddings from both streams, while Chung *et al.* [511] improved this approach with a multi-way classification among multiple negatives. Halperin *et al.* [512] explored audio-visual alignment of a re-dubbed visual scene, which is a more general case of solving synchronization, and used Dynamic Time Wrapping [513] to this end. Khosravan *et al.* [514] showed that spatio-temporal attention improves the synchronization performance. Kim *et al.* [515] suggested classifying the audio-visual embedding similarity matrix as an image to determine the offset. Meanwhile, Kandale *et al.* [516] used multiple transformer’s decoders to determine if audio and visual streams are in-sync. Although these methods reach promising performance, the main focus is the synchronization of talking faces rather than “in the wild” videos.

Audio-visual synchronization of open-domain videos Others explored the synchronization of open-domain videos. More specifically, a handful of data classes was used in Casanovas *et al.* [517] who investigated audio-visual synchronization of streams from multiple cameras. While, more recently, Chen *et al.* [518] introduced a novel general-purpose dataset of 160 classes and proposed a new transformer-based method for audio-visual synchronization. In contrast to these methods, we focus on videos with sparse sync signals, which is more challenging compared to previous approaches.

Shortcut training with temporal artefacts The negative influence of artefacts was brought to attention in the seminal work of Doersch *et al.* [519] in the self-supervised setting. The potential impact of temporal artefacts located in black regions of a video frame was explored with respect to predicting if the video frames are reversed or not (aka. “arrow-of-time”) by Wei *et al.* [520]. Another piece of evidence was reported by Arandjelović *et al.* [521] who noticed the difference in performance with MPEG-

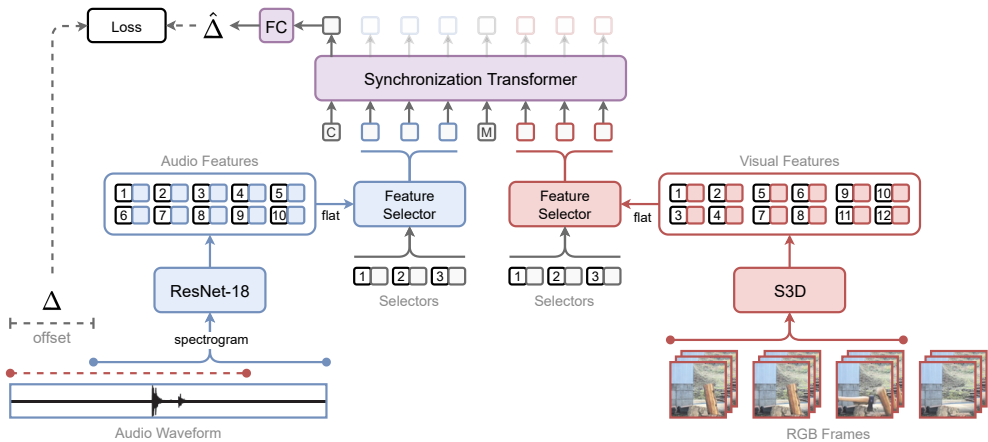


Figure 5.1 The overview of the audio-visual synchronization model: SparseSync. The model inputs audio and visual streams that are potentially out of sync. First, an audio waveform is encoded into a spectrogram. Then, audio features are extracted from the spectrogram with a variant of ResNet-18. Next, a small set of trainable selector vectors (bottom) is passed to the Feature Selector module along with full-length audio features. Feature Selector effectively picks useful cues from audio features via cross-attention modules, and outputs refined selector vectors. Similarly, the RGB stream is encoded into spatio-temporal features via the S3D network and passed to the corresponding Feature Selector module that outputs another set of refined selectors. Both sets of selectors form an input to the transformer after concatenation with auxiliary tokens for class (C) and modality separation (M). The outputs of the Synchronization Transformer are used to predict a temporal offset. The behaviour of SparseSync during training is shown in dashed lines. For visualization purposes, we zoom in on the content of RGB frames.

encoded videos when trained for audio-visual correspondence. In contrast, this work investigates methods of detecting temporal artefacts in videos as well as provides suggestions that help to avoid them.

5.2 Audio-visual synchronization with sparse signals (SparseSync)

Given audio and visual tracks, the goal of the synchronization model is to predict if the tracks are in-sync and, if not, what is the size of the offset between the tracks. As it was discussed before, “in the wild” videos may be difficult to synchronize as these may contain sparse (rare) synchronization signals. In order to maximize the chance of having the sparse synchronization signal within the trimmed sequence of frames, the trim should be long (e. g. 5 seconds). Nonetheless, long input sequences pose a severe obstacle for transformer-based architectures that are state-of-the-art approaches

for sequence modelling currently. In this work, we introduce a novel transformer-based model, called *SparseSync*, which allows the processing of long input videos. It is achieved with trainable *selectors* which effectively encode useful cues from long sequences of audio and visual features. After encoding with selectors, audio and visual features form an input to the transformer which is now of a manageable length. Finally, the transformer makes a prediction regarding the potential temporal offset between audio and visual input streams. The overview of the architecture is depicted in Figure 5.1 and the details of the approach are described next.

Feature extraction The model inputs an audio spectrogram $A \in \mathbb{R}^{F \times T_a \times 1}$ extracted from a waveform (16kHz) and a sequence of T_v RGB frames $V \in \mathbb{R}^{T_v \times H \times W \times 3}$ extracted at 25 fps. The audio and spatio-temporal visual features are extracted as

$$a = E_a(A), \quad v = E_v(V), \quad (5.1)$$

where $a \in \mathbb{R}^{f \times t_a \times d_a}$ are audio features and $v \in \mathbb{R}^{t_v \times b \times w \times d_v}$ are visual features and $E_{a/v}$ are feature extractors. More specifically, we used a variant of ResNet-18 [453] as the audio feature extractor, which was pre-trained on audio classification on the VGG-Sound dataset [495]. The visual features are extracted with the S3D network [15] that was pre-trained on Kinetics-400 [418]. In our experiments, the spatio-temporal features performed better compared to frame-wise 2D features. The outputs of the pre-classification layers are used as the final features. Features are mapped to the common dimension d (e. g. 512) from d_a and d_v , respectively. During training, we fine-tune the weights of feature extractors along with the rest of the architecture.

Feature Selectors The extracted features are *flattened* in two sequences of token embeddings $\hat{a} \in \mathbb{R}^{f \cdot t_a \times d_a}$ and $\hat{v} \in \mathbb{R}^{t_v \cdot b \cdot w \times d_v}$. Notice that audio-visual synchronization requires higher visual fps to perform the synchronization than other tasks (e. g. action recognition) which makes the problem even more challenging. Thus, these sequences may easily contain hundreds of elements for seconds-long inputs which might be unattainable for a transformer due to the quadratic complexity of the attention mechanism. To this end, inspired by works like DETR [3] and Perceiver [46], we propose to have a small set of trainable query vectors, which we call *selectors* ($q_{a/v}$). The selectors are used in corresponding *Feature Selector* modules and effectively encode useful features for synchronization from the features. The design of

the Feature Selector is similar to the transformer decoder [4] (and also defined in Section 2.3). Similar to other transformer architectures, we add positional encoding (PE_*) to modality feature sequences and selectors. The Feature Selectors ($F_{a/v}$) output two sequence of refined selectors ($\hat{q}_{a/v}$) for audio and visual modalities:

$$\hat{q}_a = F_a(\hat{a} + PE_{a_s}, q_a + PE_{q_a}), \quad \hat{q}_v = F_v(\hat{v} + PE_{v_s}, q_v + PE_{q_v}), \quad (5.2)$$

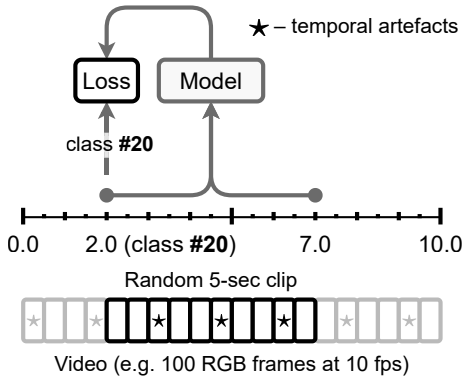
where $q_a, \hat{q}_a, q_v, \hat{q}_v \in \mathbb{R}^{k \times d}$ and k is the number of trainable selectors which we keep being small and equal for both modalities (e. g. 16). The design of the Feature Selector allows encoding useful cues from modality features in k vectors (i. e. $k \ll f \cdot t_a$ and $k \ll t_v \cdot b \cdot w$) which casts the complexity of the transformer to be linear with respect to the input length because k is fixed.

Synchronization Transformer The architecture of *Synchronization Transformer* (S) is similar to the transformer encoder (as defined in Section 2.3). The outputs of Feature Selectors ($\hat{q}_{a/v}$) are used in the transformer. More specifically, encoded selectors are concatenated with the typical classification token (<CLS> or C in Figure 5.1) and modality separation token (<MOD> or M)

$$\hat{\Delta} = S\left([\text{CLS}; \hat{q}_a; \text{MOD}; \hat{q}_v]\right). \quad (5.3)$$

To avoid notation overload, the Equation 5.3 omits the fully connected layer with softmax that is applied to the first element of the output sequence to predict the temporal offset $\hat{\Delta}$ between audio and visual tracks.

Training SparseSync Assuming that the audio and visual tracks of the majority of web videos are in-sync, we artificially create the temporal offset (Δ) between the tracks. The offset amount is picked randomly from the grid of offsets from -2 to $+2$ seconds with 0.2 seconds step, including the “no offset” class (0.0 seconds). Therefore, the offset prediction is formulated as a 21-way classification task. We train the model with a typical cross-entropy loss to predict the offset class. All weights, except for feature extractors, are initialized from scratch. Other training and implementation details are available in Publication IV (especially the supplementary material).



Codec	Acc@1	Acc@5
MPEG-4 Part 2 (mpeg4)	27.2	77.1
MPEG-4 Part 10 (H.264)	2.5	11.9
ProRes	2.7	13.4
AAC @ 44100Hz	86.7	100.0
AAC @ 22050Hz	23.0	74.3
AAC @ 16000Hz	6.3	19.3
Lossless @ 22050Hz	2.9	14.6

Table 5.1 Training to detect temporal artefacts. *Left*: the crop start-time prediction training procedure that allows to detect temporal artefacts in the data. The start of the crop is quantized to the grid of 50 classes with the step size of 0.1 second, *i. e.* one frame. *Right*: the evaluation results obtained after training a model to predict the start of the temporal crop. The results are reported across several RGB and audio encoding algorithms and with two accuracy metrics (top 1 and 5). The desired performance is 2 and 10 %.

5.3 Preventing temporal artefact leakage

Common video and audio codec algorithms leave temporal artefacts in the data streams that a model could use to solve the synchronization or correspondence task. In this section, we show a simple way of detecting temporal leakage, discuss its origin, and provide a few recommendations on how to mitigate it.

Detecting artefact leakage

In this work, we developed a simple method that may reveal if a stream exhibits temporal artefacts. More specifically, we train a model to predict the start time of the input segment. Of course, training such a model should not be possible without temporal cues in the data, but this is not what happens. The training procedure is schematically shown in Table 5.1 (left).

RGB stream In this experiment, we compare several visual codecs. Specifically, MPEG-4 Part 2 (aka. mpeg4) and MPEG-4 Part 10 (aka. H.264 or AVC) are common video codecs that rely on temporal frame predictions (inter-frame codecs). We also experiment with ProRes codec which encodes frames independently (intra-frame codec), and should not leak any temporal artefacts into the stream. Therefore, we

download the dataset of ProRes videos (MJPEG-AoT [520]) and transcode the videos into either mpeg4 or H.264. We cut the videos into 10-second clips and use random 5-second trims as inputs. The comparison is shown in Table 5.1 (right, top). In particular, we observe that only MPEG-4 Part 2 exhibits temporal artefacts as the accuracy of classifying the start of the start time of the crop is substantially higher than the ones of H.264 and ProRes, even though the videos have the same content. Although the performance of the latter two was slightly higher than the desired accuracy values (2 and 10 %), we believe it is negligible.

Audio stream We conduct a similar experiment for the extremely common audio codec, *i. e.* Advanced Audio Codec (AAC). The results are shown in Table 5.1 (right, bottom). For this experiment, we generate a tensor of Gaussian noise in a form of a 10-second waveform and save it to the disk lossless as an audio file with a specified sampling rate (16–44.1kHz). Then, we transcode the waveform file into AAC and use it as the input from which we randomly crop a 5-second clip similar to the experiments with RGB codecs. According to the results, all experiments with AAC as the input showed evidence of artefact leakage which can be observed by higher values of predicting the start time of a crop compared to the experiment with lossless data. However, the lower the sampling rate, the lower (better) the results.

Mitigating the impact of temporal leakage

RGB stream: use H.264 (AVC) instead of MPEG-4 Part 2 As it was shown empirically in Table 5.1, we were able to train a model to predict the start-time of the visual stream trim if it was encoded as MPEG-4 Part 2. We attribute this to the strict key-frame allocation pattern. For instance, I-frame¹ is allocated every 12th frame while H.264 has a more complex key-frame prediction pattern which depends strongly on the content of the frame and, therefore, it is hard for a model to learn it. Similarly, the model fails to learn to perform this task on the dataset with ProRes encoding in which every frame is independently encoded. The most obvious solution would be to transcode all MPEG-4 Part 2 into H.264 but transcoding does not remove the artefacts. Then, ideally, one would rely on ProRes when building

¹I-frames is a reference frame that is encoded independently from other frames in the video. In a sequence of, for example, 12 frames, the rest of the 11 frames (P-frames) re-use the content of the I-frame for compression purposes. The illustration of this effect is shown in the supplementary material to Publication IV.

a dataset, yet it is rarely possible as YouTube, which is a common source for most of the large-scale general-purpose video understanding datasets, does not broadcast ProRes. Therefore, we recommend avoiding MPEG-4 Part 2 in favour of H.264 (AVC).

Audio stream: reduce the sampling rate Similar to the visual stream, the audio stream also contains temporal artefacts if encoded with Advanced Audio Coding (AAC). Although the nature of temporal artefacts in the audio stream is unknown, we hypothesise that it might be due to the same reasons as for the visual stream with MPEG-4 Part 2 coding algorithm. Since AAC is an extremely common codec for any video dataset with audio tracks, it might be challenging to avoid the artefacts completely. Therefore, considering that the severity of the artefacts depends on the sampling rate (see Table 5.1), we recommend avoiding high audio sampling rates and stick to the lowest rate that is acceptable for the researcher’s needs.

5.4 Experiments and results

Dense in time dataset We consider two variations of the “dense in time” dataset: “dense in time and in space” and “dense in time but sparse in space”. To this end, two variations of Lip Reading Sentences 3 (LRS3) dataset [14] are used for experimentations: face-cropped (“dense in space”) and full-scene videos (“sparse in space”). We obtain the original videos of LRS3 from YouTube. The obtained RGB streams are encoded in H.264 and resampled at 25 fps, while the audio stream is encoded in AAC and resampled 16kHz sampling rate. We use the pretrain subset and filter out the videos that are shorter than 9 seconds for consistency with the VGGSound-Sparse dataset. We split the data into 80/10/10% as train/validation/test sets. Ultimately, we end up with ~58k video clips from ~5k videos. The model inputs 5-second clips randomly cropped out of the original videos.

Sparse in time dataset In this work, we present a novel dataset which contains videos with sparse synchronization signal. This dataset is built upon the VGGSound dataset [495] which is a collection of 200k+ ten-second YouTube clips spanning 309 data classes. Our dataset is a curated subset of VGGSound and collected by manual inspection of 5–15 randomly picked videos for each of 309 classes and annotated whether audio-visual synchronization cues are sparse in time. As a result, 12

data classes were selected or 7.1k videos in total. This new dataset is referred to as *VGGSound-Sparse*. We rely on the same train-test split as in VGGSound. Since the original VGGSound dataset is noisy due to the automatic annotation, we additionally inspect 20 videos per a selected sparse class and observed that $\sim 70\%$ of all videos are “synchronizable”. Similar to the “dense in time” setting, the models input random 5-second segments of original videos. For the full list of data classes see Table 5.3.

Problem setting and metrics Compared to most of the prior work which solves the sync/out-of-sync classification problem (two classes), our setting requires a model to predict the actual amount of offset which makes it more difficult. We formulate the synchronization problem as a 21-way classification task. During training, we pick an offset class (offset value) with equal probabilities from the grid of 21 offsets from -2 to $+2$ seconds with a step size of 0.2 seconds. The design of the offset grid is inspired by the results of the ITU Radiocommunication Assembly that conducted a subjective evaluation of the thresholds of acceptability for the delay between audio and visual tracks. In particular, it was found that the thresholds are -185 ms to $+90$ ms [522]. For this reason, we track the metrics with ± 1 class temporal tolerance (± 0.2 seconds). Considering the balanced distribution of classes, we rely on accuracy in the following experiments.

Results: audio-visual synchronization We compare the proposed model (SparseSync) to the state-of-the-art baseline by Chen *et al.* [518] (AVST). Similar to our approach, AVST is a transformer-based model. In particular, it uses audio features as *queries* to the visual features as *context*. This, however, scales poorly with the input length compared to our approach. We enhanced the baseline architecture to be suitable for predicting 21 classes instead of two classes (in-sync/out-of-sync) and using 5-second video clips at 25 fps instead of 5 fps. For the “sparse in time and space” setting we pre-train the architectures on the LRS3 (full-scene) dataset.

Table 5.2 presents the comparison of both methods performing audio-visual synchronization on three datasets with varying degrees of sparseness. According to the results, our model strongly outperforms the baseline by a large margin across all datasets. Therefore, we conclude that the proposed model achieves strong performance on both the less challenging “dense in time” datasets as well as on the “sparse in time and space” dataset. Additionally, the performance per data class on the VGGSound-Sparse dataset is reported in Table 5.3.

<i>Dense-dense: LRS3 (face crop)</i>	
Model	Accuracy $^{\pm 1 \text{ cls}}$ ↑
AVST [518]	89.8
Ours	95.6

<i>Dense-sparse: LRS3 (full-scene)</i>	
Model	Accuracy $^{\pm 1 \text{ cls}}$ ↑
AVST [518]	83.1
Ours	96.9

<i>Sparse-sparse: VGGSound-Sparse</i>	
Model	Accuracy $^{\pm 1 \text{ cls}}$ ↑
AVST [518]	29.3
Ours	44.3

Table 5.2 Results of comparison between the state-of-the-art baseline [518] and SparseSync (Ours) on the audio-visual synchronization task across three settings. First, the results of the “dense in time and space” are reported on the LRS3 (face crop) dataset. Second, the LRS3 (full-scene) dataset is used for “dense in time and sparse in space”. Third, the novel VGGSound-Sparse dataset for “sparse in time and space” setting. The evaluation is conducted on the test subsets of the datasets. The performance is measured with accuracy across 21 offset classes with ± 1 temporal class tolerance (± 0.2 seconds).

Pre-trained on LRS3 (full-scene)	Fine-tune Feature Extractors	With Selectors	Accuracy $^{\pm 1 \text{ cls}}$ ↑
\times	\checkmark	\checkmark	12.1
\checkmark	\times	\checkmark	33.5
\checkmark	\checkmark	\times	40.1
\checkmark	\checkmark	\checkmark	44.3

Data class	Accuracy $^{\pm 1 \text{ cls}}$ ↑
playing badminton	53.6
striking bowling	52.3
chopping wood	50.9
hammering nails	49.0
people sneezing	46.6
playing tennis	46.0
ice cracking	44.5
skateboarding	40.8
people eating crisps	38.7
people eating apple	34.6
dog barking	32.5
lions roaring	30.0

Table 5.3 The performance of SparseSync per data class of VGGSound-Sparse. The metric is accuracy which is measured across 21 offset classes with ± 1 temporal class tolerance (± 0.2 seconds). The average weighted performance is 44.3%.

Table 5.4 Ablation study SparseSync. The results are reported on the VGGSound-Sparse dataset. The performance is measured with accuracy across 21 offset classes with ± 1 temporal class tolerance ± 0.2 sec.

Ablation: effect of pre-training, selectors, and gradients for feature extractors

In Table 5.4, we report the results of a few ablation experiments for the introduced the SparseSync model on the “sparse in time and space” setting and highlight the following observations. *First*, we look at the effect of pre-training on a simpler dataset, *i. e.* LRS3 (full-scene) or the “dense in time and sparse in space” dataset. The results suggest that pre-training is absolutely essential as the performance drops to the near-random level (12%). *Second*, allowing gradients to update the weights of the audio and RGB feature extractors brings substantial gains compared to training with frozen feature extractors (33.5 vs. 44.3%). *Third*, the addition of the selectors compared results in a small improvement in the performance due to the increased capacity of the model. The architecture of the model without selectors is a transformer encoder that inputs concatenated sequences of audio and visual features. Notice that the model with selectors not only performs strongly but also has linear complexity with respect to the input length due to the fixed number of trainable selector vectors. Further ablation studies, including the visualization of attention maps and experiments with longer inputs, are discussed in Publication IV.

5.5 Discussion

Future research: further exploration of temporal artefacts This work investigates the problem of temporal artefacts in video data. We presented a straightforward method of detecting these artefacts, *i. e.* by training a classifier for start-time prediction. We believe these artefacts are caused by the codec algorithms which use temporal encoding. For the RGB stream, switching to H.264 codec solves the problem and having the dataset in this codec is attainable. While, in the case of Advance Audio Codec (AAC), this issue needs more exploration as most of the datasets and sources for video data encode audio in AAC. So far, we noticed that by decreasing the sampling rate of the audio coding one could reduce the impact of artefacts yet not completely. Notice, transcoding from one codec to another (*e. g.* MPEG-4 Part 2 to H.264) does not remove artefacts from the data streams and, thus, the issue requires more caution.

Future research: “sparse in time but dense in space” setting In this work, we explored three out of four possible settings in terms of the synchronization signal density. Specifically, we looked at “dense in time and space” and “dense in time but

sparse in space” settings with the variants of the LRS3 dataset. Also “sparse in time and space” setting was benchmarked with the newly proposed VGGSound-Sparse dataset. Nonetheless, “sparse in time but dense in space” was left uninvestigated. Although exploring this setting might be interesting, it is challenging to construct an open-domain dataset as one might need to rely on a pre-trained object detector which limits the applicability to the object detector classes.

Future research: larger audio-visual dataset Similar to other applications that were discussed in this thesis, the task of audio-visual synchronization should benefit from training on a large-scale audio-visual dataset. Although VGGSound is certainly a good effort to this end, the dataset is rather noisy due to the automatic annotation and, thus, audio-visual correspondence is weak in many cases. More specifically, after manual inspection, it was noticed that the majority of inspected videos had an audio track which corresponded to the annotation data class, yet the visual indication might be missing. On top of it, some classes are hard to synchronize, *e. g. running electric fan, train whistling, or wind chime*. Therefore, efforts in creating a new large-scale general-purpose audio-visual dataset are highly encouraged.

6 CONCLUSION

This thesis advances the state-of-the-art of multi-modal video understanding. In particular, it proposes inventions that push further the performance of dense video captioning, video-guided audio generation, and audio-visual synchronization. As well as, it provides a comprehensive review of the earlier and recent arts in the field. The posed research questions were thoroughly investigated in this work.

In particular, the effect of additional modalities, such as speech and audio, on the performance of a video understanding model which is a crucial research problem. We explore this question in the context of dense video captioning. To this end, Sections 3.2 and 3.3 introduce two novel multi-modal transformer-based architectures that effectively encode the multi-modal cues. Both models demonstrate strong performance gains in comparison to uni-modal approaches and outperformed state-of-the-art (see Section 3.4). We believe that building a better model for video captioning has an opportunity to help the visually impaired engage in brighter social interactions online.

Next, we outlined a new approach that makes possible the generation of sounds that are relevant to the content of an open-domain video clip. Specifically, we proposed to factorize the task into two sub-problems (Section 4.2). First, the spectrogram autoencoder with the latent codebook is trained. Second, an autoregressive model is trained to pick the codes from the codebook while being conditioned on visual cues. This two-staged approach not only allows for audio generation for open-domain videos in a single model but also produces higher-quality samples while being more than two orders of magnitude faster than the state-of-the-art. On top of this, we introduced a suite of spectrogram-based automatic evaluation metrics (Section 4.3). These metrics mitigate the need for expensive and tedious human evaluation, and speed up the development cycle of a sound generation model. The advances in a conditional sound generation find their applications in foley (sound) design for movies and digital art.

In addition, we proposed a new and effective approach to encoding long audio-visual input sequences. In particular, a small set of trainable vectors are used as “queries” to the sequence of input features (Section 5.2). We demonstrated the efficiency of this approach to the audio-visual synchronization of videos with sparse synchronization signals. These videos are often many seconds long and the use of state-of-the-art architectures (transformer) requires a substantial GPU memory footprint. Moreover, we discovered that common video and audio compression algorithms leave temporal artefacts in data streams which may allow the synchronization model to learn a shortcut (Section 5.3). To this end, we outlined a simple way of detecting them, *i. e.* training to predict the temporal crop start-time, as well as listed the recommendations on how to avoid them. Solutions for out-of-sync detection could be implemented in video editing software, which grows in demand as video editing is becoming more common among non-professionals.

Although the field of multi-modal video understanding experienced great progress in the last decade, there is still a long road ahead. Currently, the main issue in the area is the lack of large-scale open-domain datasets with strong audio-visual or visual-linguistic correspondence. Constructing such datasets would certainly move the state-of-the-art further. In connection with the previous, recent progress in the development of foundation models hints towards exploiting strong cross-modal correspondence in training a general-purpose base model that solves a variety of downstream tasks and lessens the need for task-specific architectural design.

REFERENCES

AAAI	Proc. AAAI Conference on Artificial Intelligence
BMVC	British Machine Vision Conference
CVPR	Proc. IEEE(/CVF) Conference on Computer Vision and Pattern Recognition
ECCV	Proc. European Conference on Computer Vision
EMNLP	Proc. Conference on Empirical Methods in Natural Language Processing
ICASSP	IEEE Int. Conference on Acoustics, Speech and Signal Processing
ICCV	Proc. IEEE(/CVF) Int. Conference on Computer Vision
ICLR	Int. Conference on Learning Representations
ICML	Int. Conference on Machine Learning
NAACL	Proc. Conference of the North American Chapter of the ACL: Human Language Technologies
NeurIPS	Advances in Neural Information Processing Systems
TPAMI	IEEE Transactions on Pattern Analysis and Machine Intelligence

- [1] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” in *NAACL*, Minneapolis, Minnesota: ACL, 2019.

- [2] 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 *ICML*, 2021.
- [3] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, “End-to-end object detection with transformers,” in *ECCV*, Springer, 2020.
- [4] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” *NeurIPS*, 2017.
- [5] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville, and Y. Bengio, “Generative adversarial nets,” in *NeurIPS*, 2014.
- [6] J. Pennington, R. Socher, and C. D. Manning, “GloVe: Global vectors for word representation,” in *EMNLP*, 2014.
- [7] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, *et al.*, “Improving language understanding by generative pre-training,” 2018.
- [8] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, *et al.*, “Language models are unsupervised multitask learners,” *OpenAI blog*, 2019.
- [9] 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,” *NeurIPS*, 2020.
- [10] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, “Empirical evaluation of gated recurrent neural networks on sequence modeling,” *arXiv preprint arXiv:1412.3555*, 2014.
- [11] J. Carreira and A. Zisserman, “Quo vadis, action recognition? a new model and the kinetics dataset,” in *CVPR*, 2017.
- [12] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, “High-resolution image synthesis with latent diffusion models,” in *CVPR*, 2022.
- [13] J. L. Ba, J. R. Kiros, and G. E. Hinton, “Layer normalization,” *arXiv preprint arXiv:1607.06450*, 2016.
- [14] T. Afouras, J. S. Chung, and A. Zisserman, “LRS3-TED: A large-scale dataset for visual speech recognition,” *arXiv preprint arXiv:1809.00496*, 2018.

- [15] S. Xie, C. Sun, J. Huang, Z. Tu, and K. Murphy, “Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification,” in *ECCV*, 2018.
- [16] P. Chen, Y. Zhang, M. Tan, H. Xiao, D. Huang, and C. Gan, “Generating visually aligned sound from videos,” *IEEE Transactions on Image Processing*, 2020.
- [17] S. Hershey, S. Chaudhuri, D. P. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold, *et al.*, “CNN architectures for large-scale audio classification,” in *ICASSP*, IEEE, 2017.
- [18] H. Jégou, M. Douze, C. Schmid, and P. Pérez, “Aggregating local descriptors into a compact image representation,” in *CVPR*, IEEE, 2010.
- [19] A. Van Den Oord, O. Vinyals, *et al.*, “Neural discrete representation learning,” *NeurIPS*, 2017.
- [20] J. Redmon and A. Farhadi, “YOLOv3: An incremental improvement,” *arXiv preprint arXiv:1804.02767*, 2018.
- [21] L. Smith and M. Gasser, “The development of embodied cognition: Six lessons from babies,” *Artificial life*, 2005.
- [22] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “ImageNet: A large-scale hierarchical image database,” in *CVPR*, IEEE, 2009.
- [23] A. Krizhevsky, I. Sutskever, and G. Hinton, “Imagenet classification with deep convolutional neural networks,” in *NeurIPS*, 2012.
- [24] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *ICLR*, 2015.
- [25] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in *CVPR*, 2015.
- [26] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, “Large-scale video classification with convolutional neural networks,” in *CVPR*, 2014.
- [27] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Region-based convolutional networks for accurate object detection and segmentation,” *TPAMI*, 2015.

- [28] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. Van Der Smagt, D. Cremers, and T. Brox, “FlowNet: Learning optical flow with convolutional networks,” in *ICCV*, 2015.
- [29] I. Sutskever, O. Vinyals, and Q. V. Le, “Sequence to sequence learning with neural networks,” *NeurIPS*, 2014.
- [30] L. Zhou, Y. Zhou, J. J. Corso, R. Socher, and C. Xiong, “End-to-end dense video captioning with masked transformer,” in *CVPR*, 2018.
- [31] C. Sun, A. Myers, C. Vondrick, K. Murphy, and C. Schmid, “VideoBERT: A joint model for video and language representation learning,” in *ICCV*, 2019.
- [32] H. Tan and M. Bansal, “LXMERT: Learning cross-modality encoder representations from transformers,” in *EMNLP*, 2019.
- [33] J. Lu, D. Batra, D. Parikh, and S. Lee, “VilBERT: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks,” *NeurIPS*, 2019.
- [34] W. Su, X. Zhu, Y. Cao, B. Li, L. Lu, F. Wei, and J. Dai, “VL-BERT: Pre-training of generic visual-linguistic representations,” in *ICLR*, 2020.
- [35] L. Li, Y.-C. Chen, Y. Cheng, Z. Gan, L. Yu, and J. Liu, “HERO: Hierarchical encoder for video+ language omni-representation pre-training,” in *EMNLP*, 2020.
- [36] G. Li, N. Duan, Y. Fang, M. Gong, and D. Jiang, “Unicoder-VL: A universal encoder for vision and language by cross-modal pre-training,” in *AAAI*, 2020.
- [37] A. Kamath, M. Singh, Y. LeCun, G. Synnaeve, I. Misra, and N. Carion, “MDETR-modulated detection for end-to-end multi-modal understanding,” in *ICCV*, 2021.
- [38] M. Ding, Z. Yang, W. Hong, W. Zheng, C. Zhou, D. Yin, J. Lin, X. Zou, Z. Shao, H. Yang, *et al.*, “CogView: Mastering text-to-image generation via transformers,” *NeurIPS*, 2021.
- [39] P. Esser, R. Rombach, and B. Ommer, “Taming transformers for high-resolution image synthesis,” in *CVPR*, 2021.
- [40] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever, “Zero-shot text-to-image generation,” in *ICML*, PMLR, 2021.

- [41] 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,” in *ICLR*, 2021.
- [42] H. Zhao, J. Jia, and V. Koltun, “Exploring self-attention for image recognition,” in *CVPR*, 2020.
- [43] G. Bertasius, H. Wang, and L. Torresani, “Is space-time attention all you need for video understanding?” In *ICML*, 2021.
- [44] Y. Gong, Y.-A. Chung, and J. Glass, “AST: Audio spectrogram transformer,” in *Proceedings of Interspeech*, 2021.
- [45] H. Zhao, L. Jiang, J. Jia, P. H. Torr, and V. Koltun, “Point transformer,” in *ICCV*, 2021.
- [46] A. Jaegle, F. Gimeno, A. Brock, O. Vinyals, A. Zisserman, and J. Carreira, “Perceiver: General perception with iterative attention,” in *ICML*, PMLR, 2021.
- [47] A. Jaegle *et al.*, “Perceiver IO: A general architecture for structured inputs and outputs,” in *ICLR*, 2022.
- [48] A. Nagrani, S. Yang, A. Arnab, A. Jansen, C. Schmid, and C. Sun, “Attention bottlenecks for multimodal fusion,” *NeurIPS*, 2021.
- [49] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, “Robust speech recognition via large-scale weak supervision,” *arXiv preprint arXiv:2212.04356*, 2022.
- [50] R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx, M. S. Bernstein, J. Bohg, A. Bosselut, E. Brunskill, *et al.*, “On the opportunities and risks of foundation models,” *arXiv preprint arXiv:2108.07258*, 2021.
- [51] J.-B. Alayrac, J. Donahue, P. Luc, A. Miech, I. Barr, Y. Hasson, K. Lenc, A. Mensch, K. Millican, M. Reynolds, *et al.*, “Flamingo: A visual language model for few-shot learning,” *NeurIPS*, 2022.
- [52] R. Zheng, J. Chen, M. Ma, and L. Huang, “Fused acoustic and text encoding for multimodal bilingual pretraining and speech translation,” in *ICML*, 2021.

- [53] V. Likhoshesterov, A. Arnab, K. Choromanski, M. Lucic, Y. Tay, A. Weller, and M. Dehghani, “PolyViT: Co-training vision transformers on images, videos and audio,” *arXiv preprint arXiv:2111.12993*, 2021.
- [54] L. Yuan, D. Chen, Y.-L. Chen, N. Codella, X. Dai, J. Gao, H. Hu, X. Huang, B. Li, C. Li, *et al.*, “Florence: A new foundation model for computer vision,” *arXiv preprint arXiv:2111.11432*, 2021.
- [55] T. Baltrušaitis, C. Ahuja, and L.-P. Morency, “Multimodal machine learning: A survey and taxonomy,” *TPAMI*, 2018.
- [56] H. McGurk and J. MacDonald, “Hearing lips and seeing voices,” *Nature*, 1976.
- [57] B. P. Yuhas, M. H. Goldstein, and T. J. Sejnowski, “Integration of acoustic and visual speech signals using neural networks,” *IEEE Communications Magazine*, 1989.
- [58] S. Dupont and J. Luettin, “Audio-visual speech modeling for continuous speech recognition,” *IEEE Transactions on Multimedia*, 2000.
- [59] S. Bengio, “An asynchronous hidden markov model for audio-visual speech recognition,” *NeurIPS*, 2002.
- [60] A. V. Nefian, L. Liang, X. Pi, L. Xiaoxiang, C. Mao, and K. Murphy, “A coupled hmm for audio-visual speech recognition,” in *ICASSP*, IEEE, 2002.
- [61] M. Gurban, J.-P. Thiran, T. Drugman, and T. Dutoit, “Dynamic modality weighting for multi-stream HMMs in audio-visual speech recognition,” in *Proceedings of the 10th International Conference on Multimodal Interfaces*, 2008.
- [62] C. G. Snoek and M. Worring, “Multimodal video indexing: A review of the state-of-the-art,” *Multimedia tools and applications*, 2005.
- [63] G. Evangelopoulos, A. Zlatintsi, G. Skoumas, K. Rapantzikos, A. Potamianos, P. Maragos, and Y. Avrithis, “Video event detection and summarization using audio, visual and text saliency,” in *ICASSP*, IEEE, 2009.
- [64] G. Evangelopoulos, A. Zlatintsi, A. Potamianos, P. Maragos, K. Rapantzikos, G. Skoumas, and Y. Avrithis, “Multimodal saliency and fusion for movie summarization based on aural, visual, and textual attention,” *IEEE Transactions on Multimedia*, 2013.

- [65] J. Huang, Z. Liu, and W. Yao, "Integration of audio and visual information for content-based video segmentation," in *Proceedings of International Conference on Image Processing*, IEEE, 1998.
- [66] J. Nam, M. Alghoniemy, and A. H. Tewfik, "Audio-visual content-based violent scene characterization," in *IEEE International Conference on Image Processing*, IEEE, 1998.
- [67] R. Lienbart, S. Pfeiffer, and W. Effelsberg, "Scene determination based on video and audio features," in *Proceedings IEEE International Conference on Multimedia Computing and Systems*, IEEE, 1999.
- [68] C. Saraceno and R. Leonardi, "Identification of story units in audio-visual sequences by joint audio and video processing," in *Proceedings of International Conference on Image Processing*, IEEE, 1998.
- [69] S. Tsekeridou and I. Pitas, "Speaker dependent video indexing based on audio-visual interaction," in *IEEE International Conference on Image Processing*, 1998.
- [70] Q. H. Z. Liu and A. R. D. G. B. Shahraray, "Automated generation of news content hierarchy by integrating audio, video, and text information," in *ICASSP*, IEEE, 1999.
- [71] J. Huang, Z. Liu, Y. Wang, Y. Chen, and E. K. Wong, "Integration of multimodal features for video scene classification based on hmm," in *IEEE Third Workshop on Multimedia Signal Processing*, IEEE, 1999.
- [72] A. A. Alatan, A. N. Akansu, and W. Wolf, "Multi-modal dialog scene detection using hidden markov models for content-based multimedia indexing," *Multimedia Tools and applications*, 2001.
- [73] H. Naphide and T. S. Huang, "A probabilistic framework for semantic video indexing, filtering, and retrieval," *IEEE Transactions on Multimedia*, 2001.
- [74] S. Renals, S. Bengio, and J. Fiskus, "Machine learning for multimodal interaction," in *3rd Intl. Workshop, MLMI'06. Springer Lecture Notes in Computer Science*, Springer, 2006.
- [75] J. Carletta, S. Ashby, S. Bourban, M. Flynn, M. Guillemot, T. Hain, J. Kadlec, V. Karaiskos, W. Kraaij, M. Kronenthal, *et al.*, "The ami meeting corpus: A pre-announcement," in *International Workshop on Machine Learning for Multimodal Interaction*, Springer, 2005.

- [76] M. Zancanaro, B. Lepri, and F. Pianesi, "Automatic detection of group functional roles in face to face interactions," in *Proceedings of the 8th International Conference on Multimodal interfaces*, 2006.
- [77] D. B. Jayagopi, H. Hung, C. Yeo, and D. Gatica-Perez, "Modeling dominance in group conversations using nonverbal activity cues," *IEEE Transactions on Audio, Speech, and Language Processing*, 2009.
- [78] H. Hung and D. Gatica-Perez, "Estimating cohesion in small groups using audio-visual nonverbal behavior," *IEEE Transactions on Multimedia*, 2010.
- [79] M. Al-Hames, A. Dielmann, D. Gatica-Perez, S. Reiter, S. Renals, G. Rigoll, and D. Zhang, "Multimodal integration for meeting group action segmentation and recognition," in *International Workshop on Machine Learning for Multimodal Interaction*, Springer, 2005.
- [80] M. Al-Hames, B. Hornler, R. Muller, J. Schenk, and G. Rigoll, "Automatic multi-modal meeting camera selection for video-conferences and meeting browsers," in *IEEE International Conference on Multimedia and Expo*, IEEE, 2007.
- [81] P. Ekman, "An argument for basic emotions," *Cognition & emotion*, 1992.
- [82] P. Ekman and W. V. Friesen, *Unmasking the face: A guide to recognizing emotions from facial clues*. Ishk, 2003.
- [83] M. Grimm, K. Kroschel, and S. Narayanan, "The vera am mittag german audio-visual emotional speech database," in *IEEE International Conference on Multimedia and Expo*, IEEE, 2008.
- [84] E. Douglas-Cowie, R. Cowie, C. Cox, N. Amir, and D. Heylen, "The sensitive artificial listener: An induction technique for generating emotionally coloured conversation," in *LREC workshop on corpora for research on emotion and affect*, ELRA Paris, 2008.
- [85] M. Valstar, M. Pantic, *et al.*, "Induced disgust, happiness and surprise: An addition to the MMI facial expression database," in *Proceedings of the 3rd International Workshop on EMOTION: Corpora for Research on Emotion and Affect*, Paris, France., 2010.
- [86] M. Soleymani, J. Lichtenauer, T. Pun, and M. Pantic, "A multimodal database for affect recognition and implicit tagging," *IEEE Transactions on Affective Computing*, 2011.

- [87] G. McKeown, M. F. Valstar, R. Cowie, and M. Pantic, “The semaine corpus of emotionally coloured character interactions,” in *IEEE International Conference on Multimedia and Expo*, IEEE, 2010.
- [88] B. Schuller, M. Valstar, F. Eyben, G. McKeown, R. Cowie, and M. Pantic, “AVEC 2011—the first international audio/visual emotion challenge,” in *International Conference on Affective Computing and Intelligent Interaction*, Springer, 2011.
- [89] I. Kanluan, M. Grimm, and K. Kroschel, “Audio-visual emotion recognition using an emotion space concept,” in *European Signal Processing Conference*, IEEE, 2008.
- [90] M. Glodek, S. Tschechne, G. Layher, M. Schels, T. Brosch, S. Scherer, M. Kächele, M. Schmidt, H. Neumann, G. Palm, *et al.*, “Multiple classifier systems for the classification of audio-visual emotional states,” in *International Conference on Affective Computing and Intelligent Interaction*, Springer, 2011.
- [91] G. A. Ramirez, T. Baltrušaitis, and L.-P. Morency, “Modeling latent discriminative dynamic of multi-dimensional affective signals,” in *International Conference on Affective Computing and Intelligent Interaction*, Springer, 2011.
- [92] M. A. Nicolaou, H. Gunes, and M. Pantic, “Continuous prediction of spontaneous affect from multiple cues and modalities in valence-arousal space,” *IEEE Transactions on Affective Computing*, 2011.
- [93] M. A. Nicolaou, H. Gunes, and M. Pantic, “Audio-visual classification and fusion of spontaneous affective data in likelihood space,” in *International Conference on Pattern Recognition*, IEEE, 2010.
- [94] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural computation*, 1997.
- [95] S. K. D’mello and J. Kory, “A review and meta-analysis of multimodal affect detection systems,” *ACM Computing Surveys (CSUR)*, 2015.
- [96] R. Bernardi, R. Cakici, D. Elliott, A. Erdem, E. Erdem, N. Ikizler-Cinbis, F. Keller, A. Muscat, and B. Plank, “Automatic description generation from images: A survey of models, datasets, and evaluation measures,” *Journal of Artificial Intelligence Research*, 2016.

- [97] S. Bai and S. An, “A survey on automatic image caption generation,” *Neuro-computing*, 2018.
- [98] A. Farhadi, M. Hejrati, M. A. Sadeghi, P. Young, C. Rashtchian, J. Hockenmaier, and D. Forsyth, “Every picture tells a story: Generating sentences from images,” in *ECCV*, Springer, 2010.
- [99] V. Ordonez, G. Kulkarni, and T. Berg, “Im2text: Describing images using 1 million captioned photographs,” in *NeurIPS*, 2011.
- [100] M. Hodosh, P. Young, and J. Hockenmaier, “Framing image description as a ranking task: Data, models and evaluation metrics,” *Journal of Artificial Intelligence Research*, 2013.
- [101] G. Kulkarni, V. Premraj, V. Ordonez, S. Dhar, S. Li, Y. Choi, A. C. Berg, and T. L. Berg, “Babytalk: Understanding and generating simple image descriptions,” *TPAMI*, 2013.
- [102] Y. Yang, C. Teo, H. Daumé III, and Y. Aloimonos, “Corpus-guided sentence generation of natural images,” in *EMNLP*, 2011.
- [103] S. Li, G. Kulkarni, T. Berg, A. Berg, and Y. Choi, “Composing simple image descriptions using web-scale n-grams,” in *Conference on Computational Natural Language Learning*, 2011.
- [104] P. Kuznetsova, V. Ordonez, A. Berg, T. Berg, and Y. Choi, “Collective generation of natural image descriptions,” in *Proceedings of Annual Meeting of the Association for Computational Linguistics*, 2012.
- [105] B. Z. Yao, X. Yang, L. Lin, M. W. Lee, and S.-C. Zhu, “I2T: Image parsing to text description,” *Proceedings of the IEEE*, 2010.
- [106] M. Mitchell, J. Dodge, A. Goyal, K. Yamaguchi, K. Stratos, X. Han, A. Mensch, A. Berg, T. Berg, and H. Daumé III, “Midge: Generating image descriptions from computer vision detections,” in *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics*, 2012.
- [107] D. Marr and H. K. Nishihara, “Representation and recognition of the spatial organization of three-dimensional shapes,” *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 1978.
- [108] D. Hogg, “Model-based vision: A program to see a walking person,” *Image and Vision Computing*, 1983.

- [109] K. Rohr, “Towards model-based recognition of human movements in image sequences,” *CVGIP: Image understanding*, 1994.
- [110] S. Herath, M. Harandi, and F. Porikli, “Going deeper into action recognition: A survey,” *Image and vision computing*, 2017.
- [111] R. Polana and R. Nelson, “Detecting activities,” in *CVPR*, 1993.
- [112] A. F. Bobick and J. W. Davis, “The recognition of human movement using temporal templates,” *TPAMI*, 2001.
- [113] A. Yilmaz and M. Shah, “Actions sketch: A novel action representation,” in *CVPR*, IEEE, 2005.
- [114] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri, “Actions as space-time shapes,” in *ICCV*, IEEE, 2005.
- [115] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie, “Behavior recognition via sparse spatio-temporal features,” in *2005 IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, IEEE, 2005.
- [116] P. Matikainen, M. Hebert, and R. Sukthankar, “Trajectons: Action recognition through the motion analysis of tracked features,” in *International Conference on Computer Vision Workshops*, IEEE, 2009.
- [117] I. Laptev, “On space-time interest points,” *International Journal of Computer Vision*, 2005.
- [118] R. Messing, C. Pal, and H. Kautz, “Activity recognition using the velocity histories of tracked keypoints,” in *ICCV*, IEEE, 2009.
- [119] H. Wang, M. M. Ullah, A. Klaser, I. Laptev, and C. Schmid, “Evaluation of local spatio-temporal features for action recognition,” in *BMVC*, 2009.
- [120] M. Jain, H. Jégou, and P. Bouthemy, “Better exploiting motion for better action recognition,” in *CVPR*, 2013.
- [121] Y. Zhu, X. Zhao, Y. Fu, and Y. Liu, “Sparse coding on local spatial-temporal volumes for human action recognition,” in *Asian Conference on Computer Vision*, Springer, 2010.
- [122] T. Guha and R. K. Ward, “Learning sparse representations for human action recognition,” *TPAMI*, 2011.

- [123] P. V. K. Borges, N. Conci, and A. Cavallaro, "Video-based human behavior understanding: A survey," *IEEE Transactions on Circuits and Systems for Video Technology*, 2013.
- [124] R. W. Lienhart, "Comparison of automatic shot boundary detection algorithms," in *Storage and retrieval for image and video databases VII*, SPIE, 1998.
- [125] A. Hanjalic, "Shot-boundary detection: Unraveled and resolved?" *IEEE Transactions on Circuits and Systems for Video Technology*, 2002.
- [126] N. Vasconcelos and A. Lippman, "Statistical models of video structure for content analysis and characterization," *IEEE Transactions on Image Processing*, 2000.
- [127] W. Zheng, J. Yuan, H. Wang, F. Lin, and B. Zhang, "A novel shot boundary detection framework," in *Visual Communications and Image Processing 2005*, SPIE, 2005.
- [128] J. Yuan, H. Wang, L. Xiao, W. Zheng, J. Li, F. Lin, and B. Zhang, "A formal study of shot boundary detection," *IEEE Transactions on Circuits and Systems for Video Technology*, 2007.
- [129] Z. Cernekova, I. Pitas, and C. Nikou, "Information theory-based shot cut/-fade detection and video summarization," *IEEE Transactions on Circuits and Systems for Video Technology*, 2005.
- [130] J. Baber, N. Afzulpurkar, and S. Satoh, "A framework for video segmentation using global and local features," *International Journal of Pattern Recognition and Artificial Intelligence*, 2013.
- [131] A. Amiri and M. Fathy, "Video shot boundary detection using qr-decomposition and gaussian transition detection," *EURASIP Journal on Advances in Signal Processing*, 2010.
- [132] A. Amiri and M. Fathy, "Video shot boundary detection using generalized eigenvalue decomposition and gaussian transition detection," *Computing and informatics*, 2011.
- [133] Z.-M. Lu and Y. Shi, "Fast video shot boundary detection based on svd and pattern matching," *IEEE Transactions on Image Processing*, 2013.

- [134] S. H. Abdulhussain, A. R. Ramli, M. I. Saripan, B. M. Mahmmud, S. A. R. Al-Haddad, and W. A. Jassim, "Methods and challenges in shot boundary detection: A review," *Entropy*, 2018.
- [135] M. G. Brown, J. T. Foote, G. J. Jones, K. Sparck Jones, and S. J. Young, "Automatic content-based retrieval of broadcast news," in *Proceedings of the third ACM international conference on Multimedia*, 1995.
- [136] J. R. Smith, M. Naphade, and A. Natsev, "Multimedia semantic indexing using model vectors," in *2003 International Conference on Multimedia and Expo*, IEEE, 2003.
- [137] C. G. Snoek, M. Worring, D. C. Koelma, and A. W. Smeulders, "A learned lexicon-driven paradigm for interactive video retrieval," *IEEE Transactions on Multimedia*, 2007.
- [138] C. G. Snoek, M. Worring, *et al.*, "Concept-based video retrieval," *Foundations and Trends® in Information Retrieval*, 2009.
- [139] A. Kojima, T. Tamura, and K. Fukunaga, "Natural language description of human activities from video images based on concept hierarchy of actions," *International Journal of Computer Vision*, 2002.
- [140] M. W. Lee, A. Hakeem, N. Haering, and S.-C. Zhu, "SAVE: A framework for semantic annotation of visual events," in *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, IEEE, 2008.
- [141] M. U. G. Khan, L. Zhang, and Y. Gotoh, "Towards coherent natural language description of video streams," in *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, IEEE, 2011.
- [142] A. Barbu *et al.*, "Video in sentences out," in *Proceedings of the Twenty-Eighth Conference on Uncertainty in Artificial Intelligence*, ser. UAI'12, 2012.
- [143] N. Krishnamoorthy, G. Malkarnenkar, R. Mooney, K. Saenko, and S. Guadarrama, "Generating natural-language video descriptions using text-mined knowledge," in *AAAI*, 2013.
- [144] P. Das, C. Xu, R. F. Doell, and J. J. Corso, "A thousand frames in just a few words: Lingual description of videos through latent topics and sparse object stitching," in *CVPR*, 2013.

- [145] D. G. Lowe, "Object recognition from local scale-invariant features," in *ICCV*, Ieee, 1999.
- [146] J. Sivic and A. Zisserman, "Video google: A text retrieval approach to object matching in videos," in *ICCV*, IEEE, 2003.
- [147] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, 2004.
- [148] T. Giannakopoulos, "PyAudioanalysis: An open-source python library for audio signal analysis," *PloS one*, 2015.
- [149] M. Rohrbach, W. Qiu, I. Titov, S. Thater, M. Pinkal, and B. Schiele, "Translating video content to natural language descriptions," in *ICCV*, 2013.
- [150] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, "Long-term recurrent convolutional networks for visual recognition and description," in *CVPR*, 2015.
- [151] S. Venugopalan, H. Xu, J. Donahue, M. Rohrbach, R. Mooney, and K. Saenko, "Translating videos to natural language using deep recurrent neural networks," in *NAACL*, 2015.
- [152] L. Yao, A. Torabi, K. Cho, N. Ballas, C. Pal, H. Larochelle, and A. Courville, "Describing videos by exploiting temporal structure," in *ICCV*, 2015.
- [153] J. Song, Z. Guo, L. Gao, W. Liu, D. Zhang, and H. T. Shen, "Hierarchical lstm with adjusted temporal attention for video captioning," in *Proceedings of the International Joint Conference on Artificial Intelligence*, AAAI Press, 2017.
- [154] S. Venugopalan, M. Rohrbach, J. Donahue, R. Mooney, T. Darrell, and K. Saenko, "Sequence to sequence-video to text," in *ICCV*, 2015.
- [155] Y. Xu, Y. Han, R. Hong, and Q. Tian, "Sequential video VLAD: Training the aggregation locally and temporally," *IEEE Transactions on Image Processing*, 2018.
- [156] A. Rohrbach, M. Rohrbach, and B. Schiele, "The long-short story of movie description," in *German Conference on Pattern Recognition*, Springer, 2015.
- [157] J. Dong, X. Li, W. Lan, Y. Huo, and C. G. Snoek, "Early embedding and late reranking for video captioning," in *Proceedings of the 24th ACM International Conference on Multimedia*, 2016.

- [158] Y. Pan, T. Yao, H. Li, and T. Mei, “Video captioning with transferred semantic attributes,” in *CVPR*, 2017.
- [159] L. Gao, Z. Guo, H. Zhang, X. Xu, and H. T. Shen, “Video captioning with attention-based lstm and semantic consistency,” *IEEE Transactions on Multimedia*, 2017.
- [160] Z. Gan, C. Gan, X. He, Y. Pu, K. Tran, J. Gao, L. Carin, and L. Deng, “Semantic compositional networks for visual captioning,” in *CVPR*, 2017.
- [161] N. Aafaq, N. Akhtar, W. Liu, S. Z. Gilani, and A. Mian, “Spatio-temporal dynamics and semantic attribute enriched visual encoding for video captioning,” in *CVPR*, 2019.
- [162] B. Wang, L. Ma, W. Zhang, W. Jiang, J. Wang, and W. Liu, “Controllable video captioning with pos sequence guidance based on gated fusion network,” in *ICCV*, 2019.
- [163] J. Hou, X. Wu, W. Zhao, J. Luo, and Y. Jia, “Joint syntax representation learning and visual cue translation for video captioning,” in *ICCV*, 2019.
- [164] Z. Zhang, Z. Qi, C. Yuan, Y. Shan, B. Li, Y. Deng, and W. Hu, “Open-book video captioning with retrieve-copy-generate network,” in *CVPR*, 2021.
- [165] X. Li, B. Zhao, X. Lu, *et al.*, “MAM-RNN: Multi-level attention model based RNN for video captioning,” in *Proceedings of the International Joint Conference on Artificial Intelligence*, 2017.
- [166] C. Yan, Y. Tu, X. Wang, Y. Zhang, X. Hao, Y. Zhang, and Q. Dai, “STAT: Spatial-temporal attention mechanism for video captioning,” *IEEE Transactions on Multimedia*, 2019.
- [167] Z. Yang, Y. Han, and Z. Wang, “Catching the temporal regions-of-interest for video captioning,” in *Proceedings of the 25th ACM International Conference on Multimedia*, 2017.
- [168] C.-Y. Ma, A. Kadav, I. Melvin, Z. Kira, G. AlRegib, and H. P. Graf, “Attend and interact: Higher-order object interactions for video understanding,” in *CVPR*, 2018.
- [169] X. Wu, G. Li, Q. Cao, Q. Ji, and L. Lin, “Interpretable video captioning via trajectory structured localization,” in *CVPR*, 2018.

- [170] J. Zhang and Y. Peng, “Object-aware aggregation with bidirectional temporal graph for video captioning,” in *CVPR*, 2019.
- [171] B. Pan, H. Cai, D.-A. Huang, K.-H. Lee, A. Gaidon, E. Adeli, and J. C. Niebles, “Spatio-temporal graph for video captioning with knowledge distillation,” in *CVPR*, 2020.
- [172] Z. Zhang, Y. Shi, C. Yuan, B. Li, P. Wang, W. Hu, and Z.-J. Zha, “Object relational graph with teacher-recommended learning for video captioning,” in *CVPR*, 2020.
- [173] P. Pan, Z. Xu, Y. Yang, F. Wu, and Y. Zhuang, “Hierarchical recurrent neural encoder for video representation with application to captioning,” in *CVPR*, 2016.
- [174] L. Baraldi, C. Grana, and R. Cucchiara, “Hierarchical boundary-aware neural encoder for video captioning,” in *CVPR*, 2017.
- [175] J. Wang, W. Wang, Y. Huang, L. Wang, and T. Tan, “M3: Multimodal memory modelling for video captioning,” in *CVPR*, 2018.
- [176] W. Pei, J. Zhang, X. Wang, L. Ke, X. Shen, and Y.-W. Tai, “Memory-attended recurrent network for video captioning,” in *CVPR*, 2019.
- [177] Y. Pan, T. Mei, T. Yao, H. Li, and Y. Rui, “Jointly modeling embedding and translation to bridge video and language,” in *CVPR*, 2016.
- [178] B. Wang, L. Ma, W. Zhang, and W. Liu, “Reconstruction network for video captioning,” in *CVPR*, 2018.
- [179] S. Liu, Z. Ren, and J. Yuan, “SibNet: Sibling convolutional encoder for video captioning,” in *Proceedings of the 26th ACM International Conference on Multimedia*, 2018.
- [180] R. Pasunuru and M. Bansal, “Reinforced video captioning with entailment rewards,” in *EMNLP*, Association for Computational Linguistics, 2017.
- [181] X. Wang, W. Chen, J. Wu, Y.-F. Wang, and W. Y. Wang, “Video captioning via hierarchical reinforcement learning,” in *CVPR*, 2018.
- [182] Y. Chen, S. Wang, W. Zhang, and Q. Huang, “Less is more: Picking informative frames for video captioning,” in *ECCV*, 2018.
- [183] R. Vedantam, C. Lawrence Zitnick, and D. Parikh, “CIDEr: Consensus-based image description evaluation,” in *CVPR*, 2015.

- [184] V. Ramanishka, A. Das, D. H. Park, S. Venugopalan, L. A. Hendricks, M. Rohrbach, and K. Saenko, “Multimodal video description,” in *Proceedings of the 24th ACM international conference on Multimedia*, 2016.
- [185] Q. Jin, J. Chen, S. Chen, Y. Xiong, and A. Hauptmann, “Describing videos using multi-modal fusion,” in *Proceedings of the 24th ACM International Conference on Multimedia*, 2016.
- [186] S. Chen, J. Chen, Q. Jin, and A. Hauptmann, “Video captioning with guidance of multimodal latent topics,” in *Proceedings of the 25th ACM International Conference on Multimedia*, 2017.
- [187] J. Xu, T. Yao, Y. Zhang, and T. Mei, “Learning multimodal attention LSTM networks for video captioning,” in *Proceedings of the 25th ACM International Conference on Multimedia*, 2017.
- [188] X. Wang, Y.-F. Wang, and W. Y. Wang, “Watch, listen, and describe: Globally and locally aligned cross-modal attentions for video captioning,” in *NAACL, ACL*, 2018.
- [189] C. Hori, T. Hori, T.-Y. Lee, Z. Zhang, B. Harsham, J. R. Hershey, T. K. Marks, and K. Sumi, “Attention-based multimodal fusion for video description,” in *ICCV*, 2017.
- [190] W. Hao, Z. Zhang, and H. Guan, “Integrating both visual and audio cues for enhanced video caption,” in *AAAI*, 2018.
- [191] M. Chen, Y. Li, Z. Zhang, and S. Huang, “TvT: Two-view transformer network for video captioning,” in *Asian Conference on Machine Learning*, PMLR, 2018.
- [192] Q. Zheng, C. Wang, and D. Tao, “Syntax-aware action targeting for video captioning,” in *CVPR*, 2020.
- [193] K. Lin, L. Li, C.-C. Lin, F. Ahmed, Z. Gan, Z. Liu, Y. Lu, and L. Wang, “SwinBERT: End-to-end transformers with sparse attention for video captioning,” in *CVPR*, 2022.
- [194] D. Chen and W. B. Dolan, “Collecting highly parallel data for paraphrase evaluation,” in *NAACL*, 2011.
- [195] J. Xu, T. Mei, T. Yao, and Y. Rui, “MSR-VTT: A large video description dataset for bridging video and language,” in *CVPR*, 2016.

- [196] X. Wang, J. Wu, J. Chen, L. Li, Y.-F. Wang, and W. Y. Wang, “VATEX: A large-scale, high-quality multilingual dataset for video-and-language research,” in *ICCV*, 2019.
- [197] G. A. Sigurdsson, G. Varol, X. Wang, A. Farhadi, I. Laptev, and A. Gupta, “Hollywood in homes: Crowdsourcing data collection for activity understanding,” in *ECCV*, Springer, 2016.
- [198] R. Krishna, K. Hata, F. Ren, L. Fei-Fei, and J. Carlos Niebles, “Dense-captioning events in videos,” in *ICCV*, 2017.
- [199] L. Zhou, C. Xu, and J. J. Corso, “Towards automatic learning of procedures from web instructional videos,” in *AAAI*, 2018.
- [200] A. Rohrbach, M. Rohrbach, W. Qiu, A. Friedrich, M. Pinkal, and B. Schiele, “Coherent multi-sentence video description with variable level of detail,” in *German Conference on Pattern Recognition*, Springer, 2014.
- [201] H. Yu, J. Wang, Z. Huang, Y. Yang, and W. Xu, “Video paragraph captioning using hierarchical recurrent neural networks,” in *CVPR*, 2016.
- [202] S. Gella, M. Lewis, and M. Rohrbach, “A dataset for telling the stories of social media videos,” in *EMNLP*, 2018.
- [203] Y. Xiong, B. Dai, and D. Lin, “Move forward and tell: A progressive generator of video descriptions,” in *ECCV*, 2018.
- [204] Y. Song, S. Chen, and Q. Jin, “Towards diverse paragraph captioning for untrimmed videos,” in *CVPR*, 2021.
- [205] J. S. Park, M. Rohrbach, T. Darrell, and A. Rohrbach, “Adversarial inference for multi-sentence video description,” in *CVPR*, 2019.
- [206] L. Zhou, Y. Kalantidis, X. Chen, J. J. Corso, and M. Rohrbach, “Grounded video description,” in *CVPR*, 2019.
- [207] J. Lei, L. Wang, Y. Shen, D. Yu, T. Berg, and M. Bansal, “MART: Memory-augmented recurrent transformer for coherent video paragraph captioning,” in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, 2020.
- [208] T. Wang, R. Zhang, Z. Lu, F. Zheng, R. Cheng, and P. Luo, “End-to-end dense video captioning with parallel decoding,” in *ICCV*, 2021.

- [209] M. Ren, R. Kiros, and R. Zemel, “Exploring models and data for image question answering,” *NeurIPS*, 2015.
- [210] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh, “VQA: Visual question answering,” in *ICCV*, 2015.
- [211] Y. Goyal, T. Khot, D. Summers-Stay, D. Batra, and D. Parikh, “Making the V in VQA matter: Elevating the role of image understanding in visual question answering,” in *CVPR*, 2017.
- [212] D. Gurari, Q. Li, A. J. Stangl, A. Guo, C. Lin, K. Grauman, J. Luo, and J. P. Bigam, “VizWiz grand challenge: Answering visual questions from blind people,” in *CVPR*, 2018.
- [213] J. Johnson, B. Hariharan, L. Van Der Maaten, L. Fei-Fei, C. Lawrence Zitnick, and R. Girshick, “CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning,” in *CVPR*, 2017.
- [214] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang, “Bottom-up and top-down attention for image captioning and visual question answering,” in *CVPR*, 2018.
- [215] T.-Y. Lin, A. RoyChowdhury, and S. Maji, “Bilinear cnn models for fine-grained visual recognition,” in *ICCV*, 2015.
- [216] A. Fukui, D. H. Park, D. Yang, A. Rohrbach, T. Darrell, and M. Rohrbach, “Multimodal compact bilinear pooling for visual question answering and visual grounding,” *arXiv preprint arXiv:1606.01847*, 2016.
- [217] J.-H. Kim, J. Jun, and B.-T. Zhang, “Bilinear attention networks,” *NeurIPS*, 2018.
- [218] L. R. Tucker, “Some mathematical notes on three-mode factor analysis,” *Psychometrika*, 1966.
- [219] H. Ben-Younes, R. Cadene, M. Cord, and N. Thome, “MUTAN: Multimodal tucker fusion for visual question answering,” in *ICCV*, 2017.
- [220] H. Ben-Younes, R. Cadene, N. Thome, and M. Cord, “BLOCK: Bilinear superdiagonal fusion for visual question answering and visual relationship detection,” in *AAAI*, 2019.
- [221] R. Cadene, H. Ben-Younes, M. Cord, and N. Thome, “MUREL: Multimodal relational reasoning for visual question answering,” in *CVPR*, 2019.

- [222] J. Andreas, M. Rohrbach, T. Darrell, and D. Klein, “Neural module networks,” in *CVPR*, 2016.
- [223] R. Hu, J. Andreas, M. Rohrbach, T. Darrell, and K. Saenko, “Learning to reason: End-to-end module networks for visual question answering,” in *ICCV*, 2017.
- [224] J. Johnson, B. Hariharan, L. Van Der Maaten, J. Hoffman, L. Fei-Fei, C. Lawrence Zitnick, and R. Girshick, “Inferring and executing programs for visual reasoning,” in *ICCV*, 2017.
- [225] W. Norcliffe-Brown, S. Vafeias, and S. Parisot, “Learning conditioned graph structures for interpretable visual question answering,” *NeurIPS*, 2018.
- [226] L. Li, Z. Gan, Y. Cheng, and J. Liu, “Relation-aware graph attention network for visual question answering,” in *CVPR*, 2019.
- [227] Y. Jiang, V. Natarajan, X. Chen, M. Rohrbach, D. Batra, and D. Parikh, “Pythia v0.1: The winning entry to the VQA challenge 2018,” *arXiv preprint arXiv:1807.09956*, 2018.
- [228] D. Teney, P. Anderson, X. He, and A. Van Den Hengel, “Tips and tricks for visual question answering: Learnings from the 2017 challenge,” in *CVPR*, 2018.
- [229] A. Singh, V. Natarajan, M. Shah, Y. Jiang, X. Chen, D. Batra, D. Parikh, and M. Rohrbach, “Towards VQA models that can read,” in *CVPR*, 2019.
- [230] H. Jiang, I. Misra, M. Rohrbach, E. Learned-Miller, and X. Chen, “In defense of grid features for visual question answering,” in *CVPR*, 2020.
- [231] Z. Yu, J. Yu, Y. Cui, D. Tao, and Q. Tian, “Deep modular co-attention networks for visual question answering,” in *CVPR*, 2019.
- [232] H. Zhong, J. Chen, C. Shen, H. Zhang, J. Huang, and X.-S. Hua, “Self-adaptive neural module transformer for visual question answering,” *IEEE Transactions on Multimedia*, 2020.
- [233] Y. Zhou, T. Ren, C. Zhu, X. Sun, J. Liu, X. Ding, M. Xu, and R. Ji, “TRAR: Routing the attention spans in transformer for visual question answering,” in *ICCV*, 2021.

- [234] M. Tapaswi, Y. Zhu, R. Stiefelhagen, A. Torralba, R. Urtasun, and S. Fidler, “MovieQA: Understanding stories in movies through question-answering,” in *CVPR*, 2016.
- [235] Y. Jang, Y. Song, Y. Yu, Y. Kim, and G. Kim, “TGIF-QA: Toward spatio-temporal reasoning in visual question answering,” in *CVPR*, 2017.
- [236] D. Xu, Z. Zhao, J. Xiao, F. Wu, H. Zhang, X. He, and Y. Zhuang, “Video question answering via gradually refined attention over appearance and motion,” in *ACM International Conference on Multimedia*, 2017.
- [237] J. Lei, L. Yu, M. Bansal, and T. L. Berg, “TVQA: Localized, compositional video question answering,” in *EMNLP*, 2018.
- [238] J. Gao, R. Ge, K. Chen, and R. Nevatia, “Motion-appearance co-memory networks for video question answering,” in *CVPR*, 2018.
- [239] C. Fan, X. Zhang, S. Zhang, W. Wang, C. Zhang, and H. Huang, “Heterogeneous memory enhanced multimodal attention model for video question answering,” in *CVPR*, 2019.
- [240] Y. Jang, Y. Song, C. D. Kim, Y. Yu, Y. Kim, and G. Kim, “Video question answering with spatio-temporal reasoning,” *International Journal of Computer Vision*, 2019.
- [241] J. Jiang, Z. Chen, H. Lin, X. Zhao, and Y. Gao, “Divide and conquer: Question-guided spatio-temporal contextual attention for video question answering,” in *AAAI*, 2020.
- [242] S. Xiao, L. Chen, K. Gao, Z. Wang, Y. Yang, and J. Xiao, “Rethinking multi-modal alignment in video question answering from feature and sample perspectives,” *arXiv preprint arXiv:2204.11544*, 2022.
- [243] J. Mun, P. Hongsuck Seo, I. Jung, and B. Han, “MarioQA: Answering questions by watching gameplay videos,” in *ICCV*, 2017.
- [244] D. Huang, P. Chen, R. Zeng, Q. Du, M. Tan, and C. Gan, “Location-aware graph convolutional networks for video question answering,” in *AAAI*, 2020.
- [245] T. M. Le, V. Le, S. Venkatesh, and T. Tran, “Hierarchical conditional relation networks for video question answering,” in *CVPR*, 2020.
- [246] P. Jiang and Y. Han, “Reasoning with heterogeneous graph alignment for video question answering,” in *AAAI*, 2020.

- [247] J. Park, J. Lee, and K. Sohn, “Bridge to answer: Structure-aware graph interaction network for video question answering,” in *CVPR*, 2021.
- [248] F. Liu, J. Liu, W. Wang, and H. Lu, “HAIR: Hierarchical visual-semantic relational reasoning for video question answering,” in *ICCV*, 2021.
- [249] J. Wang, B.-K. Bao, and C. Xu, “DualVGR: A dual-visual graph reasoning unit for video question answering,” *IEEE Transactions on Multimedia*, 2021.
- [250] A. Seo, G.-C. Kang, J. Park, and B.-T. Zhang, “Attend what you need: Motion-appearance synergistic networks for video question answering,” in *International Joint Conference on Natural Language Processing (ACL)*, 2021.
- [251] J. Xiao, A. Yao, Z. Liu, Y. Li, W. Ji, and T.-S. Chua, “Video as conditional graph hierarchy for multi-granular question answering,” in *AAAI*, 2022.
- [252] H. Yun, Y. Yu, W. Yang, K. Lee, and G. Kim, “Pano-AVQA: Grounded audio-visual question answering on 360deg videos,” in *ICCV*, 2021.
- [253] A. Yang, A. Miech, J. Sivic, I. Laptev, and C. Schmid, “Just ask: Learning to answer questions from millions of narrated videos,” in *ICCV*, 2021.
- [254] P. H. Seo, A. Nagrani, and C. Schmid, “Look before you speak: Visually contextualized utterances,” in *CVPR*, 2021.
- [255] W. Yu, H. Zheng, M. Li, L. Ji, L. Wu, N. Xiao, and N. Duan, “Learning from inside: Self-driven siamese sampling and reasoning for video question answering,” *NeurIPS*, 2021.
- [256] Y. Zhong, W. Ji, J. Xiao, Y. Li, W. Deng, and T.-S. Chua, “Video question answering: Datasets, algorithms and challenges,” *arXiv preprint arXiv:2203.01225*, 2022.
- [257] A. Yang, A. Miech, J. Sivic, I. Laptev, and C. Schmid, “Zero-shot video question answering via frozen bidirectional language models,” 2022.
- [258] A. Zadeh, M. Chan, P. P. Liang, E. Tong, and L.-P. Morency, “Social-IQ: A question answering benchmark for artificial social intelligence,” in *CVPR*, 2019.
- [259] K. Yi, C. Gan, Y. Li, P. Kohli, J. Wu, A. Torralba, and J. B. Tenenbaum, “CLEVRER: CoLLision Events for Video REpresentation and Reasoning,” *ICLR*, 2020.

- [260] N. Garcia, M. Otani, C. Chu, and Y. Nakashima, “KnowIT VQA: Answering knowledge-based questions about videos,” in *AAAI*, 2020.
- [261] M. Grunde-McLaughlin, R. Krishna, and M. Agrawala, “AGQA: A benchmark for compositional spatio-temporal reasoning,” in *CVPR*, 2021.
- [262] J. Xiao, X. Shang, A. Yao, and T.-S. Chua, “Next-QA: Next phase of question-answering to explaining temporal actions,” in *CVPR*, 2021.
- [263] L. Zhu, Z. Xu, Y. Yang, and A. G. Hauptmann, “Uncovering the temporal context for video question answering,” *International Journal of Computer Vision*, 2017.
- [264] K.-M. Kim, M.-O. Heo, S.-H. Choi, and B.-T. Zhang, “DeepStory: Video story QA by deep embedded memory networks,” in *Proceedings of the International Joint Conference on Artificial Intelligence*, AAAI Press, 2017.
- [265] T. Maharaj, N. Ballas, A. Rohrbach, A. Courville, and C. Pal, “A dataset and exploration of models for understanding video data through fill-in-the-blank question-answering,” in *CVPR*, 2017.
- [266] Z. Yu, D. Xu, J. Yu, T. Yu, Z. Zhao, Y. Zhuang, and D. Tao, “ActivityNet-QA: A dataset for understanding complex web videos via question answering,” in *AAAI*, 2019.
- [267] H. Alamri, V. Cartillier, A. Das, J. Wang, A. Cherian, I. Essa, D. Batra, T. K. Marks, C. Hori, P. Anderson, *et al.*, “Audio visual scene-aware dialog,” in *CVPR*, 2019.
- [268] G. Li, Y. Wei, Y. Tian, C. Xu, J.-R. Wen, and D. Hu, “Learning to answer questions in dynamic audio-visual scenarios,” in *CVPR*, 2022.
- [269] I. Schwartz, A. G. Schwing, and T. Hazan, “A simple baseline for audio-visual scene-aware dialog,” in *CVPR*, 2019.
- [270] Y. Li, Y. Song, L. Cao, J. Tetreault, L. Goldberg, A. Jaimes, and J. Luo, “TGIF: A new dataset and benchmark on animated gif description,” in *CVPR*, 2016.
- [271] T. Mikolov, K. Chen, G. S. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” in *ICLR*, 2013.

- [272] R. Xu, C. Xiong, W. Chen, and J. Corso, “Jointly modeling deep video and compositional text to bridge vision and language in a unified framework,” in *AAAI*, 2015.
- [273] L. Anne Hendricks, O. Wang, E. Shechtman, J. Sivic, T. Darrell, and B. Russell, “Localizing moments in video with natural language,” in *ICCV*, 2017.
- [274] Y. Yu, H. Ko, J. Choi, and G. Kim, “End-to-end concept word detection for video captioning, retrieval, and question answering,” in *CVPR*, 2017.
- [275] Y. Yu, J. Kim, and G. Kim, “A joint sequence fusion model for video question answering and retrieval,” in *ECCV*, 2018.
- [276] J. Dong, X. Li, C. Xu, S. Ji, Y. He, G. Yang, and X. Wang, “Dual encoding for zero-example video retrieval,” in *CVPR*, 2019.
- [277] S. Chen, Y. Zhao, Q. Jin, and Q. Wu, “Fine-grained video-text retrieval with hierarchical graph reasoning,” in *CVPR*, 2020.
- [278] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, “NetVLAD: CNN architecture for weakly supervised place recognition,” in *CVPR*, 2016.
- [279] A. Miech, I. Laptev, and J. Sivic, “Learning a text-video embedding from incomplete and heterogeneous data,” *arXiv preprint arXiv:1804.02516*, 2018.
- [280] H. Fan and Y. Yang, “Person tube retrieval via language description,” in *AAAI*, 2020.
- [281] J. Dong, X. Li, and C. G. Snoek, “Word2VisualVec: Image and video to sentence matching by visual feature prediction,” *arXiv preprint arXiv:1604.06838*, 2016.
- [282] R. Kiros, Y. Zhu, R. R. Salakhutdinov, R. Zemel, R. Urtasun, A. Torralba, and S. Fidler, “Skip-thought vectors,” *NeurIPS*, 2015.
- [283] M. Otani, Y. Nakashima, E. Rahtu, J. Heikkilä, and N. Yokoya, “Learning joint representations of videos and sentences with web image search,” in *European Conference on Computer Vision Workshops*, Springer, 2016.
- [284] N. C. Mithun, J. Li, F. Metze, and A. K. Roy-Chowdhury, “Learning joint embedding with multimodal cues for cross-modal video-text retrieval,” in *Proceedings of the ACM on International Conference on Multimedia Retrieval*, 2018.
- [285] J. Chen, H. Hu, H. Wu, Y. Jiang, and C. Wang, “Learning the best pooling strategy for visual semantic embedding,” in *CVPR*, 2021.

- [286] M. Zolfaghari, Y. Zhu, P. Gehler, and T. Brox, “CrossCLR: Cross-modal contrastive learning for multi-modal video representations,” in *ICCV*, 2021.
- [287] K. Wen, J. Xia, Y. Huang, L. Li, J. Xu, and J. Shao, “Cookie: Contrastive cross-modal knowledge sharing pre-training for vision-language representation,” in *ICCV*, 2021.
- [288] V. Gabeur, C. Sun, K. Alahari, and C. Schmid, “Multi-modal transformer for video retrieval,” in *ECCV*, Springer, 2020.
- [289] X. Wang, L. Zhu, and Y. Yang, “T2VLAD: Global-local sequence alignment for text-video retrieval,” in *CVPR*, 2021.
- [290] X. Song, J. Chen, Z. Wu, and Y.-G. Jiang, “Spatial-temporal graphs for cross-modal text2video retrieval,” *IEEE Transactions on Multimedia*, 2021.
- [291] Y. Liu, S. Albanie, A. Nagrani, and A. Zisserman, “Use what you have: Video retrieval using representations from collaborative experts,” *arXiv preprint arXiv:1907.13487*, 2019.
- [292] S. Ging, M. Zolfaghari, H. Pirsiavash, and T. Brox, “COOT: Cooperative hierarchical transformer for video-text representation learning,” *NeurIPS*, 2020.
- [293] M. Dzabraev, M. Kalashnikov, S. Komkov, and A. Petiushko, “MDMMT: Multidomain multimodal transformer for video retrieval,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2021.
- [294] A. v. d. Oord, Y. Li, and O. Vinyals, “Representation learning with contrastive predictive coding,” *arXiv preprint arXiv:1807.03748*, 2018.
- [295] X. Wang, L. Zhu, Z. Zheng, M. Xu, and Y. Yang, “Align and tell: Boosting text-video retrieval with local alignment and fine-grained supervision,” *IEEE Transactions on Multimedia*, 2022.
- [296] Y. Liu, P. Xiong, L. Xu, S. Cao, and Q. Jin, “TS2-Net: Token shift and selection transformer for text-video retrieval,” in *ECCV*, Springer, 2022.
- [297] B. Zhang, H. Hu, and F. Sha, “Cross-modal and hierarchical modeling of video and text,” in *ECCV*, 2018.
- [298] D. Dwibedi, Y. Aytar, J. Tompson, P. Sermanet, and A. Zisserman, “Temporal cycle-consistency learning,” in *CVPR*, 2019.

- [299] A. Rohrbach, A. Torabi, M. Rohrbach, N. Tandon, C. Pal, H. Larochelle, A. Courville, and B. Schiele, “Movie description,” *International Journal of Computer Vision*, 2017.
- [300] S. Li, T. Xiao, H. Li, B. Zhou, D. Yue, and X. Wang, “Person search with natural language description,” in *CVPR*, 2017.
- [301] J. Lei, T. L. Berg, and M. Bansal, “Detecting moments and highlights in videos via natural language queries,” *NeurIPS*, 2021.
- [302] A. Miech, D. Zhukov, J.-B. Alayrac, M. Tapaswi, I. Laptev, and J. Sivic, “HowTo100M: Learning a Text-Video Embedding by Watching Hundred Million Narrated Video Clips,” in *ICCV*, 2019.
- [303] A. Miech, J.-B. Alayrac, I. Laptev, J. Sivic, and A. Zisserman, “Thinking fast and slow: Efficient text-to-visual retrieval with transformers,” in *CVPR*, 2021.
- [304] J. Yang, Y. Bisk, and J. Gao, “TACO: Token-aware cascade contrastive learning for video-text alignment,” in *ICCV*, 2021.
- [305] Y.-B. Lin, J. Lei, M. Bansal, and G. Bertasius, “ECLIPSE: Efficient long-range video retrieval using sight and sound,” in *ECCV*, 2022.
- [306] J. Gao, C. Sun, Z. Yang, and R. Nevatia, “TALL: Temporal activity localization via language query,” in *ICCV*, 2017.
- [307] M. Liu, X. Wang, L. Nie, Q. Tian, B. Chen, and T.-S. Chua, “Cross-modal moment localization in videos,” in *Proceedings of the 26th ACM International Conference on Multimedia*, 2018.
- [308] L. A. Hendricks, O. Wang, E. Shechtman, J. Sivic, T. Darrell, and B. Russell, “Localizing moments in video with temporal language,” in *EMNLP, ACL*, 2018.
- [309] M. Liu, X. Wang, L. Nie, X. He, B. Chen, and T.-S. Chua, “Attentive moment retrieval in videos,” in *ACM SIGIR Conference on Research & Development in Information Retrieval*, 2018.
- [310] A. Wu and Y. Han, “Multi-modal circulant fusion for video-to-language and backward,” in *Proceedings of the International Joint Conference on Artificial Intelligence*, 2018.

- [311] J. Chen, X. Chen, L. Ma, Z. Jie, and T.-S. Chua, “Temporally grounding natural sentence in video,” in *EMNLP*, 2018.
- [312] J. Chen, L. Ma, X. Chen, Z. Jie, and J. Luo, “Localizing natural language in videos,” in *AAAI*, 2019.
- [313] H. Xu, K. He, B. A. Plummer, L. Sigal, S. Sclaroff, and K. Saenko, “Multi-level language and vision integration for text-to-clip retrieval,” in *AAAI*, 2019.
- [314] H. Xu, A. Das, and K. Saenko, “R-C3D: Region convolutional 3D network for temporal activity detection,” in *ICCV*, 2017.
- [315] Y. Yuan, L. Ma, J. Wang, W. Liu, and W. Zhu, “Semantic conditioned dynamic modulation for temporal sentence grounding in videos,” *NeurIPS*, 2019.
- [316] H. Wang, Z.-J. Zha, X. Chen, Z. Xiong, and J. Luo, “Dual path interaction network for video moment localization,” in *Proceedings of the 28th ACM International Conference on Multimedia*, 2020.
- [317] H. Wang, Z.-J. Zha, L. Li, D. Liu, and J. Luo, “Structured multi-level interaction network for video moment localization via language query,” in *CVPR*, 2021.
- [318] J. Gao and C. Xu, “Fast video moment retrieval,” in *ICCV*, 2021.
- [319] D. Zhang, X. Dai, X. Wang, Y.-F. Wang, and L. S. Davis, “MAN: Moment alignment network for natural language moment retrieval via iterative graph adjustment,” in *CVPR*, 2019.
- [320] M. Welling and T. N. Kipf, “Semi-supervised classification with graph convolutional networks,” in *ICLR*, 2016.
- [321] R. Ge, J. Gao, K. Chen, and R. Nevatia, “MAC: Mining activity concepts for language-based temporal localization,” in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, IEEE, 2019.
- [322] N. C. Mithun, S. Paul, and A. K. Roy-Chowdhury, “Weakly supervised video moment retrieval from text queries,” in *CVPR*, 2019.
- [323] S. Zhang, H. Peng, J. Fu, and J. Luo, “Learning 2D temporal adjacent networks for moment localization with natural language,” in *AAAI*, 2020.
- [324] Y. Zeng, D. Cao, X. Wei, M. Liu, Z. Zhao, and Z. Qin, “Multi-modal relational graph for cross-modal video moment retrieval,” in *CVPR*, 2021.

- [325] X. Lan, Y. Yuan, X. Wang, Z. Wang, and W. Zhu, “A survey on temporal sentence grounding in videos,” *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 2021.
- [326] Y. Yuan, T. Mei, and W. Zhu, “To find where you talk: Temporal sentence localization in video with attention based location regression,” in *AAAI*, 2019.
- [327] S. Chen and Y.-G. Jiang, “Semantic proposal for activity localization in videos via sentence query,” in *AAAI*, 2019.
- [328] S. Ghosh, A. Agarwal, Z. Parekh, and A. Hauptmann, “ExCL: Extractive Clip Localization Using Natural Language Descriptions,” in *NAACL, ACL*, 2019.
- [329] H. Zhang, A. Sun, W. Jing, and J. T. Zhou, “Span-based localizing network for natural language video localization,” in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online: ACL, 2020.
- [330] D. He, X. Zhao, J. Huang, F. Li, X. Liu, and S. Wen, “Read, watch, and move: Reinforcement learning for temporally grounding natural language descriptions in videos,” in *AAAI*, 2019.
- [331] W. Wang, Y. Huang, and L. Wang, “Language-driven temporal activity localization: A semantic matching reinforcement learning model,” in *CVPR*, 2019.
- [332] J. Wu, G. Li, X. Han, and L. Lin, “Reinforcement learning for weakly supervised temporal grounding of natural language in untrimmed videos,” in *Proceedings of the 28th ACM International Conference on Multimedia*, 2020.
- [333] D. Cao, Y. Zeng, M. Liu, X. He, M. Wang, and Z. Qin, “STRONG: Spatio-temporal reinforcement learning for cross-modal video moment localization,” in *Proceedings of the 28th ACM International Conference on Multimedia*, 2020.
- [334] C. Lu, L. Chen, C. Tan, X. Li, and J. Xiao, “DEBUG: A dense bottom-up grounding approach for natural language video localization,” in *EMNLP*, 2019.
- [335] R. Zeng, H. Xu, W. Huang, P. Chen, M. Tan, and C. Gan, “Dense regression network for video grounding,” in *CVPR*, 2020.
- [336] J. Mun, M. Cho, and B. Han, “Local-global video-text interactions for temporal grounding,” in *CVPR*, 2020.

- [337] S. Chen and Y.-G. Jiang, “Hierarchical visual-textual graph for temporal activity localization via language,” in *ECCV*, Springer, 2020.
- [338] S. Chen, W. Jiang, W. Liu, and Y.-G. Jiang, “Learning modality interaction for temporal sentence localization and event captioning in videos,” in *ECCV*, Springer, 2020.
- [339] X. Yang, F. Feng, W. Ji, M. Wang, and T.-S. Chua, “Deconfounded video moment retrieval with causal intervention,” in *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2021.
- [340] G. Nan, R. Qiao, Y. Xiao, J. Liu, S. Leng, H. Zhang, and W. Lu, “Interventional video grounding with dual contrastive learning,” in *CVPR*, 2021.
- [341] Y. Zhao, Z. Zhao, Z. Zhang, and Z. Lin, “Cascaded prediction network via segment tree for temporal video grounding,” in *CVPR*, 2021.
- [342] S. Zhang, J. Su, and J. Luo, “Exploiting temporal relationships in video moment localization with natural language,” in *Proceedings of the 27th ACM International Conference on Multimedia*, 2019.
- [343] Z. Zhang, Z. Lin, Z. Zhao, and Z. Xiao, “Cross-modal interaction networks for query-based moment retrieval in videos,” in *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2019.
- [344] J. Wang, L. Ma, and W. Jiang, “Temporally grounding language queries in videos by contextual boundary-aware prediction,” in *AAAI*, 2020.
- [345] Z. Lin, Z. Zhao, Z. Zhang, Q. Wang, and H. Liu, “Weakly-supervised video moment retrieval via semantic completion network,” in *AAAI*, 2020.
- [346] D. Liu, X. Qu, X.-Y. Liu, J. Dong, P. Zhou, and Z. Xu, “Jointly cross-and self-modal graph attention network for query-based moment localization,” in *Proceedings of the 28th ACM International Conference on Multimedia*, 2020.
- [347] X. Qu, P. Tang, Z. Zou, Y. Cheng, J. Dong, P. Zhou, and Z. Xu, “Fine-grained iterative attention network for temporal language localization in videos,” in *Proceedings of the 28th ACM International Conference on Multimedia*, 2020.

- [348] X. Yu, M. Malmir, X. He, J. Chen, T. Wang, Y. Wu, Y. Liu, and Y. Liu, “Cross interaction network for natural language guided video moment retrieval,” in *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2021.
- [349] Y. Wang, W. Zhou, and H. Li, “Fine-grained semantic alignment network for weakly supervised temporal language grounding,” in *Findings of the Association for Computational Linguistics: EMNLP 2021*, Association for Computational Linguistics, 2021.
- [350] J. Huang, Y. Liu, S. Gong, and H. Jin, “Cross-sentence temporal and semantic relations in video activity localisation,” in *ICCV*, 2021.
- [351] M. Zhang, Y. Yang, X. Chen, Y. Ji, X. Xu, J. Li, and H. T. Shen, “Multi-stage aggregated transformer network for temporal language localization in videos,” in *CVPR*, 2021.
- [352] D. Liu, X. Qu, J. Dong, P. Zhou, Y. Cheng, W. Wei, Z. Xu, and Y. Xie, “Context-aware biaffine localizing network for temporal sentence grounding,” in *CVPR*, 2021.
- [353] T. Dozat and C. D. Manning, “Deep biaffine attention for neural dependency parsing,” in *ICLR*, 2016.
- [354] Z. Wang, J. Chen, and Y.-G. Jiang, “Visual co-occurrence alignment learning for weakly-supervised video moment retrieval,” in *Proceedings of the 29th ACM International Conference on Multimedia*, 2021.
- [355] N. Srivastava, E. Mansimov, and R. Salakhudinov, “Unsupervised learning of video representations using LSTMs,” in *ICML*, PMLR, 2015.
- [356] C. Nash, J. Carreira, J. Walker, I. Barr, A. Jaegle, M. Malinowski, and P. Battaglia, “Transframer: Arbitrary frame prediction with generative models,” *arXiv preprint arXiv:2203.09494*, 2022.
- [357] W. Harvey, S. Naderiparizi, V. Masrani, C. Weilbach, and F. Wood, “Flexible diffusion modeling of long videos,” *arXiv preprint arXiv:2205.11495*, 2022.
- [358] J. Ho, T. Salimans, A. A. Gritsenko, W. Chan, M. Norouzi, and D. J. Fleet, “Video diffusion models,” in *NeurIPS*, 2022.

- [359] R. Yang, P. Srivastava, and S. Mandt, “Diffusion probabilistic modeling for video generation,” *arXiv preprint arXiv:2203.09481*, 2022.
- [360] T. Höppe, A. Mehrjou, S. Bauer, D. Nielsen, and A. Dittadi, “Diffusion models for video prediction and infilling,” *Transactions on Machine Learning Research*, 2022.
- [361] Y. He, T. Yang, Y. Zhang, Y. Shan, and Q. Chen, “Latent video diffusion models for high-fidelity video generation with arbitrary lengths,” *arXiv preprint arXiv:2211.13221*, 2022.
- [362] C. Finn, I. Goodfellow, and S. Levine, “Unsupervised learning for physical interaction through video prediction,” *NeurIPS*, 2016.
- [363] N. Kalchbrenner, A. Oord, K. Simonyan, I. Danihelka, O. Vinyals, A. Graves, and K. Kavukcuoglu, “Video pixel networks,” in *ICML*, PMLR, 2017.
- [364] M. Babaeizadeh, C. Finn, D. Erhan, R. H. Campbell, and S. Levine, “Stochastic variational video prediction,” in *ICLR*, 2018.
- [365] E. Denton and R. Fergus, “Stochastic video generation with a learned prior,” in *ICML*, PMLR, 2018.
- [366] R. Villegas, A. Pathak, H. Kannan, D. Erhan, Q. V. Le, and H. Lee, “High fidelity video prediction with large stochastic recurrent neural networks,” *NeurIPS*, 2019.
- [367] D. Weissenborn, O. Täckström, and J. Uszkoreit, “Scaling autoregressive video models,” in *ICLR*, 2020.
- [368] W. Yan, Y. Zhang, P. Abbeel, and A. Srinivas, “VideoGPT: Video generation using vq-vae and transformers,” *arXiv preprint arXiv:2104.10157*, 2021.
- [369] V. Voleti, A. Jolicoeur-Martineau, and C. Pal, “MCVD - masked conditional video diffusion for prediction, generation, and interpolation,” in *NeurIPS*, 2022.
- [370] C. Vondrick, H. Pirsiavash, and A. Torralba, “Generating videos with scene dynamics,” *NeurIPS*, 2016.
- [371] M. Saito, E. Matsumoto, and S. Saito, “Temporal generative adversarial nets with singular value clipping,” in *ICCV*, 2017.
- [372] S. Tulyakov, M.-Y. Liu, X. Yang, and J. Kautz, “MoCoGAN: Decomposing motion and content for video generation,” in *CVPR*, 2018.

- [373] M. Saito, S. Saito, M. Koyama, and S. Kobayashi, “Train sparsely, generate densely: Memory-efficient unsupervised training of high-resolution temporal GAN,” *International Journal of Computer Vision*, 2020.
- [374] S. Yu, J. Tack, S. Mo, H. Kim, J. Kim, J.-W. Ha, and J. Shin, “Generating videos with dynamics-aware implicit generative adversarial networks,” in *ICLR*, 2022.
- [375] C. Schuldt, I. Laptev, and B. Caputo, “Recognizing human actions: A local SVM approach,” in *Proceedings of the 17th International Conference on Pattern Recognition*, IEEE, 2004.
- [376] T. Gupta, D. Schwenk, A. Farhadi, D. Hoiem, and A. Kembhavi, “Imagine this! scripts to compositions to videos,” in *ECCV*, 2018.
- [377] G. Mittal, T. Marwah, and V. N. Balasubramanian, “Sync-DRAW: Automatic video generation using deep recurrent attentive architectures,” in *Proceedings of the 25th ACM International Conference on Multimedia*, 2017.
- [378] Y. Li, M. Min, D. Shen, D. Carlson, and L. Carin, “Video generation from text,” in *AAAI*, 2018.
- [379] Y. Pan, Z. Qiu, T. Yao, H. Li, and T. Mei, “To create what you tell: Generating videos from captions,” in *Proceedings of the 25th ACM International Conference on Multimedia*, 2017.
- [380] Y. Liu, X. Wang, Y. Yuan, and W. Zhu, “Cross-modal dual learning for sentence-to-video generation,” in *Proceedings of the 27th ACM International Conference on Multimedia*, 2019.
- [381] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, “Unpaired image-to-image translation using cycle-consistent adversarial networks,” in *ICCV*, 2017.
- [382] C. Wu, L. Huang, Q. Zhang, B. Li, L. Ji, F. Yang, G. Sapiro, and N. Duan, “GODIVA: Generating open-domain videos from natural descriptions,” *arXiv preprint arXiv:2104.14806*, 2021.
- [383] C. Wu, J. Liang, L. Ji, F. Yang, Y. Fang, D. Jiang, and N. Duan, “NÜWA: Visual synthesis pre-training for neural visual world creation,” in *ECCV*, Springer, 2022.

- [384] W. Hong, M. Ding, W. Zheng, X. Liu, and J. Tang, “CogVideo: Large-scale pretraining for text-to-video generation via transformers,” *arXiv preprint arXiv:2205.15868*, 2022.
- [385] M. Ding, W. Zheng, W. Hong, and J. Tang, “CogView2: Faster and better text-to-image generation via hierarchical transformers,” in *NeurIPS*, 2022.
- [386] R. Villegas, M. Babaeizadeh, P.-J. Kindermans, H. Moraldo, H. Zhang, M. T. Saffar, S. Castro, J. Kunze, and D. Erhan, “Phenaki: Variable length video generation from open domain textual description,” *arXiv preprint arXiv:2210.02399*, 2022.
- [387] S. Ge, T. Hayes, H. Yang, X. Yin, G. Pang, D. Jacobs, J.-B. Huang, and D. Parikh, “Long video generation with time-agnostic vqgan and time-sensitive transformer,” in *ECCV*, 2022.
- [388] P. Dhariwal and A. Nichol, “Diffusion models beat gans on image synthesis,” *NeurIPS*, 2021.
- [389] A. Nichol, P. Dhariwal, A. Ramesh, P. Shyam, P. Mishkin, B. McGrew, I. Sutskever, and M. Chen, “GLIDE: Towards photorealistic image generation and editing with text-guided diffusion models,” *arXiv preprint arXiv:2112.10741*, 2021.
- [390] 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.
- [391] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional networks for biomedical image segmentation,” in *International Conference on Medical Image Computing and Computer-assisted Intervention*, Springer, 2015.
- [392] J. Ho and T. Salimans, “Classifier-free diffusion guidance,” in *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021.
- [393] U. Singer, A. Polyak, T. Hayes, X. Yin, J. An, S. Zhang, Q. Hu, H. Yang, O. Ashual, O. Gafni, *et al.*, “Make-a-video: Text-to-video generation without text-video data,” *arXiv preprint arXiv:2209.14792*, 2022.
- [394] J. Ho, W. Chan, C. Saharia, J. Whang, R. Gao, A. Gritsenko, D. P. Kingma, B. Poole, M. Norouzi, D. J. Fleet, *et al.*, “Imagen video: High definition video generation with diffusion models,” *arXiv preprint arXiv:2210.02303*, 2022.

- [395] M. Bain, A. Nagrani, G. Varol, and A. Zisserman, “Frozen in time: A joint video and image encoder for end-to-end retrieval,” in *ICCV*, 2021.
- [396] T. Hayes, S. Zhang, X. Yin, G. Pang, S. Sheng, H. Yang, S. Ge, Q. Hu, and D. Parikh, “MUGEN: A playground for video-audio-text multimodal understanding and generation,” in *ECCV*, Springer, 2022.
- [397] N. Neverova, C. Wolf, G. Taylor, and F. Nebout, “ModDrop: Adaptive multi-modal gesture recognition,” *TPAMI*, 2015.
- [398] X. Yang, P. Ramesh, R. Chitta, S. Madhvanath, E. A. Bernal, and J. Luo, “Deep multimodal representation learning from temporal data,” in *CVPR*, 2017.
- [399] Y. Bian, C. Gan, X. Liu, F. Li, X. Long, Y. Li, H. Qi, J. Zhou, S. Wen, and Y. Lin, “Revisiting the effectiveness of off-the-shelf temporal modeling approaches for large-scale video classification,” *arXiv preprint arXiv:1708.03805*, 2017.
- [400] X. Long, C. Gan, G. De Melo, J. Wu, X. Liu, and S. Wen, “Attention clusters: Purely attention based local feature integration for video classification,” in *CVPR*, 2018.
- [401] B. Korbar, D. Tran, and L. Torresani, “Cooperative learning of audio and video models from self-supervised synchronization,” *NeurIPS*, 2018.
- [402] A. Owens and A. A. Efros, “Audio-visual scene analysis with self-supervised multisensory features,” in *ECCV*, 2018.
- [403] E. Kazakos, A. Nagrani, A. Zisserman, and D. Damen, “EPIC-fusion: Audio-visual temporal binding for egocentric action recognition,” in *ICCV*, 2019.
- [404] W. Wang, D. Tran, and M. Feiszli, “What makes training multi-modal classification networks hard?” In *CVPR*, 2020.
- [405] F. Xiao, Y. J. Lee, K. Grauman, J. Malik, and C. Feichtenhofer, “Audiovisual SlowFast networks for video recognition,” *arXiv preprint arXiv:2001.08740*, 2020.
- [406] Z. Wu, Y.-G. Jiang, X. Wang, H. Ye, and X. Xue, “Multi-stream multi-class fusion of deep networks for video classification,” in *ACM International Conference on Multimedia*, 2016.

- [407] Y.-G. Jiang, Z. Wu, J. Wang, X. Xue, and S.-F. Chang, “Exploiting feature and class relationships in video categorization with regularized deep neural networks,” *TPAMI*, 2017.
- [408] X. Long, C. Gan, G. Melo, X. Liu, Y. Li, F. Li, and S. Wen, “Multimodal keyless attention fusion for video classification,” in *AAAI*, 2018.
- [409] H. R. V. Joze, A. Shaban, M. L. Iuzzolino, and K. Koishida, “MMTM: Multimodal transfer module for cnn fusion,” in *CVPR*, 2020.
- [410] E. Kazakos, J. Huh, A. Nagrani, A. Zisserman, and D. Damen, “With a little help from my temporal context: Multimodal egocentric action recognition,” in *BMVC*.
- [411] M. M. Islam and T. Iqbal, “Multi-GAT: A graphical attention-based hierarchical multimodal representation learning approach for human activity recognition,” *Robotics and Automation Letters*, 2021.
- [412] M. M. Islam and T. Iqbal, “MuMu: Cooperative multitask learning-based guided multimodal fusion,” in *AAAI*, 2022.
- [413] J. Chen and C. M. Ho, “MM-ViT: Multi-modal video transformer for compressed video action recognition,” in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2022.
- [414] J.-M. Pérez-Rúa, V. Vielzeuf, S. Pateux, M. Baccouche, and F. Jurie, “Mfas: Multimodal fusion architecture search,” in *CVPR*, 2019.
- [415] X. Xiong, A. Arnab, A. Nagrani, and C. Schmid, “M²m mix: A multimodal multiview transformer ensemble,” *arXiv preprint arXiv:2206.09852*, 2022.
- [416] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, “HMDB: A large video database for human motion recognition,” in *ICCV*, IEEE, 2011.
- [417] K. Soomro, A. R. Zamir, and M. Shah, “UCF101: A dataset of 101 human actions classes from videos in the wild,” *arXiv preprint arXiv:1212.0402*, 2012.
- [418] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, *et al.*, “The kinetics human action video dataset,” *arXiv preprint arXiv:1705.06950*, 2017.

- [419] D. Damen, H. Doughty, G. M. Farinella, S. Fidler, A. Furnari, E. Kazakos, D. Moltisanti, J. Munro, T. Perrett, W. Price, *et al.*, “Scaling egocentric vision: The EPIC-KITCHENS dataset,” in *ECCV*, 2018.
- [420] D. Damen, H. Doughty, G. M. Farinella, A. Furnari, E. Kazakos, J. Ma, D. Moltisanti, J. Munro, T. Perrett, W. Price, *et al.*, “Rescaling egocentric vision: Collection, pipeline and challenges for EPIC-KITCHENS-100,” *International Journal of Computer Vision*, 2022.
- [421] E. Amrani, R. Ben-Ari, D. Rotman, and A. Bronstein, “Noise estimation using density estimation for self-supervised multimodal learning,” in *AAAI*, 2021.
- [422] Z. Tang, J. Lei, and M. Bansal, “DeCEMBERT: Learning from noisy instructional videos via dense captions and entropy minimization,” in *NAACL*, 2021.
- [423] T. Han, W. Xie, and A. Zisserman, “Temporal alignment networks for long-term video,” in *CVPR*, 2022.
- [424] H. Luo, L. Ji, M. Zhong, Y. Chen, W. Lei, N. Duan, and T. Li, “CLIP4Clip: An empirical study of CLIP for end to end video clip retrieval and captioning,” *Neurocomputing*, 2022.
- [425] X. Zhu, J. Zhu, H. Li, X. Wu, H. Li, X. Wang, and J. Dai, “Uni-Perceiver: Pre-training unified architecture for generic perception for zero-shot and few-shot tasks,” in *CVPR*, 2022.
- [426] M. Cao, T. Yang, J. Weng, C. Zhang, J. Wang, and Y. Zou, “LocVTP: Video-text pre-training for temporal localization,” in *ECCV*, 2022.
- [427] J. Wang, D. Chen, Z. Wu, C. Luo, L. Zhou, Y. Zhao, Y. Xie, C. Liu, Y.-G. Jiang, and L. Yuan, “OmniVL: One foundation model for image-language and video-language tasks,” *arXiv preprint arXiv:2209.07526*, 2022.
- [428] B. Ni, H. Peng, M. Chen, S. Zhang, G. Meng, J. Fu, S. Xiang, and H. Ling, “Expanding language-image pretrained models for general video recognition,” *arXiv preprint arXiv:2208.02816*, 2022.
- [429] Y. Ge, Y. Ge, X. Liu, D. Li, Y. Shan, X. Qie, and P. Luo, “Bridging video-text retrieval with multiple choice questions,” in *CVPR*, 2022.

- [430] H. Lu, N. Fei, Y. Huo, Y. Gao, Z. Lu, and J.-R. Wen, “COTS: Collaborative two-stream vision-language pre-training model for cross-modal retrieval,” in *CVPR*, 2022.
- [431] L. Zhu and Y. Yang, “ActBERT: Learning global-local video-text representations,” in *CVPR*, 2020.
- [432] J. Lei, T. L. Berg, and M. Bansal, “Revealing single frame bias for video-and-language learning,” *arXiv preprint arXiv:2206.03428*, 2022.
- [433] Y. Lin, C. Wei, H. Wang, A. Yuille, and C. Xie, “SMAUG: Sparse masked autoencoder for efficient video-language pre-training,” *arXiv preprint arXiv:2211.11446*, 2022.
- [434] A. Miech, J.-B. Alayrac, L. Smaira, I. Laptev, J. Sivic, and A. Zisserman, “End-to-end learning of visual representations from uncurated instructional videos,” in *CVPR*, 2020.
- [435] M. Patrick, Y. M. Asano, P. Kuznetsova, R. Fong, J. F. Henriques, G. Zweig, and A. Vedaldi, “On compositions of transformations in contrastive self-supervised learning,” in *ICCV*, 2021.
- [436] R. Zellers, X. Lu, J. Hessel, Y. Yu, J. S. Park, J. Cao, A. Farhadi, and Y. Choi, “MERLOT: Multimodal neural script knowledge models,” *NeurIPS*, 2021.
- [437] J. Lei, L. Li, L. Zhou, Z. Gan, T. L. Berg, M. Bansal, and J. Liu, “Less is more: ClipBERT for video-and-language learning via sparse sampling,” in *CVPR*, 2021.
- [438] M. Patrick, P.-Y. Huang, Y. Asano, F. Metze, A. G. Hauptmann, J. F. Henriques, and A. Vedaldi, “Support-set bottlenecks for video-text representation learning,” in *ICLR*, 2021.
- [439] T.-J. Fu, L. Li, Z. Gan, K. Lin, W. Y. Wang, L. Wang, and Z. Liu, “VIOLET: End-to-end video-language transformers with masked visual-token modeling,” *arXiv preprint arXiv:2111.12681*, 2021.
- [440] H. Fang, P. Xiong, L. Xu, and Y. Chen, “CLIP2Video: Mastering video-text retrieval via image clip,” *arXiv preprint arXiv:2106.11097*, 2021.

- [441] J.-B. Alayrac, A. Recasens, R. Schneider, R. Arandjelović, J. Ramapuram, J. De Fauw, L. Smaira, S. Dieleman, and A. Zisserman, “Self-Supervised MultiModal Versatile Networks,” *NeurIPS*, 2020.
- [442] H. Akbari, L. Yuan, R. Qian, W.-H. Chuang, S.-F. Chang, Y. Cui, and B. Gong, “VATT: Transformers for multimodal self-supervised learning from raw video, audio and text,” *NeurIPS*, 2021.
- [443] R. Zellers, J. Lu, X. Lu, Y. Yu, Y. Zhao, M. Salehi, A. Kusupati, J. Hessel, A. Farhadi, and Y. Choi, “MERLOT RESERVE: Neural script knowledge through vision and language and sound,” in *CVPR*, 2022.
- [444] H. Mittal, P. Morgado, U. Jain, and A. Gupta, “Learning state-aware visual representations from audible interactions,” *arXiv preprint arXiv:2209.13583*, 2022.
- [445] H. Xu, G. Ghosh, P.-Y. Huang, D. Okhonko, A. Aghajanyan, F. Metze, L. Zettlemoyer, and C. Feichtenhofer, “VideoCLIP: Contrastive pre-training for zero-shot video-text understanding,” in *EMNLP*, 2021.
- [446] L. Li, J. Lei, Z. Gan, L. Yu, Y.-C. Chen, R. Pillai, Y. Cheng, L. Zhou, X. E. Wang, W. Y. Wang, *et al.*, “VALUE: A multi-task benchmark for video-and-language understanding evaluation,” in *Conference on Neural Information Processing Systems Track on Datasets and Benchmarks*, 2021.
- [447] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, “Swin transformer: Hierarchical vision transformer using shifted windows,” in *ICCV*, 2021.
- [448] Y. Jiang, S. Chang, and Z. Wang, “TransGAN: Two pure transformers can make one strong gan, and that can scale up,” *NeurIPS*, 2021.
- [449] A. Arnab, M. Dehghani, G. Heigold, C. Sun, M. Lučić, and C. Schmid, “ViViT: A video vision transformer,” in *ICCV*, 2021.
- [450] L. Chen, K. Lu, A. Rajeswaran, K. Lee, A. Grover, M. Laskin, P. Abbeel, A. Srinivas, and I. Mordatch, “Decision transformer: Reinforcement learning via sequence modeling,” *NeurIPS*, 2021.
- [451] M. Schuster and K. Nakajima, “Japanese and korean voice search,” in *ICASSP*, IEEE, 2012.

- [452] R. Sennrich, B. Haddow, and A. Birch, “Neural machine translation of rare words with subword units,” in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, 2016.
- [453] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *CVPR*, 2016.
- [454] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the inception architecture for computer vision,” in *CVPR*, 2016.
- [455] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A simple way to prevent neural networks from overfitting,” *The Journal of Machine Learning Research*, 2014.
- [456] P. Dhariwal, H. Jun, C. Payne, J. W. Kim, A. Radford, and I. Sutskever, “Jukebox: A generative model for music,” *arXiv preprint arXiv:2005.00341*, 2020.
- [457] J. Johnson, A. Karpathy, and L. Fei-Fei, “Densecap: Fully convolutional localization networks for dense captioning,” in *CVPR*, 2016.
- [458] V. Escorcia, F. Caba Heilbron, J. C. Niebles, and B. Ghanem, “DAPs: Deep action proposals for action understanding,” in *ECCV*, Springer, 2016.
- [459] J. Wang, W. Jiang, L. Ma, W. Liu, and Y. Xu, “Bidirectional attentive fusion with context gating for dense video captioning,” in *CVPR*, 2018.
- [460] S. Buch, V. Escorcia, C. Shen, B. Ghanem, and J. Carlos Niebles, “SST: Single-stream temporal action proposals,” in *CVPR*, 2017.
- [461] X. Duan, W. Huang, C. Gan, J. Wang, W. Zhu, and J. Huang, “Weakly supervised dense event captioning in videos,” *NeurIPS*, 2018.
- [462] S. J. Rennie, E. Marcheret, Y. Mroueh, J. Ross, and V. Goel, “Self-critical sequence training for image captioning,” in *CVPR*, 2017.
- [463] Y. Li, T. Yao, Y. Pan, H. Chao, and T. Mei, “Jointly localizing and describing events for dense video captioning,” in *CVPR*, 2018.
- [464] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, “SSD: Single shot multibox detector,” in *ECCV*, Springer, 2016.
- [465] Y. Zhao, Y. Xiong, L. Wang, Z. Wu, X. Tang, and D. Lin, “Temporal action detection with structured segment networks,” in *ICCV*, 2017.

- [466] J. Mun, L. Yang, Z. Ren, N. Xu, and B. Han, “Streamlined dense video captioning,” in *CVPR*, 2019.
- [467] O. Vinyals, M. Fortunato, and N. Jaitly, “Pointer networks,” *NeurIPS*, 2015.
- [468] T. Rahman, B. Xu, and L. Sigal, “Watch, listen and tell: Multi-modal weakly supervised dense event captioning,” in *ICCV*, 2019.
- [469] B. Shi, L. Ji, Y. Liang, N. Duan, P. Chen, Z. Niu, and M. Zhou, “Dense procedure captioning in narrated instructional videos,” in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019.
- [470] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, “Learning spatiotemporal features with 3D convolutional networks,” in *ICCV*, 2015.
- [471] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards real-time object detection with region proposal networks,” *NeurIPS*, 2015.
- [472] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, “BLEU: A method for automatic evaluation of machine translation,” in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, 2002.
- [473] S. Banerjee and A. Lavie, “METEOR: An automatic metric for mt evaluation with improved correlation with human judgments,” in *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, 2005.
- [474] T. Wang, H. Zheng, M. Yu, Q. Tian, and H. Hu, “Event-centric hierarchical representation for dense video captioning,” *IEEE Transactions on Circuits and Systems for Video Technology*, 2020.
- [475] M. Suin and A. Rajagopalan, “An efficient framework for dense video captioning,” in *AAAI*, 2020.
- [476] S. Chen and Y.-G. Jiang, “Towards bridging event captioner and sentence localizer for weakly supervised dense event captioning,” in *CVPR*, 2021.
- [477] C. Deng, S. Chen, D. Chen, Y. He, and Q. Wu, “Sketch, ground, and refine: Top-down dense video captioning,” in *CVPR*, 2021.
- [478] W. Choi, J. Chen, and J. Yoon, “Parallel pathway dense video captioning with deformable transformer,” *IEEE Access*, 2022.
- [479] A. Owens, P. Isola, J. McDermott, A. Torralba, E. H. Adelson, and W. T. Freeman, “Visually indicated sounds,” in *CVPR*, 2016.

- [480] L. Chen, S. Srivastava, Z. Duan, and C. Xu, “Deep cross-modal audio-visual generation,” in *Proceedings of the on Thematic Workshops of ACM Multimedia*, 2017.
- [481] B. Li, X. Liu, K. Dinesh, Z. Duan, and G. Sharma, “Creating a multitrack classical music performance dataset for multimodal music analysis: Challenges, insights, and applications,” *IEEE Transactions on Multimedia*, 2018.
- [482] W. Hao, Z. Zhang, and H. Guan, “CMCGAN: A uniform framework for cross-modal visual-audio mutual generation,” in *AAAI*, 2018.
- [483] Y. Zhou, Z. Wang, C. Fang, T. Bui, and T. L. Berg, “Visual to sound: Generating natural sound for videos in the wild,” in *CVPR*, 2018.
- [484] H. Tan, G. Wu, P. Zhao, and Y. Chen, “Spectrogram analysis via self-attention for realizing cross-model visual-audio generation,” in *ICASSP*, IEEE, 2020.
- [485] K. Su, X. Liu, and E. Shlizerman, “Audeo: Audio generation for a silent performance video,” *Advances in Neural Information Processing Systems*, 2020.
- [486] V. K. Kurmi, V. Bajaj, B. N. Patro, K. Venkatesh, V. P. Namboodiri, and P. Jyothi, “Collaborative learning to generate audio-video jointly,” in *ICASSP*, IEEE, 2021.
- [487] K. Chen, C. Zhang, C. Fang, Z. Wang, T. Bui, and R. Nevatia, “Visually indicated sound generation by perceptually optimized classification,” in *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, 2018.
- [488] 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 *ICASSP*, IEEE, 2017.
- [489] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, “GANs trained by a two time-scale update rule converge to a local Nash equilibrium,” *NeurIPS*, 2017.
- [490] K. Kilgour, M. Zuluaga, D. Roblek, and M. Sharifi, “Fr\`echet audio distance: A metric for evaluating music enhancement algorithms,” *arXiv preprint arXiv:1812.08466*, 2018.

- [491] M. Bińkowski, J. Donahue, S. Dieleman, A. Clark, E. Elsen, N. Casagrande, L. C. Cobo, and K. Simonyan, “High fidelity speech synthesis with adversarial networks,” in *ICLR*, 2020.
- [492] P. Manocha, A. Finkelstein, R. Zhang, N. J. Bryan, G. J. Mysore, and Z. Jin, “A differentiable perceptual audio metric learned from just noticeable differences,” in *Proceedings of Interspeech*, 2020.
- [493] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, “Image-to-image translation with conditional adversarial networks,” in *CVPR*, 2017.
- [494] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, “The unreasonable effectiveness of deep features as a perceptual metric,” in *CVPR*, 2018.
- [495] H. Chen, W. Xie, A. Vedaldi, and A. Zisserman, “VGGSound: A large-scale audio-visual dataset,” in *ICASSP, IEEE*, 2020.
- [496] M. Chen, A. Radford, R. Child, J. Wu, H. Jun, D. Luan, and I. Sutskever, “Generative pretraining from pixels,” in *ICML, PMLR*, 2020.
- [497] D. Griffin and J. Lim, “Signal estimation from modified short-time fourier transform,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 1984.
- [498] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, “WaveNet: A Generative Model for Raw Audio,” in *Proc. 9th ISCA Workshop on Speech Synthesis Workshop (SSW 9)*, 2016.
- [499] K. Kumar, R. Kumar, T. de Boissiere, L. Gestin, W. Z. Teoh, J. Sotelo, A. de Brébisson, Y. Bengio, and A. C. Courville, “MelGAN: Generative adversarial networks for conditional waveform synthesis,” *NeurIPS*, 2019.
- [500] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, “Improved techniques for training gans,” *NeurIPS*, 2016.
- [501] S. Ghose and J. J. Prevost, “FoleyGAN: Visually guided generative adversarial network-based synchronous sound generation in silent videos,” *IEEE Transactions on Multimedia*, 2022.
- [502] K. Choi, S. Oh, M. Kang, and B. McFee, “A proposal for foley sound synthesis challenge,” *arXiv preprint arXiv:2207.10760*, 2022.
- [503] R. Sheffer and Y. Adi, “I hear your true colors: Image guided audio generation,” *arXiv preprint arXiv:2211.03089*, 2022.

- [504] A. Razavi, A. Van den Oord, and O. Vinyals, “Generating diverse high-fidelity images with VQ-VAE-2,” *NeurIPS*, 2019.
- [505] C. Cui, Y. Ren, J. Liu, R. Huang, and Z. Zhao, “Varietysound: Timbre-controllable video to sound generation via unsupervised information disentanglement,” *arXiv preprint arXiv:2211.10666*, 2022.
- [506] S. Lee, J. Chung, Y. Yu, G. Kim, T. Breuel, G. Chechik, and Y. Song, “ACAV100M: Automatic curation of large-scale datasets for audio-visual video representation learning,” in *ICCV*, 2021.
- [507] 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,” *arXiv preprint arXiv:2211.06687*, 2022.
- [508] J. Hershey and J. Movellan, “Audio vision: Using audio-visual synchrony to locate sounds,” *NeurIPS*, 1999.
- [509] M. Slaney and M. Covell, “FaceSync: A linear operator for measuring synchronization of video facial images and audio tracks,” *NeurIPS*, 2000.
- [510] J. S. Chung and A. Zisserman, “Out of time: Automated lip sync in the wild,” in *Asian Conference on Computer Vision*, Springer, 2017.
- [511] S.-W. Chung, J. S. Chung, and H.-G. Kang, “Perfect match: Improved cross-modal embeddings for audio-visual synchronisation,” in *ICASSP*, IEEE, 2019.
- [512] T. Halperin, A. Ephrat, and S. Peleg, “Dynamic temporal alignment of speech to lips,” in *ICASSP*, IEEE, 2019.
- [513] L. Rabiner and B.-H. Juang, *Fundamentals of speech recognition*. Prentice-Hall, Inc., 1993.
- [514] N. Khosravan, S. Ardeshtir, and R. Puri, “On attention modules for audio-visual synchronization.,” in *CVPR Workshops*, 2019.
- [515] Y. J. Kim, H. S. Heo, S.-W. Chung, and B.-J. Lee, “End-to-end lip synchronisation based on pattern classification,” in *2021 IEEE Spoken Language Technology Workshop (SLT)*, IEEE, 2021.
- [516] V. S. Kadandale, J. F. Montesinos, and G. Haro, “VocaLiST: An audio-visual synchronisation model for lips and voices,” *arXiv preprint arXiv:2204.02090*, 2022.

- [517] A. Llagostera Casanovas and A. Cavallaro, “Audio-visual events for multi-camera synchronization,” *Multimedia Tools and Applications*, 2015.
- [518] H. Chen, W. Xie, T. Afouras, A. Nagrani, A. Vedaldi, and A. Zisserman, “Audio-visual synchronisation in the wild,” in *BMVC*, 2021.
- [519] C. Doersch, A. Gupta, and A. A. Efros, “Unsupervised visual representation learning by context prediction,” in *ICCV*, 2015.
- [520] D. Wei, J. J. Lim, A. Zisserman, and W. T. Freeman, “Learning and using the arrow of time,” in *CVPR*, 2018.
- [521] R. Arandjelovic and A. Zisserman, “Objects that sound,” in *ECCV*, 2018.
- [522] I. Radiocommunication, *Relative timing of sound and vision for broadcasting (bt.1359-1)*, 1998.

PUBLICATIONS

PUBLICATION

|

Multi-modal dense video captioning

V. Iashin and E. Rahtu

*In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition
Workshops, IEEE, 2020, pp. 958–959*

Publication reprinted with the permission of the copyright holders.

Multi-modal Dense Video Captioning

Vladimir Iashin
Tampere University

vladimir.iashin@tuni.fi

Esa Rahtu
Tampere University

esa.rahtu@tuni.fi

Abstract

Dense video captioning is a task of localizing interesting events from an untrimmed video and producing textual description (captions) for each localized event. Most of the previous works in dense video captioning are solely based on visual information and completely ignore the audio track. However, audio, and speech, in particular, are vital cues for a human observer in understanding an environment. In this paper, we present a new dense video captioning approach that is able to utilize any number of modalities for event description. Specifically, we show how audio and speech modalities may improve a dense video captioning model. We apply automatic speech recognition (ASR) system to obtain a temporally aligned textual description of the speech (similar to subtitles) and treat it as a separate input alongside video frames and the corresponding audio track. We formulate the captioning task as a machine translation problem and utilize recently proposed Transformer architecture to convert multi-modal input data into textual descriptions. We demonstrate the performance of our model on ActivityNet Captions dataset. The ablation studies indicate a considerable contribution from audio and speech components suggesting that these modalities contain substantial complementary information to video frames. Furthermore, we provide an in-depth analysis of the ActivityNet Caption results by leveraging the category tags obtained from original YouTube videos. Code is publicly available: github.com/v-iashin/MDVC.

1. Introduction

The substantial amount of freely available video material has brought up the need for automatic methods to summarize and compactly represent the essential content. One approach would be to produce a short video skim containing the most important video segments as proposed in the *video summarization* task [25]. Alternatively, the video content could be described using natural language sentences. Such an approach can lead to a very compact and intuitive representation and is typically referred to as video captioning

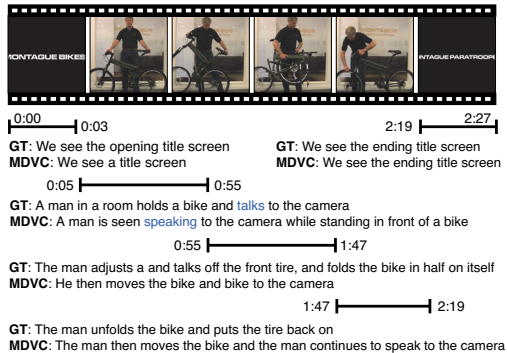


Figure 1. Example video with ground truth captions and predictions of Multi-modal Dense Video Captioning module (MDVC). It may account for any number of modalities, *i.e.* audio or speech.

in the literature [58]. However, producing a single description for an entire video might be impractical for long unconstrained footage. Instead, *dense video captioning* [24] aims, first, at temporally localizing events and, then, at producing natural language description for each of them. Fig. 1 illustrates dense video captions for an example video sequence.

Most recent works in dense video captioning formulate the captioning problem as a machine translation task, where the input is a set of features extracted from the video stream and the output is a natural language sentence. Thus, the captioning methods can be leveraged by recent developments in machine translation field, such as *Transformer* model [45]. The main idea in the transformer is to utilise *self-attention* mechanism to model long-term dependencies in a sequence. We follow the recent work [59] and adopt the transformer architecture in our dense video captioning model.

The vast majority of previous works are generating captions purely based on visual information [59, 48, 26, 28, 53, 30, 54]. However, almost all videos include an audio track, which could provide vital cues for video understanding. In particular, what is being said by people in the video, might make a crucial difference to the content description. For in-

stance, in a scene when someone knocks the door from an opposite side, we only see the door but the audio helps us to understand that somebody is behind it and wants to enter. Therefore, it is impossible for a model to make a useful caption for it. Also, other types of videos as instruction videos, sport videos, or video lectures could be challenging for a captioning model.

In contrast, we build our model to utilize video frames, raw audio signal, and the speech content in the caption generation process. To this end, we deploy *automatic speech recognition* (ASR) system [1] to extract time-aligned captions of *what is being said* (similar to subtitles) and employ it alongside with video and audio representations in the transformer model.

The proposed model is assessed using the challenging ActivityNet Captions [24] benchmark dataset, where we obtain competitive results to the current state-of-the-art. The subsequent ablation studies indicate a substantial contribution from audio and speech signals. Moreover, we retrieve and perform breakdown analysis by utilizing previously unused video category tags provided with the original YouTube videos [2]. The program code of our model and the evaluation approach will be made publicly available.

2. Related Work

2.1. Video Captioning

Early works in video captioning applied *rule-based models* [22, 31, 7], where the idea was to identify a set of video objects and use them to fill predefined templates to generate a sentence. Later, the need for sentence templates was omitted by casting the captioning problem as a machine translation task [37]. Following the success of neural models in translation systems [42], similar methods became widely popular in video captioning [57, 46, 47, 58, 5, 38, 18, 9, 52]. The rationale behind this approach is to train two *Recurrent Neural Networks* (RNNs) in an *encoder-decoder* fashion. Specifically, an encoder inputs a set of video features, accumulates its *hidden state*, which is passed to a decoder for producing a caption.

To further improve the performance of the captioning model, several methods have been proposed, including shared memory between visual and textual domains [49, 34], spatial and temporal attention [56], reinforcement learning [50], semantic tags [11, 32], other modalities [55, 19, 51, 13], and by producing a paragraph instead of one sentence [36, 58].

2.2. Dense Video Captioning

Inspired by the idea of the *dense image captioning* task [20], Krishna *et al.* [24] introduced a problem of dense video captioning and released a new dataset called ActivityNet Captions which leveraged the research in the field

[59, 48, 26, 28, 53, 30, 35, 54]. In particular, [48] adopted the idea of the context-awareness [24] and generalized the temporal event proposal module to utilize both past and future contexts as well as an *attentive fusion* to differentiate captions from highly overlapping events. Meanwhile, the concept of *Single Shot Detector* (SSD) [27] was also used to generate event proposals and *reward maximization* for better captioning in [26].

In order to mitigate the intrinsic difficulties of RNNs to model long-term dependencies in a sequence, Zhou *et al.* [59] tailored the recent idea of Transformer [45] for dense video captioning. In [28] the authors noticed that the captioning may benefit from interactions between objects in a video and developed recurrent higher-order interaction module to model these interactions. Xiong *et al.* [53] noticed that many previous models produced redundant captions, and proposed to generate captions in a progressive manner, conditioned on the previous caption while applying paragraph- and sentence-level rewards. Similarly, a “bird-view” correction and two-level reward maximization for a more coherent story-telling have been employed in [30].

Since the human annotation of a video with temporal boundaries and captions for each of them can be laborious, several attempts have been made to address this issue [10, 29]. Specifically, [10] employed the idea of *cycle-consistency* to translate a set of captions to a set of temporal events without any paired annotation, while [29] automatically-collected dataset of an unparalleled-scale exploiting the structure of instructional videos.

The most similar work to our captioning model is [59] that also utilizes a version of the Transformer [45] architecture. However, their model is designed solely for visual features. Instead, we believe that dense video captioning may benefit from information from other modalities.

2.3. Multi-modal Dense Video Captioning

A few attempts has been made to include additional cues like audio and speech [35, 16, 39] for dense video captioning task. Rahman *et al.* [35] utilized the idea of *cycle-consistency* [10] to build a model with visual and audio inputs. However, due to weak supervision, the system did not reach high performance. Hessel *et al.* [16] and Shi *et al.* [39] employ a transformer architecture [45] to encode both video frames and speech segments to generate captions for instructional (cooking) videos. Yet, the high results on a dataset which is restricted to instructional video appear to be not evidential as the speech and the captions are already very close to each other in such videos [29].

In contrast to the mentioned multi-modal dense video captioning methods: (1) we present the importance of the speech and audio modalities on a domain-free dataset, (2) propose a multi-modal dense video captioning module (MDVC) which can be scaled to any number of modalities.

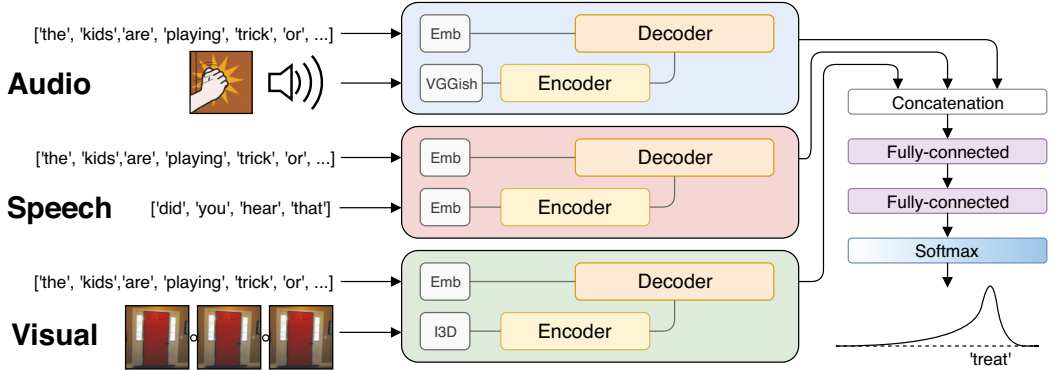


Figure 2. The proposed Multi-modal Dense Video Captioning (MDVC) framework. Given an input consisting of several modalities, namely, audio, speech, and visual, internal representations are produced by a corresponding feature transformer (middle). Then, the features are fused in the multi-modal generator (right) that outputs the distribution over the vocabulary.

3. Proposed Framework

In this section, we briefly outline the workflow of our method referred to as Multi-modal Dense Video Captioning (MDVC) which is shown in Fig. 2. The goal of our method is to temporally localize events on a video and to produce a textual description for each of them. To this end, we apply a two-stage approach.

Firstly, we obtain the temporal event locations. For this task, we employ the *Bidirectional Single-Stream Temporal* action proposals network (Bi-SST) proposed in [48]. Bi-SST applies 3D Convolution network (C3D) [44] to video frames and extracts features that are passed to subsequent *bi-directional LSTM* [17] network. The LSTM accumulates visual cues over time and predicts confidence scores for each location to be start/end point of an event. Finally, a set of event proposals (start/end times) is obtained and passed to the second stage for caption generation.

Secondly, we generate the captions given a proposal. To produce inputs from audio, visual, and speech modalities, we use *Inflated 3D convolutions* (I3D) [6] for visual and *VGGish network* [15] for audio modalities. For speech representation as a text, we employ an external ASR system [1]. To represent the text into a numerical form, we use a similar text embedding which is used for caption encoding. The features are, then, fed to individual transformer models along with the words of a caption from the previous time steps. The output of the transformer is passed into a *generator* which fuses the outputs from all modalities and estimates a probability distribution over the word vocabulary. After sampling the next word, the process is repeated until a special end *token* is obtained. Fig. 1 illustrates an example modality and the corresponding event captions.

3.1. Temporal Event Localization Module

An event localization module is dedicated to generating a set of temporal regions which might contain an event. To achieve this, we employ pre-trained *Bidirectional Single-Stream Temporal* action proposals network (Bi-SST) proposed in [48] as it has been shown to reach good performance in the proposal generation task.

Bi-SST inputs a sequence of T RGB frames from a video $V = (x_1, x_2, \dots, x_T)$ and extracts a set of 4096-d features $V' = (f_1, f_2, \dots, f_T)$ by applying a 3D Convolution network (C3D) on non-overlapping segments of size 16 with a stride of 64 frames. To reduce the feature dimension, only 500 principal components were selected using PCA.

To account for the video context, events are proposed during forward and backward passes on a video sequence V' , and, then, the resulting scores are fused together to obtain the final proposal set. Specifically, during the *forward pass*, LSTM is used to accumulate the visual clues from the “past” context at each position t which is treated as an *ending* point and produce confidence scores for each proposal.

Afterwards, a similar procedure is performed during the *backward pass* where the features V' are used in a reversed order. This empowers the model to have a sense of the “future” context in a video. In contrast to the forward pass, each position is treated as a *starting* point of the proposal. Finally, the confidence scores from both passes are fused by multiplication of corresponding scores for each proposal at each time step, and, then, filtered according to a predefined threshold.

Finally, we obtain a set of N_V event proposals for caption generation $P_V = \{p_j = (\text{start}_j, \text{end}_j, \text{score}_j)\}_{j=1}^{N_V}$.

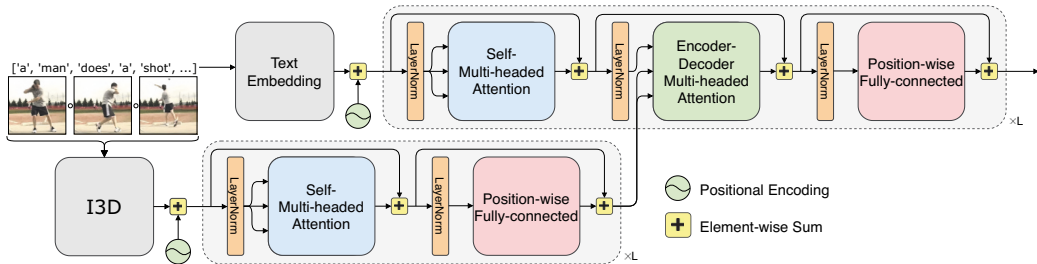


Figure 3. The proposed feature transformation architecture that consists of an encoder (bottom part) and a decoder (top part). The encoder inputs pre-processed and position-encoded features from I3D (in case of the visual modality), and outputs an internal representation. The decoder, in turn, is conditioned on both position-encoded caption that is generated so far and the output of the encoder. Finally, the decoder outputs its internal representation.

3.2. Captioning Module

In this section we explain the captioning based for an example modality, namely, visual. Given a video V and a set of proposals P_V from the event localization module, the task of the captioning module is to provide a caption for each proposal in P_V . In order to extract features from a video V , we employ I3D network [6] pre-trained on the Kinetics dataset which produces 1024-d features. The gap between the extracted features and the generated captions is filled with Transformer [45] architecture which was proven to effectively encode and decode the information in a sequence-to-sequence setting.

3.2.1 Feature Transformer

As shown in Fig. 3, Feature Transformer architecture mainly consists of three blocks: an *encoder*, *decoder*, and *generator*. The encoder inputs a set of extracted features $\mathbf{v}^j = (v_1, v_2, \dots, v_{T_j})$ temporally corresponding to a proposal p_j from P_V and maps it to a sequence of internal representations $\mathbf{z}^j = (z_1, z_2, \dots, z_{T_j})$. The decoder is conditioned on the output of the encoder \mathbf{z}^j and the *embedding* $\mathbf{e}_{\leq t}^j = (e_1, e_2, \dots, e_t)$ of the words in a caption $\mathbf{w}_{\leq t}^j = (w_1, w_2, \dots, w_t)$. It produces the representation $\mathbf{g}_{\leq t}^j = (g_1, g_2, \dots, g_t)$ which, in turn, is used by the generator to model a distribution over a vocabulary for the next word $p(w_{t+1} | \mathbf{g}_{\leq t}^j)$. The next word is selected greedily by obtaining the word with the highest probability until a special *ending token* is sampled. The captioning is initialized with a *starting token*. Both are added to the vocabulary.

Before providing an overview of the encoder, decoder, and generator, we presenting the notion of *multi-headed attention* that acts as an essential part of the decoder and encoder blocks. The concept of the multi-head attention, in turn, heavily relies on *dot-product attention* which we describe next.

Dot-product Attention The idea of the multi-headed attention rests on the *scaled dot-product attention* which calculates the weighted sum of *values*. The weights are obtained by applying the softmax function on the dot-product of each pair of rows of *queries* and *keys* scaled by $\frac{1}{\sqrt{D_k}}$. The scaling is done to prevent the softmax function from being in the small gradient regions [45]. Formally the scaled dot-product attention can be represented as follows

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{D_k}}\right)V, \quad (1)$$

where Q, K, V are queries, keys, and values, respectively.

Multi-headed Attention The multi-headed attention block is used once in each encoder layer and twice in each decoder layer. The block consists of H heads that allows to cooperatively account for information from several representations sub-spaces at every position while preserving the same computation complexity [45]. In a transformer with dimension D_T , each head is defined in the following way

$$\text{head}_h(q, k, v) = \text{Attention}(qW_h^q, kW_h^k, vW_h^v), \quad (2)$$

where q, k, v are matrices which have D_T columns and the number of rows depending on the position of the multi-headed block, yet with the same number of rows for k and v to make the calculation in (1) to be feasible. The $W_h^q, W_h^k, W_h^v \in \mathbb{R}^{D_T \times D_k}$ are trainable projection matrices that map q, k, v from D_T into $D_k = \frac{D_T}{H}$, asserting D_T is a multiple of H . The multi-head attention, in turn, is the concatenation of all attention heads mapped back into D_T by trainable parameter matrix $W^o \in \mathbb{R}^{D_k \cdot H \times D_T}$:

$$\text{MultiHead}(q, k, v) = \begin{bmatrix} \text{head}_1(q, k, v) \\ \dots \\ \text{head}_H(q, k, v) \end{bmatrix} W^o. \quad (3)$$

Encoder The encoder consists of L layers. The first layer inputs a set of features \mathbf{v}^j and outputs an internal representation $\mathbf{z}_1^j \in \mathbb{R}^{T_j \times D_T}$ while each of the next layers treats the output of a previous layer as its input. Each encoder layer l consist of two sub-layers: *multi-headed attention* and *position-wise fully connected network* which are explained later in this section. The input to both sub-layers are normalized using layer normalization [3], each sub-layer is surrounded by a residual connection [14] (see Fig. 3). Formally, the l -th encoder layer has the following definition

$$\bar{\mathbf{z}}_l^j = \text{LayerNorm}(\mathbf{z}_l^j) \quad (4)$$

$$\mathbf{r}_l^j = \mathbf{z}_l^j + \text{MultiHead}(\bar{\mathbf{z}}_l^j, \bar{\mathbf{z}}_l^j, \bar{\mathbf{z}}_l^j) \quad (5)$$

$$\bar{\mathbf{r}}_l^j = \text{LayerNorm}(\mathbf{r}_l^j) \quad (6)$$

$$\mathbf{z}_{l+1}^j = \mathbf{r}_l^j + \text{FCN}(\bar{\mathbf{r}}_l^j), \quad (7)$$

where FCN is the position-wise fully connected network. Note, the multi-headed attention has identical queries, keys, and values ($\bar{\mathbf{z}}_l^j$). Such multi-headed attention block is also referred to as *self*-multi-headed attention. It enables an encoder layer l to account for the information from all states from the previous layer \mathbf{z}_{l-1}^j . This property contrasts with the idea of RNN which accumulates only the information from the past positions.

Decoder Similarly to the encoder, the decoder has L layers. At a position t , the decoder inputs a set of embedded words $\mathbf{e}_{\leq t}^j$ with the output of the encoder \mathbf{z}^j and sends the output to the next layer which is conditioned on this output and, again, the encoder output \mathbf{z}^j . Eventually, the decoder producing its internal representation $\mathbf{g}_{\leq t}^j \in \mathbb{R}^{t \times D_T}$. The decoder block is similar to the encoder but has an additional sub-layer that applies multi-headed attention on the encoder output and the output of its previous sub-layer. The decoder employs the layer normalization and residual connections at all three sub-layers in the same fashion as the encoder. Specifically, the l -th decoder layer has the following form:

$$\bar{\mathbf{g}}_l^j = \text{LayerNorm}(\mathbf{g}_{l, \leq t}^j) \quad (8)$$

$$\mathbf{b}_l^j = \mathbf{g}_{l, \leq t}^j + \text{MultiHead}(\bar{\mathbf{g}}_l^j, \bar{\mathbf{g}}_l^j, \bar{\mathbf{g}}_l^j) \quad (9)$$

$$\bar{\mathbf{b}}_l^j = \text{LayerNorm}(\mathbf{b}_l^j) \quad (10)$$

$$\mathbf{u}_l^j = \mathbf{g}_{l, \leq t}^j + \text{MultiHead}(\bar{\mathbf{b}}_l^j, \mathbf{z}^j, \mathbf{z}^j) \quad (11)$$

$$\bar{\mathbf{u}}_l^j = \text{LayerNorm}(\mathbf{u}_l^j) \quad (12)$$

$$\mathbf{g}_{l+1, \leq t}^j = \mathbf{u}_l^j + \text{FCN}(\bar{\mathbf{u}}_l^j), \quad (13)$$

where \mathbf{z}^j is the encoder output. Note, similarly to the encoder, (9) is a self-multi-headed attention function while the second multi-headed attention block attends on both the encoder and decoder and is also referred to as *encoder-decoder* attention. This block enables each layer of the decoder to attend all state of the encoder’s output \mathbf{z}^j .

Position-wise Fully-Connected Network The fully connected network is used in each layer of the encoder and the decoder. It is a simple two-layer neural network that inputs x with the output of the multi-head attention block, and, then, projects each row (or position) of the input x from D_T space onto D_P , ($D_P > D_T$) and back, formally:

$$\text{FCN}(x) = \text{ReLU}(xW_1 + b_1)W_2 + b_2, \quad (14)$$

where $W_1 \in \mathbb{R}^{D_T \times D_P}$, $W_2 \in \mathbb{R}^{D_P \times D_T}$, and biases b_1, b_2 are trainable parameters, ReLU is a rectified linear unit.

Generator At the position t , the generator consumes the output of the decoder $\mathbf{g}_{\leq t}^j$ and produces a distribution over the vocabulary of words $p(w_{t+1} | \mathbf{g}_{\leq t}^j)$. To obtain the distribution, the generator applies the softmax function of the output of a fully connected layer with a weight matrix $W_G \in \mathbb{R}^{D_T \times D_V}$ where D_V is a vocabulary size. The word with the highest probability is selected as the next one.

Input Embedding and Positional Encoding Since the representation of textual data is usually sparse due to a large vocabulary, the dimension of the input of a neural language model is reduced with an embedding into a dimension of a different size, namely D_T . Also, following [45], we multiply the embedding weights by $\sqrt{D_T}$. The *position encoding* is required to allow the transformer to have a sense of the order in an input sequence. We adopt the approach proposed for a transformer architecture, *i.e.* we add the output of the combination of sine and cosine functions to the embedded input sequence [45].

3.2.2 Multi-modal Dense Video Captioning

In this section, we present the multi-modal dense video captioning module which, utilises visual, audio, and speech modalities. See Fig. 3 for a schematic representation of the module.

For the sake of speech representation $\mathbf{s}^j = (s_1, s_2, \dots, s_{T_s^j})$, we use the text embedding of size 512-d that is similar to the one which is employed in the embedding of a caption $\mathbf{w}_{\leq t}^j$. To account for the audio information, given a proposal p_j we extract a set of features $\mathbf{a}_j = (a_1, a_2, \dots, a_{T_a^j})$ applying the 128-d embedding layer of the pre-trained VGGish network [15] on an audio track. While the visual features $\mathbf{v}^j = (v_1, v_2, \dots, v_{T_v^j})$ are encoded with 1024-d vectors by Inflated 3D (I3D) convolutional network [6].

To fuse the features, we create an encoder and a decoder for each modality with dimensions corresponding to the size of the extracted features. The outputs from all decoders are fused inside of the generator, and the distribution of a next word w_{t+1} is formed.

In our experimentation, we found that a simple two-layer fully-connected network applied a matrix of concatenated features performs the best with the *ReLU* activation after the first layer and the softmax after the second one. Each layer of the network has a matrix of trainable weights: $W_{F_1} \in \mathbb{R}^{D_F \times D_V}$ and $W_{F_2} \in \mathbb{R}^{D_V \times D_V}$ with $D_F = 512 + 128 + 1024$ and D_V is a vocabulary size.

3.3. Model Training

As the training is conducted using mini-batches of size 28, the features in one modality must be of the same length so the features could be stacked into a tensor. In this regard, we *pad* the features and the embedded captions to match the size of the longest sample.

The model is trained by optimizing the Kullback–Leibler divergence loss which measures the “distance” between the ground truth and predicted distributions and averages the values for all words in a batch ignoring the *masked* tokens.

Since many words in the English language may have several synonyms or human annotation may contain mistakes, we undergo the model to be less certain about the predictions and apply Label Smoothing [43] with the smoothing parameter γ on the ground truth labels to mitigate this. In particular, the ground truth distribution over the vocabulary of size D_V , which is usually represented as one-hot encoding vector, the identity is replaced with probability $1 - \gamma$ while the rest of the values are filled with $\frac{\gamma}{D_V - 1}$.

During training, we exploit the *teacher forcing* technique which uses the ground truth sequence up to position t as the input to predict the next word instead of using the sequence of predictions. As we input the whole ground truth sequence at once and predicting the next words at each position, we need to prevent the transformer from peeping for the information from the next positions as it attends to all positions of the input. To mitigate this, we apply masking inside of the self-multi-headed attention block in the decoder for each position higher than $t - 1$, following [45].

The details on the feature extraction and other implementation details are available in the supplementary materials.

4. Experiments

4.1. Dataset

We perform our experiments using ActivityNet Captions dataset [24] that is considered as the standard benchmark for dense video captioning task. The dataset contains approximately 20k videos from YouTube and split into 50/25/25 % parts for training, validation, and testing, respectively. Each video, on average, contains 3.65 temporally localized captions, around 13.65 words each, and two minutes long. In addition, each video in the validation set is annotated twice by different annotators. We report all results using the validation set (no ground truth is provided for the test set).

Method	GT Proposals			Learned Proposals		
	B@3	B@4	M	B@3	B@4	M
<i>Seen full dataset</i>						
Krishna <i>et al.</i> [24]	4.09	1.60	8.88	1.90	0.71	5.69
Wang <i>et al.</i> [48]	–	–	10.89	2.55	1.31	5.86
Zhou <i>et al.</i> [59]	5.76	2.71	11.16	2.42	1.15	4.98
Li <i>et al.</i> [26]	4.55	1.62	10.33	2.27	0.73	6.93
<i>Seen part of the dataset</i>						
Rahman <i>et al.</i> [35]	3.04	1.46	7.23	1.85	0.90	4.93
MDVC	4.12	1.81	10.09	2.31	0.92	6.80
MDVC, no missings	5.83	2.86	11.72	2.60	1.07	7.31

Table 1. The results of the dense video captioning task on the ActivityNet Captions validation sets in terms of BLEU–3,4 (B@3, B@4) and METEOR (M). The related methods are compared with the proposed approach (MDVC) in two settings: on the full validation dataset and a part of it with the videos with all modalities present for a fair comparison (“no missings”). Methods are additionally split into the ones which “saw” all training videos and another ones which trained on partially available data. The results are presented for both ground truth (GT) and learned proposals.

The dataset itself is distributed as a collection of links to YouTube videos, some of which are no longer available. Authors provide pre-computed C3D features and frames at 5fps, but these are not suitable for our experiments. At the time of writing, we found 9,167 (out of 10,009) training and 4,483 (out of 4,917) validation videos which is, roughly, 91 % of the dataset. Out of these 2,798 training and 1,374 validation videos (approx. 28 %) contain at least one speech segment. The speech content was obtained from the *closed captions* (CC) provided by the YouTube ASR system which can be thought as subtitles.

4.2. Metrics

We are evaluating the performance of our model using BLEU@N [33] and METEOR [8]. We regard the METEOR as our primary metric as it has been shown to be highly correlated with human judgement in a situation with a limited number of references (only one, in our case).

We employ the official evaluation script provided in [23]. Thus, the metrics are calculated if a proposed event and a ground truth location of a caption overlaps more than a specified *temporal Intersection over Union* (tIoU) and zero otherwise. All metric values are averaged for every video, and, then, for every threshold tIoU in [0.3, 0.5, 0.7, 0.9]. On the validation, we average the resulting scores for both validation sets. For the learned proposal setting, we report our results on at most 100 proposals per video.

Notably, up to early 2017, the evaluation code had an issue which previously overestimated the performance of the algorithms in the learned proposal setting [30]. Therefore, we report the results using the new evaluation code.

Model	Params. ($\times 10^6$)	Metric	
		B@4	M
Feature Transf. (random)	42	0.88	7.16
Bi-GRU	55	1.44	9.47
Feature Transformer	42	1.84	9.62

Table 2. Comparison of the Feature Transformer and the Bi-directional GRU (Bi-GRU) architectures in terms of BLEU-4 (B@4), METEOR (M), and a number of model parameters. The input to all models is visual modality (I3D). The results indicate the superior performance of the Feature Transformer on all metrics. Additionally, we report the random input baseline which acts as a lower performance bound. The best results are highlighted.

4.3. Comparison with Baseline Methods

We compare our method with five related approaches, namely Krishna *et al.* [24], Wang *et al.* [48], Zhou *et al.* [59], Li *et al.* [26], and Rahman *et al.* [35]. We take the performance values from the original papers, except for [26], and [59], which are taken from [30] due to the evaluation issue (see Sec. 4.2).

The lack of access to the full ActivityNet Captions dataset makes strictly fair comparison difficult as we have less training and validation videos. Nevertheless, we present our results in two set-ups: 1) full validation set with random input features for missing entries, and 2) videos with all three modalities present (video, audio, and speech). The first one is chosen to indicate the lower bound of our performance with the full dataset. Whereas, the second one (referred to as “no missings”) concentrates on the multi-modal setup, which is the main contribution of our work.

The obtained results are presented in Tab. 1. Our method (MDVC) achieves comparable or better performance, even though we have access to smaller training set and 9% of the validation videos are missing (replaced with random input features). Furthermore, if all three modalities are present, our method outperforms all baseline approaches in the case of both GT and learned proposals. Notably, we outperform [59] which is also based on the transformer architecture and account for the optical flow. This shows the superior performance of our captioning module which, yet, trained on the smaller amount of data.

4.4. Ablation Studies

In this section, we perform an ablation analysis highlighting the effect of different design choices of our method. For all experiments, we use the full unfiltered ActivityNet Captions validation set with ground truth event proposals.

Firstly, we assess the selection of the model architecture. To this end, we implemented a version of our method where the transformer was replaced by *Bidirectional Recurrent Neural Network with Gated Recurrent Units with atten-*

Modality			Fusion	Params. ($\times 10^6$)	Metric	
V	A	S			B@4	M
✓			–	42	1.61	9.64
	✓		–	5	1.03	8.01
✓	✓		Average probs.	46	1.68	9.71
✓	✓		Concat. + 2 FC	149	1.73	9.87
✓			No, 2 FC	145	1.62	9.69
✓	✓	✓	Concat. + 2 FC	179	1.81	10.09

Table 3. The performance of the proposed MDVC framework with different input modalities (V-visual, A-audio, S-speech) and feature fusion approaches: probability averaging and concatenation of two fully-connected layers (Concat. + 2 FC). Also, we report the comparison between audio-visual MDVC with visual-only MDVC with similar model capacities (2 FC).

tion (Bi-GRU), proposed in [4]. To distil the effect of the change in architecture, the results are shown for visual-only models. Both Bi-GRU and the transformer input I3D features extracted from 64 RGB and optical flow frames (the final model inputs 24 frames). Finally, we set a lower bound for the feature performance by training a transformer model with random video features. Tab. 2 shows the comparison. To conclude, we observe that the feature transformer-based model is not only uses less parameters but also achieves better performance in dense video captioning task. Moreover, both method clearly surpasses the random baseline.

Secondly, we evaluate the contribution of different modalities in our framework. Tab. 3 contains the results for different modality configurations as well as for two feature fusion approaches. Specifically, averaging of the output probabilities and concatenation of the outputs of all modalities and applying two fully connected (FC) layers on top. We observe that audio-only model has the worst performance, followed by the visual only model, and the combination of these two. Moreover, the concatenation and FC layers result in better performance than averaging. To further assess if the performance gain is due to the additional modalities or to the extra capacity in the FC layers, we trained a visual-only model with two additional FC layers. The results indicate that such configuration performs worse than any bi-modal setup. Overall, we conclude that the final model with all three modalities performs best among all tested set-ups, which highlights the importance of multi-modal setting in dense video captioning task.

Fig. 4 shows a qualitative comparison between different models in our ablation study. Moreover, we provide the corresponding captions from the best performing baseline method (Zhuo *et al.* [59]). We noticed the following pattern: the audio-modality produces coherent sentences and captures the concepts of speaking in the video. However, there are clear mistakes in the caption content. In contrast, the model with all three modalities manages to capture

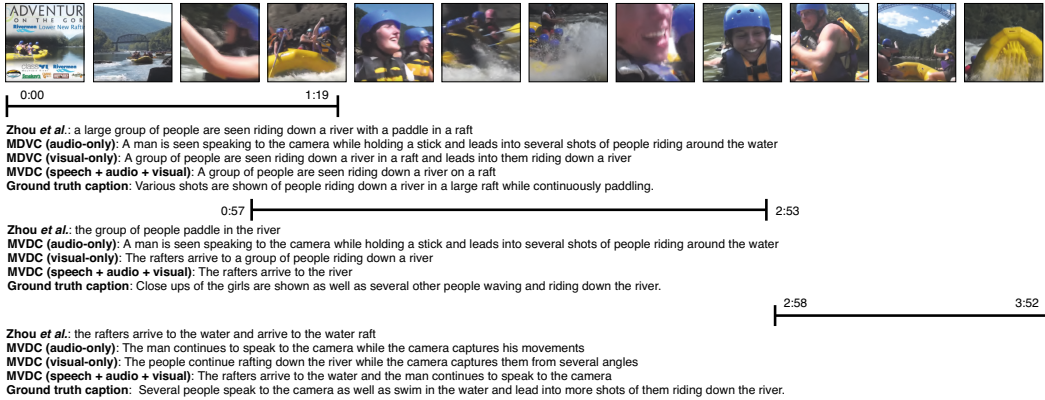


Figure 4. The qualitative captioning results for an example video from the ActivityNet Captions validation set. In the video, the speaker describes the advantages of rafting on this particular river and their club. Occasionally, people are shown rapturously speaking about how fun it is. Models that account for audio modality tend to grasp the details of the speaking on the scene while the visual-only models fail at this. We invite the reader to watch the example YouTube video for a better impression ([xS5imfBbWmw](https://www.youtube.com/watch?v=xS5imfBbWmw)).

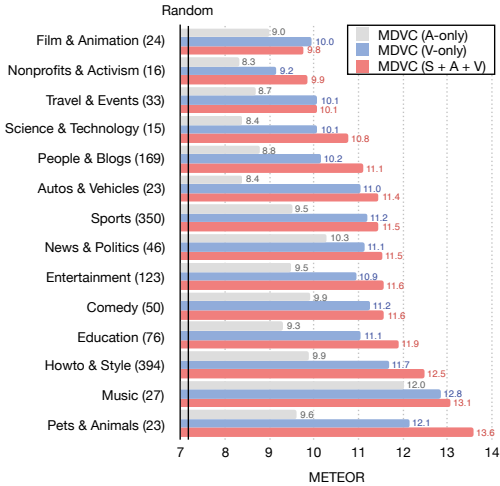


Figure 5. The results are split for category and version of MDVC. The number of samples per category is given in parenthesis. The METEOR axis is cut up to the random performance level (7.16).

the man who speaks to the camera which is also present in the ground truth. Both visual-only MDVC and Zhou et al. struggle to describe the audio details.

Finally, to test whether our model improves the performance in general rather than in a specific video category, we report the comparison of the different versions of MDVC per category. To this end, we retrieve the category labels from the YouTubeAPI [2] (US region) for every available

ActivityNet Captions validation video. These labels are given by the user when uploading the video and roughly represent the video content type. The comparison is shown in Fig. 5. The results imply a consistent gain in performance within each category except for categories: “Film & Animation” and “Travel & Events” which might be explained by the lack of correspondence between visual and audio tracks. Specifically, the video might be accompanied by music, e.g. promotion of a resort. Also, “Film & Animation” contains cartoon-like movies which might have a realistic soundtrack while the visual track is goofy.

5. Conclusion

The use of different modalities in computer vision is still an underrepresented topic and, we believe, deserves more attention. In this work, we introduced a multi-modal dense video captioning module (MDVC) and shown the importance of the audio and speech modalities for dense video captioning task. Specifically, MDVC is based on the transformer architecture which encodes the feature representation of each modality for a specific event proposal and produces a caption using the information from these modalities. The experimentation, conducted employing the ActivityNet Captions dataset, shows the superior performance of a captioning module to the visual-only models in the existing literature. Extensive ablation study verifies this conclusion. We believe that our results firmly indicate that future works in video captioning should utilize a multi-modal input.

Acknowledgments Funding for this research was provided by the Academy of Finland projects 327910 & 324346. The authors acknowledge CSC — IT Center for Science, Finland, for computational resources.

References

- [1] YouTube Data API Video Captions. <https://developers.google.com/youtube/v3/docs/captions>, [Accessed 1 November 2019].
- [2] YouTube Data API Video Categories. <https://developers.google.com/youtube/v3/docs/videoCategories>, [Accessed 1-November-2019].
- [3] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization, 2016.
- [4] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473, 2014.
- [5] L. Baraldi, C. Grana, and R. Cucchiara. Hierarchical boundary-aware neural encoder for video captioning. In *CVPR*, 2017.
- [6] J. Carreira and A. Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, 2017.
- [7] P. Das, C. Xu, R. F. Doell, and J. J. Corso. A thousand frames in just a few words: Linguistic description of videos through latent topics and sparse object stitching. In *CVPR*, 2013.
- [8] Michael Denkowski and Alon Lavie. Meteor universal: Language specific translation evaluation for any target language. In *ACL Workshop*, 2014.
- [9] J. Donahue, L. A. Hendricks, M. Rohrbach, S. Venugopalan, S. Guadarrama, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. *TPAMI*, 2017.
- [10] Xuguang Duan, Wenbing Huang, Chuang Gan, Jingdong Wang, Wenwu Zhu, and Junzhou Huang. Weakly supervised dense event captioning in videos. In *NeurIPS*, 2018.
- [11] Z. Gan, C. Gan, X. He, Y. Pu, K. Tran, J. Gao, L. Carin, and L. Deng. Semantic compositional networks for visual captioning. In *CVPR*, 2017.
- [12] Jort F. Gemmeke, Daniel P. W. Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R. Channing Moore, Manoj Plakal, and Marvin Ritter. Audio set: An ontology and human-labeled dataset for audio events. In *ICASSP*, 2017.
- [13] Wang-Li Hao, Zhaoxiang Zhang, and He Guan. Integrating both visual and audio cues for enhanced video caption. In *AAAI*, 2017.
- [14] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.
- [15] Shawn Hershey, Sourish Chaudhuri, Daniel P. W. Ellis, Jort F. Gemmeke, Aren Jansen, Channing Moore, Manoj Plakal, Devin Platt, Rif A. Saurous, Bryan Seybold, Malcolm Slaney, Ron Weiss, and Kevin Wilson. CNN architectures for large-scale audio classification. In *ICASSP*, 2017.
- [16] Jack Hessel, Bo Pang, Zhenhai Zhu, and Radu Soricut. A case study on combining ASR and visual features for generating instructional video captions. In *CoNLL*, 2019.
- [17] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 1997.
- [18] C. Hori, T. Hori, T. Lee, Z. Zhang, B. Harsham, J. R. Hershey, T. K. Marks, and K. Sumi. Attention-based multimodal fusion for video description. In *ICCV*, 2017.
- [19] Chiori Hori, Takaaki Hori, Gordon Wichern, Jue Wang, Teng-Yok Lee, Anoop Cherian, and Tim K. Marks. Multimodal attention for fusion of audio and spatiotemporal features for video description. In *CVPR Workshops*, 2018.
- [20] Justin Johnson, Andrej Karpathy, and Li Fei-Fei. Densecap: Fully convolutional localization networks for dense captioning. In *CVPR*, 2016.
- [21] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.
- [22] Atsuhiko Kojima, Takeshi Tamura, and Kunio Fukunaga. Natural language description of human activities from video images based on concept hierarchy of actions. *IJCV*, 2002.
- [23] Ranjay Krishna. Evaluation code for dense-captioning events in videos. https://github.com/ranjaykrishna/densevid_eval/tree/9d4045aced3d827834a5d2da3c9f0692e3f33c1c.
- [24] Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. Dense-captioning events in videos. In *ICCV*, 2017.
- [25] Yong Jae Lee, Joydeep Ghosh, and Kristen Grauman. Discovering important people and objects for egocentric video summarization. In *CVPR*, 2012.
- [26] Yehao Li, Ting Yao, Yingwei Pan, Hongyang Chao, and Tao Mei. Jointly localizing and describing events for dense video captioning. In *CVPR*, 2018.
- [27] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. SSD: Single shot multibox detector. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *ECCV*, 2016.
- [28] C. Ma, A. Kadav, I. Melvin, Z. Kira, G. AlRegib, and H. P. Graf. Attend and interact: Higher-order object interactions for video understanding. In *CVPR*, 2018.
- [29] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. HowTo100M: Learning a text-video embedding by watching hundred million narrated video clips. In *ICCV*, 2019.
- [30] Jonghwan Mun, Linjie Yang, Zhou Ren, Ning Xu, and Bohyung Han. Streamlined dense video captioning. In *CVPR*, 2019.
- [31] Mun Wai Lee, A. Hakeem, N. Haering, and Song-Chun Zhu. Save: A framework for semantic annotation of visual events. In *CVPR Workshops*, 2008.
- [32] Y. Pan, T. Yao, H. Li, and T. Mei. Video captioning with transferred semantic attributes. In *CVPR*, 2017.
- [33] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: A method for automatic evaluation of machine translation. In *ACL*, 2002.
- [34] Wenjie Pei, Jiyuan Zhang, Xiangrong Wang, Lei Ke, Xiaoyong Shen, and Yu-Wing Tai. Memory-attended recurrent network for video captioning. In *CVPR*, 2019.
- [35] Tanzila Rahman, Bicheng Xu, and Leonid Sigal. Watch, listen and tell: Multi-modal weakly supervised dense event captioning. In *ICCV*, 2019.
- [36] Anna Rohrbach, Marcus Rohrbach, Wei Qiu, Annemarie Friedrich, Manfred Pinkal, and Bernt Schiele. Coherent multi-sentence video description with variable level of detail.

- In Xiaoyi Jiang, Joachim Hornegger, and Reinhard Koch, editors, *Pattern Recognition*. Springer International Publishing, 2014.
- [37] M. Rohrbach, W. Qiu, I. Titov, S. Thater, M. Pinkal, and B. Schiele. Translating video content to natural language descriptions. In *ICCV*, 2013.
- [38] Z. Shen, J. Li, Z. Su, M. Li, Y. Chen, Y. Jiang, and X. Xue. Weakly supervised dense video captioning. In *CVPR*, 2017.
- [39] Botian Shi, Lei Ji, Yaobo Liang, Nan Duan, Peng Chen, Zhendong Niu, and Ming Zhou. Dense procedure captioning in narrated instructional videos. In *ACL*, 2019.
- [40] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014.
- [41] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume. In *CVPR*, 2018.
- [42] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *NeurIPS*, 2014.
- [43] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, 2016.
- [44] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3D convolutional networks. In *ICCV*, 2015.
- [45] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017.
- [46] Subhashini Venugopalan, Marcus Rohrbach, Jeffrey Donahue, Raymond Mooney, Trevor Darrell, and Kate Saenko. Sequence to sequence – video to text. In *ICCV*, 2015.
- [47] Subhashini Venugopalan, Huijuan Xu, Jeff Donahue, Marcus Rohrbach, Raymond Mooney, and Kate Saenko. Translating videos to natural language using deep recurrent neural networks. In *NAACL-HLT*, 2015.
- [48] J. Wang, W. Jiang, L. Ma, W. Liu, and Y. Xu. Bidirectional attentive fusion with context gating for dense video captioning. In *CVPR*, 2018.
- [49] J. Wang, W. Wang, Y. Huang, L. Wang, and T. Tan. M3: Multimodal memory modelling for video captioning. In *CVPR*, 2018.
- [50] X. Wang, W. Chen, J. Wu, Y. Wang, and W. Y. Wang. Video captioning via hierarchical reinforcement learning. In *CVPR*, 2018.
- [51] Xin Wang, Yuan-Fang Wang, and William Yang Wang. Watch, listen, and describe: Globally and locally aligned cross-modal attentions for video captioning. In *NAACL-HLT*, 2018.
- [52] Xian Wu, Guanbin Li, Qingxing Cao, Qingge Ji, and Liang Lin. Interpretable video captioning via trajectory structured localization. In *CVPR*, 2018.
- [53] Yilei Xiong, Bo Dai, and Dahua Lin. Move forward and tell: A progressive generator of video descriptions. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *ECCV*, 2018.
- [54] H. Xu, B. Li, V. Ramanishka, L. Sigal, and K. Saenko. Joint event detection and description in continuous video streams. In *WACV*, 2019.
- [55] Jun Xu, Ting Yao, Yongdong Zhang, and Tao Mei. Learning multimodal attention lstm networks for video captioning. In *ACM*, 2017.
- [56] C. Yan, Y. Tu, X. Wang, Y. Zhang, X. Hao, Y. Zhang, and Q. Dai. STAT: Spatial-temporal attention mechanism for video captioning. *Transactions on Multimedia*, 2020.
- [57] Li Yao, Atousa Torabi, Kyunghyun Cho, Nicolas Ballas, Christopher Pal, Hugo Larochelle, and Aaron Courville. Describing videos by exploiting temporal structure. In *ICCV*, 2015.
- [58] H. Yu, J. Wang, Z. Huang, Y. Yang, and W. Xu. Video paragraph captioning using hierarchical recurrent neural networks. In *CVPR*, 2016.
- [59] Luowei Zhou, Yingbo Zhou, Jason J. Corso, Richard Socher, and Caiming Xiong. End-to-end dense video captioning with masked transformer. In *CVPR*, 2018.

PUBLICATION

II

A better use of audio-visual cues: Dense video captioning with bi-modal transformer

V. Iashin and E. Rahtu

In 31st British Machine Vision Conference 2020, BMVA Press, 2020

Publication reprinted with the permission of the copyright holders.

A Better Use of Audio-Visual Cues: Dense Video Captioning with Bi-modal Transformer

Vladimir Iashin
vladimir.iashin@tuni.fi
Esa Rahtu
esa.rahtu@tuni.fi

Computing Sciences
Tampere University
Tampere, Finland

Abstract

Dense video captioning aims to localize and describe important events in untrimmed videos. Existing methods mainly tackle this task by exploiting only visual features, while completely neglecting the audio track. Only a few prior works have utilized both modalities, yet they show poor results or demonstrate the importance on a dataset with a specific domain. In this paper, we introduce *Bi-modal Transformer* which generalizes the Transformer architecture for a bi-modal input. We show the effectiveness of the proposed model with audio and visual modalities on the dense video captioning task, yet the module is capable of digesting any two modalities in a sequence-to-sequence task. We also show that the pre-trained bi-modal encoder as a part of the bi-modal transformer can be used as a feature extractor for a simple proposal generation module. The performance is demonstrated on a challenging *ActivityNet Captions* dataset where our model achieves outstanding performance. The code is available: [v-iashin.github.io/bmt](https://github.com/v-iashin/bmt)

1 Introduction

Current video sharing platforms contain a large amount of video material. The ability to generate descriptions of this content would be highly valuable for many tasks, such as content-based retrieval or recommendation [25, 44]. Moreover, they would enable visually-impaired people to consume video material and improve their quality of life [38].

This kind of video descriptions are usually provided as natural language sentences or *captions*, a compact and intuitive format and, most importantly, can be digested by humans. Early works [46, 47, 56, 58] described the video content with only one sentence, which might be too “sparse” for long videos – one might try to think up a relatively short sentence which describes the whole film. To mitigate this issue, [20] proposed *dense video captioning* which requires a model to, first, localize “events”, and, then, to produce one-sentence description for each of them instead of generating one caption for the entire film (see Fig. 1).

The task is usually formulated as a *sequence-to-sequence* (video to caption) task. Therefore, the progress in the field is significantly influenced by advances in machine translation. Hence, many models rely on an encoder-decoder architecture which consists of two *recurrent neural networks* (RNNs) or, recently-proposed *Transformer*-like model [45]. An event

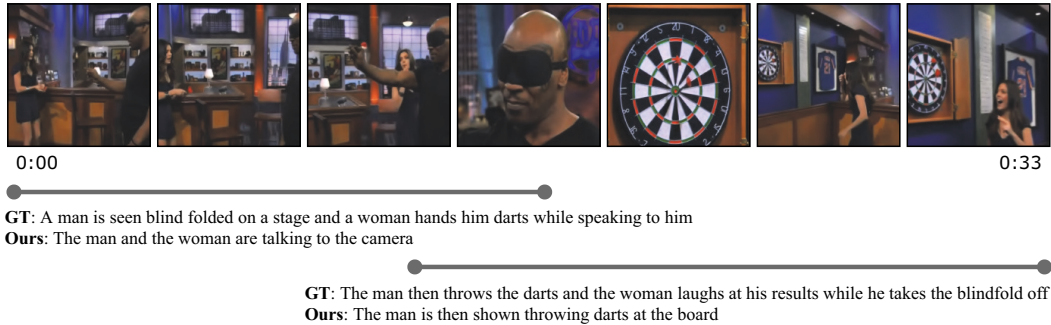


Figure 1: Example video with the predictions of our model alongside the ground truth.

localization module usually utilizes an RNN structure which first encodes the input to produce a hidden representation and, then, makes predictions using this representation.

Considering the natural co-occurrence of visual and audio tracks in a video and the fact that human perception is multi-modal, recent advances in deep learning practice audio-visual training [24, 27, 59, 63, 64]. Yet, most of the existing works on dense video captioning employ only visual inputs. In this work, we address this issue by introducing a novel bi-modal transformer with the multi-headed proposal generator. Our captioning module is inspired by the transformer architecture and, more precisely, how the attention module fuses the information from both sequences. While an efficient object detector *YOLO* [35] inspires the design of each proposal head in the bi-modal multi-headed proposal generator.

The proposed method effectively utilizes audio and visual cues. We demonstrate the performance of our model on the challenging open-domain ActivityNet Captions dataset [20]. The results show the state-of-the-art performance of our bi-modal dense video captioning module as well as our bi-modal proposal generator on BLEU@3-4 and F1 metrics.

2 Related Work

The dense video captioning task requires a model to, first, localize events within a video and, then, to produce a textual one-sentence description of what is happening during the event. The dense video captioning task branches out from the *video captioning* which task is to caption a video without localizing the event. The video captioning field evolved from hand-crafted rule models [6, 19, 21] to *encoder-decoder* architectures [46, 47, 56, 58] inspired by advances in machine translation [39]. Later, the captioning models were further enhanced by *semantic tagging* [11, 28], *reinforcement learning* [51], *attention* [55], *extended memory* [31, 50], and other modalities [13, 16, 52, 54].

2.1 Dense Video Captioning

The task of dense video captioning, as well as a test-bed, ActivityNet Captions dataset, were introduced by Krishna *et al.* [20] who utilized the idea of the *Deep Action Proposals* network [10] to generate event proposals and an LSTM network to encode the context and generate captions. The idea of context-awareness was further developed in [49] who employed a bi-directional variant of *Single-Stream Temporal* Action proposal network (SST) [3] which makes better use of the video context, an LSTM network with *attentive fusion and context*

gating was used to generate context-aware captions. Zhou *et al.* [62] adapted *Transformer* architecture [45] to tackle the task and used transformer *encoder*'s output as input to a modification of *ProcNets* [61] to generate proposals.

Recently, the idea of reinforcement learning was found to be beneficial for image captioning (*Self-critical Sequence Training* (SCST)) [37] and, hence, applied in dense video captioning as well. More precisely, the SCST was used in a captioning module to optimize the non-differentiable target metric, *e.g.* METEOR [7]. Specifically, Li *et al.* [22] integrated the reward system and enriched *Single-Shot-Detector*-like structure [23] with descriptiveness regression for proposal generation. Similarly, Xiong *et al.* [53] used an LSTM network trained with the sentence- and paragraph-level rewards for maintaining coherent and concise story-telling, while the event proposal module was adopted from *Structured Segment Networks* [60]. Mun *et al.* [26] further developed the idea of coherent captioning by observing the overall context and optimizing two-level rewards, an SST module is used for proposal generation, and a *Pointer Network* [48] to distill proposal candidates.

Another direction of research relies on weak supervision which is designed to mitigate the problem of laborious annotation of the datasets. To this end, Duan *et al.* [9] proposed an *autoencoder* architecture which generates proposals and, then, captions them while being supervised only with a set of non-localized captions in a *cycle-consistency* manner. However, the results appeared to be far from the supervised methods.

2.2 Multi-modal Dense Video Captioning

It is natural to assume that, besides visual information, a video understanding system might benefit from the cues contained in other modalities like audio [33], speech (subtitles) [40], or both [17]. Specifically, Rahman *et al.* [33] were the first to include audio modality into the dense video captioning set up. They borrowed the idea of cycle-consistency from [9] and employed *multi-modal Tucker decomposition* [2] to combine information from both modalities and pass it to a *GRU*-based [5] caption decoder. However, since the model is trained in a weakly supervised setting, the results do not reach the performance of the supervised models.

Shi *et al.* [40] proposed to utilize the corresponding speech along with frame features to further improve captioning performance on cooking videos. They suggested employing a transformer's encoder to encode video frames and subtitle *embeddings* produced by a pre-trained *BERT* model [8]. Next, an LSTM generates proposals, and the other two LSTMs were used for the encoder-decoder captioning module. Despite the significant gains in captioning performance, we believe these findings are not conclusive as instructional videos is an ill-suited domain to show the benefits of the speech modality for a captioning task since subtitles alone can be a very accurate proxy for captions in such videos (see [25]).

In contrast, Iashin *et al.* [17] showed the importance of the speech modality on a free-domain dataset. They proposed to train three transformers for each modality individually and fuse features by concatenation before predicting the next caption word while borrowing the proposal generator from [49]. However, the suggested approach for feature fusion is rather straightforward and inefficient. Moreover, the adopted proposal generator is based solely on video features which contrasts with the idea of the dense video captioning task.

Our method is mostly similar to [17], yet we show significantly better results on the task while utilizing only visual and audio cues. Besides, our proposal generator does employ both modalities and significantly outperforms the state-of-the-art. Furthermore, we present a single model which utilizes bi-modal encoder for both: the proposal generator and captioning module, making it an elegant approach for the dense video captioning task.

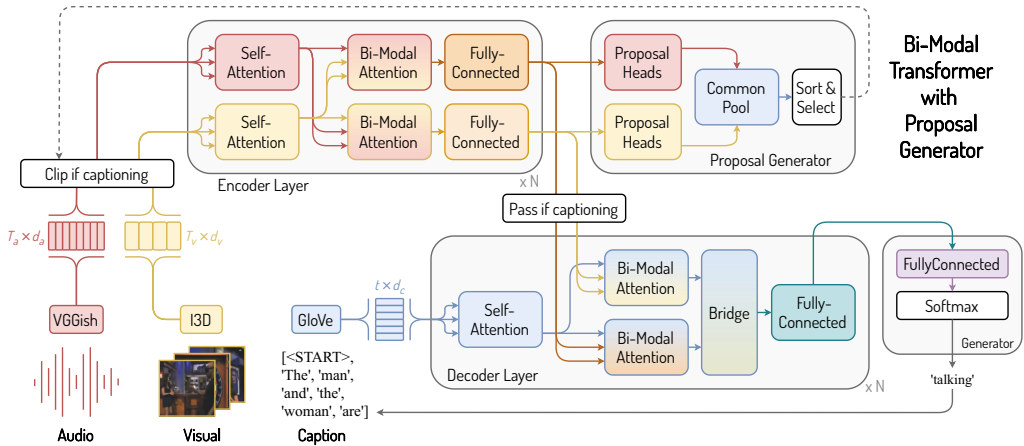


Figure 2: The design of Bi-modal Transformer with Multi-headed Proposal Generator. The proposed model inputs features extracted by VGGish, I3D, and GloVe pre-trained models (bottom left). Then, the bi-modal encoder with N layers processes the audio and visual features and passes its bi-modal representation to the proposal generator (top). After, the generated proposals are used to clip the input features (left). The clipped features are passed through the encoder again. The output of the encoder, then, is used at every layer (N) of the bi-modal decoder (bottom). The decoder attends to the bi-modal encoder’s representation as well as the previous caption words and produces its internal representation of the context. This representation is passed to the generator (right) to generate the next word. Residual connections are removed for clarity. Best viewed in color.

3 Our Framework

Our approach consists of two parts: the *bi-modal transformer* and *multi-headed proposal generator* (see Fig. 2). The model expects the input to be a set of continuous features stacked together in a sequence. To represent a visual stream, we use a pre-trained *Inflated 3D* (I3D) network [4] while for the audio stream we employ pre-trained *VGGish* [15], the tokens (roughly, words) are embedded with pre-trained *GloVe* [32] (see Sec. 6.2 for implementation details). Also, since the transformer is *permutation invariant* it has no sense of recurrence. Thus, the order of features within a sequence is preserved by adding the *positional encoding* to the output of the embedding layers. Following [45], we use *cosine* and *sine* functions.

Next, the audio and visual sequences, are passed through the transformer’s bi-modal N -layered *encoder* to produce bi-modal sequence representations utilizing novel *bi-modal multi-headed attention* blocks to fuse the features from both sequences. Then, the novel proposal generator utilizes these features to generate proposals and their confidence scores. After, a pre-defined number of most confident proposals are selected to clip the input feature sequences. Next, the clipped features are processed with the encoder to re-represent the features considering only the features which are left after clipping.

The bi-modal encoder’s representation is used at every layer in the bi-modal *decoder*. Concretely, the encoder’s outputs are passed to the corresponding bi-modal attention blocks in the decoder layer along with the representation of the previously generated caption words. The last-layer representation of the decoder is used in the *generator* where the next caption word is produced. To avoid an empty input to the decoder in the beginning, a special *start-token* is used. The caption is generated word-by-word until a special *end-token* is sampled.

This section, first, presents the design of the captioning module (Sec. 3.1) and, second, the proposal generator (Sec. 3.2) while the training procedure is explained in Sec. 3.3.

3.1 Captioning Module

The task of dense video captioning requires to produce a caption for each proposal. Therefore, *bi-modal encoder* inputs audio A and visual V feature sequences which temporally correspond to the proposal and outputs two sequences: audio-attended visual features V^a and visual-attended audio features A^v . These features are used by the *bi-modal decoder* which attends to these features and the previous caption words (c_1, c_2, \dots, c_t) . Finally, the bi-modal decoder outputs the representation which is employed to model a distribution of the next caption word (c_{t+1}) over the vocabulary. The proposal index is omitted for clarity.

Bi-modal Encoder In contrast to the encoder in [45], our bi-modal encoder inputs two streams: audio ($A \in \mathbb{R}^{T_a \times d_a}$) and visual ($V \in \mathbb{R}^{T_v \times d_v}$) features corresponding to the proposal. Then, the features are passed in a stack of N encoder layers. Instead of two, each layer has three sub-layers: *self-attention*, *bi-modal attention* (new), and *position-wise fully-connected* layers. Specifically, given $A_0^{\text{fc}} = A$ and $V_0^{\text{fc}} = V$, an n^{th} encoder layer is defined as

$$A_n^{\text{self}} = \text{MultiHeadAttention}(A_{n-1}^{\text{fc}}, A_{n-1}^{\text{fc}}, A_{n-1}^{\text{fc}}), \quad // \text{ audio self-attention} \quad (1)$$

$$V_n^{\text{self}} = \text{MultiHeadAttention}(V_{n-1}^{\text{fc}}, V_{n-1}^{\text{fc}}, V_{n-1}^{\text{fc}}), \quad // \text{ visual self-attention} \quad (2)$$

$$A_n^{\text{mm}} = \text{MultiHeadAttention}(A_n^{\text{self}}, V_n^{\text{self}}, V_n^{\text{self}}), \quad // \text{ visual-attended audio feats.} \quad (3)$$

$$V_n^{\text{mm}} = \text{MultiHeadAttention}(V_n^{\text{self}}, A_n^{\text{self}}, A_n^{\text{self}}), \quad // \text{ audio-attended visual feats.} \quad (4)$$

$$A_n^{\text{fc}} = \text{TwoFullyConnected}(A_n^{\text{mm}}), \quad // \mathbb{R}^{T_a \times d_a} \leftarrow \mathbb{R}^{T_a \times 4d_a} \leftarrow \mathbb{R}^{T_a \times d_a} \quad (5)$$

$$V_n^{\text{fc}} = \text{TwoFullyConnected}(V_n^{\text{mm}}), \quad // \mathbb{R}^{T_v \times d_v} \leftarrow \mathbb{R}^{T_v \times 4d_v} \leftarrow \mathbb{R}^{T_v \times d_v} \quad (6)$$

where all sub-layers have distinct sets of trainable weights and mostly resemble the blocks of Transformer [45], yet we allow the dimension of the weights in multi-headed attention in (3) & (4) to be different for both modalities because we expect them to have a different size. We define the multi-headed attention in Sec. 6.1. The encoder outputs visual-attended audio features ($A^v = A_N^{\text{fc}}$) and audio-attended visual features ($V^a = V_N^{\text{fc}}$), which are used the decoder.

Bi-modal Decoder The bi-modal decoder inputs the previous sequence of caption words $C_t = (c_1, c_2, \dots, c_t) \in \mathbb{R}^{t \times d_c}$ and, opposed to the original Transformer’s decoder [45], ours gets the output from the bi-modal encoder ($A^v \in \mathbb{R}^{T_a \times d_a}$, $V^a \in \mathbb{R}^{T_v \times d_v}$). Thus, instead of three, it has four sub-layers: *self-attention*, *bi-modal encoder-decoder attention* (new), *bridge* (new), & *position-wise fully-connected* layers. For $C_0^{\text{fc}} = C_t$, an n^{th} decoder layer is defined as

$$C_n^{\text{self}} = \text{MultiHeadAttention}(C_{n-1}^{\text{fc}}, C_{n-1}^{\text{fc}}, C_{n-1}^{\text{fc}}), \quad // \text{ caption self-attention} \quad (7)$$

$$C_n^{A^v} = \text{MultiHeadAttention}(C_n^{\text{self}}, A^v, A^v), \quad // \text{ audio-visual attended prev. caps.} \quad (8)$$

$$C_n^{V^a} = \text{MultiHeadAttention}(C_n^{\text{self}}, V^a, V^a), \quad // \text{ visual-audio attended prev. caps.} \quad (9)$$

$$C_n^{\text{mm}} = \text{OneFullyConnected}([C_n^{A^v}, C_n^{V^a}]), \quad // \mathbb{R}^{t \times d_c} \leftarrow \mathbb{R}^{t \times 2d_c}; [\cdot, \cdot] \text{ — concat.} \quad (10)$$

$$C_n^{\text{fc}} = \text{TwoFullyConnected}(C_n^{\text{mm}}), \quad // \mathbb{R}^{t \times d_c} \leftarrow \mathbb{R}^{t \times 4d_c} \leftarrow \mathbb{R}^{t \times d_c} \quad (11)$$

where, as in the encoder, trainable weights have distinct dimensions depending on a modality and are not shared across sub-layers. The decoder outputs caption features ($C_t^{\text{av}} = C_N^{\text{fc}}$).

Generator The purpose of the generator is to model the distribution for the next caption word c_{t+1} given the output of the decoder $C_t^{\text{av}} \in \mathbb{R}^{t \times d_c}$. Therefore, the generator is, usually, a fully-connected layer with the softmax activation which maps the caption features of size d_c into a dimension corresponding to the size of the vocabulary in the training set.

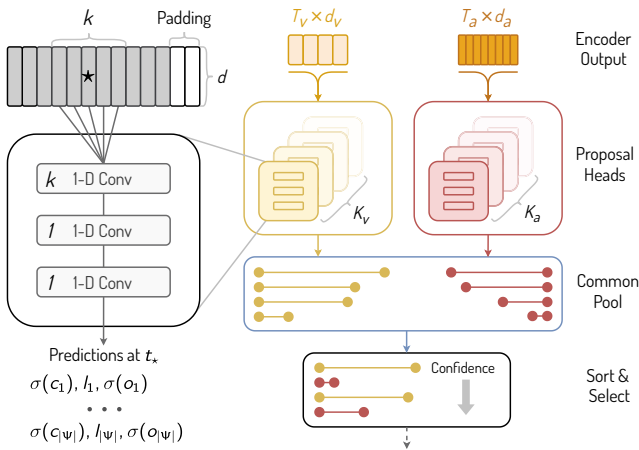


Figure 3: The Bi-modal Multi-headed Proposal Generator inputs the two-stream output from the bi-modal encoder, processes it with two stacks of proposal generation heads. The predictions from all heads form a common pool of predictions. Thus, the pool consists of $T_v \cdot K_v \cdot |\Psi_v| + T_a \cdot K_a \cdot |\Psi_a|$ proposals, which are sorted on the confidence score and passed back to clip input features to the captioning module.

Residual Connection Following the original Transformer architecture, we employ the *residual connection* [14] surrounding each sub-layer of the encoder and decoder except for the bridge layer since in- and out-dimensions are different. Additionally, we adopt Layer Normalization [1] before applying a sub-layer: $x + \text{sub-layer}(\text{LayerNorm}(x))$.

Dropout We also regularize our model with *dropout* [41] which is applied: a) before adding the residual in the residual connection, b) before the activation in the bridge layer, c) on outputs of the positional encoding, d) between layers in the position-wise fully-connected network, and e) after the softmax operation in the scaled dot-product attention (see Sec. 6.1).

3.2 Event Proposal Generation Module

The proposal generator generates a set of proposals for a given video. It consists of two blocks: a bi-modal encoder and *bi-modal multi-headed proposal generator* (not related to multi-headed attention). The bi-modal encoder in this module inputs the whole sequence opposed to the bi-modal encoder in the captioning module, which inputs a sequence of features corresponding to a proposal. Specifically, it inputs both: visual-attended audio features $A^v \in \mathbb{R}^{T_a \times d_a}$ and audio-attended visual features $V^a \in \mathbb{R}^{T_v \times d_v}$. Since the sequence lengths (T_a, T_v) might be distinct, the fusion of predictions cannot be done at each time-step. To this end, we propose the module which makes predictions for each modality at every timestamp individually forming a common pool of cross-modal predictions (see Fig. 3).

Proposal Generation Head The proposal generation head inputs a sequence of T features, and makes predictions at each timestamp on the interval $[1, T]$, and for every prior segment length *anchor* in the set Ψ . The design of the proposal generation head is partly inspired by YOLO object detector [34, 35, 36]. Specifically, it is a *fully-convolutional* network which, in our case, consists of only three layers. Opposed to YOLO, we preserve the sequence length across all layers using *padding* and identity *stride*. Moreover, YOLO utilizes predictions from three different scales to predict different-scale objects. Hence, only three sizes of receptive fields are used. Instead, our model makes predictions at a single scale while controlling the receptive field with a *kernel size* k which is distinct in each proposal generation head. More precisely, the 1st convolutional layer has a kernel size k while in the 2nd and the 3rd the kernel size is 1. The layers are separated with *ReLU* activations and dropout.

Predictions Temporal boundaries and confidence for a proposal are obtained using three values which were predicted by the proposal generation head: a location of a segment center

$\sigma(c)$ relative to a position p in the sequence while $\sigma(\cdot)$ is a sigmoid function which bounds the values into $[0, 1]$ interval, a coefficient $\exp(l)$ for an anchor, and *objectness score* $\sigma(o)$

$$\text{center} = p + \sigma(c); \quad \text{length} = \text{anchor} \cdot \exp(l); \quad \text{confidence} = \sigma(o). \quad (12)$$

The prediction of the center and length are in grid-cells (not in seconds). To obtain seconds, both are multiplied by a cell size which corresponds to a temporal span of the feature.

Bi-modal Multi-headed Proposal Generator The common pool of predictions is formed with predictions made by each of the proposal generation heads. Specifically, our model has K_a and K_v heads for audio and visual modalities with distinct sets of kernel sizes. Overall, our model generates $(T_a \cdot K_a \cdot |\Psi_a| + T_v \cdot K_v \cdot |\Psi_v|)$ proposals. For the final predictions, we select top-100 proposals out of the common pool based on the confidence score.

Segment Length Priors & Kernel Sizes To select a set of anchors, we use *K-Means* clustering algorithm with the *Euclidean distance* metric, as opposed to *intersection over the union* in YOLO. Due to granularity of feature extractors, feature lengths (T_a, T_v) might not necessarily equal. Thus, we obtain distinct numbers of anchors for audio and visual modalities ($|\Psi_a|, |\Psi_v|$) to keep $T_a \cdot |\Psi_a|$ close to $T_v \cdot |\Psi_v|$ to balance the impact of each modality to the common pool of predictions. Similarly, the kernel sizes are determined by K-Means. We motivate it with an expectation that the receptive field will correspond to an event with a higher probability. We scale the resulting cluster centroids (in secs) by the feature time span to obtain values in grid-cell coordinates. Next, we round the values to the next odd integer for more elegant padding. Again, to preserve the balance in the share of predictions from each modality, we obtain an equal number of kernel sizes $K_a = K_v$ both modalities.

3.3 Training Procedure

Our model is trained in two stages: first, the captioning module is trained with ground truth proposals and, then, the proposal generator is trained using the pre-trained bi-modal encoder from the captioning model. Similar to [45] and [17], we optimize *KL-divergence* loss and apply *Label Smoothing* [43] to force a model to be less confident about predictions anticipating noisy annotations. Also, *masking* is used to ignore padding and prevent the model from attending to the next positions in the ground truth caption. During training of the event proposal generation module, all proposal generation heads for each modality are trained simultaneously summing up losses from all heads and both modalities. Each head uses YOLO-like loss: MSE for the localization losses (no square root) and *cross-entropy* for (no)objectness losses. The NMS is avoided for efficiency and to preserve the possibility of *dense* events. For the implementation details, a reader is referred to supplementary material (Sec. 6.3).

4 Experiments

We employ ActivityNet Captions dataset [20], which consists of 100k temporally localized sentences for 20k YouTube videos. The dataset is split into 50/25/25 % parts for training, validation, and testing. The validation set of videos is annotated by two different annotators. We report the results on the validation subsets as ground truth is not available for the testing set. Since the dataset is distributed as a set of links to YouTube videos, it is not possible to collect the whole dataset as some videos became unavailable. The authors also provide C3D features which are not suitable for our experimentation as they are missing audio information. In total, we had 91 % of the videos. We omit the unavailable videos from the validation

	RL	Full Dataset was Available	GT Proposals			Learned Proposals		
			B@3	B@4	M	B@3	B@4	M
Li <i>et al.</i> [22]	yes	yes	4.55	1.62	10.33	2.27	0.73	6.93
Xiong <i>et al.</i> [53]	yes	yes	–	–	–	2.84	1.24	7.08
Mun <i>et al.</i> [26]	yes	yes	4.41	1.28	13.07	2.94	0.93	8.82
Krishna <i>et al.</i> [20]	no	yes	4.09	1.60	8.88	1.90	0.71	5.69
Li <i>et al.</i> [22]	no	yes	4.51	1.71	9.31	2.05	0.74	6.14
Zhou <i>et al.</i> [62]	no	yes	5.76	2.71	11.16	2.91	1.44	6.91
Wang <i>et al.</i> [49]	no	yes	–	–	10.89	2.27	1.13	6.10
Mun <i>et al.</i> [26]	no	yes	–	–	–	–	–	6.92
Iashin <i>et al.</i> [17]	no	no	4.52	1.98	11.07	2.53	1.01	7.46
Rahman <i>et al.</i> [33]	no	no	3.04	1.46	7.23	1.85	0.90	4.93
Ours	no	no	4.63	1.99	10.90	3.84	1.88	8.44

Table 1: Comparison with state-of-the-art results on the dense video captioning task. The results are reported on the validation subset of ActivityNet Captions in both settings: captioning ground truth (GT) and learned proposals on BLEU@3–4 (B@3–4) and METEOR (M) metrics. For a fair comparison on METEOR, we additionally report the results of models without the reward (METEOR) maximization (RL) and indicate whether full dataset was available for training. The best and the 2nd best results are highlighted.

sets. We compared the results of other methods on the 91 % and 100 % of videos in Sec. 6.4.1 and observed similar performance suggesting the videos to be *missing completely at random*.

To evaluate the event proposal generation module we employ precision, recall, and mainly rely on F1-score (harmonic mean of precision and recall). While METEOR [7] and BLEU@3–4 [29] were used for captioning as they are highly correlated with human judgement. All metrics are averaged for every video and *temporal Intersection over Union* thresholds: [0.3, 0.5, 0.7, 0.9]. As it has been noted in [26], the original evaluation script had a critical issue which resulted in an incorrect evaluation of previous models. Therefore, we re-implement [49, 62] and compare with the results obtained with the corrected script.

4.1 Comparison to the State-of-the-art

We present the comparison between the bi-modal transformer with multi-headed proposal generator (Ours) and other methods in the existing literature [17, 20, 22, 26, 33, 49, 53, 62] on the dense video captioning task. The results of the comparison for captioning both ground truth (GT) and learned proposals are shown in Tab. 1. Since evaluating captioning is still challenging and METEOR is probably the best among other options, yet it only provides a *proxy* for how good a caption is. Therefore we believe that the direct optimization of METEOR using a reinforcement learning technique (RL) might not necessarily result in a better caption. To this end, we also include the results of [22, 26] without the RL module. Moreover, we obtained the results of [17] on the same subset of videos as we have since they additionally removed the videos with no speech modality from the evaluation.

According to the results, in the learned proposals setup, our dense video captioning model outperforms all of the models, which have no reward maximization on METEOR (no RL) while being on par when captioning ground truth proposals. Notably, our model has

	Full Dataset was Available	Prec.	Rec.	F1
Xiong <i>et al.</i> [53]	yes	51.41	24.31	33.01
Wang <i>et al.</i> [49]	yes	44.80	57.60	50.40
Zhou <i>et al.</i> [62]	yes	38.57	86.33	53.31
Mun <i>et al.</i> [26]	yes	57.57	55.58	56.56
Ours	no	48.23	80.31	60.27

Table 2: Comparison with state-of-the-art proposal generation methods on dense video captioning task. Results are reported on the validation set of ActivityNet Captions. Metrics: Precision, Recall, & F1-measure. The top-2 is highlighted.

the highest BLEU metrics in the learned proposal setup yet lies far away from [62] when captioning ground truth proposals on BLEU and performs on par with this model on METEOR.

Comparing to the RL methods, our model still outperforms them on BLEU metrics in both setups but loses in METEOR due to the absence of reward-maximization module. We draw the attention of a reader to the performance of [22] with and without the RL module — METEOR has dropped significantly yet other metrics remained on the same level.

Interestingly, we also outperform [17] who also use the transformer in multi-modal setup yet has more parameters (149M vs 51M). We note again that the results are not fair to neither of [17, 33] and ours since models have been trained on fewer videos.

Next, we compare our bi-modal multi-headed proposal generation module with other proposal generation modules from other dense video captioning models. The results for [62] and [49] are reported for 100 proposals per video. The results of the comparison are presented in Tab. 2. Despite our model being trained on fewer videos, our proposal generation model achieves state-of-the-art performance on the F1 metric. Specifically, our model provides impressive ground truth segment coverage while being accurate in its predictions.

4.2 Ablation Study

In this section, we show how the training procedure and modality impact the final results. The results are presented in Tab. 3 for both settings: captioning ground truth (performance of the captioning module) and leaned proposal (full dense video captioning model).

Training Procedures Our final model is trained in the following way. First, we train the captioning model on the ground truth proposal. Second, we freeze the weights on the encoder and train the proposal generator using the frozen encoder. The final results are obtained by captioning the proposals obtained from the trained proposal generator. Hence, the acronym “Cap \rightarrow Prop” which reads as: “the proposal generator is trained using the pre-trained encoder from the captioning module”. We compare this training procedure to other two methods: a) when both captioning and proposal generator modules are trained separately and b) when, first, the proposal module is trained and, then, the captioning module uses the pre-trained encoder with frozen weights during training. This is the opposite of the training procedure used for the final model, thus, abbreviated to “Prop \rightarrow Cap”.

Different Sets of Modalities The final model uses both audio and visual modalities to make predictions. We compare the performance of a bi-modal model with uni-modal ones. Specifically, for uni-modal settings, we employ the uni-modal transformer architecture similar to one in [17]. The difference between the hyper-parameters used for the final model and the uni-modal transformer is in the input dimension. For the uni-modal transformer, we follow the original paper where the input is first embedded into D_q dimension (see (14)) and

Training Procedure	Modality	GT Proposals			Learned proposals		
		B@3	B@4	M	B@3	B@4	M
Separately	Audio	2.85	1.14	8.81	2.50	1.11	6.89
	Visual	3.77	1.66	10.29	2.94	1.36	7.69
	Bi-modal	4.62	1.99	10.89	3.47	1.65	8.05
Prop \rightarrow Cap	Audio	2.59	0.99	8.81	2.23	0.93	6.88
	Visual	3.62	1.56	10.16	3.08	1.45	7.81
	Bi-modal	4.10	1.78	10.48	3.07	1.47	7.67
Cap \rightarrow Prop	Audio	2.85	1.14	8.81	2.58	1.15	6.98
	Visual	3.77	1.66	10.29	2.85	1.30	7.47
	Bi-modal	4.62	1.99	10.89	3.84	1.88	8.44

Table 3: The impact of training procedures and input modalities. We compare the training procedure of the final model when the proposal generator uses the pre-trained encoder on the captioning task (“Cap \rightarrow Prop”) to an opposite scenario (“Prop \rightarrow Cap”), and the situation when both of them are trained separately. The results are shown on validation sets of ActivityNet Captions when captioning ground truth (GT) and learned proposals.

remains the same everywhere later. We select 1024 for visual-only and 128 for audio-only transformers; the size of the pre-trained GloVe is projected with a FC layer to match the size.

Results We report every combination of the settings in Tab. 3. Specifically, we observed that the captioning module does not benefit from the pre-training for the proposal generation (“Prop \rightarrow Cap” vs “Cap \rightarrow Prop” & “Separate”). The results of the learned proposal setting show the importance of the pre-training but only in the “Cap \rightarrow Prop” setting. Overall, we claim that the captioning training does not benefit from utilizing the pre-trained proposal generator’s encoder and, even, performs worse with it. While, the proposal generator ends up with better performance if pre-trained captioning module’s encoder is used.

The comparison of the cross-modal performance shows that using both modalities (audio and visual) gives the best result in nearly all cases in both settings. However, it is shown that the audio modality is the *weakest* among the three implying that visual modality might contain a stronger signal for video understanding. Nevertheless, the gap between the visual-only and bi-modal case is consistent in all settings. This suggests that the audio still provides essential cues for dense video captioning. More ablations studies can be found in Sec. 6.4.

5 Conclusion

We believe that the handling of multiple modalities is under-explored in the computer vision community. In this paper, we present a novel bi-modal transformer with a bi-modal multi-headed proposal generation module showing how audio might facilitate the performance of dense video captioning. We perform our experimentation on the ActivityNet Captions dataset and achieve state-of-the-art results on F1 and BLEU metrics. The the ablation study results show that the proposed model provides an effective and elegant way of fusing audio and visual features while outperforming the uni-modal configurations in all settings.

Acknowledgments Funding for this research was provided by the Academy of Finland projects 327910 & 324346. We also acknowledge CSC – IT Center for Science, Finland, for computational resources.

References

- [1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization, 2016.
- [2] Hedi Ben-Younes, Rémi Cadene, Matthieu Cord, and Nicolas Thome. MUTAN: Multimodal tucker fusion for visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2612–2620, 2017.
- [3] Shyamal Buch, Victor Escorcia, Chuanqi Shen, Bernard Ghanem, and Juan Carlos Niebles. SST: Single-stream temporal action proposals. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2911–2920, 2017.
- [4] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017.
- [5] Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *NIPS 2014 Workshop on Deep Learning, December 2014*, 2014.
- [6] Pradipto Das, Chenliang Xu, Richard F Doell, and Jason J Corso. A thousand frames in just a few words: Lingual description of videos through latent topics and sparse object stitching. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2634–2641, 2013.
- [7] Michael Denkowski and Alon Lavie. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the ninth workshop on statistical machine translation*, pages 376–380, 2014.
- [8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina 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)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423.
- [9] Xuguang Duan, Wenbing Huang, Chuang Gan, Jingdong Wang, Wenwu Zhu, and Junzhou Huang. Weakly supervised dense event captioning in videos. In *Advances in Neural Information Processing Systems*, pages 3059–3069, 2018.
- [10] Victor Escorcia, Fabian Caba Heilbron, Juan Carlos Niebles, and Bernard Ghanem. DAPs: Deep action proposals for action understanding. In *European Conference on Computer Vision*, pages 768–784, 2016.
- [11] Zhe Gan, Chuang Gan, Xiaodong He, Yunchen Pu, Kenneth Tran, Jianfeng Gao, Lawrence Carin, and Li Deng. Semantic compositional networks for visual captioning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5630–5639, 2017.
- [12] Jort F Gemmeke, Daniel PW Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R Channing Moore, Manoj Plakal, and Marvin Ritter. Audio set: An ontology and human-labeled dataset for audio events. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 776–780, 2017.

- [13] Wangli Hao, Zhaoxiang Zhang, and He Guan. Integrating both visual and audio cues for enhanced video caption. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.
- [14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [15] Shawn Hershey, Sourish Chaudhuri, Daniel PW Ellis, Jort F Gemmeke, Aren Jansen, R Channing Moore, Manoj Plakal, Devin Platt, Rif A Saurous, Bryan Seybold, et al. Cnn architectures for large-scale audio classification. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 131–135, 2017.
- [16] Chiori Hori, Takaaki Hori, Teng-Yok Lee, Ziming Zhang, Bret Harsham, John R Hershey, Tim K Marks, and Kazuhiko Sumi. Attention-based multimodal fusion for video description. In *Proceedings of the IEEE international conference on computer vision*, pages 4193–4202, 2017.
- [17] Vladimir Iashin and Esa Rahtu. Multi-modal dense video captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 958–959, 2020.
- [18] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.
- [19] Atsuhiko Kojima, Takeshi Tamura, and Kunio Fukunaga. Natural language description of human activities from video images based on concept hierarchy of actions. *International Journal of Computer Vision*, 50(2):171–184, 2002.
- [20] Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. Dense-captioning events in videos. In *Proceedings of the IEEE international conference on computer vision*, pages 706–715, 2017.
- [21] Mun Wai Lee, Asaad Hakeem, Niels Haering, and Song-Chun Zhu. Save: A framework for semantic annotation of visual events. In *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 1–8, 2008.
- [22] Yehao Li, Ting Yao, Yingwei Pan, Hongyang Chao, and Tao Mei. Jointly localizing and describing events for dense video captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7492–7500, 2018.
- [23] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. SSD: Single shot multibox detector. In *European conference on computer vision*, pages 21–37, 2016.
- [24] Xiang Long, Chuang Gan, Gerard De Melo, Jiajun Wu, Xiao Liu, and Shilei Wen. Attention clusters: Purely attention based local feature integration for video classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7834–7843, 2018.

- [25] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. In *Proceedings of the IEEE international conference on computer vision*, pages 2630–2640, 2019.
- [26] Jonghwan Mun, Linjie Yang, Zhou Ren, Ning Xu, and Bohyung Han. Streamlined dense video captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6588–6597, 2019.
- [27] Andrew Owens and Alexei A Efros. Audio-visual scene analysis with self-supervised multisensory features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 631–648, 2018.
- [28] Yingwei Pan, Ting Yao, Houqiang Li, and Tao Mei. Video captioning with transferred semantic attributes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6504–6512, 2017.
- [29] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318, 2002.
- [30] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. PyTorch: An imperative style, high-performance deep learning library. In *Advances in neural information processing systems*, pages 8026–8037, 2019.
- [31] Wenjie Pei, Jiyuan Zhang, Xiangrong Wang, Lei Ke, Xiaoyong Shen, and Yu-Wing Tai. Memory-attended recurrent network for video captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8347–8356, 2019.
- [32] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.
- [33] Tanzila Rahman, Bicheng Xu, and Leonid Sigal. Watch, listen and tell: Multi-modal weakly supervised dense event captioning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 8908–8917, 2019.
- [34] Joseph Redmon and Ali Farhadi. YOLO9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271, 2017.
- [35] Joseph Redmon and Ali Farhadi. YOLOv3: An incremental improvement. *arXiv*, 2018.
- [36] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.
- [37] Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. Self-critical sequence training for image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7008–7024, 2017.

- [38] Anna Rohrbach, Atousa Torabi, Marcus Rohrbach, Niket Tandon, Christopher Pal, Hugo Larochelle, Aaron Courville, and Bernt Schiele. Movie description. *International Journal of Computer Vision*, 123(1):94–120, 2017.
- [39] Marcus Rohrbach, Wei Qiu, Ivan Titov, Stefan Thater, Manfred Pinkal, and Bernt Schiele. Translating video content to natural language descriptions. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 433–440, 2013.
- [40] Botian Shi, Lei Ji, Yaobo Liang, Nan Duan, Peng Chen, Zhendong Niu, and Ming Zhou. Dense procedure captioning in narrated instructional videos. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6382–6391, 2019.
- [41] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.
- [42] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8934–8943, 2018.
- [43] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.
- [44] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 4489–4497, 2015.
- [45] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.
- [46] Subhashini Venugopalan, Marcus Rohrbach, Jeffrey Donahue, Raymond Mooney, Trevor Darrell, and Kate Saenko. Sequence to sequence-video to text. In *Proceedings of the IEEE international conference on computer vision*, pages 4534–4542, 2015.
- [47] Subhashini Venugopalan, Huijuan Xu, Jeff Donahue, Marcus Rohrbach, Raymond Mooney, and Kate Saenko. Translating videos to natural language using deep recurrent neural networks. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1494–1504, Denver, Colorado, May–June 2015. Association for Computational Linguistics. doi: 10.3115/v1/N15-1173.
- [48] Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. Pointer networks. In *Advances in neural information processing systems*, pages 2692–2700, 2015.
- [49] Jingwen Wang, Wenhao Jiang, Lin Ma, Wei Liu, and Yong Xu. Bidirectional attentive fusion with context gating for dense video captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7190–7198, 2018.

- [50] Junbo Wang, Wei Wang, Yan Huang, Liang Wang, and Tieniu Tan. M3: Multimodal memory modelling for video captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7512–7520, 2018.
- [51] Xin Wang, Wenhui Chen, Jiawei Wu, Yuan-Fang Wang, and William Yang Wang. Video captioning via hierarchical reinforcement learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4213–4222, 2018.
- [52] Xin Wang, Yuan-Fang Wang, and William Yang Wang. Watch, listen, and describe: Globally and locally aligned cross-modal attentions for video captioning. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 795–801, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-2125.
- [53] Yilei Xiong, Bo Dai, and Dahua Lin. Move forward and tell: A progressive generator of video descriptions. In *Proceedings of the European Conference on Computer Vision*, pages 468–483, 2018.
- [54] Jun Xu, Ting Yao, Yongdong Zhang, and Tao Mei. Learning multimodal attention lstm networks for video captioning. In *Proceedings of the 25th ACM international conference on Multimedia*, pages 537–545, 2017.
- [55] Chenggang Yan, Yunbin Tu, Xingzheng Wang, Yongbing Zhang, Xinhong Hao, Yongdong Zhang, and Qionghai Dai. STAT: spatial-temporal attention mechanism for video captioning. *IEEE Transactions on Multimedia*, 22(1):229–241, 2019.
- [56] Li Yao, Atousa Torabi, Kyunghyun Cho, Nicolas Ballas, Christopher Pal, Hugo Larochelle, and Aaron Courville. Describing videos by exploiting temporal structure. In *Proceedings of the IEEE international conference on computer vision*, pages 4507–4515, 2015.
- [57] YouTube Data API. Video Categories. <https://developers.google.com/youtube/v3/docs/videoCategories>, [Accessed 1 November 2019].
- [58] Haonan Yu, Jiang Wang, Zhiheng Huang, Yi Yang, and Wei Xu. Video paragraph captioning using hierarchical recurrent neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4584–4593, 2016.
- [59] Hang Zhao, Chuang Gan, Andrew Rouditchenko, Carl Vondrick, Josh McDermott, and Antonio Torralba. The sound of pixels. In *Proceedings of the European conference on computer vision (ECCV)*, pages 570–586, 2018.
- [60] Yue Zhao, Yuanjun Xiong, Limin Wang, Zhirong Wu, Xiaoou Tang, and Dahua Lin. Temporal action detection with structured segment networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2914–2923, 2017.
- [61] Luwei Zhou, Chenliang Xu, and Jason J Corso. Towards automatic learning of procedures from web instructional videos. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

- [62] Luowei Zhou, Yingbo Zhou, Jason J Corso, Richard Socher, and Caiming Xiong. End-to-end dense video captioning with masked transformer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8739–8748, 2018.
- [63] Lingyu Zhu and Esa Rahtu. Separating sounds from a single image. *arXiv preprint arXiv:2007.07984*, 2020.
- [64] Lingyu Zhu and Esa Rahtu. Visually guided sound source separation using cascaded opponent filter network. *arXiv preprint arXiv:2006.03028*, 2020.

PUBLICATION

III

Taming visually guided sound generation

V. Iashin and E. Rahtu

In 32nd British Machine Vision Conference 2021, BMVA Press, 2021

Publication reprinted with the permission of the copyright holders.

Taming Visually Guided Sound Generation

Vladimir Iashin
vladimir.iashin@tuni.fi
Esa Rahtu
esa.rahtu@tuni.fi

Computing Sciences
Tampere University
Tampere, Finland

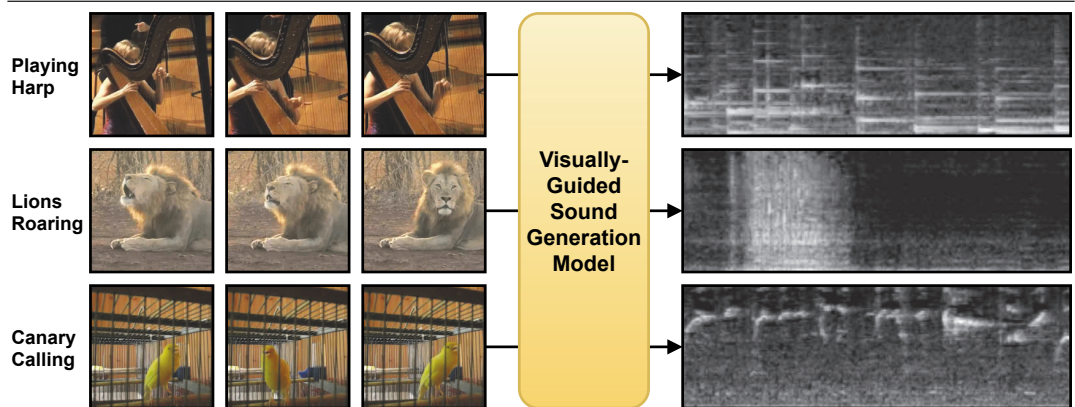


Figure 1: A single model supports the generation of visually guided, high-fidelity sounds for multiple classes from an open-domain dataset faster than the time it will take to play it.

Abstract

Recent advances in visually-induced audio generation are based on sampling short, low-fidelity, and one-class sounds. Moreover, sampling 1 second of audio from the state-of-the-art model takes minutes on a high-end GPU. In this work, we propose a single model capable of generating visually relevant, high-fidelity sounds prompted with a set of frames from open-domain videos in less time than it takes to play it on a single GPU.

We train a transformer to sample a new spectrogram from the pre-trained spectrogram codebook given the set of video features. The codebook is obtained using a variant of VQGAN trained to produce a compact sampling space with a novel spectrogram-based perceptual loss. The generated spectrogram is transformed into a waveform using a window-based GAN that significantly speeds up generation. Considering the lack of metrics for automatic evaluation of generated spectrograms, we also build a family of metrics called FID and MKL. These metrics are based on a novel sound classifier, called Melception, and designed to evaluate the fidelity and relevance of open-domain samples.

Both qualitative and quantitative studies are conducted on small- and large-scale datasets to evaluate the fidelity and relevance of generated samples. We also compare our model to the state-of-the-art and observe a substantial improvement in quality, size, and computation time. Code, demo, and samples: [v-iashin.github.io/SpecVQGAN](https://github.com/v-iashin/SpecVQGAN)

1 Introduction

A user-controlled sound generation has many applications for *e.g.* movie and music production. Currently, foley designers are required to search through large databases of sound effects to find a suitable sound for a scene. A less painstaking approach would be to auto-

matically generate a novel and relevant sound, given a few visual cues. Recent advances in deep learning brought to light many promising models for user-controlled content synthesis.

Previous works have proposed models to controllably generate *e.g.* images [13, 17, 35, 41, 44, 46, 50, 52, 64, 66, 67], videos [6, 12, 25, 34, 38, 42, 59, 60, 60, 63], and audios [1, 9, 15, 22, 24, 43, 57, 58], or separate sounds [18, 19, 69, 70, 74]. However, most of the audio works are music-related, and only a few attempts have been made to generate visually guided audio in an open domain setup [11, 73]. These methods rely on a one-model-per-class approach, which can be prohibitively expensive to scale to hundreds of classes.

Our goal in this paper is to build a single model that is capable of generating sounds conditioned on visual input from multiple classes with a restricted time budget. To address this, we propose to learn a prior in a form of the Vector Quantized Variational Autoencoder (VQVAE) codebook [61] and operate on spectrograms for efficiency. To shrink the sampling space more aggressively, we draw on advances in controlled image generation [17] relying on a variant of VQVAE with adversarial loss and introduce a novel spectrogram perceptual loss.

Such an approach allows us to reliably reconstruct a high-fidelity spectrogram from a smaller representation resolution. We, thus, can train a transformer on a shorter sequence to sample from the codebook and autoregressively construct a high-fidelity spectrogram while being conditioned on the visual cues. Finally, we vocode the spectrogram into a waveform using a variant of MelGAN [32] suitable for open-domain applications.

Human evaluation of content generation models is an expensive and tedious procedure. In the image generation field, this problem is bypassed with the automatic evaluation of fidelity using a family of metrics based on an ImageNet-pretrained [14] Inception model [56] *e.g.* Inception Score [53], Fréchet- [27] and Kernel Inception Distance [4] (FID & KID). The automatic evaluation of a sound generation model, however, remains an open question.

FID was adapted to assess fidelity of the generated audio in [30]. This metric is designed for very short sounds (<1 second) and, therefore, has limited applicability for long audio as it may miss long-term cues. Another challenge in the visually guided sound generation is to reliably estimate the relevance of produced samples. To mitigate both problems, we propose a family of metrics for fidelity and relevance evaluation based on a novel architecture called Melception, trained as a classifier on VGGSound [7], a large-scale open-domain dataset.

The main contributions of this work are: **(1)** a novel efficient approach for multi-class visually guided sound synthesis that relies on a transformer trained to sample from a codebook-based prior; **(2)** a new perceptual loss for spectrogram synthesis, called LPAPS. The loss relies on a novel general-purpose sound classifier, referred to as VGGish-ish, and helps VQVAE to learn reconstruction of higher-fidelity spectrograms from small-scale representations; **(3)** a novel set of metrics suitable for automatic evaluation of the fidelity and relevance of spectrogram synthesis, called Melception-based FID and MKL. We show the effectiveness of our approach in comparison with prior work and provide an extensive ablation study on small- and large-scale datasets (VAS and VGGSound) for visually guided sound synthesis.

2 Related Work

Codebook-based Content Generation The use of condensed prior information in a form of a codebook has been shown to effectively reduce the sampling space of generative algorithms. The initial idea was proposed in the seminal work [61] (VQVAE) and further improved in [51] (VQVAE-2). Applications of VQVAE for content generation include images [51, 61], audio [15, 37, 61, 71], and videos [49, 65]. Recently, it was found to be beneficial to train a transformer to sample from the codebook given a rich condition *e.g.*

text [16, 50], low-resolution image, semantic, edge, and depth-maps [17]. Our method, in contrast, is conditioned on a sequence of video frames and generates spectrograms.

Automatic Evaluation of Audio Synthesis While still being an open research question, few promising ideas have been proposed for the automatic evaluation of audio synthesis. Specifically, Kilgour *et al.* [30] adapted FID [27] to evaluate the fidelity of music enhancement algorithms. Unfortunately, the proposed method operates on 1-second windows and, therefore, does not utilize long-term cues. A similar approach was shown on a text-to-speech task in [5]. Alternatively, a model trained on human judgments has been employed as a perceptual loss during training [39]. However, collecting training material for a large-scale dataset poses significant budget requirements. In this paper, we propose a set of metrics designed to measure both the fidelity and relevance of prolonged open-domain spectrograms.

Instrument Music Generation With Visual Cues Generating short music audios became a testbed for many cross-modal generation algorithms. Owens *et al.* [45] pioneered the task by collecting a dataset of short videos containing hitting/scratching drumsticks against objects and used a combination of AlexNet [31] and LSTM [28] as a baseline. Chen *et al.* [9] focused on the generation of an image from the audio and vice-versa for single-instrument performance videos from the URMP dataset [36] using two Generative Adversarial Nets (GAN) [21] while Hao *et al.* [24] improved the performance of the GAN with cross-modal cycle-consistency [72]. Furthermore, Tan *et al.* [57] incorporated self-attention [62] into the GAN architecture and Su *et al.* [55] proposed to generate a piano sound by vocoding Midi predicted from a video. Recently, Kurmi *et al.* [33] brought a generation of short (1s) musical videos into the picture. These methods, however, focus on short (~ 1 second) music videos recorded in a controlled setting while our model operates on open-domain 10-second videos.

Open-domain Audio Generation Based on Visual Cues The generation of audio given a set of open-domain visual cues is a novel and challenging task. The first attempt to solve the task was published by Chen *et al.* [8] who proposed to employ a subset of AudioSet [20] to train a model to learn a residual to an average spectrogram for a video class. However, more relevant and higher-fidelity results were obtained by training a separate model for each video class. Namely, Zhou *et al.* [73] trained a separate SampleRNN [40] to generate a waveform for each of the 10 classes in the proposed dataset (VEGAS). Current state-of-the-art results in the generation of relevant and high-fidelity sounds for a video were shown by Chen *et al.* [11] (RegNet). They noticed the negative impact of “unseen” background sound on training dynamics and introduced a ground-truth-based regularizer and an enhanced version of the VEGAS dataset (VAS). While producing the most appealing results, the models are trained for each data class and the sampling speed is slow limiting the applicability of the model. In this paper, we propose a model that is capable of generating visually relevant sounds from videos of multiple classes in a time that is less than it takes to play the sound.

3 Framework

We aim to generate visually relevant and high-fidelity sounds. The main challenge is to design a model that handles videos of multiple categories and operates in real-time. Thus, we train a transformer to autoregressively compose a concise codebook representation of a spectrogram primed with a small set of frame-wise features obtained from a video (Sec. 3.2). The representation is then used in the pretrained codebook decoder to produce a spectrogram as outlined in Sec. 3.1. Finally, a waveform is reconstructed from the spectrogram using a pretrained vocoder as defined in Sec. 3.3. An overview of the architecture is shown in Fig. 2.

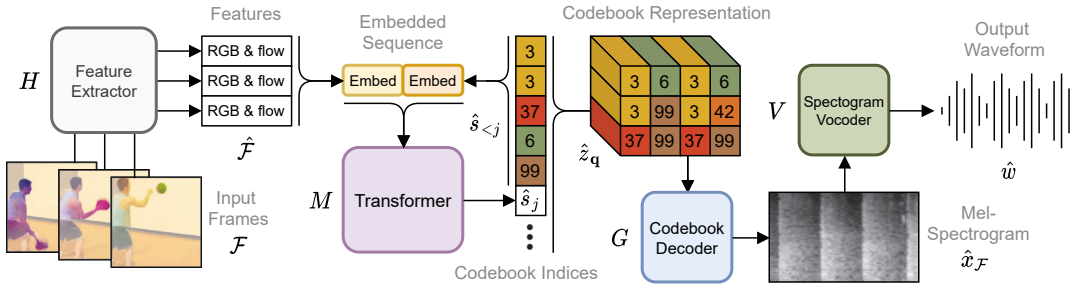


Figure 2: **Vision-based Conditional Cross-modal Autoregressive Sampler.** A transformer autoregressively samples the next codebook index given a sequence of visual features along with previously generated codebook indices. Once sampling is done, a sequence of generated indices is used to look up a pretrained codebook. Next, a pretrained codebook decoder is used to decode a spectrogram from a codebook representation. Finally, the generated spectrogram is turned into a waveform using a pretrained general-purpose spectrogram vocoder.

3.1 Perceptually-rich Spectrogram Codebook

The transformer requires the input to be represented as a sequence. A direct operation on wave samples or raw spectrogram pixels, however, quickly becomes intractable due to the quadratic nature of the dot-product attention. Alternatively, one could apply an encoder such as VQVAE [61] but the quantized bottleneck representation would be still infeasibly large. Our approach draws on VQGAN [17], an efficient autoencoder that allows decoding an image from a smaller-size representation than of VQVAE. To bridge the gap between image and audio signals, we operate on spectrograms and propose a new perceptual loss (LPAPS).

Spectrogram VQVAE Vector-Quantized Variational Autoencoder (VQVAE) [61] is trained to approximate an input using a compressed intermediate representation, retrieved from a discrete codebook. Our adaption of VQVAE, *Spectrogram VQVAE*, inputs a spectrogram $x \in \mathbb{R}^{F \times T}$ and outputs a reconstructed version of it $\hat{x} \in \mathbb{R}^{F \times T}$. First, the input x is encoded into a small-scale representation $\hat{z} = E(x) \in \mathbb{R}^{F' \times T' \times n_z}$ where n_z is the dimension of the codebook entries and $F' \times T'$ is a reduced frequency and time dimension. Next, the elements of the encoded representation \hat{z} are mapped onto the closest items in a codebook $\mathcal{Z} = \{z_k\}_{k=1}^K \subset \mathbb{R}^{n_z}$, forming a quantized representation $z_{\mathbf{q}} \in \mathbb{R}^{F' \times T' \times n_z}$:

$$z_{\mathbf{q}} = \mathbf{q}(\hat{z}) := \left(\arg \min_{z_k \in \mathcal{Z}} \|\hat{z}_{ft} - z_k\| \quad \text{for all } (f, t) \text{ in } (F' \times T') \right). \quad (1)$$

Since (1) is non-differentiable, we approximate the gradient by a straight-through estimator [2]. The reconstructed spectrogram \hat{x} is subsequently decoded from the codebook representation as $\hat{x} = G(z_{\mathbf{q}}) = G(\mathbf{q}(E(x)))$. The full VQVAE objective is defined by

$$\mathcal{L}_{\text{VQVAE}} = \underbrace{\|x - \hat{x}\|}_{\text{recons loss}} + \underbrace{\|E(x) - \text{sg}[z_{\mathbf{q}}]\|_2^2 + \beta \|\text{sg}[E(x)] - z_{\mathbf{q}}\|_2^2}_{\text{codebook loss}} \quad (2)$$

where sg is the stop-gradient operation that acts as an identity during the forward pass but has zero gradient at the backward pass.

The resolution of the intermediate codebook representation ($F' \times T'$) produced by VQVAE remains to be too large for a transformer to operate on. However, more suitable down-sampling rates, e.g. 1/16 of the input size, lead to poor reconstructions as shown in [17].

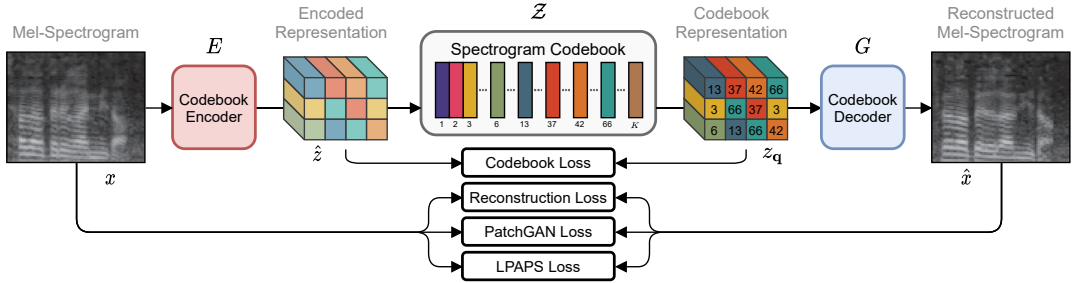


Figure 3: **Training Perceptually-Rich Spectrogram Codebook.** A spectrogram is passed through a 2D codebook encoder that effectively shrinks the spectrogram. Next, each element of a small-scale encoded representation is mapped to its closest neighbor from the codebook. A 2D codebook decoder is then used to reconstruct the input spectrogram. The training of the model is guided by codebook, reconstruction, adversarial, and LPAPS losses.

Spectrogram VQGAN and LPAPS VQGAN [17] is a version of VQVAE, extended with a patch-based adversarial loss [29] and perceptual loss (LPIPS) [68], that help to preserve the reconstruction quality when upsampled from a smaller-scale representation. Since the perceptual loss, used in the original VQGAN, relies on the ImageNet [14] pretrained VGG-16 [54], it is unreasonable to expect decent performance on sound spectrograms. Therefore, we introduce a novel way of guiding spectrogram-based audio synthesis, referred to as Learned Perceptual *Audio* Patch Similarity (LPAPS).

The closest relative of VGG-16 in audio classification is VGGish [26], which has the same capacity as VGG-9. However, we cannot directly build LPAPS on the pretrained VGGish or its architecture, since VGGish digests spectrograms with a rather short time span (< 1 second), while our application requires operating on spectrograms spanning up to 10 seconds. Moreover, the lack of depth and, therefore, downsampling operations prevents the model from extracting larger-scale features that could be useful in separating real and fake spectrograms. To address this, we train a variant of the VGG-16 architecture on the VGGSound dataset [7]. We refer to the obtained model as VGGish-ish.

Fig. 3 shows the training procedure for Spectrogram VQGAN with the final loss:

$$\mathcal{L}_{\text{SpecVQGAN}} = \mathcal{L}_{\text{VQVAE}} + \underbrace{\log D(x) + \log(1 - D(\hat{x}))}_{\text{patch-based adversarial loss}} + \underbrace{\sum_s \frac{1}{F^s T^s} \|x^s - \hat{x}^s\|_2^2}_{\text{LPAPS loss}}, \quad (3)$$

where D is a patch-based discriminator and $x^s, \hat{x}^s \in \mathbb{R}^{F^s \times T^s \times C^s}$ are features from real and fake spectrograms extracted at the s^{th} scale of VGGish-ish.

3.2 Vision-based Conditional Cross-modal Autoregressive Sampler

The sampler (transformer) is trained to sample a sequence of the codebook indices given a set of visual features. These should match the indices formed by the codebook encoder for the original audio. The conditional prediction of the next token can be formulated as a machine translation task and modeled by the vanilla Encoder-Decoder transformer architecture [62]. Alternatively, the problem can be defined in terms of language modeling, that is often approached with a Decoder-only transformer such as GPT [47]. In this paper, we employ a variant of GPT-2 [48] inspired by its success in autoregressive image synthesis [10, 17].

As outlined in Fig. 2, the sampling starts with the extraction of a sequence of features $\hat{\mathcal{F}} = \{\hat{f}_i\}_{i=1}^N \subset \mathbb{R}^{D_r + D_o}$ formed from a stack of RGB and optical flow frames $\mathcal{F} = \{f_i^r, f_i^o\}_{i=1}^N$.

The sequence of features $\hat{\mathcal{F}}$ is obtained by applying a frame-wise feature extractor H that consists of two pretrained models (for RGB and flow modalities) such that $\hat{\mathcal{F}} = H(\mathcal{F})$. Given a sequence of previously generated codebook indices $\hat{s}_{<j} = (\hat{s}_1, \hat{s}_2, \dots, \hat{s}_{j-1})$ along with the features $\hat{\mathcal{F}}$, an autoregressive step for the transformer M is defined by

$$p(s_j | \hat{s}_{<j}, \hat{\mathcal{F}}) = M([\hat{\mathcal{F}} : \hat{s}_{<j}]), \quad (4)$$

where $[\cdot]$ is a stacking operation and $p(s_j | \hat{s}_{<j}, \hat{\mathcal{F}}) \in [0, 1]^{n_z}$ is a probability distribution over all codebook indices. The next codebook index \hat{s}_j is sampled from the multinomial distribution with weights provided by p . The sampling is initialized at $j = 1$ and primed only with the input features $\hat{\mathcal{F}}$. Once $j = F' \cdot T'$, the sampling stops. The sequence of predicted codebook indices $\hat{\mathcal{S}} = \{\hat{s}_j\}_{j=1}^{F' \cdot T'}$ is used to lookup the codebook \mathcal{Z} so that, after unflattening, the codebook representation $\hat{z}_{\mathbf{q}} \in \mathbf{R}^{F' \times T' \times n_z}$ is formed. The transformer is trained with a typical cross-entropy loss, comparing the predicted codebook indices to those obtained from the ground truth spectrogram. Finally, given the codebook representation $\hat{z}_{\mathbf{q}}$, we decode a spectrogram $\hat{x}_{\mathcal{F}}$ using the decoder G pretrained during the codebook training stage (Sec. 3.1).

We note the importance of unflattening the sequence into a 2D form in a column-major way, precisely as shown in the middle part of Fig. 2, opposed to the row-major approach used for image synthesis [10, 17]. Employing the row-major unflattening during training restricts model applications as it would correspond to reconstructing the lower frequencies given the higher ones. Specifically, we found that a model trained this way produces poor samples when prompted with a few seconds of real audio.

3.3 Spectrogram Vocoder

During the final stage, a waveform \hat{w} is reconstructed from the decoded spectrogram using the pretrained vocoder V . Natural candidates for such vocoding are the Griffin-Lim algorithm [23] and WaveNet (used in prior work [11]). The Griffin-Lim procedure is fast, easy to implement, and it handles the diversity of an open-domain dataset. However, it produces low-fidelity results when operating on mel-spectrograms. In contrast, WaveNet provides high-quality results but remains to be relatively slow on test-time (25 mins per 10-sec sample on a GPU). For these reasons, we employ MelGAN [32] that is a non-autoregressive approach to reconstruct a waveform and, therefore, takes only 2 secs per sample on a CPU, while still achieving decent quality. Since MelGAN is originally trained for speech or music data, the pretrained models cannot be used in our open-domain scenario. Therefore, we train a MelGAN on the open-domain dataset (VGGSound).

3.4 Automatic Quality Assessment of Spectrogram-based Synthesis

Fidelity Our goal is to automatically evaluate both the fidelity and relevance of the generated samples. In the image generation domain, ImageNet pretrained InceptionV3 [56] is often used to form an opinion on the fidelity of the generated samples. Specifically, Inception Score [53] hypothesizes low entropy in conditional label distribution and high entropy on a marginal probability distribution for high-fidelity and diverse samples. More consistent evaluation results were achieved by computing Fréchet Distance between the distributions of pre-classification layer’s features of InceptionV3 between fake and real samples (FID) [27]. Considering the domain gap between spectrograms and RGB images, we adapt the Inception architecture for a spectrogram input size and train the model on the VGGSound dataset.

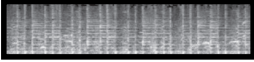
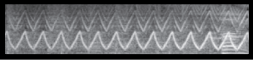
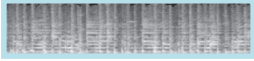
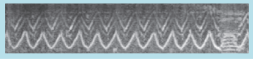
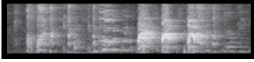
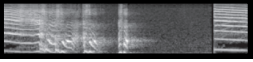
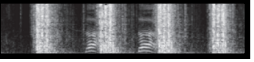
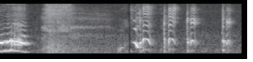

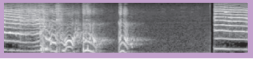
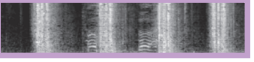

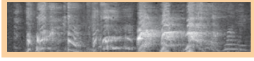
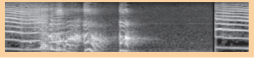
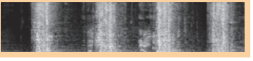
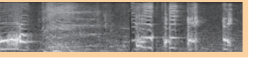
Trained on	Evaluated on	FID↓	MKL↓	Playing Jembe (VGGSound)	Ambulance Siren (VGGSound)
VGGSound	VGGSound	1.0	0.8		
VGGSound	VAS	3.2	0.7		
VAS	VAS	6.0	1.0		
					
					
					
					
					

Table 1: **Spectrogram VQGAN shows strong reconstruction ability on hold-out sets of VGGSound and VAS.** Metrics are Melception-based FID and mean MKL. On the top-right: ground truth reconstruction results for two classes are shown for a model trained on VGGSound. The bottom triplets show a comparison of VGGSound-trained and VAS-trained models on four classes from VAS. Adobe Reader can be used to listen for reconstructions.

Relevance Since Inception Score and FID metrics rely on dataset-level distributions, they are not suitable to assess the conditional content synthesis. To this end, we propose a metric, called MKL, that individually compares the distances between output distributions of fake and real audio associated with a condition (*e.g.* frames from a video). As the distance measure, we rely on KL-divergence and use the Melception classifier to build the distributions.

4 Experiments

VGGSound and VAS Datasets VAS dataset [11] consists of 12.5k \sim 6.73-second clips for 8 classes: *Dog, Fireworks, Drum, Baby, Gun, Sneeze, Cough, Hammer*. We follow the same train-test splitting procedure as [11] for a fair comparison. VGGSound dataset [7] consists of \sim 200k+ 10-second clips from YouTube spanning 309 classes with audio-visual correspondence. The classes can be grouped as *people, sports, nature, home, tools, vehicles, music*, etc. VGGSound is substantially larger, but less curated than VAS due to the automatic collecting procedure. We managed to download \sim 190k clips from the dataset as some of the videos were removed from YouTube. Our split is similar to the original with the exception that the train part is further split into train and validation. The validation split is formed to match the same number of *videos* per class as in the test set. As a result, we have 156.5k *clips* in the train, 19.1k in the validation, and 14.5k in the test sets. This splitting strategy is used across all training procedures including Melception, MelGAN, and VGGish-ish. To the best of our knowledge, we are the first to use the VGGSound dataset for sound synthesis.

Metrics The proposed model is evaluated in quantitative and qualitative studies. In quantitative evaluation, we rely on Melception-based metrics, namely MKL (averaged across the dataset) and FID for relevance and fidelity evaluation (as defined in Sec. 3.4).

Details We extract log mel-spectrograms of size 80×848 and 212 visual features with dimension $D_r = D_o = 1024$ from \sim 9.8-second videos before training. The codebook encoder and decoder are generic 2D Conv stacks with two extra attention layers before \hat{z} and after z_q . The downsampling and upsampling operations are parametrized. The variant of GPT-2 has 24 layers. Visual features and codebook indices are embedded to match the transformer dimension (1024). Training requires at least one 12GB GPU. See more in the supplementary.

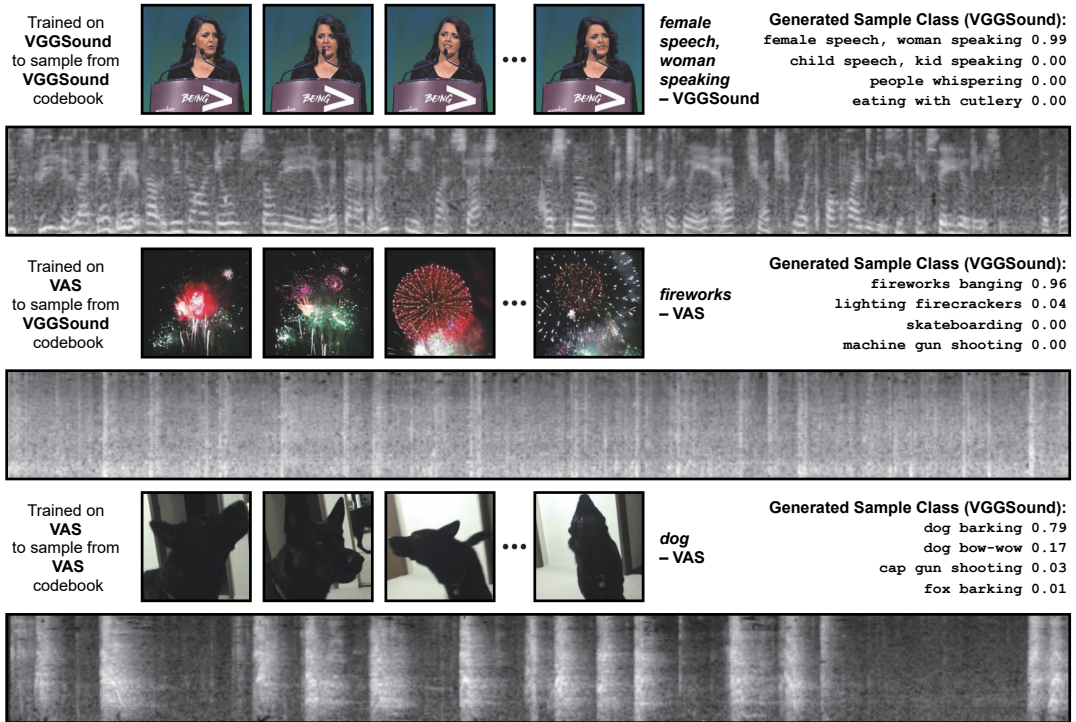


Figure 4: Samples produced by conditional cross-modal sampler are relevant and have high fidelity. The top row shows results of a model trained on VGGSound to sample from a VGGSound codebook (“from VGGSound for VGGSound”), the middle is “from VGGSound for VAS”, the bottom is: “from VAS to VAS”. An “opinion” of Melception is on the right.

4.1 Results

Reconstruction with Spectrogram VQGAN When compared to ground truth spectrograms, the reconstructions are expected to have high fidelity (low FID) and to be relevant (low mean MKL). Tab. 1 contains quantitative and qualitative results produced by our Spectrogram VQGAN (Sec. 3.1). The results imply high fidelity and relevance on a variety of classes from both VGGSound (test) and VAS (validation) datasets. Notably, the performance of the VGGSound-pretrained codebook is better than of the VAS-pretrained codebook even when applied on the VAS validation set due to larger and more diverse data seen during training. The implementation details and more examples are provided in the Supplementary. Moreover, in Tab. 2 we show the results of the ablation study on the impact of losses on reconstruction quality. In particular, the absence of the adversarial loss results in significant blurriness (which agrees with the findings in [17]) in reconstructed spectrograms and expected substantial downgrade in metrics.

Visually-Guided Sound Generation We benchmark the visually-guided sound generation qualitatively and quantitatively using three different settings: **a)** trained the transformer on VGGSound to sample from the VGGSound codebook, **b)** trained on VAS with the VGGSound codebook, and **c)** trained on VAS with the VAS codebook. Fig. 4 shows a few examples obtained with different settings along with the “opinion” of the Melception classifier on the generated sample label and in Tab. 3, we compare a different number of priming features including sampling without a condition (*No Feats*), which can be seen as the upper-bound on the relevance metric (mean MKL). The quantitative results are provided for two sets of ImageNet-pretrained features: BN-Inception (RGB + flow) and ResNet-50 (RGB).

GAN LPAPS		FID↓	MKL↓	Table 2: Adversarial and perceptual losses improve reconstruction results on the VGGSound test set.				
		130.4	9.6					
✓		1.4	1.1					
✓	✓	1.0	0.8					

Condition		FID↓	MKL↓	FID↓	MKL↓	FID↓	MKL↓	⊙↓
No Feats		13.5	9.7	33.7	9.6	28.7	9.2	7.7
ResNet	1 Feat	11.5	7.3	26.5	6.7	25.1	6.3	7.7
	5 Feats	11.3	7.0	22.3	6.5	20.9	6.1	7.9
	212 Feats	10.5	6.9	20.8	6.2	22.6	5.8	11.8
Inception	1 Feat	8.6	7.7	38.6	7.3	25.1	6.6	7.7
	5 Feats	9.4	7.0	29.1	6.9	24.8	6.2	7.9
	212 Feats	9.6	6.8	20.5	6.0	25.4	5.9	11.8
Codebook	VGGSound	VGGSound		VAS				
Sampling for	VGGSound	VAS		VAS				
Setting	(a)	(b)		(c)				

Table 3: **The number of features is an important factor for relevance and sampling speed on both datasets.** Fidelity and relevance are measured by FID and mean MKL, speed is in seconds to generate a ~ 10 -second audio sample.

We observe that: 1) In general, the more features from a corresponding video are used, the better the result in terms of relevance. However, there is a trade-off imposed by the sampling speed which decreases with the size of the conditioning. 2) A large gap (log-scale) in mean MKL between visual and “empty” conditioning suggests the importance of visual conditioning in producing relevant samples. 3) When the sampler and codebook are trained on the same dataset—settings (a) and (c)—the fidelity remains on a similar level if visual conditioning is used. This suggests that it is easier for the model to learn “features-codebook” (visual-audio) correspondence even from just a few features. However, if trained on different datasets (b), the sampler benefits from more visual information. 4) Both BN-Inception and ResNet-50 features achieve comparable performance, with BN-Inception being slightly better on VGGSound and with longer conditioning in each setting. Notably, the ResNet-50 features are RGB-only which significantly eases practical applications. We attribute the small difference between the RGB+flow features and RGB-only features to the fact that ResNet-50 is a stronger architecture than BN-Inception on the ImageNet benchmark [3]. See the technical details, more examples, ablations, and human studies in Supplementary Material.

Comparison with the state-of-the-art In Tab. 4, we compare our model to RegNet [11], which is currently the strongest baseline in generating relevant sounds for a visual sequence. For a fair comparison, we employ the same data preprocessing for audio and visual features as in RegNet [11]. We use the settings (b) & (c) (see Tab. 3) with 212 features in the condition, which is similar to the RegNet input. Since RegNet limits the sampling space explicitly by training a separate model for each class, it is difficult to fairly compare relevance with our model that is trained on all classes. To mitigate this to some extent, we include a class label into the transformer conditioning sequence allowing the model to learn to separate parameter subspaces for all 8 classes. The results suggest that our model produces higher quality spectrograms than RegNet, which is also supported by the lower FID scores. Moreover, RegNet uses two times more parameters. See more examples in the Supplementary Material.

	Ground Truth	RegNet	Ours		
gun (VAS)					
baby (VAS)					
		Params	FID↓	MKL↓	⊙↓
Ours (b)		379M	20.5	6.0	12
Ours (c)		377M	25.4	5.9	12
RegNet [11]	$8 \times 105M$	78.8	5.7	1500	
Ours (b) + cls		379M	20.2	5.7	12
Ours (c) + cls		377M	24.9	5.5	12

Table 4: **Compared to state-of-the-art, our model generates higher fidelity samples faster and with similar relevance w/ and w/o providing the class label.** RegNet size is multiplied by the num. of classes in VAS.

4.2 Qualitative Analysis of the Model Properties

We conduct a human study by single-handedly inspecting over 2k samples for test-set videos of the VGGSound dataset. Despite the biasedness of the study, we believe that the results are worth reporting. The samples are drawn for a random class and using the model trained on the VGGSound dataset with the VGGSound codebook (the setting (a), *5 Feats*, see Sec. 4.1). We divide our observations into three parts: **general properties of the model**, **problems with data preprocessing**, and **dataset-related issues** (see supplementary).

General Properties of the Model The proposed model supports multiple classes and, especially with some patience budget, generates relevant audio for the majority of classes in the VGGSound. The mistakes are not rare, but they are often associated with a poor audio-visual correspondence in the video or because the model generates a sound of another musical instrument instead of the specific one (*e.g.*, *violin* instead of *cello* – both are string instruments). However, the generation of a sample that belongs to a completely different class group is a rare event, *e.g.*, for a bird singing video the model will not generate an audio appropriate for indoor sports activities. We also observed, for classes such as *zebra braying*, *cat purring*, *pig oinking*, *bee*, *wasp*, *etc. buzzing*, *cattle mooing*, *alarm clock ringing*, the model struggles to produce a relevant sample possibly due to the unobservable source of the signal (*e.g.*, the flies are flying around the camera pointed to a tree and the flies are never captured but heard).

The model may confuse visually similar sounds, *e.g.*, *people whistling*, *singing*, *talking*, *whispering*, *burping*, *etc.* Also, if a video shows a close-up of hands, *e.g.*, *machine sewing*, the model may generate a sound of *keyboard typing* or *computer mouse clicking*. We also found that an ASMR setup (Autonomous Sensory Meridian Response) enforces the model to produce clean sounds similar to ASMR but often of a different class. The model struggles to differentiate different types of birds (*e.g.*, *swallow chickadee*, *pheasant*, *etc*) or hitting instruments (*e.g.*, *bongo*, *timbales*, *timpani*, *steelpan*, *etc*), yet it tends to produce the sounds of a similar class from, *e.g.*, another bird or instrument. These properties are expected from a model trained on a relatively noisy dataset with a vague separation between classes.

Data Preprocessing Issues After transformation into the mel-scale spectrogram, the audio signal loses the phase and a range of essential frequencies to differentiate sounds from some classes. For instance, by transforming the waveform into mel-scale spectrogram and back, we observed that the sound of *cat caterwauling* became indiscernible from *person sobbing*, *crying*, or *dog howling* classes. Although the speech segments are recognizable, the words are indecipherable. To this end, the model can be trained directly on top of the STFT spectrograms at the cost of efficiency during sampling, however.

5 Conclusion

We introduced a new efficient approach for multi-class visually-guided sound generation, which operates on spectrograms and relies on a prior in a form of a codebook representation. To train the prior, we proposed a new perceptual loss (LPAPS) which is based on a general-purpose classifier (VGGish-ish). This loss allows the model to learn to reconstruct higher-fidelity spectrograms from a small-scale representation. In addition, a novel automatic evaluation procedure is outlined to estimate both fidelity and relevance of generated spectrograms with a new family of metrics based on the Melception classifier. Our experiments on small- and large-scale datasets show the power and efficiency of our model in both quantitative and qualitative studies compared to the state-of-the-art.

Acknowledgments Funding for this research was provided by the Academy of Finland projects 327910 & 324346. We also acknowledge CSC – IT Center for Science, Finland, for computational resources.

References

- [1] Théis Bazin, Gaëtan Hadjeres, Philippe Esling, and Mikhail Malt. Spectrogram In-painting for Interactive Generation of Instrument Sounds. In *Joint Conference on AI Music Creativity*, 2020.
- [2] Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013.
- [3] Simone Bianco, Remi Cadene, Luigi Celona, and Paolo Napoletano. Benchmark analysis of representative deep neural network architectures. *IEEE Access*, 2018.
- [4] Mikolaj Binkowski, Danica J. Sutherland, Michael Arbel, and Arthur Gretton. Demystifying MMD GANs. In *ICLR*, 2018.
- [5] Mikolaj Binkowski, Jeff Donahue, Sander Dieleman, Aidan Clark, Erich Elsen, Norman Casagrande, Luis C. Cobo, and Karen Simonyan. High Fidelity Speech Synthesis with Adversarial Networks. In *ICLR*, 2020.
- [6] Caroline Chan, Shiry Ginosar, Tinghui Zhou, and Alexei A. Efros. Everybody Dance Now. In *ICCV*, 2019.
- [7] Honglie Chen, Weidi Xie, Andrea Vedaldi, and Andrew Zisserman. Vggsound: A Large-Scale Audio-Visual Dataset. In *ICASSP*, 2020.
- [8] Kan Chen, Chuanxi Zhang, Chen Fang, Zhaowen Wang, Trung Bui, and Ram Nevatia. Visually indicated sound generation by perceptually optimized classification. In *ECCV Workshops*, 2018.
- [9] Lele Chen, Sudhanshu Srivastava, Zhiyao Duan, and Chenliang Xu. Deep cross-modal audio-visual generation. In *Thematic Workshops of ACM Multimedia*, 2017.
- [10] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative Pretraining From Pixels. In *ICML*, 2020.
- [11] Peihao Chen, Yang Zhang, Mingkui Tan, Hongdong Xiao, Deng Huang, and Chuang Gan. Generating visually aligned sound from videos. *IEEE Transactions on Image Processing*, 2020.
- [12] Yang Chen, Yingwei Pan, Ting Yao, Xinmei Tian, and Tao Mei. Mocycle-GAN: Unpaired Video-to-Video Translation. In *ACM International Conference on Multimedia*, 2019.
- [13] Yunjey Choi, Min-Je Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. StarGAN: Unified Generative Adversarial Networks for Multi-Domain Image-to-Image Translation. In *CVPR*, 2018.
- [14] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009.
- [15] Prafulla Dhariwal, Heewoo Jun, Christine Payne, Jong Wook Kim, Alec Radford, and Ilya Sutskever. Jukebox: A generative model for music. *arXiv preprint arXiv:2005.00341*, 2020.

- [16] Ming Ding, Zhuoyi Yang, Wenyi Hong, Wendi Zheng, Chang Zhou, Da Yin, Junyang Lin, Xu Zou, Zhou Shao, Hongxia Yang, and Jie Tang. CogView: Mastering Text-to-Image Generation via Transformers. In *NeurIPS*, 2021.
- [17] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *CVPR*, 2021.
- [18] Chuang Gan, Deng Huang, Peihao Chen, Joshua B Tenenbaum, and Antonio Torralba. Foley music: Learning to generate music from videos. In *ECCV*. Springer, 2020.
- [19] Chuang Gan, Deng Huang, Hang Zhao, Joshua B. Tenenbaum, and Antonio Torralba. Music Gesture for Visual Sound Separation. In *CVPR*, 2020.
- [20] Jort F. Gemmeke, Daniel P. W. Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R. Channing Moore, Manoj Plakal, and Marvin Ritter. Audio Set: An ontology and human-labeled dataset for audio events. In *ICASSP*, 2017.
- [21] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Nets. In *NeurIPS*, 2014.
- [22] Gal Greshler, Tamar Rott Shaham, and Tomer Michaeli. Catch-A-Waveform: Learning to Generate Audio from a Single Short Example. In *NeurIPS*, 2021.
- [23] Daniel Griffin and Jae Lim. Signal estimation from modified short-time Fourier transform. *IEEE Transactions on acoustics, speech, and signal processing*, 1984.
- [24] Wang-Li Hao, Zhaoxiang Zhang, and He Guan. CMCGAN: A uniform framework for cross-modal visual-audio mutual generation. In *AAAI*, 2018.
- [25] Zekun Hao, Xun Huang, and Serge J. Belongie. Controllable Video Generation With Sparse Trajectories. In *CVPR*, 2018.
- [26] Shawn Hershey, Sourish Chaudhuri, Daniel P. W. Ellis, Jort F. Gemmeke, Aren Jansen, R. Channing Moore, Manoj Plakal, Devin Platt, Rif A. Saurous, Bryan Seybold, Malcolm Slaney, Ron J. Weiss, and Kevin W. Wilson. CNN architectures for large-scale audio classification. In *ICASSP*, 2017.
- [27] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. In *NeurIPS*, 2017.
- [28] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 1997.
- [29] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-Image Translation with Conditional Adversarial Networks. In *CVPR*, 2017.
- [30] Kevin Kilgour, Mauricio Zuluaga, Dominik Roblek, and Matthew Sharifi. Fréchet Audio Distance: A Metric for Evaluating Music Enhancement Algorithms. *arXiv preprint arXiv:1812.08466*, 2018.

- [31] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In *NeurIPS*, 2012.
- [32] Kundan Kumar, Rithesh Kumar, Thibault de Boissiere, Lucas Gestin, Wei Zhen Teoh, Jose Sotelo, Alexandre de Brébisson, Yoshua Bengio, and Aaron C. Courville. Mel-GAN: Generative Adversarial Networks for Conditional Waveform Synthesis. In *NeurIPS*, 2019.
- [33] Vinod K Kurmi, Vipul Bajaj, Badri N Patro, KS Venkatesh, Vinay P Namboodiri, and Preethi Jyothi. Collaborative Learning to Generate Audio-Video Jointly. In *ICASSP*. IEEE, 2021.
- [34] Alex X Lee, Richard Zhang, Frederik Ebert, Pieter Abbeel, Chelsea Finn, and Sergey Levine. Stochastic adversarial video prediction. *arXiv preprint arXiv:1804.01523*, 2018.
- [35] Cheng-Han Lee, Ziwei Liu, Lingyun Wu, and Ping Luo. MaskGAN: Towards Diverse and Interactive Facial Image Manipulation. In *CVPR*, 2020.
- [36] Bochen Li, Xinzhao Liu, Karthik Dinesh, Zhiyao Duan, and Gaurav Sharma. Creating a multitrack classical music performance dataset for multimodal music analysis: Challenges, insights, and applications. *IEEE Transactions on Multimedia*, 2018.
- [37] Xubo Liu, Turab Iqbal, Jinzheng Zhao, Qiushi Huang, Mark D Plumbley, and Wenwu Wang. Conditional Sound Generation Using Neural Discrete Time-Frequency Representation Learning. In *MLSP Workshop*, 2021.
- [38] Arun Mallya, Ting-Chun Wang, Karan Sapra, and Ming-Yu Liu. World-consistent video-to-video synthesis. In *ECCV*, 2020.
- [39] Pranay Manocha, Adam Finkelstein, Richard Zhang, Nicholas J. Bryan, Gautham J. Mysore, and Zeyu Jin. A differentiable perceptual audio metric learned from just noticeable differences. In *Interspeech*, October 2020.
- [40] Soroush Mehri, Kundan Kumar, Ishaan Gulrajani, Rithesh Kumar, Shubham Jain, Jose Sotelo, Aaron C. Courville, and Yoshua Bengio. SampleRNN: An Unconditional End-to-End Neural Audio Generation Model. In *ICLR*, 2017.
- [41] Takeru Miyato and Masanori Koyama. cGANs with Projection Discriminator. In *ICLR*, 2018.
- [42] Medhini Narasimhan, Shiry Ginosar, Andrew Owens, Alexei A Efros, and Trevor Darrell. Strumming to the Beat: Audio-Conditioned Contrastive Video Textures. In *CVPR Workshops*, 2021.
- [43] Javier Nistal, Stefan Lattner, and Gael Richard. Drumgan: Synthesis of drum sounds with timbral feature conditioning using generative adversarial networks. In *ISMIR*, 2020.
- [44] Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional Image Synthesis with Auxiliary Classifier GANs. In *ICML*, 2017.

- [45] Andrew Owens, Phillip Isola, Josh H. McDermott, Antonio Torralba, Edward H. Adelson, and William T. Freeman. Visually Indicated Sounds. In *CVPR*, 2016.
- [46] Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. Styleclip: Text-driven Manipulation of Stylegan Imagery. In *ICCV*, 2021.
- [47] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018.
- [48] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language Models are Unsupervised Multitask Learners. 2019.
- [49] Ruslan Rakhimov, Denis Volkhonskiy, Alexey Artemov, Denis Zorin, and Evgeny Burnaev. Latent video transformer. In *VISIGRAPP*, 2021.
- [50] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-Shot Text-to-Image Generation. In *ICML*, 2021.
- [51] Ali Razavi, Aäron van den Oord, and Oriol Vinyals. Generating Diverse High-Fidelity Images with VQ-VAE-2. In *NeurIPS*, 2019.
- [52] Scott E. Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative Adversarial Text to Image Synthesis. In *ICML*, *JMLR Workshop and Conference Proceedings*, 2016.
- [53] Tim Salimans, Ian J. Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved Techniques for Training GANs. In *NeurIPS*, 2016.
- [54] Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *ICLR*, 2015.
- [55] Kun Su, Xiulong Liu, and Eli Shlizerman. Audeo: Audio Generation for a Silent Performance Video. In *NeurIPS*, 2020.
- [56] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the Inception Architecture for Computer Vision. In *CVPR*, 2016.
- [57] Huadong Tan, Guang Wu, Pengcheng Zhao, and Yanxiang Chen. Spectrogram Analysis Via Self-Attention for Realizing Cross-Model Visual-Audio Generation. In *ICASSP*, 2020.
- [58] Maciej Tomczak, Masataka Goto, and Jason Hockman. Drum Synthesis and Rhythmic Transformation with Adversarial Autoencoders. In *ACM International Conference on Multimedia*, 2020.
- [59] Soumya Tripathy, Juho Kannala, and Esa Rahtu. FACEGAN: Facial Attribute Controllable rEenactment GAN. In *WACV*, 2021.
- [60] Sergey Tulyakov, Ming-Yu Liu, Xiaodong Yang, and Jan Kautz. MoCoGAN: Decomposing Motion and Content for Video Generation. In *CVPR*, 2018.
- [61] Aäron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural Discrete Representation Learning. In *NeurIPS*, 2017.

- [62] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In *NeurIPS*, 2017.
- [63] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Nikolai Yakovenko, Andrew Tao, Jan Kautz, and Bryan Catanzaro. Video-to-Video Synthesis. In *NeurIPS*, 2018.
- [64] Tao Xu, Pengchuan Zhang, Qiuyuan Huang, Han Zhang, Zhe Gan, Xiaolei Huang, and Xiaodong He. AttnGAN: Fine-Grained Text to Image Generation With Attentional Generative Adversarial Networks. In *CVPR*, 2018.
- [65] Wilson Yan, Yunzhi Zhang, Pieter Abbeel, and Aravind Srinivas. VideoGPT: Video Generation using VQ-VAE and Transformers. *arXiv preprint arXiv:2104.10157*, 2021.
- [66] Han Zhang, Tao Xu, and Hongsheng Li. StackGAN: Text to Photo-Realistic Image Synthesis with Stacked Generative Adversarial Networks. In *ICCV*, 2017.
- [67] Han Zhang, Ian J. Goodfellow, Dimitris N. Metaxas, and Augustus Odena. Self-Attention Generative Adversarial Networks. In *ICML*, 2019.
- [68] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *CVPR*, 2018.
- [69] Hang Zhao, Chuang Gan, Andrew Rouditchenko, Carl Vondrick, Josh McDermott, and Antonio Torralba. The sound of pixels. In *ECCV*, 2018.
- [70] Hang Zhao, Chuang Gan, Wei-Chiu Ma, and Antonio Torralba. The Sound of Motions. In *ICCV*, 2019.
- [71] Yi Zhao, Haoyu Li, Cheng-I Lai, Jennifer Williams, Erica Cooper, and Junichi Yamagishi. Improved Prosody from Learned F0 Codebook Representations for VQ-VAE Speech Waveform Reconstruction. In *Interspeech*, 2020.
- [72] Tinghui Zhou, Philipp Krähenbühl, Mathieu Aubry, Qi-Xing Huang, and Alexei A. Efros. Learning Dense Correspondence via 3D-Guided Cycle Consistency. In *CVPR*, 2016.
- [73] Yipin Zhou, Zhaowen Wang, Chen Fang, Trung Bui, and Tamara L. Berg. Visual to Sound: Generating Natural Sound for Videos in the Wild. In *CVPR*, 2018.
- [74] Lingyu Zhu and Esa Rahtu. Visually guided sound source separation using cascaded opponent filter network. In *ACCV*, 2020.

PUBLICATION

IV

Sparse in space and time: Audio-visual synchronisation with trainable selectors

V. Iashin, W. Xie, E. Rahtu, and A. Zisserman

In 33rd British Machine Vision Conference 2022, BMVA Press, 2022

Publication reprinted with the permission of the copyright holders.

Sparse in Space and Time: Audio-visual Synchronisation with Trainable Selectors

Vladimir Iashin¹

vladimir.iaschin@tuni.fi

Weidi Xie^{2,3}

weidi@robots.ox.ac.uk

Esa Rahtu¹

esa.rahtu@tuni.fi

Andrew Zisserman³

az@robots.ox.ac.uk

¹ Computing Sciences

Tampere University

Tampere, Finland

² Coop. Medianet Innovation Center

Shanghai Jiao Tong University

Shanghai, China

³ Visual Geometry Group

Department of Engineering Science

University of Oxford

Oxford, UK

Abstract

The objective of this paper is audio-visual synchronisation of general videos ‘in the wild’. For such videos, the events that may be harnessed for synchronisation cues may be spatially small and may occur only infrequently during a many seconds-long video clip, *i.e.* the synchronisation signal is ‘sparse in space and time’. This contrasts with the case of synchronising videos of talking heads, where audio-visual correspondence is dense in both time and space.

We make four contributions: *(i)* in order to handle longer temporal sequences required for sparse synchronisation signals, we design a multi-modal transformer model that employs ‘selectors’ to distil the long audio and visual streams into small sequences that are then used to predict the temporal offset between streams. *(ii)* We identify artefacts that can arise from the compression codecs used for audio and video and can be used by audio-visual models in training to artificially solve the synchronisation task. *(iii)* We curate a dataset with only sparse in time and space synchronisation signals; and *(iv)* the effectiveness of the proposed model is shown on both dense and sparse datasets quantitatively and qualitatively. Project page: v-iaschin.github.io/SparseSync

1 Introduction

Audio-visual synchronisation is the task of determining the temporal offset between the audio (sound) and visual (image) streams in a video. In recent literature, this task has been explored by exploiting strong correlations between the audio and visual streams, *e.g.* in human speech [2, 8, 10] and playing instruments [3, 23], to provide a training signal for deep neural networks. In such scenarios, effective signals for synchronisation can be discovered between the lip or body movements and audio at almost every second. Despite the tremendous success achieved by these methods, for the most part, existing models are still limited to specialised domains, and not directly applicable to general (non-face, non-music) classes.

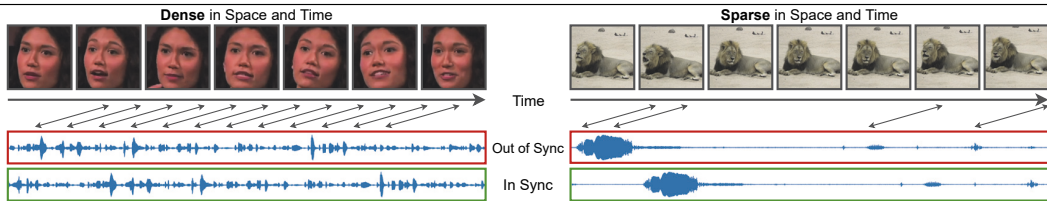


Figure 1: Audio-visual synchronisation requires a model to relate changes in the visual and audio streams. Open-domain videos often have a small visual indication, *i.e.* sparse in space. Moreover, cues may be intermittent and scattered, *i.e.* sparse across time, *e.g.* a lion only roars once during a video clip. This differs from a tight face crop of a speaker where cues are dense in space and time.

Our goal in this paper is to develop the next-generation audio-visual (AV) synchroniser. Rather than focusing on a specialised domain, such as human speech, we explore architectures for AV synchronisation for videos of general thematic content, *e.g.* daily videos [6, 20, 22] and live sports [13]. A solution for this task would be extremely useful for a number of applications that improve a user’s viewing experience – in order to avoid or at least automatically detect AV synchronisation offsets. Applications such as video conferencing, television broadcasts, and video editing, are currently largely done by ‘off-line’ measurements or heavy manual processing [11, 26, 28].

However, upgrading the existing audio-visual synchronisation systems to general videos is not straightforward, due to the following challenges: (i), in general videos, the synchronisation signal is often *sparse and instantaneous in time*, (a lion roaring or a tennis volley), rather than *dense in time* (a recorded monologue); (ii), objects that emit sounds can vary in size or appear in the distance making their presence on the frame small or *sparse in space* (a ball being hit in tennis), whereas synchronisation of a talking-head video may rely on visual cues from the localised mouth region, *i.e.* *dense in space*; (iii), some sound sources do not have a useful visual signal for synchronisation, *e.g.* stationary sounds (a car engine or electric trimmer), ambient sounds (wind, water, crowds, or traffic), and off-screen distractors (commentary track or advertisements); (iv), video encoding algorithms compress unperceived redundancy of a signal, this, however, can introduce artefacts that may lead to a trivial solution when training for audio-visual synchronisation; lastly, (v) due to its challenging nature, a public benchmark to measure progress has not yet been established.

In this paper, (i) we introduce a novel multi-modal transformer architecture, **SparseSelector**, that can digest long videos with linear scaling complexity with respect to the number of input tokens, and predict the temporal offset between the audio and visual streams. We achieve this by using a set of learnable *queries* to select informative signals from the ‘sparse’ video events across a wide time span. (ii) We show that for specific common audio and visual coding standards, a model can detect compression artefacts during training. We present a few simple indicators to determine if a model has learnt using these artefacts, as well as suggest several ways to mitigate the problem. Specifically, for the RGB stream, we recommend avoiding the MPEG-4 Part 2 codec, as well as reducing the sampling rate for audio. (iii) Additionally, to measure the progress of audio-visual synchronisation on general thematic content, we curate a subset of VGGSound with ‘sparse’ audio-visual correspondence called VGGSound-Sparse. We validate the effectiveness of the new model with thorough experiments on the existing lip reading benchmark (LRS3) and natural videos from VGGSound-Sparse and demonstrate state-of-the-art performance.

2 Related Work

Audio-visual synchronisation. During the pre-deep-learning era, the audio-visual human face synchronisation models relied on manually crafted features and statistical models [16, 27]. With the advent of deep learning, [9] introduced a two-stream architecture that was trained in a self-supervised manner using a binary contrastive loss. Later improvements were brought by multi-way contrastive training [10], and Dynamic Time Warping [24] used by [14]. Khosravan *et al.* [20] demonstrated the benefits of spatio-temporal attention and Kim *et al.* [21] employed a cross-modal embedding matrix to predict the offset for synchronisation. The progress was followed by [18] who introduced an architecture called VocaLiST with three transformer decoders [29]: two that cross-attend individual modalities and a third that fuses the outputs of the first two. These methods achieve impressive performance but focus on human speech rather than open-domain videos.

Although audio-visual synchronisation of general classes is a novel task, a few promising attempts have been made. In particular, Casanovas *et al.* [5] studied a handful of different scenes captured from a set of cameras. More recently, Chen *et al.* [7] adapted the transformer architecture and used a subset of VGGSound [6] covering 160 classes. In contrast to prior work, we focus on more challenging classes that have ‘sparse’ rather than ‘dense’ synchronisation signals.

Video coding artefacts. Since the early work of Doersch *et al.* on self-supervision [12], it has been known that network training can find shortcuts. Similarly, shortcuts due to video editing and coding artefacts have been noted in Wei *et al.* [30] and Arandjelović *et al.* [3]. In particular, [30] tackled the arrow-of-time in videos and studied artificial cues caused by black regions on video frames. While [3] noticed a slight impact of MPEG-encoding on audio-visual correspondence training and attributed it to the way negative samples are picked with respect to the start time of a positive sample. In this work, we study the ways to easily spot that the data contains artificial signals, as well as provide a few recommendations on how to prevent leaking such artefacts into data.

3 SparseSelector: an Audio-visual Synchronisation Model

In this section, we describe our audio-visual synchronisation model, where the audio-visual correspondence may only be available at sparse events in the ‘in the wild’ videos. This requires the model to handle longer video clips so that there is a high probability that a synchronisation event will occur. To this end, we propose *SparseSelector*, a transformer-based architecture that enables the processing of long videos with linear complexity with respect to the duration of a video clip. It achieves this by ‘compressing’ the audio and visual input tokens into two small sets of learnable *selectors*. These selectors form an input to a transformer which predicts the temporal offset between the audio and visual streams.

Architecture overview. The overview of the model is shown in Fig. 2. Given an audio spectrogram $\mathcal{A} \in \mathbb{R}^{H_a \times W_a \times 1}$ (H_a , W_a are frequency and time dimensions) and a stack of RGB frames $\mathcal{V} \in \mathbb{R}^{T_v \times H_v \times W_v \times 3}$, the audio-visual synchronisation model outputs the offset Δ between audio (\mathcal{A}) and visual (\mathcal{V}) streams:

$$\Delta = \Phi_{\text{Sync}} \left(\Phi_{\text{A-Sel}} \left(\Phi_{\text{A-Feat}}(\mathcal{A}) \right), \Phi_{\text{V-Sel}} \left(\Phi_{\text{V-Feat}}(\mathcal{V}) \right) \right). \quad (1)$$

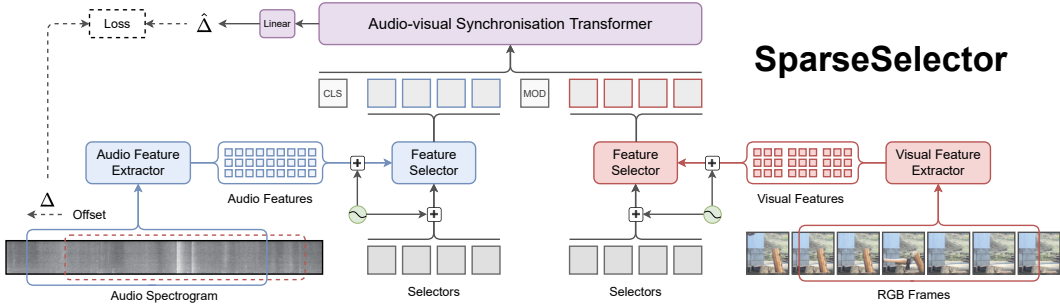


Figure 2: **An overview of SparseSelector.** The input is a spectrogram of the audio waveform and RGB frames from the video stream. These are passed through corresponding feature extractors, and the resulting features are refined with trainable selectors that ‘pick’ useful cues for synchronisation. As a result, the synchronisation transformer operates on substantially shorter sequences than the original input. The visual and audio selector queries are concatenated with classification (CLS) and separation tokens (MOD) as input to the transformer. Finally, the CLS token of the transformer output is used to predict the audio offset using a linear classification head. RGB frames are zoomed-in for visualisation purposes. The model is trained by off-setting the audio spectrogram. Dashed lines illustrate train-time behaviour.

First, audio and visual streams are independently encoded in feature extraction modules $\Phi_{A/V-Feat}$. Next, trainable *selectors* are passed to *Feature Selectors* ($\Phi_{A/V-Sel}$) along with the encoded features where they ‘summarise’ informative signals from the features that contain ‘sparse’ information across time and space. Finally, the selectors are used in *Synchronisation Transformer* (Φ_{Sync}) to predict the temporal offset Δ between audio and visual streams.

Feature encoding. Audio & visual inputs are encoded in spatio-temporal feature extractors:

$$a = \Phi_{A-Feat}(\mathcal{A}) \in \mathbb{R}^{h_a \times w_a \times d_a}, \quad v = \Phi_{V-Feat}(\mathcal{V}) \in \mathbb{R}^{t_v \times h_v \times w_v \times d_v}, \quad (2)$$

where t , h , w , and d denote time, height, width and channel dimensions, respectively. For the audio backbone, we use a variant of ResNet18 [15], which we pre-train on VGGSound [6] for sound classification. As for the visual backbone, we adopt S3D [31] pre-trained for action recognition on Kinetics 400 [19]. Although the setting allows employing any visual recognition network, we found that training a synchronisation model with a frame-wise feature extractor was significantly more difficult.

Feature Selectors. To utilize sparsely occurring synchronisation cues, the model should be able to handle longer input sequences. Moreover, accurate synchronisation requires a higher visual frame rate than other video understanding tasks (*e.g.* action recognition), which further increases the input size. For this reason, drawing on the idea of trainable queries [4, 17, 32], we propose to use a small number of trainable ‘selectors’ that learn to attend to the most useful modality features for synchronisation and, thus, reducing sequence length.

The architecture of the *Feature Selector* is similar to the transformer decoder [29]. Specifically, we start by flattening audio and visual features into sequences $a \in \mathbb{R}^{h_a w_a \times d_a}$ and $v \in \mathbb{R}^{t_v h_v w_v \times d_v}$. After adding trainable positional encoding (PE_*) for each dimension, trainable selectors and modality features are passed to the separate Feature Selectors as follows:

$$\hat{q}_a = \Phi_{A-Sel}(a + PE_a, q_a + PE_{q_a}), \quad \hat{q}_v = \Phi_{V-Sel}(v + PE_v, q_v + PE_{q_v}), \quad (3)$$

where $q_a, \hat{q}_a \in \mathbb{R}^{k_a \times d}$ and $q_v, \hat{q}_v \in \mathbb{R}^{k_v \times d}$ while $k_{a/v}$ are the numbers of selectors.

Note that the selectors provide a ‘short summary’ of the context features through the cross-attention mechanism while making the memory footprint more manageable. The reduced memory requirement is a consequence of (a) casting the complexity from quadratic to linear w.r.t. the input length, and (b) setting $k_v \ll t_v \cdot h_v \cdot w_v$ and $k_a \ll h_a \cdot w_a$. The number of selectors (k_v or k_a) can be conveniently tweaked according to the memory budget.

Audio-visual synchronisation transformer. To fuse the audio-visual cues from individual selectors, we adopt the standard transformer encoder layers to jointly process them, and to predict the offset, *i.e.* relative temporal shift between audio and visual streams, as follows:

$$\hat{\Delta} = \Phi_{\text{Sync}}([\text{CLS}; q_v; \text{MOD}; q_a]) \quad (4)$$

Here we concatenate the visual-audio selectors with two learnable special tokens, namely the classification token [CLS], and the modality token [MOD] that separates the two modalities. The offset prediction is obtained by applying a linear prediction head on the first token of the output sequence (omitted from Eq. (1) and (4) for clarity).

Training procedure. We assume that the majority of videos in the public datasets are synchronised to a good extent. With this assumption, we can artificially create temporal offsets between audio and visual streams from a video. We formulate the audio-visual synchronisation as a classification task onto a set of offsets from a pre-defined temporal grid space as $[-2.0, -1.8, \dots, 0.0, +0.2, \dots, +2.0]$ sec. The step size is motivated by the ± 0.2 sec human tolerance, where the ITU performed strictly controlled tests with expert viewers and found that the threshold for acceptability is -0.19 sec to $+0.1$ sec [25]. To train the model, we employ the cross-entropy loss. For our experiments, we randomly trim a 5-sec segment out of 9 seconds such that both audio and visual streams are within the 9-second clip to make inputs of the same size and avoid padding that could hint if the input is off-sync.

4 Avoiding Temporal Artefacts

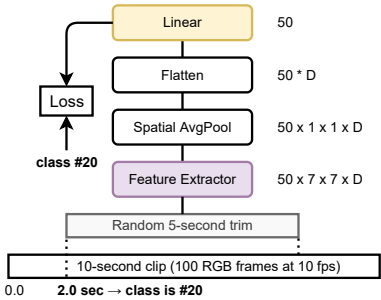
In this section, we describe our discovery of trivial solutions for training audio-visual synchronisation, that is, the model is able to exploit the video compression artefacts, to infer the time stamp for the specific video clip. Additionally, we also detail a suite of techniques that allows us to probe the artefacts and provide some practical suggestions to avoid them.

4.1 Identifying Temporal Artefact Leakage

We present two ways of identifying the temporal artefact leakage. In particular, training to predict the start time of a temporal crop (discussed next) and tracking metrics with temporal tolerance (discussed in the supplementary material).

Training to predict a video clip’s time stamp. A synchronisation model should rely solely on temporal positions of conceptual cues instead of, what we call, temporal artefacts. To check if data is polluted with artefacts, we suggest training a model to predict the start time of a random trim of an available video clip as shown in Tab. 1, left. Of course, it should not be possible to determine the start time of the trim in the original clip from the trim itself, and a network trained for this task should achieve only chance performance. However, for some audio and video codecs, the performance is far higher indicating artefact leakage.

The start-time classifier is a simple feature extractor (ResNet18). It is trained on three variations of the MJPEG-AoT dataset [30] obtained from the Vimeo streaming service: original ProRes videos, and ProRes videos transcoded into either MPEG-4 Pt. 2 (aka. mpeg4)



Codec	Acc@1	Acc@5
MPEG-4 Part 2 (mpeg4)	27.2	77.1
MPEG-4 Part 10 (H.264)	2.5	11.9
ProRes	2.7	13.4
AAC @ 44100Hz	86.7	100.0
AAC @ 22050Hz	23.0	74.3
AAC @ 16000Hz	6.3	19.3
Lossless @ 22050Hz	2.9	14.6

Table 1: **Commonly used coding standards may leak temporal artefacts – it is easy to test.** *Left:* a simple architecture trained to classify the start of a trim from a 10s clip to a pre-defined 0.1 sec-step grid (here for RGB). *Right:* Accuracy comparison for RGB and audio stream codecs predicting the start of an RGB or audio trim. Metrics are accuracy at 1 and 5 on 50 classes. Chance performance is 2 and 10 %. The higher accuracies indicate that an artefact is being used – see text for discussion.

or MPEG-4 Pt. 10 (aka. H.264). Note, frames in ProRes are compressed independently from others. If the visual stream of the video is encoded using `mpeg4`, the model trained to predict the start of the trim can do it significantly beyond a chance performance (Tab. 1, top-right).

Similarly, Advanced Audio Coding (AAC) might also leak temporal cues to the audio signal (Tab. 1, bottom-right). Since it is challenging to find a large set of videos with lossless audio compression, we used audio of randomly generated noise with a specified sampling rate and saved it to a disk losslessly (PCM) to obtain the performance with lossless compression. To obtain results on AAC, we transcoded these files to AAC with `ffmpeg`.

4.2 Preventing Temporal Artefact Leakage

Avoiding MPEG-4 Part 2 in favour of H.264. The algorithm that selects key-frames in MPEG-4 Part 2 (`mpeg4`) is less flexible than the one of H.264. In particular, `ffmpeg`, which is commonly used in practice, by default, encodes key frames every 12 frames. This means that each of the following 11 frames is merely a residual of the key frame and it is noticeable on the RGB stream (as we show in the supplementary). Such a temporal regularity can be picked up by a model and used to solve the task relying mostly on these artefacts. In contrast, each frame encoded by H.264 can reference up to 16 key-frames, which can be allocated more sparsely and their presence depends heavily on the scene rather than a rather strict interval as in MPEG-4 Pt. 2. This benefit is apparent when training a model to predict the start of a trim (see Tab. 1, top-right). A potential solution would be to avoid *inter*-frame codecs (`mpeg4` and H.264) in favor of an *intra*-frame codec (e.g. MJPEG, ProRes). However, this is a strong requirement for research datasets because it requires avoiding YouTube which stores videos compressed with inter-frame codecs (H.264 or VP9, according to view count).

Reducing audio sampling rate. There is a substantial difference in the model’s ability to predict the start of a trim depending on the sampling rate of the audio track (Fig. 1, bottom-right). While the reason behind the temporal artefacts in AAC is unknown, we recommend avoiding higher sampling rates. In our experiments, we rely on a 16kHz sampling rate as it provides a reasonable trade-off between audio quality and the start prediction performance. Ultimately one would want to have a dataset with lossless audio tracks yet, again, it is a strong requirement for a dataset as it is commonly used by YouTube.

5 Experiments

Dense in time dataset. The dataset is Lip Reading Sentences (LRS3) [1] which is obtained from TED talks for many speakers. We use two variations of the dataset. The first employs strict rectangular face crop coordinates that are extended to make a square (‘dense in time and space’). The second variation consists of full-frame videos without cropping (‘spatially sparse and temporally dense’). The raw videos are obtained from YouTube with RGB (25fps, H.264) and audio (16kHz, AAC) streams and referred to as ‘LRS3-H264’ and ‘LRS3-H264 (“No face crop”)’. We utilise the `pretrain` subset and split video ids into 8:1:1 parts for train, validation, and test sets. Only videos longer than 9 sec are used to unify it with the sparse dataset (discussed next). In total, we use $\sim 58k$ clips from $\sim 4.8k$ videos.

Sparse in time dataset. The dataset uses VGGSound [6] which consists of 10s clips collected from YouTube for 309 sound classes. A subset of ‘temporally sparse’ classes is selected using the following procedure: 5–15 videos are randomly picked from each of the 309 VGGSound classes, and manually annotated as to whether audio-visual cues are only sparsely available. After this procedure, 12 classes are selected ($\sim 4\%$) or 6.5k and 0.6k videos in the train and test sets, respectively (for class names see Fig. 3). Next, the second round of manual verification of a different subset of 20 videos from each class determines if it is feasible to align the sound based on the visual content. It is observed that $\sim 70\%$ of these video clips are synchronisable. We refer to this dataset as *VGGSound-Sparse*.

Baseline. Drawing on architectural details proposed in [7], we design a baseline as a transformer decoder that uses audio features as *queries* and visual features as context (*keys* and *values*) to predict the offset. The audio features are pooled across the spectrogram frequency dimension and trained from scratch. Apart from that, the feature extractors resemble ours.

Offset grid. We define the synchronisation task as classifying the offset on a 21-class grid ranging from -2 to $+2$ seconds with the 0.2-sec step size, as explained in Sec. 3. This can be regarded as a more challenging variant of the sync/off-sync task that prior work solves. We also experiment with a simpler setting with only 3 offset classes $[-1, 0, +1]$, that test if a model could predict if one track either lags, is in sync, or is ahead of the other one.

Metrics. Considering the human off-sync perception tolerance, in our experiments, we mainly report the Top-1 Accuracy with a ± 1 class of temporal tolerance (as described in Sec. 4.1). Note, that the training loss does not account for this tolerance. In the supplementary section, we additionally provide performance on accuracy without tolerance.

5.1 Results

Dense in time and space. Tab. 2 shows the comparison between the baseline and proposed architecture. As the task becomes more difficult (more sparse data or finer offset grid), we observe a larger gap between our model and the baseline. In particular, the baseline performs strongly on LRS3 (dense in time and space) in the setting with just three classes (98.4%). However, once the task gets more challenging, *i.e.* when training on finer offset shifts (21 class, 0.2 sec apart), the baseline performance deteriorates significantly ($\sim 89.8\%$). In contrast, the proposed model performs strongly even in the setting with finer offsets. This suggests that the proposed model is better suited for more challenging data tasks.

	Dense-Dense		Dense-Sparse		Sparse-Sparse	
	LRS3 (Face crop)		LRS3 (W/o face crop)		VGGSound-Sparse	
	3 cls	21 cls	3 cls	21 cls	3 cls*	21 cls
AVST _{dec}	98.4	89.8	95.8	83.1	52.2	29.3
Ours	96.4	95.6	95.5	96.9	60.3	44.3

Table 2: **The proposed model handles the increasing complexity of the setting and dataset better than the baseline while reaching a strong performance compared to the oracle.** ‘Dense-Dense’ refers to the face-cropped speech videos (LRS3), ‘Dense-Sparse’ for spatially-sparse LRS3 (‘No face crop’), ‘Sparse-Sparse’ is reported on VGGSound-Sparse which is sparse in time and space, *e.g.* lion roars once during a clip. The synchronisation performance is measured in two settings: the 3-class with $(-1, 0, +1)$ offsets given 5-sec clips, and 21 classes of offsets from -2.0 to $+2.0$ sec with 0.2-sec step size. The latter setting allows ± 1 temporal class tolerance (± 0.2 sec). *: oracle performance is 70 %.

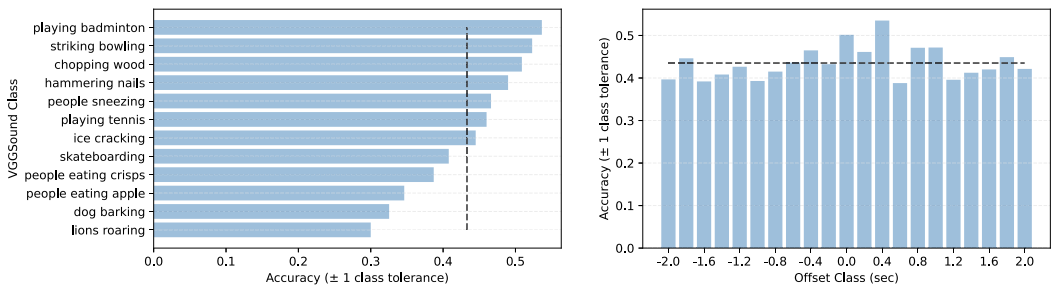


Figure 3: **Performance per data and offset class on VGGSound-Sparse (test).**

Dense in time and sparse in space. A similar effect is observed on the dataset that is dense in time and sparse in space, *i.e.* LRS3 (No face crop). As both architectures drop their performance slightly on the 3-class setting after switching to a more difficult dataset, the drop is more significant for the baseline than for our model. Moreover, the baseline performance drops substantially in the 21-class setting ($>6\%$), while our model performs strong.

Sparse in time and space. Finally, the experiments on the VGGSound-Sparse reveal an even larger difference between the baseline method and our final model. For this experiment, we add additional data augmentation to mitigate overfitting. In particular, our model significantly outperforms the baseline showing the benefit of selectors on a more challenging dataset and setting. Ultimately, our model reaches 60% in the 3-class setting which is close to the oracle performance ($\sim 70\%$: a human performance on 240 randomly picked videos), while achieving 44% in the 21-class setting. We report performance per class in Fig. 3.

5.2 Ablation Study

In Tab. 3 we provide results for an ablation study. The results are reported on the VGGSound-Sparse dataset with 21 offset classes. More results are provided in the supplementary.

Feature selectors. The architecture with feature selectors outperforms the vanilla transformer showing the effectiveness of selectors in ‘compressing’ signals from the audio and visual features. Also, Fig. 4 shows a memory footprint comparison of the two, omitting the memory consumed by feature extractors that are the same. It is evident that memory demand grows rapidly with the input duration making it impossible to work with longer sequences and the transformer that inputs concatenated audio and visual streams.

Selectors	Sync Model pre-trained on LRS3	Pre-trained Feature Extractors	Unfrozen Feature Extractors	Accuracy ₂₁
✗	✓	✓	✓	40.1
✓	✗	✓	✓	12.1
✓	✓	✗	✓	29.6
✓	✓	✓	✗	33.5
✓	✓	✓	✓	44.3

Table 3: **Results of the ablation study.** The experiments are conducted on VGGSound-Sparse with 21 off-set classes. Metric is Accuracy with ± 1 class tolerance.

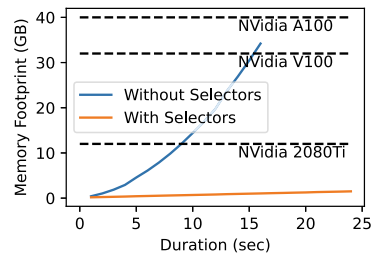


Figure 4: Working with longer sequences quickly becomes infeasible without selectors.

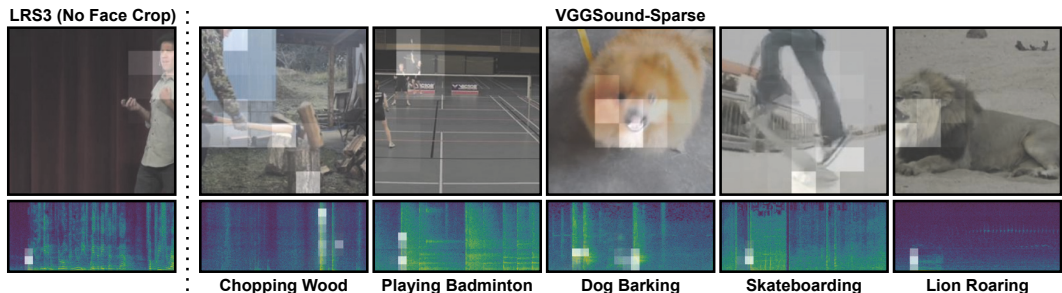


Figure 5: **Visual feature selectors focus on specific parts of the sparse signal that is useful for synchronisation.** Examples are from the hold-out set of LRS3 (‘No face crop’) and VGGSound-Sparse. Attention is captured from a selector to a visual or spectrogram feature token from a head within one of the layers. Attention values are min-max scaled.

Pre-training on dense signals. Pre-training a model on LRS3 (‘No face crop’) is an essential part of the training procedure on VGGSound-Sparse: 44.3 vs. 12.1 (near chance performance). For this reason, we also pre-train the baseline architecture (see Tab. 2).

Pre-trained feature extractors. The initialisation of audio and visual feature extractors with pre-trained weights has a strong positive effect on model performance. To initialise our feature extractors, we use weights of S3D pre-trained on Kinetics 400 for action recognition and ResNet18 pre-trained sound classification on VGGSound. The initialisation not only improves the final performance (43.3 vs. 33.5 %) but also significantly speeds up training. We attribute this improvement to the fact that such initialisation allows the model to ‘skip’ learning of the generic low-layer features and focus on training for synchronisation.

Frozen feature extractors during training. Allowing the gradients to reach raw data pixels is useful for audio-visual synchronisation as it makes the model sensitive to the smallest variations in the signal which is useful for synchronisation. In particular, having feature extractors to be trainable significantly boosts the performance from 34 to above 43 %, and the difference is even more pronounced on LRS3 (‘No face crop’) – see supplementary material.

Attention visualisation. Fig. 5 shows examples from LRS3-H.264 (‘No Face Crop’) and VGGSound-Sparse. Specifically, the attention exhibits spatial locality as the selectors learned to attend to the features extracted from the mouth region as expected from a model trained on a speech dataset. For a more challenging and diverse dataset, VGGSound-Sparse, the model highlights important parts of the visual and audio streams. In particular, the model accounts for the hit of the second badminton player who is far away in the background or attends to

Length (sec.)	VGGSound-Sparse	
	3 classes	21 classes
2	55.6	—
3	59.4	36.8
4	60.8	43.0
5	60.3	44.3
6	61.2	45.6
7	62.9	46.5

Table 4: Synchronisation accuracy improves with input length. We report results on two settings: with 3 offset classes ($-1, 0, +1$ sec) and 21 classes (± 2.0 sec grid with 0.2-sec step size). The results are reported on the test subset and accuracy is used as the metric. The accuracy for the 21-class setting is reported with ± 1 class tolerance. We use the same input lengths for pre-training, fine-tuning, and testing.

the axe during the chop, or the roaring mouth of the lion, yet these occur just once per video clip. Similarly, audio feature selectors point to specific parts of the spectrogram when the change occurs. More examples are provided in the supplementary material.

Input length. As the sparse synchronisation cues occur only occasionally within a video clip, processing shorter temporal crops decrease the chance of having sufficient cues for synchronisation, which, in turn, should decrease the performance. In Tab. 4, we show how performance varies with respect to the duration of input video clips. The results on the 3 and 21 offset classes illustrate the upward trend in model performance as the input duration extends. Note that the longer the inputs, the less unseen training data the model processes at each epoch. Specifically, a 10-second clip may be split into non-overlapping 3-second clips, which is not possible with clips longer than 5 seconds. Thus, this effect may undermine the current performance.

6 Conclusion

In this work, we study ‘in the wild’ videos that often have a synchronisation signal that is sparse in time. This requires a model to efficiently process longer input sequences as these synchronisation cues occur only rarely. To this end, we designed a transformer-based synchronisation model that has linear complexity with respect to the input length. This was made possible by using a small set of learnable ‘selectors’ that summarise long audio and visual features that are employed to solve the synchronisation task. To evaluate models in this challenging setup, we curate a dataset with only sparse events and train it on 5-second long clips. Finally, we discovered that compression artefacts caused by audio and video codecs might pose a threat to training for synchronisation, yet, as we show, these artefacts are easily identifiable and could be avoided to a certain extent.

Limitations. First, considering the complicated input-output relationship in the proposed model, it is challenging to determine which part of the input signal influences the output. Second, in this work, we considered signals that are ‘dense in time and space’, ‘dense in time but sparse in space’, and ‘sparse in time and space’. However, there is another interesting setting ‘sparse in time but dense in space’ yet it is not clear how to design such a dataset without making it ‘too artificial’. Third, despite showing strong performance on the proposed dataset, VGGSound-Sparse, there is still room for improvement.

Acknowledgements. Funding for this research was provided by the Academy of Finland projects 327910 and 324346, EPSRC Programme Grant VisualAI EP/T028572/1, and a Royal Society Research Professorship. We also acknowledge CSC – IT Center for Science, Finland, for computational resources.

References

- [1] Triantafyllos Afouras, Joon Son Chung, and Andrew Zisserman. LRS3-TED: a large-scale dataset for visual speech recognition. *arXiv preprint arXiv:1809.00496*, 2018.
- [2] Triantafyllos Afouras, Andrew Owens, Joon Son Chung, and Andrew Zisserman. Self-supervised learning of audio-visual objects from video. In *Proceedings of the European Conference on Computer Vision*, 2020.
- [3] Relja Arandjelovic and Andrew Zisserman. Objects that sound. In *Proceedings of the European Conference on Computer Vision*, pages 435–451, 2018.
- [4] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *Proceedings of the European Conference on Computer Vision*, 2020.
- [5] Anna Llagostera Casanovas and Andrea Cavallaro. Audio-visual events for multi-camera synchronization. *Multimedia Tools and Applications*, 2015.
- [6] Honglie Chen, Weidi Xie, Andrea Vedaldi, and Andrew Zisserman. VGG-Sound: A large-scale audio-visual dataset. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, 2020.
- [7] Honglie Chen, Weidi Xie, Triantafyllos Afouras, Arsha Nagrani, Andrea Vedaldi, and Andrew Zisserman. Audio-visual synchronisation in the wild. In *Proceedings of the British Machine Vision Conference*, 2021.
- [8] Joon Son Chung and Andrew Zisserman. Out of time: automated lip sync in the wild. In *Workshop on Multi-view Lip-reading, ACCV*, pages 251–263, 2016.
- [9] Joon Son Chung and Andrew Zisserman. Lip reading in the wild. In *Proceedings of the Asian Conference on Computer Vision*, pages 87–103, 2016.
- [10] Soo-Whan Chung, Joon Son Chung, and Hong-Goo Kang. Perfect match: Improved cross-modal embeddings for audio-visual synchronisation. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, 2019.
- [11] Vansh Dassani, Jon Bird, and Dave Cliff. Automated composition of picture-synched music soundtracks for movies. In *Proceedings of the European Conference on Computer Vision*, 2019.
- [12] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *Proceedings of the International Conference on Computer Vision*, pages 1422–1430, 2015.
- [13] Joshua P. Ebeneze, Yongjun Wu, Hai Wei, Sriram Sethuraman, and Zongyi Liu. Detection of audio-video synchronization errors via event detection. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, 2021.
- [14] Tavi Halperin, Ariel Ephrat, and Shmuel Peleg. Dynamic temporal alignment of speech to lips. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, 2019.

- [15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.
- [16] John Hershey and Javier Movellan. Audio vision: Using audio-visual synchrony to locate sounds. *Advances in Neural Information Processing Systems*, 1999.
- [17] Andrew Jaegle, Felix Gimeno, Andrew Brock, Andrew Zisserman, Oriol Vinyals, and Joao Carreira. Perceiver: General perception with iterative attention. *arXiv preprint arXiv:2103.03206*, 2021.
- [18] Venkatesh S Kadandale, Juan F Montesinos, and Gloria Haro. VocaLiST: An audio-visual synchronisation model for lips and voices. *arXiv preprint arXiv:2204.02090*, 2022.
- [19] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.
- [20] Naji Khosravan, Shervin Ardeshir, and Rohit Puri. On attention modules for audio-visual synchronization. In *Workshop on Sight and Sound, CVPR*, 2019.
- [21] You Jin Kim, Hee Soo Heo, Soo-Whan Chung, and Bong-Jin Lee. End-to-end lip synchronisation based on pattern classification. In *SLT Workshop*, 2021.
- [22] Bruno Korbar, Du Tran, and Lorenzo Torresani. Co-training of audio and video representations from self-supervised temporal synchronization. In *Advances in Neural Information Processing Systems*, 2018.
- [23] Andrew Owens and Alexei A. Efros. Audio-visual scene analysis with self-supervised multisensory features. In *Proceedings of the European Conference on Computer Vision*, 2018.
- [24] Lawrence Rabiner and Biing-Hwang Juang. *Fundamentals of speech recognition*. Prentice-Hall, Inc., 1993.
- [25] ITU Radiocommunication. Relative timing of sound and vision for broadcasting.
- [26] Prarthana Shrestha, Mauro Barbieri, Hans Weda, and Dragan Sekulovski. Synchronization of multiple camera videos using audio-visual features. *IEEE Transactions on Multimedia*, 2010. doi: 10.1109/TMM.2009.2036285.
- [27] Malcolm Slaney and Michele Covell. Facesync: A linear operator for measuring synchronization of video facial images and audio tracks. *Advances in Neural Information Processing Systems*, 2000.
- [28] Nicolas Staelens, Jonas De Meulenaere, Lizzy Bleumers, Glenn Van Wallendael, Jan De Cock, Koen Geeraert, Nick Vercammen, Wendy Van den Broeck, Brecht Vermeulen, Rik Van de Walle, et al. Assessing the importance of audio/video synchronization for simultaneous translation of video sequences. *Multimedia systems*, 2012.

-
- [29] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, 2017.
- [30] Donglai Wei, Joseph J Lim, Andrew Zisserman, and William T Freeman. Learning and using the arrow of time. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8052–8060, 2018.
- [31] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *Proceedings of the European Conference on Computer Vision*, pages 305–321, 2018.
- [32] Mingze Xu, Yuanjun Xiong, Hao Chen, Xinyu Li, Wei Xia, Zhuowen Tu, and Stefano Soatto. Long short-term transformer for online action detection. In *Advances in Neural Information Processing Systems*, 2021.

