



Manifold Structure of Artificial and Biological Neural Networks

Zhang Liu

**Capstone Final Report for BSc (Honours) in
Mathematical, Computational and Statistical Sciences**

Supervised by: Francesca Spagnuolo

Co-supervised by: Steven W. Zucker

AY 2021/2022

Abstract

Keywords: manifold learning, neural computation, early primate vision, computer vision

Our work builds on a long line of research aiming to develop more accurate computational models of the visual system. Despite decades of research, we have not fully understood the structure of neural circuits responsible for visual perception [26]. Additionally, among the computer vision (CV) community, there is a growing interest in investigating the limitations of the CV models by comparing them against biological vision. To this end, we seek to answer two specific open questions: First, how do CV models (VGG16 [48], Vision Transformer [15], and Convolutional Recurrent Neural Network [47]) compare to biological vision, at retina and primary visual cortex (V1), in terms of their respective neural circuits¹? Second, what specific mechanisms are important in causing such differences and/or similarities?

We first build the biological neural tensors using experimental neural spiking data and artificial neural tensors using numerical simulations on computer vision models (VGG16 [48], Vision Transformer [15], and Convolutional Recurrent Neural Network [47]). Using tensor CP decomposition, we obtain a manifold of neurons. The discrete data graph underlying the manifold then reflects both the neural circuit connections and the neurons' role in those circuits. By comparing the manifold structure of neurobiological networks in retina and V1 with that of computer vision models, we can

¹A neural circuit is a population of neurons interconnected by synapses to carry out a specific function when activated [23].

make precise inferences about similarities and differences in their respective functional circuits. For the first time, we find that the underlying neural circuit of feed-forward computer vision models including CNN and ViT form disconnected network clusters, making them poor approximations of the visual cortex, contrary to popular belief. In order to model the highly connected neural circuits in the visual cortex, recurrent structure is likely necessary.

Contents

Abstract	i
1 Introduction	1
2 Related works	9
3 Dimensionality Reduction	13
4 Biological Neural Networks	16
5 Artificial Neural Networks	22
6 Conclusion	35
Bibliography (Main)	38
Bibliography (Supplementary)	39
Acknowledgements	44
A Complete List of Experiments	45
B Diffusion Map	51
C Principal Component Analysis	56
D Tensor CP Decomposition on Face Recognition Dataset	58
E Persistent Homology Theory	60
F Topological Methods	70

1 Introduction

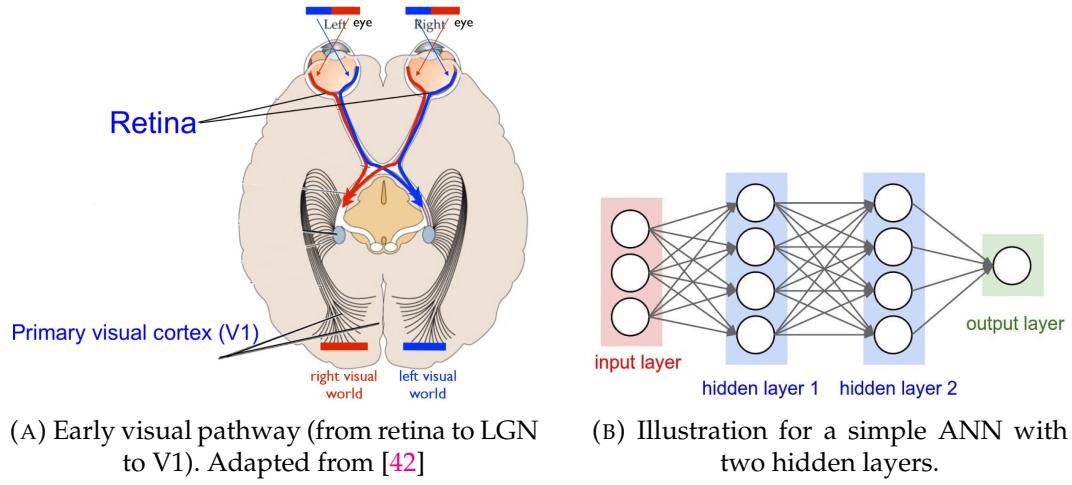
1.1 Motivation and significance

In vision science, core object recognition has largely been modeled as a feed-forward process down the visual ventral stream [13]. First of such feed-forward models was Hubel and Wiesel's hierarchy model [31], which classified the neurons in the visual cortex into simple and complex cells. This idea directly inspired the Neocognitron model [20], which in turn provided the basis for Convolutional Neural Networks (CNNs) [34],[35], which have become a dominant model in computer vision [59]. However, recent research starts to suggest that these feed-forward neural networks fail to model various key features in early primate vision [40], [51], [45], [33], [4]. In addition, across the different stages of early visual processing, the nature of neural computations vary drastically, particularly between the retina and the primary visual cortex (V1) [16].

To this end, we seek to investigate the following related open questions:

- How do computer vision (CV) models (VGG16 [48], Vision Transformer [15], and Convolutional Recurrent Neural Network [47]) compare to biological vision, at retina and primary visual cortex (V1), in terms of their respective neural circuits²?
- What specific mechanisms are important in causing such differences and/or similarities?

²A neural circuit is a population of neurons interconnected by synapses to carry out a specific function when activated [23].



The significance of answering these questions is two-fold. First, modeling the visual system has always been an important task in the field of neuroscience. Despite decades of research, we have not fully understood the structure of neural circuits responsible for visual perception [26]. Second, for the computer vision community, in addition to the existing research on explainability of computer vision models, there is a growing interest in investigating the limitations of these models by comparing them against biological vision. A growing number of research starts to reveal tasks that biological vision excels at while the state-of-the-art computer vision models fail. One notable example is that CNN is highly inferior to human vision in solving visual problems that involve relational reasoning [24], such as recognizing same-different relations in images [45], categorizing images based on the constituent parts [19], and comparing features in images [52]. The hope is that with a rigorous comparison, we can discover the specific mechanisms responsible for certain desired features of biological vision that the current computer vision models are lacking and adapt those mechanisms into computer vision models with relevant computational constraints in mind.

1.2 Framework of comparison

Both subjects of our study, artificial and biological neural networks, are instances of black box problem where the underlying function is unknown. The best approach to investigate the internal structure of a black box system is to probe the system with some input and record the output:

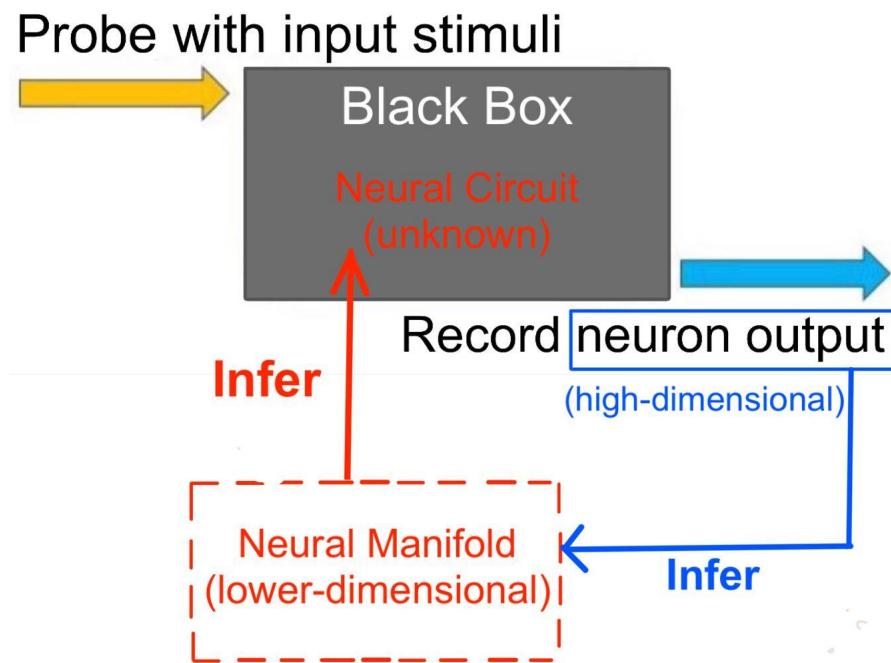


FIGURE 1.2: Illustration of the black box problem.

Applying this approach to our problem, we compare the neural circuits of artificial and biological neural networks (both unknown *a priori*) by comparing the neural output from each network when given the appropriate input. Since the neural output is encoded in high-dimensional representations, it is impossible to analyze the organization of neurons directly based on their responses to visual stimuli. For this reason, we need to infer an intermediate lower-dimensional representation, which we call "neural manifold" and will be our basis for comparison.

1.2.1 What is a neural manifold?

In order to formalize the concept of neural manifold, we first introduce the definition of a manifold as a mathematical object.

Definition 1 (Homeomorphism). *Let $f : X \rightarrow Y$ be a function between topological spaces X and Y . If f is bijective, then the inverse f^{-1} exists. If both f and f^{-1} are continuous, then f is a homeomorphism. The two topological spaces, X and Y , are said to be homeomorphic if there exists a homeomorphism between X and Y .*

Definition 2 (Manifold). *In brief, a real n -dimensional manifold is a topological space \mathcal{M} for which every point $x \in \mathcal{M}$ has a neighborhood homeomorphic to Euclidean space \mathbb{R}^n . Informally, a manifold is a topological space that locally resembles Euclidean space.*

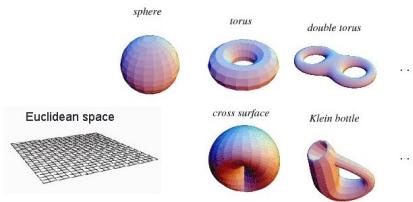


FIGURE 1.3: Examples of 2-dimensional manifolds.

Formally, the neural manifold is a lower-dimensional topological space embedded in the high-dimensional Euclidean space where the original neural output belongs. Why is the neural manifold still a good representation of the organization of neurons even if it is of a lower dimension? The general justification rests on “the Manifold Hypothesis³,” which states that naturally occurring data form lower-dimensional manifolds in the embedding space [39]. The neuroscientific justification is based on the established fact that despite the original high-dimensional representations, the possible patterns of

³the Manifold Hypothesis forms the basis of a collection of methods called Manifold Learning

neural output are confined to a lower-dimensional manifold spanned by a few independent patterns called “neural modes” [21], [54], [60].

Remark 3. *For ease of reference, we also use the term “neural manifold” for the point-cloud of data points with an underlying manifold structure. In addition, neural manifolds from real neural data are often no longer “manifolds” in a mathematical sense, due to the sparse input sampling and the presence of neural noise.*

1.2.2 *What can we analyze from a neural manifold?*

Neural manifold is an intermediate step from the neural output towards inferring the neural circuits. On the neural manifold, each point represents a neuron. The distance between a pair of points indicates how similar the firing patterns of the corresponding neurons are, and equivalently, how likely they participate in the same neural circuit. To infer the (unknown) neural circuit from a neural manifold (known), we note that there are three possible cases: if neurons were originally sampled

1. from a collection of isolated neural circuits (decomposable networks),
then the neurons would have distinct stimuli preferences, yielding a discontinuous neural manifold.
2. from overlapping neural circuits (partially decomposable networks),
most neurons would respond to multiple stimuli, leading to a smooth transition in stimuli preferences and a continuous neural manifold.
3. from fully connected neural circuits (non-decomposable networks), all neurons would respond to all stimuli, and there would be no stimuli preferences across groups of neurons. This would yield a zero-dimensional (i.e., degenerate) manifold.

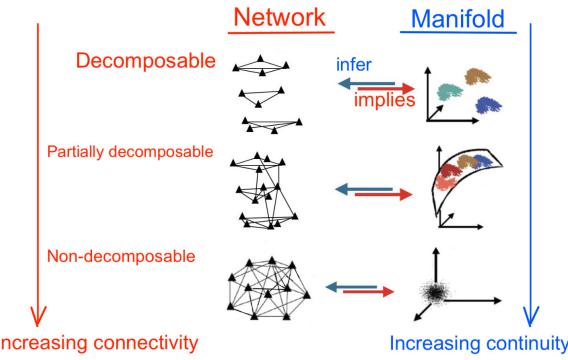


FIGURE 1.4: Connections between neural manifolds and neural circuits. Adapted from [16].

To sum up, by comparing the continuity of neural manifolds arising from artificial and biological neural networks, we can make precise inferences about their similarities and differences of their respective neural circuits, which is what we set out to compare in 1.1.

1.3 Problem statement

Our starting point was the result in [16]: based on the respective neural manifolds, the retinal ganglion cells differ fundamentally from the functional cells in V1 which are distributed more continuously.

Therefore, our objective, stated in concrete terms, is to find out among the computer vision models, which yield continuous neural manifold like V1 and which yield discontinuous neural manifold like the retina. The candidate models are CNN, Vision Transformer (ViT), and Recurrent Neural Networks (RNN), and their respective input and output are summarized in the following table.

	Biological	Artificial
Models	retina, V1	CNN, ViT, RNN
Input	flow stimuli [17]	ImageNet images [12]
Output	single-neuron recording	artificial neuron output

TABLE 1.1: Summary of comparison.

1.4 Highlights and thesis organization

During the project, we have made the following original contributions:

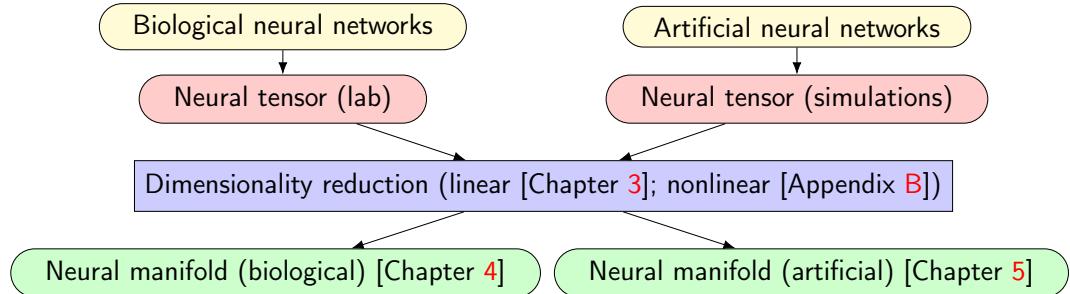
1. conducted new computational experiments on neural data in the retina,
2. developed an original algorithm to investigate the manifold structure of pre-trained Convolutional Neural Network (CNN) and how it evolves over the course of training,
3. developed an original algorithm to investigate the manifold structure of pre-trained Convolutional Recurrent Neural Network (CRNN),
4. developed an original algorithm to investigate the manifold structure of pre-trained Vision Transformer (ViT),
5. compared and analyzed (both qualitatively and quantitatively) the manifold structure of CNN, ViT, and CRNN, against that of the neurobiological networks in retina and V1,
6. conclusion that could advance research in both biological vision and computer vision.

The code for this project is available at the GitHub repository⁴.

The organization of the thesis follows the structure of the project in the flowchart below. After situating our work within the backdrop of related research in Chapter 2, we will provide the theoretical preliminaries for the linear dimensionality reduction method in Chapter 3 (its nonlinear counterpart is outlined in Appendix B). The experiments and results for biological and artificial neural networks will be elaborated in Chapters 4 and 5 respectively. In the end, we propose the conclusion of our work and future

⁴<https://github.com/zhang-liu-official/capstone-liu-2022>

directions.



2 Related works

2.1 Modeling biological vision

2.1.1 Hierarchy Models

Hubel and Wiesel's hierarchy model [31] was the first seminal work on modeling the visual system, which classified the neurons in the primary visual cortex (V1) into simple and complex cells. The essential difference between simple and complex cells is that the responses of simple cells are modulated by the spatial phase of a sine grating, whereas the responses of complex cells are largely phase invariant. In other words, as we progress from simple cells to complex cells, the neurons become selective for increasingly complex stimuli and become more tolerant to the exact position within their receptive fields. Based on this, a natural way to construct complex cells is to group responses from simple cells that have the same orientation preference but different phase preferences.

This idea directly inspired the Neocognitron model [20]. In the Neocognitron model, simple cells (termed as "S-cells" in the original paper) are tuned to simple stimuli at the convolution layers. Their outputs are then combined at the pooling layers by taking the maximum or average to form complex cells (termed as "C-cells" in the original paper). As a result, complex cells are tuned to more complex stimuli. The Neocognitron model was among a myriad of hierarchical models of the visual system and inspired the structure of CNN.

The figure below shows a systematic illustration of the general idea behind the hierarchical models:

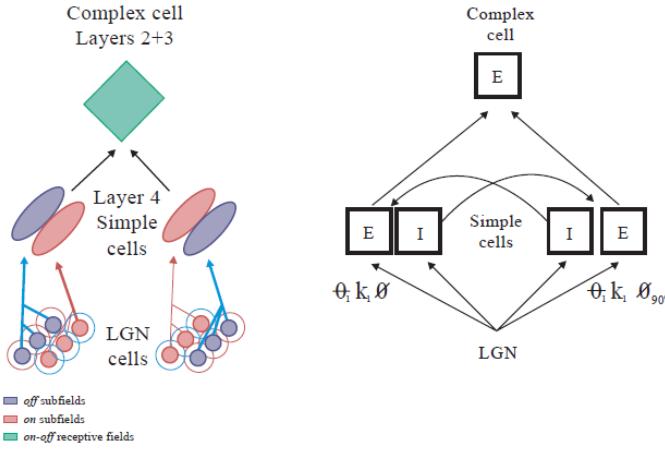


FIGURE 2.1: Illustration of the hierarchical models [37].

Hierarchical models have several advantages :

- If a visual recognition task can be decomposed into low-complexity learning tasks through the layers, then each layer would need only a small set of training data [43].
- The lower levels of the hierarchy might represent a dictionary of features that can be shared across various classification tasks [22], thus increasing efficiency.

There are also some known limitations of hierarchical models:

- Hierarchical models assume that the computations at each successive stage being largely feed-forward [46], [14]. This is limited because back-projections are also likely to be a key part of the visual system.
- The anatomical hierarchy should be considered an idealization instead of a strict flowchart of visual information [27].

2.1.2 Recurrent Models

Increasingly, recent experimental and computational evidence has suggested alternatives to the hierarchical model, notably recurrent models [37].

Recurrent models changed the focus of attention from single cells to networks of cortical connections [37]. An illustration for recurrent models are shown in Figure 2.2.

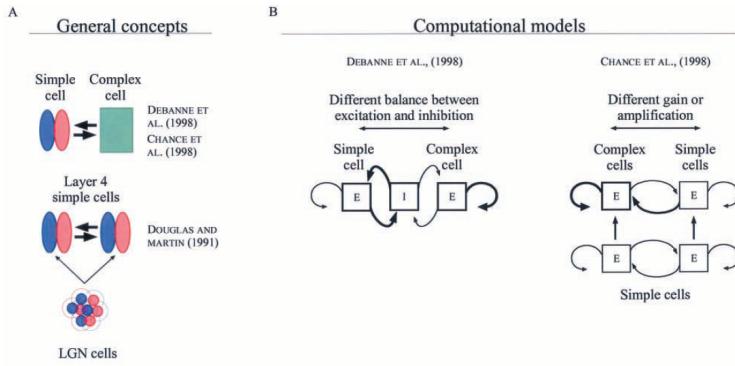


FIGURE 2.2: Illustration of the recurrent models [37].

In addition, more recently, recurrent models of the visual system has inspired a new topic of research that focus on applying the idea behind recurrence model to show and/or address limitations of computer vision models. Among other similar works ([40], [51], [33], [4]), two notable recent works are [45], which proposed that feedback mechanisms including attention and perceptual grouping are key to visual relational reasoning tasks and [33], which showed that recurrence is necessary to model the representational dynamics in the visual system.

Till today, there are still many debates over which models best capture the neural circuits in the visual system and whether and how computer vision models can incorporate certain features of biological vision. Our work builds on the above existing line of research and provides some original insights to open questions that are relevant to these debates.

2.2 Neural manifold theory

Throughout the historical progress in neuroscience, there have been numerous efforts to apply geometric and topological methods from related fields

in mathematics and computer science, including computational geometry, differential geometry, spectral graph theory, and algebraic topology, among many others. Recently, studying the structure of neural population response with high-dimensional neural spiking data has become a commonly used technique in studying the structure of brain cortices. The high-dimensional nature of neural spiking data have thus motivated various linear and non-linear dimensionality reduction methods.

In [8], neural manifold is defined as a set of geometric structures in neural population activity underlying different cognitive tasks.

The authors of [21] were the first to propose the concept of “neural modes,” which are specific patterns of correlated neural activity that span the neural manifolds. They thus proposed a generative model of individual neuronal activity based on the activation of neural modes. The parameters of such model can be obtained using dimensionality reduction methods. [58] used linear dimensionality reduction method of tensor CANDECOMP/-PARAFAC (CP) decomposition to discover low-dimensional neural dynamics by extracting three “tensor factors,” which are interconnected, low-dimensional descriptions of neural data.

3 Dimensionality Reduction

As explained in section 1.2, our basis for comparison is the neural manifold, which is a lower-dimensional representation of the original neural output. This requires the use of an appropriate dimensionality reduction method⁵. In this thesis, we focus on the linear dimensionality reduction method, tensor CANDECOMP/PARAFAC (CP) decomposition, a higher-order generalization of Principal Component Analysis (PCA)⁶. This higher-order generalization is necessary because the neural output has three dimensions, namely neurons, stimuli, and time steps (which will be further explained in Chapter 4).

3.1 Tensor CP decomposition

Definition 4 (Tensor). *An N-way tensor is defined as an element of the tensor product of N vector spaces. A 1-way tensor is a vector. A 2-way tensor is a matrix.*

Definition 5 (Tensor norm). *The norm of tensor $X \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ is*

$$\|X\| = \sqrt{\sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \dots \sum_{i_N=1}^{I_N} x_{i_1 i_2 \dots i_N}^2}. \quad (3.1)$$

Definition 6 (Rank-one tensor). *An N-way tensor $X \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ is rank-one if it can be written as the outer product of N vectors, that is,*

$$X = \mathbf{a}^{(1)} \circ \mathbf{a}^{(2)} \circ \dots \circ \mathbf{a}^{(N)} \quad (3.2)$$

Analogous to PCA, tensor CP decomposition aims to approximate the original tensor by a sum of rank-one tensors, each of which can then be

⁵Linear dimensionality reduction methods work well when the data lie near a linear subspace of high-dimensional space. They suffice for the purpose of the project thus far, but when the data lie near a nonlinear manifold, a nonlinear dimensionality reduction method might be more suitable [18]. Therefore, we also introduce the nonlinear dimensionality reduction technique of diffusion map in Appendix B.

⁶In this chapter we assume familiarity with PCA, but interested reader could refer to Appendix C for notes on PCA.

written as the outer product of N vectors. Given a 3-way tensor $X \in \mathbb{R}^{I \times J \times K}$, the tensor CP decomposition model is formulated as

$$X \approx [[\mathbf{A}, \mathbf{B}, \mathbf{C}]] = \sum_{r=1}^R \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r, \quad (3.3)$$

where R is the rank of the tensor, i.e., the minimal number of rank-1 tensors necessary to represent the tensor. $\mathbf{a}_r \in \mathbb{R}^I, \mathbf{b}_r \in \mathbb{R}^J, \mathbf{c}_r \in \mathbb{R}^K$ for $r = 1, \dots, R$. Determining the rank R of a specific given tensor is difficult; in fact, the problem is NP-hard. Thus, typically the goal is to find the low-rank approximation of the true rank, i.e., R as an approximation of the true rank M and $R < M$.

With tensor CP decomposition, we obtain the tensor factors which are analogous to the principle components in PCA. These neural factors can then be used as lower-dimensional representation of the tensor data. This can be illustrated with the diagram below. These neural factors can be used as lower-dimensional representation of the tensor data.

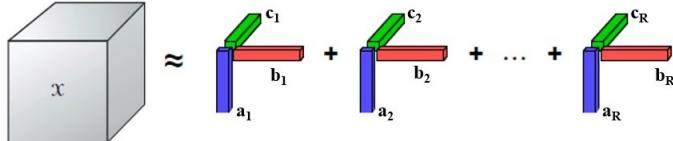


FIGURE 3.1: Illustration for tensor CP decomposition.
(Adapted from [58])

In selecting the optimization algorithm, we experimented with different optimization methods using the the generalized CP decomposition framework described in [29]. In the end, gradient-based direct optimization approach (OPT) [1] was shown to give higher accuracy when the number of factors chosen is larger than the true rank (“overfactoring”). This could help inform us the number of factors to choose.

Additionally, when the data has a non-negative constraint, Non-negative

Tensor Factorization (NTF) is used. In NTF, by adding the non-negative constraint, the original optimization problem becomes:

$$\begin{aligned} & \underset{\mathbf{A}, \mathbf{B}, \mathbf{C}}{\text{minimize}} && \|\mathcal{X} - [\mathbf{A}, \mathbf{B}, \mathbf{C}] \|_F^2 \\ & \text{subject to} && \mathbf{A}, \mathbf{B}, \mathbf{C} \geq 0. \end{aligned} \tag{3.4}$$

Henceforth we will refer to the chosen linear dimensionality reduction method (tensor CP decomposition using direct optimization with non-negative constraint) as Non-negative Tensor Factorization (NTF).

4 Biological Neural Networks

In this chapter, we describe our experiments to test and extend the findings in [16] on a different data set collected from the retina.

4.1 Neural tensor for retina

The neural spiking data [16] were collected from lab experiments with the setup shown in the figure below. Six different types of flow stimuli developed in [17] were flashed in front of the mouse. The electrodes recorded the neural output from the mouse's retina while it viewed each flow stimuli moving in eight directions respectively. The neural recordings were encoded in peristimulus (PSTH) diagrams, each of which showed the firing rate of one neuron over time for the eight directions respectively. In the PSTH diagram the brighter pixels indicate higher firing rates.

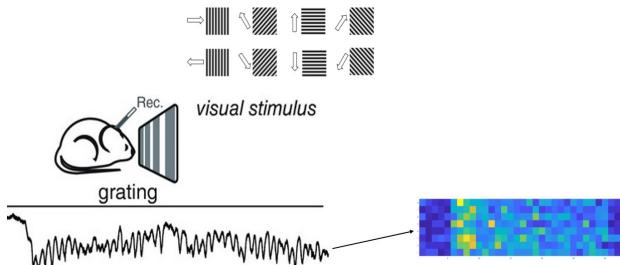


FIGURE 4.1: Visualising neural data from lab experiments.

The neural spiking data is of three dimensions: the first dimension represents the 698 neurons; the second dimension represents the six different types of stimuli; the third dimension represents the vectorized PSTH diagrams, each of which has $8 \times 33 = 264$ pixels. This gives us a 3-way tensor:

Definition 7 (Neural tensors). *Suppose \mathcal{S} is a set of flow stimuli $\mathcal{S} = \{s_1, s_2, \dots, s_K\}$, each moving over T time steps. The neural population response of a set of N neurons to some stimulus s_i over time is $\mathcal{N} = \{\vec{n}_1, \vec{n}_2, \dots, \vec{n}_N\}$, where $\vec{n}_i \in \mathbb{R}^T$.*

Each neural tensor encodes the neural population response of N neurons to K flow stimuli over T time steps and is thus a 3-way N -by- K -by- T tensor.

4.2 Neural factors for retina

We implement the Non-negative Tensor Factorization (NTF) using the Tensor Toolbox for MATLAB [6]. Our algorithm is as follows:

Algorithm 4.2: Algorithm for NTF on retina tensor.

Data: Neural tensor for retina.
Result: R tensor factors for retina.

```

1 import data as X, X = tensor(X)
2 chosen rank, R = 35
3 n_repetitions = 10, initialize errors with zeros
4 for r from 1 to R do
5   for rep from 1 to n_repetitions do
6     NTF: M = cp_opt(X, r, 'init', 'rand', 'lower', 0)
7     error = norm(X - tensor(M)) / norm(X);
8     add error to the array of errors
9 F_neuron = M.u{1}, F_stimuli = M.u{2}, F_time = M.u{3}
10 save the factors

```

We plot the reconstruction error curves to compare the performance of tensor CP decomposition when different optimization methods are used. ALS with random initialization and regular direct optimization perform better than non-negative direct optimization. However, since the neural output from biological neural networks is always non-negative, in order to ensure the interpretability of the resulting tensor factors in the context of our application, we choose to use non-negative direct optimization despite its relatively worse performance (which is not a crucial concern in our application since the neural factors obtained in this process are still good enough approximation of the original neural data).

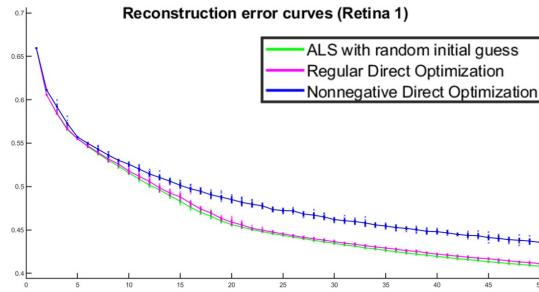


FIGURE 4.2: Reconstruction errors of different optimization methods.

We can visualize the resulting tensor factors:

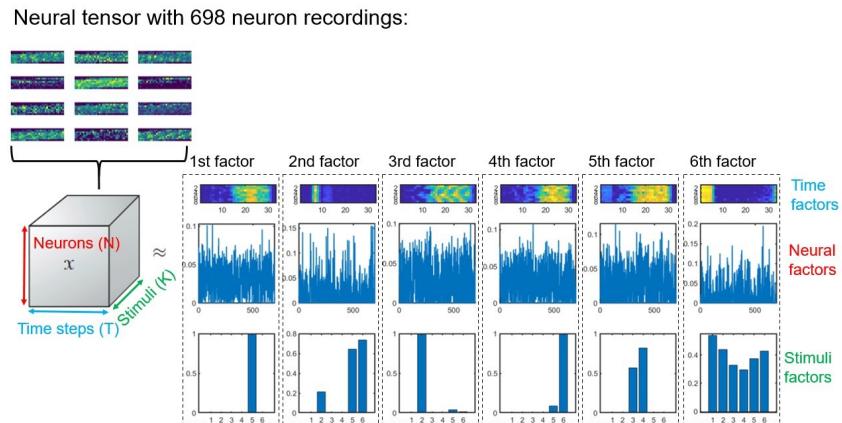


FIGURE 4.3: First six tensor factors for retina.

4.3 Neural manifold for retina

The tensor factors form what we call the “neural matrix” of dimension (N, R) , where N is the number of neurons and R is the chosen rank. Since the tensor factors provide an approximation of the original neural spiking data, we use PCA to project neural matrix onto a linear subspace with the same dimension, i.e., the number of components is R . We then use k-means clustering to label the clusters by distances in the lower-dimensional linear subspace. By choosing the optimal number of clusters to be six based on the experiments, we obtain the neural manifold for retina. In the following plot, each

point represents a neuron. From the visualization below, the neural manifold for retina is discontinuous and the PSTH diagrams within each cluster are similar. This implies that neurons form disconnected clusters grouped by their firing patterns, implying that neurons have distinct stimuli preferences. By using the connections between neural manifolds and neural circuits explained in , we can already infer that the neural networks in retina are characterized by isolated neural circuits. We extend this result further by quantifying the degree of continuity in the next section.

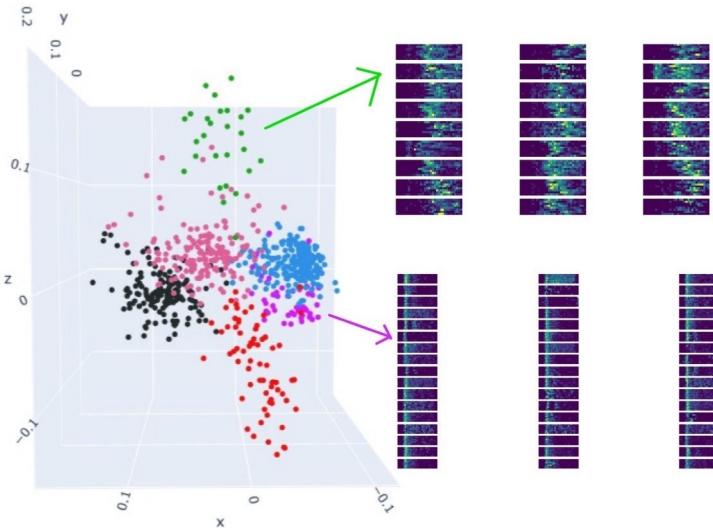


FIGURE 4.4: Neural manifold for retina is discontinuous.

4.3.1 Quantifying the continuity of the manifold

Following the method proposed in [16], we compute the mean flow ratio (MFR) as a quantifiable measure to compare the continuity of the respective neural manifolds: the smaller the MFR, the more discontinuous the neural manifold is, and vice versa. The algorithm is outlined on the next page. The MFR of the retina neural manifold computed from our algorithm is $\phi_G = 0.56$. Now we compare the results with V1 using results from [16]: the neural manifold for V1 is continuous in the visualizatoin below and the MFR of V1 is $\phi_G = 0.91$ as concluded in . The difference in the MFRs suggests that the

neural manifold for V1 is much more continuous than retina. Thus, we can conclude that the neural circuit in V1 has much higher connectivity than retina. The results in this chapter will be used again in the next chapter to compare with results from the artificial neural networks.

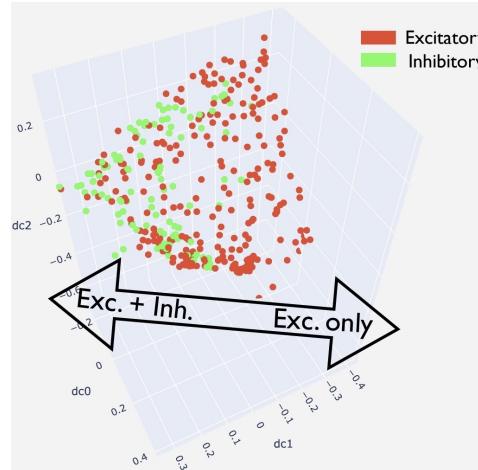


FIGURE 4.5: Neural manifold for V1 is continuous. The neurons are colored by excitatory and inhibitory functions and the two clusters overlap each other.

Algorithm 4.3: Algorithm for computing the mean flow ratio.

Data: points on the neural manifold, $\mathbf{x}_1, \dots, \mathbf{x}_N \in \mathcal{M}$.
Result: mean flow ratio for the neural manifold, ϕ_G .

- 1 kernel bandwidth, $\varepsilon := \frac{1}{N} \sum_{i=1}^N \min_{j: \mathbf{x}_j \neq \mathbf{x}_i} \|\mathbf{x}_i - \mathbf{x}_j\|^2$
- 2 Gaussian kernel^a, $k(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/\varepsilon)$
- 3 weight matrix $W, W_{i,j} := k_\varepsilon(\mathbf{x}_i, \mathbf{x}_j)$
- 4 degree of node $\mathbf{x}_i, d(\mathbf{x}_i) := \sum_{j=1}^N W_{i,j}$
- 5 anisotropic Gaussian kernel, $\tilde{k}_\varepsilon(\mathbf{x}_i, \mathbf{x}_j) := \frac{k_\varepsilon(\mathbf{x}_i, \mathbf{x}_j)}{d(\mathbf{x}_i)d(\mathbf{x}_j)}$
- 6 anisotropic weight matrix, $\tilde{W}, \tilde{W}_{i,j} := \tilde{k}_\varepsilon(\mathbf{x}_i, \mathbf{x}_j)$
- 7 build graph $G = (V, E)$, weight of the edge $(\mathbf{x}_i, \mathbf{x}_j)$ is $\tilde{W}_{i,j}$
- 8 edge resistance, $R_{i,j} = 1/\tilde{W}_{i,j}$
- 9 compute effective resistance $R_{i,j}^{\text{eff}}$, using [50]
- 10 effective conductance, $C_{i,j}^{\text{eff}} = 1/R_{i,j}^{\text{eff}}$
- 11 build graph G_C , weight of the edge $(\mathbf{x}_i, \mathbf{x}_j)$ is $C_{i,j}^{\text{eff}}$
- 12 **for** i from 1 to N **do**
 - 13 **for** j from 1 to $N, j \neq i$ **do**
 - 14 SumAll += maxflow(i, j)
 - 15 **for** $k: k$ is a neighbor of i **do**
 - 16 SumNeighbors += maxflow(i, k)
 - 17 flow ratio of node $\mathbf{x}_i, \rho_i = \text{SumAll}/\text{SumNeighbors}$
- 18 mean flow ratio of graph $G, \phi_G = \sum_{i=1}^N \rho_i$.

^aIn fact, we can choose any symmetric, positive semi-definite similarity kernel. But Gaussian kernel gives a physically intuitive construction as it effectively considers a ball of radius ε around each data point.

5 Artificial Neural Networks

As we have established in the introduction, the key basis of comparison between artificial and biological neural networks is the continuity of the neural manifold. Qualitatively, we can make this comparison by visualizing both the tensor factors and the neural manifolds. Quantitatively, we can compare the mean flow ratio which provides a measure for how continuous the neural manifolds are. This chapter outlines our algorithms, key experiments, and findings for candidate models proposed in 1.2, which allow for a rigorous comparative analysis - these constitute our main contribution to the research literature.

5.1 CNN

Our first candidate model is VGG16, one of the most successful CNN models in computer vision tasks such as image classification. Its architecture is shown in the figure below.

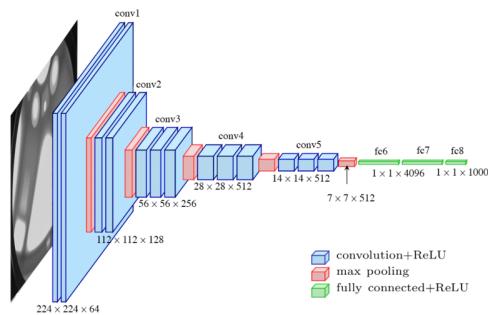


FIGURE 5.1: Visualizing the structure of the VGG16 model.

5.1.1 “Visual stimuli” in CNN

Analogous to the flow stimuli in lab experiments, the “visual stimuli” for CNN are natural images of different objects selected from the benchmark

dataset ImageNet [12]. To keep the size of the artificial neural tensor manageable, we randomly selected 20 images from four classes as the visual stimuli. In order to simulate the movement of flow stimuli over time, we create multiple shifts of the original image in vertical and horizontal direction, with the shift step equal to the size of the filter.

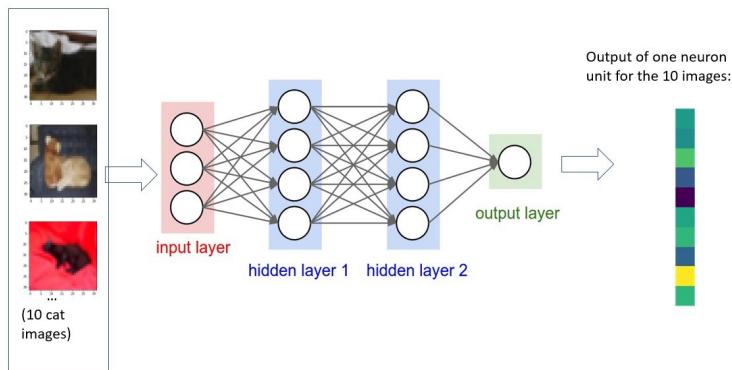


FIGURE 5.2: Computing output of one neuron unit in the CNNs given 10 input cat images.

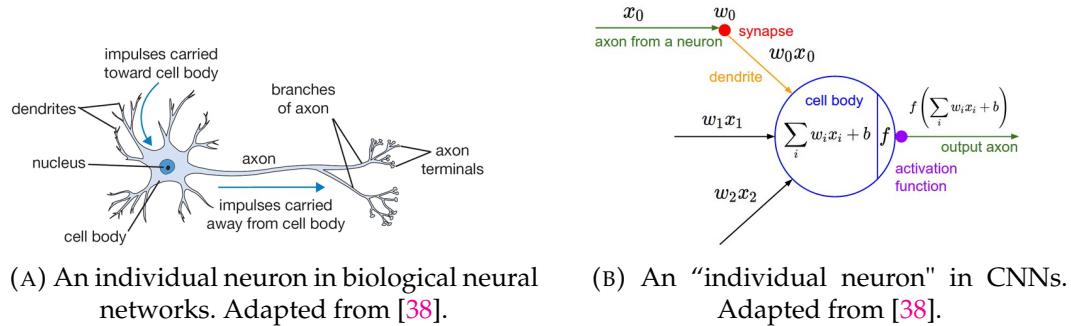
Remark 8. At first it might seem that using different stimuli for biological and artificial neural networks will lead to an unfair comparison. However, this approach is appropriate for the following reason: The mouse visual system was trained on the flow stimuli is shown to resemble the naturalistic visual information that the mouse's vision has adapted to⁷. The visual stimuli for VGG16 model came from the same data set that it is trained on (ImageNet). Thus, in both cases, the visual stimuli are similar to what the respective network was trained to recognize.

5.1.2 “Individual neuron” in CNN

In the lab experiments for biological neural networks, moving visual stimuli trigger spikes in the activation potential in the neurons in the retina. The individual neuron response is the firing rate of the neuron which is dependent

⁷Based on the paper that proposed flow stimuli [17], flow stimuli are more like what the mouse would encounter in the natural world than are sine-wave gratings but is more tractable for analysis than are natural images.

on the electrostatics processes that take place in the synapse. The biological neuron is coarsely modeled by the neuron unit in the CNNs. At the “dendrite,” the input is taken from the “axons” of the connected neurons.⁸ The weight of each neuron models the synaptic process in biological neuron. At the “cell body,” all the inputs are multiplied with the weights and summed. We then add the bias term to the sum and apply the nonlinear activation to obtain the final output from the individual neuron unit at the “axon.” The parallel between biological and artificial neuron is shown in the figures below.



(A) An individual neuron in biological neural networks. Adapted from [38].

(B) An “individual neuron” in CNNs. Adapted from [38].

5.1.3 “Receptive field” of an artificial neuron

Each layer in the CNN has several different feature maps. Each feature map corresponds to a filter of a specific size and specific weights. The filter is essentially a matrix that, when taking inner product with the input image, give an output that highlights some specific features in the input image. To illustrate this, we can visualize the 64 feature map in the first convolutional layer of the VGG16 model, given the input image.

⁸Note, however, that in our experiments, instead of taking the input from previously connected neuron unit, each neuron takes the image as the input.



(A) Input image. Adapted from [30].



(B) Visualizing the 64 feature maps in the first convolutional layer of the VGG16 model. Adapted from [30].

The neurons within each feature map share the same weights and each attends to a specific subregion of the image which is referred to as their receptive field. This term originated from biological vision defined as a restricted region of visual space where a stimulus could illicit responses in a retinal ganglion cell. The size of the receptive field of a neuron is the same as the filter size corresponding to the feature map in which the neuron is located.

5.1.4 “Neuron population response” in CNN

Analogous to biological neural networks, we computed the neural output given the input image. Unlike biological neurons, most of the artificial neurons will have output of small values because most pre-trained weights decay to zero. In other words, the “firing” of neurons in CNN will be sparse. We select the feature maps that contain the neurons with highest average firing rate.

Algorithm 5.1: Algorithm for CNN artificial neural tensor.

```

1 Function apply_all_shifts(image, shift_step):
2   add the original image to image_all_shifts
3   for i from 0 to # vertical shifts do
4     shift the image vertically by shift_step
5     add the shifted image to image_all_shifts
6     for j from 0 to # horizontal shifts do
7       shift the image horizontally by shift_step
8       add the shifted image to image_all_shifts

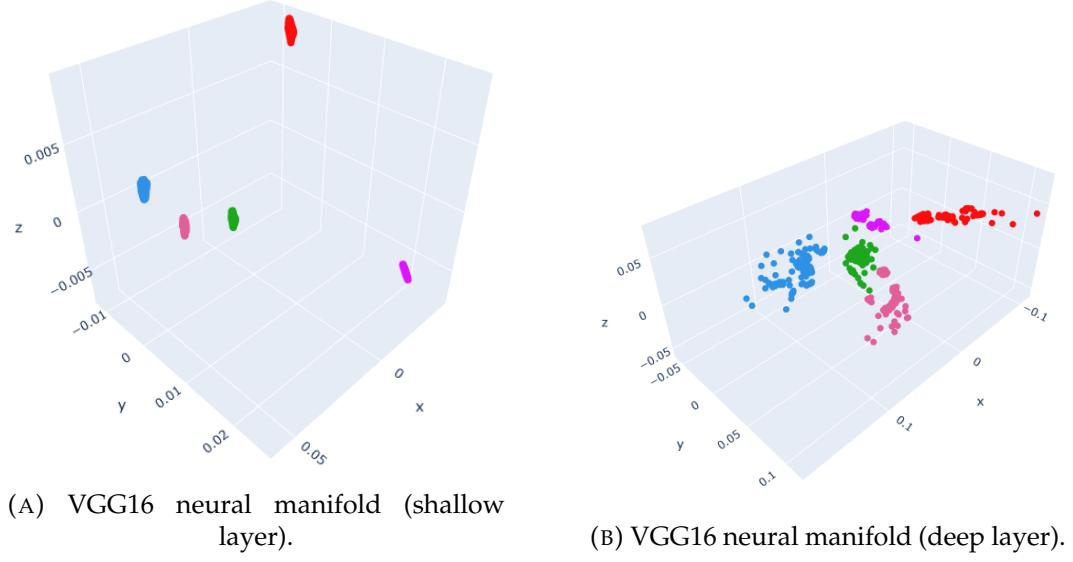
9 Function compute_neuron_output(model, image_all_shifts):
10  for layer in all convolutional layers do
11    pass the image through the model
12    take the neuron output from the specified layer after ReLU
13    (optional) remove the neurons at the edges
14    compute the average neuron output in each feature map
15    select the feature maps with highest average
16    normalize the neuron outputs in each feature map

17 Function show_stimuli(model, all_images):
18  for image in images_selected_classes do
19    call apply_all_shifts
20    call compute_neuron_output
21    take average over outputs of all shifts of current image
22    add to neuron_output_all_images

```

5.1.5 Neural manifold for CNN

In the plot below, each point in the data cloud represents a neuron. As shown by the disconnected clusters of neurons, the neural manifold for CNN is clearly discontinuous. When labeling the neurons according to their feature maps, we realize that these disconnected clusters are in fact organized by feature maps. This result aligns with the theoretical explanation that the neurons within the same feature map share the same filter weights. From shallow layer to deep layer, it is observed that although the clusters become more diffused, they were still form separate clusters with little overlap.



In addition, by modifying line 24 in Algorithm 5.1 to appending the output for each shift instead of taking the average over all shifts, we obtain the neural manifold that preserves the spatial ordering of the neurons in the same feature map. The center and right plots below show the neurons within one cluster colored by their vertical and horizontal position. Since the neuron's coordinate on the neural manifold and the color aligns perfectly, this is good evidence that within a cluster, each neuron's response corresponds to its spatial position given by the subregion it attends to.

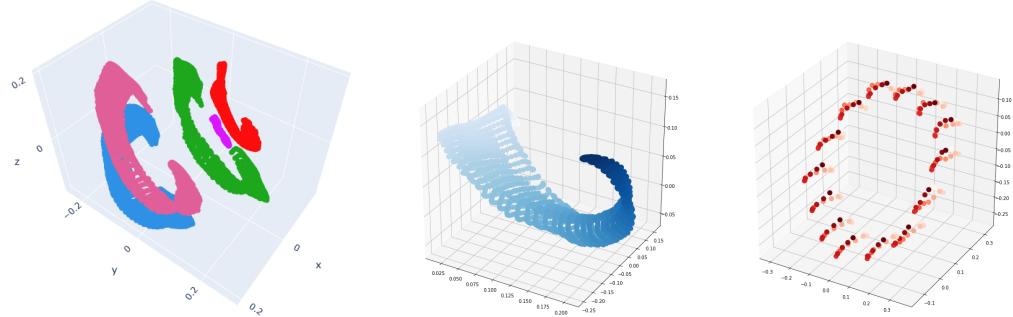


FIGURE 5.6: Organization within the CNN neural manifold corresponds to neurons' spatial positions.

5.2 ViT

A recent model ViT has been shown to outperform CNN in image recognition tasks while requiring fewer computational resources [41]. Transformer [57] is originally designed for sequence-to-sequence tasks primarily used in Natural Language Processing (NLP). In ViT, the given input image is divided into a sequence of 16-by-16 image patches, which allows the image recognition problem to be solved with Transformer. The distinguishing feature of the Transformer model is entirely built on the self-attention mechanisms without using sequence-aligned recurrent architecture.

5.2.1 “Individual neuron” in ViTs

The challenge in building artificial neural tensor for ViT lies in setting up a reasonable analogy to CNN. To do this, we need to scrutinize the structure of ViT.

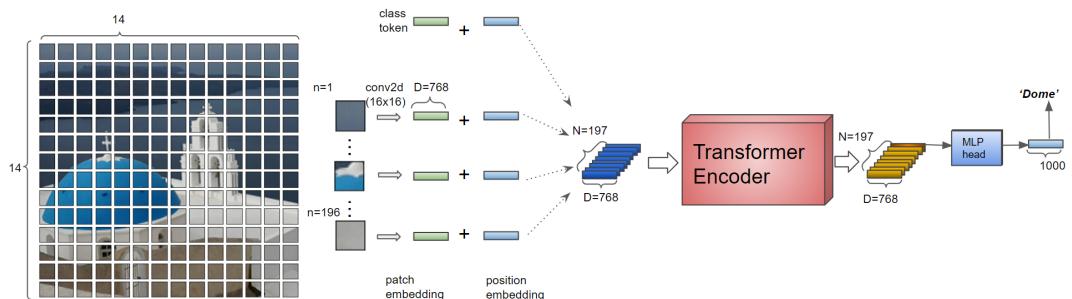


FIGURE 5.7: ViT input (image credit: Hiroto Honda).

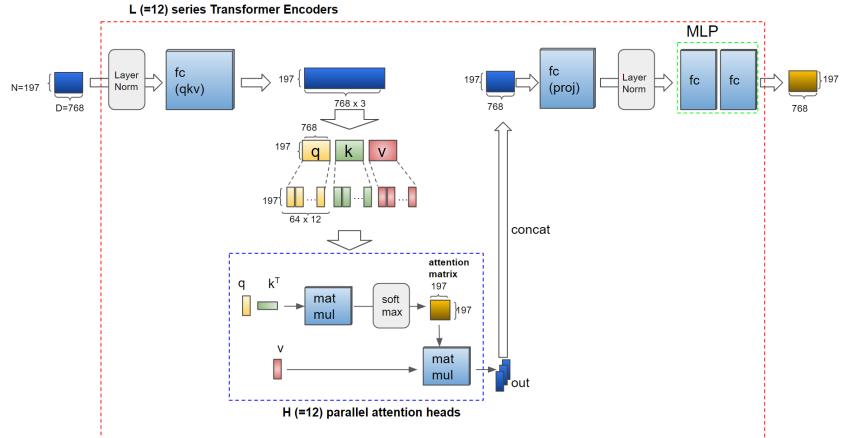
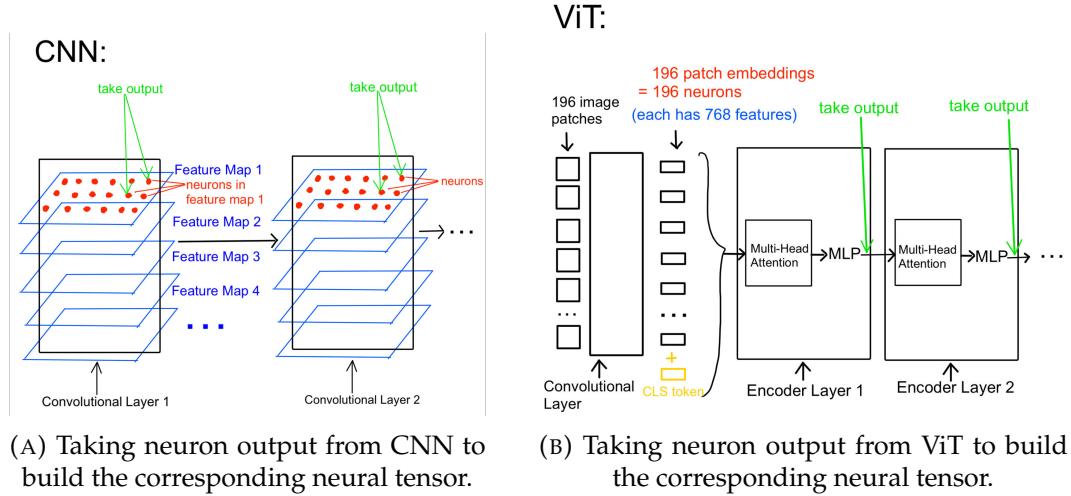


FIGURE 5.8: ViT encoder (image credit: Hiroto Honda).

The following diagram compares the structures of ViT and CNN, which suggests the modification we need to make for ViT based on the previous algorithm for CNN.



Using Algorithm 5.2, we generate the neural manifold for ViT by taking the output from each encoder layer. The manifold structure for ViT is similar to that of CNN: the manifold is discontinuous and shows discrete clusters organized based on the features obtained from the convolutional

layer before the encoder layers.

Algorithm 5.2: Algorithm for ViT artificial neural tensor.

```

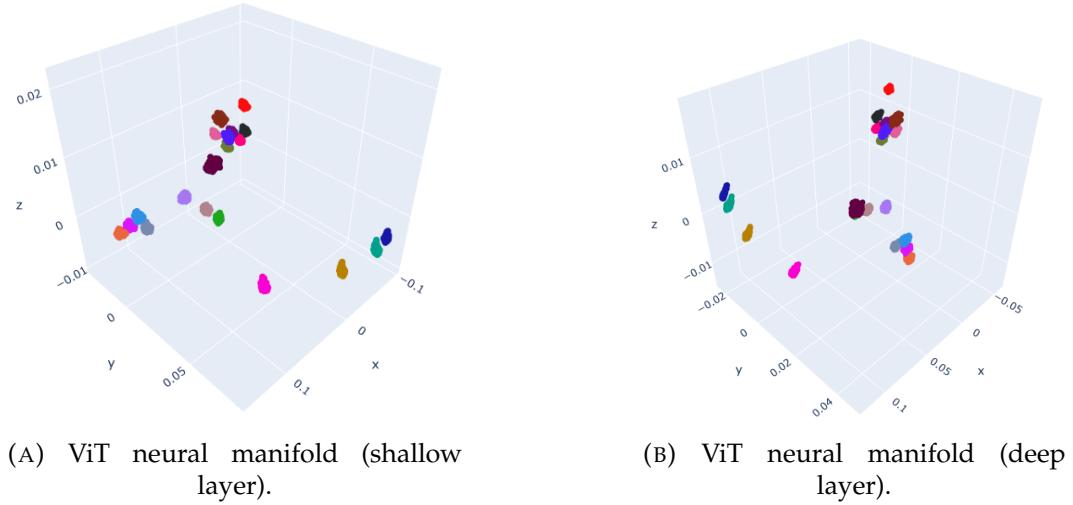
1 Function compute_neuron_output(model, image_all_shifts):
2   for layer in all encoder layers do
3     pass the image through the model
4     take the neuron output from the specified layer
5     apply ReLU activation on the neuron output
6     compute the average of neuron outputs in each feature
7     select the feature maps with highest average
8     normalize the neuron outputs in each feature map
  
```

Remark 9. On line 5, we need to impose ReLU⁹ activation to make the neuron output non-negative. The non-negative constraint is desired because the tensor factors from NTF are more interpretable. The pre-trained ViT uses GELU¹⁰ as the activation function. This results in negative values and would prevent us from using non-negative tensor decomposition. Fortunately, the negative portion of GELU is negligible given the purpose of our investigation, which justifies this step.

An interesting observation is that from shallow layer to deep layer, the sizes of clusters mostly remain the same except for their relative positions, which is slightly different from what is observed for CNN. We provide one possible explanation for this observation. The key mechanism in ViT is self-attention while the key mechanism in CNN is convolution. A crucial difference between the two mechanisms is the receptive field size. Based on findings from [44], [11], convolution receptive fields are highly local and grow gradually whereas self-attention has global receptive field. As a result, ViT has more uniform representations and greater similarity between lower and deeper layers.

⁹ReLU: Rectified Linear Unit [2].

¹⁰GELU: Gaussian Error Linear Unit [28].



We also visualize the neural manifold for ViT that preserves the spatial information, which shows that similar to CNN, neuron responses in ViT also have similar spatial organization within the neural manifold. This similarity is likely because ViT encoder layer takes the features from the convolutional layer as its input.

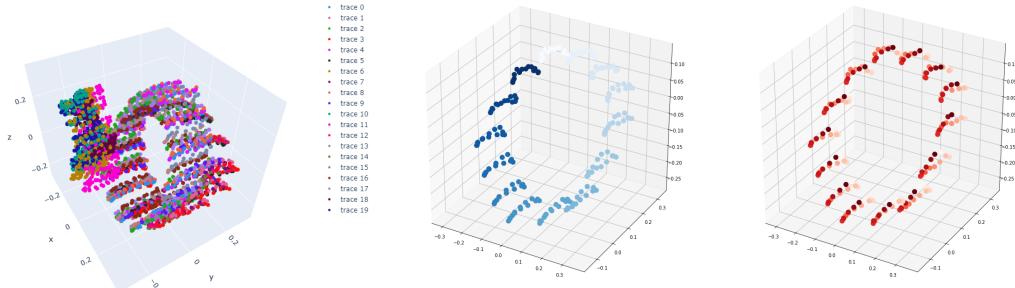
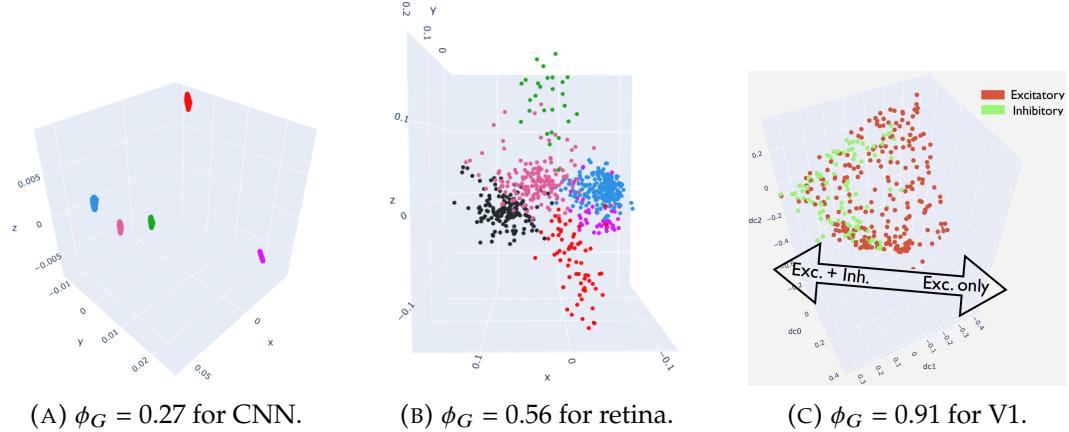


FIGURE 5.11: Organization within the ViT neural manifold corresponds to neurons' spatial positions.

5.2.2 Quantifying the continuity of the manifold

Following the method proposed in [16], we further compute the MFR as we have done for biological neural networks. Since it is clear from the visualization that the neural manifolds CNN and ViT are similarly discontinuous, it suffices to compute the MFR for one of them. We compare the respective neural manifolds and their respective MFR for CNN, retina, and V1:



The quantitative comparison aligns with our qualitative comparison: from the visualization, we have observed that V1 has a much more continuous manifold than CNN and the retina; similarly, the MFR of V1 is notably higher than CNN and the retina. MFR additionally reveals (which is not entirely obvious from the visualizations) that there seems to be a considerable gap between between CNN and retina as well.

5.3 Adding recurrence

Since neither the convolution in CNN nor the attention mechanism in ViT yields continuous neural manifold like V1, this leads us to investigate other computational mechanisms that could result in a more continuous neural manifold. Based on our literature review in 2.1.2, we hypothesize that recurrent models might be promising. For this reason, we apply the same method to investigate computer vision models with recurrent structures. One challenge is that recurrent neural networks (RNNs) themselves are not commonly used for image recognition tasks. Fortunately, there are computer vision models that combine convolution and recurrent units. For this reason, we choose two representative models: first, convolutional RNN (CRNN) [47] that adds recurrent layers after the convolutional layers and second,

γ -net [36] that adds recurrent connections directly on the convolutional layers. Algorithm 5.1 for CNN applies in this case as well. We include here our preliminary findings for neural manifold obtained from CRNN [47].

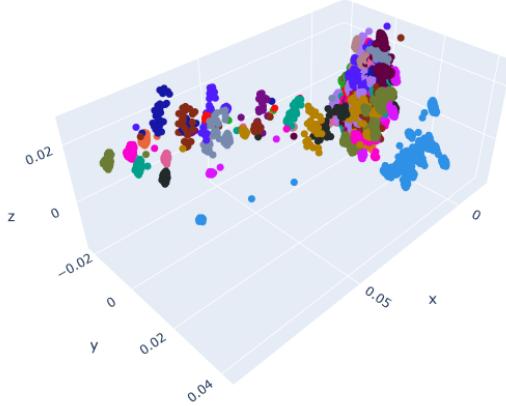


FIGURE 5.13: Neural manifold for CRNN (shallow layer).

As observed from the figure above, the clusters become less separated and significantly overlap with each other, which is in contrast to the neural manifolds for CNN and ViT . This suggests that by adding the recurrently layers and jointly training the unified network we can obtain a more continuous manifold. A possible reason for this observation is that due to the structure of this model, the convolutional layers are affected by the recurrent units in the RNN layers in that the RNN layers can back-propagate error differentials to its inputs from the convolutional layers.

Another intriguing result is that when visualizing the neural factors (obtained by applying NTF on respective neural tensors), the neural factors for CRNN seem to have very distinct properties from those for CNN and ViT. The neural factors visualize the independent neural firing patterns that can approximately summarize all the neural firing patterns in the original neural tensor. In the following figure, we see that the for CNN and ViT there is mostly just one bright region in each factor, suggesting that the neurons mostly respond to a single targeted feature/region in the input image. This

likely implies the neurons have strong stimuli preferences and they participate in isolated neural circuits (thus yielding a continuous manifold). However, neural factors for CRNN indicate some sequential patterns with many bright regions in each factor, suggesting that the neurons might respond to multiple stimuli. This implies that the underlying neural circuit is more interconnected, which is why we have observed a more continuous manifold.

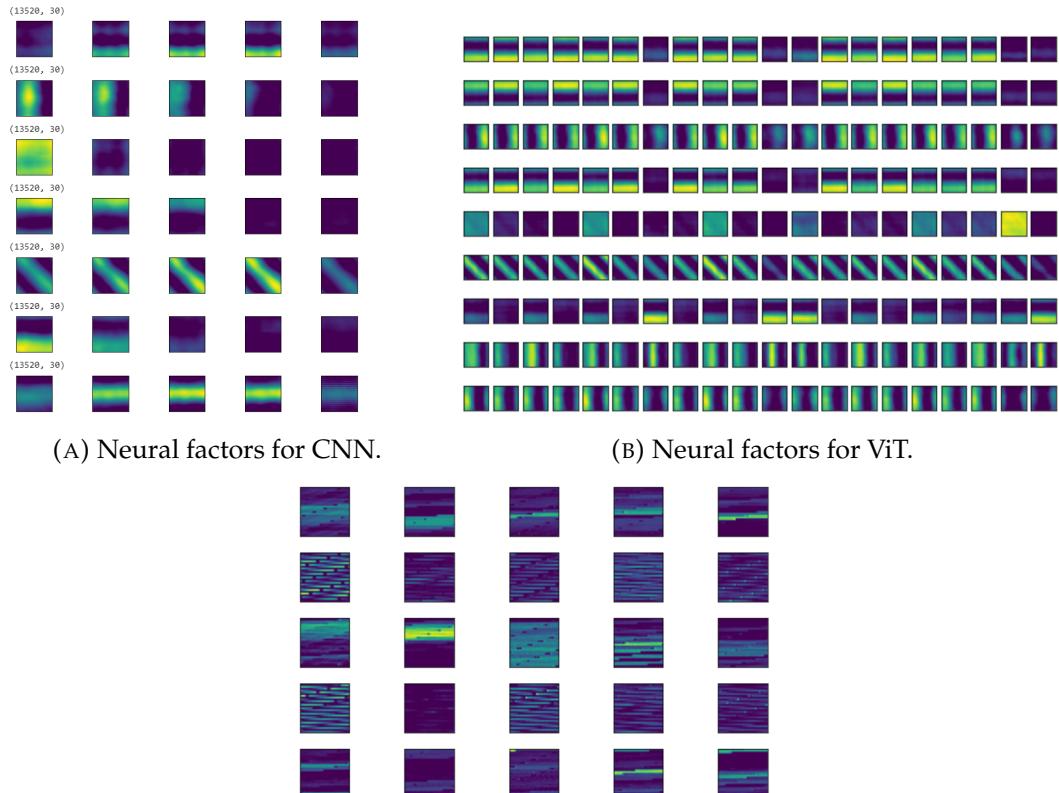


FIGURE 5.15: Neural factors for CRNN are distinct from CNN and ViT.

6 Conclusion

6.1 Summary and discussion of results

Our results were the first to show that CNN and ViT form discontinuous neural manifold whereas adding recurrent units to the network forms a more continuous manifold. From this we can infer that CNN and ViT form isolated neural circuits and recurrent connections increase the connectivity of the underlying neural circuits. We reformulate our results in terms of conjectures that address the open questions raised in section 1.1:

- *How do computer vision models (VGG16 [48], Vision Transformer [15], and Convolutional Recurrent Neural Network [47]) compare to biological vision, specifically in the regions retina and V1?*

Conjecture 1. The underlying neural circuits in feed-forward networks, including CNN and ViT, are closer to that in retina than that in V1. In other words, they are decent models for retina but are severely limited as models for V1.

- *What specific feature(s) in computer vision models lead to the differences in the neural circuits?*

Conjecture 2. Incorporating recurrent mechanisms in computer vision models seems to be crucial in increasing the connectivity of the underlying neural circuits.

The implication of **Conjecture 1** is that, contrary to popular belief, current computer vision models turn out to be poor approximations of visual cortical circuits (although they are indeed inspired by early studies of the

visual cortex and perform well on image recognition tasks). These feed-forward networks are good enough as models for the retina, but fail to capture the characteristics of the functional neural circuits in the visual cortex. **Conjecture 2** suggests that recurrence seems to be the missing key to a more accurate model of the visual cortex. In the next section, we will propose future research in three directions:

6.2 Future directions

6.2.1 Understanding the role of recurrence in vision

Recurrence is a characteristic feature in not only biological vision but also most of our important brain functions. However, recurrence is largely missing from most of the popular computer vision models today. Our results have suggested that recurrence is key to increasing the connectivity of the underlying functional neural circuits in the neural networks (both artificial and biological). Since richer functional connectivity is a desirable feature for neural networks when solving visual tasks that involve more complex reasoning, it could potentially have significant implication in advocating the merit of incorporating recurrent mechanisms into computer vision models. Thus, we propose a more extensive analysis of the role of recurrence in both biological vision and computer vision.

6.2.2 Experimenting with recurrent models

The experiments in Chapter 5 have solved some open questions that we set out to tackle, but they also leave us with many more intriguing observations that we have not understood fully. Building on the results we have obtained during the capstone, we will continue and focus on studying how various

recurrent structures affect the the neural manifold. We are currently in the process of obtaining an additional pre-trained network model [36], which includes recurrent connections directly to the convolutional layers. Our current investigation on the neural manifold of artificial neural networks with recurrent structures is still limited: we have only studied CRNN which add recurrently layers after the convolutional layers. This might not be sufficient and we believe that the model in [36] that adds recurrence directly to the convolutional layers would be a promising place to continue. It would be of interest to find out whether (and if so, why) adding recurrence in different ways may result in manifolds with different structure. We can apply the mean flow algorithm to precisely quantify the extent to which each kind of recurrent structure change the continuity of the neural manifold.

6.2.3 Applying nonlinear dimensionality reduction

Although the linear dimensionality reduction method (tensor CP decomposition) has been working well so far, this could be a potential limitation of our current study, especially since that both network science and neural dynamics are typical contexts where nonlinear systems arise. It might be interesting to apply the nonlinear dimensionality reduction method (diffusion map) to compare the results.

Bibliography (Main)

- [1] Evrin Acar, Daniel M. Dunlavy, and Tamara G. Kolda. "A scalable optimization approach for fitting canonical tensor decompositions". In: *Journal of Chemometrics* 25.2 (2011), pp. 67–86. DOI: <https://doi.org/10.1002/cem.1335>.
- [6] Tamara G. Kolda Brett W. Bader et al. "Tensor Toolbox for MATLAB". Version 3.2.1. In: (Apr. 5, 2021). URL: <https://www.tensortoolbox.org/>.
- [8] SueYeon Chung and L F Abbott. "Neural population geometry: An approach for understanding biological and artificial neural networks". In: *Current Opinion in Neurobiology* (2021), p. 12.
- [15] Alexey Dosovitskiy et al. "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale". en. In: *arXiv:2010.11929 [cs]* (June 2021). arXiv: 2010.11929. URL: <http://arxiv.org/abs/2010.11929> (visited on 01/01/2022).
- [16] Luciano Dyballa. "The Manifold of Neural Responses Informs Physiological Circuits in the Visual System". English. ISBN: 9798534651942 Publication Title: ProQuest Dissertations and Theses 28321876. Ph.D. New Haven: Yale University, 2021.
- [17] Luciano Dyballa et al. "Flow stimuli reveal ecologically appropriate responses in mouse visual cortex". In: *Proceedings of the National Academy of Sciences* 115.44 (2018), pp. 11304–11309. DOI: <10.1073/pnas.1811265115>. eprint: <https://www.pnas.org/doi/pdf/10.1073/pnas.1811265115>. URL: <https://www.pnas.org/doi/abs/10.1073/pnas.1811265115>.
- [20] Kunihiko Fukushima. "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position". In: *Biol. Cybernetics* 36.4 (1980), pp. 193–202. ISSN: 0340-1200, 1432-0770. DOI: <10.1007/BF00344251>. URL: <http://link.springer.com/10.1007/BF00344251> (visited on 10/25/2021).
- [31] D H Hubel and T N Wiesel. "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex". eng. In: *The Journal of physiology* 160.1 (Jan. 1962), pp. 106–154. ISSN: 0022-3751. DOI: <10.1113/jphysiol.1962.sp006837>. URL: <https://pubmed.ncbi.nlm.nih.gov/14449617>.
- [33] Tim C. Kietzmann et al. "Recurrence is required to capture the representational dynamics of the human visual system". en. In: *Proceedings of the National Academy of Sciences* 116.43 (Oct. 2019), pp. 21854–21863. ISSN: 0027-8424, 1091-6490. DOI: <10.1073/pnas.1905544116>. URL: <https://pnas.org/doi/full/10.1073/pnas.1905544116> (visited on 04/03/2022).
- [36] Drew Linsley et al. "Recurrent neural circuits for contour detection". In: (2020). DOI: <10.48550/ARXIV.2010.15314>. URL: <https://arxiv.org/abs/2010.15314>.
- [47] Baoguang Shi, Xiang Bai, and Cong Yao. "An End-to-End Trainable Neural Network for Image-based Sequence Recognition and Its Application to Scene Text Recognition". en. In: *arXiv:1507.05717 [cs]* (July 2015). arXiv: 1507.05717. URL: <http://arxiv.org/abs/1507.05717> (visited on 03/27/2022).
- [48] Karen Simonyan and Andrew Zisserman. "Very Deep Convolutional Networks for Large-Scale Image Recognition". en. In: *arXiv:1409.1556 [cs]* (Apr. 2015). arXiv: 1409.1556. URL: <http://arxiv.org/abs/1409.1556> (visited on 03/27/2022).
- [50] Daniel A. Spielman and Nikhil Srivastava. "Graph Sparsification by Effective Resistances". In: (2008). DOI: <10.48550/ARXIV.0803.0929>. URL: <https://arxiv.org/abs/0803.0929>.
- [58] Alex H. Williams et al. "Unsupervised Discovery of Demixed, Low-Dimensional Neural Dynamics across Multiple Timescales through Tensor Component Analysis". In: *Neuron* 98.6 (2018), 1099–1115.e8. ISSN: 08966273. DOI: <10.1016/j.neuron.2018.05.015>. (Visited on 06/10/2021).

Bibliography (Supplementary)

- [2] Abien Fred Agarap. "Deep Learning using Rectified Linear Units (ReLU)". In: *CoRR* abs/1803.08375 (2018). arXiv: 1803 . 08375. URL: <http://arxiv.org/abs/1803.08375>.
- [3] R Ben-Yishai, R L Bar-Or, and H Sompolinsky. "Theory of orientation tuning in visual cortex." In: *Proceedings of the National Academy of Sciences* 92.9 (Apr. 1995), p. 3844. DOI: 10.1073/pnas.92.9.3844. URL: <http://www.pnas.org/content/92/9/3844.abstract>.
- [4] Ruben S van Bergen and Nikolaus Kriegeskorte. "Going in circles is the way forward: the role of recurrence in visual inference". en. In: *Current Opinion in Neurobiology* 65 (Dec. 2020), pp. 176–193. ISSN: 09594388. DOI: 10.1016/j.conb.2020.11.009. URL: <https://linkinghub.elsevier.com/retrieve/pii/S0959438820301768> (visited on 04/03/2022).
- [5] Phyllis Blumenfeld, Toni Rogat, and Joseph Krajcik. "Motivation and Cognitive Engagement in Learning Environments." In: *The Cambridge Handbook of the Learning Sciences* (Jan. 2006).
- [7] Gunnar Carlsson. "Topology and data". en. In: *Bulletin of the American Mathematical Society* 46.2 (Jan. 2009), pp. 255–308. ISSN: 0273-0979. DOI: 10.1090/S0273-0979-09-01249-X. URL: <http://www.ams.org/journal-getitem?pii=S0273-0979-09-01249-X> (visited on 11/02/2021).
- [9] R. R. Coifman et al. "Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps". In: *Proceedings of the National Academy of Sciences* 102.21 (2005), pp. 7426–7431. ISSN: 0027-8424, 1091-6490. DOI: 10.1073/pnas.0500334102. URL: <http://www.pnas.org/cgi/doi/10.1073/pnas.0500334102> (visited on 08/05/2021).
- [10] Ronald R. Coifman and Stéphane Lafon. "Diffusion maps". In: *Applied and Computational Harmonic Analysis* 21.1 (2006), pp. 5–30. ISSN: 10635203. DOI: 10.1016/j.acha.2006.04.006. URL: <https://linkinghub.elsevier.com/retrieve/pii/S1063520306000546> (visited on 08/05/2021).
- [11] Zihang Dai et al. *CoAtNet: Marrying Convolution and Attention for All Data Sizes*. 2021. DOI: 10.48550/ARXIV.2106.04803. URL: <https://arxiv.org/abs/2106.04803>.
- [12] Jia Deng et al. "Imagenet: A large-scale hierarchical image database". In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee. 2009, pp. 248–255.
- [13] James J. DiCarlo, Davide Zoccolan, and Nicole C. Rust. "How Does the Brain Solve Visual Object Recognition?" en. In: *Neuron* 73.3 (Feb. 2012), pp. 415–434. ISSN: 08966273. DOI: 10.1016/j.neuron.2012.01.010. URL: <https://linkinghub.elsevier.com/retrieve/pii/S089662731200092X> (visited on 03/28/2022).
- [14] James J DiCarlo, Davide Zoccolan, and Nicole C Rust. "How does the brain solve visual object recognition?" In: *Neuron* 73.3 (2012), pp. 415–434.
- [18] Charles Fefferman, Sanjoy Mitter, and Hariharan Narayanan. "Testing the manifold hypothesis". In: *J. Amer. Math. Soc.* 29.4 (2016), pp. 983–1049. ISSN: 0894-0347, 1088-6834. DOI: 10.1090/jams/852. URL: <https://www.ams.org/jams/2016-29-04/S0894-0347-2016-00852-4/> (visited on 10/25/2021).
- [19] François Fleuret et al. "Comparing machines and humans on a visual categorization test". In: *Proceedings of the National Academy of Sciences* 108.43 (2011), pp. 17621–17625. DOI: 10.1073/pnas.1109168108. eprint: <https://www.pnas.org/doi/pdf/10.1073/pnas.1109168108>. URL: <https://www.pnas.org/doi/abs/10.1073/pnas.1109168108>.
- [21] Juan A. Gallego et al. "Neural Manifolds for the Control of Movement". In: *Neuron* 94.5 (2017), pp. 978–984. ISSN: 08966273. DOI: 10.1016/j.neuron.2017.05.025. URL: <https://linkinghub.elsevier.com/retrieve/pii/S0896627317304634> (visited on 10/23/2021).

- [22] Stuart Geman. "Hierarchy in machine and natural vision". In: *Proceedings of the Scandinavian Conference on Image Analysis*. Vol. 1. 1999, pp. 179–184.
- [23] Danielle Gerhard. "Neuroscience. 5th Edition". eng. In: *The Yale Journal of Biology and Medicine* 86.1 (Mar. 2013). Publisher: YJBM, pp. 113–114. ISSN: 0044-0086. URL: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3584490/>.
- [24] Xavier Glorot and Yoshua Bengio. "Understanding the difficulty of training deep feedforward neural networks". In: *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*. Ed. by Yee Whye Teh and Mike Titterington. Vol. 9. Proceedings of Machine Learning Research. Chia Laguna Resort, Sardinia, Italy: PMLR, 2010, pp. 249–256. URL: <https://proceedings.mlr.press/v9/glorot10a.html>.
- [25] Richard M. Goldberg et al. "A Randomized Controlled Trial of Fluorouracil Plus Leucovorin, Irinotecan, and Oxaliplatin Combinations in Patients With Previously Untreated Metastatic Colorectal Cancer". en. In: *Journal of Clinical Oncology* 22.1 (Jan. 2004), pp. 23–30. ISSN: 0732-183X, 1527-7755. DOI: [10.1200/JCO.2004.09.046](https://doi.org/10.1200/JCO.2004.09.046). URL: <http://ascopubs.org/doi/10.1200/JCO.2004.09.046> (visited on 11/05/2021).
- [26] Laura Gwilliams and Jean-Rémi King. "Performance-optimized hierarchical models only partially predict neural responses during perceptual decision making". In: *bioRxiv* (2017). DOI: [10.1101/221630](https://doi.org/10.1101/221630).
- [27] Jay Hegde and Daniel J Felleman. "Reappraising the functional implications of the primate visual anatomical hierarchy". In: *The Neuroscientist* 13.5 (2007), pp. 416–421.
- [28] Dan Hendrycks and Kevin Gimpel. "Bridging Nonlinearities and Stochastic Regularizers with Gaussian Error Linear Units". In: *CoRR* abs/1606.08415 (2016). arXiv: [1606.08415](https://arxiv.org/abs/1606.08415). URL: <http://arxiv.org/abs/1606.08415>.
- [29] David Hong, Tamara G. Kolda, and Jed A. Duersch. "Generalized Canonical Polyadic Tensor Decomposition". In: *SIAM Review* 62.1 (2020), pp. 133–163. ISSN: 0036-1445, 1095-7200. DOI: [10.1137/18M1203626](https://doi.org/10.1137/18M1203626). arXiv: [1808.07452](https://arxiv.org/abs/1808.07452). URL: <http://arxiv.org/abs/1808.07452> (visited on 06/10/2021).
- [30] *How to Visualize Filters and Feature Maps in Convolutional Neural Networks*. <https://machinelearningmastery.com/how-to-visualize-filters-and-feature-maps-in-convolutional-neural-networks/>. Accessed: 2021-11-08. 2021.
- [32] Michael Kerber, Dmitriy Morozov, and Arnur Nigmetov. "Geometry Helps to Compare Persistence Diagrams". en. In: *arXiv:1606.03357 [cs]* (June 2016). arXiv: 1606.03357. URL: <http://arxiv.org/abs/1606.03357> (visited on 10/01/2021).
- [34] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*. NIPS'12. Lake Tahoe, Nevada: Curran Associates Inc., 2012, 1097–1105.
- [35] Y. Lecun et al. "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324. DOI: [10.1109/5.726791](https://doi.org/10.1109/5.726791).
- [37] Luis M. Martinez and Jose-Manuel Alonso. "Complex Receptive Fields in Primary Visual Cortex". In: *The Neuroscientist* 9.5 (Oct. 2003). Publisher: SAGE Publications Inc STM, pp. 317–331. ISSN: 1073-8584. DOI: [10.1177/1073858403252732](https://doi.org/10.1177/1073858403252732). URL: <https://doi.org/10.1177/1073858403252732> (visited on 11/05/2021).
- [38] *Notes for CS231n: Convolutional Neural Networks for Visual Recognition*. <https://cs231n.github.io/>. Accessed: 2021-11-08. 2021.
- [39] Christopher Olah. *Neural Networks, Manifolds, and Topology*. 2014. URL: <http://colah.github.io/posts/2014-03-NN-Manifolds-Topology/>.
- [40] Randall O'Reilly et al. "Recurrent Processing during Object Recognition". In: *Frontiers in Psychology* 4 (2013). ISSN: 1664-1078. URL: <https://www.frontiersin.org/article/10.3389/fpsyg.2013.00124>.

- [41] Sayak Paul and Pin-Yu Chen. "Vision Transformers are Robust Learners". In: (2021). DOI: [10.48550/ARXIV.2105.07581](https://doi.org/10.48550/ARXIV.2105.07581). URL: <https://arxiv.org/abs/2105.07581>.
- [42] Jonathan Pillow. "Vision: From Eye to Brain (Chap 3, Part II)". en. In: (2022), p. 34.
- [43] Tomaso Poggio, Steve Smale, et al. "The mathematics of learning: Dealing with data". In: *Notices of the AMS* 50.5 (2003), pp. 537–544.
- [44] Maithra Raghu et al. "Do Vision Transformers See Like Convolutional Neural Networks?" en. In: (2021), p. 13.
- [45] Matthew Ricci, Jinkyung Kim, and Thomas Serre. "Same-different problems strain convolutional neural networks". en. In: (2018), p. 6.
- [46] Maximilian Riesenhuber and Tomaso Poggio. "Hierarchical models of object recognition in cortex". In: *Nature neuroscience* 2.11 (1999), pp. 1019–1025.
- [49] Gurjeet Singh et al. "Topological analysis of population activity in visual cortex". In: *Journal of Vision* 8.8 (June 2008), pp. 11–11. ISSN: 1534-7362. DOI: [10.1167/8.8.11](https://doi.org/10.1167/8.8.11). eprint: https://arvojournals.org/arvo/content_public/journal/jov/933530/jov-8-8-11.pdf. URL: <https://doi.org/10.1167/8.8.11>.
- [51] Courtney J. Spoerer, Patrick McClure, and Nikolaus Kriegeskorte. "Recurrent Convolutional Neural Networks: A Better Model of Biological Object Recognition". In: *Frontiers in Psychology* 8 (2017). ISSN: 1664-1078. URL: <https://www.frontiersin.org/article/10.3389/fpsyg.2017.01551>.
- [52] Sebastian Stabinger, Antonio Rodríguez-Sánchez, and Justus Piater. *25 years of CNNs: Can we compare to human abstraction capabilities?* 2016. DOI: [10.48550/ARXIV.1607.08366](https://doi.org/10.48550/ARXIV.1607.08366). URL: <https://arxiv.org/abs/1607.08366>.
- [53] Jay S. Stanley III. "Geometric Data Organization: Algorithms and Applications". English. ISBN: 9798516931055 Publication Title: ProQuest Dissertations and Theses 27736931. Ph.D. Ann Arbor: Yale University, 2020. URL: <https://www.proquest.com/dissertations-theses/geometric-data-organization-algorithms/docview/2544526068/se-2?accountid=10267>.
- [54] Mark Stopfer, Vivek Jayaraman, and Gilles Laurent. "Intensity versus Identity Coding in an Olfactory System". In: *Neuron* 39.6 (Sept. 2003), pp. 991–1004. ISSN: 0896-6273. DOI: [10.1016/j.neuron.2003.08.011](https://doi.org/10.1016/j.neuron.2003.08.011). URL: <https://www.sciencedirect.com/science/article/pii/S089662730300535X>.
- [55] The GUDHI Project. *GUDHI User and Reference Manual*. 3.4.1. GUDHI Editorial Board, 2021. URL: https://gudhi.inria.fr/python/latest/wasserstein_distance_user.html.
- [56] Christopher Tralie, Nathaniel Saul, and Rann Bar-On. "Ripser.py: A Lean Persistent Homology Library for Python". In: *The Journal of Open Source Software* 3.29 (2018), p. 925. DOI: [10.21105/joss.00925](https://doi.org/10.21105/joss.00925). URL: <https://doi.org/10.21105/joss.00925>.
- [57] Ashish Vaswani et al. "Attention is All you Need". en. In: (2017), p. 11.
- [59] Rikiya Yamashita et al. "Convolutional neural networks: an overview and application in radiology". In: *Insights into Imaging* 9.4 (Aug. 2018), pp. 611–629. ISSN: 1869-4101. DOI: [10.1007/s13244-018-0639-9](https://doi.org/10.1007/s13244-018-0639-9). URL: <https://doi.org/10.1007/s13244-018-0639-9>.
- [60] Byron M. Yu et al. "Gaussian-Process Factor Analysis for Low-Dimensional Single-Trial Analysis of Neural Population Activity". In: *Journal of Neurophysiology* 102.1 (July 2009). Publisher: American Physiological Society, pp. 614–635. ISSN: 0022-3077. DOI: [10.1152/jn.90941.2008](https://doi.org/10.1152/jn.90941.2008). URL: <https://doi.org/10.1152/jn.90941.2008> (visited on 11/02/2021).

“In a shorter time, more will be known about the most remote objects, namely the stars, than about the most nearby topic, namely perception.”

Aristotle

Yale-NUS College Capstone Project

DECLARATION & CONSENT

1. I declare that the product of this Project, the Thesis, is the end result of my own work and that due acknowledgement has been given in the bibliography and references to ALL sources be they printed, electronic, or personal, in accordance with the academic regulations of Yale-NUS College.
2. I acknowledge that the Thesis is subject to the policies relating to Yale-NUS College Intellectual Property ([Yale-NUS HR 039](#)).

ACCESS LEVEL

3. I agree, in consultation with my supervisor(s), that the Thesis be given the access level specified below: [check one only]

Unrestricted access

Make the Thesis immediately available for worldwide access.

Access restricted to Yale-NUS College for a limited period

Make the Thesis immediately available for Yale-NUS College access only from _____
(mm/yyyy) to _____ (mm/yyyy), up to a maximum of 2 years for the following reason(s): (please specify; attach a separate sheet if necessary):

After this period, the Thesis will be made available for worldwide access.

Other restrictions: (please specify if any part of your thesis should be restricted)
this thesis should not be made public, either within the Yale-NUS community or world-wide

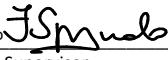
Zhang Liu, Cendana College
Name & Residential College of Student



Signature of Student

1 April 2022

Date

Francesca Spagnuolo 
Name & Signature of Supervisor

2 April 2022

Date

Acknowledgements

I would like to thank my supervisor Francesca Spagnuolo and co-supervisor and mentor Steven W. Zucker and Luciano Dyballa for their generous guidance and advice. They have made possible the many serendipitous moments in this project. There have been ups and downs in this project, but their support has helped me keep a keen spirit for learning and discovery. To the faculty at Yale-NUS for nurturing me into a good human being, and for bearing with my countless weaknesses and mistakes, particularly the MCS faculty who have had paramount influence in my development as a researcher: Prof. Francesca Spagnuolo, Prof. Olivier Danvy, Prof. David Smith, Prof. Robby Tan, and Prof. Timothy Wertz. Remarkably, they somehow believed in me even in times when I myself did not and provided me with the right guidance. To my family, thank you for loving me for who I am. To my suitemates, friends, and alumni, thank you for patiently bearing with my endless sharing of my work and ideas. Specifically, I acknowledge Xiyao Fu, Aparna Gupte, Mark Yuen for providing feedback; alumnus Abhinav Natarajan, Dion Ho, Sultan Aitzhan, Yiming Ng, Wei Han, Xin Run for supporting me throughout my college years; friends whom I have had conversations with about this capstone: Amanda Leong, Zhala Sarmast, Amberly Yeo, Glenda Chin, Dasha Haryfullina, Aiman Imtiaz, Lize Cai, Yanhua Wang, Linda Li, Jia Tang, Eric Lan, ... To the trees on campus, thank you for being there and reminding me of the life force within me. To whoever has read thus far, it's my pleasure to take you on this short tour with me. I have flaws, and my thesis has flaws - I sincerely invite more conversations about them.

A Complete List of Experiments

I have also conducted many exploratory experiments. A complete list of the experiments are included below. For the biological neural tensor, I conducted the following experiments:

1. applying tensor CP decomposition to generate the neural manifold
2. applying diffusion map to generate the neural manifold
3. applying persistent homology and Wasserstein distance to compare the topological features of the neural manifold
4. over training AlexNet

For the artificial neural tensor, I conducted the following experiments:

1. building the artificial neural tensor without considering shifts in images
2. building the artificial neural tensor with shifts in images
3. applying tensor CP decomposition on the artificial neural tensor (with shifts in images)
4. studying how the neural manifold changes during the training of an ANN (instead of using the pre-trained ANN)
5. applying persistent homology (though with limited explainability)

List of Figures

1.2	Illustration of the black box problem.	3
1.3	Examples of 2-dimensional manifolds.	4
1.4	Connections between neural manifolds and neural circuits. Adapted from [16].	6
2.1	Illustration of the hierarchical models [37].	10
2.2	Illustration of the recurrent models [37].	11
3.1	Illustration for tensor CP decomposition. (Adapted from [58])	14
4.1	Visualising neural data from lab experiments.	16
4.2	Reconstruction errors of different optimization methods. . .	18
4.3	First six tensor factors for retina.	18
4.4	Neural manifold for retina is discontinuous.	19
4.5	Neural manifold for V1 is continuous. The neurons are colored by excitatory and inhibitory functions and the two clusters overlap each other.	20
5.1	Visualizing the structure of the VGG16 model.	22
5.2	Computing output of one neuron unit in the CNNs given 10 input cat images.	23
5.6	Organization within the CNN neural manifold corresponds to neurons' spatial positions.	27
5.7	ViT input (image credit: Hiroto Honda).	28
5.8	ViT encoder (image credit: Hiroto Honda).	29

5.11	Organization within the ViT neural manifold corresponds to neurons' spatial positions.	31
5.13	Neural manifold for CRNN (shallow layer).	33
5.15	Neural factors for CRNN are distinct from CNN and ViT.	34
B.1	Visualising the original spiral data.	54
B.2	First non-trivial coordinate function.	54
B.3	Sample data from the MNIST.	55
B.4	First two non-trivial coordinate functions (plotting only five of the digits classes).	55
C.1	Illustration for PCA. (Adapted from [58])	56
D.1	First 5 tensor factors for face image data.	58
D.2	Some arbitrary clusters showing the results of tensor CP decomposition for face data.	59
F.1	Results for applying persistent homology on the three-dimensional embedding of X_1 .	71
F.2	Results for applying persistent homology on the three-dimensional embedding of X_2 .	71
F.3	Results for applying persistent homology on the three-dimensional embedding of X_3 .	71
F.4	Results for applying persistent homology on the three-dimensional embedding of X_4 .	72
F.5	Results for applying persistent homology on the three-dimensional embedding of X_5 .	72
F.6	Results for applying persistent homology on the three-dimensional embedding of X_6 .	72

F.7	Results for applying persistent homology on the 2-sphere.	74
F.8	Results for applying persistent homology on the three-dimensional torus.	74

List of Algorithms

4.2	Algorithm for NTF on retina tensor.	17
4.3	Algorithm for computing the mean flow ratio.	21
5.1	Algorithm for CNN artificial neural tensor.	26
5.2	Algorithm for ViT artificial neural tensor.	30

List of Tables

1.1	Summary of comparison.	6
F.1	Pairwise Wasserstein distance between persistent diagrams for homology group H_0	73
F.2	Pairwise Wasserstein distance between persistent diagrams for homology group H_1	73
F.3	Pairwise Wasserstein distance between persistent diagrams for homology group H_2	73
F.4	Wasserstein distance between persistent diagrams of the point clouds and 2-sphere and torus (homology group H_0).	74
F.5	Wasserstein distance between persistent diagrams of the point clouds and 2-sphere and torus (homology group H_1).	75
F.6	Wasserstein distance between persistent diagrams of the point clouds and 2-sphere and torus (homology group H_2).	75

B Diffusion Map

Besides tensor CP decomposition, we also explore the non-linear dimensionality reduction method of diffusion map. (Note that most of the results presented in this report can be obtained from tensor CP decomposition. However, when the former failed to provide meaningful results, we turn to the diffusion map.) The main references for this section are [9], [10] and [53]. We will now outline the steps in the diffusion map algorithm.

1. Discretize the underlying manifold as a weighted graph.

Definition 10 (Discretization of a manifold). *Given $X = \{x_1, x_2, \dots, x_n\}$, a set of data sampled from a Riemannian manifold \mathcal{M} , we can construct the discretization of \mathcal{M} as a weighted, undirected graph, $\mathcal{G}_X = (X, g_\varepsilon)$ where X is the set of vertices given by the data and g_ε is a Gaussian kernel function $g_\varepsilon : (x_i, x_j) \rightarrow [0, \infty]$ defined by the following rule*

$$g_\varepsilon(x_i, x_j) = \frac{1}{2} \left(\exp \frac{-\|x_i - x_j\|_2^2}{2\varepsilon(x_i)} + \exp \frac{-\|x_i - x_j\|_2^2}{2\varepsilon(x_j)} \right), \quad \forall x_i, x_j \in X, \quad (\text{B.1})$$

where $\varepsilon : X \rightarrow \mathbb{R}$ is the bandwidth function by which the Gaussian kernel is parameterized.

Gaussian kernel is only one of the possible kernel functions, but it gives a physically intuitive construction for learning the manifold underlying the data sample because it effectively considers a ball of radius ε around each data point.

The choice of ε determines the radius of neighborhood of $x_i \in X$. ε is application-specific and is usually a global bandwidth, that is, $\varepsilon(x_i) = \varepsilon(x_j)$ for all $x_i, x_j \in X$.

2. Compute the similarity matrix associated with the weighted graph.

Given the weighted graph \mathcal{G}_X , we can compute the corresponding similarity matrix. Each entry in the similarity matrix, W_{ij} , is the pairwise similarity value between data/vertices x_i and x_j and is taken as the weight of the edge between nodes x_i and x_j .

$$W_{ij} = \exp\left\{-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right\}. \quad (\text{B.2})$$

3. Construct a lazy random walk on \mathcal{G}_X .

First, we define random walk on the weighted, undirected graph \mathcal{G}_X :

Definition 11 (Random walk on graphs). A random walk on $\mathcal{G}_X = (X, g_\varepsilon)$ is a process that begins at some vertex x_i , and at each time step moves to another vertex x_j . Since the graph is weighted, it moves to a neighbor x_j with probability $p(j | i)$, proportional to the weight of the corresponding edge, and is defined as follows:

$$p(j | i) = \frac{W_{ij}}{\sum_k W_{ik}}. \quad (\text{B.3})$$

The random walk is lazy if we allow $p(i | i) > 0$, i.e., the probability of staying at some point i as the time step moves forward is non-zero.

The probabilities $p(j | i)$ can be represented by a Markov matrix M , which is essentially the similarity matrix normalized to have row sums equal to 1:

$$M = D^{-1}W, \text{ where } D_{ii} = \sum_k W_{ik}. \quad (\text{B.4})$$

The probability of reaching node x_j from x_i after t steps is then

$$p(t, j | i) = e_i^T M^t e_j. \quad (\text{B.5})$$

4. Eigendecomposition of the Markov matrix:

The eigendecomposition of M is derived from the eigendecomposition of $M_s = D^{1/2}MD^{-1/2} = \Omega\Lambda\Omega^T$:

$$M = D^{-1/2}\Omega\Lambda\Omega^TD^{-1/2} := \Psi\Lambda\Phi^T. \quad (\text{B.6})$$

Note that since Ψ and Φ are mutually orthogonal, Ψ contains the right eigenvectors, as shown below:

$$M\Psi = \Psi\Lambda\Phi^T\Psi = \Psi\Lambda = \Lambda\Psi. \quad (\text{B.7})$$

The eigendecomposition of M after t steps is then

$$M = \Psi\Lambda^t\Phi^T. \quad (\text{B.8})$$

- Diffusion coordinate functions are the right eigenvectors of Markov matrix scaled by their corresponding eigenvalues:

$$\Upsilon := \Psi\Lambda. \quad (\text{B.9})$$

- Diffusion distance after t steps is the following:

$$\|e_i^T\Upsilon - e_j^T\Upsilon\|^2 = \sum_k (p(t, k | i) - p(t, k | j))^2(D_{kk}^{-1}). \quad (\text{B.10})$$

- Infer geometric properties from the growth of eigenvalues of M .

For intuition behind the relation between the growth of eigenvalues and the geometric properties of the underlying manifold that we begin with, we consider the following two extreme situations:

- (a) If the discretization of the underlying manifold is a disconnected graph (none of the nodes are connected), then:

$$P = I, \lambda_i = \lambda_j \quad \forall i, j,$$

which implies a flat spectrum with zero decay rate.

- (b) If the discretization of the underlying manifold is a fully connected graph (each of the node is connected to all the rest of the nodes), assuming weights of all edges are 1, then:

$$\lambda_1 = 1, \lambda_i = 0 \quad \forall i \neq 1.$$

Key ideas of diffusion maps:

- The similarity kernel gives us the *local* geometry. As the time steps move forward, we integrate the local geometry and thus reveal the geometric structures at different scales.
- A cluster from a random walk is a region where the probability of escaping this region is low.

B.1 Demonstration of the method

As a demonstration of the diffusion maps method, we apply it on synthetic spiral data and MNIST handwritten digits images data:

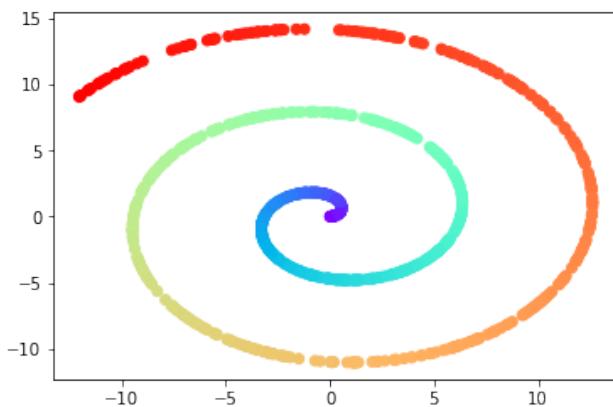


FIGURE B.1: Visualising the original spiral data.

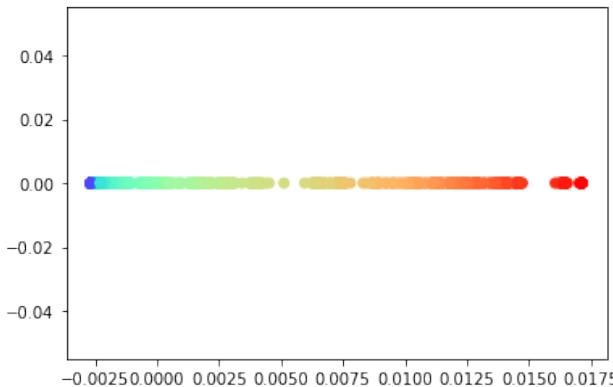


FIGURE B.2: First non-trivial coordinate function.

```

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5
6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6
7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7
8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8
9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9

```

FIGURE B.3: Sample data from the MNIST.

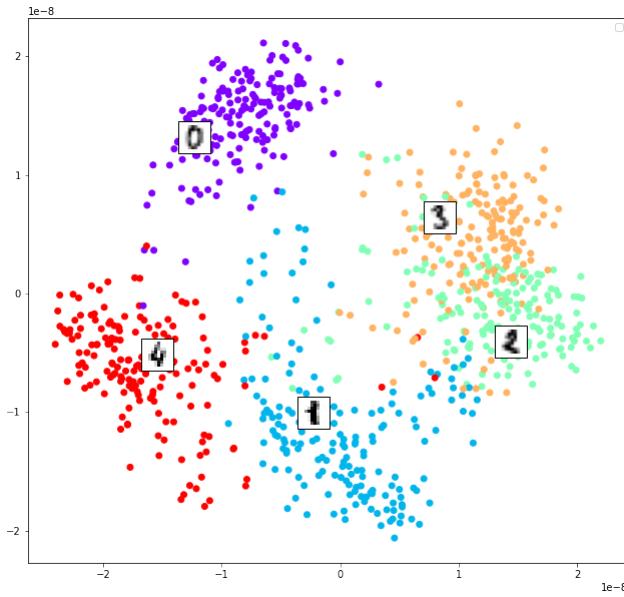


FIGURE B.4: First two non-trivial coordinate functions (plotting only five of the digits classes).

B.2 Remarks on non-linear dimensionality reduction

Setting the right parameters for non-linear dimensionality reduction is very challenging, especially for lab data where sampling errors are prevalent. For this reason, we will be investigating this method in greater details in the second half of the project instead.

C Principal Component Analysis

Principal component analysis (PCA) is the simplest matrix decomposition model. The PCA model can be formulated as an optimization problem. Given a N -by- M matrix \mathbf{X} with N variables and M features, we can approximate \mathbf{X} with the product of two orthogonal rank-one matrices:

$$\mathbf{X} \approx \mathbf{U}\mathbf{V}^T = \sum_{r=1}^R u_r \circ v_r, \quad (\text{C.1})$$

where \circ denotes the outer product operator. The equivalent element-wise formulation of the PCA model is

$$x_{ij} \approx \sum_{r=1}^R u_i^r v_j^r. \quad (\text{C.2})$$

With PCA, the dimensionality of the original matrix can be reduced from from N -by- M to N -by- R . This can be illustrated with the diagram below

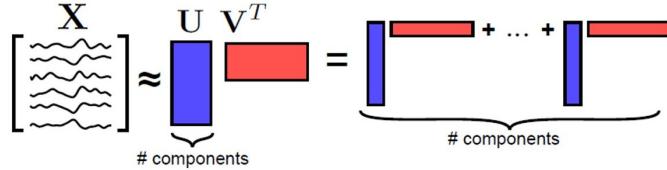


FIGURE C.1: Illustration for PCA. (Adapted from [58])

PCA seeks to solve a sequence of optimization problems:

- maximize variance:

$$\underset{\mathbf{V}}{\text{maximize}} \quad \|\mathbf{X}\mathbf{V}\mathbf{V}^T\|_F^2 \quad (\text{C.3})$$

subject to \mathbf{V} orthonormal,

- minimize residuals:

$$\underset{\mathbf{U}, \mathbf{V}}{\text{minimize}} \quad \|\mathbf{X} - \mathbf{U}\mathbf{V}^T\|_F^2 \quad (\text{C.4})$$

subject to \mathbf{U}, \mathbf{V} orthogonal,

where $\|\cdot\|_F$ denotes the Frobenius norm.

When the data has a non-negative constraint, non-negative matrix factorization (NMF) is used. In NMF, the second part of the optimization becomes the following instead:

$$\begin{aligned} & \underset{\mathbf{U}, \mathbf{V}}{\text{minimize}} \quad \|\mathbf{X} - \mathbf{UV}^T\|_F^2 \\ & \text{subject to} \quad \mathbf{U} \geq 0, \mathbf{V} \geq 0. \end{aligned} \tag{C.5}$$

D Tensor CP Decomposition on Face Recognition Dataset

D.1 Demonstrate tensor CP decomposition by using the face image dataset

As an intuitive demonstration for the tensor CP decomposition method, we apply tensor CP decomposition on face image data with 1000 face images, with each image having dimension 96-by-96. Thus, the face image data are encoded in a 3-way tensor of dimension 1000-by-32-by-32. Using tensor CP decomposition, we obtain the first 5 tensor components, which intuitively represent the prominent facial features in the face images:

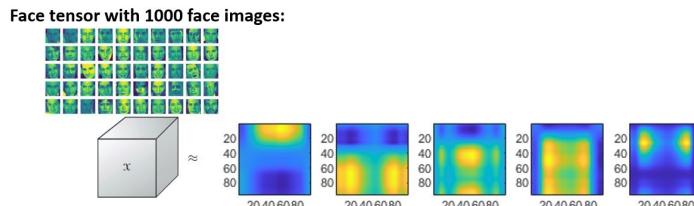


FIGURE D.1: First 5 tensor factors for face image data.

By projecting the data onto the first few principal components and apply the k -means clustering method, we obtain the following clusters grouped by similar face images:

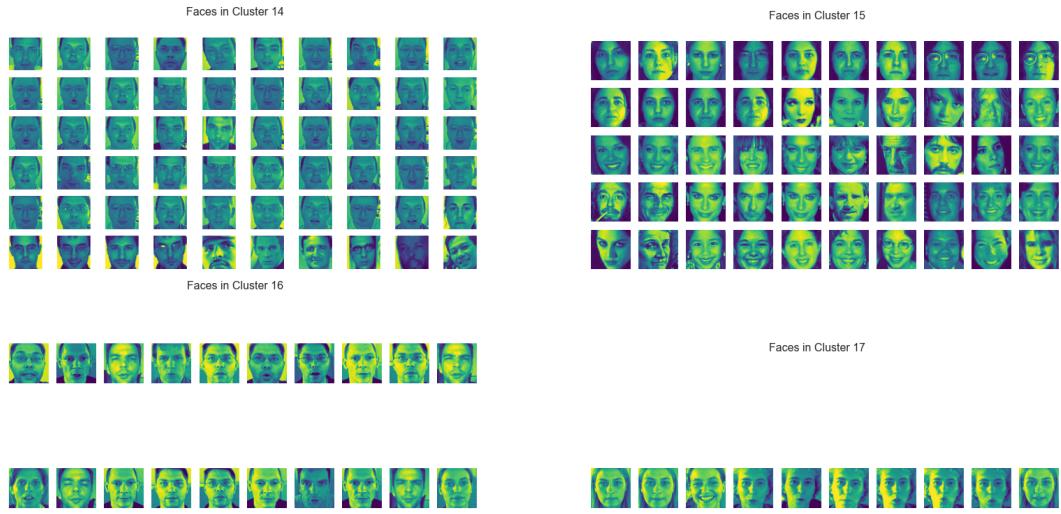


FIGURE D.2: Some arbitrary clusters showing the results of tensor CP decomposition for face data.

E Persistent Homology Theory

In this appendix, we provide the full details on the theoretical background of persistent homology. The main reference for the following exposition is [7]

E.1 Motivation

First of all, there are four major advantages for using topological methods to analyze high-dimensional point clouds.

1. Topology provides qualitative information which is required for data analysis.
2. Metrics are not theoretically justified. Compared to straightforward geometric methods, Topology is less sensitive to the actual choice of metrics.
3. Studying geometric objects using Topology does not depend on the coordinates.
4. Functoriality. This is the most important advantage.

Definition 12 (Functoriality). *For any topological space X , abelian group A , and integer $k \geq 0$, there is assigned a group $H_k(X, A)$. For any A and k , and any continuous map $f : X \rightarrow Y$, there is an induced homomorphism $H_k(f, A) : H_k(X, A) \rightarrow H_k(Y, A)$. Then functoriality refers to the following conditions:*

- $H_k(f \circ g, A) : H_k(f, A) \circ H_k(g, A)$
- $H_k(Id_X; A) = Id_{H_k(X, A)}$.

Functoriality addresses the ambiguities in statistical clustering methods – in particular the arbitrariness of various threshold choices. We now illustrate how exactly functoriality could be used in questions related to clustering.

Let X be the full data set and X_1, X_2 are the subsamples from the data set. If the set of clusterings $C(X_1), C(X_2), C(X_1 \cup X_2)$ correspond well, then we can conclude that the subsample clusterings correspond to clusterings in the full data set X .

E.2 Homotopy

Definition 13 (Homotopy). Homotopy is a family of maps $f_t : X \rightarrow Y$ where $t \in I$ such that $F : X \times I \rightarrow Y$ defined by $F(x, t) \mapsto f_t(x)$ is continuous.

Two maps f_0, f_1 are homotopic if there exists a homotopy f_t between f_0 and f_1 .

Definition 14 (Homotopy equivalence). A map $f : X \rightarrow Y$ is a homotopy equivalence if there is a map $g : Y \rightarrow X$ such that

- $f \circ g$ is homotopic to the identity map on Y , and
- $g \circ f$ is homotopic to f .

Two spaces X, Y are homotopy equivalent if there exists a homotopy equivalence $f : X \rightarrow Y$.

Theorem 15. *If f and g are homotopic, then $H_k(f, A) = H_k(g, A)$.*

If X and Y are homotopy equivalent, then $H_k(X, A) \cong H_k(Y, A)$.

Note that if two spaces are homotopy equivalent, then all their Betti numbers (defined in subsequent section) are equal.

Definition 16 (Topological space). *Associated to a simplicial complex (V, Δ) is a topological space $|(V, \Delta)|$. $|(V, \Delta)|$ may be defined using a bijection $\phi : V \rightarrow \{1, 2, \dots, N\}$ as the subspace of \mathbb{R}^N given by the union*

$$\bigcup_{\sigma \in \Delta} c(\sigma),$$

where $c(\sigma)$ is the convex hull of the set $\{e_{\phi(s)}\}_{s \in \sigma}$, where e_i denotes the i -th standard basis vector.

We often use abstract simplicial complexes to approximate topological spaces. For simplicial complexes the homology can be computed using only the linear algebra of finitely generated \mathbb{Z} -modules. In particular, for simplicial complexes, homology is algorithmically computable.

Definition 17 (Nerve). *Let X be a topological space, and let $\mathcal{U} = \{U_\alpha\}_{\alpha \in A}$ be any covering of X .*

The nerve of \mathcal{U} , denoted by $N(\mathcal{U})$, will be the abstract simplicial complex with vertex set A , and where a family $\{\alpha_0, \dots, \alpha_k\}$ spans a k -simplex if and only if $U_{\alpha_0} \cap U_{\alpha_1} \cap \dots \cap U_{\alpha_k} \neq \emptyset$.

One reason that this construction is very useful in the following “Nerve Theorem.” This theorem gives the criteria for $N(\mathcal{U})$ to be homotopy equivalent to the underlying topological space X .

Theorem 18 (Nerve Theorem). *Suppose that X and \mathcal{U} are as above, and suppose that the covering consists of open sets and is numerable. Suppose further that for all $\emptyset \subseteq S \subseteq A$, we have that $\bigcap_{s \in S} U_s$ is either contractible or empty. Then $N(\mathcal{U})$ is homotopy equivalent to X .*

The Nerve Theorem is very important in TDA since it provides a way to encode the topology of continuous spaces into abstract combinatorial structures that are more useful for designing computationally efficient data structures and algorithms.

E.3 Simplicial Complexes

Definition 19 (Simplicial complex). An abstract simplicial complex is a pair (V, Δ) , where V is a finite set, and Δ is a family of non-empty subsets of V such that

$$\tau \in \Delta \text{ and } \sigma \subseteq \tau \implies \sigma \in \Delta.$$

$\tau \in \Delta$ is face of Δ . The dimension of a face τ is $|\tau| - 1$.

(Intuition) A simplicial complex Δ in \mathbb{R}^n is a collection of simplices in \mathbb{R}^n such that

1. Every face of a simplex of Δ is in Δ .
2. The intersection of any two simplices of Δ is a face of each.

Definition 20 (f-vector). For each $n \geq -1$, let $f_n = f_n(\Delta)$ be the number of faces of Δ of dimension n .

The f-vector of Δ is the vector $(f_{-1}, f_0, f_1, \dots, f_d)$ where d is the dimension of Δ .

Example 21. Suppose the family of sets

$$\emptyset, \{a\}, \{b\}, \{c\}, \{d\}, \{a, b\}, \{a, c\}, \{b, c\}, \{b, d\}, \{c, d\}, \{a, b, c\}$$

form a simplicial complex E_1 . Then the f-vector of E_1 is $(1, 4, 5, 1)$.

Definition 22 (Oriented complex). (Intuition) An oriented simplex is a simplex σ together with an orientation of σ . If $\{a_0, a_1, \dots, a_p\}$ spans a p -complex σ , then we denote the oriented simplex with $[a_0, a_1, \dots, a_p]$.

(Formal) Let \mathbb{F} be a commutative ring, Δ be a simplicial complex. For each $n \geq -1$, we form a free \mathbb{F} -module $\tilde{C}_n(\Delta; \mathbb{F})$ with a basis indexed by the n -dimensional faces of Δ . For each n -dimensional face $a_0a_1 \cdots a_n$, we have a basis element $\vec{e}_{a_0, a_1, \dots, a_n}$.

We refer to the basis element $\vec{e}_{a_0, a_1, \dots, a_n}$ as an oriented simplex. The concept of an oriented simplex is analogous to the idea of a unit vector (which gives the direction and has unit length).

Definition 23 (Chain group of degree n). We refer to the free \mathbb{F} -module $\tilde{C}_n(\Delta; \mathbb{F})$ in the previous definition as the chain group of degree n . The rank of $\tilde{C}_n(\Delta; \mathbb{F})$ is the n -th value $f_n(\Delta)$ in the f -vector of Δ .

Example 24. For $E_1 = \{\emptyset, \{a\}, \{b\}, \{c\}, \{d\}, \{a, b\}, \{a, c\}, \{b, c\}, \{b, d\}, \{c, d\}, \{a, b, c\}\}$, we have the following chain groups of degrees $-1, 0, 1, 2$, respectively:

- $\tilde{C}_{-1}(E_1) = \{\lambda e_\emptyset : \lambda \in \mathbb{F}\} \cong \mathbb{F}$.
- $\tilde{C}_0(E_1) = \{\lambda_a e_a + \lambda_b e_b + \lambda_c e_c + \lambda_d e_d : \lambda_a, \lambda_b, \lambda_c, \lambda_d \in \mathbb{F}\} \cong \mathbb{F}^4$.
- $\tilde{C}_1(E_1) = \{\lambda_{ab} e_{a,b} + \dots + \lambda_{cd} e_{c,d} : \lambda_{ab}, \dots, \lambda_{cd} \in \mathbb{F}\} \cong \mathbb{F}^5$.
- $\tilde{C}_2(E_1) = \{\lambda e_{a,b,c} : \lambda \in \mathbb{F}\} \cong \mathbb{F}$.

Definition 25 (Simplicial chain complex). The simplicial chain complex of a simplicial complex Δ , denoted with $C(\Delta)$, is defined as:

$$\dots \xrightarrow{\partial_{n+2}} \tilde{C}_{n+1}(\Delta) \xrightarrow{\partial_{n+1}} \tilde{C}_n(\Delta) \xrightarrow{\partial_n} \tilde{C}_{n-1}(\Delta) \xrightarrow{\partial_{n-1}} \dots$$

Example 26. The simplicial chain complex of E_1 , $C(E_1)$ is

$$\begin{array}{ccccccc} 0 & \longrightarrow & \tilde{C}_2(E_1) & \xrightarrow{\partial_2} & \tilde{C}_1(E_1) & \xrightarrow{\partial_1} & \tilde{C}_0(E_1) \xrightarrow{\partial_0} \tilde{C}_{-1}(E_1) \longrightarrow 0 \\ & & \downarrow \cong & & \downarrow \cong & & \downarrow \cong \\ 0 & \longrightarrow & \mathbb{F} & \xrightarrow{\partial_2} & \mathbb{F}^5 & \xrightarrow{\partial_1} & \mathbb{F}^4 \xrightarrow{\partial_0} \mathbb{F} \longrightarrow 0 \end{array} \quad \text{Re-}$$

call that E_1 has f -vector $(1, 4, 5, 1)$.

Proposition 27 (The double boundary condition). Boundary maps satisfy the double boundary condition, that is, $\partial_n \circ \partial_{n+1} = 0 \quad \forall n$.

Definition 28 (Simplicial homology of degree k). (*Intuition*) The simplicial homology gives an algebraic measure on the amount of cycles that are not the boundaries.

(*Formal*) Define the \mathbb{F} -module $Z_k(\Delta; \mathbb{F})$ of cycles and the \mathbb{F} -module of the boundary group $B_k(\Delta; \mathbb{F})$ by the following formulas:

- $Z_k(\Delta; \mathbb{F}) = \ker(\partial_k) = \{Z \in \tilde{C}_k(\Delta; \mathbb{F}) : \partial_k(Z) = 0\}$
- $B_k(\Delta; \mathbb{F}) = \text{im}(\partial_{k+1}) = \{Z \in \tilde{C}_k(\Delta; \mathbb{F}) : \partial_{k+1}(x), \quad x \in \tilde{C}_{k+1}(\Delta; \mathbb{F})\}$

By 27, $B_k(\Delta; \mathbb{F})$ is a submodule of $Z_k(\Delta; \mathbb{F})$, i.e., $B_k(\Delta; \mathbb{F}) \subseteq Z_k(\Delta; \mathbb{F}) \subseteq C_k(\Delta; \mathbb{F})$.

We define the simplicial homology in degree k of Δ to be the quotient group

$$\tilde{H}_k(\Delta; \mathbb{F}) = \ker(\partial_k)/\text{im}(\partial_{k+1}) \quad (\text{E.1})$$

$$= Z_k(\Delta; \mathbb{F})/B_k(\Delta; \mathbb{F}) \quad (\text{E.2})$$

Definition 29 (k -th Betti number). The k -th Betti number of the simplicial complex Δ is

$$\beta_k(\Delta; \mathbb{F}) := \dim \tilde{H}_k(\Delta; \mathbb{F}) \quad (\text{E.3})$$

$$= \dim \ker(\partial_k) - \dim \text{im}(\partial_{k+1}). \quad (\text{E.4})$$

- elements in $\ker(\partial_k)$ are called k -cycles.
- elements in $\text{im}(\partial_k)$ are called k -boundaries.
- k -cycles that are not boundaries represent k -holes, which means that the k -th Betti number $\beta_k(\Delta; \mathbb{F})$ is the number of k -holes.

Definition 30 (Homology as “holes”). In certain well-behaved cases, homology can be interpreted as an algebraic measure on the amount of “holes” in a simplicial complex Δ .

Formally, given $X \subseteq \mathbb{R}^n$, a “hole” of X is a bounded connected component of $\mathbb{R}^n \setminus X$. For a given $k \geq 0$, suppose the $(k+1)$ -skeleton $\Delta^{(k+1)}$ of Δ has a geometric realization $X \subseteq \mathbb{R}^n$. Then $\tilde{H}_k(\Delta; \mathbb{F})$ is a free \mathbb{F} -module of rank equal to the number of holes of X .

Homology associates a vector space $H_k(X)$ to a topological space X for each $k \in \mathbb{N}$.

- $H_0(X)$ is the number of path components in X .
- $H_1(X)$ is the number of holes in X .
- $H_2(X)$ is the number of voids in X .

Remark 31. Note, however, that in the general metric spaces, the interpretation that homology measures the amount of “holes” does not hold.

E.4 Persistent homology

The main idea of *persistence* is that instead of selecting a fixed value of the threshold ε , we would like to obtain a useful summary of the homological information for all the different values of ε at once. As ε increases, we add simplicies to the complexes and detect which features “persist.”

The main advantages of persistent homology are threefold:

1. It is based on algebraic topology.
2. It is computable via linear algebra.
3. It is robust with regards to perturbations in the input data.

Definition 32 (Filtered simplicial complex). A subcomplex of Δ is a subset $\Delta^i \subseteq \Delta$ that is also a simplicial complex. Let Δ be a finite simplicial complex and let $\Delta^1 \subset$

$\Delta^2 \subset \cdots \subset \Delta^m = \Delta$ be a finite sequence of nested subcomplexes of Δ . The simplicial complex Δ with such a sequence of subcomplexes, $\emptyset \subseteq \Delta^1 \subseteq \Delta^2 \subseteq \cdots \subseteq \Delta^m = \Delta$, is called filtered simplicial complex.

Definition 33 (p -persistent k -th homology group). Given a filtered complex, for the i -th subcomplex Δ^i we can compute the associated

- the boundary maps ∂_k^i for all dimensions k .
- boundary matrices M_k^i for all dimensions k .
- groups C_k^i, Z_k^i (cycle group), B_k^i (boundary group), and H_k^i (homology group).

Then the p -persistent k -th homology group $H_k^{i,p}$ of Δ^i is

- $H_k^{i,p} = Z_k^i / (B_k^{i+p} \cap Z_k^i)$
- (equivalently) $H_k^{i,p} \cong \text{Im}(\eta_k^{i,p})$, where $\eta_k^{i,p}$ is a bijection $\eta_k^{i,p} : H_k^i \rightarrow H_k^{i+p}$ that maps a homology class into another homology class containing it.

Note that this is simply the definition of homology group of degree k in Definition 28 with the additional notion of persistence.

Definition 34 (p -persistent k -th homology group). We now have the persistent version of 29:

The p -persistent k -th Betti number of Δ^i is $\beta_{k,p} =$ the rank of the free subgroup of $H_k^{i,p}$.

Definition 35 (Persistence complex). A persistence complex \mathcal{C} is a family of chain complexes $\{C_*^i\}_{i \geq 0}$ over R , together with chain map's $f^i : C_*^i \rightarrow C_*^{i+1}$, so that we have the following diagram:

$$C_*^0 \xrightarrow{f^0} C_*^1 \xrightarrow{f^1} C_*^2 \xrightarrow{f^2} \dots$$

The filtered simplicial complex defined in 32 with inclusion maps for the simplices then becomes a persistence complex. To illustrate, below is a portion of a persistence complex with the chain complexes expanded.

$$\begin{array}{ccccccc}
 & \downarrow \partial_3 & \downarrow \partial_3 & \downarrow \partial_3 & & & \\
 C_2^0 & \xrightarrow{f^0} & C_2^1 & \xrightarrow{f^1} & C_2^2 & \xrightarrow{f^2} & \dots \\
 \downarrow \partial_2 & & \downarrow \partial_2 & & \downarrow \partial_2 & & \\
 C_1^0 & \xrightarrow{f^0} & C_1^1 & \xrightarrow{f^1} & C_1^2 & \xrightarrow{f^2} & \dots \\
 \downarrow \partial_1 & & \downarrow \partial_1 & & \downarrow \partial_1 & & \\
 C_0^0 & \xrightarrow{f^0} & C_0^1 & \xrightarrow{f^1} & C_0^2 & \xrightarrow{f^2} & \dots
 \end{array}$$

Definition 36 (Persistence modules). A persistence module \mathcal{M} is a family of R -modules M^i , together with homomorphisms $\phi^i : M^i \rightarrow M^{i+1}$.

Theorem 37 (Classification Theorem). For a finite persistence module \mathcal{M} with coefficients in the field F ,

$$H_*(\mathcal{M}; F) \cong \underbrace{\bigoplus_i x^{t_i} F(x)}_{\text{free module}} \oplus \underbrace{\left(\bigoplus_j x^{r_j} (F(x)/(x^{s_j} F[x])) \right)}_{\text{torsion module}} \quad (\text{E.5})$$

There exists a bijection between the set of free elements and the set of homology generators with birth at t_i and persist for all future parameter values.

There exists a bijection between the set of torsion elements and the set of homology generators with birth at r_j and death at $r_j + s_j$.

The classification theorem gives the fundamental characterization of persistence barcode.

Theorem 38 (Barcode as the persistence analogue of Betti number). The rank of $H_k^{i \rightarrow j}(\mathcal{C}; F)$ gives the number of intervals in the barcode of $H_k^{i \rightarrow j}(\mathcal{C}; F)$ spanning the parameter interval $[i, j]$. In particular, $H_*^{i \rightarrow j}(\mathcal{C}_*^i; F)$ gives the number of intervals that contain i .

Remark 39. As with Betti number, the barcode for H_k does not give the actual structure of the homology group, but just a continuously parameterized rank. The barcode is useful in that it can qualitatively filter out topological noise (since they are “short-lived” features) and capture significant topological features (features that persist over increasing values of ε).

F Topological Methods

F.1 Topological data analysis

This chapter provides the details for an explorative study that extends the method by further extracting topological features from the neural manifolds in response to different visual stimuli. We present here an overview of the steps of our implementation and the results. Full details on the theoretical background of persistent homology can be found in Appendix ??.

Recall that in Chapter 6, we showed the results of using diffusion map to generate the three-dimensional neural manifolds corresponding to six stimuli types, respectively. In this chapter we describe further experiments that investigated the topological structure of the neural manifolds.

F.2 Step 1: Applying persistent homology

We first applied persistent homology to extract the topological features from the six neural manifolds respectively. In this step, these topological features are represented by persistence barcodes and persistence diagrams.

Our implementation used the package ripser ([56]) to obtain the respective persistence diagrams from the embeddings. Using the (birth, death)-intervals from each persistence diagram, we drew the equivalent persistence barcode representation. The complete code can be found in Appendix ?? . The following figures show the results from applying persistent homology.

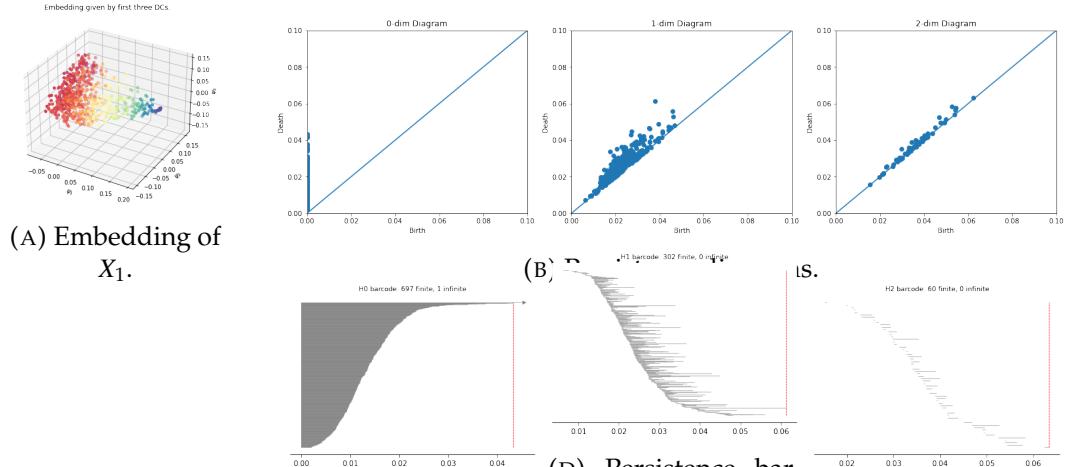


FIGURE F.1: Results for applying persistent homology on the three-dimensional embedding of X_1 .

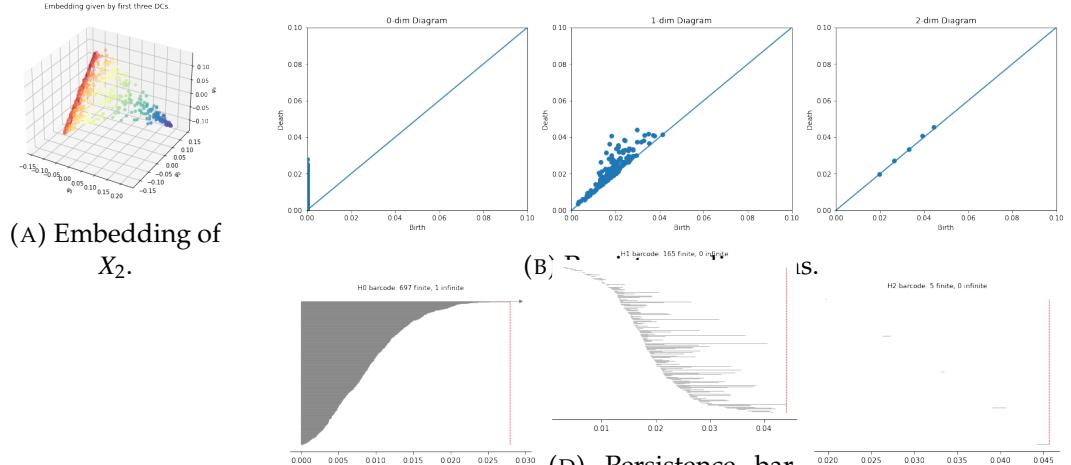


FIGURE F.2: Results for applying persistent homology on the three-dimensional embedding of X_2 .

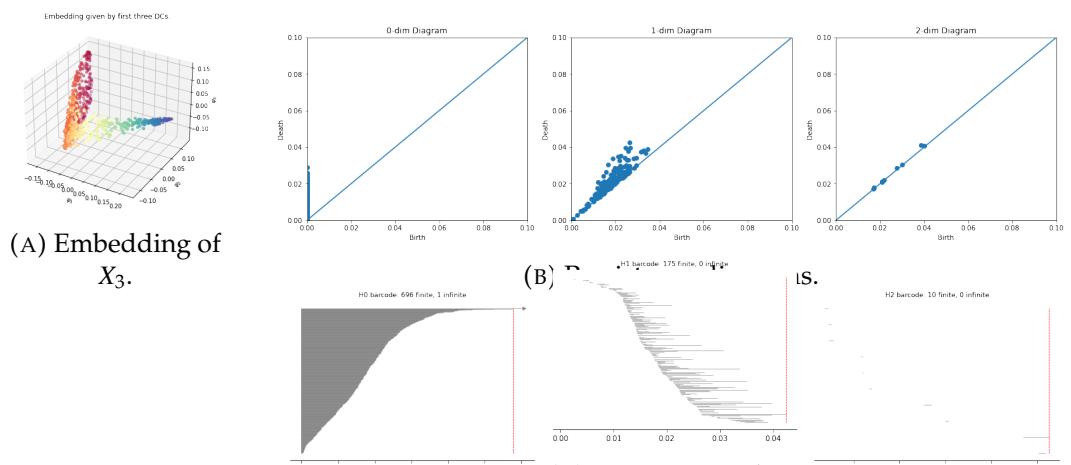


FIGURE F.3: Results for applying persistent homology on the three-dimensional embedding of X_3 .

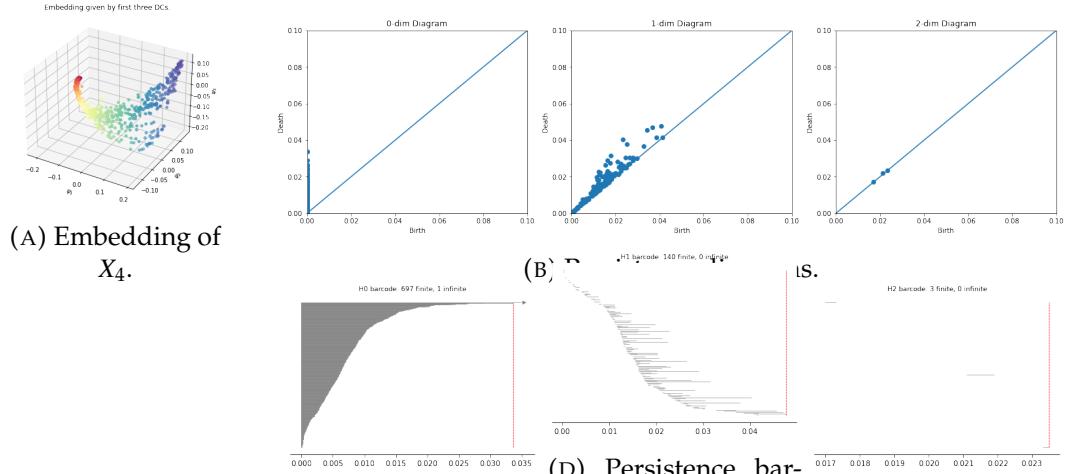


FIGURE F.4: Results for applying persistent homology on the three-dimensional embedding of X_4 .

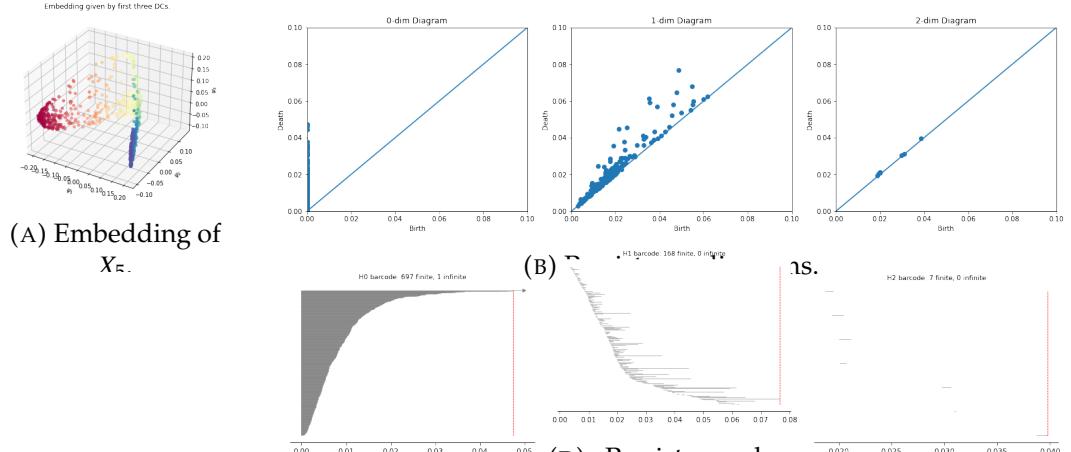


FIGURE F.5: Results for applying persistent homology on the three-dimensional embedding of X_5 .

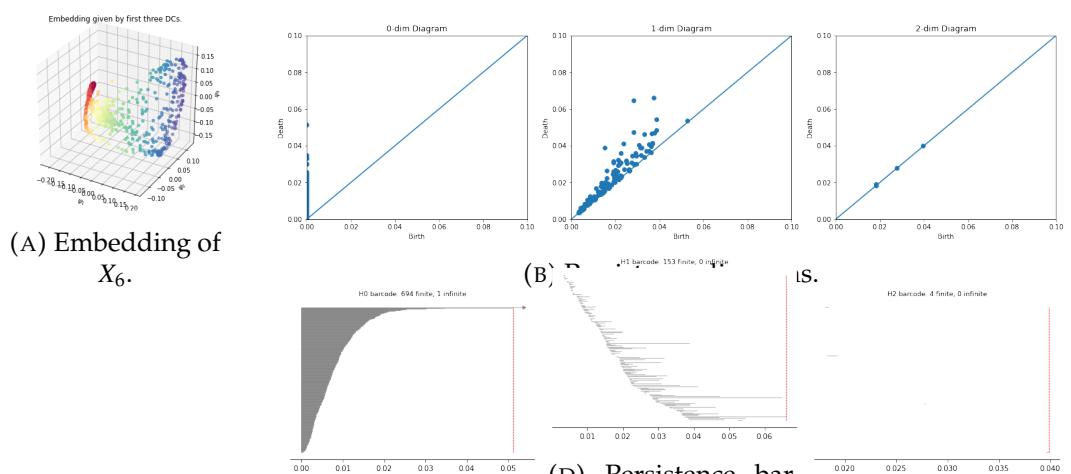


FIGURE F.6: Results for applying persistent homology on the three-dimensional embedding of X_6 .

F.3 Step 2: Pairwise Wasserstein distance

In this step, we used the gudhi package ([55]) to compute the pairwise Wasserstein distance between the persistence diagrams. The method used is based on ([32]).

We first computed the first pairwise distances between the persistence diagrams for each of the six point clouds.

H_0	X_1	X_2	X_3	X_4	X_5	X_6
X_1	0.0	2.77	3.05	3.95	3.08	3.42
X_2	2.77	0.0	0.43	1.35	1.0	0.84
X_3	3.05	0.43	0.0	1.03	0.94	0.66
X_4	3.95	1.35	1.03	0.0	1.11	0.72
X_5	3.08	1.0	0.94	1.11	0.0	0.51
X_6	3.42	0.84	0.66	0.72	0.51	0.0

TABLE F.1: Pairwise Wasserstein distance between persistent diagrams for homology group H_0 .

H_1	X_1	X_2	X_3	X_4	X_5	X_6
X_1	0.0	0.46	0.5	0.64	0.64	0.55
X_2	0.46	0.0	0.17	0.29	0.36	0.3
X_3	0.5	0.17	0.0	0.25	0.33	0.3
X_4	0.64	0.29	0.25	0.0	0.27	0.29
X_5	0.64	0.36	0.33	0.27	0.0	0.26
X_6	0.55	0.3	0.3	0.29	0.26	0.0

TABLE F.2: Pairwise Wasserstein distance between persistent diagrams for homology group H_1 .

H_2	X_1	X_2	X_3	X_4	X_5	X_6
X_1	0.0	0.06	0.06	0.06	0.06	0.06
X_2	0.06	0.0	0.01	0.0	0.01	0.0
X_3	0.06	0.01	0.0	0.0	0.01	0.0
X_4	0.06	0.0	0.0	0.0	0.0	0.0
X_5	0.06	0.01	0.01	0.0	0.0	0.0
X_6	0.06	0.0	0.0	0.0	0.0	0.0

TABLE F.3: Pairwise Wasserstein distance between persistent diagrams for homology group H_2 .

Building upon the same method, we further compared the six point clouds with known three-dimensional shapes (2-sphere and torus) based on the Wasserstein distance.

First, we extracted the topological features of the 2-sphere and three-dimensional torus, as we did for the point clouds in Step 2.

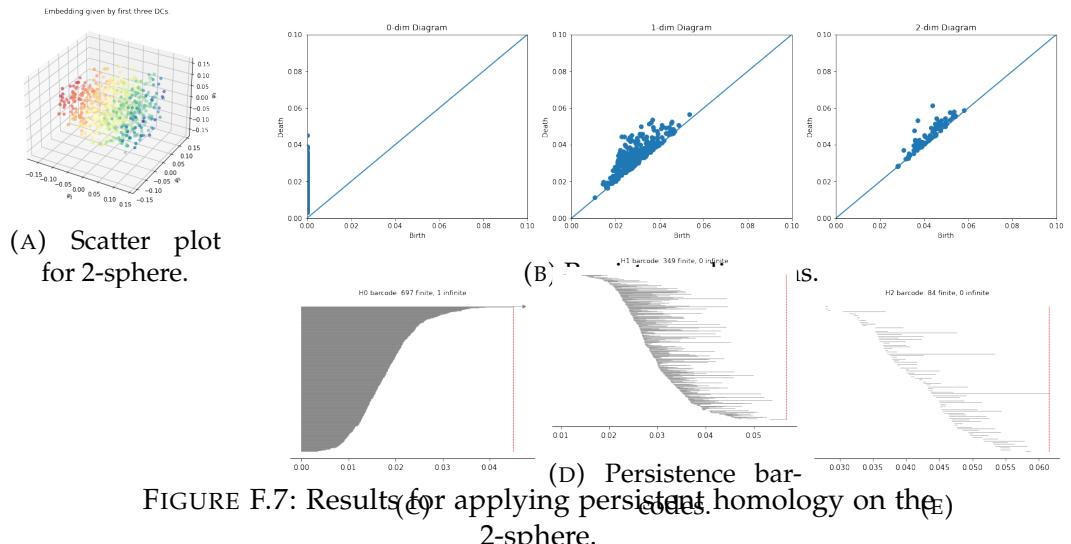


FIGURE F.7: Results for applying persistent homology on the 2-sphere.

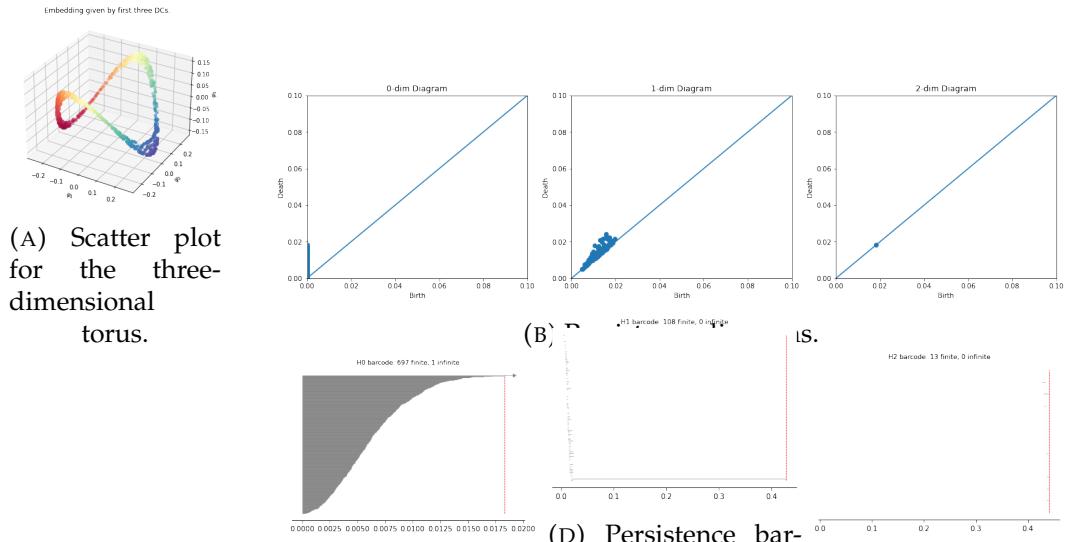


FIGURE F.8: Results for applying persistent homology on the three-dimensional torus.

Then, we computed the Wasserstein distance between the six point clouds and the known shapes:

H_0	X_1	X_2	X_3	X_4	X_5	X_6
2-sphere	2.48	4.46	4.64	5.18	4.53	4.83
torus	3.63	1.46	1.31	0.73	1.38	1.06

TABLE F.4: Wasserstein distance between persistent diagrams of the point clouds and 2-sphere and torus (homology group H_0).

H_1	X_1	X_2	X_3	X_4	X_5	X_6
2-sphere	0.62	0.77	0.80	0.87	0.85	0.76
torus	0.74	0.49	0.44	0.34	0.45	0.47

TABLE F.5: Wasserstein distance between persistent diagrams of the point clouds and 2-sphere and torus (homology group H_1).

H_2	X_1	X_2	X_3	X_4	X_5	X_6
2-sphere	0.11	0.12	0.12	0.12	0.13	0.12
torus	0.057	0.017	0.018	0.016	0.018	0.016

TABLE F.6: Wasserstein distance between persistent diagrams of the point clouds and 2-sphere and torus (homology group H_2).

F.4 Analysis of results

Based on the results from the above tables of Wasserstein distances, our observations and inferences are as follows.

1. The topological structure for X_1 is distinct from the rest.

Based on the pairwise Wasserstein distances in Tables F.1 and F.2, X_1 is notably different from the rest of the point clouds in terms of topological structure. In the context of our application, this implies that the neural population response evoked by stimulus type 1 is significantly different from the other stimulus types. This observation leads us to hypothesize that there is some neuroscientific reason behind this distinction in the topological structure of neural population response. It might be interesting to conduct further lab experiments to investigate what special properties this stimulus has and why it causes such different neural population response.

2. Wasserstein distances between persistence diagrams are nearly negligible for H_2 .

Based on Table F.3, pairwise Wasserstein distances between persistence diagrams for H_2 are small enough to be nearly negligible. This

implies that the intrinsic dimensionality of this neural data might be even lower than three-dimensional since there is no significant differences in homology groups H_2 for the point clouds.

3. Shape comparison with 2-sphere and torus.

Based on Tables F.4 and F.5, the Wasserstein distances between the point clouds and the 2-sphere are smaller than the Wasserstein distances between the point clouds and the torus for all X_i except for X_1 . We can thus infer that except for X_1 , all other point clouds are more similar to the shape of a torus than the 2-sphere. This implies that the topological structure of the neural population response evoked by stimuli type 1 is more similar to a sphere while the neural population response evoked by the rest of the stimuli types used in the experiments are more similar to a torus.

It would be interesting to compare this result with the hypothesis in ([3], [5], [25], [49]): If we are given an oriented stimulus, and if the orientation is a circular variable, then the hypothesis is that the neural population response evoked by such stimulus must have a topological structure equivalent to that of a circle. However, to fully test this hypothesis, further experiments with different types of stimuli need to be conducted.

F.5 Discussions

In this chapter, we demonstrated our proposed approach to compare the point clouds that represent the neural population response to different stimuli. Our proposed approach is contingent on the topological structures of the point clouds.

One significant limitation in this specific application is that lab data usually involve a lot of sampling errors such as missing data and inconsistent densities, causing noise to the true underlying geometry of the neural population response. Persistent homology works well on synthetic data, but might not work when the sampling errors are too significant.

The advantages of topological approach is that we can provide a succinct and useful summary of the global geometric structure of the neural spiking data, thus obtaining a useful account of how similarly the neurons collectively respond to visual stimuli. This is especially important in applications where the notion of connectedness and clusters are salient. As an emerging field, TDA will certainly see more applications in solving problems that involve understanding the geometric and topological structure of high-dimensional data.