TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

MATTI MOLKKARI

AUTOMATIC CLASSIFICATION OF PERTURBATION-INDUCED QUANTUM SCARS

Bachelor of Science Thesis

Examiners: Janne Solanpää, Joonas Keski-Rahkonen and Prof. Esa Räsänen

Submitted for review on 8 June 2018

ABSTRACT

MATTI MOLKKARI: Automatic Classification of Perturbation-Induced Quantum Scars
Tampere University of Technology
Bachelor of Science Thesis, 26 pages, 21 Appendix pages
June 2018
Degree Programme in Science and Engineering
Major: Advanced Engineering Physics
Examiners: Janne Solanpää, Joonas Keski-Rahkonen and Prof. Esa Räsänen

Keywords: Quantum scar, unsupervised learning, clustering, convolutional network, self-organizing map

A recently discovered perturbation-induced quantum scarring phenomenon is an example of quantum mechanical suppression of chaos. The phenomenon is seen on some eigenstates of the system as *quantum scars*, which are remnants of the periodic orbits of the corresponding unperturbed classical system.

The phenomenon allows efficient propagation of wave packets through two-dimensional potential wells. The scarring is controllable by an external magnetic field and by manipulating the properties of the perturbations, indicating exciting possibilities in quantum transport. Exploiting this phenomenon requires knowledge about the scarred eigenstates, found among thousands of eigenstates in the systems of interest. Therefore an automated procedure for classifying the different types of scars and quantifying their abundance and strength in each system is desired.

A solution generalizable to any system is sought by approaching the problem from an unsupervised learning viewpoint, to avoid the laborious task of labeling training data in the studied systems. An integral tool utilized in this thesis is the clustering of feature vectors extracted from the two-dimensional probability density grids of the eigenstates. The feature vectors are obtained by considering local probability density histograms and by employing convolutional networks. Self-organizing maps are also utilized for forming a representation of typical eigenstates in the system.

The methods are found to perform satisfactorily if the scarring is strong enough. Weaker scarring poses challenges, particularly if the scars of the same kind are present in multiple orientations and scales. In this case the number of clusters required for adequate portrayal of the system grows impractically large. The inverse participation ratio provides a useful measure for intra-cluster scarring strength, but it is not globally applicable to the whole system.

TIIVISTELMÄ

MATTI MOLKKARI: Kvanttiarpien automaattinen luokittelu
Tampereen teknillinen yliopisto
Kandidaatintyö, 26 sivua, 21 liitesivua
Kesäkuu 2018
Teknis-luonnontieteellinen koulutusohjelma
Pääaine: Teknillinen fysiikka
Tarkastajat: Janne Solanpää, Joonas Keski-Rahkonen ja prof. Esa Räsänen

Avainsanat: Kvanttiarpi, ohjaamaton oppiminen, klusterointi, konvoluutioverkko, itseorganisoituva kartta

Hiljattain havaittu häiriöiden aiheuttama kvanttiarpeutumisilmiö on esimerkki, kuinka kvanttimekaniikka voi tukahduttaa klassisen kaaoksen. Ilmiössä klassisen mekaniikan mukaiset periodiset liikeradat häiriöttömässä systeemissä jättävät jäljen, *kvanttiarven*, vastaavan kvanttimekaanisen systeemin joihinkin ominaistiloihin.

Ilmiö mahdollistaa aaltopakettien tehokkaan etenemisen kaksiulotteisissa potentiaalikaivoissa. Arpeutumista voidaan hallita ulkoisen magneettikentän avulla ja häiriöiden ominaisuuksia muokkaamalla, mikä antaa odottaa ilmiöltä mielenkiintoisia mahdollisuuksia kvanttikuljetuksen sovelluksissa. Ilmiön hyödyntäminen vaatii tietoa arpeutuneista ominaistiloista, jotka täytyy löytää mielenkiintoisten systeemien tuhansien ominaistilojen joukosta. Tämän takia olisi hyödyllistä kehittää automaattinen menetelmä erityyppisten arpien luokitteluun sekä niiden esiintymistiheyden ja voimakkuuden määrittämiseksi eri systeemeissä.

Yleispätevän ratkaisun löytämiseksi ongelmaa lähestytään ohjaamattoman oppimisen näkökulmasta, ettei jokaisessa tutkittavassa systeemissä tarvitsisi ensin käsin tehdä aikaavievää opetusaineiston luokittelua. Työssä käytettävä keskeinen työkalu on esitysvektoreiden klusterointi. Esitysvektorit rakennetaan tarkasteltavista ominaistiloista hyödyntäen tilojen todennäköisyystiheyksiin pohjautuvia histogrammeja ja konvoluutioverkkoja. Lisäksi itseorganisoituvien karttojen avulla luodaan esitys systeemissä esiintyvistä tyypillisistä ominaistiloista.

Menetelmät tuottavat halutunlaisen tuloksen, jos arpeutuminen on riittävän voimakasta. Heikompi arpeutuminen on ongelmallista, etenkin jos tietyntyyppiset arvet esiintyvät useissa asennoissa erikokoisina. Tällöin systeemissä esiintyvien tyypillisten ominaistilojen riittävä klusterointi vaatii liian monta luokkaa. Arpeutumisen voimakkuutta voidaan kuvata käänteisellä osallistumissuhteella, joka ei kuitenkaan sovellu systeemin ominaistilojen luokitteluun. Kuitenkin tämä luku osoittautuu mielekkääksi arpeutumisasteen mittariksi eri luokkien sisällä.

CONTENTS

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

IPR	Inverse participation ratio
PI	Perturbation-induced
SOM	Self-organizing map
c	index of the best matching node in SOM
$D,\ D(c,j)$	topological distance between the node j and the best matching node c in SOM
$d,\ d(z,m_j)$	distance between sample of data and model vector of SOM node j
F_c	probability density image cumulative distribution high intensity cutoff value
$h_{cj}(t)$	neighborhood function in SOM
i	index of an eigenstate
j	index of a SOM node
$m_j,\ m_j(t)$	model vector at node j in SOM
$m_j(x,y)$	model image at node j in SOM
N_{bins}	number of bins in a histogram
N_c	number of clusters
n_c	number of clusters during each iteration of iterative clustering
r_c	location of the best matching node in SOM
r_j	location of node j in SOM
T	total learning time in SOM
$T^{(1)},\ T^{(2)}$	total learning time during the initial and final learning phases
t	time step index in SOM during learning
x	x-coordinate
y	y-coordinate
z	sample of data shown to SOM during learning
$\alpha(t)$	learning rate in SOM
α_0	initial learning rate in SOM
$\alpha_0^{(1)},\ \alpha_0^{(2)}$	initial learning rate during the initial and final learning phases
$\beta(t)$	neighborhood size in SOM
β_0	initial neighborhood size in SOM
$\beta_0^{(1)},\ \beta_0^{(2)}$	initial neighborhood size during the initial and final learning phases
γ	parameter for contrast enhancement
$\Theta_i,\ \Theta_i(x,y)$	preprocessed probability density image of eigenstate i
Θ_c	probability density image high intensity cutoff value
$\Theta_{\text{min}},\ \Theta_{\text{max}}$	minimum and maximum for probability density image normalization
σ	standard deviation defining the extent of Gaussian blur
$\psi_i,\ \psi_i(x,y)$	wave function of an eigenstate
$\lvert\psi_i\rvert^2,\ \lvert\psi_i(x,y)\rvert^2$	probability density associated with eigenstate i

1. INTRODUCTION

Classical chaotic systems are characterized by extreme sensitivity to initial conditions, with tiny perturbations eventually leading to completely different phase space trajectories. Quantum systems, on the other hand, are decidedly non-chaotic in the traditional sense, as any difference between two initial states remains constant in time. In quantum systems chaos is manifested in the statistical properties of the eigenenergies. [1]

Quantum fingerprints of chaos can also be seen in the eigenstates themselves in the form of quantum scars. Quantum scars are anomalous enhancements of probability density along periodic orbits of corresponding classical systems, originally discovered on short unstable periodic orbits of classically chaotic systems [2]. While periodic orbits exist in chaotic systems, their total phase space volume is zero. Thus the instability of the orbits is what differentiates scarring from any expected enhancements of the probability density due to the correspondence principle [2].

A new kind of strong quantum scarring was recently discovered in two-dimensional potential wells perturbed by local impurities where the scars form along the periodic orbits of the corresponding *unperturbed* classical system. The appearance of the scars can be explained with degenerate perturbation theory. [3] The theory predicts that the scarred eigenstates extremize the overlap with the impurities, allowing some control over the orientations of the scars for applications [3, 4].

The geometry of these perturbation-induced (PI) scars is controllable with an external magnetic field, and even a single perturbation is sufficient to facilitate scarring while exercising control over the orientations of the scars [4]. The particularly strong enhancements of the probability density in these PI scars results in especially efficient propagation of wave packets along the scars [3]. These two properties, the controllability and efficient wave packet propagation, indicate exciting possibilities for applications in quantum transport [3, 4].

Exploiting this phenomenon for applications requires knowledge about the scarred states in the studied systems. As each system contains thousands of eigenstates in the relevant energy range, manual inspection of the states does not lend itself to large scale investigations of different systems. Therefore an automated procedure for quantifying the abundance and strength of different types of scars in each system is highly desired, motivating the study in this thesis.

This thesis is structured as follows: Chapter 2 briefly explains the theory behind the phenomenon and then proceeds with the descriptions of the methods utilized in this thesis.

Chapter 3 explains how the methods are applied to the problem. The main results are presented in chapter 4. The thesis concludes with chapter 5 where the results are briefly discussed and suggestions are given for further research.

2. THEORY

At the atomic level classical mechanics is not applicable anymore. The behavior of particles, such as electrons, is governed by the laws of quantum mechanics. An important consequence of the quantum theory is the abolition of exact knowledge of particle positions and momenta. Instead, information about possible values for these observables is encoded in a complex-valued wave function ψ describing the quantum state. [5]

When considering stationary states with time-independent observables, the wave functions are the eigenfunction solutions of the time-independent Schrödinger equation

$$\hat{H}\psi = E\psi, \tag{2.1}$$

where $\hat{H}$ is the Hamiltonian operator and E is the eigenenergy associated with the eigenfunction ψ. In position basis the Hamiltonian operator for a particle in a two-dimensional potential well $V(x,y)$ is given by

$$\hat{H} = -\frac{\hbar^2}{2m}\left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}\right) + V(x,y), \tag{2.2}$$

where $\hbar$ is the reduced Planck's constant and m is the mass of the particle. Due to the linearity of the equation, an eigenfunction ψ multiplied by a constant remains a solution with the same eigenenergy. Physically meaningful eigenfunctions are square-integrable and thus normalizable according to the condition

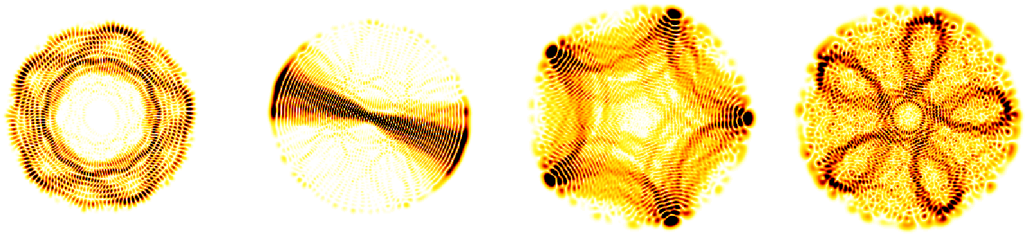
$$\iint_{\Omega} |\psi(x,y)|^2 \,\mathrm{d}x\,\mathrm{d}y = 1, \tag{2.3}$$

where the integration is performed over the whole domain Ω. The square modulus $|\psi(x,y)|^2$ of a properly normalized wave function can be interpreted as the probability density of finding the particle at the location (x,y). [5]

For states bound by the potential, boundary conditions result in a discrete set of solutions ψ_i with quantized energy levels E_i. Atomic orbitals are perhaps the most common example of such quantized states. Only the simplest potentials permit exact analytical solution of the equation, and in many applications numerical approximation methods must be utilized. [5] The eigenstates studied in this thesis are calculated with the open source `itp2d` software, which incorporates the most recent advances in the imaginary time propagation method [6].

2.1 Perturbation-induced quantum scars

In two-dimensional potential wells scarring may occur if the otherwise separable system is perturbed by local impurities. In this case some of the high-energy eigenstates form



(a) **(b)** **(c)** **(d)**

Figure 2.1. Examples of scarred eigenstates. *The probability densities of the states are depicted with the brightest percentile of the probability density cut off to aid the visualization of the scars with a linear colormap. A pentagram-shaped scar (a) with fivefold symmetry and a "bouncing ball" type scar (b) corresponding to zero angular momentum in a system with circularly symmetric $V(r) = \frac{1}{2}r^5$ unperturbed potential. Star-shaped (c) and flower-like (d) scars with fivefold symmetries in a system with circularly symmetric $V(r) = \frac{1}{2}r^2$ unperturbed potential in an external magnetic field.*

scars along the periodic orbits of the unperturbed system. [3, 4] Some examples of these perturbation-induced quantum scars are shown in Fig. 2.1.

The scarring phenomenon can be explained by degenerate perturbation theory. There are near-degenerate eigenstates in the unperturbed system, and the scars are a result of the small perturbations causing the eigenstates to mostly localize into subspaces spanned by the linear combinations of these near-degenerate eigenstates. Localized perturbations, such as small Gaussian bumps, generate scarred eigenstates from these subspaces as such states extremize the expectation value with the perturbations, explaining the orientations of the scars. [3]

The strength of the scarring is traditionally evaluated by propagating Gaussian wave packets initialized on the periodic orbits of the scars, and studying the periodic peaks of the wave packet autocorrelation function after every completed orbit [2]. The magnitude of these peaks, i.e. the recurrence strength, is particularly strong in PI scars, exceeding that of the unperturbed systems, especially on later recurrences [3]. The counter-intuitive result that perturbations enhance the propagation of wave packets is another encouraging factor for applications and further necessitates the demand for an automated procedure for detection and classification of the scars.

The wave packet method requires knowledge about the periodic orbits. While the periodic orbits are relatively easy to enumerate in spherically symmetric potentials, a general solution is not readily available [3]. Indeed, a generic method is desired for the classification that is applicable to systems with arbitrary potential wells, without the laborious task of first manually identifying different kinds of scars.

Alternative approach for quantifying the scarring is based on the premise that in the scarred eigenstates the probability density is localized into smaller regions. This can be measured

by the second moment of the probability density

$$\Psi_i^4 = \iint_{\Omega} |\psi_i(x,y)|^4 \, \mathrm{d}x \, \mathrm{d}y, \tag{2.4}$$

known as the inverse participation ratio (IPR) [7, ch 5.2], and denoted by Ψ_i^4 for the purposes of this thesis. The IPR is computationally light approach applicable without knowledge about the periodic orbits, but it only characterizes the localization of the wave function and not the effectiveness of wave packet propagation.

2.2 Clustering

The problem of inferring hidden categories in data without first showing examples of labeled data is called unsupervised learning [8, 9]. A central tool in unsupervised learning is clustering, in which objects are grouped to common classes according to their similarity. To mathematically assess the similarity, the properties of the objects are usually represented by feature vectors, whose components are real numbers describing the properties in a suitable manner. [9] Two clustering algorithms relevant for this thesis, k-means clustering and hierarchical clustering, are described below.

2.2.1 k-means

The k-means algorithm attempts to cluster data consisting of N samples $x_i \in \mathbb{R}^n$ by choosing k centroids $\mu_i \in \mathbb{R}^n$ that minimize the total squared distance ϕ between the samples and the closest centroids. The problem is NP-hard but heuristic iterative algorithms exist and the most common variant is described in algorithm 1. While the algorithm is simple and generally fast, the resulting clustering can be arbitrarily far away from the true optimum. [10]

Choosing the initial centroids appropriately can guarantee strict bounds for the expectation value of the total squared distance ϕ relative to the true optimum. The k-means++ algorithm proposes that the first centroid μ_1 should be chosen uniformly at random from all samples x_i. The remaining centroids should be chosen randomly from the remaining samples in such a way that the probability that the sample x_i is chosen is proportional to its squared distance from the closest centroid already chosen. [10]

Algorithm 1 The k-means algorithm.

1. Initialize by choosing k centroids μ_i.
2. Each data point x_i belongs to the closest centroid.
3. Move the centroids to the means of the data points belonging to each centroid.
4. Repeat from step 2 until convergence.

2.2.2 Hierarchical clustering

In hierarchical clustering samples of data are hierarchically grouped together resulting in a tree-like representation of similar samples, called a dendrogram. An agglomerative approach is considered, where each sample starts in its own cluster and then the clusters are combined according to their similarity. [9, p. 393]

A suitable distance metric is required to calculate the distances between pairs of samples. Any pairwise distance may be utilized, as long as the choice can be justified for the application in question. Additionally, a linkage criterion determines how the distances are calculated between clusters containing more than a single element. [9, ch. 10.3.2] Three options are discussed here: average and complete linkage, and Ward's minimum variance method.

In average linkage the inter-cluster distance between two clusters is defined as the arithmetic mean of all the pairwise distances between the elements of the different clusters. In complete linkage the maximum of the pairwise distances determines the inter-cluster distance instead. [9, ch. 10.3.2] Ward's method links the two clusters which results in the smallest increase in variance within the cluster [11].

Finally, a criterion is required for extracting the desired clusters from the dendrogram. The simplest solution is to merely stop the agglomerative grouping process after reaching a desired number of remaining clusters. [9, ch. 10.3.2]

2.3 Convolutional networks

Convolution of a function $f(t)$ with a convolution kernel $k(t)$ is denoted by $(f * k)(t)$ and is defined as

$$(f * k)(t) = \int_{-\infty}^{\infty} f(\tau)k(t-\tau)\,\mathrm{d}\tau. \tag{2.5}$$

With discrete data, the integration is replaced by summation [8, p. 327–328]. The eigenstates are two-dimensional and convolution networks utilize multiple convolutions with separate kernels to extract different features. These two-dimensional convolutions, and how the kernels highlight features resembling them, is illustrated in Fig. 2.2.

With multiple kernels a third dimension, called channel, is needed to keep track of the results of each convolution. Let $f_{i,j,k}$ be the input to the convolution and $g_{i,j,k}$ the output. The indices i and j are the regular two-dimensional indices and k is the index of the channel. This leads to 4-dimensional convolution kernels $k_{i,j,k,l}$, where i and j are again the normal indices to the original two-dimensional data. The indices k and l describe the connection strength between the output channel k and input channel l. The convolution can then be calculated by

$$g_{i,j,k} = \sum_{m,n,l} f_{i+m,j+n,l}k_{m,n,k,l}, \tag{2.6}$$

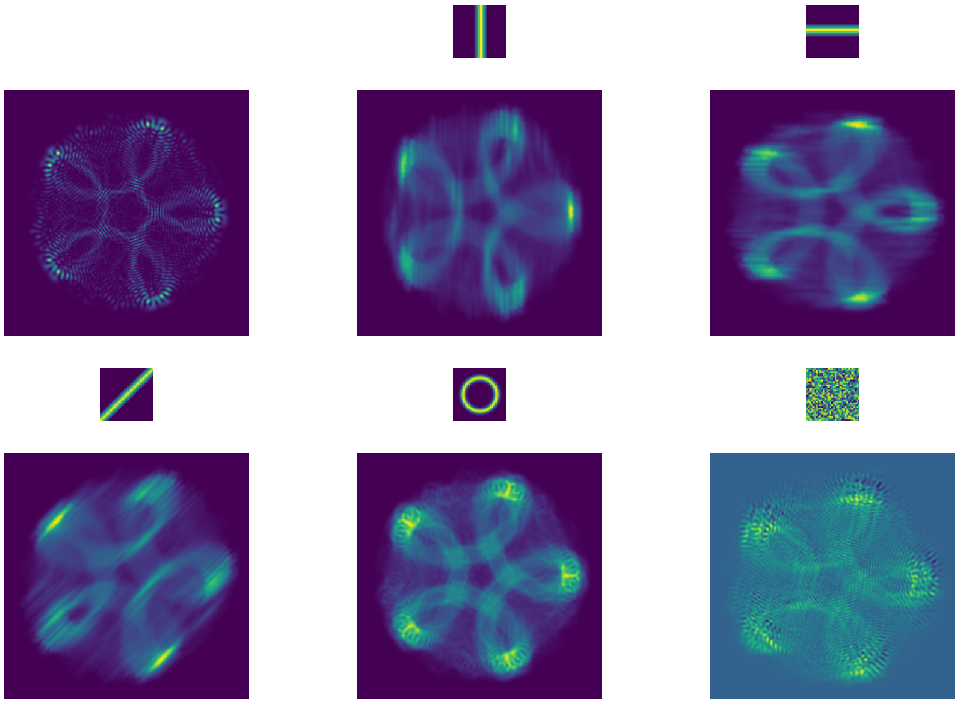


Figure 2.2. Convolution examples. *The original image is shown on top left. The smaller images depict the convolution kernels and the results of the convolutions are illustrated below the kernels.*

where the index l is summed over all the channels in the input, with m and n summed over the regular two-dimensional part of the discrete convolution kernel. [8, p. 342]

When the data is discretized on a finite grid there are complications arising from boundary effects. Defined as above, the result of the convolution would shrink by one element less than the size of the kernel at each dimension. This can be compensated by zero-padding the input function $f_{i,j,k}$ around the edges in such a manner that the size of the output remains the same as the input. [8, p. 343–345]

Convolution networks usually consists of many layers with the resulting convolutions fed as inputs for the subsequent layers. The output of the previous layer is commonly processed with a pooling layer before proceeding to the next convolution layer. The purpose of the pooling layer is to produce summary statistics about its input to reduce the size of the output for computational and statistical efficiency. Pooling also makes the output approximately invariant to small translations in the input data. The most common type is a max pooling layer, which replaces rectangular regions in the input data with the maximum value in that region. [8, ch. 9.3]

The coefficients of the convolution kernels are usually teachable parameters in supervised learning when the classes are known during teaching. However, the unsupervised approach pursued in this thesis discards that possibility. A simple solution is to populate the kernels randomly, which function reasonably well, as the convolution layers followed by pooling

become frequency selective and translation invariant. Other options are to design the kernels by hand or devise an unsupervised scheme to learn them from the data. [8, p. 357]

2.4 Self-organizing maps

The self-organizing map (SOM) is a tool for visualization of high-dimensional data by producing a similarity graph of the data in lower dimensions. The dimensionality reduction is accomplished by converting nonlinear statistical relationships in the data into geometric relationships in the graph; i.e. similar data will reside in neighboring nodes in the graph. [12, p. 106]

Each node j has a model vector $m_j \in \mathbb{R}^n$ associated with it. The model vectors represent typical data at the nodes' locations in the graph. The SOM maps samples of input data $z \in \mathbb{R}^n$ into nodes with the most similar model vectors. Formally the best matching node c can be defined as

$$c = \arg\min_j \left\{ d(z, m_j) \right\}, \tag{2.7}$$

where d is a suitable distance metric for the problem in consideration. [12, p. 106]

The objective of the SOM algorithm is learning model vectors that result in an ordered and descriptive mapping of the distribution of the input data [12, p. 106]. The model vectors are traditionally initialized randomly, but similarly to k-means, the results depend on the initialization procedure. Another alternative for initialization is principal component analysis, which performs better for quasilinear data, whereas random initialization is recommended for nonlinear data. [13]

After initializing the model vectors the learning process is accomplished by showing samples of input data z and updating the model vectors m_j according to the formula

$$m_j(t+1) = m_j(t) + \alpha(t) h_{cj}(t) \left[z - m_j(t) \right], \tag{2.8}$$

where t is a discrete time coordinate, α is a learning rate factor, and h_{cj} is a neighborhood function that defines to what extent the nodes topologically closest to the best matching node are translated towards the input data. [12, p. 109–111]

The neighborhood function is a crucial component of the algorithm that is responsible for the ordering in the map. A simple choice is a discrete neighborhood function that assigns topologically nearby nodes to the neighborhood

$$h_{cj}(t) = \begin{cases} 1, & D(c, j) \leq \beta(t) \\ 0, & \text{otherwise} \end{cases}, \tag{2.9}$$

where $D(c, j)$ is the topological distance between the node j and the best matching node c. The neighborhood size is controlled by the monotonically decreasing parameter $\beta(t)$.

Another usual choice for the neighborhood is a Gaussian function

$$h_{cj}(t) = e^{-\frac{1}{2}\left(\frac{D(c,j)}{\beta(t)}\right)^2}, \tag{2.10}$$

which provides a smoother continuous neighborhood. [12, p. 111]

The initial neighborhood size should be large enough to ensure that the map will be ordered globally instead of being composed of locally ordered patches. On the other hand, convergence requires that the changes to the map decrease over time. This is enforced by having monotonically decreasing learning rate factor $\alpha(t)$ and neighborhood size $\beta(t)$. Some reasonable choices are linear or exponential functions. [12, p. 111–112] The following linear functions are adopted for this thesis

$$\alpha(t) = \alpha_0 \left(1 - \frac{t}{T}\right) \tag{2.11}$$

$$\beta(t) = \beta_0 \left(1 - \frac{t}{T}\right), \tag{2.12}$$

where α_0 and β_0 are the initial values for the learning coefficient and neighborhood size, respectively, and T is the total learning time.

The learning process should be performed in two phases. The goal of the first phase is to attain rough global ordering by utilizing large initial values for the learning rate factor and neighborhood size. A good rule of thumb is that this global ordering phase should last for the order of $T^{(1)} \approx 1000$ steps, with initial neighborhood radius $\beta_0^{(1)}$ being roughly half the size of the network, and initial learning rate $\alpha_0^{(1)}$ close to unity. [12, p. 112]

Convergence is achieved during the final phase, which should continue for an extended number of steps $T^{(2)}$, at least 500 times the amount of nodes. To ensure convergence, the learning should proceed slower than in the previous phase. The learning rate $\alpha_0^{(2)}$ should not exceed values of roughly 0.02 and the neighborhood radius $\beta_0^{(2)}$ should only contain the nearest neighbors. [12, p. 112]

Common choices for the topology of the map include rectangular and hexagonal grids in two dimensions and linear topologies in one dimension. The edges of the map may also be connected, resulting in toroidal topologies in two dimensions and closed loops in one dimension.

3. METHODS

Two principal methods are employed for the classification: methods based on clustering and methods based on self-organizing maps. Clustering-based methods can further be categorized according to the methods used for extracting feature vectors from the eigenstates. Initial experiments were performed with feature vectors based on probability density histograms. Later experiments focused on feature vectors extracted with convolutional networks.

Two different systems are considered in this study, with circularly symmetric unperturbed potentials $V_2(r) = \frac{1}{2}r^2$ and $V_5(r) = \frac{1}{2}r^5$. The potentials are disturbed by Gaussian-like bumps defined by amplitudes $A_2 = 4$, $A_5 = 20$ and standard deviations $\sigma_2 = 0.1$, $\sigma_5 = 0.08$ in Hartree atomic units. The bumps are randomly placed so that on average there are 2 bumps per unit area. The r^2-system also has an external magnetic field with the strength $B_2 = 1.5$ in the SI-units-based convention of magnetism-related Hartree atomic units. These parameters were chosen because the scarring phenomenon is clearly visible in these systems [3, 4].

In both systems the first 5000 eigenstates are calculated, of which at least 4000 are required to converge. The ground state and other lower energy states are highly distinct compared to the higher energy states, which would complicate clustering. Additionally, the scarring phenomenon cannot be clearly detected in the lowest energy states due to the relatively small number of nodes in the eigenstates that are confined to a relatively small region at the center of the system. Therefore, the 2000 lowest energy states are exempted from the classification. The eigenstates ψ_i of the studied systems are calculated on a 300-by-300 grid with the open source `itp2d` code [6].

3.1 Preprocessing of probability density images

The classification is performed by analyzing the probability densities $|\psi_i(x,y)|^2$ calculated on a two-dimensional grid. The eigenstates $\psi_i(x,y)$ are normalized such that the integral of the probability density over the whole domain Ω is unity

$$\iint_\Omega |\psi_i(x,y)|^2 \,\mathrm{d}x\,\mathrm{d}y = 1. \tag{3.1}$$

Better classification results are facilitated by suitable preprocessing of the probability densities prior to classification. However, these manipulations ruin the proper normalization of the probability density. Therefore, these preprocessed probability density images are denoted by $\Theta_i(x,y)$. Different preprocessing techniques are described below and their effects are illustrated in Fig. 3.1.

3.1.1 High intensity cutoff

The very highest values of the probability density are usually concentrated on very small areas. Cutting off the highest peaks of the probability density increases the importance of the overall structure of the scars instead of focusing on the very highest peaks. This also aids the visual inspection of the scars.

The cutoff procedure is accomplished by choosing a threshold value $F_c \in [0,1]$ from the cumulative distribution function $F[\Theta_i(x,y)]$ of the probability density images $\Theta_i(x,y)$. Values of images for which $F[\Theta_i(x,y)] > F_c$ are replaced with the value Θ_c of the image at $F(\Theta_c) = F_c$, resulting in the following mapping

$$\tilde{\Theta}_i(x,y) = \begin{cases} \Theta_i(x,y), & F[\Theta_i(x,y)] < F_c \\ \Theta_c, & F[\Theta_i(x,y)] \geq F_c \end{cases}. \tag{3.2}$$

In practice this means that F_c is the highest percentile of allowed pixel intensities after which they are cut off. Values close to unity already significantly promote the visibility of the scars, as the strongest peaks of the probability density are highly localized, as shown in Fig. 3.1b.

The cumulative distribution function F, and consequently the value Θ_c, can be calculated either globally for all the states or separately for each state i. Global cumulative distribution is required if the relative strength of the scarring between different states is to be maintained. However, if only the visibility of the scarring within individual states is considered, then it is beneficial to calculate the distribution for each state individually.

3.1.2 Normalization

Instead of normalizing for the probabilistic interpretation, the images Θ_i are normalized to enable efficient application of machine learning techniques. Many machine learning algorithms are best suited for data in the unit interval [8, 9].

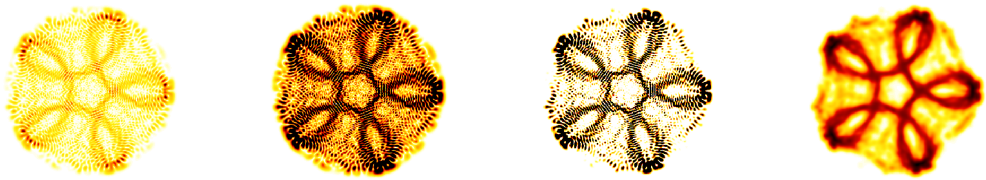
The normalization is performed by a simple linear mapping

$$\tilde{\Theta}_i(x,y) = \frac{\Theta_i(x,y) - \Theta_{\min}}{\Theta_{\max} - \Theta_{\min}}, \tag{3.3}$$

where $\Theta_{\min}$ and $\Theta_{\max}$ are the minimum and maximum values of the images. Similarly to the high intensity cutoff, the normalization may be performed globally for all states or for each state individually, employing the corresponding minima and maxima.

3.1.3 Contrast enhancement

The scars are characterized by high probability density. Thus the prominence of the scars can be increased by enhancing the contrast between higher and lower probability density



(a) **(b)** **(c)** **(d)**

Figure 3.1. Preprocessing example. *The Original 300-by-300 probability density grid (a) is subsequently preprocessed by high intensity cutoff with $F_c = 0.90$ (b), contrast enhancement with $\gamma = 4$ (c), and Gaussian blur with $\sigma = 3$ (d).*

areas. The relative distances between low and high values can be increased by raising the values to power higher than unity. For images normalized into the unit interval, the transformation

$$\tilde{\Theta}_i(x,y) = [\Theta_i(x,y)]^\gamma \tag{3.4}$$

with $\gamma > 1$ retains the desired normalization.

3.1.4 Gaussian blur

The probability density has an overall two-dimensional wavelike structure consisting of low probability density troughs within the high probability density patterns. Emphasizing the importance of the large scale structures instead of the small scale details in the ripple pattern can be accomplished by blurring the image. Gaussian blurring is considered an efficient method for smoothing images, and is efficiently achieved by convolving the target image with a two-dimensional Gaussian kernel function $g(x,y;\sigma)$

$$g(x,y;\sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}, \tag{3.5}$$

where the standard deviation σ controls the magnitude of the smoothing effect. [14, ch. 3.4.4]

3.2 Clustering of feature vectors

Calculating the eigenstates on an n-by-m grid results in an nm-dimensional representation for the states. With $n = m = 300$ the direct clustering of the resulting 90 000-dimensional feature vectors is impractical. Therefore a dimensionality reduction procedure capturing the essential features of the states is required. Two main schemes are considered for extracting feature vectors from the eigenstates, employing probability density histograms and convolutional networks.

Each dimension of the feature vectors describes a single feature represented as a numeric value. If the magnitudes of these values are not comparable to each other, the clustering will be dominated by features with the highest absolute variance. The feature vectors may be normalized to equalize the effect of variations within each dimension of the vectors. The normalization is performed by subtracting the mean and dividing by the standard deviation within each dimension, resulting in each feature having zero mean and unit variance.

3.2.1 Probability density histograms

Probability density histograms have the desirable property that they are invariant to rotations, which is beneficial as the scars may occur in different orientations. The range of probability density values is divided into N_{bins} bins of equal width and the number of probability density values falling within each bin are utilized as feature vectors. It is expected that the distribution of these bin counts may be very uneven, resulting in the clustering being dominated by the most populated bins. Therefore the feature vectors are normalized to zero mean and unit variance.

A single histogram calculated from the whole state may disregard too much information about the structure of the state. This can be overcome by subdividing the state into smaller grids and considering histograms in each subgrid. The histograms from all the subgrids, across all the eigenstates, are first clustered by k-means. The final feature vectors for clustering the eigenstates themselves are constructed from the clusters of the histograms within each state. To maintain the rotational invariance only the relative frequencies of the different clusters within each state are considered. These relative frequencies form the final representation for the states as feature vectors.

An alternative approach is to consider the two-dimensional image formed by the labels of the clustered histograms organized as a grid. This preserves more information about the structure of the state at the cost of losing rotational invariance. Another issue is the definition of a good pairwise distance metric for these images, as it is not meaningful to consider the regular Euclidean distance when the feature vector values are indices to different clusters. A simple solution incorporated here is to consider the total number of different cluster labels in the corresponding elements of the two feature vectors being compared. As a consequence of this distance metric it is not straightforward to apply the k-means algorithm, and instead hierarchical clustering is performed.

With both approaches the histograms can be calculated at multiple scales by subdividing the original states into smaller grids of various sizes. The final feature vector is then obtained by simply concatenating the feature vectors of each different scale.

3.2.2 Convolutional networks

Convolutional networks are utilized for extracting feature vectors from the probability density images. The images are normalized into unit interval prior to propagating them

through the network, as explained in section 3.1.2. The effect of other preprocessing techniques on the resulting feature vectors and subsequent clustering is also considered.

The network consists of five convolutional layers with uniformly random kernels. Each layer is followed by a max pooling layer with a 2-by-2 neighborhood, and the final output is flattened into one-dimensional vector. The structure of the network is described in Table 3.1. Each component of the resulting 648-dimensional feature vectors v_i is normalized to zero mean and unit variance before clustering. The normalized vectors are clustered with k-means and hierarchical clustering.

Table 3.1. *Structure of the convolutional network and the number of kernels and kernel sizes for convolution layers. The first two dimensions in the output shape are the size of the probability density grid and the final dimension is the number of channels.*

Layer	Kernels	Kernel size	Output shape
Input			$300 \times 300 \times 1$
Convolution 1	64	32×32	$300 \times 300 \times 64$
Max pooling 1			$150 \times 150 \times 64$
Convolution 2	32	16×16	$150 \times 150 \times 32$
Max pooling 2			$75 \times 75 \times 32$
Convolution 3	16	8×8	$75 \times 75 \times 16$
Max pooling 3			$37 \times 37 \times 16$
Convolution 4	8	4×4	$37 \times 37 \times 8$
Max pooling 4			$18 \times 18 \times 8$
Convolution 5	8	4×4	$18 \times 18 \times 8$
Max pooling 5			$9 \times 9 \times 8$
Flatten			648

3.2.3 Iterative clustering

An iterative approach to clustering is considered to assist in categorizing the clusters into scarred and non-scarred clusters. The underlaying assumption for this procedure is that scarred states are mutually more similar within a cluster than non-scarred states. Therefore during each iteration of clustering only the most similar cluster is kept and the clustering is repeated with the remaining data for the next iteration. The procedure is explained in more detail in Algorithm 2.

Test vectors associated with each state are considered for determining the most similar cluster. The feature vectors can be directly utilized for this purpose. Alternatively, the vectorized probability density grids, or their preprocessed versions, could be employed as test vectors.

Within each cluster the mean and the maximum values are determined for each component of the test vectors. Then the root-mean-square differences are calculated between the mean

and the maximum vectors of each cluster. The cluster with the smallest root-mean-square difference is considered to be the most similar one. Comparing only the calculated mean and maximum states eliminates the need for pairwise comparison of all the states. The mean-max comparison also innately penalizes clusters with outliers.

Algorithm 2 Iterative clustering.

1. Cluster the data $\mathscr{D}$ with any clustering algorithm into many clusters n_c.
2. Determine the cluster $\mathscr{C}_{\text{best}}$ containing the most mutually similar elements according to some suitable criterion.
3. Insert $\mathscr{C}_{\text{best}}$ into the set of final clusters $\mathscr{F}$ and remove its elements from the data $\mathscr{D}$.
4. Repeat from 1 with the remaining data $\mathscr{D}$ until a desired number of clusters N_c has been extracted.
5. Insert all the remaining elements in the data $\mathscr{D}$ as their own cluster into the final clusters $\mathscr{F}$.

3.3 Self-organizing maps

Self-organizing maps are utilized for clustering by having the nodes of the map learn typical probability density images found in the system. The nodes are arranged in a rectangular n-by-m grid of unit length in both directions. The topological distance $D(c,j)$ between the nodes is then simply the Euclidean distance of the nodes from each other in the grid:

$$D(c,j) = \left\| r_c - r_j \right\|. \tag{3.6}$$

The nodes directly represent probability density images with their model vectors. These vectors, or more precisely model images $m_j(x,y;t)$ in this case, are learned by showing downscaled probability density images from all the states at each time step t. The best matching node c is decided according to the distances $d(\Theta_i, m_j)$ given as the total squared differences between the shown image $\Theta_i(x,y)$ and the model image:

$$d(\Theta_i, m_j) = \sum_{x,y} \left[\Theta_i(x,y) - m_j(x,y;t) \right]^2. \tag{3.7}$$

The model images are updated according to Eq. (2.8) with m_j and Θ_i interpreted as matrices.

Preprocessing is necessary to provide good results, and suitable values for the preprocessing parameters F_c, γ and σ, are experimented with. Normalization and the computation of the high intensity cutoff value Θ_c are performed globally for all the states simultaneously, so that weaker scars are less likely to add noise to the nodes representing distinctly scarred states.

4. RESULTS

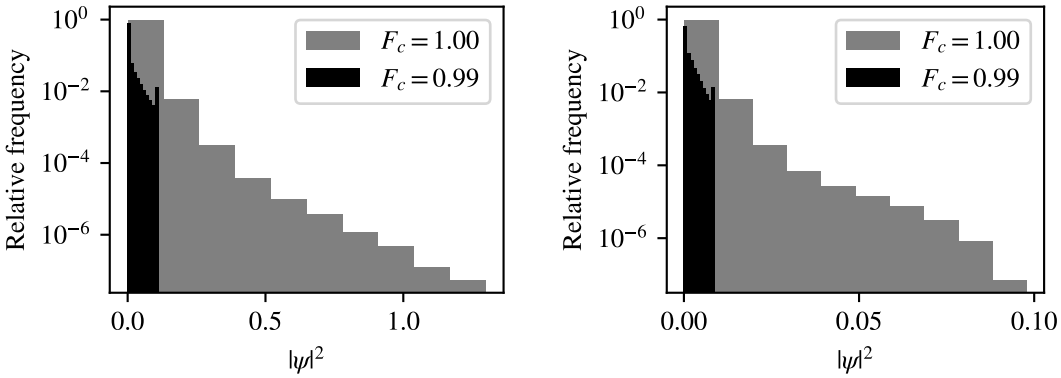
The IPR, or Ψ^4-value, was also considered as a means to classify the eigenstates. While it can differentiate particularly localized scars, such as "bouncing balls", from the other states, it is not useful measure for detecting weaker scars, e.g. pentagrams, amongst ordinary eigenstates. However, the IPR provides a meaningful measure for ordering the eigenstates within classes, as more prominent scars are generally more localized.

It may also be possible to characterize the scarred clusters by the distribution of the state indices within the clusters. As can be seen in the figures in the appendices, the scarred states are distributed somewhat evenly and the spacing appears to depend on the type of the scar, at least to some extent. This may not be generalizable phenomenon and could depend on the unperturbed potential being circularly symmetric homogeneous function. Additionally the scarred states also usually appear in pairs in adjacent states.

The scars appear in a few preferred orientations, determined by the affinity to extremizing the overlap with the perturbations. These different orientations complicate the clustering by requiring more clusters to capture all the orientations in addition to the different kinds of scars. Scaling is a subtler effect. The scale of the scars increases with energy, as larger area becomes available in the potential well. This poses some difficulties for the clustering, as the scars will also be divided into different clusters according to their energy, especially in the r^2-system, as the available area grows faster as a function of energy. Clustering the r^2-system is more complicated for other reasons as well, as the scars are generally weaker and the shapes are more varied. Increasing the number of clusters could produce higher quality clusters, but complicates classification.

Approximately 15 clusters seems to be a good compromise between capturing the essential scarred states and having too many clusters. On the other hand, the number of clusters could be increased if the methods are combined with secondary clustering after the initial clusters have been formed. The clusters could be represented by the mean values of the states belonging to each clusters. These mean representations could be clustered by taking the desired symmetry operations into account by exhaustive search and comparison with all the possible transformations applied in turn. Such brute force method may be more suitable to the much smaller and smoother set of the mean representations than to the full set of the calculated eigenstates.

The best clustering method depends on the utilized feature vectors. A problem with k-means is the stochastic initialization, which adds variance to the results and complicates reproducibility. Hierarchical clustering with Euclidean mean and Ward's linkage criterion is largely equivalent with deterministic results. Complete linkage is also useful, but average



(a) **(b)**

Figure 4.1. Probability density histograms. *The probability density histograms with 10 bins are calculated over all the states in both the r^5-system (a) and the r^2-system (b).*

linkage seems to be inferior in all the studied cases. The results from the different methods are presented below.

4.1 Probability density histograms

Probability density histograms of the studied systems are shown in Fig. 4.1, with full range of values ($F_c = 1.00$) and with the brightest percentile cut off ($F_c = 0.99$). Note that the y-axis is logarithmic and the highest values of the probability density are very infrequent; with them cut off the whole histogram fits inside the first bin of the full distribution. This long and thin tail is of decisive importance for successful clustering, and the quality of the clustering declines by removing the tail. On the other hand, normalization of the histogram bin counts to zero mean and unit variance appears to attribute predominant importance to the tail, as it comprises the majority of the bins.

As suspected, a single histogram from the whole state disregards too much information to be practical for clustering. Subdividing the probability density grid into smaller grids for higher resolution alleviates the problem to some extent. Utilizing the relative frequencies of these subgrid histogram clusters as the feature vectors preserves rotational invariance, at least partially. However, some mixing of scarred states still occurs and the method requires many clusters to extract all different kinds of scars into their own clusters. Especially the pentagram-like scars are easily mixed with other states.

An example of such clustering is presented in Fig. A.1. It can be seen that pentagrams are partly embedded in the cluster $\mathscr{C}_1$, which is illustrated in more detail in Fig. A.3. Some weaker bouncing balls are also lost to the larger clusters. The bouncing balls and circular scars appear to be clustered according to their widths. The system also contains some heptagram-like scars, but these are not assigned to any distinct cluster.

The most interesting results with histograms are achieved when the subgrid histogram

clusters are considered as two-dimensional image for each eigenstate. Surprisingly the best results are obtained when the subgrid histograms are clustered into only two clusters. This may be due to the distance metric introduced in section 3.2.1 not being very robust. The binary nature of the images also allows other more common metrics to be meaningfully utilized for the pairwise distances between the eigenstates.

For the r^5-system the results are shown in Fig. A.2 with hierarchical clustering utilizing L^1 metric and complete linkage. The most compelling observation is that the non-scarred states are mostly collected into the cluster $\mathscr{C}_1$, which is however contaminated by some bouncing ball scars. This cluster is visualized in more detail in Fig. A.4, which confirms that the widest bouncing balls are also placed into that cluster. Increasing the number of clusters would separate them, but an excessive amount of clusters is counter-productive.

The feature vectors were also clustered utilizing Euclidean metric and Ward's linkage criterion in hierarchical clustering. For comparison the results are illustrated in Fig. A.5. The non-scarred states are spread over several clusters and the variance within clusters appears to be slightly larger. However, the overall quality of the clustering is arguably higher, as there are hardly any misclustered states.

Corresponding clustering results for the r^2-system are show in Fig. A.6 and Fig. A.7. It is immediately clear that the clustering is not as successful in this system. Strong scars are rarer but weak scars are quite plentiful. Even though almost all the scars have fivefold symmetry, there is a lot more variation in the exact shapes of the scars due to the magnetic field. Scaling also becomes more crucial issue in the r^2-system, as the classically allowed area is growing significantly faster than in the r^5-system. Nevertheless, the Euclidean/Ward combination for the clustering metrics is the more successful approach for isolating the strongest scars.

Employing high intensity cutoff reduces the effectiveness of clustering with the histogram based methods, but the other preprocessing methods do not seem to have a significant effect. There also appears to be no benefit to the iterative approach of section 3.2.3. Extracting the feature vectors from multiple scales with subgrids of different sizes for concatenated multiscale feature vectors does not seem to appreciably affect the results.

4.2 Convolutional networks

The quality of the feature vectors extracted by the convolutional network depends on preprocessing. Moderate contrast enhancement appears to be useful. The amount may loosely depend on the system, but for the two studied systems a value of approximately $\gamma \approx 4$ yields good results. High intensity cutoff is the most important preprocessing step and a suitable threshold value F_c depends on the system. The r^5-system requires higher threshold value for good results, approximately $0.9 \lesssim F_c \lesssim 0.99$, and conversely the r^2-system benefits from lower threshold values, approximately $0.8 \lesssim F_c \lesssim 0.9$. The max pooling layers reduce the necessity of applying Gaussian blur. However, it would

appear that light blurring may encourage the clustering of the weakest scars into the noisier non-scarred clusters. Therefore it may be beneficial to apply blur with the width of approximately $\sigma \approx 3$ to obtain more distinctive scarred clusters. Evaluating the effect of the preprocessing parameters is partly overshadowed by the stochasticity introduced by the random convolution kernels, which generates some variance to the results.

The feature vectors should also be normalized to zero mean and unit variance before clustering. Hierarchical clustering with Euclidean metric and Ward's linkage criterion consistently outperforms the other considered clustering algorithms. The utilized metric has smaller effect on the quality of the results than the linkage criterion. The average and complete linkages are distinctly inferior choices for the present feature vectors, and the Ward's linkage criterion mandates the usage of the Euclidean metric.

An example of clustering the r^5-system is shown in Fig. B.1. Overall the clustering seems to slightly outperform the clusterings obtained from histograms. The fading of the scars is noticeable as the eigenstates become less localized with decreasing Ψ^4-value. It is questionable whether the least localized states should be clustered into non-scarred clusters instead. The weak heptagram-shaped scars are also not discernible in this clustering.

The same feature vectors are iteratively clustered in Fig. B.2. The benefit of the iterative approach is that most non-scarred states are in a single cluster. The fading of pentagram-scars is also illustrated in more detail in Fig. B.3 and Fig. B.4. Unfortunately there does not appear to be any clear-cut threshold Ψ^4-value to determine boundaries for scarring.

In Fig. B.5 the iterated clustering is performed by utilizing the probability density grids as test vectors, instead of the feature vectors themselves. Some circular eigenstates are erroneously assigned to the cluster $\mathscr{C}_0$, which is further illustrated in Fig. B.6. These thin circular states with strong probability density rims are very sensitive to the scaling of the eigenstates as a function of energy when employing direct comparison of the states for similarity, which is probably the reason for the misclustering. The weak heptagram-shaped scars are barely visible in the cluster $\mathscr{C}_{11}$, which is shown in more detail in Fig. B.7.

For the r^2-system a clustering example is presented in Fig. B.8. However, the clustering does not function adequately due to the reasons discussed in the previous section. The most prominent kinds of scars are apparent but in general the intra-cluster variance is higher than desirable. The iterative clustering approach mitigates the issue to some extent, and the results are shown in Fig. B.9 and Fig. B.10. Nonetheless, some clearly scarred states are left in the remainder cluster, which is inspected in greater detail in Fig. B.11 and Fig. B.12.

4.3 Self-organizing maps

Similarly to convolutional networks, the results of the self-organizing map depend on the preprocessing. Downscaling the probability density images is an additional preprocessing

step performed to reduce the computational complexity of the algorithm. Downscaled images of 50-by-50 pixels are sufficient for good clustering and computationally reasonable. Good values for the other preprocessing parameters are approximately the same as for convolutional networks, but the importance of light blurring is enhanced. Likewise, different systems require preprocessing with different parameters for the best results, which necessitates subjective judgment from the researcher.

The SOM should contain enough nodes so that all interesting scarred states are captured. In practice a 5-by-3 grid appears to be a reasonable choice. However, this will result in many of the nodes containing non-scarred states. Additionally, scars of the same kind, but with different orientations, may also occupy multiple nodes. Therefore, the final classification of the SOM nodes is left for the researcher.

An example of the model images of the SOM for the r^5-system is presented in Fig. 4.2. The global ordering property of the map can be noticed. The resulting clustering is shown in Fig. C.1. The quality of the clustering is comparable to that of achieved by the convolutional network. The clusters with circular states are more distinct, resulting in arguably better clustering. The heptagram-shaped scars remain problematic and are not properly clustered due to being too weak. It is possible to further cluster the states associated with some particular node to gain further insight into the structure and purity of the cluster. The states belonging to node $\mathscr{C}_{14}$ are further clustered in Fig. 4.3. This reveals scaling in the central region of the scar, and the reason for the fuzziness around the edges in the original node, as some states are closer to circular with some minor overlapping pentagram-like characteristics.

Similarly an example of the model images in the r^2-system are illustrated in Fig. 4.4 and the resulting clustering in Fig. C.2. It can be noticed that the SOM does not solve the problems encountered in clustering the r^2-system. Regardless, the most prominent scars are clustered to their own nodes, while for the weaker scars the decisive factor for clustering is the orientation of the brightest vertices at the outer edges of the scars. This is evident in Fig. 4.5 where the node $\mathscr{C}_2$ has been further clustered. All the scars share the same orientation and fivefold symmetry, while the exact shape of the scar varies.

The global ordering property of the SOM is convenient for exploring the distribution of different states within the system. This is further illustrated in Fig. C.3 where a SOM was taught to represent the r^5-system on a 12-by-6 grid. A disadvantage of the SOMs, especially with larger maps, is that the method is computationally more demanding than the other studied methods.

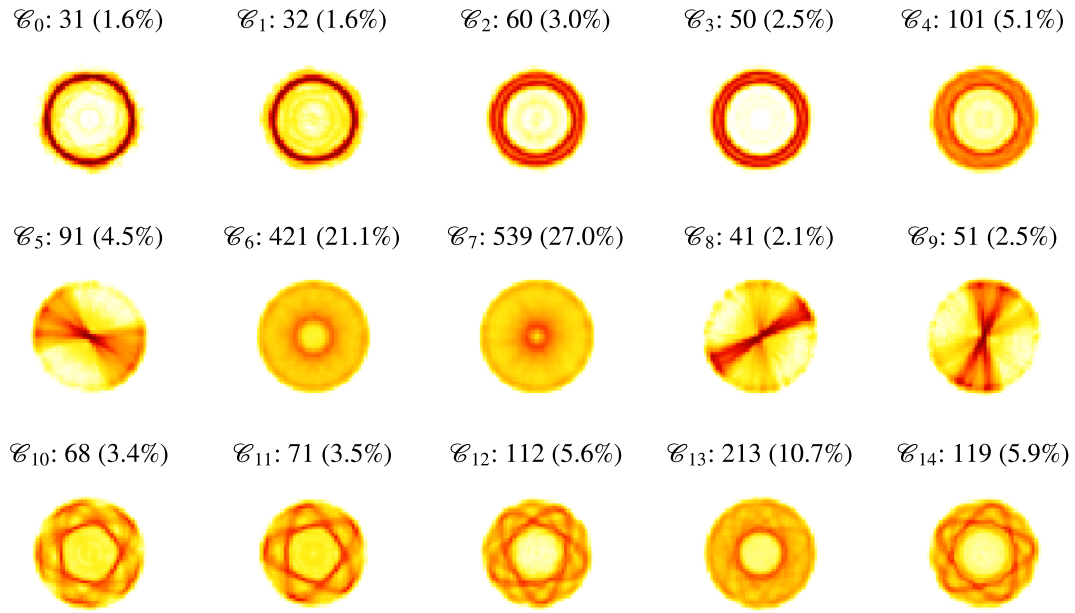


Figure 4.2. Example of SOM nodes the r^5-system. *The model images for the nodes of the SOM are shown. The SOM was taught to cluster 50-by-50 downscaled probability density images with preprocessing parameters* $F_c = 0.95$, $\gamma = 4$, *and* $\sigma = 3$. *The learning parameters of the SOM were* $T^{(1)} = 1000$, $\alpha_0^{(1)} = 0.9$, $\beta_0^{(2)} = 0.5$, $T^{(2)} = 10000$, $\alpha_0^{(2)} = 0.02$, *and* $\beta_0^{(2)} = 0.05$. *The labels indicate the absolute and relative number of states associated with each node.*

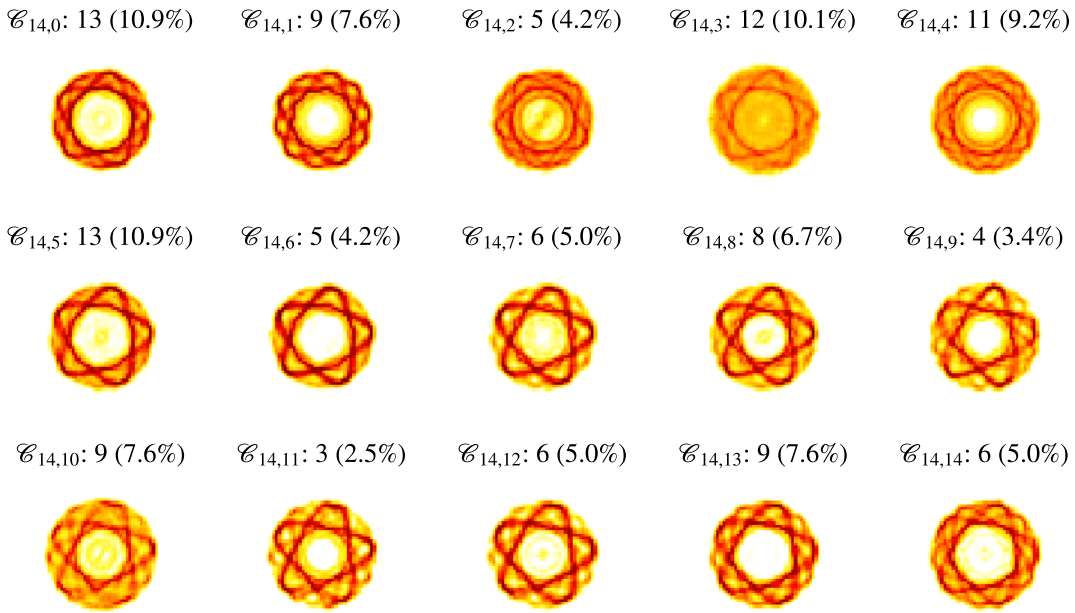


Figure 4.3. Example of SOM nodes for subclustering the node $\mathscr{C}_{14}$ from Fig. 4.2.

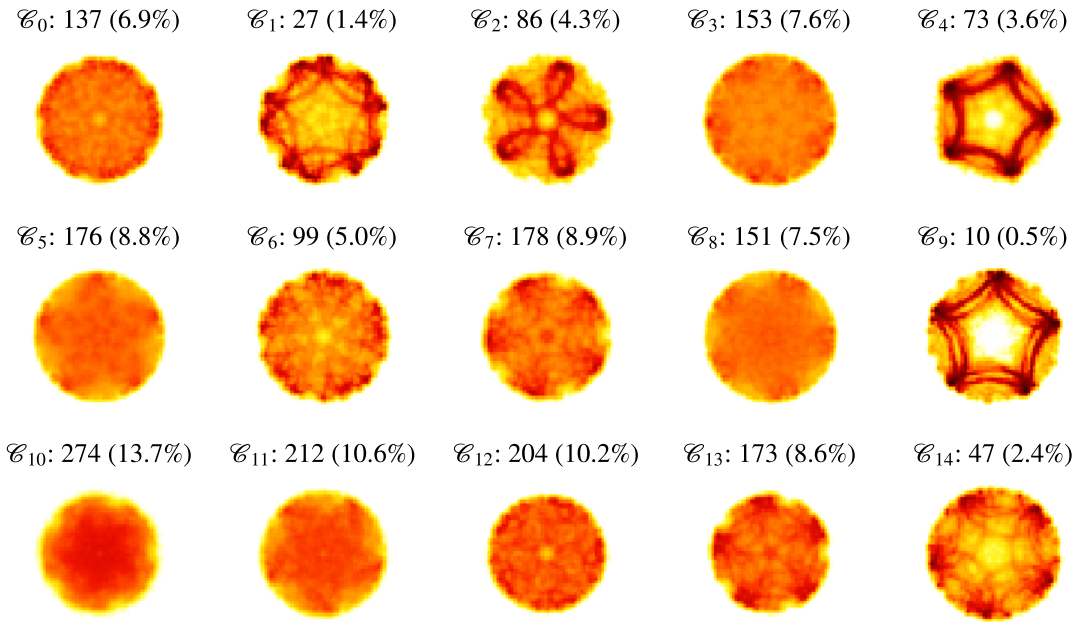


Figure 4.4. Example of SOM nodes the r^2-system. *The model images for the nodes of the SOM are shown. The SOM was taught to cluster 50-by-50 downscaled probability density images with preprocessing parameters* $F_c = 0.85$, $\gamma = 4$, *and* $\sigma = 3$. *The learning parameters of the SOM were* $T^{(1)} = 1000$, $\alpha_0^{(1)} = 0.9$, $\beta_0^{(2)} = 0.5$, $T^{(2)} = 10000$, $\alpha_0^{(2)} = 0.02$, *and* $\beta_0^{(2)} = 0.05$. *The labels indicate the absolute and relative number of states associated with each node.*

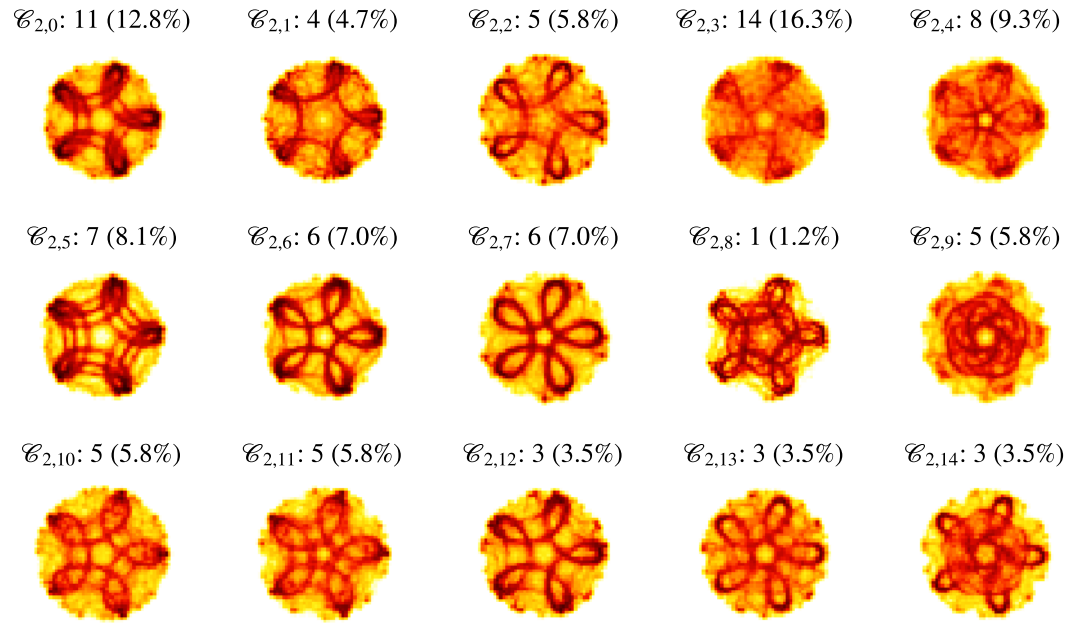


Figure 4.5. Example of SOM nodes for subclustering the node $\mathscr{C}_2$ from Fig. 4.4.

5. DISCUSSION AND CONCLUSIONS

During the thesis it was noticed that the general unsupervised problem of detecting and classifying the scarred eigenstates is more complicated than expected. Nevertheless, the investigation provides valuable insight about the challenges of the problem; inadequate approaches were recognized and promising methods for further development were identified.

Further research into histogram-based feature vectors is unlikely to result in a breakthrough, but a more thorough investigation of convolutional networks is warranted. Besides experimenting with different network layouts, some elements of the network are immediately apparent targets for improvements. While random kernels generally perform adequately [8, p. 357], kernels adapted to the actual data are expected to provide superior performance. A possible solution would be to cluster random segments of the images with k-means, and utilize the learned centroids as kernels [15]. Another enhancement would be to design the network to take desired symmetries into account. Max pooling already makes the network invariant to small translations and minor scaling. Rotations, and other symmetry operations, could be taken into account by constructing parallel layers in the network where these symmetry operations have been applied to the kernels. The consecutive layers would choose the outputs from the preceding layers with the highest activations.

Entirely different methods for constructing the feature vectors could be considered as well. The eigenstates of the perturbed system can be approximated by a linear combination of the eigenstates of the corresponding *unperturbed* system. The coefficients of these expansions could be utilized as feature vectors. Additionally, if good quantum numbers can be constructed for the unperturbed states, these combined with the coefficients could be exploited for actual classification instead of merely clustering.

The self-organizing maps are also good candidates for further improvements. The number of nodes in the maps is an important parameter that has to be adjusted by hand. It may not be clear how many nodes are required to properly represent all the different kinds of scars present in the system. Even though relatively low number of nodes may reveal the prominent scar geometries, the clusters are noisier and likely to contain many outliers. These problems could be overcome by modifying the algorithm to dynamically adjust the number of nodes in the network, and more generally, allow the dynamic adjustment of the network topology.

The scars form closed orbits marked by high probability density. Therefore it could be beneficial to attempt to directly trace these orbits and form a representation for them. A suitable approximation of the orbits could be easier to normalize for rotations, scale, and other desired symmetries. It can be a fair assumption that in the absence of any clear orbits

the representations would not converge towards any definite representation resembling the orbits, allowing scarred and non-scarred states to be distinguished. Furthermore, if the orbits are precise enough, they could allow wave packet analysis for more quantitative study of scar strengths.

The orbit tracing was preliminarily attempted by one-dimensional SOMs with closed loop topology for its nodes. The nodes hold (x, y)-coordinates of points in the probability density image, learned by randomly sampling points proportionally to the probability density. While these representations do not exactly follow the orbits, they are distinct for different kinds of scars. Nevertheless, the subsequent clustering of these representations was not of comparable quality to the other methods considered in this thesis, while being significantly more expensive computationally.

Another strategy for finding the orbits could be based on convolution with suitably designed kernels. As can be seen in Fig. 2.2, convolution by a kernel containing a line element highlights areas where brighter contents of the image are aligned with the line. Systematically rotating the kernel and following the hot spots in the convolution could allow tracing the orbits.

If the requirement for generality is relaxed, a specialized solution could be devised for systems with circularly symmetric unperturbed potentials. In this case the classical periodic orbits can be enumerated relatively easily [3], which allows straightforward matching of these orbits onto the eigenstates. Not only would this allow meaningful classification, but also direct quantification of the strength of the scarring by the standard method of wave packet analysis. Computationally lighter alternative could be to determine the fraction of the probability density residing nearby the orbits.

Presently the methods rely on the researcher performing the final classification of the produced clustering. Nevertheless, the current methods already enable further study of the scarred systems. For example, they can be utilized for exploring the effects of the perturbations on the scarring phenomenon. This is due to the observation that the orientations of the scars are relatively insensitive to the amplitude and standard deviation of the noise [3]. Therefore, as long as the bump positions remain fixed, the effects of these parameters can be studied by the same clustering that has been once classified.

REFERENCES

[1] H.J. Stöckmann, Quantum Chaos: An Introduction, Cambridge University Press, 1999.

[2] E.J. Heller, Bound-state eigenfunctions of classically chaotic hamiltonian systems: Scars of periodic orbits, Physical Review Letters, Vol. 53, Iss. 16, 1984, pp. 1515–1518.

[3] P.J.J. Luukko, B. Drury, A. Klales, L. Kaplan, E.J. Heller, E. Räsänen, Strong quantum scarring by local impurities, Scientific Reports, Vol. 6, 2016, p. 37656.

[4] J. Keski-Rahkonen, P.J.J. Luukko, L. Kaplan, E.J. Heller, E. Räsänen, Controllable quantum scars in semiconductor quantum dots, Phys. Rev. B, Vol. 96, Sep, 2017, p. 094204.

[5] D. Griffiths, Introduction to Quantum Mechanics, Pearson Prentice Hall, Pearson international edition, 2005.

[6] P.J.J. Luukko, E. Räsänen, Imaginary time propagation code for large-scale two-dimensional eigenvalue problems in magnetic fields, Computer Physics Communications, Vol. 184, Iss. 3, 2013, pp. 769–776.

[7] B. Kramer, A. MacKinnon, Localization: theory and experiment, Reports on Progress in Physics, Vol. 56, Iss. 12, 1993, p. 1469.

[8] I. Goodfellow, Y. Bengio, A. Courville, Deep learning, MIT Press, Cambridge, MA, 2017.

[9] G. James, D. Witten, T. Hastie, R. Tibshirani, An introduction to statistical learning: with applications in R, Springer, New York, 2013.

[10] D. Arthur, S. Vassilvitskii, K-means++: The advantages of careful seeding, in: Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, Philadelphia, PA, USA, 2007, Society for Industrial and Applied Mathematics, SODA '07, New Orleans, Louisiana, pp. 1027–1035.

[11] J.H. Ward, Hierarchical grouping to optimize an objective function, Journal of the American Statistical Association, Vol. 58, Iss. 301, 1963, pp. 236–244.

[12] T. Kohonen, Self-organizing maps, 3rd ed., Springer, Berlin, 2001.

[13] A.A. Akinduko, E.M. Mirkes, A.N. Gorban, Som: Stochastic initialization versus principal components, Information Sciences, Vol. 364-365, 2016, pp. 213–221.

[14] M.S. Nixon, A.S. Aguado, Feature Extraction and Image Processing for Computer Vision, 3rd ed., Academic Press, GB, 2012.

[15] A. Coates, A. Ng, H. Lee, An analysis of single-layer networks in unsupervised feature learning, in: Gordon, G., Dunson, D., Dudík, M. (eds.), Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, Fort Lauderdale, FL, USA, 11–13 Apr, 2011, Proceedings of Machine Learning Research 15, PMLR, pp. 215–223.

APPENDIX A: EXAMPLES OF HISTOGRAM-BASED CLUSTERING

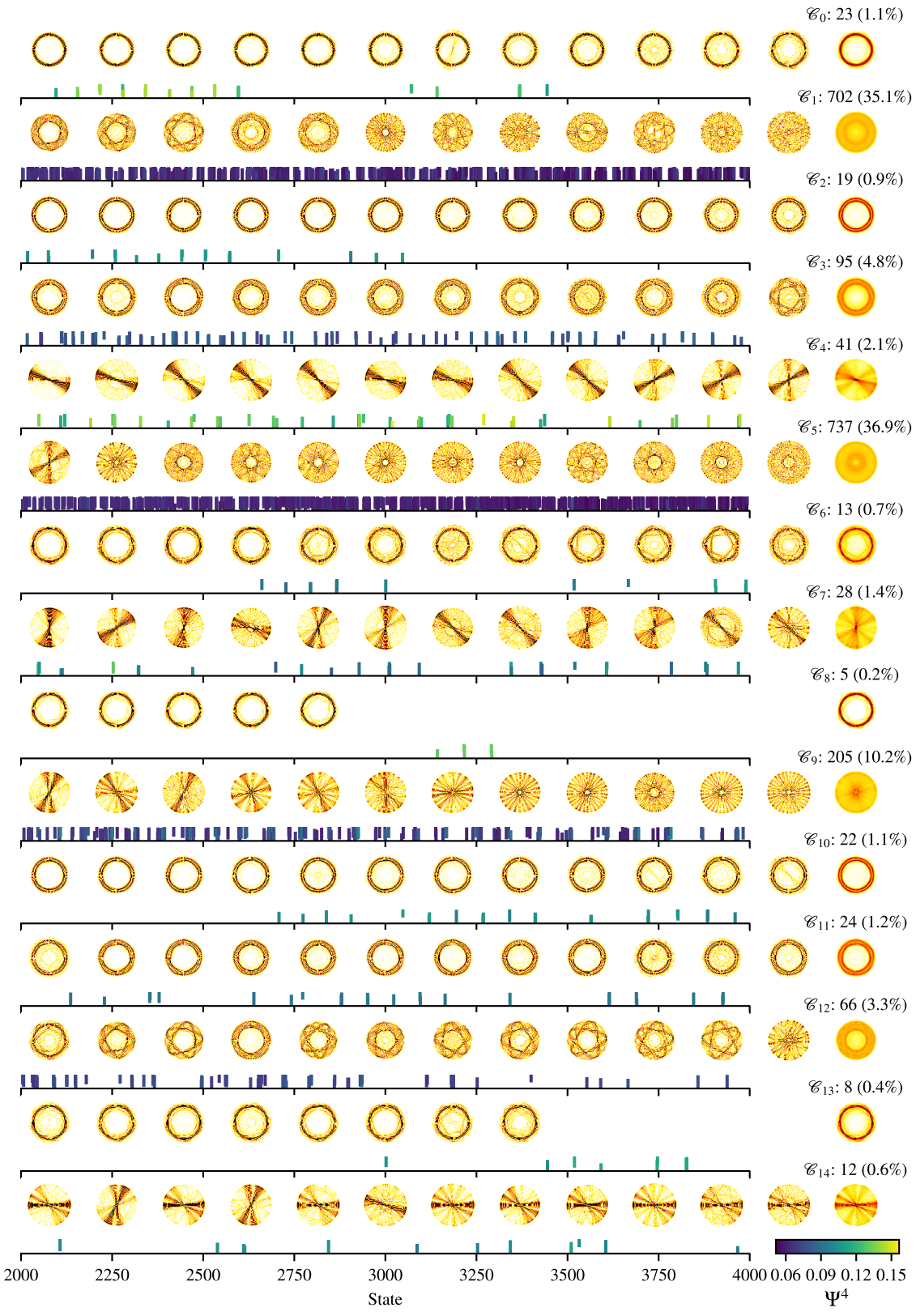


Figure A.1. Example of relative frequency subgrid histogram clustering. *The probability density grids in the r^5-system were divided into 10-by-10 subgrids. Histograms with 10 bins were clustered by k-means into 10 clusters. The relative frequencies of the subgrid histogram clusters were normalized to zero mean and unit variance, and then clustered with hierarchical clustering utilizing L^1 metric and complete linkage.*

Figure A.2. Example of subgrid histogram image clustering in the r^5-system. *The probability density grids were divided into 10-by-10 subgrids. Histograms with 10 bins were clustered by k-means into 2 clusters. The histogram clusters of each state were interpreted as binary images and were clustered with hierarchical clustering utilizing L^1 metric and complete linkage.*

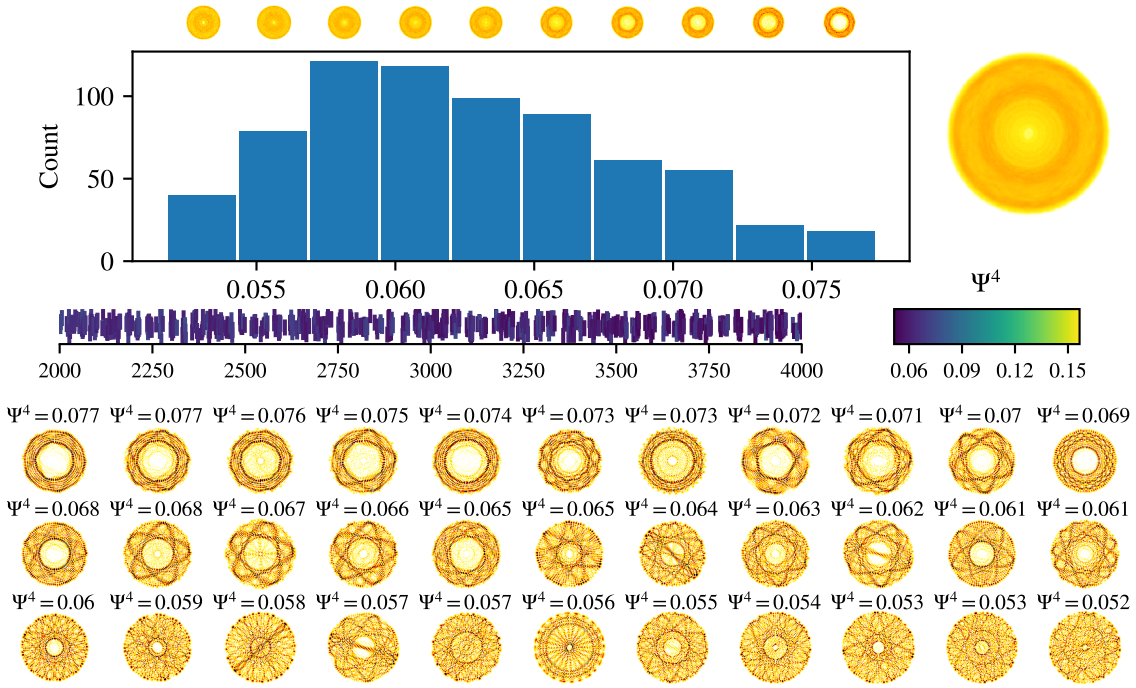


Figure A.3. Cluster $\mathscr{C}_1$ from Fig. A.1.

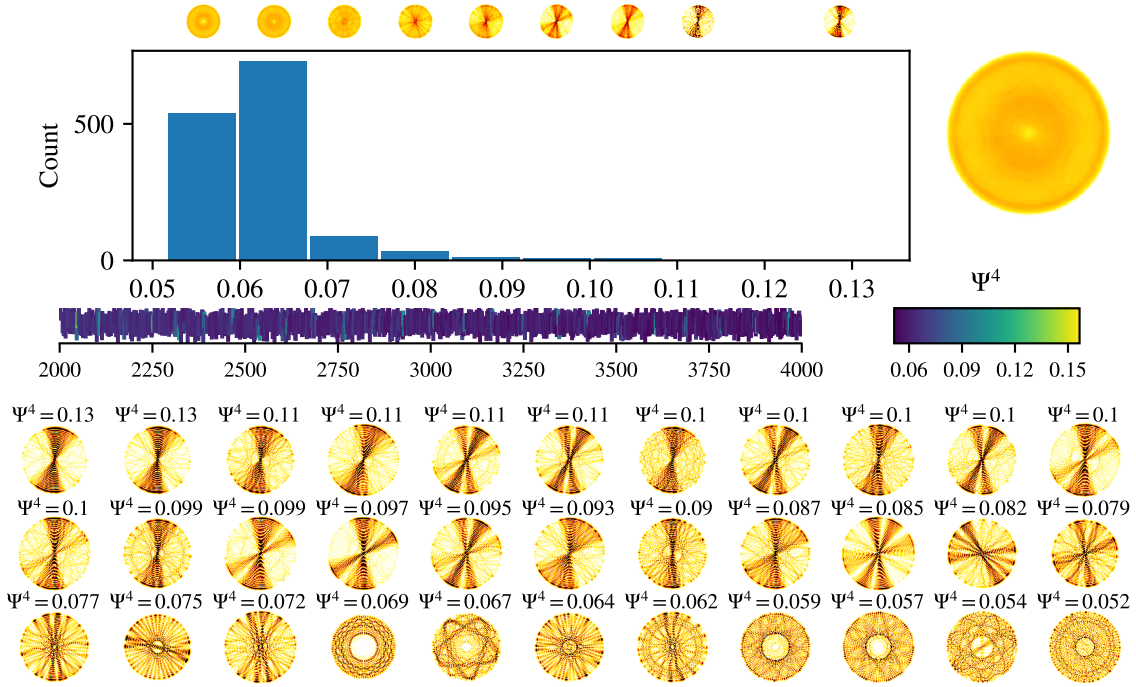


Figure A.4. Cluster $\mathscr{C}_1$ from Fig. A.2.

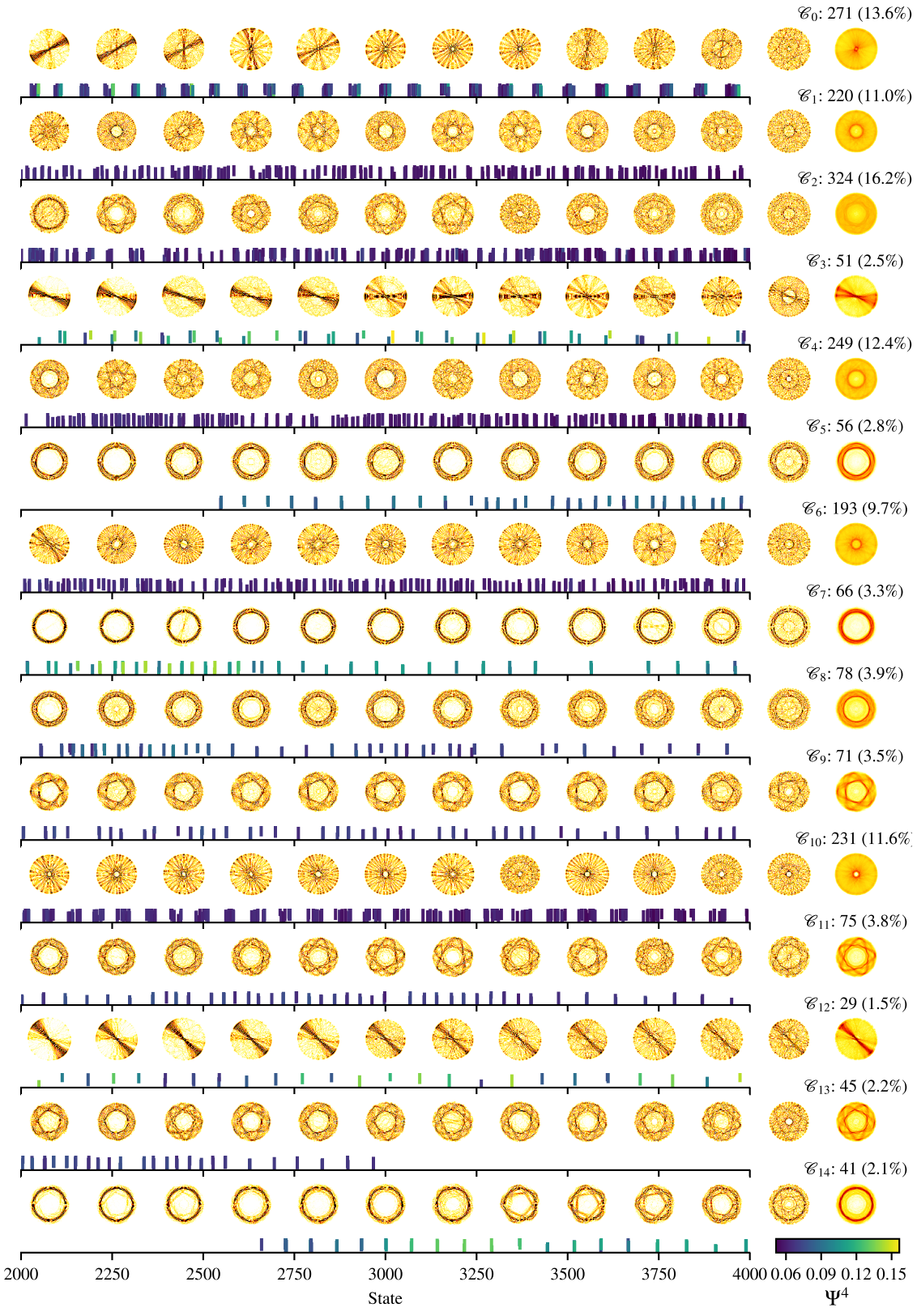


Figure A.5. Example of subgrid histogram image clustering in the r^5-system. *The probability density grids were divided into 10-by-10 subgrids. Histograms with 10 bins were clustered by k-means into 2 clusters. The histogram clusters of each state were interpreted as binary images and were clustered with hierarchical clustering utilizing Euclidean metric and Ward's linkage criterion.*

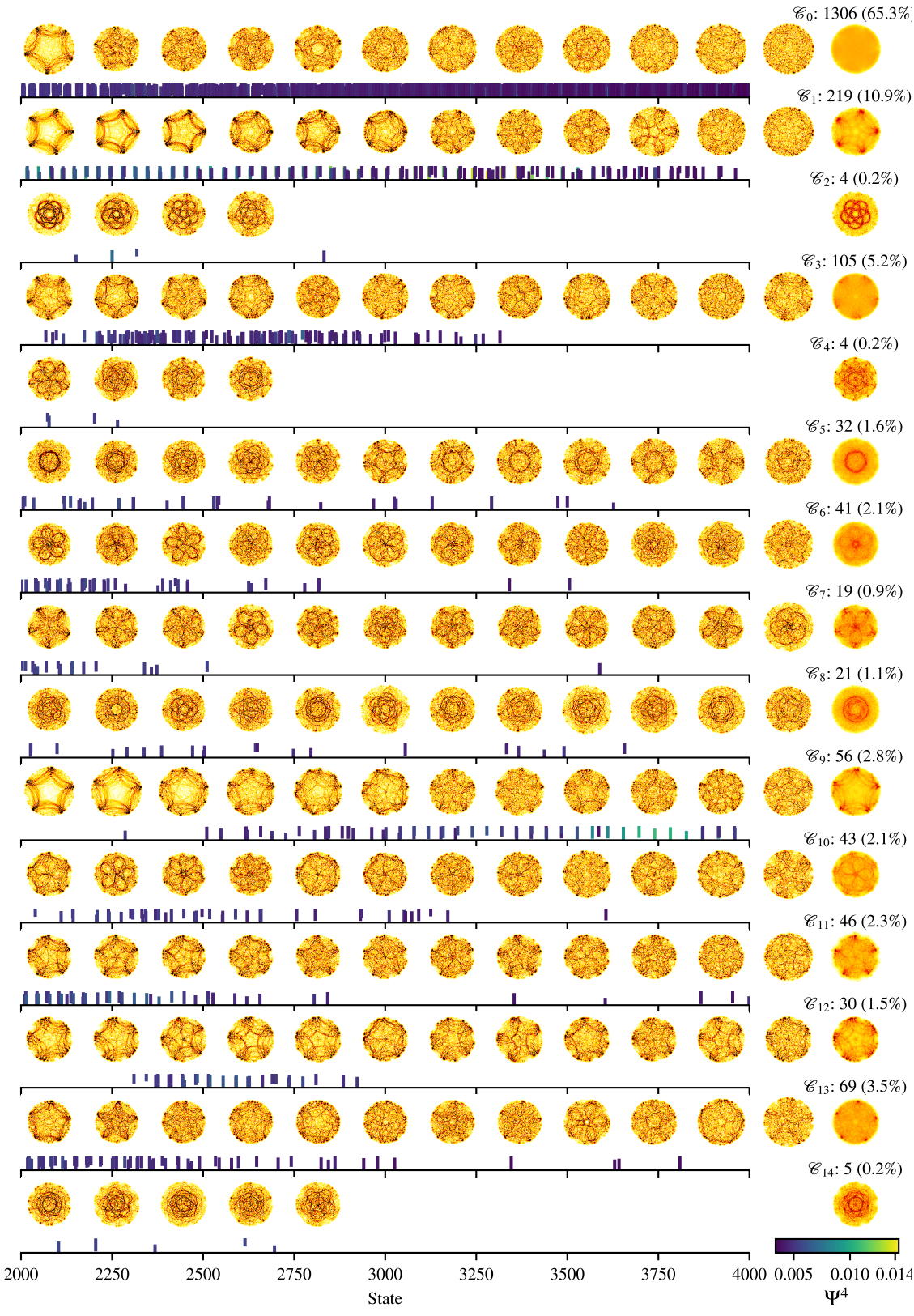


Figure A.6. Example of subgrid histogram image clustering in the r^2-system. *The probability density grids were divided into 10-by-10 subgrids. Histograms with 10 bins were clustered by k-means into 2 clusters. The histogram clusters of each state were interpreted as binary images and were clustered with hierarchical clustering utilizing L^1 metric and complete linkage.*

Figure A.7. Example of subgrid histogram image clustering in the r^2-system. *The probability density grids were divided into 10-by-10 subgrids. Histograms with 10 bins were clustered by k-means into 2 clusters. The histogram clusters of each state were interpreted as binary images and were clustered with hierarchical clustering utilizing Euclidean metric and Ward's linkage criterion.*

APPENDIX B: EXAMPLES OF CONVOLUTIONAL NETWORK CLUSTERING

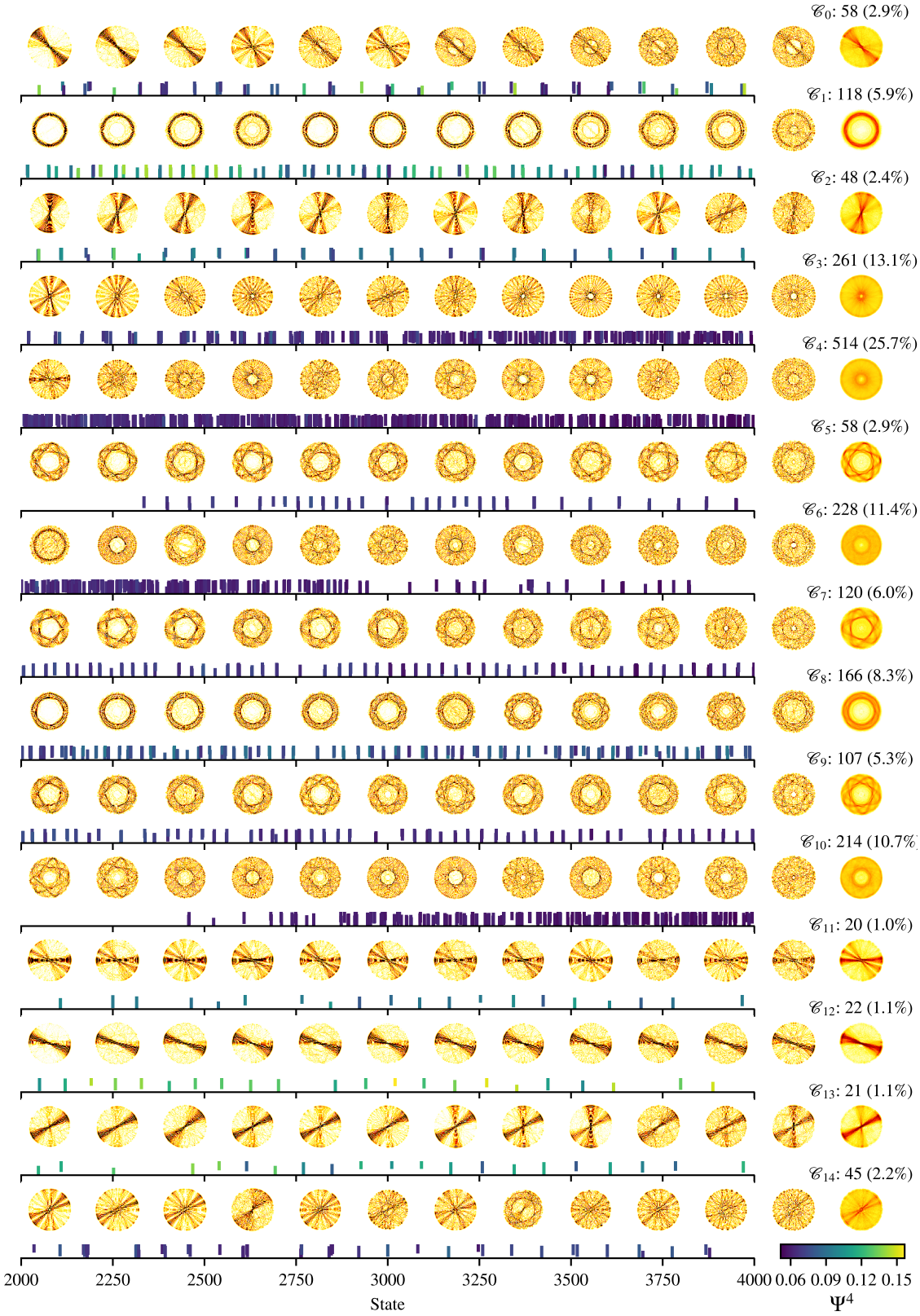


Figure B.1. Clustering of the r^5-system by convolutional network. *The probability density images were preprocessed with parameters* $F_c = 0.95$, $\gamma = 4$, *and* $\sigma = 3$ *prior to extracting the feature vectors. The normalized feature vectors were clustered by hierarchical clustering utilizing Euclidean metric and Ward's linkage criterion.*

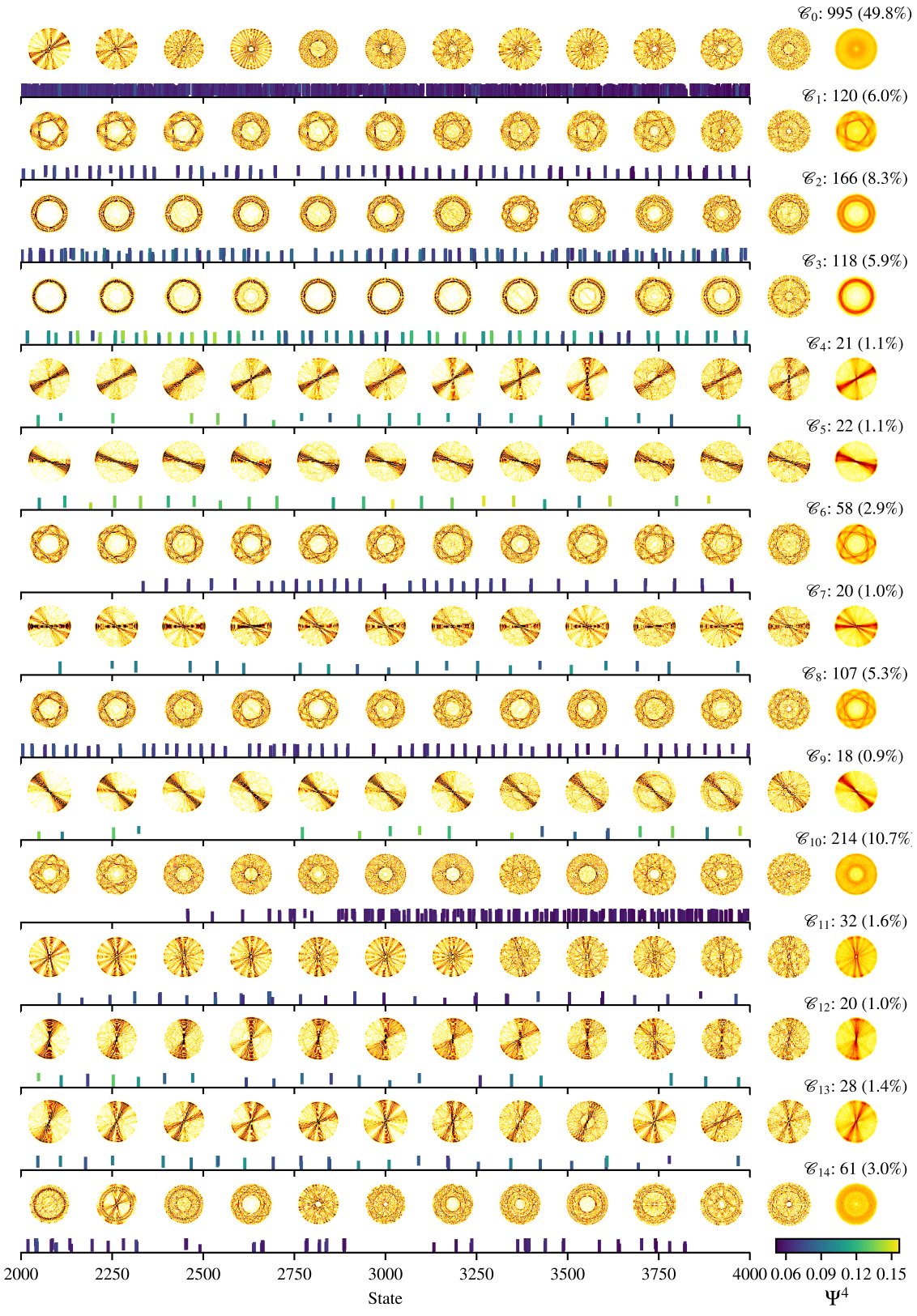


Figure B.2. Iterative clustering of the r^5-system by convolutional network. *The probability density images were preprocessed with parameters* $F_c = 0.95$, $\gamma = 4$, *and* $\sigma = 3$ *prior to extracting the feature vectors. The normalized feature vectors were iteratively clustered with the feature vectors as test vectors. Each iteration employed hierarchical clustering with 8 clusters. The clustering utilized Euclidean metric and Ward's linkage criterion.*

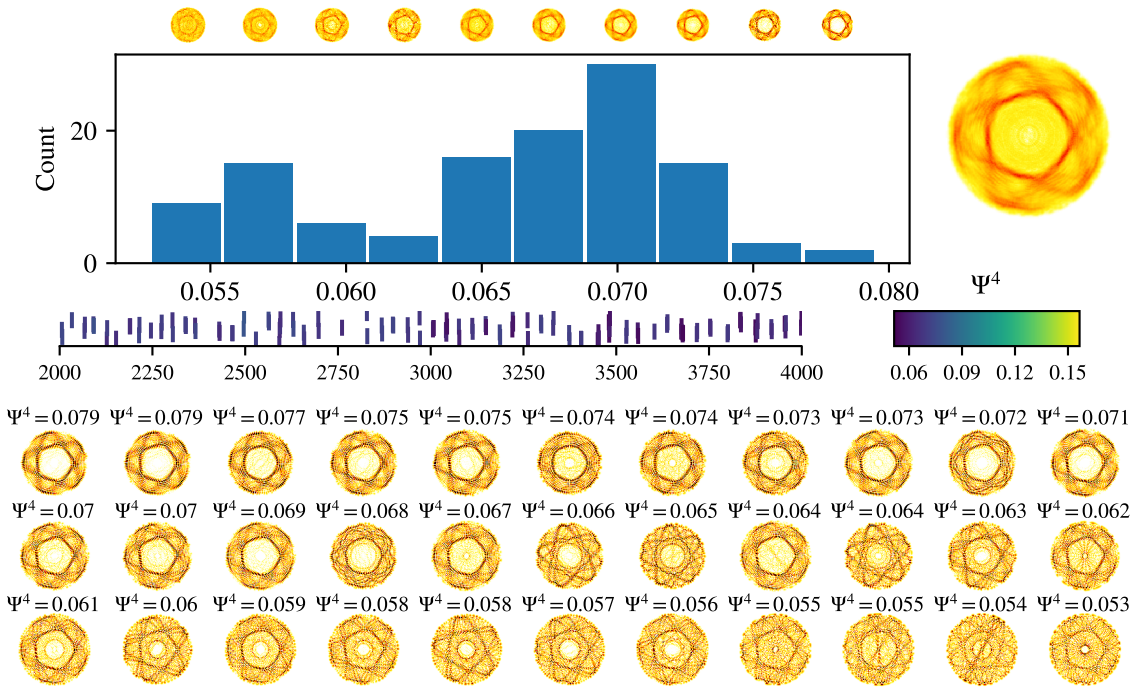


Figure B.3. Cluster $\mathscr{C}_1$ from Fig. B.2.

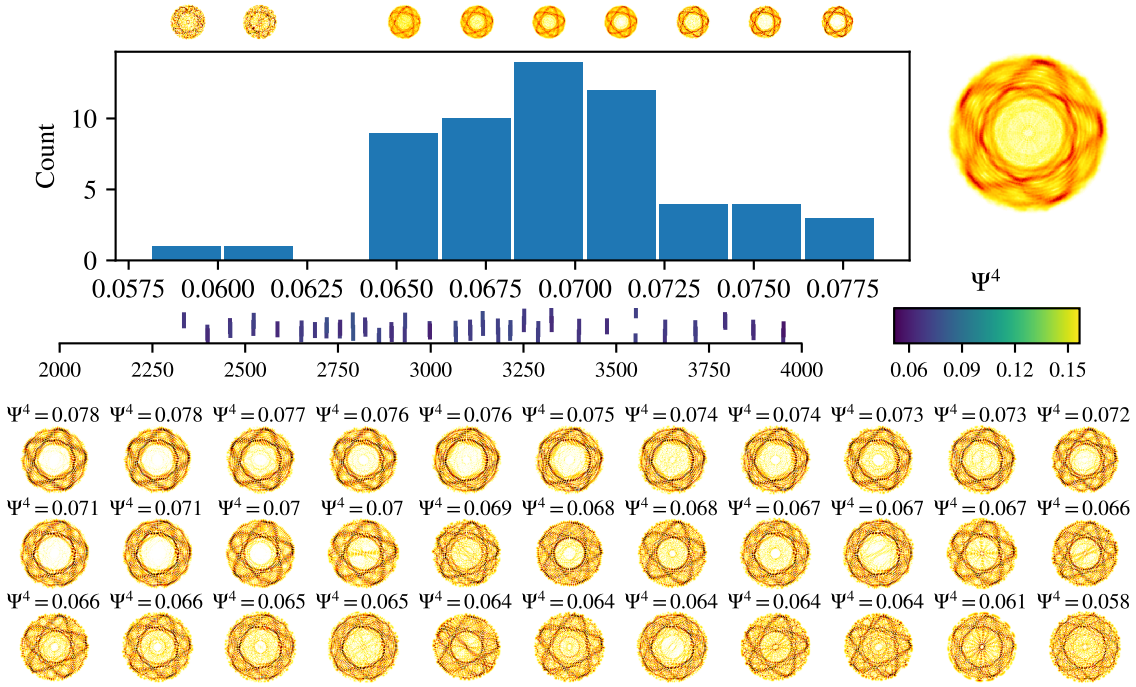


Figure B.4. Cluster $\mathscr{C}_6$ from Fig. B.2.

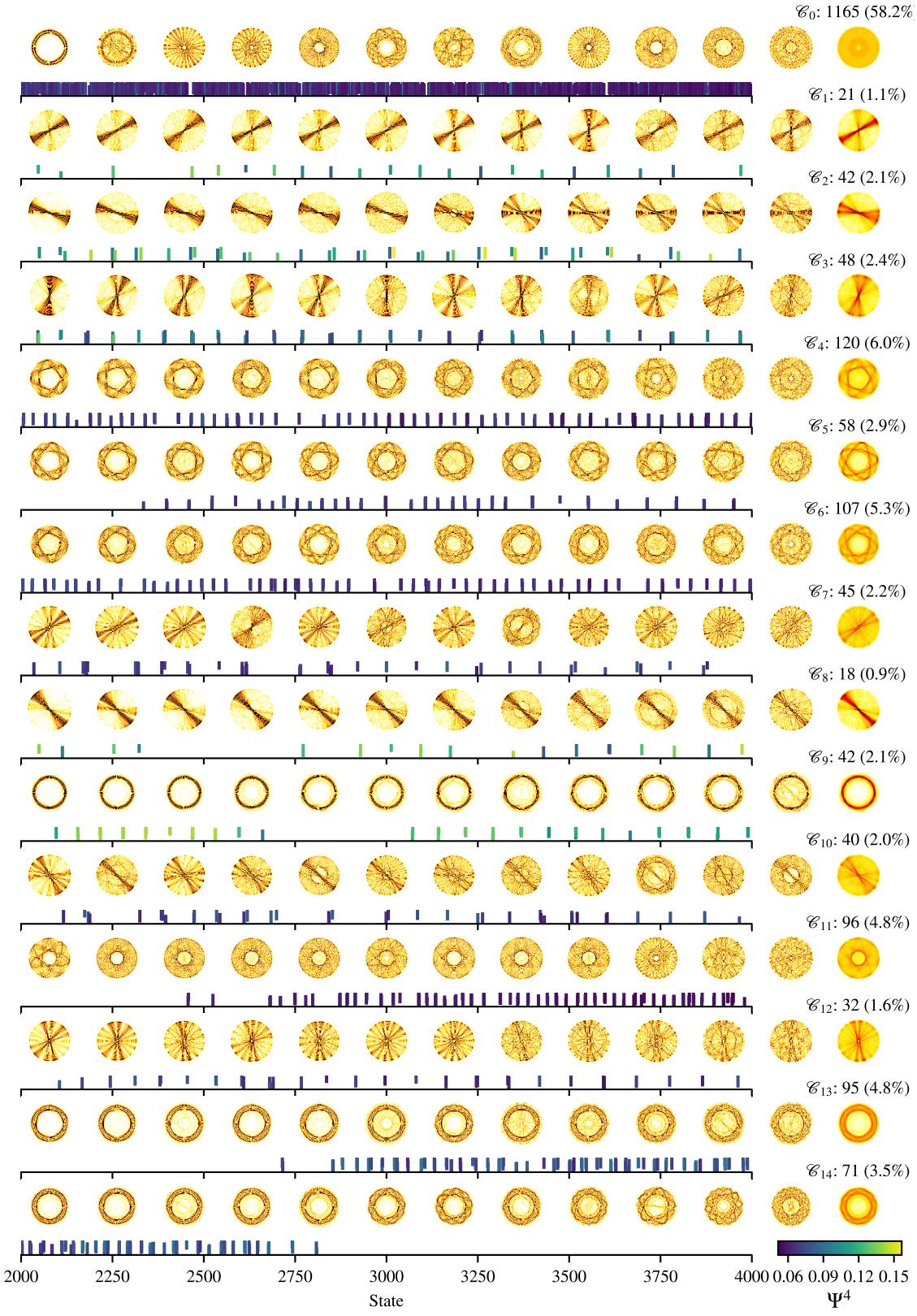


Figure B.5. Iterative clustering of the r^5-system by convolutional network. *The probability density images were preprocessed with parameters* $F_c = 0.95$, $\gamma = 4$, *and* $\sigma = 3$ *prior to extracting the feature vectors. The normalized feature vectors were iteratively clustered with the probability density grids as test vectors. Each iteration employed hierarchical clustering with 8 clusters. The clustering utilized Euclidean metric and Ward's linkage criterion.*

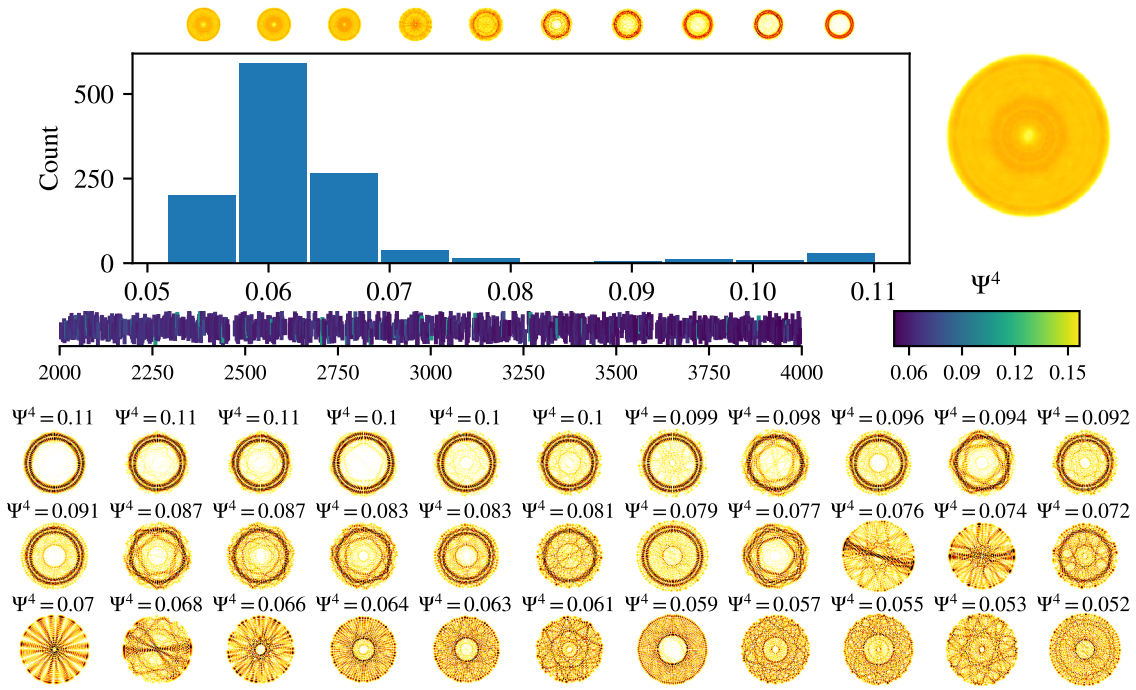


Figure B.6. Cluster $\mathscr{C}_0$ from Fig. B.5.

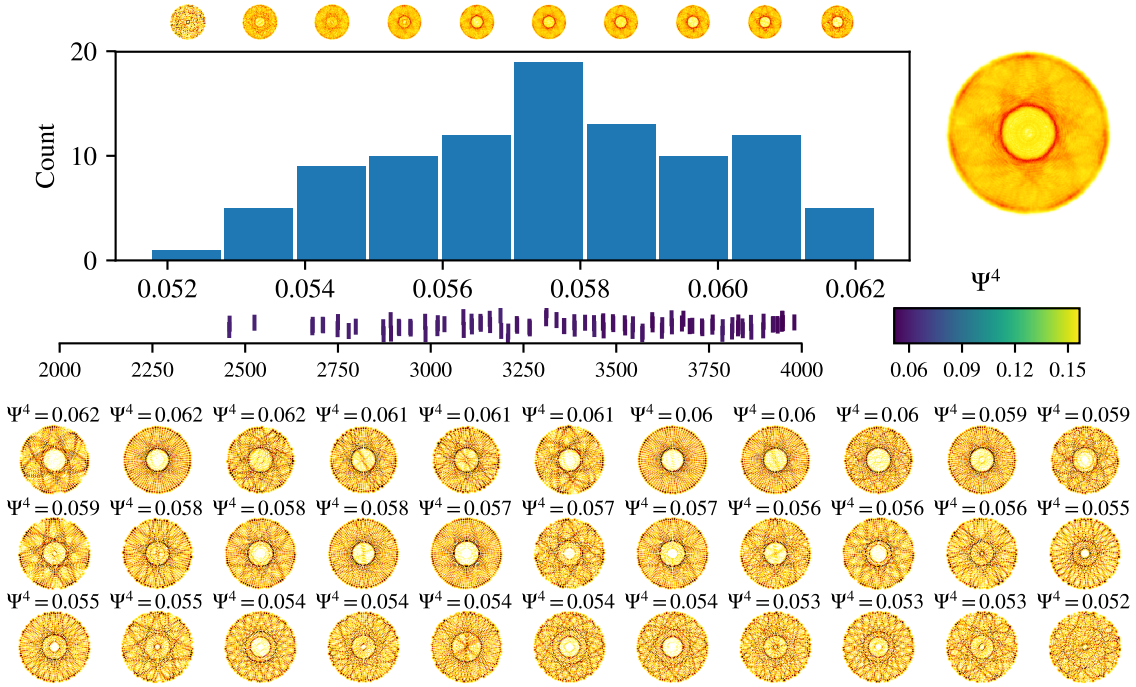


Figure B.7. Cluster $\mathscr{C}_{11}$ from Fig. B.5.

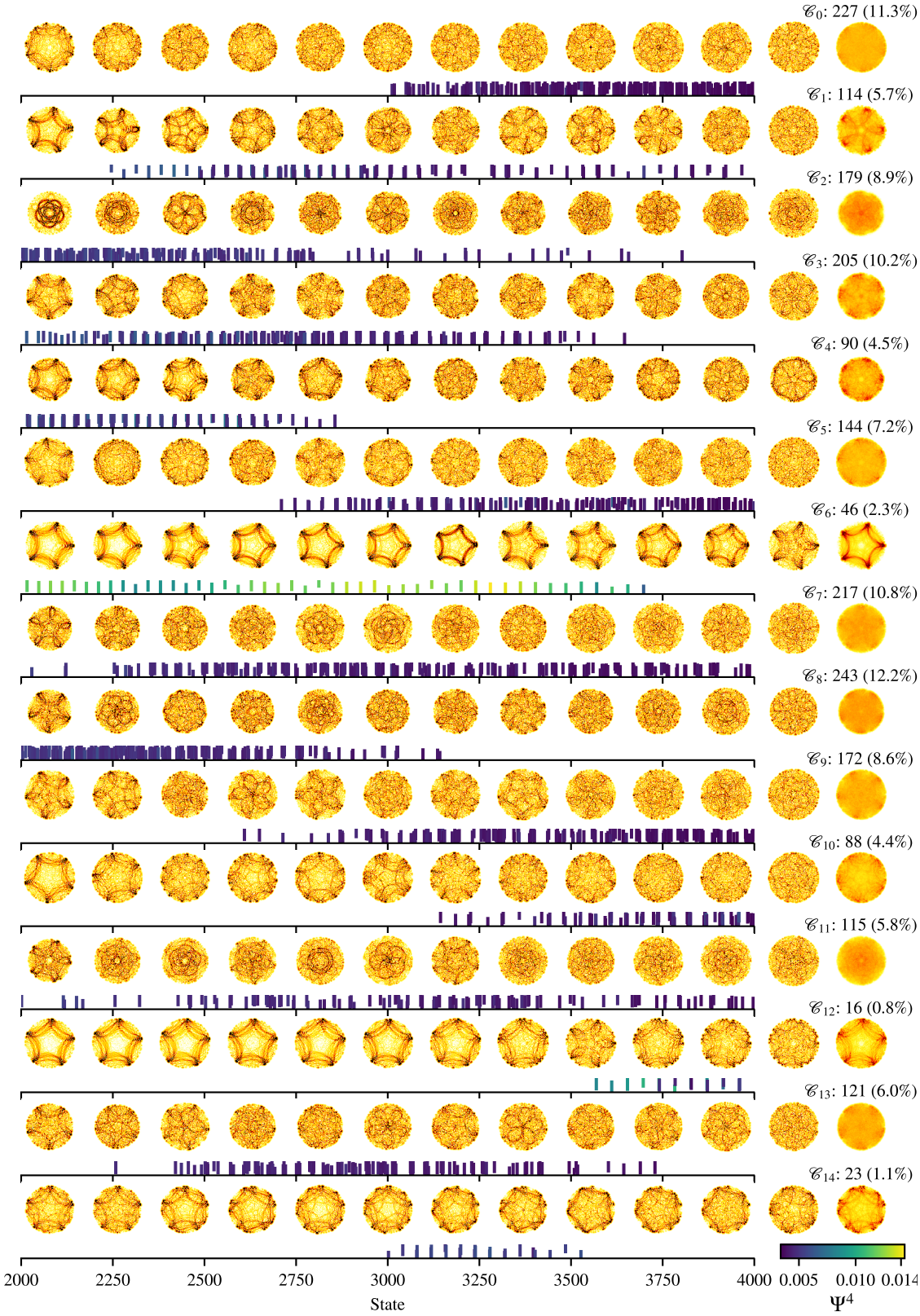


Figure B.8. Clustering of the r^2-system by convolutional network. *The probability density images were preprocessed with parameters* $F_c = 0.85$, $\gamma = 4$, *and* $\sigma = 3$ *prior to extracting the feature vectors. The normalized feature vectors were clustered by hierarchical clustering utilizing Euclidean metric and Ward's linkage criterion.*

Figure B.9. Iterative clustering of the r^2-system by convolutional network. *The probability density images were preprocessed with parameters* $F_c = 0.85$, $\gamma = 4$, *and* $\sigma = 3$ *prior to extracting the feature vectors. The normalized feature vectors were iteratively clustered with the feature vectors as test vectors. Each iteration employed hierarchical clustering with 15 clusters. The clustering utilized Euclidean metric and Ward's linkage criterion.*

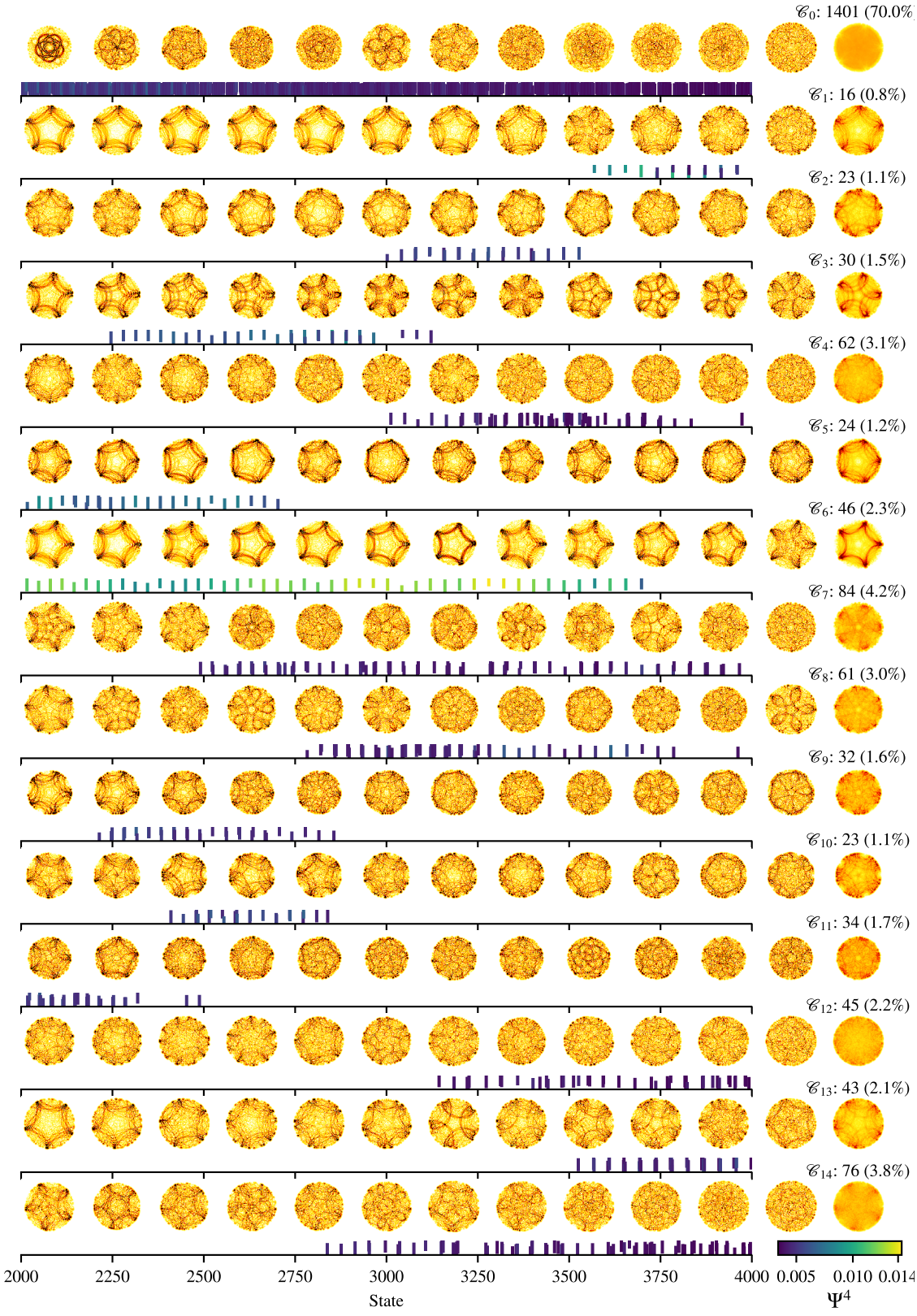


Figure B.10. Iterative clustering of the r^2-system by convolutional network. *The probability density images were preprocessed with parameters* $F_c = 0.85$, $\gamma = 4$, *and* $\sigma = 3$ *prior to extracting the feature vectors. The normalized feature vectors were iteratively clustered with the probability density grids as test vectors. Each iteration employed hierarchical clustering with 15 clusters. The clustering utilized Euclidean metric and Ward's linkage criterion.*

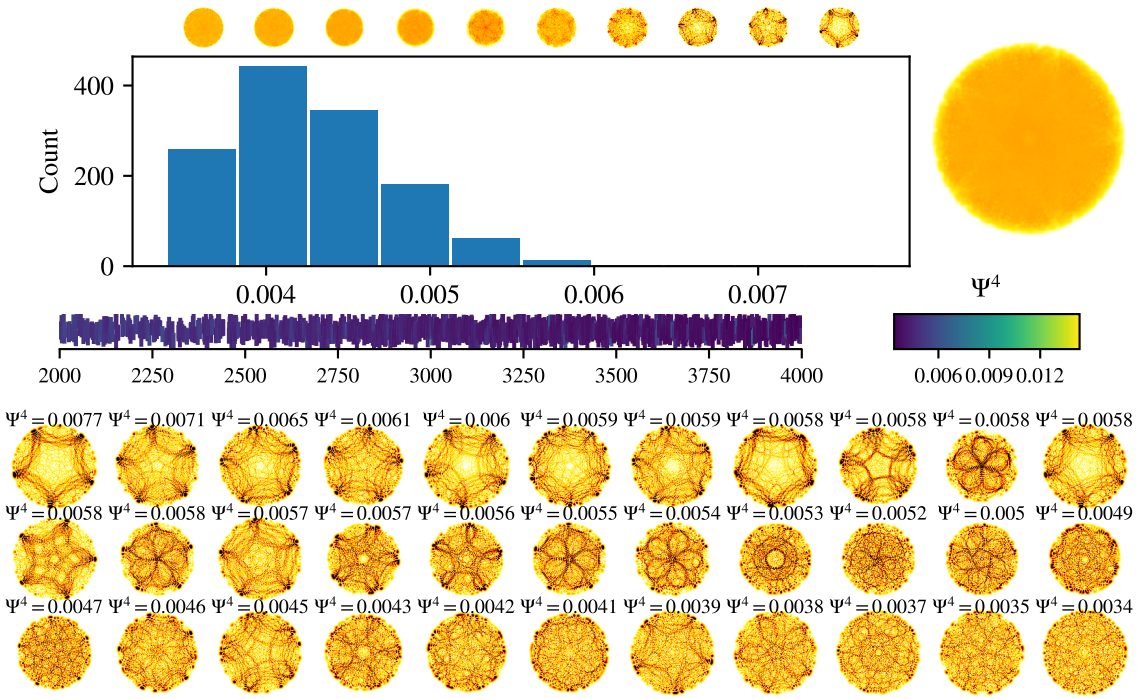


Figure B.11. Cluster $\mathscr{C}_0$ from Fig. B.9.

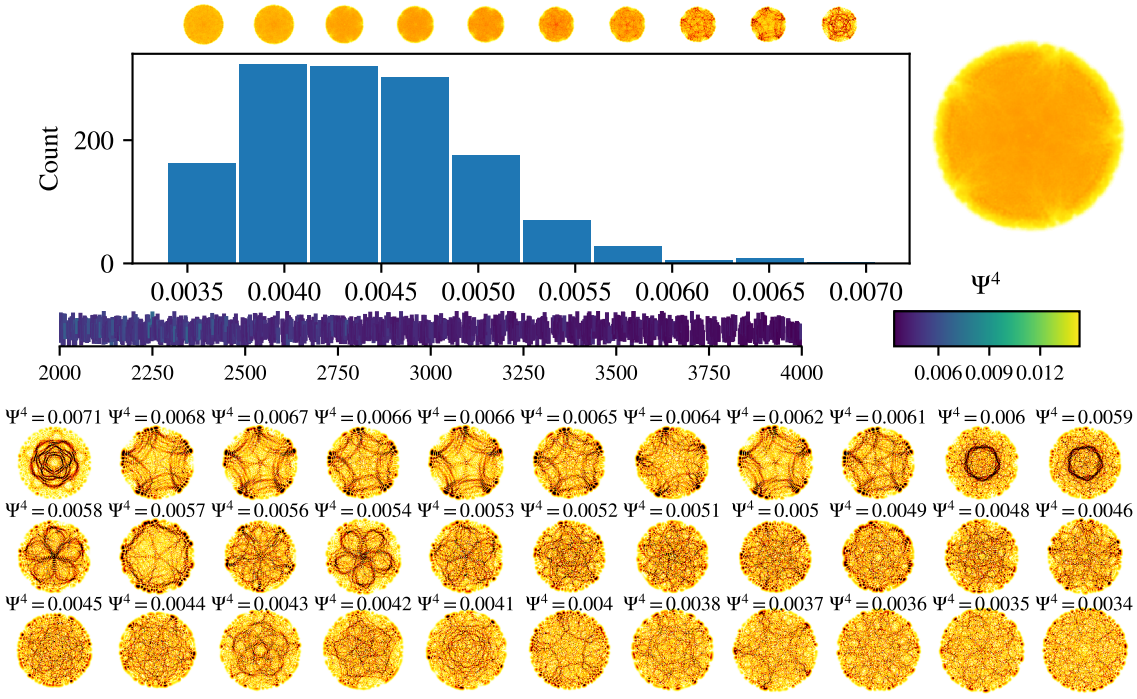


Figure B.12. Cluster $\mathscr{C}_0$ from Fig. B.10.

APPENDIX C: EXAMPLES OF SOM CLUSTERING

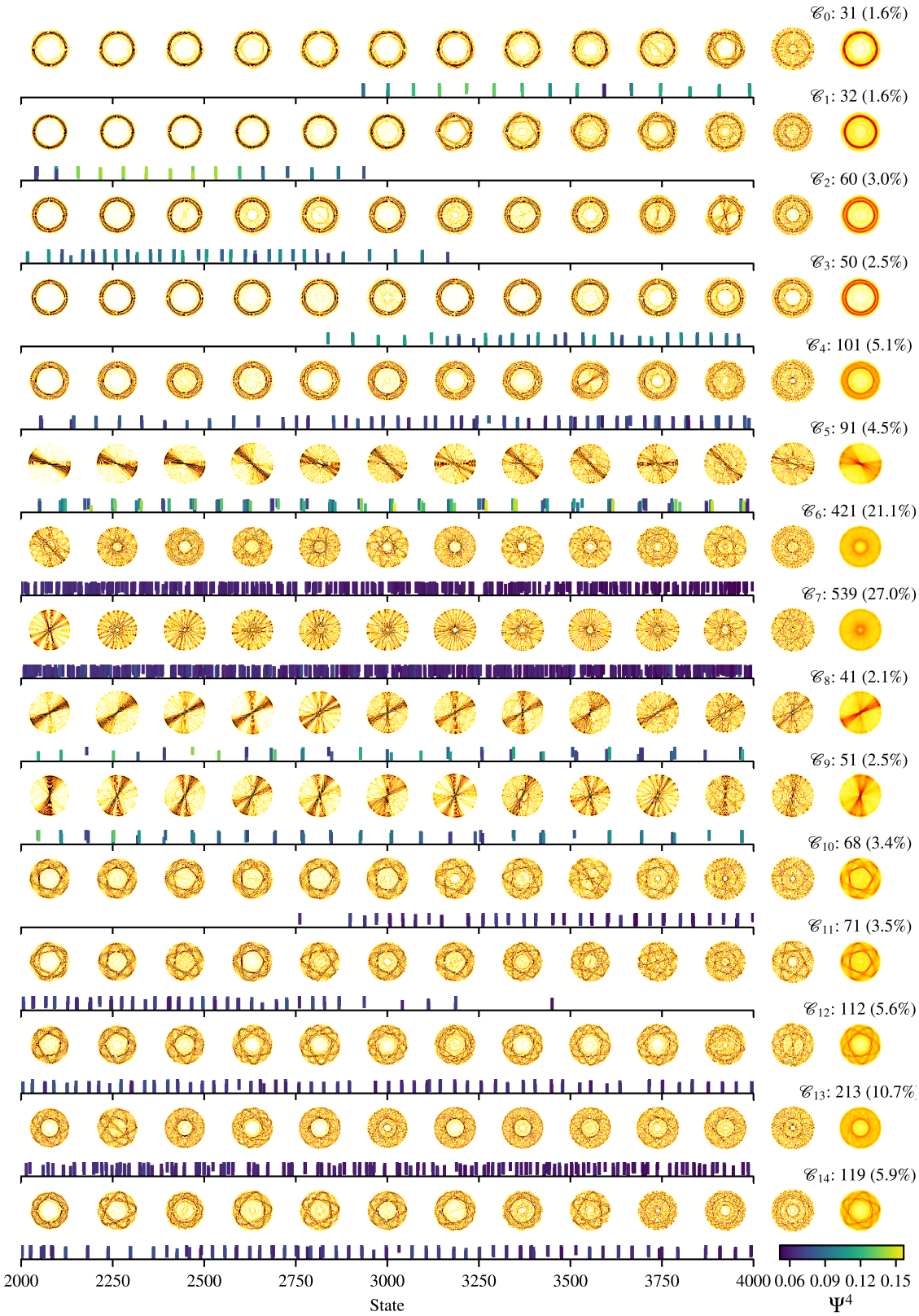


Figure C.1. Clustering of the r^5-system by SOM. *The SOM was taught to cluster 50-by-50 downscaled probability density images with preprocessing parameters* $F_c = 0.95$, $\gamma = 4$, *and* $\sigma = 3$. *The learning parameters of the SOM were* $T^{(1)} = 1000$, $\alpha_0^{(1)} = 0.9$, $\beta_0^{(2)} = 0.5$, $T^{(2)} = 10000$, $\alpha_0^{(2)} = 0.02$, *and* $\beta_0^{(2)} = 0.05$.

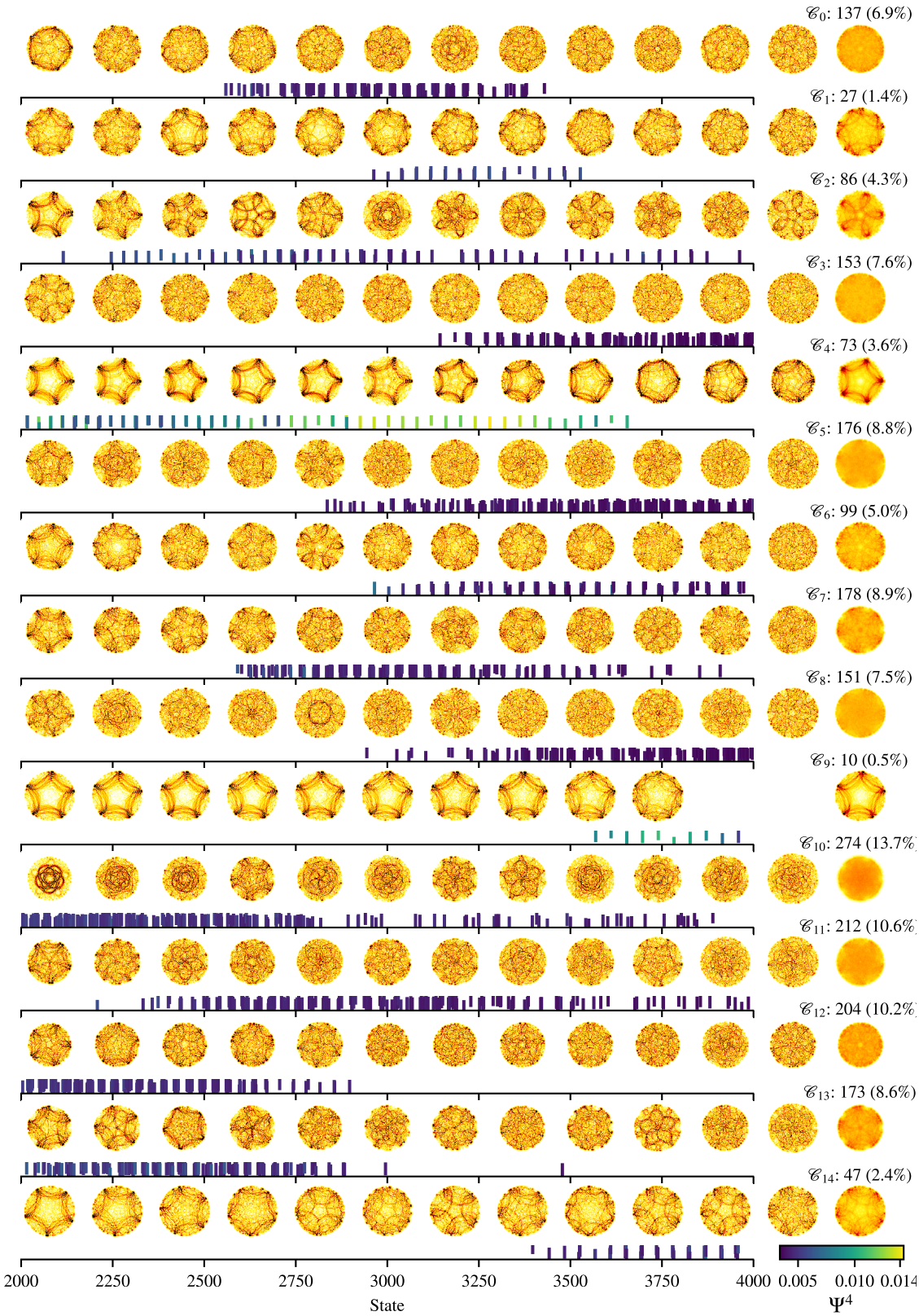


Figure C.2. Clustering of the r^2-system by SOM. *The SOM was taught to cluster 50-by-50 downscaled probability density images with preprocessing parameters* $F_c = 0.85$, $\gamma = 4$, *and* $\sigma = 3$. *The learning parameters of the SOM were* $T^{(1)} = 1000$, $\alpha_0^{(1)} = 0.9$, $\beta_0^{(2)} = 0.5$, $T^{(2)} = 10000$, $\alpha_0^{(2)} = 0.02$, *and* $\beta_0^{(2)} = 0.05$.

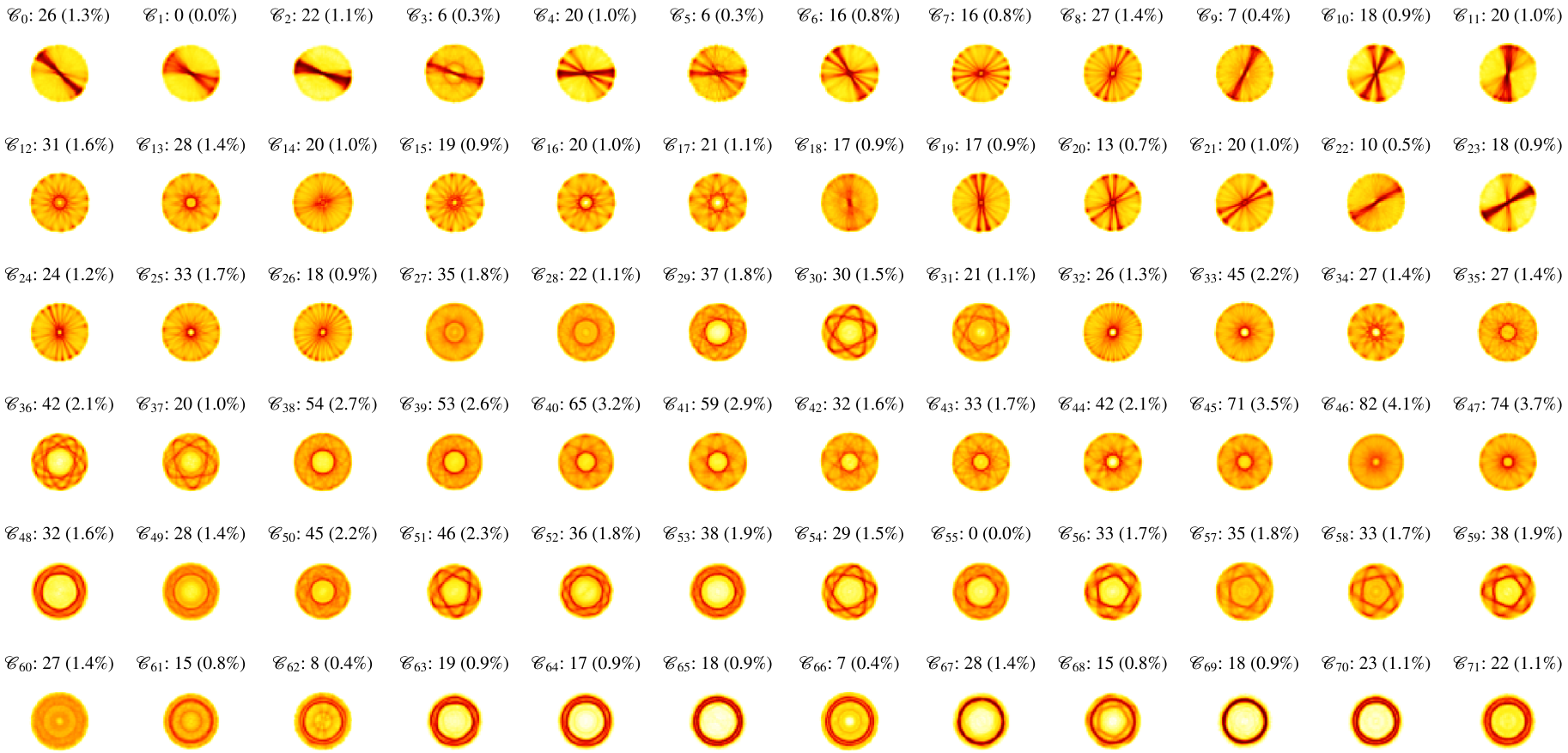


Figure C.3. 12-by-6 SOM for the r^5-system. *The model images for the nodes of the SOM are shown. The SOM was taught to cluster 50-by-50 downscaled probability density images with preprocessing parameters* $F_c = 0.98$, $\gamma = 1$, *and* $\sigma = 3$. *The learning parameters of the SOM were* $T^{(1)} = 1000$, $\alpha_0^{(1)} = 0.9$, $\beta_0^{(2)} = 0.5$, $T^{(2)} = 50000$, $\alpha_0^{(2)} = 0.02$, *and* $\beta_0^{(2)} = 0.05$.