Representations in Biological and Artificial Neural Networks

Jianghong Shi

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2022

Reading Committee:

Eric Shea-Brown, Chair

Michael Buice

Adrienne Fairhall

Program Authorized to Offer Degree:
Applied Mathematics

University of Washington

Abstract

Representations in Biological and Artificial Neural Networks

Jianghong Shi

Chair of the Supervisory Committee:
Eric Shea-Brown
Department of Applied Mathematics

Remarkably, artificial neural networks (ANNs) have shown astounding success in almost all aspects of artificial intelligence. Meanwhile, large scale experiments have gathered an unprecedented amount of data about the biological brain, both anatomical and functional. In this thesis, we make a series of interconnected endeavors to link ANNs with biological brains, and show that such links can shed light on our understanding of both systems. First, we establish and validate a paradigm for comparing ANN models with large scale functional data sets from mouse visual cortex. We show that comparing ANNs to the real brain is not only possible to do in a reliable way, but also helpful in revealing insights about computation in the biological brain. Second, we present the first (to our knowledge) ANN model of the mouse visual cortex (MouseNet) that is constrained by large-scale mesoscopic anatomical data. With the MouseNet model, we demonstrate the computational capabilities of a mouse-sized architecture and quantify the extent to which it recapitulates the neural representation of images in mouse visual cortex. Finally, we utilize the mathematical framework of linear ANNs to study learning dynamics in a simple model with parallel pathways, an important network feature that appears in MouseNet and is common in biological brains. We examine and quantify the surprisingly rich dynamics by which the learning process distributes the task-related knowledge among different network pathways.

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGMENTS

I want to thank the wonderful people I have come across throughout my PhD journey. It is the presence of every one of you that has made this journey colorful and full.

Firstly, I want to thank my co-advisors Eric Shea-Brown and Michael Buice. Thank you both for your guidance and support through all the years, especially for granting me the freedom to wander in various research directions. Thank you Eric for being so responsible and organized that I can always trust your wisdom for steering the wheels whenever I am uncertain about my direction. Thank you Michael for your critical and careful thinking in drawing scientific conclusions in our paper drafts. Your straightforward criticisms have paved ways for further improvements.

Thank you also to my collaborators Stefan Mihalas and Bryan Tripp. The CNN MouseNet project would not have been possible without your guidance and support. Thank you Stefan for providing your unmatched knowledge and deep understanding about the mouse brain. Thank you Bryan for your pioneering work and guidance on constructing anatomically constrained networks and for connecting us with Toronto groups. The discussions with Shahab Bakhtiari, Blake Richards and Graham Taylor are both eye-opening and fun.

I own special thanks to Adrienne Fairhall. You and Eric have created the best possible computational neuroscience community in UW one could wish for. From the joint group meetings to the journal clubs, and to all sorts of local seminars, conferences and summer schools, the opportunities to learn from local colleagues and outside experts are countless.

I must also thank my committee members Nathan Kutz and Wyeth Bair. Thank you Nathan for your contagious enthusiasm, positive energy and encouragement. Thank you Wyeth for your inspiring work and wonderful class on Artiphysiology.

Having been wandering in the interdisciplinary terrain, I am fortunate enough to come across and learn from people with different areas of expertise. I owe my thanks to all these wonderful people who have helped and supported me in one way or another.

In the UW computational neuroscience community, I want to thank the other members in Eric's group: Kameron Decker Harris, Alison Weber, Doris Voina, Matthew Farrell, Daniel Zdeblick, Shane Shang, Helena Liu, Tim Oleskiw, N. Alex Cayco Gajic, Guillaume Lajoie, Hannah Choi, Gabrielle Gutierrez, Merav Stern, Leenoy Meshulam, Stefano Recanatesi, Joel Zylberberg, Braden Brinkman. Also thanks to the members from Adrienne's group: Alison Duffy, Rich Pang, Hengji Wang, Anatoly Buchin, Ben Lansdell, Yoni Browning, Fereshteh Lagzi. And the broader UW community: Dean Pospisil, Satpreet Singh, Natalia Mesa, Nick Steinmetz, Rajesh Rao, Ariel Rokem, Fred Rieke, Bing Brunton and Jessica Huszar.

In the Allen Institute for Brain Science, I owe my thanks to all the scientists and staff who have contributed to the invaluable unprecedented large scale datasets, without which my thesis would not be possible. And thanks to the Allen Institute for Brain Science founder, Paul G. Allen, for his vision, encouragement, and support. Special thanks to the scientists I was fortunate to have opportunities to lean from: Saskia de Vries, Michael Oliver, Gabe Ocker, Nicholas Cain, Julie Harris, Séverine Durand, Jun Zhuang, Daniel Millman, Corinne Teeter, Koosha Khalvati, Brian Hu, Clay Reid, Christof Koch.

In my home department, I want to thank Bernard Deconinck, Hong Qian, Loyce Adams, Randy LeVeque, Anne Greenbaum, Aleksandr Aravkin, Lauren Lederer, Tony Garcia, Yian Ma, Yue Wang, Felix Ye, Yuan Gao, Yuying Liu, Kelly Liu and many more. The summer study for the qualifying exams with Xin Yang, Jize Zhang, Brian DeSilva, Kathleen Champion, Jeremy Upsal, Andreas Freund, Samuel Rudy and Weston Barger was memorable and fun.

Thank you also to my former advisor Ping Ao. Discussions with you in Seattle throughout the years have always been inspiring and encouraging.

Finally, I owe huge thanks to my friends and family members for their unconditional love and support. I am blessed with my kindest parents who unconditionally support their only child wandering on the other side of the world in her own pace with wholehearted love and yearning. And my amazing husband Tianqi Chen, who has always been holding my hands with warmest love and support, sharing every bit of happiness and sorrow in every aspect of life.

DEDICATION

to the singing brain

Chapter 1

INTRODUCTION

Remarkably, artificial neural networks (ANNs) have shown astounding success in almost all aspects of artificial intelligence. ANNs, though originally inspired by the biological brain, have deviated far from the real brain throughout their development. Interestingly, a recent surge of research studies indicate that a remarriage between the two can lead to fruitful results. In this thesis, we seek to extend the links between the ANNs and the real brain, by developing meaningful comparisons between activity patterns in the two, drawing inspirations from the anatomy of the biological brain, and utilizing the mathematical framework of ANNs' learning dynamics to understand the impact of basic features of this anatomy.

The biological brain area we focus on in this thesis is the mouse visual cortex. Compared to primates' visual cortex, the computation in rodents' visual cortex is less understood. On the other hand, large scale experiments have gathered an unprecedented amount of data about the mouse brain, both anatomical and functional. An ANN model with comparable scale serves a good candidate to assimilate these large scale data sets. In Chapter 2, we take the first step to compare ANN models with large scale functional data sets from mouse visual cortex. We show that comparing the real brain to artificial neural networks is not only possible, but also fruitful for our understanding of the brain. More importantly, by establishing the validity of a framework and metrics for comparing ANNs and brain data, we contribute criteria that can guide further development of models that increasingly match the biological brain.

While task-driven ANNs (ANNs that have been trained to perform some tasks, such as image classification) have shown great potential in studying biological neurons, they are not precise biological analogues. Constructing an ANN architecture which more closely

matches the anatomy and connectivity of the brain can further advance our analysis, opening doors to demonstrating the computational power with given biological resources, enabling comparisons of homologous groups of neurons in models and the mouse brain. In Chapter 3, we present the first (to our knowledge) ANN model of the mouse visual cortex (MouseNet) that is constrained by large scale mesoscopic anatomical data. With the MouseNet model, we can address key questions about mesoscale brain architecture and its role in task learning and performance.

A key feature in our constructed MouseNet architecture is its parallel pathways. These raise the question of how these different pathways learn different aspects of a task, and to what degree. It is not easy to address this question in general for a complicated task and architecture, but we can start from a simplified setting where relevant quantities are quantifiable and analytically tractable. In Chapter 4, we utilize the mathematical framework of linear ANNs' learning dynamics to study a simple model with parallel pathways. We examine and quantify the surprisingly rich dynamics by which the learning process distributes the task-related knowledge onto different pathways.

This thesis is organized as follows. Chapter 2-4 are self-contained papers, with interludes that connect their findings and approaches. In Chapter 5, we discuss conclusions and future directions. In the remainder of this chapter, we introduce the background of the work in more detail for the general audience.

1.1 The success of deep neural networks

Since the unprecedented success in large scale image classification [Krizhevsky et al., 2012], deep multi-layer artificial neural networks are replacing classical algorithms in solving innumerable hard problems in various areas. From speech recognition [Graves et al., 2013] to machine translation [Wu et al., 2016], from protein folding [Senior et al., 2020] to medical image analysis [Milletari et al., 2016], all the way to the game of go [Silver et al., 2016], they have produced results comparable to and in some cases surpassing human expert performance. In addition to their marvelous computational power, deep neural networks also

grant researchers with accessibility to all their trained parameters and activity patterns, enabling myriad manipulations to probe their representations, which empower them as great tools for scientific discovery as well [Raghu and Schmidt, 2020]. In the following, we will briefly introduce the deep learning framework and associated concepts, for a jump start aimed at the general audience. For a systematic treatment of this subject, one may refer to deep learning textbooks [Goodfellow et al., 2016, Zhang et al., 2019].

1.1.1 The deep learning framework in view of linear regression

Intriguingly, the way deep neural networks solve these very different tasks is universal in general: with a large scale **data set**, a network with some **architecture**, a **loss function**, and a **learning algorithm** to train the network to distill knowledge from the data under the guidance of the loss function.

Let's take the simplest linear regression problem under this framework to illustrate how it works. Say we have a **data set** of P examples $\{x^i, y^i\}, i = 1, 2, \ldots, P$, where x^i are input vectors and y^i are the corresponding output vectors. To build a relation from the input to the output, we try the simplest linear model $\hat{y} = w^T x + b$ as our **architecture**. To find the parameters w, b that match the model to the data as closely as possible, we set a square **loss function**

$$L(w, b) = \frac{1}{2} \sum_{i=1}^{P} ||y^i - w^T x^i - b||^2.$$

Then the final step is to find the parameters w^*, b^* which minimize the loss function, i.e. to solve

$$w^*, b^* = \underset{w,b}{\operatorname{argmin}}\, L(w, b).$$

For this simple linear regression problem, we can find an analytical solution. Nevertheless in the world of deep learning, analytically solvable problems are rare. Instead, a numerical **learning algorithm** such as gradient descent is applied to solve the optimization problem in practice. In this example, we initialize the weights with small random numbers, and calculate the gradient of loss function with respect to the weights and update the weights in

each iteration until convergence:

$$(w, b) \leftarrow (w, b) - \eta \frac{\partial L}{\partial (w, b)}$$

where η is a learning rate that controls the learning speed. In practice, a stochastic gradient descent learning algorithm is mostly used, where the loss is calculated with small batches of data instead of the whole data set.

1.1.2 The deep learning framework for image classification

The above simple linear regression example has demonstrated the components of the deep learning framework. For a more complicated problem, such as image classification, the four basic components are the same as in this basic example.

For instance, the deep learning framework has revolutionized the ImageNet [Deng et al., 2009] classification task since the development of the AlexNet [Krizhevsky et al., 2012] architecture. Here, the ImageNet data set provide millions of high resolution images with class labels as a large scale **data set**. (Note that the unprecedented scale of the ImageNet data set is one of the key ingredient which advanced the deep learning literature.)

In these image classification tasks, a deep neural network model takes a high resolution image as an input and transforms it into the probabilities of the input image belonging to each class. For transformations as complicated as this, we need more sophisticated architectures than the linear model described above, to obtain the needed "expressive" power. It turns out that convolutional neural network **architectures** such as AlexNet and VGG16 [Simonyan and Zisserman, 2014], which employ nonlinear transformations between network layers that are designed for the statistical properties of images, work very well for processing image inputs.

In the linear regression problem above, we used a squared error loss function. For the image classification task, we need to compare the predicted probability distribution over image labels with the true labels. A relative entropy **loss function** in the following form is

used for regressing class labels

$$H(p,q) = -\sum_{x} p(x) \log q(x) \tag{1.1}$$

Here $p(x)$ is the distribution provided by the true labels, and $q(x)$ is the distribution predicted by the network. The predicted probability distribution is generated by adding a softmax function to the output of the network, in the following form

$$softmax(z_i) = \exp(z_i) / \sum_{j} \exp(z_j).$$

Finally a stochastic gradient descent descent **learning algorithm** is applied to find the weights that minimize the loss function. Among these four components of the deep learning framework, this thesis will put an emphasis on the architecture, since it can be directly contrasted and compared between the ANNs and the real brain.

1.1.3 The architecture of convolutional neural networks

Although it has been pointed out by the universal approximation theory [Hornik et al., 1989] that a two-layer nonlinear neural network is able to approximate any function given enough width (the number of units in the layer), it is hard to find the network's weights through training with a poorly designed architecture when dealing with a large scale data set. As a matter of fact, the specific design of convolution neural network architecture is a key ingredient for its triumph in the image classification task.

For a typical convolutional neural network such as VGG16 [Simonyan and Zisserman, 2014](Figure 1.1A), the architecture is a single pathway feed-forward structure which consists of several building blocks such as a convolution layer, pooling layer, fully connected layer etc. When making a prediction or during training, images are fed as inputs into the bottom layer of the network and pass through a series of computations, leading to the prediction of the input image labels as the output of the network.

As for the building blocks of the VGG16 network, the fully connected layer is simply the linear model as we discussed above plus a nonlinear activation function. The shortcoming of

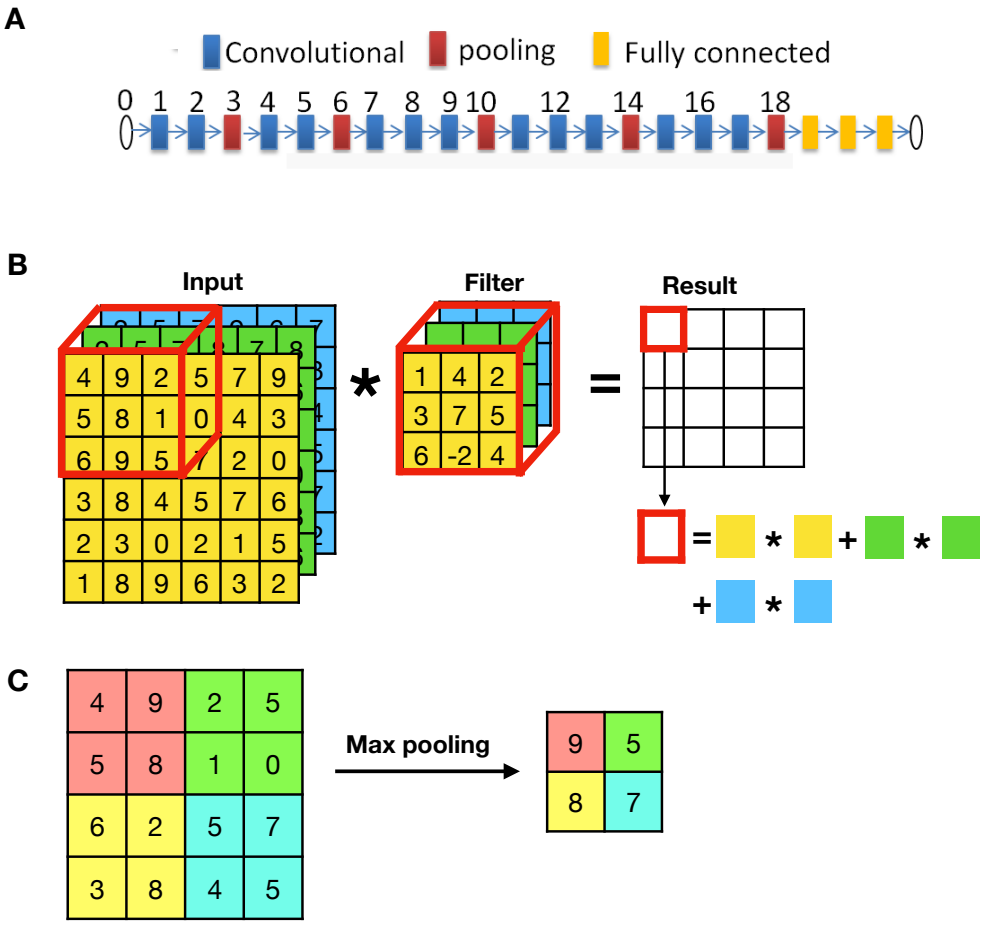

Figure 1.1: A typical deep neural network architecture and its building blocks. (A) Architecture of a VGG16 network. (B) Illustration of a convolutional layer. (C) Illustration of a max pooling layer.

the fully connected layer is that its parameters are dense, hence computationally expensive in terms of training or making predictions. Thus, fully connected layers are often used only at the top layers of a deep neural network. The convolutional layer, on the other hand, by sharing filter weights for different locations of a input, significantly reduces the number of its parameters, making the model sparser (meaning less connections between the units) and more computationally efficient.

An illustration of a convolutional layer is shown in Figure 1.1B. The standard convolutional layer is parameterized with a four dimensional filter $F(x, y, i, j)$, where x, y denote the horizontal and vertical location of the filter, and i, j index the input and output channels (For example, a color image input has three channels: red, green and blue). In the example of Figure 1.1B, we have $x, y, i \in \{0, 1, 2\}$, $j = 0$, with a total number of weights as $3 \times 3 \times 3 = 27$. If we denote the input layer activation as $A^l(\alpha, \beta, i)$ and the output activation as $A^{l+1}(a, b, j)$ ($\alpha, \beta \in \{0, \ldots, 5\}, a, b \in \{0, \ldots, 3\}$ denotes the horizontal and vertical location), then the convolution layer sets the relation between them as

$$A^{l+1}(a, b, j) = \sum_i \sum_x \sum_y A^l(as + x, bs + y, i) F(x, y, i, j) \tag{1.2}$$

where s is a meta-parameter called stride, which sets the step size of the convolution operation. In this example we have $s = 1$. The output activation will then be transformed with a non-linear function, typically a Rectified Linear Unit(ReLU) funcion $f(x) = max(0, x)$.

The pooling layer can be thought as a zoom out operation, where summary statistics are used to represent small neighbourhood regions. Figure 1.1C shows an example of max pooling layer, where the input matrix is separated into four smaller regions, with each region outputs its own max value. The output size is hence reduced to 2×2 from a 4×4 input. The pooling layers enable a deep neural network to work on different levels of abstraction, with bottom layers extracting lower level features (such as color and texture) and top layers extracting higher level features (such as object).

There are more building blocks in addition to these introduced above in the deep learning literature. With these building blocks, one can design all kinds of deep neural network architectures. In Chapter 3, we will develop a convolutional neural network architecture with real biological constraints and examine its properties.

1.2 The emerging large scale brain data sets

In contrast to ANNs, our knowledge about the biological brain can only be obtained through laboratory experiments. Thanks to the advancing imaging technology and developing exper-

imental pipelines, an unprecedented scale of brain data is being gathered, creating exciting and new opportunities for analysis.

An important animal species about which we have obtained such large scale data is the mouse. As a rapidly reproducing and genetically malleable mammal, the mouse serves a good candidate for producing large scale data sets with adequate functional complexity [Abbott et al., 2020]. Key data sets of this form have been produced at the Allen Institute. Anatomically, the Allen Mouse Brain Connectivity Atlas [Oh et al., 2014] contains mesoscale whole brain connectivity strength data from viral tract-tracing experiments. Physiologically, the Allen Brain Observatory [de Vries et al., 2020] provides a survey of tens of thousands of neurons' activities in the mouse visual cortex for various types of visual stimuli. In this thesis, we will integrate these unique data sets to reveal interesting properties of visual processing in the mouse brain.

1.3 The interplay between ANNs and the real brain

With marvelous computational power and adequate complexity, ANNs serve as a good model candidate for integrating these large scale data sets. Historically, ANNs were originally inspired by the design of the real brain, dating back to the 1940s [Mcculloch and Pitts, 1943], but have in many ways been developed independently since. Recent times have seen a resurgence of interest in linking the two.

At the physiological level, researchers have shown that ANNs can predict primates' brain activities better than classical models [Yamins and DiCarlo, 2016]. By examining the internal representations of the ANNs, it can also be shown that the features ANNs extract also align with those in the real brain [Pospisil et al., 2018]. At the more abstract psychological level, it has been shown that the learning dynamics of the ANNs mimic the semantic development learning in children [Saxe et al., 2019].

In this thesis, we extend the links between ANNs and brains, in the context of massive new mouse brain data sets. We will demonstrate that linking the ANNs to the mouse brain is both possible and fruitful.

1.3.1 Comparing ANNs with the mouse brain

To link the ANNs with real brain activities, the first step is to make meaningful quantitative comparisons between the two. Although ANNs and the mouse brain are quite different, they both can be seen as functional transformations acting upon the input data (Figure 1.2). The activities from the ANNs or the brains can then be constructed into "representation matrices" to analyze the properties of these transformations.

There are numerous metrics that can be used to compare those representation matrices [Diedrichsen and Kriegeskorte, 2017, Raghu et al., 2017]. However, the robustness of such comparisons are unclear, considering the limitation that the biological experiments, despite their large scale, still have only limited input stimuli and recorded neurons; this contrasts the complete information available on activity in ANNs. In Chapter 2, by using systematic comparisons between ANNs and themselves under data limited constraints, we demonstrate conditions when comparing ANNs with the data from the mouse visual cortex is expected to be robust, thus promising to shed light on our understanding of the underlying biological brain.

1.3.2 Adding biological constraints to ANNs

Despite their capabilities, ANNs are not precise biological analogues. To make ANNs closer to the real brain, three directions stand out: getting more realistic data sets, constructing more realistic architectures, and defining more realistic loss functions (tasks). Fortunately, with the availability of large scale tract tracing data sets, a new approach to the central of these directions is available. Thus equipped, we build ANN models of the mouse visual cortex that more closely resemble the underlying biology.

In Chapter 3, we build an ANN model for mouse visual cortex that is biologically constrained in detail, not only in terms of the network structure, but also the densities and other details of connections between specific groups of neurons. We call this model MouseNet. Equipped with the MouseNet model, we can address key questions about mesoscale brain

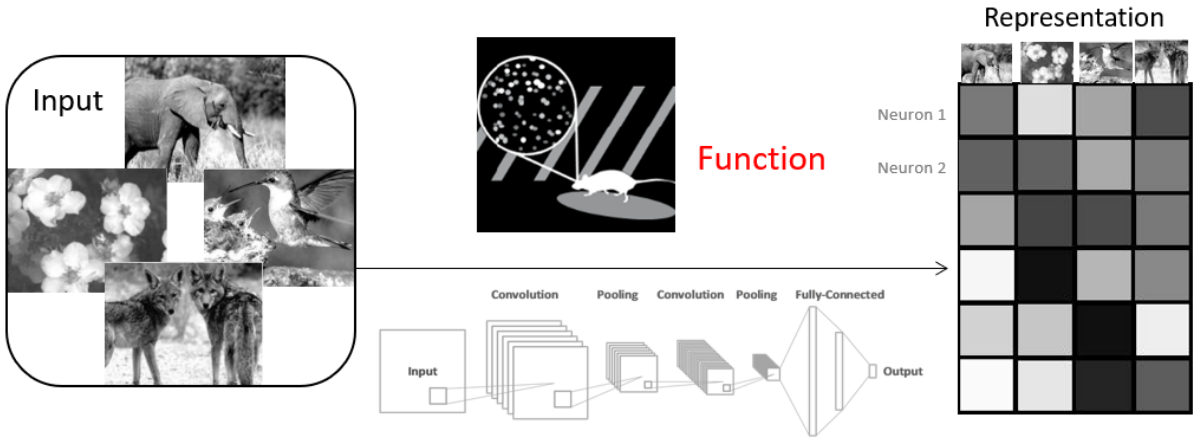

Figure 1.2: ANNs and mouse brain both act as functions acting on a common set of input stimuli. The output of these functions can be constructed into representation matrices. Figure adapted from Allen Brain Observatory [de Vries et al., 2020].

architecture and its role in task learning and performance. We ask, and provide a first set of answers, to: What is the performance of a mouse brain-sized – and mouse brain-structured – model on benchmark image classification tasks? How does the training of a network on this task affect the functional properties of specified layers within the biologically constrained architecture – both overall, and in comparison with recorded function of mouse neurons?

1.4 Using ANNs to understand the role of biological architectural design

In constructing the anatomically constrained MouseNet, we notice that it has a unique characteristic as opposed to common ANN models – that is, instead of just a single pathway, it has multiple parallel pathways. We have relatively little knowledge about the functional difference between these parallel pathways in the mouse and what information they learn to carry. Interestingly, however, in primates' visual cortex, there are also analogous parallel "ventral" and "dorsal" pathways about which we have more knowledge: the "ventral" pathway deals with object recognition (what) and the "dorsal" pathway deals with movement

information (where).

The role of these parallel pathways, as a common design in biological systems, invites a general analysis. In Chapter 5, we analyze the simplest possible ANN architecture with parallel pathways, to develop new analytical and numerical results on the learning behaviour within this architectural design.

Chapter 2

COMPARISON AGAINST TASK DRIVEN ARTIFICIAL NEURAL NETWORKS REVEALS FUNCTIONAL ORGANIZATION OF MOUSE VISUAL CORTEX

Partially inspired by features of computation in visual cortex, deep neural networks compute hierarchical representations of their inputs. While these networks have been highly successful in machine learning, it remains unclear to what extent they can aid our understanding of cortical function. Several groups have developed metrics that provide a quantitative comparison between representations computed by networks and representations measured in cortex. At the same time, neuroscience is well into an unprecedented phase of large-scale data collection, as evidenced by projects such as the Allen Brain Observatory. Despite the magnitude of these efforts, in a given experiment only a fraction of units are recorded, limiting the information available about the cortical representation. Moreover, only a finite number of stimuli can be shown to an animal over the course of a realistic experiment. These limitations raise the question of how and whether metrics that compare representations of deep networks are meaningful on these datasets. Here, we empirically quantify the capabilities and limitations of these metrics due to limited image presentations and neuron samples. We find that the comparison procedure is robust to different choices of stimulus set and the level of subsampling that one might expect in a large-scale brain survey with thousands of neurons. Using these results, we compare the representations measured in the Allen Brain Observatory in response to natural image presentations to deep neural networks. We show that the visual cortical areas are relatively high order representations (in that they map to deeper layers of convolutional neural networks). Furthermore, we see evidence of a broad, more parallel organization rather than a sequential hierarchy, with the primary area VISp

(V1) being lower order relative to the other areas.

2.1 Introduction

Deep neural networks, originally inspired in part by observations of function in visual cortex, have been highly successful in machine learning [Krizhevsky et al., 2012, Goodfellow et al., 2016, Simonyan and Zisserman, 2014], but it is less clear to what extent they can provide insight into cortical function. Using coarse-grained neural activity from fMRI and MEG, it has been shown that comparing against task-driven DNNs provides insights for functional organization of primates' brain areas [Güçlü and van Gerven, 2015, Cichy et al., 2016]. At the single-neuron level, it has been shown that deep neural networks with convolutional structure and a hierarchical architecture outperform simpler models in predicting single-neuron responses in primates' visual pathway [Cadieu et al., 2014, Yamins et al., 2014, Khaligh-Razavi and Kriegeskorte, 2014a, Yamins and DiCarlo, 2016].

To understand the overall structure and function of cortex, we require models that describe both the population representation as well as single cell properties. Artificial neural network models such as convolutional networks discard complexity in individual units (compared to real biological neurons) but provide a useful structure to model large-scale organization of cortex, e.g. by describing the progressive development of specific feature response through successive layers of computation. Conversely, given an artificial network, we can use its patterns of response as a "yardstick" to assess the nature and complexity of representations in real neural networks. Naturally, such an assessment requires a metric for comparing representations and a suitable model for comparison. Here we choose such models from the family of convolutional networks. We aim to assess the complexity and hierarchical structure of a real cortical system relative to a computational hierarchy originally inspired by biological response.

Additionally we must choose a metric. While there exist metrics in the literature to compare representations between models or networks, even the largest scale neuroscience experiments only record from a fraction of the population of neurons and limited imaging or

recording time implies that one can only cover a very small portion of stimulus space, raising the question of whether metrics that compare representations of deep networks to those of cortical neurons are meaningful. For example, the Allen Brain Observatory, despite being a massive dataset, includes only a small fraction of the neurons in the mouse visual cortex. Similarly, despite over three hours of imaging per experiment, only 118 unique natural images are shown due to the inclusion of a diverse array of stimulus types.

In this work, we empirically investigate the limitations imposed on representational comparison metrics due to limited presentations of stimuli and sampling of the space of units or neurons. Specifically, given a metric M that computes a similarity score between two representations, we choose a fiducial task-trained network X and ask about the robustness of mapping representations to depth in the network X as a measure of feature complexity, we call this the X-pseudo-depth for metric M, d_X^M, of the representation. In the following, we use VGG16 [Simonyan and Zisserman, 2014] to illustrate the idea. We provide results of some other network variants in the appendix. We use two metrics available in the literature, the similarity of similarity matrices (SSM) [Diedrichsen and Kriegeskorte, 2017] and singular value canonical correlation analysis (SVCCA) [Raghu et al., 2017, Morcos et al., 2018].

For both metrics, we compute the effect on VGG16-pseudo-depth and similarity score of the size of the image set and the number of units sub-sampled (as would happen, e.g. when a measurement precludes access to the entire population). We find that although the similarity score degrades with subsampling neurons, it can be well approximated with number of sampled neurons on the order of thousands. The pseudo-depth is also reasonably preserved with number of sampled neurons on the order of thousands.

Using these observations, we find that the data from the Allen Brain Observatory meets criteria that allow us to use the model VGG16 as a comparison model to assess functional organization and feature complexity via the similarity score and VGG16-pseudo-depth. We find that all regions of mouse visual cortex have the pseudo-depth close to the midpoint of the network, indicating that the representations as a whole are higher-order than the "simple" type of cell responses that are typically used to describe early visual layers. The

primary area VISp (also called V1) is of consistently lower VGG16-pseudo-depth than other layers, while the higher visual areas have no clear ordering, suggesting the fact that mouse visual cortex is organized in a broader, more parallel structure, a finding consistent with anatomical results [Zhuang et al., 2017b]. VISam and VISrl have such low similarity scores that this may suggest an alternative function, i.e. a network trained on another task may yield more similar features.

2.2 Methodology

Problem Formalization and definitions Define a "representation matrix" of a system X, $R_X \in \mathbb{R}^{n \times m}$, to be the set of responses of m units or neurons to n images. Choosing a set of images, we choose a model network and a similarity metric $M \in \{SSM, SVCCA\}$ and compute the **VGG16-pseudo-depth** as $d^M_{VGG16} = \text{argmax}_{i \in \text{layers of VGG16}} M(R_X, R_{VGG16_i})$. We use d^* as short hand notation for d^M_{VGG16}, and compute the corresponding **similarity score**, as $s^* = M(R_X, R_{VGG16_{d^*}})$. Our goal is to investigate the stability of d^* and similarity score s^* under subsampling of neuron number n and both the number of images m and which images are shown, and to use these quantities to study representations across different mouse cortical areas. We also provide additional results about other model variants in the appendix.

The Allen Brain Observatory data set The Allen Brain Observatory data set [de Vries et al., 2020] is a large-scale standardized *in vivo* survey of physiological activity in the mouse visual cortex, featuring representations of visually evoked calcium responses from GCaMP6f-expressing neurons. It includes cortical activity from nearly 60,000 neurons collected from 6 visual areas, 4 layers, and 12 transgenic mouse Cre lines from 243 adult mice, in response to a range of of visual stimuli.

In this work, we use the population neural responses to natural image stimuli, which contains 118 natural images selected from three different databases (Berkeley Segmentation Dataset [Martin et al., 2001], van Hateren Natural Image Dataset [van Hateren and van der Schaaf, 1998] and McGill Calibrated Colour Image Database [Olmos and Kingdom, 2004]).

The images were presented for 250 ms each, with no inter-image delay or intervening "gray" image. The neural responses we use are events detected from $\Delta F/F$ using an L0 regularized deconvolution algorithm, which deconvolves pointwise events assuming a linear calcium response for each event and penalizes the total number of events included in the trace [Jewell and Witten, 2018, Jewell et al., 2019]. Full information about the experiment is given in [de Vries et al., 2020].

Representation matrices for mouse visual cortical areas To construct the representation matrix for a certain mouse visual cortical area, we take the trial-averaged mean responses of the neurons in the first 500ms upon image presentation. We group activities of neurons in different experiments for the same brain area and construct the representation matrix. Note that for the Allen observatory dataset, the number of images (118) is much less than the number of observed neurons.

Representation matrices for DNN layers Unless explicitly stated, the representation matrices for DNN layers are obtained from feeding the same set of 118 images (resized to 64 by 64, see section 4.3 below) to the DNN and collecting all the activations from a certain layer.

Two similarity metrics for comparing representation matrices We investigate two metrics suitable for comparing representation matrices with $n << m$, i.e., many fewer images than neurons. One is similarity of similarity matrices (SSM) [Diedrichsen and Kriegeskorte, 2017]. Another is an extension of the recently developed singular value canonical correlation analysis (SVCCA) [Raghu et al., 2017, Morcos et al., 2018] to the $n << m$ regime.

For the SSM metric, we calculate the Pearson correlation coefficient between every pair of rows in one representation matrix to get a size n by n "similarity matrix" where each entry describes the similarity of the response to two images. Importantly, this collapses the data along the neuron dimensions, so that representations with different numbers of neurons can be compared. To compare the two similarity matrices, we flatten the matrices to vectors

and compute the Spearman rank correlation of their elements. Like the Pearson correlation coefficient, the rank correlation lies in the range $[-1, 1]$ indicating how similar (close to 1) or dissimilar (close to -1) the two representations are.

Following the established approaches [Raghu et al., 2017], we first run singular value decomposition (SVD) to reduce the neuron dimension to a fixed number r which is smaller than the dimension of both representations. We fix r to be the most important (largest variance) 40 dimensions for each representation. We then perform a canonical correlation analysis (CCA) on the reduced representation matrices. CCA compares two representation matrices by sequentially finding orthogonal directions along which the two representations are most correlated. We can then read out the strength of similarity by looking at the values of the corresponding correlation coefficients. We take the mean of the r correlation coefficients resulting from CCA as the SVCCA similarity value.

Note that SVCCA is invariant to invertible linear transformations of the representations. SSM is invariant to transformations of representations that induce monotonic transformations of the elements of similarity matrices. An excellent review of similarity metrics and their properties can be found in [Kornblith et al., 2019].

2.3 Robustness of estimates of similarity score and pseudo-depth to subsampling of images and neural units

In this section, we study the robustness of VGG16-pseudo-depth and similarity score estimates in the face of limited stimuli and limited access to neurons in the representation of interest. Recall that we have full access to all neurons in the pretrained VGG network [Simonyan and Zisserman, 2014] that we are using as a "yardstick". We begin with the simplest possible setting: using this yardstick to measure VGG16-pseudo-depth and similarity score of another copy of VGG16, but for which we observe only a random subsample of units (neurons).

We will show that (1) the similarity scores are robust to including only the 118 images in the Allen brain observatory data set, as well as the specific images within this set, and

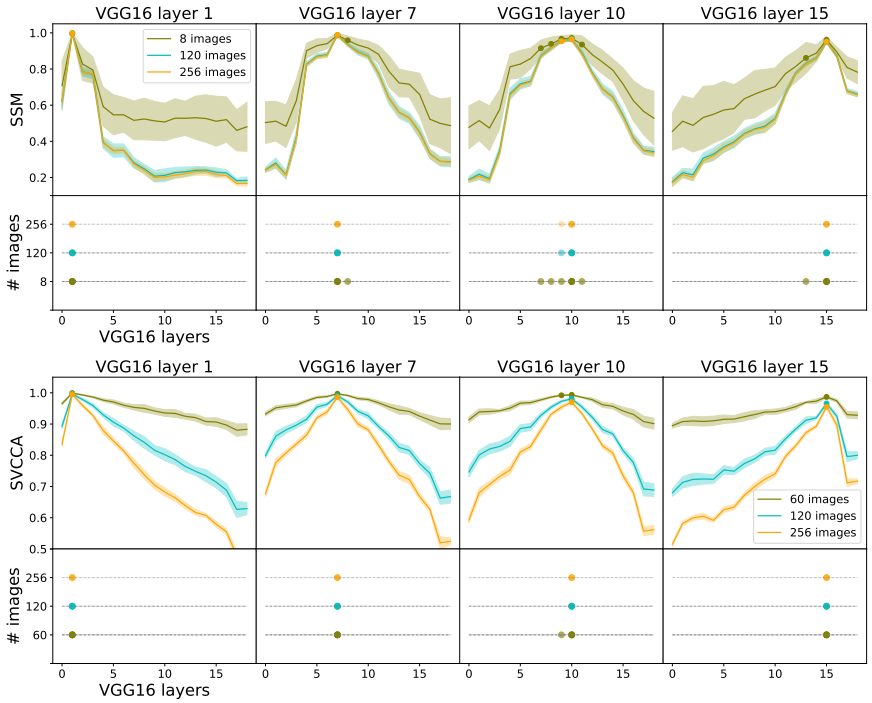

Figure 2.1: Testing the self-consistency of d^* by varying the number of images included in the dataset. Shown are SSM (top) and SVCCA (bottom) d^* computed for several layers of VGG16 (1, 7, 10, 15 from left to right) using different numbers of stimuli from tiny ImageNet. The shaded areas denote the standard deviation computed from different randomly chosen sets of images. The shaded circles denote the layers indistinguishable from d^* (highlighted).

(2) the similarity scores decrease with neuron subsampling, whereas the pseudo-depth stays constant given enough neurons.

2.3.1 VGG-pseudo-depth and similarity scores can be estimated stably with limited image sets

The Allen Brain Observatory dataset includes neural responses to 118 natural image stimuli. We first study how the number of stimuli influences estimates of VGG16-pseudo-depth and similarity score, and how much variation arises when we present different sets of images.

For this, we randomly select different numbers of images from tiny ImageNet [Le and Yang, 2015] and calculate the similarity values between VGG16 model layers. The results for four representative layers are shown in Figure 2.1. We see that the VGG16-pseudo-depth identifies the corresponding layer that is chosen for comparison, and the similarity score is always one for the corresponding layer given different number of randomly chosen images. In addition, the variance introduced by the random choices of images is small for 120 images. Thus the metrics are robust to different choices of stimulus set, including the image set used in the Allen Brain Observatory.

Note that the sharpness of the peak of the similarity curve represents how precisely the metric can find the best layer. We see that the sharpness of the peak of SSM curve does not further improve when more than 120 images are shown (approximately the number presented in the biological data set), while the sharpness of the SVCCA curve can still improve if we add more images to the data set.

2.3.2 VGG-pseudo-depth and similarity scores can be estimated stably with sufficient subsampling of neuronal populations

In biological experimental settings, we only observe a small portion of neurons from a brain area. Here, we investigate how this affects our ability to reliably use the VGG network to estimate pseudo-depth and similarity scores. Recalling that the network that we use as a yardstick can be completely observed, we take a sub-subsampled population from a certain layer in VGG16, and compare it to the whole population of VGG16 layers. The results for four representative layers are given in Figure 2.2. This shows that the similarity scores are severely reduced by subsampling. As we increase the number of neurons, the similarity curves also rise, reaching values with 2000 neurons that are close to those with no subsampling. Thus, at least for comparing the VGG model with a partially observed version of itself, a rule of thumb is that if including at least 2000 neurons in the sampled population, then the similarity score is a good approximation to those that would be found from observing the whole population.

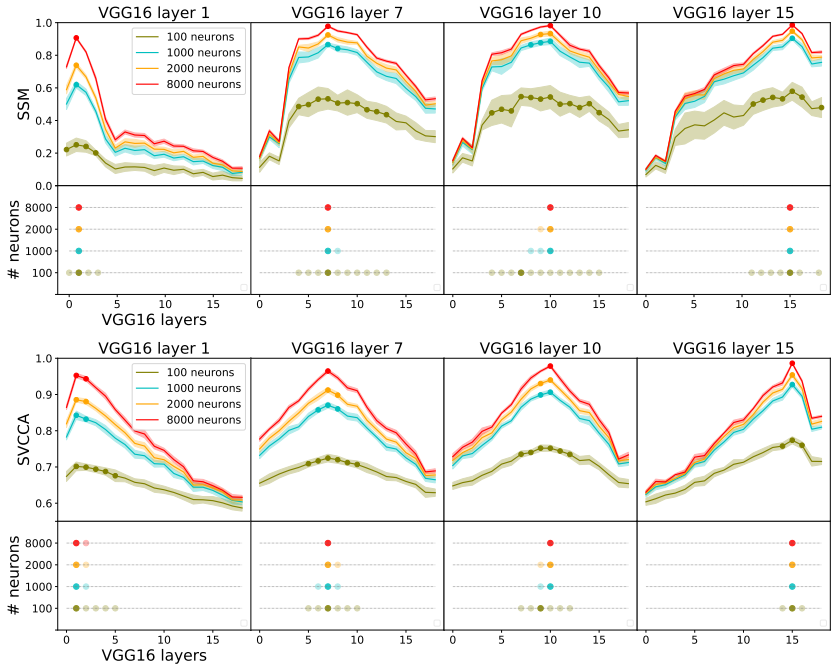

Figure 2.2: Testing the self-consistency of d^* by varying the number of units subsampled. Shown for SSM (top) and SVCCA (bottom) is d^* computed for several layers of VGG16 (1, 7, 10, 15 from left to right). The shaded areas denote the standard deviation computed from different random draws of sub-samples.

The relative order of similarity values across layers is consistent for a wider range of the number of sampled neurons. Even with less than 2000 neurons sampled, say 1000, we can already find which layers are more similar to the population of interest. Thus, the corresponding rule of thumb for VGG16-pseudo-depth is that around 1000 neurons must be sampled for it to be consistently estimated.

2.3.3 Robustness of similarity score and pseudo-depth extend to a different network

To see whether the approaches above remain robust when comparing representations from a different network against representations generated by VGG16, we choose neurons from 4 layers of VGG19 and compare them with entire layers of VGG16. The results are given

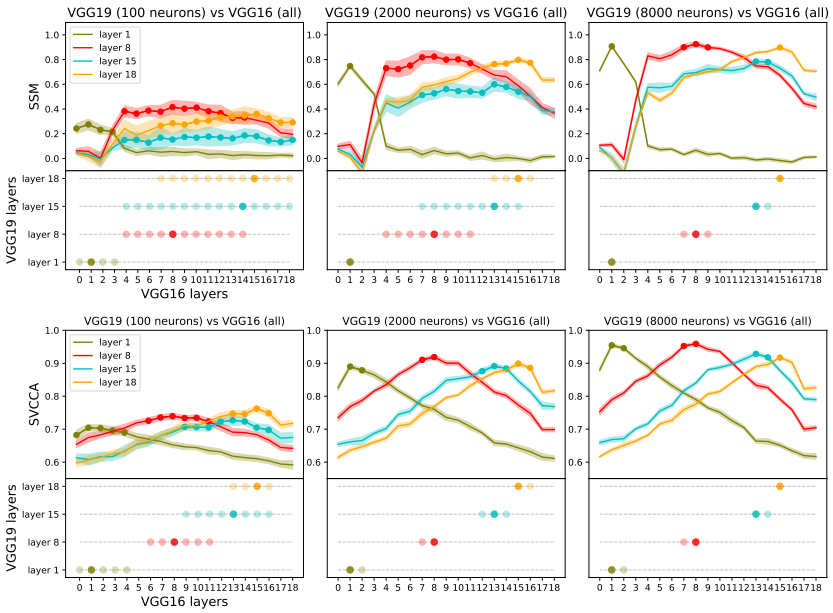

Figure 2.3: d^* computed on the layers of VGG19. d^* is relatively consistent across with large numbers of sub-sampled units. Shown for SSM (top) and SVCCA (bottom) is d^* for the layers of VGG19 with different numbers of sub-sampled units (left to right: 100, 2000, 8000).

in Figure 2.3. We see that the curves with 2000 neurons are very close to the ones with 8000 neurons, suggesting that this remains an adequate level of sampling when comparing between these two networks. Moreover, our metrics show that early layers in VGG19 are more similar to early layers in VGG16, and later layers in VGG19 are more similar to later layers in VGG16, as we would expect intuitively, reflecting the functional hierarchy of the four VGG19 layers based on VGG16-pseudo-depth estimated from 2000 neurons.

2.4 VGG16-pseudo-depth and similarity scores for mouse cortex and interpretations for the visual hierarchy

In this section, we compare mouse visual cortex representations against VGG16 and discuss the resulting insights for the mouse visual hierarchy. In the Allen brain observatory data set,

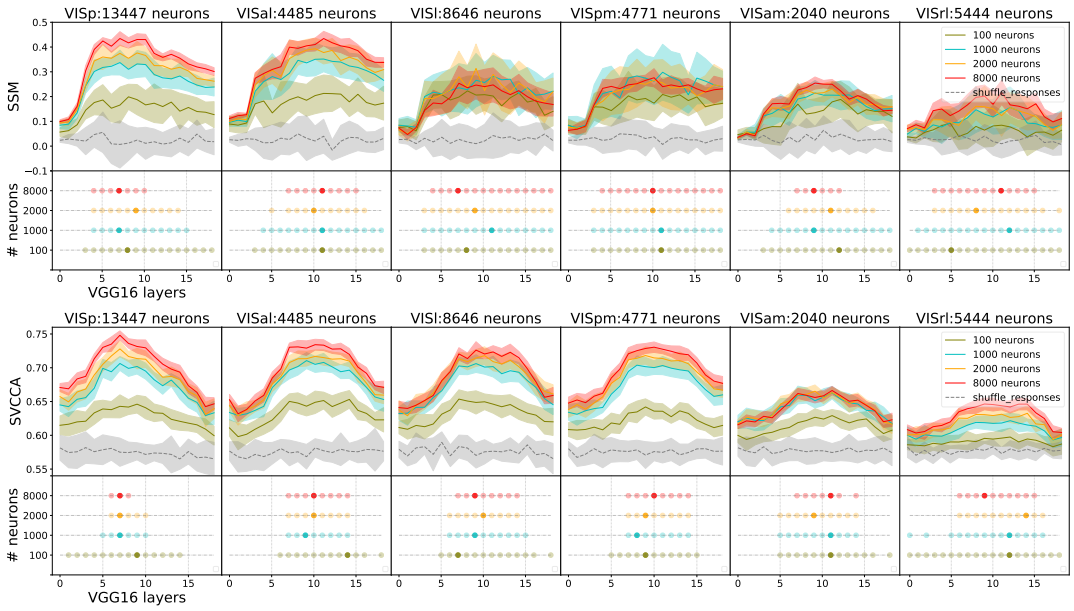

Figure 2.4: d^* computed for representations from the Allen Brain Observatory, shows a relatively broad, parallel structure, rather than a strict hierarchy, although VISp is of lower d^* than the other areas. Shown for SSM (top) and SVCCA (bottom) is d^* for the Allen Brain Observatory. The dashed gray curve is comparing the whole population to VGG16 with responses shuffled. The shaded areas denote the standard deviation computed from different random draws of sub-samples. The shaded circles denote the layers indistinguishable from d^* (highlighted).

each neuron belongs to a specific visual area (VISp, VISl, VISal, VISpm, VISal, VISam), cortical layer (layer23, layer4, layer5, layer6) and has a specific cell type (Cre-line). By grouping neurons in different areas, cortical layer or cell types, we can study the functional properties of the specific neuron groups. In the following, we separately compare VGG16 with entire cortical areas (Figure 2.4), distinct cortical layers in the same area (Figure 2.6), and distinct cell types in the same area (Figure 2.7).

2.4.1 Whole brain area comparisons show functional properties for mouse visual cortical areas

To study visual representations within and across whole brain areas, we group all the neurons in the same visual area and compare all six areas in our data set to VGG16. Note that different areas have different total numbers of recorded neurons available. In order to make fair comparisons across areas, each time we compute a similarity curve we sample the same number of neurons with replacement from each area. As always, we compare representations in the sub-sampled brain area to representations in all neurons in the VGG16 layers that we are using as our yardstick. The results are shown in Figure 2.4. To give a baseline for these comparisons, we shuffled the rows of the representation matrices and calculate the similarity curves for it (dashed gray curves).

Similarity curves computed using both SSM and SVCCA metrics show that:

1. The pseudo-depth for the mouse brain areas corresponds to the middle layers of VGG16. This suggests that mouse visual cortical representations are *higher-order*, involving multiple stages of processing.

2. The pseudo-depth of VISp is lower than that of other brain areas, a fact that is partially but not completely attributable due to its receptive field size (see section 4.3 below). Meanwhile, the higher visual areas have no clear ordering. This suggests that, following initial stages of processing after VISp, mouse visual cortex is organized in a broadly parallel structure, as apposed to a hierarchical one.

3. VISam and VISrl have the lowest similarity scores among all brain areas, according to both metrics. Based on our studies in Section 3 (Figure 2.2) that suggest sub-samples of 2000 neurons are sufficient to approximate similarity scores, this indicates that VISam and VISrl are less similar overall to VGG16 than the other areas. A natural hypothesis is that VISam and VISrl perform a different type of processing – one that demands

visual features that are more distinct from those required to classify the large set of categories used to train VGG16.

In addition to these principal observations that are common to both SSM and SVCCA metrics, we note that these metrics do show some different properties when applied to brain areas VISl and VISpm. Specifically, SSM produces a relatively larger variance in similarity curves across subsamples of VISl and VISpm neurons, and as a consequence a broader range of possible pseudo-depths for these areas. We leave investigating the cause and possible interpretations of such differences to future work. We also used different input image resolutions to do the comparison. This shows our main conclusions remain valid, but increasing image resolution causes the pseudo-depth to shift to the right, which suggests that the pseudo-depth could be associated with receptive field size (Section 2.4.3). To numerically quantify the effects of trial-to-trial variability, we repeated the calculation of the SSM value as in Figure 2.4 by bootstrapping across trials (Figure 2.5). The results show that our main conclusions are robust to trial-to-trial variability.

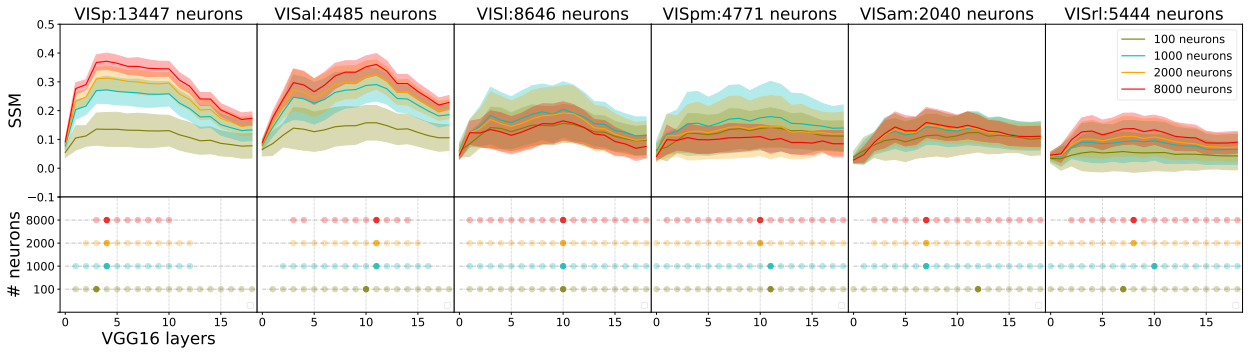

Figure 2.5: Same comparison as in Figure 2.4, with shaded area showing standard deviation from simultaneously performing bootstrap resampling for trials' mean responses and random draws of different sub-samples of neurons.

2.4.2 Cortical layer and cell-type subpopulations show similar trends but can have higher similarity scores than brain areas taken as a whole

How do the trends for pseudo-depth and similarity scores that we have identified above depend on the fact that we have grouped together neurons across type and cortical depth (*cortical* layer) into 'whole" areas? To answer this question, we separate neurons from the same brain area according to their cortical layer, Figure 2.6, and genetically encoded cell line (a coarse measure of cell type), Figure 2.7. In producing the resulting similarity scores we sample 2000 neurons with replacement from each subpopulation of mouse neurons. Note that these subpopulations have less than 2000 neurons in general, so that resampling is significant; in Figure 2.7, we only show the results for cell types with more than 900 neurons.

We find that SVCCA reveals the same basic trends in similarity curves when brain areas are divided into subpopulations as for the whole area comparisons in Section 2.4.1. The SSM metric produces curves that are suggestive of some possible differences. For example, for the whole area comparisons, we see that the SSM curves values for VISl and VISpm have lower mean and larger variance compared to those for VISp and VISal. However, when their subpopulations are considered separately, there are some cortical layers (layer23 of VISl and layer5 of VISpm) and cell types (Slc17a7 of VISpm) that have higher SSM similarity scores than their areas as a whole. This suggests that these cortical layers and cell types may, taken as components of a larger system, represent visual features that are in fact more similar to those extracted by VGG16.

2.4.3 Impact of image resolution on VGG pseudo-depth

A natural question about our conclusions about pseudo-depth above is whether they are an automatic consequence of the image resolution (sometimes referred to as the receptive field size) that occurs at different stages through both the VGG network and the mouse brain – in other words, whether they simply follow from matching the resolution in a given VGG layer with that in a given mouse brain area, rather than matching their complexity.

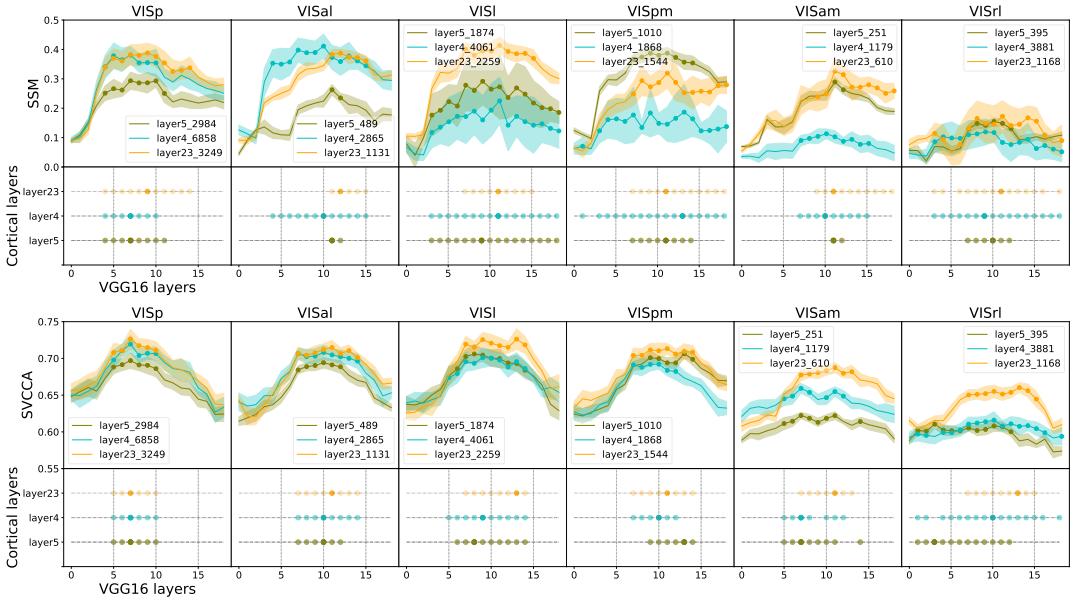

Figure 2.6: Separate cortical layer comparisons. SSM (top) and SVCCA (bottom) result for comparing different cortical layers in the same area to VGG16.

To address this, we first note that we have chosen our input images to be downsampled to a very limited size (64 by 64) that roughly corresponds to the limited visual acuity of the mouse [Prusky et al., 2000]. Thus we do not believe that our overall finding that the VGG-pseudo-depth of mouse visual brain areas corresponds to the middle layers of VGG is an automatic consequence of needing to look sufficiently deep into the VGG network for receptive field sizes that are as large as those in the mouse visual system. In Figure 2.8, we further test this by recomputing similarity curves for the VGG network responding to images with both substantially lower (resized input images to 32 by 32) and higher (128 by 128) resolution. We find that there is little effect of this input resolution for SSM pseudo-depth. Moreover, while SVCCA pseudo-depth is impacted by input resolution, pseudo-depths remain in the middle layers of SVCCA even when the input resolution is doubled or halved. Based on this we conclude that our result that the pseudo-depth of mouse visual cortex corresponds to the middle layers of VGG16 is robust to reasonable assumptions about the

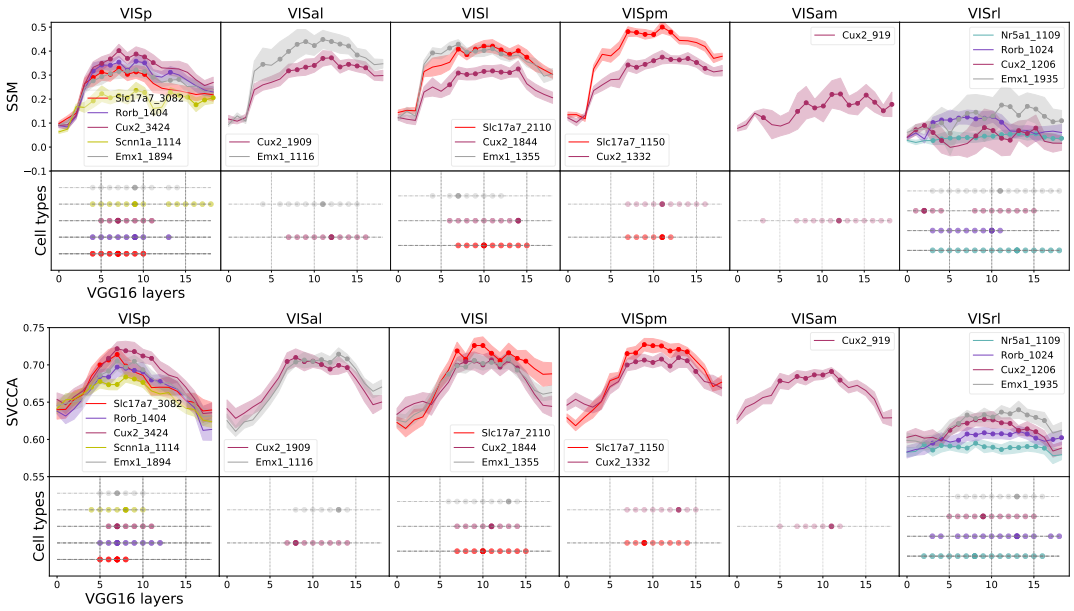

Figure 2.7: Separate cell type comparisons. SSM (top) and SVCCA (bottom) result for comparing different cell types in the same area to VGG16. Only cell types with more than 900 neurons are shown.

visual resolution. However, conclusions about the *relative* depth of visual areas could still be impacted by the resolution issue. For example, area VISp is known [de Vries et al., 2020] to have smaller receptive fields than other mouse visual cortical areas. Thus, the fact that SVCCA (but not SSM) pseudo-depths are earlier for VISp than other areas could be due to the resolution effects, rather than the level of complexity of its representations. We note a final possible limitation in interpreting our results. The VGG16 network that we use as a yardstick was pretrained on high resolution visual inputs. It is an interesting and open question whether our findings would be the same for a network retrained with the lower resolution inputs which we use and describe above.

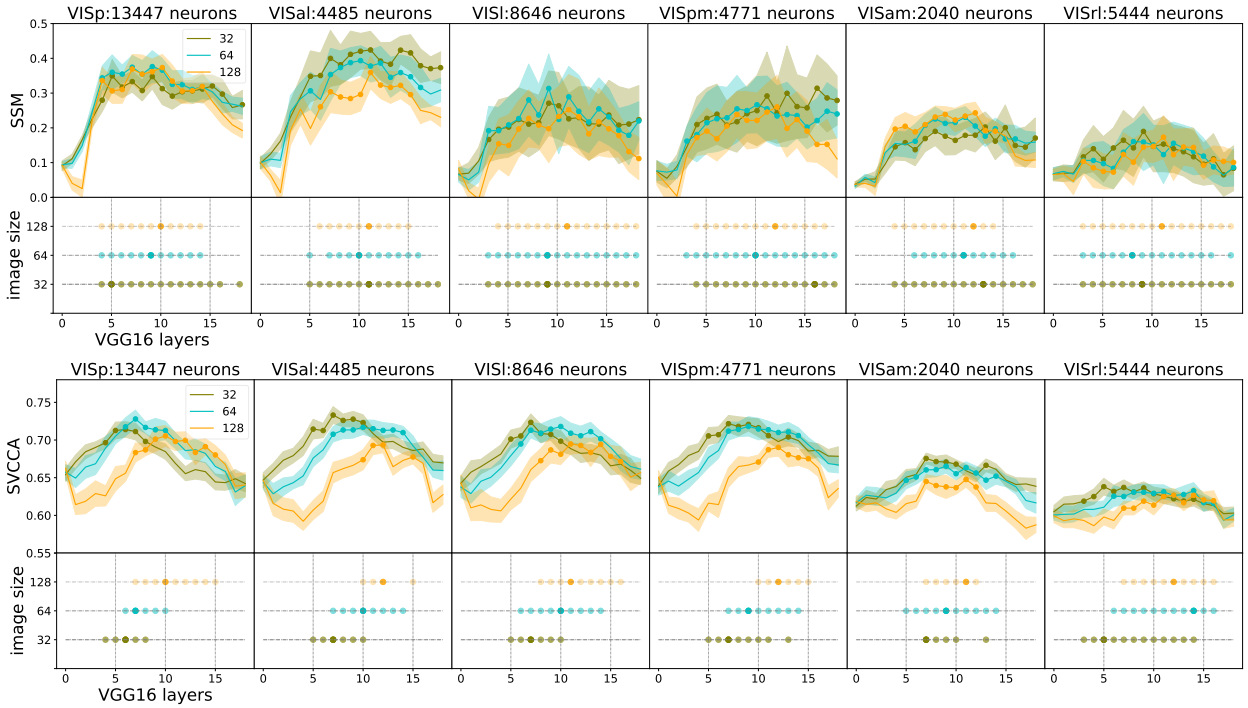

Figure 2.8: SSM (top) and SVCCA (bottom) result for comparing mouse visual cortical areas with 2000 sample neurons to VGG16 with different resizing (32x32, 64x64, 128x128) of input images. The shaded areas are the standard deviation computed from different random draws of sub-samples. The shaded circles denote the layers indistinguishable from the layer with highest similarity (highlighted).

2.5 Conclusion

Deep artificial neural networks can now produce task behavior that rivals the performance of biological brains in many settings. This opens the door to a fascinating question: what is similar, and what is different, in the way in which artificial and biological networks solve the underlying tasks [Yamins and DiCarlo, 2016, Hénaff et al., 2019]. A natural place to start is in comparing the stimulus representations that each produces.

Our first goal was to assess the robustness of this comparison to unavoidable challenges:

the set of stimuli, and number of neurons, that can be probed in biological experiments is necessarily limited. Our empirical results show that pseudo-depth and similarity scores are indeed robust to choices of stimuli on the order of hundreds and subsampling of neurons on the order of thousands.

Our second goal was to use this comparison to investigate visual representations in the mouse visual cortex, a system of explosively increasing interest in the neuroscience community and for which curated, massive public data sets on visual representations are now available [de Vries et al., 2020]. Functionally, very little is known about the visual areas in mice, compared with the primate visual cortex. This said, anatomical studies are developing the inter-area wiring diagram ([Harris et al., 2018]), and functional studies have provided evidence of some specialization across areas in terms of spatial and temporal frequency processing (e.g. [Andermann et al., 2011, Marshel et al., 2011]). Our results with data from the Allen Brain Observatory data set show that, according to VGG pseudo-depth and similarity scores, mouse visual cortical areas are relatively high order representations in a broad, more parallel organization rather than a sequential hierarchy, with the primary area VISp being lower order relative to the other areas. This is consistent with the relatively flat hierarchy observed in [Harris et al., 2018]. This approach and finding invites future insights from other artificial network systems, e.g. recurrent networks, and helps open doors for analyzing emerging large-scale datasets across species and tasks.

Acknowledgements

We thank Tianqi Chen, Saskia de Vries, Michael Oliver for helpful discussions, and Rich Pang, Gabrielle Gutierrez for comments on the draft. We thank the Allen Institute for Brain Science founder, Paul G. Allen, for his vision, encouragement, and support. We acknowledge the NIH Graduate training grant in neural computation and engineering (R90DA033461).

2.6 Appendix

In the following, we repeat all the experiments in the main text for AlexNet, ResNet18, and two Pytorch VGG16 models (VGG16a and VGG16b) trained from different initializations. The VGG16 model in the main text is a pre-trained Tensorflow model, which has a different preprocessing scheme from the Pytorch model. The type of each numbered layer, for each model, is given in Table 2.1. These results show that our main conclusions are preserved by models with different architectures or initializations.

Table 2.1: Specification of layers for each network model.

	VGG16	VGG19	ResNet18	AlexNet
0	Input	Input	Input	Input
1	Conv	Conv	bn1	Conv
2	Conv	Conv	Maxpool	Maxpool
3	Maxpool	Maxpool	layer1.0.bn1	Conv
4	Conv	Conv	layer1.0.bn2	Maxpool
5	Conv	Conv	layer1.1.bn1	Conv
6	Maxpool	Maxpool	layer1.1.bn2	Conv
7	Conv	Conv	layer2.0.bn1	Conv
8	Conv	Conv	layer2.0.bn2	Maxpool
9	Conv	Conv	layer2.1.bn1	
10	Maxpool	Conv	layer2.1.bn2	
11	Conv	Maxpool	layer3.0.bn1	
12	Conv	Conv	layer3.0.bn2	
13	Conv	Conv	layer3.1.bn1	
14	Maxpool	Conv	layer3.1.bn2	
15	Conv	Conv	layer4.0.bn1	
16	Conv	Maxpool	layer4.0.bn2	
17	Conv	Conv	layer4.1.bn1	
18	Maxpool	Conv	layer4.1.bn2	
19		Conv	Avgpool	
20		Conv		
21		Maxpool		

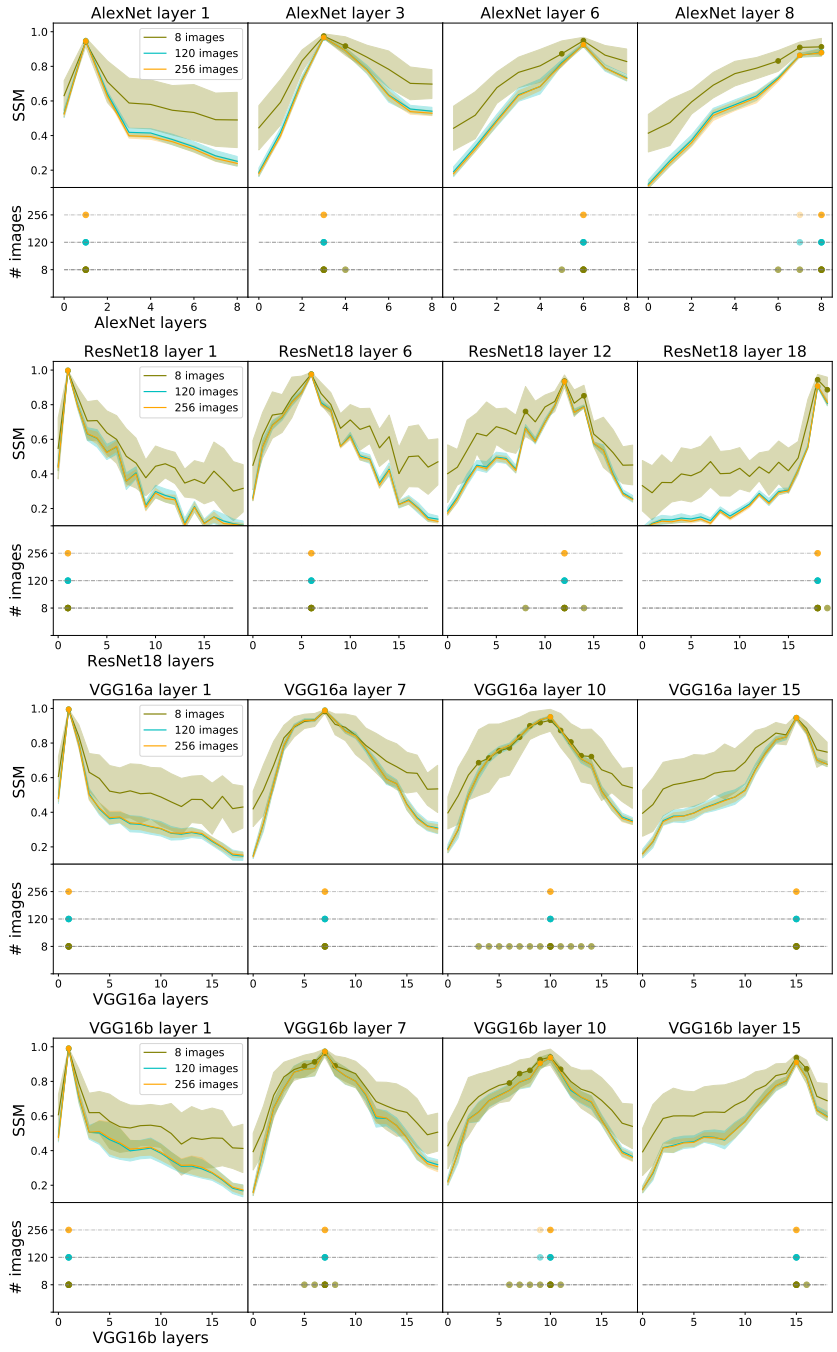

Figure 2.9: Same experiments as Figure 2.1 for AlexNet, ResNet18, VGG16a, VGG16b (top to bottom). These results show that, as for the VGG16 model in the main text, 120 images is generally an adequate number to identify layers via SSM for the other models as well.

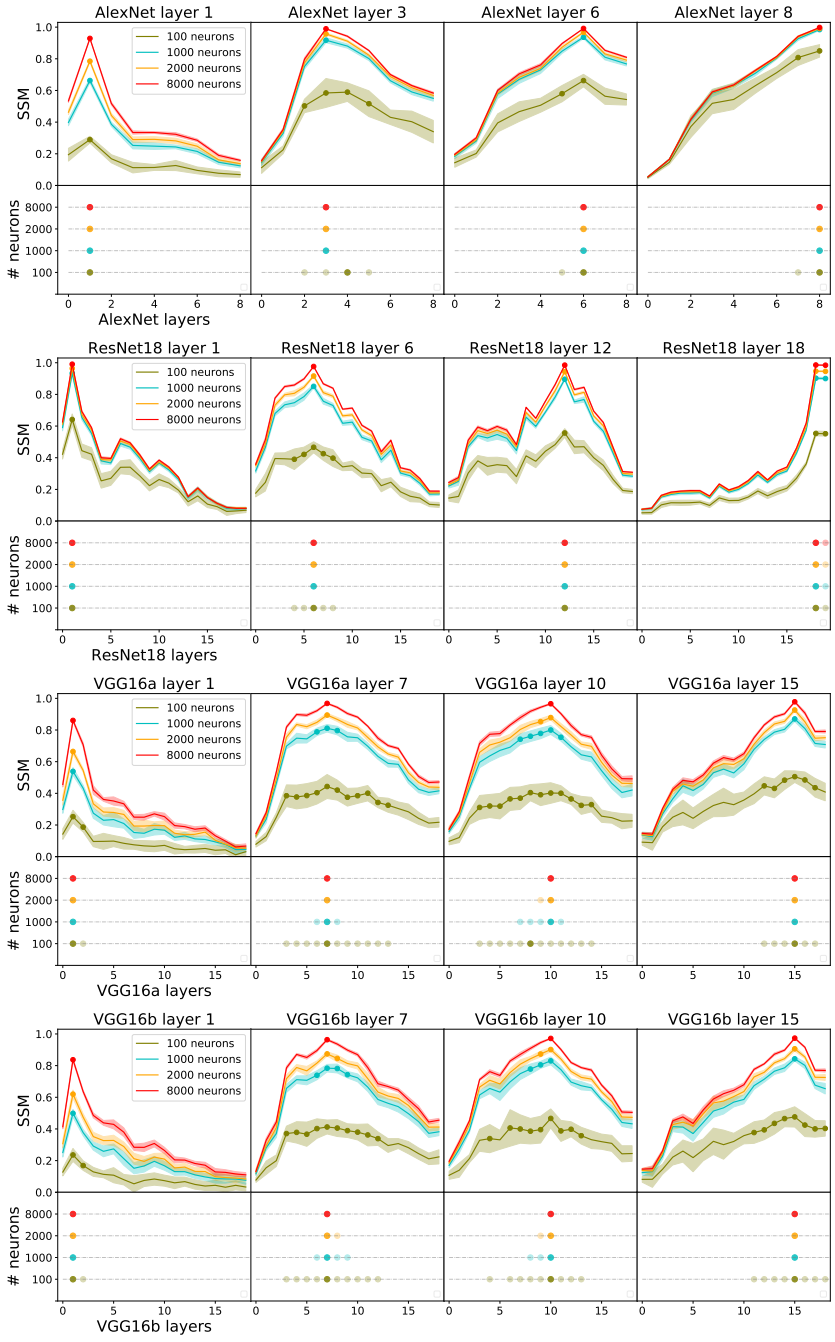

Figure 2.10: Same experiments as Figure 2.2 for AlexNet, ResNet18, VGG16a, VGG16b (top to bottom). These results show that, as for the VGG16 model in the main text, subsampling neurons on the order of thousands will give a good approximation of SSM similarity values.

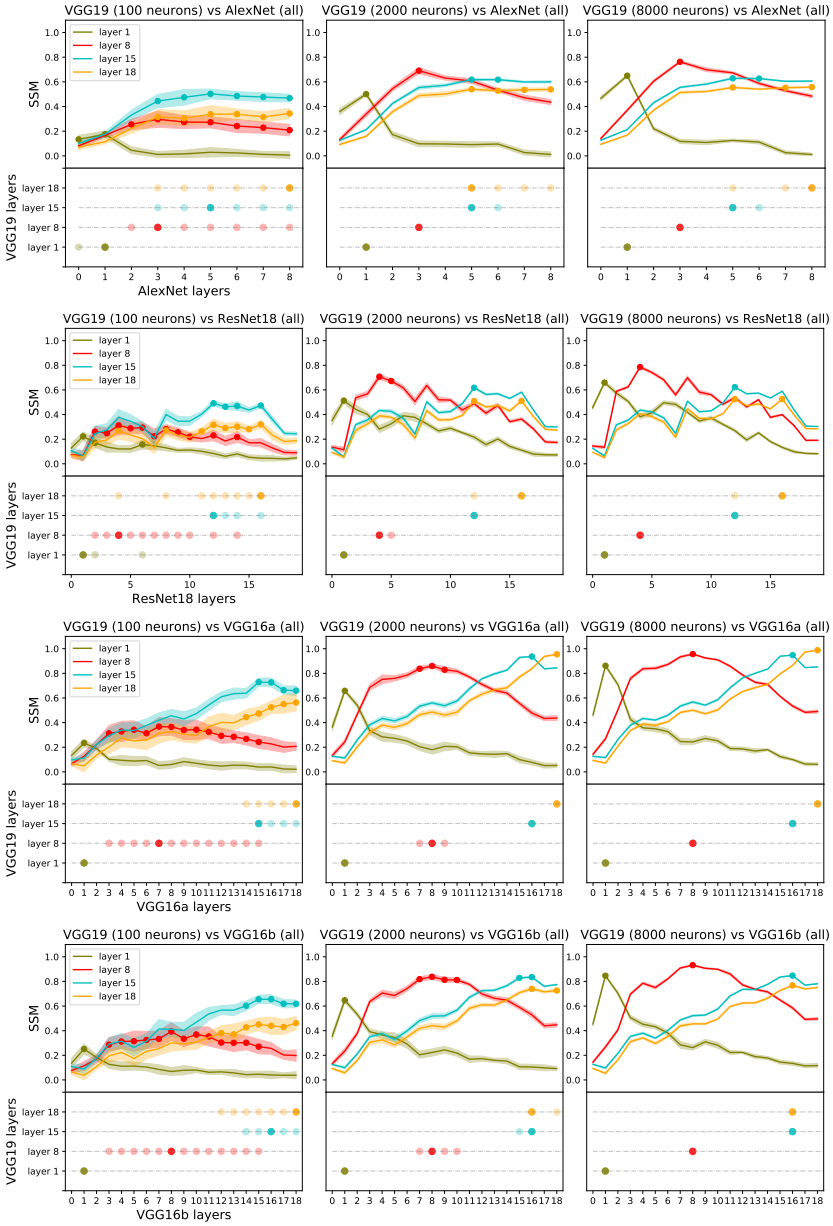

Figure 2.11: Same experiments as Figure 2.3 for AlexNet, ResNet18, VGG16a, VGG16b (top to bottom). These results show that, as for the VGG16 model in the main text, other models can be used as yardsticks to differentiate VGG19 layers via SSM as well.

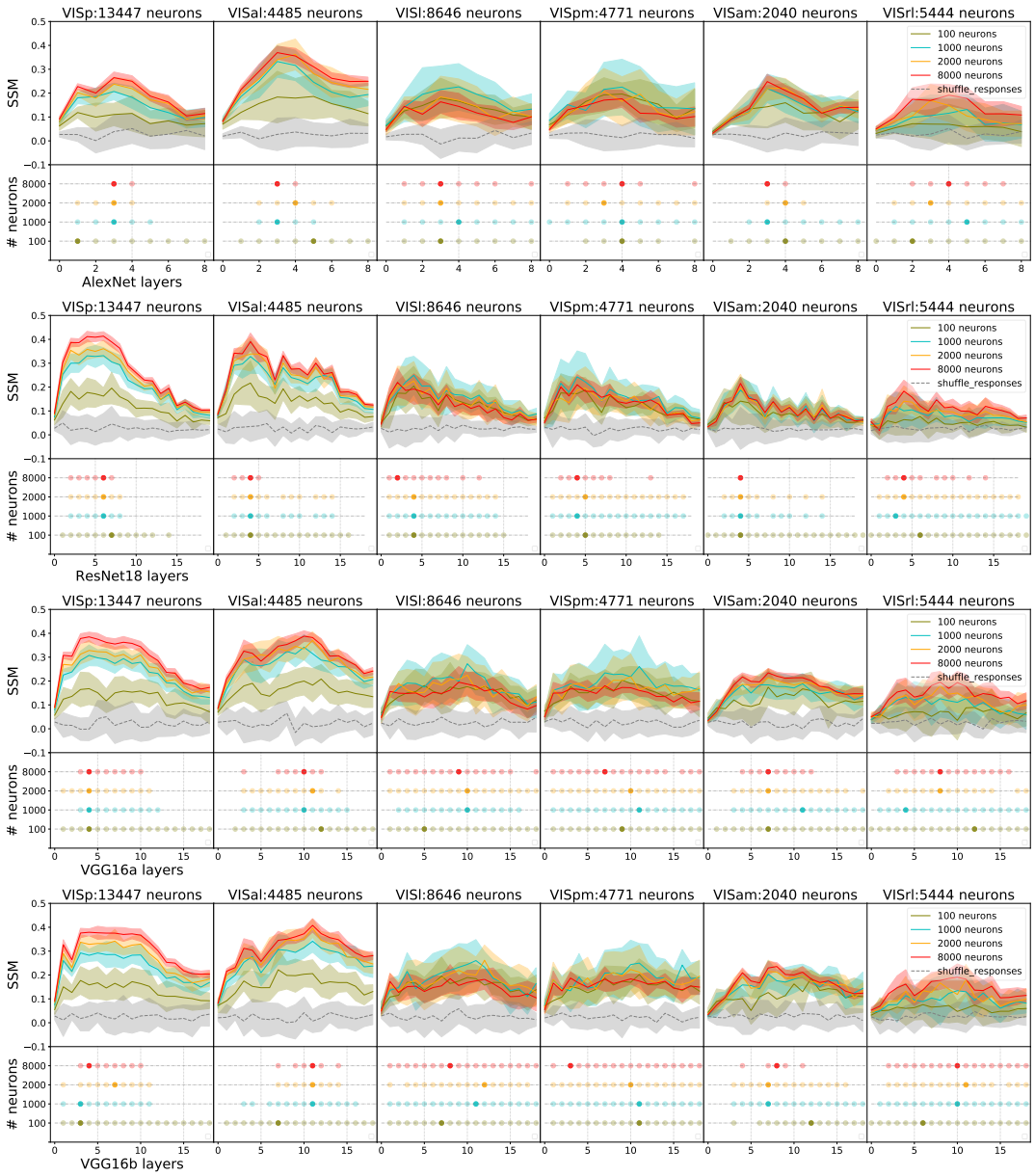

Figure 2.12: Same experiments as Figure 2.4 for AlexNet, ResNet18, VGG16a, VGG16b (top to bottom). These results show that our main conclusions about mouse visual cortex are qualitatively preserved by different models.

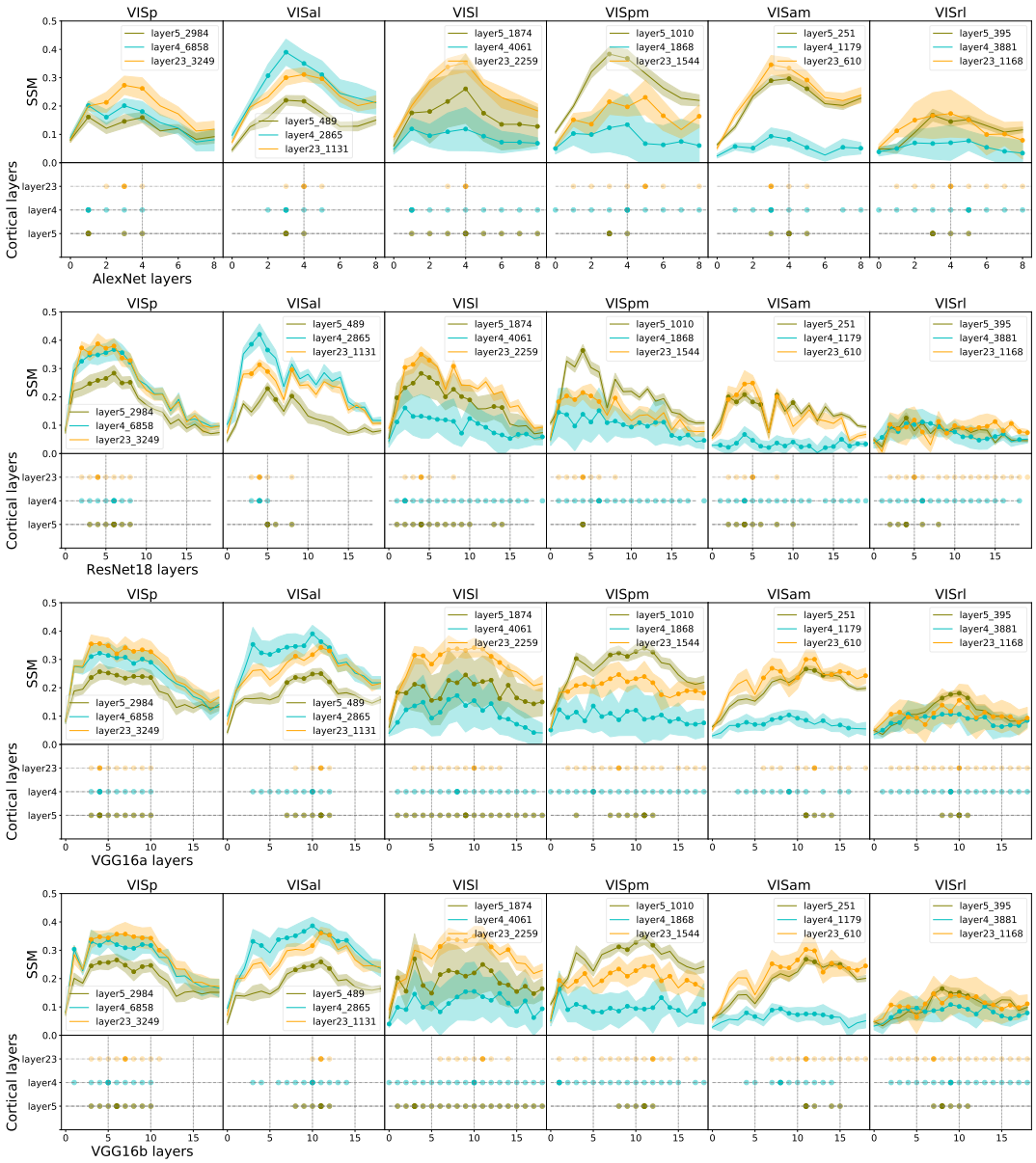

Figure 2.13: Same experiments as Figure 2.6 for AlexNet, ResNet18, VGG16a, VGG16b (top to bottom). These results show that our main conclusions about mouse visual cortex are qualitatively preserved by different models.

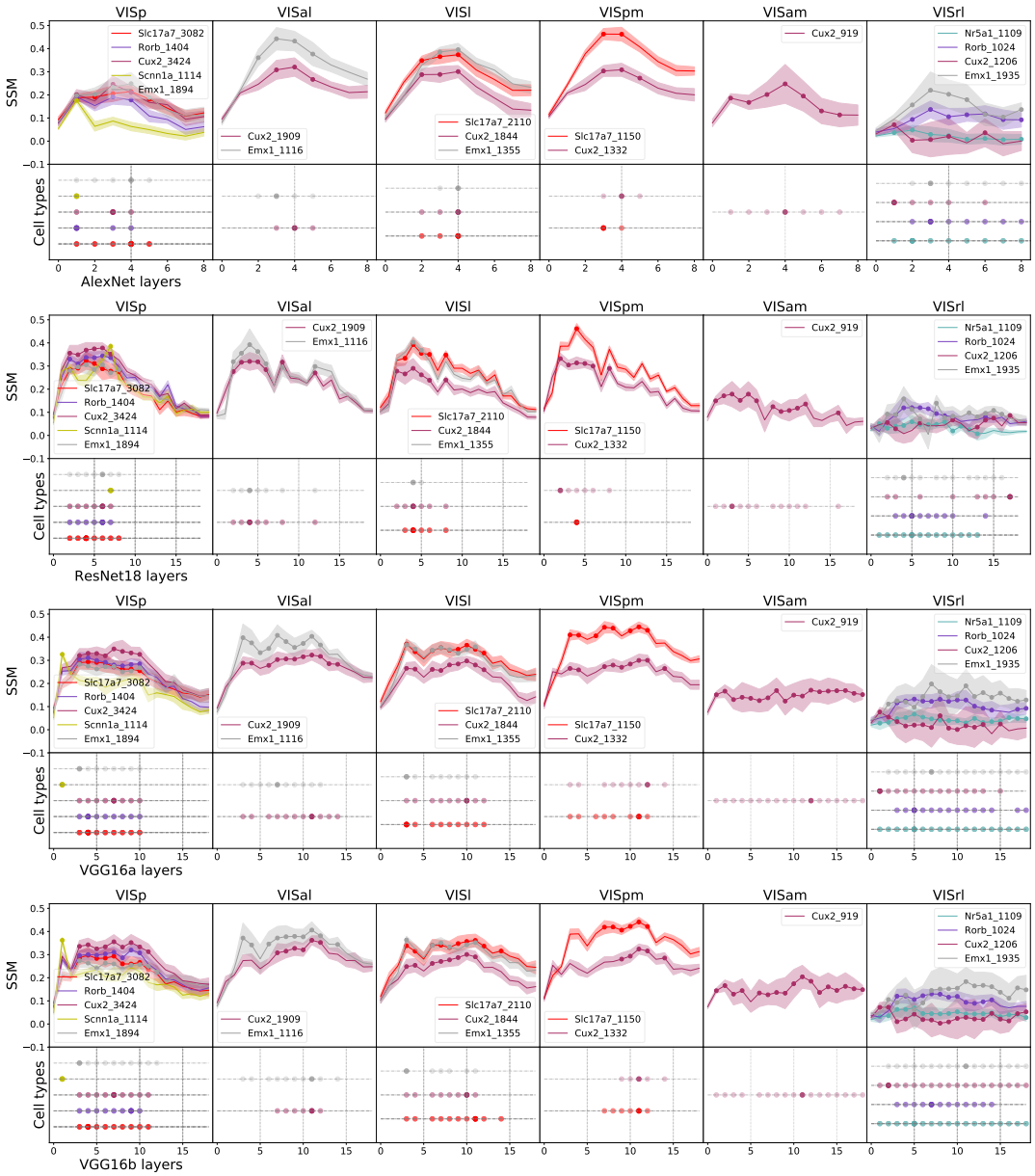

Figure 2.14: Same experiments as Figure 2.7 for AlexNet, ResNet18, VGG16a, VGG16b (top to bottom). These results show that our main conclusions about mouse visual cortex are qualitatively preserved by different models.

INTERLUDE

In the previous chapter, it has been shown that comparing the real brain to artificial neural networks is not only possible, but also fruitful for our understanding of the brain. More importantly, the validity of the comparison framework and metrics provides a valuable criterion for the further development of models that more closely match the real brain.

While task-driven deep neural networks have shown great potential in studying responses in biological brains, they are not precise analogues. Constructing a deep neural network architecture which matches closer to the brain can further benefit our analysis, for example through demonstrating the computational power of a given biological resource, enabling comparisons corresponding homologous groups of neurons in models and the mouse brain, etc.

In the following chapter, we present the first (to our knowledge) deep neural network models of the mouse visual cortex (MouseNet) that are biologically constrained in detail, not only in terms of the basic structure of their connectivity, but also in terms of the count and hence density of neurons within each area, and the spatial extent of their projections.

Chapter 3

CNN MOUSENET: A BIOLOGICALLY CONSTRAINED CONVOLUTIONAL NEURAL NETWORK MODEL FOR MOUSE VISUAL CORTEX

Convolutional neural networks trained on object recognition derive inspiration from the neural architecture of the visual system in primates, and have been used as models of the feedforward computation performed in the primate ventral stream. In contrast to the deep hierarchical organization of primates, the visual system of the mouse has a shallower arrangement. Since mice and primates are both capable of visually guided behavior, this raises questions about the role of architecture in neural computation. In this work, we introduce a novel framework for building a biologically constrained convolutional neural network model of the mouse visual cortex. The architecture and structural parameters of the network are derived from experimental measurements, specifically the 100-micrometer resolution interareal connectome, the estimates of numbers of neurons in each area and cortical layer, and the statistics of connections between cortical layers.

This network is constructed to support detailed task-optimized models of mouse visual cortex, with neural populations that can be compared to specific corresponding populations in the mouse brain. Using a well-studied image classification task as our working example, we demonstrate the computational capability of this mouse-sized network. Given its relatively small size, MouseNet achieves roughly 2/3rds the performance level on ImageNet as VGG16. In combination with the large scale Allen Brain Observatory Visual Coding dataset, we use representational similarity analysis to quantify the extent to which MouseNet recapitulates the neural representation in mouse visual cortex. Importantly, we provide evidence that optimizing for task performance does not improve similarity to the corresponding biological

system beyond a certain point. We demonstrate that the distributions of some physiological quantities are closer to the observed distributions in the mouse brain after task training. We encourage the use of the MouseNet architecture by making the code freely available.

3.1 Introduction

Convolutional neural networks (CNNs) trained on object recognition derive some inspiration from the neuroscience of the visual system in primates, and have been used as models of feedforward computation performed in the primate ventral stream [Fukushima, 1988, Riesenhuber and Poggio, 1999, DiCarlo et al., 2012]. Indeed, the activity in these networks often resembles activity recorded from areas of the primate visual system, from oriented Gabor-like features in early layers [Krizhevsky et al., 2012] to responses to curves and more complex geometries [Serre et al., 2007] and even functional, or *representational*, similarity at the population level [Güçlü and van Gerven, 2015, Kriegeskorte, 2015]. Task-trained artificial neural networks have been shown to produce similar neural representations or develop predictive models of neural activity in visual[Yamins et al., 2014, Yamins and DiCarlo, 2016, Khaligh-Razavi and Kriegeskorte, 2014b], auditory[Kell et al., 2018], rodent whisker areas[Zhuang et al., 2017a], and more[Sandbrink et al., 2020, Michaels et al., 2020, Lindsay, 2020]. Despite these successes and the clear power of CNNs to solve machine learning problems in the visual domain, among others[Krizhevsky et al., 2012, Simonyan and Zisserman, 2014], they are not structural or architectural analogues for the underlying biological circuits. Recent endeavors [Kubilius et al., 2018, Kubilius et al., 2019] show that imposing brain like structure such as shallowness and recurrence in network models can improve their functional similarity to the primate brain. The interplay of architecture and computation remains an open problem in both machine learning and neuroscience.

This issue is especially pronounced for studies of mouse visual cortex, a field undergoing explosive growth. Large scale tract tracing data sets have revealed neuro-anatomical structure in unprecedented detail [Bakker et al., 2012, Oh et al., 2014, Knox et al., 2019, Harris et al., 2016]. From these efforts we have learned, in contrast to the hierarchical organization

of primates, that the visual system of the mouse has a much more parallel structure [Harris et al., 2019]. Since rodents are capable of visually guided behavior including invariant object recognition [Zoccolan et al., 2009, Huberman and Niell, 2011], this raises questions about the role of architecture in neural computation. Recently, data from a large-scale physiological survey of neural activity in the mouse visual system [de Vries et al., 2020] was used to compare the representations of visual stimuli in cortex with those of modern deep networks [Shi et al., 2019, Cadena et al., 2019, Vinken and Op de Beeck, 2020]. It was found that even purportedly "early" regions such as V1 in mouse cortex are more similar at the level of representation to middle layers of networks such as VGG16, rather than to early layers that respond optimally to simple visual features and bear more resemblance to the "simple" and "complex" cells normally supposed to describe the early visual pathway. However, the profound difference in architecture between modern CNNs and the mouse cortex raises significant challenges in interpreting these findings. To begin, many modern computational models of vision (in particular CNNs, which often have a high input resolution) have a larger number of units than the number of neurons in mouse visual cortex. Moreover, CNNs from computational vision are largely of feedforward type, either purely so or with some skip connections (e.g., in ResNet architectures), which ignores the large amount of recurrence present in real neural circuits. Furthermore, the mouse thalamo-cortical system is quite shallow [Harris et al., 2019]. Most importantly, as stated above and detailed more below, the mouse visual cortex has an intriguing parallel structure.

Here we introduce a novel framework for incorporating these data to build a biologically constrained convolutional neural network model of the mouse visual cortex — the CNN MouseNet. Convolutional neural networks share weights across the visual field, and thus form an approximation of the functional properties that may be imposed by translation invariance of natural stimuli leading to equivariant representations in neural systems [Fukushima, 1988, Riesenhuber and Poggio, 1999, DiCarlo et al., 2012].

This weight sharing makes them much easier to train, which is an important practical consideration for model development. The structural parameters of MouseNet are derived

from experimental measurements, specifically estimates of numbers of neurons in each area and cortical layer, the 100-micrometer resolution interareal connectome, and the statistics of connections between cortical layers.

MouseNet is constructed to support detailed task-optimized models of mouse visual cortex, with neural populations that can be compared to specific corresponding populations in the mouse brain. To demonstrate the usage of MouseNet, we use standard image classification tasks as working examples; specifically, we train MouseNet to perform classification using the ImageNet Large Scale Visual Recognition Challenge 2012 (ILSVRC2012) [Russakovsky et al., 2015] as well as the CIFAR10 [Krizhevsky, 2012] data sets.

We find that, although MouseNet is much smaller than a typical CNN and has specific architectural differences, it can reach above 90% validation accuracy on CIFAR10, and roughly 2/3rds of the performance level of a typical CNN (VGG16) on the more challenging ImageNet classification benchmark.

Next, using the large-scale functional data sets from the Allen Brain Observatory [de Vries et al., 2020] on visual responses of neurons across visual cortex, we investigate the functional properties of the MouseNet architecture after training on the ImageNet dataset. We use representational similarity analysis [Diedrichsen and Kriegeskorte, 2017, Barrett et al., 2019, Shi et al., 2019] to investigate the relative effects of task-training on the different computational layers in the model. We see that ImageNet classification training of MouseNet makes responses in its corresponding level of layers more similar to responses recorded from the mouse brain.

We then contrast these results for the biologically constrained MouseNet with those for a standard CNN network, VGG16, trained on the same task. We show that the representational similarity of MouseNet to the mouse brain is comparable to that of VGG16, even though VGG16 produces significantly higher task performance.

We study the training process for both networks, and find that the highest SSM values between a model neural network and the mouse brain areas are not necessarily achieved by the best performing models, rather at early or intermediate points during the training

process. We take this as an indication that image classification using ImageNet is not the appropriate task to describe the mouse visual cortex (or at least those regions measured in the Allen Brain Observatory) rather than a failure of the task-training approach. This conclusion is perhaps to be expected. However, we feel that the use of object recognition is an important baseline in comparison with established results in primate.

Furthermore, in addition to broad measures of representational similarity across images, we also demonstrate the effect of image classification training on MouseNet by showing how it affects the other functional properties such as lifetime sparseness and orientation selectivity index [de Vries et al., 2020]. We find that training drives both of these properties to become more similar between MouseNet and the biological mouse brain. Finally, by comparing both VGG16 and MouseNet representations in individual layers before and after training, we find that the image classification task makes MouseNet layers more diverse after training, a phenomenon we attribute to the parallel pathways in the MouseNet architecture.

Overall, we describe an open framework for constructing MouseNet that is general and can be easily modified to incorporate new data on the structure of the mouse brain [Abbott et al., 2020]. Likewise, MouseNet can be readily trained on other tasks [Nayebi et al., 2021], including those corresponding more closely to natural behavior. We encourage future research along these lines by making the Python code publicly available at `https://github.com/mabuice/Mouse_CNN`, together with the step-by-step description of the model construction that we present next.

3.2 Construction of CNN MouseNet

In this section, we introduce our framework for constructing the CNN MouseNet. Fig 3.1 shows an overview of this framework. The basic idea is to use available sources of anatomical data (e.g. tract tracing data, cell counts, and statistics of short-range connections) to constrain the CNN network structure and architectural hyperparameters. We discuss the details of each step below.

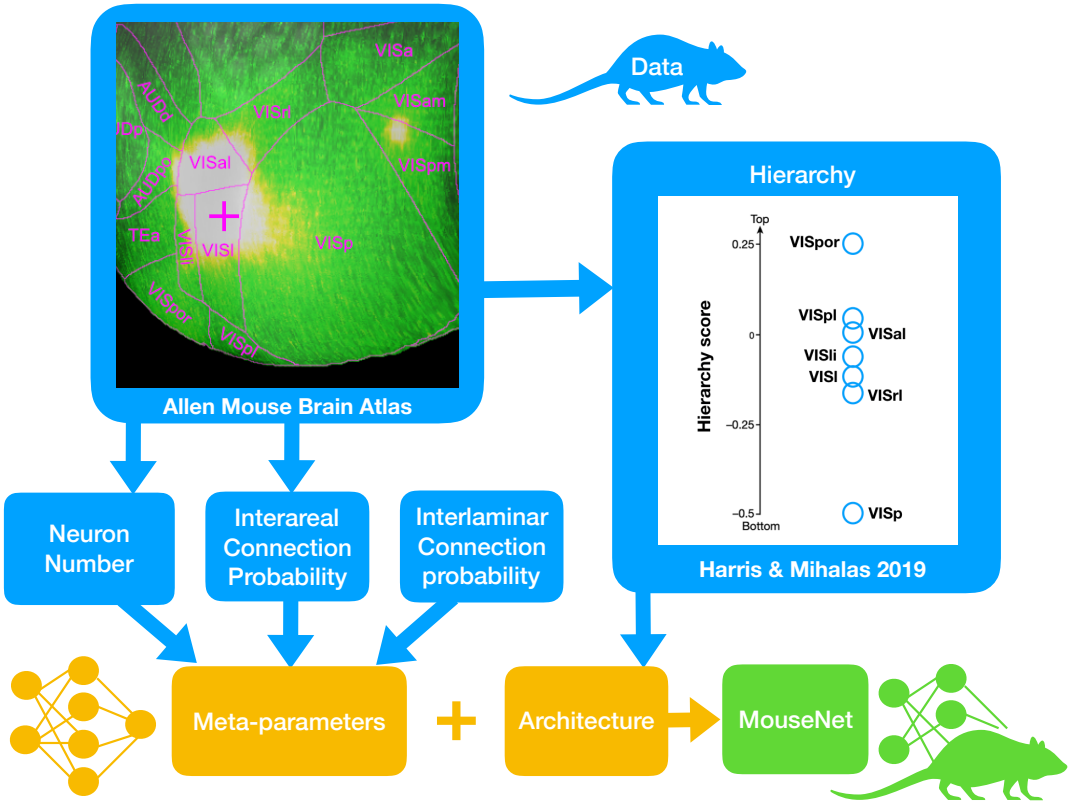

Figure 3.1: **Modeling framework.** Framework for constructing MouseNet from biological constraints on anatomy, via publicly available data from large-scale experiments. The CNN architecture is set by the analysis of hierarchy [Harris et al., 2019] on the Allen Mouse Brain Connectivity Atlas [Oh et al., 2014] (Image credit: Allen Institute); and the meta-parameters are mostly fixed by the combination of the 100-micrometer resolution interareal connectome [Knox et al., 2019] with detailed estimates of neuron density [Erö et al., 2018], and the statistics of connections between cortical layers from the literature [Billeh et al., 2020, Levy and Reyes, 2012, Stepanyants et al., 2007].

Network architecture

MouseNet spans the dorsal lateral geniculate nucleus (dLGN) and six visual areas (Fig 3.2A). Input to the network passes first through dLGN, and then to the primary visual area VISp.

After VISp, the architecture branches into five parallel pathways, representing five secondary lateral visual areas: VISl (lateral visual area), VISal (anterolateral), VISpl (posterolateral), VISli (laterointermediate), and VISrl (rostrolateral). Finally, the output of VISp together with all five lateral visual areas provide input to VISpor (postrhinal). We include only the lateral areas because they are more associated to object recognition while the medial areas are more involved in multimodal integration [Glickfeld and Olsen, 2017].

The three-level architecture among the VIS areas was derived from an analysis of the hierarchy of mouse cortical and thalamic areas (Fig. 6e in [Harris et al., 2019]), which considered feedforward and feedback connection structures in each area. In this analysis, VISp was clearly low in the hierarchy, and VISpor was clearly high, but the other lateral visual areas had similar intermediate positions.

In the MouseNet model, each VIS area is represented by three separate cortical layers: layer 4 (L4), layer 2/3 (L2/3) and layer 5 (L5). We call a specific cortical layer within a specific area a "region". Here we only consider the feedforward pathway, thought in primate to drive responses within $\approx$ 100ms of stimulus presentation [Riesenhuber and Poggio, 1999, Yamins et al., 2014]. Following depictions of the canonical microcircuit (e.g. as summarized in Fig 5 in [Amorim Da Costa and Martin, 2010]), we consider the feedforward pathway to consist of laminar connections from L4 to L2/3, and from L2/3 to L5. Input from other areas feed into L4 and all of L4, L2/3 and L5 output to downstream areas, as shown in Fig 3.2B. This is consistent with broad connectivity among visual areas from each of these layers (Fig. 2f of [Harris et al., 2019]). Fig 3.2C shows the MouseNet architecture in full detail, including all 22 regions and associated connections.

From architecture to convolutional neural net

Similar to the CALC model for the primate visual cortex by one of the authors [Tripp, 2019], the general idea is to use convolution (Conv) operations to model the projections between different regions in the mouse visual cortex. Conv operations are linear combinations of many inputs, so they impose the assumption of linear synaptic integration. They are widely

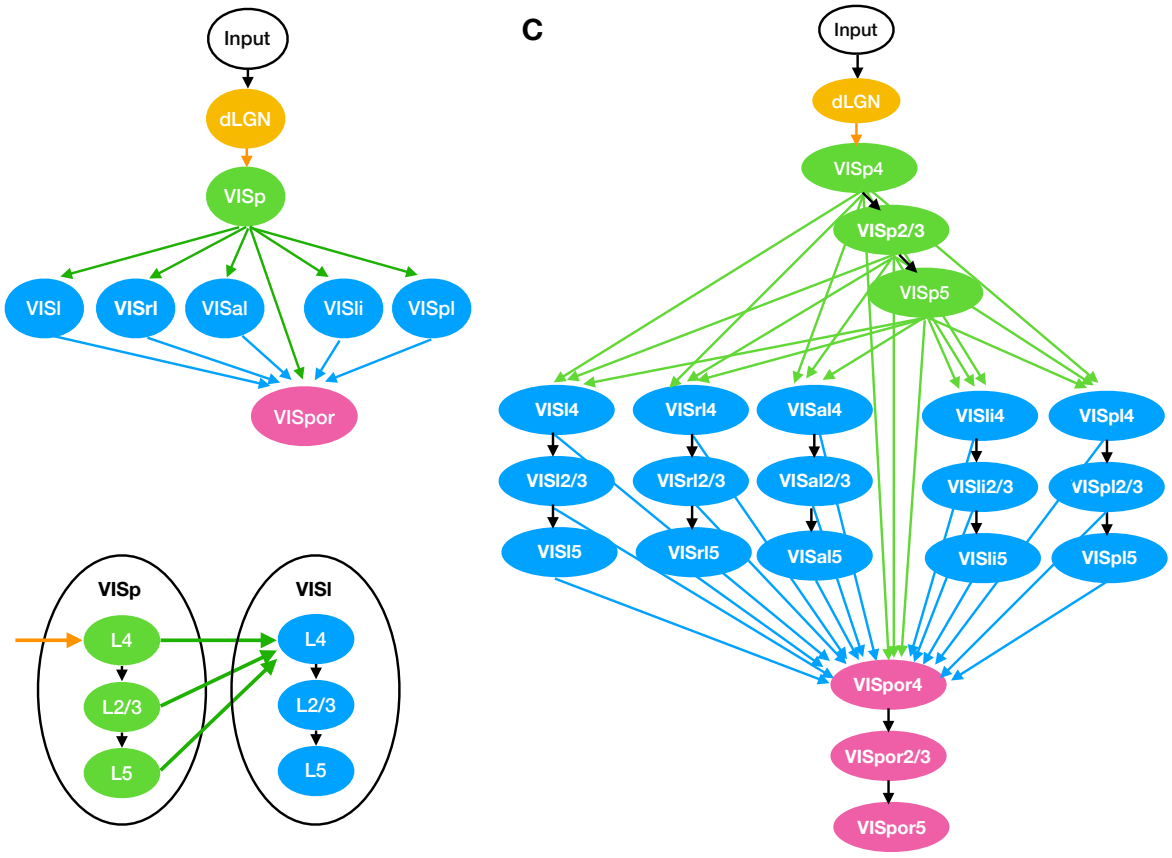

Figure 3.2: **Illustration of MouseNet architecture.** Only feedforward connections are included. (A) High-level organization of MouseNet, based on analysis of the hierarchy of lateral visual areas ([Harris et al., 2019]). (B) Connection patterns at the level of cortical layers. (C) Full MouseNet architecture.

used in machine learning, because they run efficiently on graphical processing units, and they share parameters across the visual field, leading to reduced memory requirements and faster learning, relative to general linear maps.

Each connection from source brain region i to target brain region j is modelled with a Conv operation, Conv^{ij}. The input to Conv^{ij} corresponds to the neural activities in source

region i, and the output of Conv^{ij} drives neural activities in the target region j. For example, as shown in Fig 3.3A, the projection from Region 1 to Region 2 (Proj 1→2) is modeled by Conv^{12}. The neural activities in Region 1 correspond to the input to Conv^{12}, while the neural activities in Region 2 are a nonlinear function (ReLU, as shown in Fig 3.3C) of the output of Conv^{12}. In MouseNet, L4 of all areas except VISp receive multiple converging inputs, similar to Region 4 in Fig 3.3A. In this case, each projection (Proj 2→4 and Proj 3→4) is modeled by a separate Conv layer (Conv^{24} and Conv^{34}), and a nonlinear function (ReLU) is applied to the sum of the output from both of the Conv layers, to produce the neural activities in Region 4.

3.2.1 Finding meta-parameters consistent with mouse data

After fixing the architecture, we need to determine the meta-parameters for constructing the kernels for each Conv operation (Fig 3.3). The standard Conv operation is defined in terms of a four-dimensional kernel. The output of the kernel is a three-dimensional tensor of activations for region j, A^j, which pass through element-wise ReLU nonlinearities to produce non-negative rates. Element $A^j_{\alpha\beta\gamma}$ is the activation of the neuron at the α^{th} vertical and β^{th} horizontal position in the visual field, in the γ^{th} channel (or feature map). The γ^{th} channel of the activation tensor for region j, A^j_γ, depends on inbound connections as,

$$A^j_\gamma = \sum_{i \in I^j} \sum_{\delta} C^{ij}_{\gamma\delta} * A^i_\delta, \tag{3.1}$$

where I^j is the set of regions that provide input to region j. Note that both $C^{ij}_{\gamma\delta}$ and A^j_δ are two-dimensional, and they undergo standard two-dimensional convolution. The meta-parameters of kernel C^{ij} are: number of input channels c^{ij}_{in}, number of output channels c^{ij}_{out}, stride s^{ij}, padding p^{ij}, and finally kernel size k^{ij}, i.e. the height and width (which are set equal) of $C^{ij}_{\gamma\delta}$. To make the connections realistically sparse, we add a binary Gaussian mask on the Conv operations, whose parameters are also estimated from data. See Fig 3.3B for an illustration of Conv operation with Gaussian mask. We constrain these meta-parameters with

quantitative data whenever possible, and reasonable assumptions indicated by experimental observations otherwise, as indicated below.

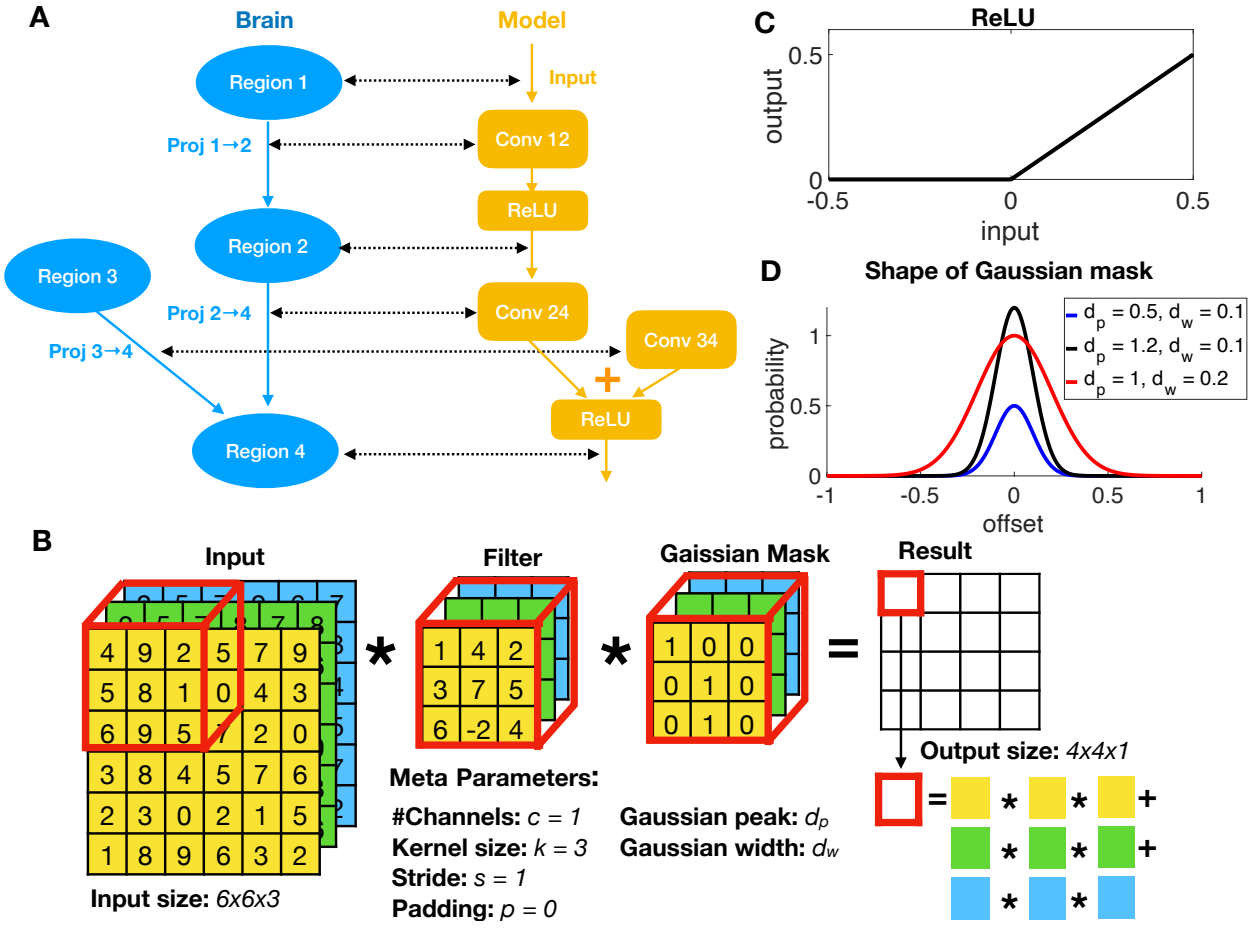

Figure 3.3: **From mouse brain to CNN model.** (A) From mouse brain hierarchy to CNN architecture. (B) An example of Conv operation with Gaussian mask. (C) ReLU operation in the CNN architecture. (D) The binary Gaussian mask is generated by a Gaussian shaped probability whose peak and width are meta-parameters.

Cortical population constraints

Assumptions about area output size We set the horizontal and vertical resolution of the input (in pixels) based on mouse visual acuity. According to [Prusky et al., 2000], the upper bound for visual acuity in mice is 0.5 cycles/degree, corresponding to a Nyquist sampling rate of 2 pixels/cycle x 0.5 cycles/degree = 1.0 pixel/degree. According to retinotopic map studies[Zhuang et al., 2017b], V1 included a visual coverage range of $\sim 60°$ in altitude and $\sim 90°$ in azimuth, we further simplified this to square input size of 64 by 64 pixels.

The resolution of the other regions depends on both the resolution of the input, and strides of the connections. The stride of a connection is the sampling period with respect to its input. For example, a Conv with a stride of one samples every element of its input, whereas a Conv with a stride of two samples every other element (both horizontally and vertically), leading to output of half the size in each dimension. Because cortical neurons are not organized into discrete channels in the same way as convolutional network layers, there is no strong anatomical constraint on the stride. However, the mean stride has to be somewhat less than two; there are ten steps in the longest path through MouseNet, but if only six of them had a stride of two, the 64x64 input would be reduced to 1x1 in VISpor, with no remaining topography. Lacking strong constraints, for simplicity, we first attempted to set all the strides to one, but this left very few channels in some of the smaller regions (due to an interaction between channels and strides that we describe below). We therefore set the strides of the connections outbound from VISp to two, and others to one. The feature maps of dLGN and VISp were therefore 64x64 (the same as the input), and all others were 32x32.

Given the resolutions of the channels in each region, the number of channels are constrained by the number of neurons. Specifically, Let n^i be the number of neurons in region i and (l_x^i, l_y^i) be the size of the output in area i, then the number of channels in area i is determined by $c^i = n^i/(l_x^i * l_y^i)$.

Estimating number of neurons in each area from data We only model the

excitatory neural population in our model, consistent with the fact that all neurons in the model project to other visual areas. In fact, neurons in convolutional networks are neither excitatory or inhibitory, but have both positive and negative output weights. However, past modelling work [Parisien et al., 2008, Tripp and Eliasmith, 2016] has shown that such mixed-weight projections can be transformed so that the original neurons are all excitatory, and an additional population of inhibitory neurons recovers the functional effects of the original weights.

According to [Evangelio et al., 2018], the estimated number of excitatory neurons in dLGN is 21200. For VISp, VISal, VISl, VISpl, we use estimated density for excitatory neurons given by [Erö et al., 2018][1], which is summarized in Tabel 3.1. Note that we use neuron density instead of counts to get a more stable estimation of number of neurons out of different versions of brain parcellations. For the remaining areas VISrl, VISli and VISpor, we approximate their density by taking the average across the above four areas with separated cortical layers.

Table 3.1: **Exitatory population density [$\mathbf{mm^{-3}}$] [Erö et al., 2018].**

	L4	L2/3	L5
VISp	106114.7	86668.2	86643.4
VISal	93176.9	79070.6	78540.9
VISl	86559.9	73937.9	66215.6
VISpl	106783.0	87368.3	82538.1
Average	98158.6	81761.1	78484.5

Combined with the number of 10µm voxels counted in the Allen Mouse Brain Common Coordinate Framework (CCFv3) [Wang et al., 2020] (Table 3.2), we summarize the estimated number for all the regions in our model in Table 3.3.

[1] `https://bbp.epfl.ch/nexus/cell-atlas/`

Table 3.2: **Number of 10µm voxels in each region.**

	L4	L2/3	L5
VISp	1023640	1999040	1552688
VISal	104152	199314	202942
VISl	179084	301588	314522
VISpl	36638	205150	242812
VISrl	146294	276390	244294
VISli	57256	117252	147946
VISpor	60632	373972	385168

Table 3.3: **Estimated number of exitatory neurons in each region.**

	L4	L2/3	L5	Total
dLGN				21200
VISp	108623	173253	134530	
VISal	9705	15760	15939	
VISl	15501	22299	20826	
VISpl	3912	17924	20041	
VISrl	14360	22598	19173	
VISli	5620	9587	11611	
VISpor	5952	30576	30230	

Cortical connection constraints

Neurons tend to receive relatively dense inputs from other neurons that are above or below them, in other cortical layers, and the connection density decreases with increasing horizontal distance. Similarly, inputs from other cortical areas tend to have a point of highest density, with smoothly decreasing density around that point. We approximate such connection-

density profiles with two-dimensional Gaussian functions. Specifically, the fan-in connection probability from source region i to target region j at position (x, y) (position offset from center in µm) is modeled as,

$$P^{ij}(x,y) = d_p^{ij} \exp\left(-\frac{x^2}{2(d_x^{ij})^2} - \frac{y^2}{2(d_y^{ij})^2}\right). \tag{3.2}$$

where d_p^{ij} is the peak probability at the center and d_x^{ij} and d_y^{ij} are the widths in the x and y directions. For simplicity, we assume $d_x^{ij} = d_y^{ij} \triangleq d_w^{ij}$ and let $r = \sqrt{x^2 + y^2}$ denote the offset from the center of the source layer, the above equation then simplifies to,

$$P^{ij}(r) = d_p^{ij} \exp\left(-\frac{r^2}{2(d_w^{ij})^2}\right), \tag{3.3}$$

where d_w^{ij} (µm) is the Gaussian width.

Both d_p^{ij} and d_w^{ij} are estimated from mouse data. The parameters for interlaminar connections are estimated from statistics of connections between cortical layers in paired electrode studies (Section Estimating d_w^{ij}, d_p^{ij} for interlaminar connections), and the parameters for interareal connections are estimated from the mesoscale mouse connectome (Section Estimating d_w^{ij} and d_p^{ij} for interareal connections).

Conv layer with Gaussian mask

To translate our Gaussian models of connection density into network meta-parameters, we apply a binary mask to the weights of the Conv layers (Fig 3.3B). To do that, we first change the unit of d_w^{ij} in Eq.3.3 from micrometers to source area-dependent "pixels" (unit of output size of source area i) by multiplying it with $\sigma_i = \sqrt{(l_x^i * l_y^i)/a_i}$ (pixel/µm), where a_i denotes the surface size of area i, estimated from the voxel model (See Estimating d_w^{ij} and d_p^{ij} for interareal connections). We then have,

$$P^{ij}(\tilde{r}) = d_p^{ij} \exp\left(-\frac{\tilde{r}^2}{2(\tilde{d}_w^{ij})^2}\right), \quad \tilde{d}_w^{ij} = \sigma_i d_w^{ij}, \tag{3.4}$$

where both $\tilde{r}$ and $\tilde{d}_w^{ij}$ are in the "pixel" unit. The kernel size of the Conv layer is set to be $k^{ij} = 2 \times \lfloor \tilde{d}_w^{ij} \rfloor + 1$, with padding calculated by $p^{ij} = (k^{ij} - s^{ij})/2$, where s^{ij} is the stride of

the Conv layer. During initialization, a mask containing zeros and ones is generated for each Conv layer, with size $(c_{out}^{ij}, c_{in}^{ij}, k^{ij}, k^{ij})$. The probability of each element being one is $P^{ij}(\tilde{r})$, where $\tilde{r}$ (pixel) is the offset from the center of mask. The weights of the Conv layer are then multiplied by the mask. This gives the connections realistic densities (or sparsities), with realistic spatial profiles.

Estimating d_w^{ij}, d_p^{ij} for interlaminar connections

For the interlaminar connections, we estimate the Gaussian width d_w^{ij} from multiple experimental resources. Firstly, from Table 3 in [Levy and Reyes, 2012], we get the estimation of d_w^{ij} to be 114 micrometers for functional connections between pairs of L4 pyramidal cells in mouse auditory cortex. Secondly, manually extracted from [Stepanyants et al., 2007] Fig 8B, we obtain the variation of the Gaussian width depending on source and target layer from cat V1. Finally, we use this variation to scale the L4 to L4 width of 114 µm to other layers in the mouse cortex. We summarize the Gaussian widths from cat cortex, along with corresponding scaled estimates for mouse cortex, in Table 3.4. Note that in the current model, we only use the values for connections from L4 to L2/3 and from L2/3 to L5 (Fig 3.2B).

Table 3.4: **Estimated Gaussian width d_w^{ij} for interlaminar excitatory connections.** The values outside of the parenthesis are extracted from [Stepanyants et al., 2007]; the values inside the parenthesis are scaled to mouse cortex, using the width 114 µm for L4-to-L4 connections in mouse auditory cortex [Levy and Reyes, 2012]. Units are micrometers (µm).

		Target		
		L2/3 (scaled)	L4 (scaled)	L5 (scaled)
Source	L2/3	225 (142.5)	50 (31.67)	100 (63.33)
	L4	220 (139.33)	180 (114)	140 (88.67)
	L5	150 (95)	100 (63.33)	210 (133)

To estimate the Gaussian peak probability d_p^{ij}, we first collect the connection probability between excitatory populations offset at 75 micrometer d_{75}^{ij} (Fig. 4A in [Billeh et al., 2020]). We then get the peak probability d_p^{ij} by the relation

$$d_p^{ij} = d_{75}^{ij} / \exp\left(-\frac{75^2}{2(d_w^{ij})^2}\right) \tag{3.5}$$

We summarize the probability at 75 micrometers d_{75}^{ij} along with the peak probability d_p^{ij} in Table 3.5.

Table 3.5: **The connection probability between excitatory populations offset at 75 micrometer** d_{75}^{ij} The numbers are from Fig 4A in [Billeh et al., 2020]). The calculated Gaussian peak probability d_p^{ij} are given in parenthesis.

		Target		
		L2/3 (peak)	L4 (peak)	L5 (peak)
Source	L2/3	0.160 (0.184)	0.016 (0.264)	0.083 (0.167)
	L4	0.140 (0.162)	0.243 (0.302)	0.104 (0.149)
	L5	0.021 (0.029)	0.007 (0.014)	0.116 (0.136)

Estimating d_w^{ij} and d_p^{ij} for interareal connections

To estimate interareal connection strengths and spatial profiles, we use the mesoscale model of the mouse connectome [Knox et al., 2019, Harris et al., 2016]. This model estimates connection strengths between 100 micrometer resolution voxels, based on 428 individual anterograde tracing experiments mapping fluorescent labeled neuronal projections in wild type C57BL/6J mice.

Flat map The voxel model is in 3 dimensional space. For the purpose of our analysis, we need to map the 3 dimensional structure into 2 dimensions. First, we fit the visual area positions by a sphere with center $c \in \mathcal{R}^3$ and radius r. Each position $x \in \mathcal{R}^3$ is then mapped

to $\bar{x} \in \mathcal{R}^2$ with relation

$$\bar{x}_1 = v \cdot r \cdot \arctan\left(\frac{x_1 - c_1}{x_2 - c_2}\right), \tag{3.6}$$

$$\bar{x}_2 = v \cdot r \cdot \arctan\left(\frac{x_3 - c_3}{\sqrt{(x_1 - c_1)^2 + (x_2 - c_2)^2}}\right) \tag{3.7}$$

where $v = 100\mu$m is the size of the voxel.

Area size Approximations of the surface area for each brain region are needed to convert the widths of connection profiles (see Conv layer with Gaussian mask) from voxels in the mesoscale model to convolutional-layer pixels in MouseNet. For this purpose, each region's surface area size is approximated by the area of a convex hull of its mapped two-dimensional positions. These estimates are summarized in Table 3.6.

Table 3.6: **Area size (mm^2) estimated from the voxel model.**

	VISp	VISal	VISl	VISli	VISpl	VISrl	VISpor
L4	4.3271	0.4909	0.8793	0.3355	0.2865	0.6182	0.5264
L2/3	4.7406	0.5477	0.9279	0.4356	0.6659	0.6980	1.3937
L5	4.2511	0.4972	0.8651	0.4039	0.6785	0.6748	1.2445

Estimating d_w^{ij} For each connection from source region i to target region j, we estimate d_w^{ij} from the mesoscale model. The first step is to estimate the widths of connections to individual voxels in j. The incoming width d_k^{ij} for target voxel k in j is estimated by the standard deviation of the connection strength about its center of mass. Specifically, $d_k^{ij} = (\sum_l w_{lk} d_l^2 / \sum_l w_{lk})^{1/2}$, where l indexes the voxels in source region j, w_{lk} is the connection weight between source and target voxels l and k in the mesoscale model, and d_l is the distance from voxel l to the center of mass of these connection weights. We then estimate d_w^{ij} as the average of these widths over the voxels in j. We omit from this average any target voxels that have multi-modal input profiles. This procedure provides an upper bound for d_w^{ij}, because a target voxel may include multiple neurons with partially overlapping input areas.

Estimating d_p^{ij} The mesoscale model provides estimates of relative densities of connections between pairs of voxels. But an additional factor is needed to convert these relative densities into neuron-to-neuron connection probabilities. For this purpose, we assumed that each neuron received inputs from 1000 neurons in other areas (we call this number the extrinsic in-degree, e). This is on the order of the estimate from Fig S9 M in [Markram et al., 2015]. Given this assumption, we calculated d_p^{ij} by the relation,

$$e \cdot \frac{w_{ij}}{\sum_i w_{ij}} = 2\pi(\tilde{d}_w^{ij})^2 \cdot d_p^{ij} \cdot c^i, \tag{3.8}$$

where w_{ij} is the connection strength from source area i to target area j, estimated from integrating the connection weights of the corresponding areas in the mesoscale model. The estimated values for d_w^{ij} and d_p^{ij} are given in Table 3.7.

Conv kernel size for dLGN

The above methods allowed us to set kernel sizes for intracortical connections, but not subcortical ones. We set the kernel sizes for inputs to dLGN and VISp L4 according to receptive field sizes in these regions. Receptive fields are about 9 degrees in dLGN and 11 degrees in VISp [Durand et al., 2016]. As mentioned in Section Cortical population constraints, mouse visual acuity is approximately 1 pixel/degree, therefore we set kernel size of the connection from input to dLGN to 9x9. We then set the kernel size of the connection from dLGN to VISp to 3x3, such that the receptive field size for VISp is 11x11 pixels.

Summary tables

In Table 3.8, we summarize the calculated number of channels in each area (in parenthesis) and the kernel size for each Conv layer.

The parameters used in the model based on biological sources and assumptions are summarized in Table 3.9 and the formulae for calculating the Conv layer meta-parameters are sumarized in Table 3.10.

Table 3.7: **The estimated d_w^{ij} (μm) and d_p^{ij} for interareal connections.**

Source	Target	d_w^{ij} (μm)	d_p^{ij}
VISp4	VISal4	277.1	0.039
	VISl4	313.2	0.030
	VISli4	296.7	0.032
	VISpl4	290.6	0.032
	VISrl4	306.7	0.032
	VISpor4	276.8	0.013
VISp2/3	VISal4	266.1	0.063
	VISl4	325.8	0.038
	VISli4	303.3	0.045
	VISpl4	284.2	0.047
	VISrl4	339.4	0.032
	VISpor4	307.4	0.013
VISp5	VISal4	239.0	0.064
	VISl4	311.5	0.042
	VISli4	314.3	0.043
	VISpl4	278.3	0.053
	VISrl4	311.4	0.042
	VISpor4	298.3	0.016

Source	Target	d_w^{ij} (μm)	d_p^{ij}
VISal4	VISpor4	37.98	0.551
VISal2/3		13.81	4.362
VISal5		14.87	4.213
VISl4		204.7	0.014
VISl2/3		210.6	0.016
VISl5		215.4	0.019
VISli4		155.1	0.017
VISli2/3		169.3	0.022
VISli5		148.4	0.028
VISpl4		22.2	0.190
VISpl2/3		59.5	0.054
VISpl5		54.4	0.079
VISrl4		97.2	0.074
VISrl2/3		105.4	0.064
VISrl5		110.4	0.064

Table 3.8: **The calculated meta-parameters for the Conv layers.**

Source(#channel)	Target	kernel size
input(3)	LGNv	9×9
dLGN(5)	VISp4	3×3
VISp4(26)	VISp2/3	9×9
	VISal4	17×17
	VISl4	19×19
	VISli4	19×19
	VISpl4	19×19
	VISrl4	19×19
	VISpor4	17×17
VISp2/3(42)	VISp5	3×3
	VISal4	15×15
	VISl4	19×19
	VISli4	17×17
	VISpl4	17×17
	VISrl4	21×21
	VISpor4	19×19
VISp5(32)	VISal4	15×15
	VISl4	19×19
	VISli4	19×19
	VISpl4	17×17
	VISrl4	19×19
	VISpor4	19×19
VISpor4(5)	VISpor2/3	13×13
VISpor2/3(29)	VISpor5(29)	3×3

Source(#channel)	Target	kernel size
VISal4(9)	VISal2/3	13×13
	VISpor4	3×3
VISal2/3(15)	VISal5	5×5
	VISpor4	1×1
VISal5(15)	VISpor4	1×1
VISl4(15)	VISl2/3	9×9
	VISpor4	15×15
VISl2/3(21)	VISl5	5×5
	VISpor4	15×15
VISl5(20)	VISpor4	15×15
VISli4(5)	VISli2/3	17×17
	VISpor4	17×17
VISli2/3(9)	VISli5	7×7
	VISpor4	17×17
VISli5(11)	VISpor4	15×15
VISpl4(3)	VISpl2/3	19×19
	VISpor4	3×3
VISpl2/3(17)	VISpl5	5×5
	VISpor4	5×5
VISpl5(19)	VISpor4	5×5
VISrl4(14)	VISrl2/3	11×11
	VISpor4	7×7
VISrl2/3(22)	VISrl5	5×5
	VISpor4	9×9
VISrl5(18)	VISpor4	9×9

Table 3.9: **Parameters from data or assumptions.**

Notation	CNN parameter	Biological source or assumptions
n^i	Number of neurons in area i	Based on [Erö et al., 2018] combined with the voxel model [Knox et al., 2019]
a_i	Two dimensional area size for area i	Estimated from voxel model [Knox et al., 2019]
e	Total fan-in connections for all areas	Set to be 1000 based on [Markram et al., 2015]
(l_x^0, l_y^0)	Input size to the model	Set to be 64x64 based on mouse visual acuity [Prusky et al., 2000]
(l_x^i, l_y^i)	Output size of area i	Set to be 64x64 up to VISp, 32x32 after VISp (Assumption)
d_w^{ij}	Gaussian width (interlaminar)	Estimated from mouse [Levy and Reyes, 2012] and cat [Stepanyants et al., 2007] cortical properties
	Gaussian width (interareal)	Estimated from voxel model [Knox et al., 2019]
d_p^{ij}	Guassian peak (interlaminar)	Based on statistics of connections in paired electrode studies [Billeh et al., 2020]
	Gaussian peak (interareal)	Estimated from voxel model [Knox et al., 2019]

Table 3.10: **Meta-parameters for Conv layer connecting source area i to target area j.**

Notation	CNN parameter	Formula
c^i	number of channels in area i	$c^i = n^i/(l_x^i \cdot l_y^i)$
k^{ij}	kernel size	$k^{ij} = 2 \times \lfloor \tilde{d}_w^{ij} \rfloor + 1$
s^{ij}	stride	$s^{ij} = l_x^i/l_x^j = l_y^i/l_y^j$
p^{ij}	padding	$p^{ij} = (k^{ij} - s^{ij})/2$
$\tilde{d}_w^{ij}$	Gaussian width	$\tilde{d}_w^{ij} = \sqrt{(l_x^i \cdot l_y^i)/a_i} \cdot d_w^{ij}$
d_p^{ij}	Gaussian peak	d_p^{ij}

3.3 Results

In this section, we use a well established image classification task as a working example to demonstrate the usage of the CNN MouseNet and to derive novel findings relating architecturally constrained CNNs and the mouse brain. We first assess the computational performance of this mouse-architecture network on an image classification task. Then, through systematic comparisons with the large scale Allen Brain Observatory dataset, we show how MouseNet can be used to probe the effect of a CNN's specific task training and architecture on its similarities and differences with responses in the biological brain.

Implementation of MouseNet

To enable training of MouseNet on a standard image classification task, we implemented the MouseNet structure in PyTorch [Paszke et al., 2019]. Each Conv layer was followed by a batch normalization layer and a ReLU non-linearity. For regions such as VISpor L4 that receive input from multiple Conv layers, the outputs of the Conv layers are summed before being fed into the batch normalization layer and ReLU non-linearity.

To train the MouseNet model on an image classification task, we added a simple classifier. Specifically, in order to include the final processing output from each individual area such that the information is not bottlenecked by the relatively small VISpor area, we took the L5 output from all seven areas and reduce them to 4x4 by an average pooling layer. We then flattened, concatenated, and fed this to a linear fully-connected layer, which reduced the dimension to the number of classes of the task. The outputs were then transformed to probabilities by the softmax function, and the cross-entropy loss of the predicted probabilities (determined from the categorical distribution where individual class probabilities are from the output of the softmax) relative to the ground truth labels was used to train on the image classification task.

Computational Performance of MouseNet on image classification

We trained MouseNet end-to-end using stochastic gradient decent with momentum, adapting the training script from the imagenet example script from the PyTorch examples github repository[2]. Full training details and scripts are available on the MouseNet github repo: `https://github.com/mabuice/Mouse_CNN`.

We first found that MouseNet achieved above 90% validation accuracy on CIFAR10 data set [Krizhevsky, 2012], a simple classification of 32x32 images into 10 categories. Interestingly, this is close to state of the art performance of modern networks, suggesting that mouse sized networks are fully capable of performing this simple task.

We then moved to the more challenging image classification benchmark of ImageNet [Deng et al., 2009], which requires classification of higher resolution images into 1000 categories. We found that, even for input images downsampled to a resolution of (64x64), MouseNet can still be trained to perform above 37% top-1 validation accuracy on ImageNet. Below, we contrast representations in MouseNet to those in VGG16 trained with the same downsampled input size (64x64), which achieved above 60% top-1 validation accuracy on ImageNet. We contrast the number of parameters in MouseNet and VGG16 in Table 3.11. Note that the number of parameters of MouseNet Conv layers without the Gaussian masks is about 14% of that for VGG16, while the number of parameters of MouseNet Conv layers with Gaussian masks is less than 1% of that for VGG16. Our simulation results are all based on MouseNet models with Gaussian masks.

Table 3.11: **Number of parameters for MouseNet and VGG16 for 1000-class ImageNet classification task.**

	Conv layers	Conv with mask	Classifier
VGG16	14.7M	14.7M	123M
MouseNet	2.1M	87K	2.3M

[2] `https://github.com/pytorch/examples/tree/master/imagenet`

3.3.1 The Effects of Task Training on Functional Properties

To examine the effect of the image classification task training on the functional similarity of the MouseNet and the biological mouse brain, we make use of the large-scale, publicly available Allen Brain Observatory dataset [de Vries et al., 2020]. We study representational similarity of MouseNet and the biological mouse brain across a set of natural images, along with the basic functional properties of sparsity and orientation selectivity.

The Allen Brain Observatory data set

The Allen Brain Observatory data set is a large-scale standardized *in vivo* survey of physiological activity in the mouse visual cortex, featuring representations of visually evoked calcium responses from GCaMP6f-expressing neurons. In this work, we use the population neural responses to a set of 118 natural image stimuli, each presented 50 times. The images were presented for 250ms each, with no inter-image delay or intervening "gray" image. The neural responses we use are events detected from fluorescence traces using an L0 regularized deconvolution algorithm, which deconvolves pointwise events assuming a linear calcium response for each event and penalizes the total number of events included in the trace. Full information about the experiment is and database given in [de Vries et al., 2020].

The Allen Brain Observatory includes data from six different brain areas, namely VISp, VISal, VISl, VISpm, VISam and VISrl. The number of neurons in the dataset, for each of the regions we use, is summarized in Table 3.12.

Table 3.12: **Number of neurons recorded from each mouse brain region.**

	VISp	VISal	VISl	VISpm	VISam	VISrl
Total	14173	4396	8748	4771	2040	5189
L2/3	4079	1042	2259	1544	610	1168
L4	6735	2967	4163	1905	1179	3626
L5	3003	387	1874	973	251	395

The Similarity of Similairy Matrices metric (SSM)

To compare functional similarity betweeen two representations – in MouseNet, and in the biological mouse brain – of a set of images, we use the Similarity of Similarity matrices (SSM) [Diedrichsen and Kriegeskorte, 2017, Shi et al., 2019] metric. We begin with a matrix of neural activities, in which each row contains the population activities for a certain image. We calculate the Pearson correlation coefficient between every pair of rows within one representation matrix, to form an n by n "similarity matrix" for this representation, where each entry describes the similarity of the population response to a pair of images. Next, to compare two similarity matrices, we flatten the matrices to vectors and compute the Spearman rank correlation between these vectors. Like the Pearson correlation coefficient, the rank correlation lies in the range $[-1, 1]$ indicating how similar (close to 1) or dissimilar (close to -1) the two representations are. Rather than examining one neuron at a time [Olah et al., 2017, Pospisil et al., 2018], this metric compares representations based on activities of the whole populations of artificial and biological neurons, revealing functional similarity at the population level. Another choice of such population similarity metrics is Singular Vector Canonical Correlation Analysis (SVCCA)[Raghu et al., 2017, Shi et al., 2019]. An excellent review of such similarity metrics and their properties can be found in [Kornblith et al., 2019].

Following the procedures in [Shi et al., 2019], we construct the representation matrix for a certain mouse visual cortex region by taking the trial-averaged mean responses of the neurons in the 250ms during the image presentation. Activities of neurons in different experiments for the same brain area are grouped together to construct the representation matrix, whose dimension is number of images by number of neurons. The representation matrices for MouseNet layers are obtained from feeding the same set of 118 images (resized to 64x64) to MouseNet and collecting all the activations from a certain layer of the model.

Neural reliability and SSM noise ceiling

We next compute the SSM noise ceiling from the Allen Brain Observatory data. We use split half reliability to quantify the reliability of a single neuron from the Allen Brain Observatory. This is done by separating the 50 trials into two non-overlapping 25 trial sets, and taking the correlation of trial-averaged responses between the two. We make ten random splits, and take the mean of the ten correlations to represent the reliability of each neuron. The reliability distributions of the neural populations are shown in Fig 3.4 (left). VISp, VISl and VISal are most reliable areas and VISpm, VISam and VISrl are less reliable areas.

To estimate the noise ceiling of the SSM metric, we compare the mouse data representation matrices with themselves. Specifically, we split the 50 trials in the dataset into two non-overlapping sets and calculate the trial averaged representation matrices for each set. The SSM between these two representation matrices are the noise ceiling of the SSM metric. Multiple splits of the dataset are computed for estimating the mean and standard deviation of the noise ceilings.

To examine how the noise ceiling changes with the reliability of the neural population, we calculate the noise ceilings by selecting neurons that surpass different levels of thresholds, as shown in Fig 3.4 (right). We see that for some regions, if we select a group of neurons using a certain reliability threshold, the noise ceiling becomes higher than without this selection. We summarize the reliability and best noise ceiling for each area in Fig 3.5. In this paper, we will concentrate our discussions on the most reliable areas, VISp, VISl and VISal, which are included in the MouseNet model. We will use the best noise ceiling to compare with the models.

Task training improves the similarity between MouseNet and the Allen Brain Observatory

To examine the effect of training to perform an image classification task on the functional similarity of MouseNet to the brain, we compute the SSM value between each layer of MouseNet with data from a brain region recorded in the Allen Brain Observatory. To

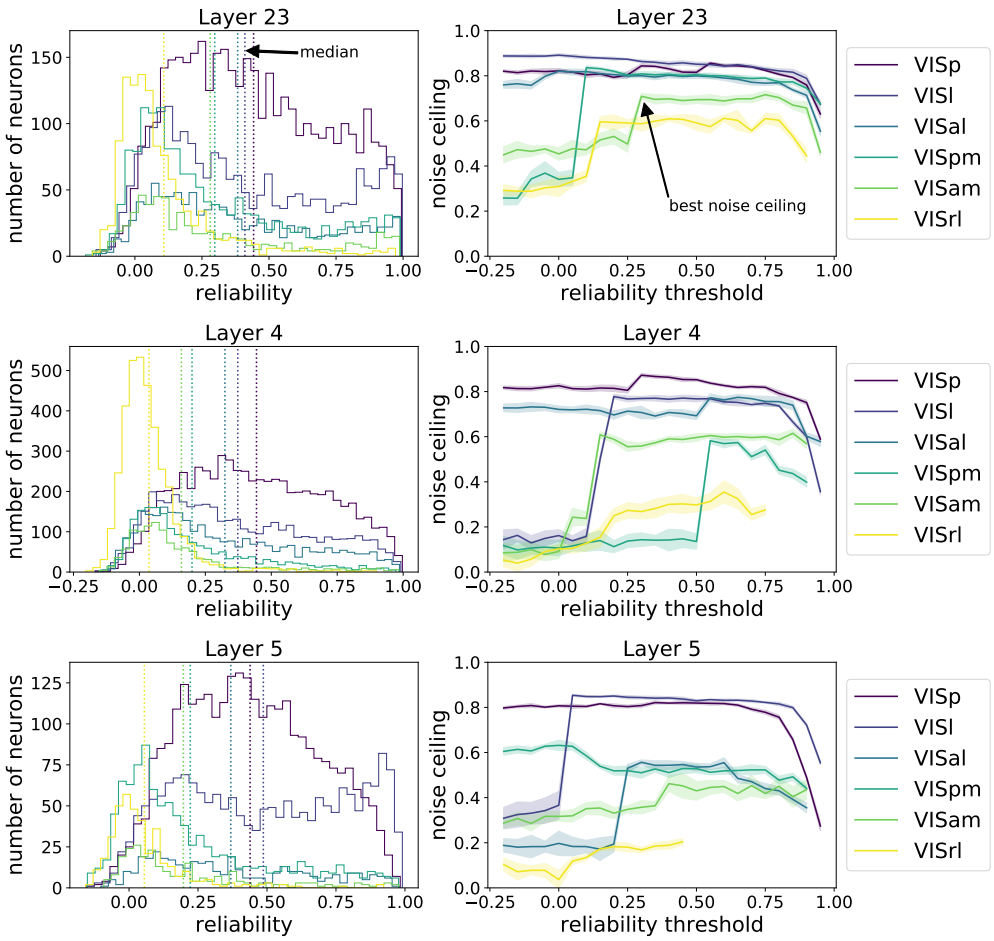

Figure 3.4: **Selecting reliable neurons improves noise ceilings.** (Left) Reliability distribution of neural populations. Each row shows all the brain areas at a specific cortical layer. The dotted lines indicate the median reliability of each neural population. (Right) The noise ceilings change with variation of the threshold for selecting reliable neurons. The higher the threshold, the fewer neurons are selected. For some populations, selecting a certain portion of reliable neurons gives best noise ceiling. Error bars are from different draws of non-overlapping trials.

account for the randomness due to initialization, we train four instances of MouseNet on ImageNet starting with different weights and look at their mean statistics. Fig. 3.6 shows

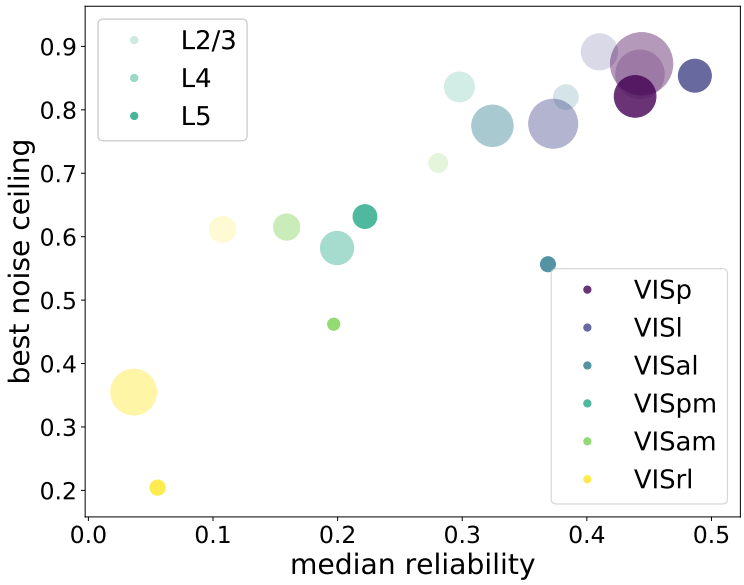

Figure 3.5: **Summary plot of median reliability and best noise ceiling for each brain area.** Each color represents a different brain area, and shades from light to dark indicate different cortical layers L2/3, L4 and L5. The circle size is proportional to the size of the population in the dataset.

the SSM values between each of the MouseNet layers with data from L2/3 of VISp, VISl and VISal. Layers 4 and 5 are shown in Fig. 3.7. The first important observation is that regions in the model do not necessarily best match to the corresponding functional area recorded in the Allen Brain Observatory. We see that for layer 2/3, area VISp in the Allen Brain Observatory, five different model areas show significant change in SSM value from the untrained model. In the following, we will add prefix "m" in front of the modeled areas from the MouseNet to contrast with the ones from the real brain. One of these is an early layer, mVISp5, while the others are in the parallel pathway portion of the architecture. Of the others, mVISl4 shows an increase in similarity with VISp_layer23, while three other model regions show a decrease in similarity. For the other two regions in Figure 3.6, mVISp5 shows a significant increase in similarity. For VISl_layer23, there are six other model regions

that all show an increase in similarity. These statements hold specifically when comparing model regions to each other for the same area in the Brain Observatory. Comparing areas of the Brain Observatory to each other requires a different adjustment for the number of comparison (see black vs. red stars in Figure 3.6). These results are consistent with the idea from Shi, et al [Shi et al., 2019] that VISp is a lower order area than VISl and VISal (VISp maps to lower "pseudo-depth" in comparing to a CNN than both VISl and VISal). Layers 4 and 5 show results that are similar, but not identical to, layer 2/3. (Fig. 3.7). VISal in Layer 4 and VISl and VISal in Layer 5 show improved similarity after training for many of the mVISp model regions. Similarly, VISp in layer 4 and 5 shows decreased similarity after training in some of regions in the parallel portion of the architecture.

Note that, although training on ImageNet improves the corresponding level of model regions' similarity to the brain, the highest SSM value does not always occur in the model layer corresponding to the specific region considered in the Brain Observatory. For example, the SSM value for mVISp regions are higher than the mVISl regions when comparing to the brain area VISl L2/3. This is possibly because the visual areas are more similar to each other than they are to the MouseNet regions (see Table 3.13 for the SSM values between the brain areas themselves), such that improving the similarity to one brain region can possibly lead to improving the similarity to some other regions. Nevertheless, by looking at all the layers globally, we see that for the earliest visual area VISp, the ImageNet classification training promotes the SSM values of the mVISp layers in the MouseNet while suppress the values for the later layers; whereas for secondary visual areas VISl, the training promotes both earlier layers and later layers in the parallel pathways, suggesting a higher place in the functional hierarchy (cf. with the results of [Shi et al., 2019]).

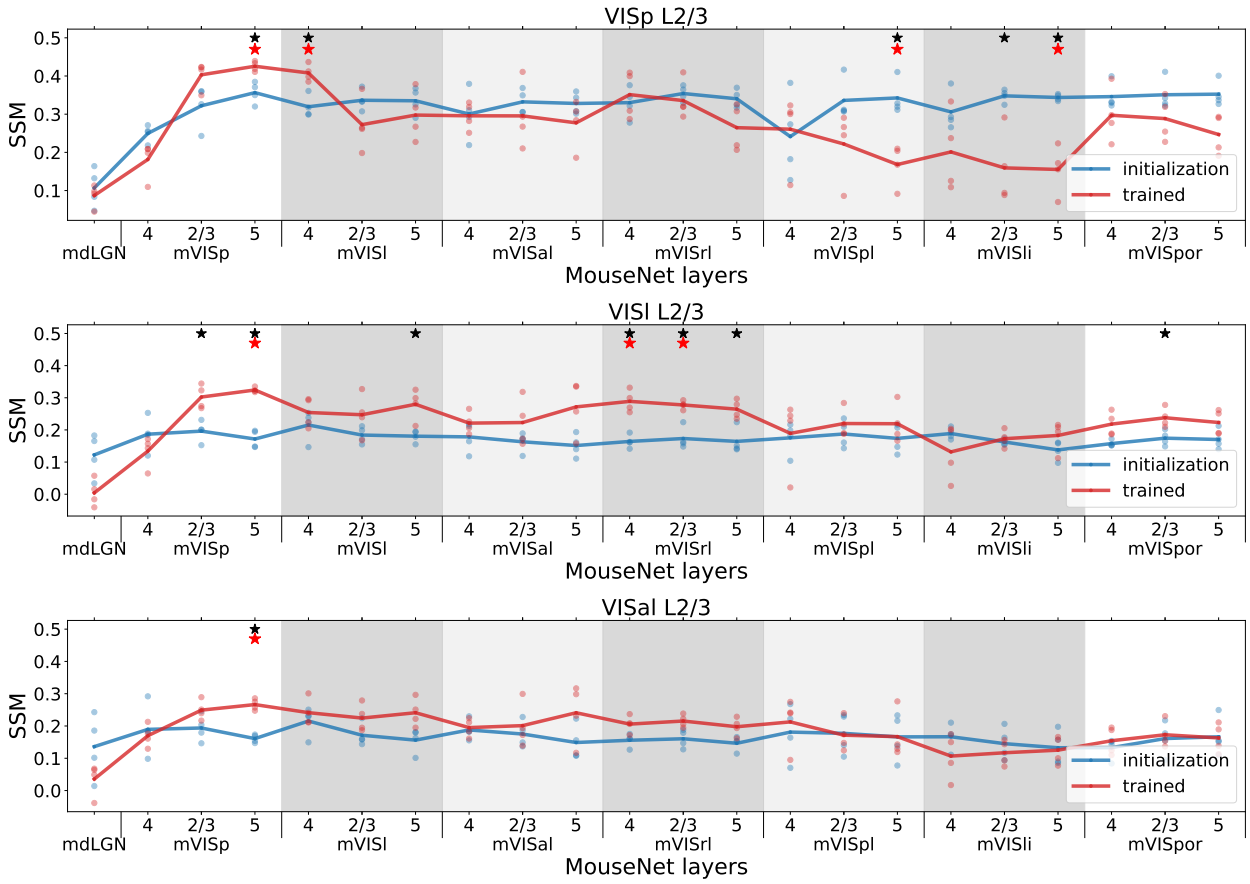

Figure 3.6: **SSM between mouse data in VISp(top)/VISl(middle)/VISal(bottom) L2/3 and all layers in the MouseNet before (blue) and after training (red).** Each line corresponds to the mean of four different MouseNet instances trained from different initialization weights (dots). The x axis includes all the layers in the model in a serial way. The five parallel secondary visual area pathways in the model are in shaded grey background. Black stars denote the the pvalues of two-sample t-test with Benjamini/Hochberg correction of 22 comparisons within one brain area is less than 0.05; Red stars denote the pvalues of two-sample t-test with Benjamini/Hochberg correction of all 9x22 comparisons across all 9 brain areas is less than 0.05.

Table 3.13: **SSM values between mouse visual cortical areas.** Note that even with the neural sub-sampling issue [Shi et al., 2019], the similarity values between VISp, VISl, and VISal are much higher than they are with the CNN models.

	VISp	VISl	VISal	VISpm	VISam	VISrl
VISp	1	0.56	0.60	0.35	0.23	0.25
VISl	0.56	1	0.51	0.35	0.24	0.25
VISal	0.60	0.51	1.	0.39	0.24	0.30
VISpm	0.35	0.35	0.39	1.	0.19	0.13
VISam	0.23	0.24	0.24	0.19	1	0.14
VISrl	0.25	0.25	0.30	0.13	0.14	1

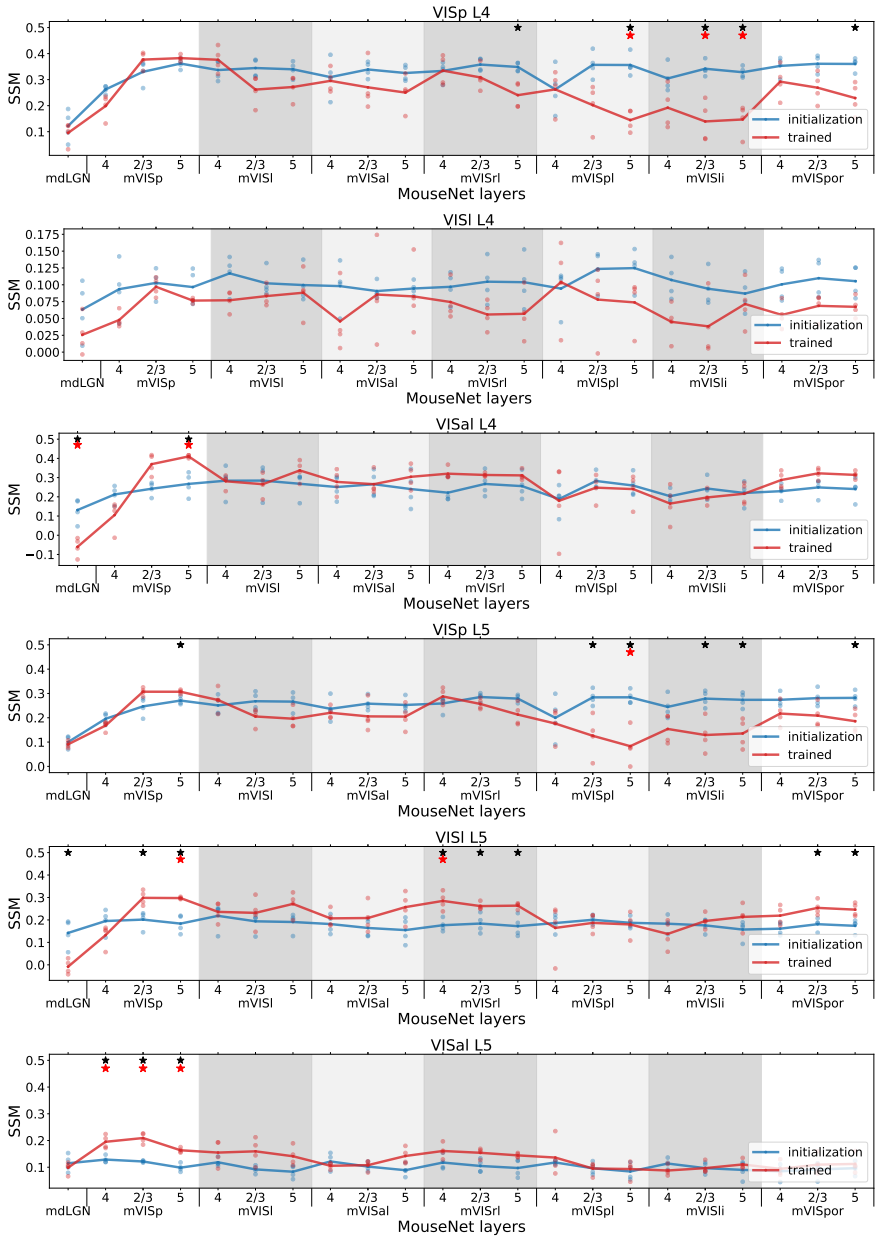

Figure 3.7: **SSM between mouse data in VISp(top)/VISl(middle)/VISal(bottom) L4 and L5 and all layers in the MouseNet before(blue) and after training(red).** Each line corresponds to the mean of 4 different MouseNet instances trained from different initialization weights (dots). The x axis includes all the layers in the model in a serial way. The five parallel secondary visual area pathways in the model are in shaded grey background. Black stars denote the the pvalues of two-sample t-test with Benjamini/Hochberg correction of 22 comparisons within one brain area is less than 0.05; Red stars denote the pvalues of two-sample t-test with Benjamini/Hochberg correction of all 9x22 comparisons across all 9 brain areas is less than 0.05.).

Higher task performance on image classification does not guarantee higher similarity to the mouse brain

To examine how performance on the ImageNet classification task affects the functional similarity to the brain, we contrast the SSM values for MouseNet with another network that can perform this task, the VGG16 network discussed above. We use the same input resolution, on the same task (see Section Computational Performance of MouseNet on image classification). Similarly as for MouseNet, we calculate the SSM values between each layer in VGG16 and the regions in the mouse visual cortex. VGG16 does not have a "corresponding layer" for each region; we choose the VGG layer that has the highest SSM with each mouse brain region. For this comparison, we do the same for MouseNet, so that for each region, we compare this 'best layer' SSM value with the best layer SSM value for MouseNet.

The best layer's SSM values for both VGG16 and MouseNet, for each mouse cortical layer in VISp, VISl, and VISal, are summarized in Fig 3.8. As we can see in the figure, although VGG16 has much higher performance on the ImageNet task (about 60% vs 40%), it does not have much higher SSM values to the brain for most regions. The saturation of functional similarity between the brain and models in terms of image classification performance is also observed in primates, albeit at a much higher performance level [Schrimpf et al., 2018].

To further grasp the limited relationship between a model's task performance and its functional similarity to the mouse brain, we look at how the models' functional similarity to brain data changes during training. As shown in Fig 3.9, the highest SSM values between a model neural network and the mouse brain areas are not necessarily achieved by the best performing models, rather at early or intermediate points during the training process. See Fig.3.10 for more instances of MouseNet during training, also showing this effect. These results show that optimizing performance on this particular task, at least beyond an early or intermediate level of performance, does not necessarily improve the model's similarity to the biological brain. If the approach of building models for neural responses via task training of artificial networks is broadly correct, then we take this as an indication that ImageNet is

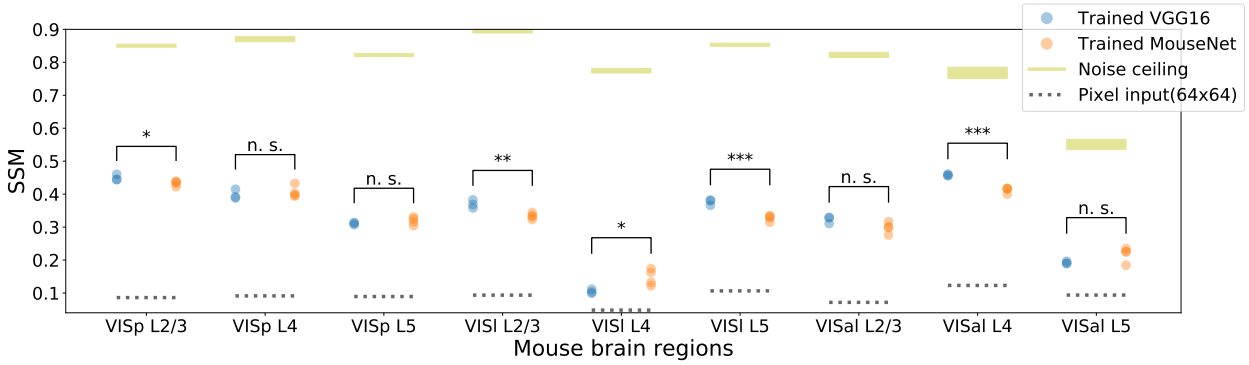

Figure 3.8: **SSM between best layer in trained VGG16/MouseNet and mouse brain regions.** The plot shows results of 3 instances of VGG16 (with validation accuracy 60.46, 60.72, 60.93) and 4 instances of MouseNet (with validation accuracy 37.46, 37.95, 37.52, 37.49) trained from different initialization weights. Yellow lines denote the best noise ceiling; their widths are standard deviations calculated from multiple draws of non-overlapping trials as in Fig.3.4. Dotted black lines are the SSM values between the 64x64 pixel input and the corresponding regions. Black stars denote the statistical significance of two-sample t-test between the mean of the trained VGG16 and the trained MouseNet instances (one star: $p < 0.05$, two stars: $p < 0.01$, three stars: $p < 0.001$).

not the correct task to consider for the representations in the mouse brain.

Task training with the MouseNet architecture increases the similarity of other functional properties to the mouse brain

As mentioned above, the SSM metric compares functional representations, based on activities of the whole neural population in a given model layer and a set of recordings from a given brain area. For a complementary view of the effect of task training on MouseNet representations, and of the role of its architecture, we can also study the statistical distributions of single neuron functional properties, such as orientation selectivity and lifetime sparseness [de Vries et al., 2020].

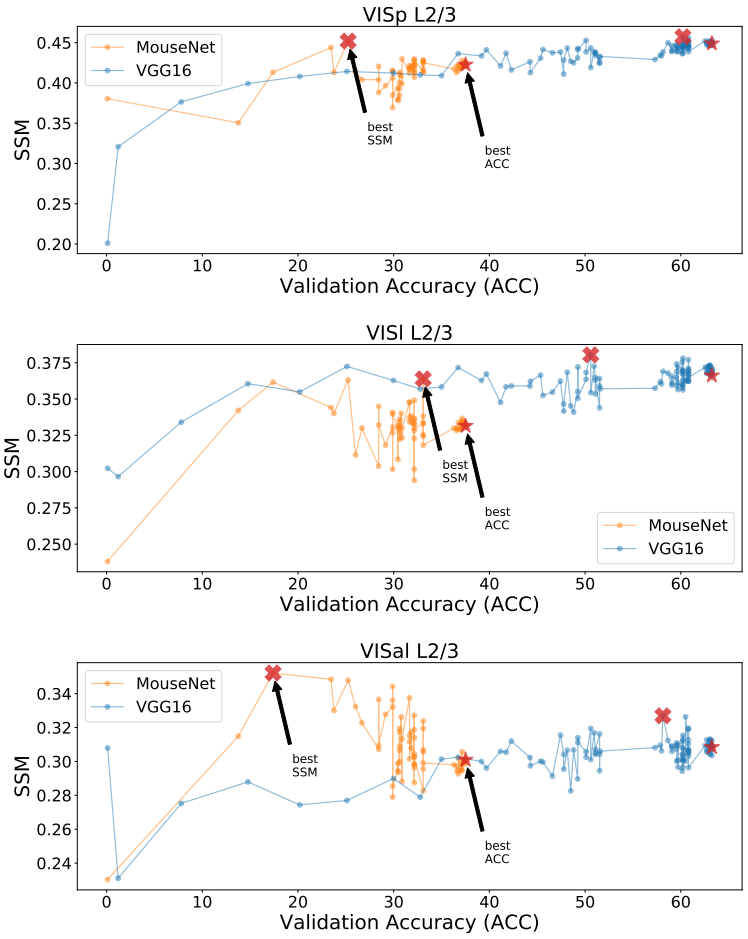

Figure 3.9: **Functional similarity and validation accuracy during the training process.** Each row compares models with a different brain area. We show one instance of MouseNet and VGG16 during their training process, where each dot represents the best layer's SSM of one model at a certain epoch to the specified brain area. The clear jumps of validation accuracy occurred when the learning rate is reduced.

Lifetime sparseness measures the selectivity of a neuron's mean response to different stimulus condition, defined as [Vinje and Gallant, 2000, de Vries et al., 2020]

$$S_L = \left(1 - \frac{1}{N}\frac{(\sum_i r_i)^2}{\sum_i r_i^2}\right) / \left(1 - \frac{1}{N}\right) \tag{3.9}$$

where N is the number of stimulus conditions and r_i is the response of the neuron to stimulus

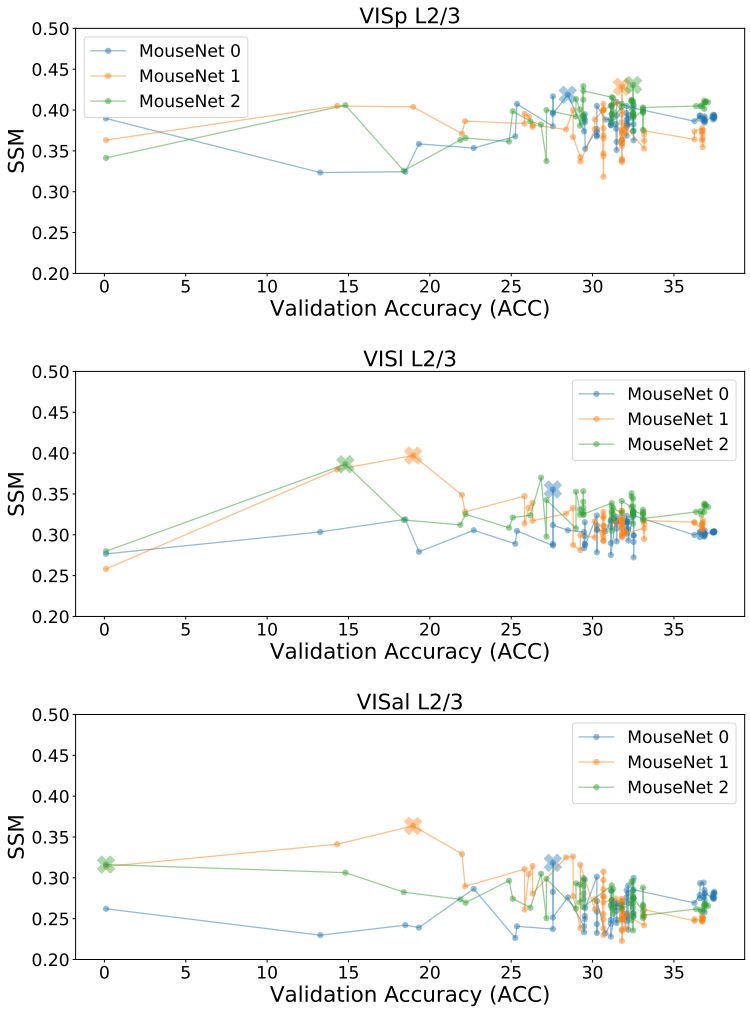

Figure 3.10: **Functional similarity and validation accuracy during the training process for multiple MouseNet instances.** Each row compares models with a different brain area. We show three instances of MouseNet during their training process. Each dot represents the best layer's SSM of one instance at a certain epoch to the specified brain area, with each instance's highest achieved SSM during training process marked by a cross. The clear jumps of validation accuracy occurred when we reduced the learning rate.

condition i averaged across trials. A neuron that responds strongly to only a few stimuli will have a lifetime sparseness close to 1, whereas a neuron that responds broadly to many stimuli

will have a lower lifetime sparseness. The statistical distribution of lifetime sparseness for the mouse data on natural scene stimuli and for all the units in trained/untrained MouseNet and VGG16 models, responding to the same natural scene stimuli as in the Allen Brain Observatory, are shown in Fig. 3.11 (top row). This demonstrates clearly that training on the image classification task makes the distribution of lifetime sparseness values much closer to the mouse brain data for MouseNet, but not as much for VGG16.

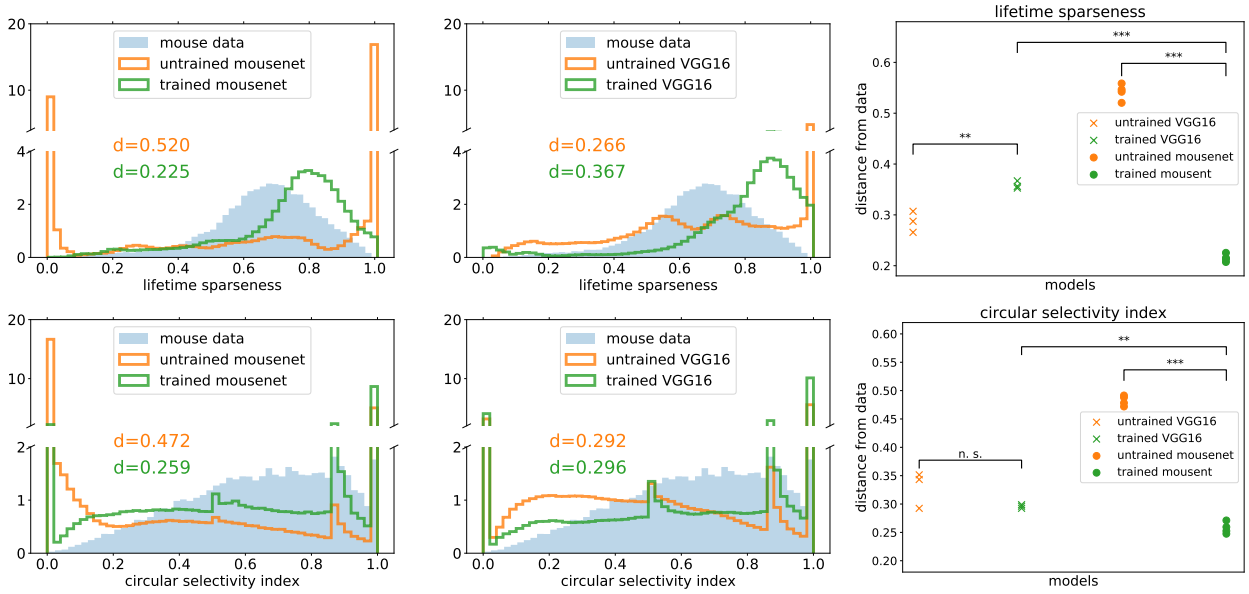

Figure 3.11: **Distributions of lifetime sparseness (top row) and circular selectivity index (bottom row) for all the units in the models and all the neurons in the mouse data.** The distributions of all units in one instance of trained/untrained MouseNet (first column) and VGG16 (second column) are plotted along with mouse data, with the Jensen-Shannon distances between the models and the data annotated. The Jensen-Shannon distances between multiple instances of models and the mouse data are summarized in the third column. Black stars denote the statistical significance of two-sample t-test between the mean of the model instances (one star: $p < 0.05$, two stars: $p < 0.01$, three stars: $p < 0.001$).

Similarly, we can study the orientation selectivity of individual neurons by using the

static grating stimuli in the Allen Brain Observatory dataset. Specifically, we calculate the circular selectivity index (which is one minus the circular variance defined in [Ringach et al., 2002]), defined as

$$S_O = \sum_k r_k e^{i2\theta_k} / \sum_k r_k \tag{3.10}$$

where r_k is the response of the neuron to a grating with angle θ_k averaged across trials. A neuron that responds strongly to only one direction will have circular selectivity index close to 1, whereas a neuron that responds broadly to many directions will have lower circular selectivity index. The statistical distributions of the circular selectivity index, for the mouse data with static grating stimuli and for trained/untrained MouseNet and VGG16 models with the same stimuli, are shown in Fig. 3.11 (bottom row). As for the case of lifetime sparsity above, task training changes the distribution of selectivity values. These distributions, after training, are closer to the mouse brain data for the MouseNet networks than for the VGG, once again showing how the more specifically matched architecture of MouseNet can lead to more similar model responses to the biological brain. Note that the spikier distributions of the models result from the deterministic nature of the models in contrast to the noisier brain data in response to the (only) six total static grating directions. If we were to simulate neural noise in the model responses, it would smooth the distributions, resulting in closer approximation of the data, as we show in Fig 3.12.

Taken together, these results show how the MouseNet model can be used to explore the impact of task training on a variety of response statistics that are commonly computed in physiology studies, and that those defined on individual neurons can demonstrate complementary and in some cases more dramatic changes with training than those averaged over entire populations.

Task training diversifies functional representation among MouseNet layers

Finally, we study how task training and network architecture affect the general 'geometric' layout of models' representations, separately from their similarity to representations in the

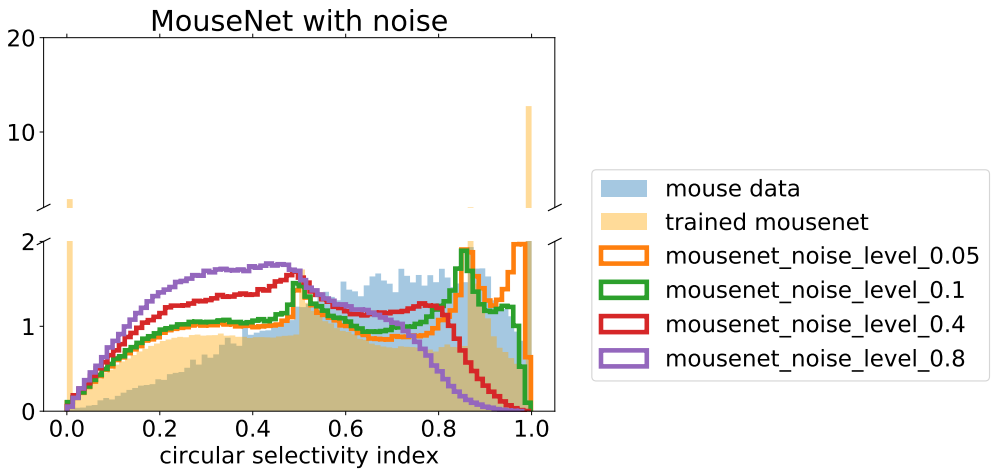

Figure 3.12: **Distribution of circular selectivity index for all the units in trained MouseNet with different levels of noise added.** The noise is added to the activations of each layer as a half-normal distribution with a standard deviation of the specified noise level multiplied by the mean activation across all units for that layer. This results shows that circular selectivity index distribution can be smoothed out by adding noise to the deterministic MouseNet model.

mouse brain data. To do this, we calculate the SSM values between every pair of layers from both trained/untrained MouseNet and VGG16, and visualize them in two dimensional space via a metric multidimensional scaling algorithm [Pedregosa et al., 2011, Borg and Groenen, 2005]. The results are shown in Fig.3.13. For VGG16, we see that representations in layers are clustered together both before and after training. By contrast, for MouseNet the representations become much more diversified after training. We hypothesize that it is the parallel architecture of MouseNet that leads it to learn this more diversified representation as it solves the image classification task. Further examinations of the various pathways and model instances show that different pathways are learning quite different representations (Fig 3.14, and that these qualitative results are consistent across multiple instances of MouseNet models (Fig 3.15. Unraveling any specific functions of each pathway, in this task

or in others, is an opportunity left for future studies.

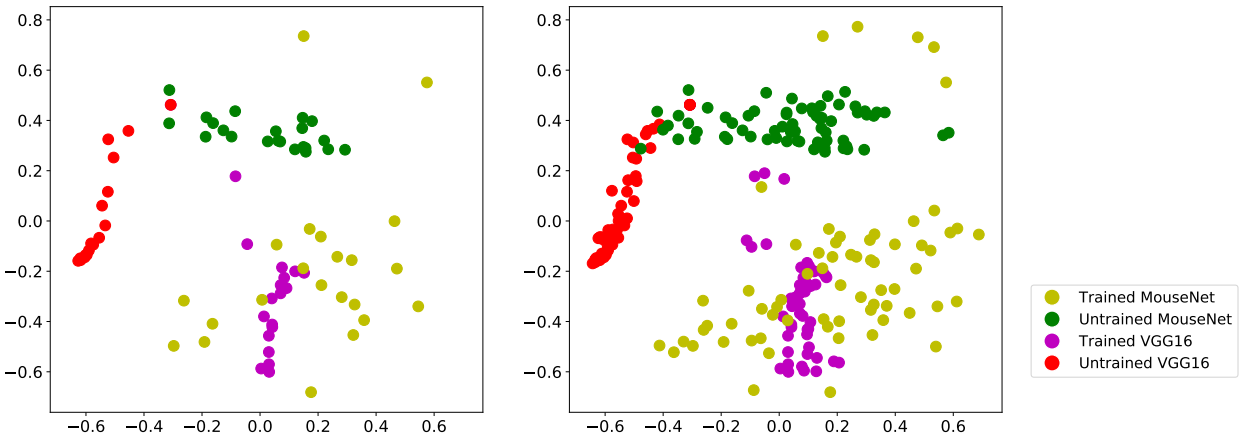

Figure 3.13: **Visualization of all layers from one instance (left) and three instances (right) of trained/untrained MouseNet and VGG16.** Each dot represents a layer from a certain model instance. The position of the dots are the two-dimensional projection from the multidimensional scaling algorithm, with the distance measure defined as one minus the SSM value.

3.4 Discussion

Task-optimized deep networks show promise for brain modelling, because they are functionally sophisticated, and they often develop internal representations that overlap strongly with representations in the brain [Lindsay, 2020, Yamins et al., 2014, Yamins and DiCarlo, 2016, Kell et al., 2018, Khaligh-Razavi and Kriegeskorte, 2014b, Zhuang et al., 2017a, Sandbrink et al., 2020, Michaels et al., 2020]. While deep network architectures are originally loosely inspired by the brain, there has been an extensive empirical exploration of the effects of architectural features in machine learning, in directions often independent from neuroscience. In parallel, however, a great deal more has been learned about the architecture of the biological brain, with that of the mouse brain having been been particularly well characterized.

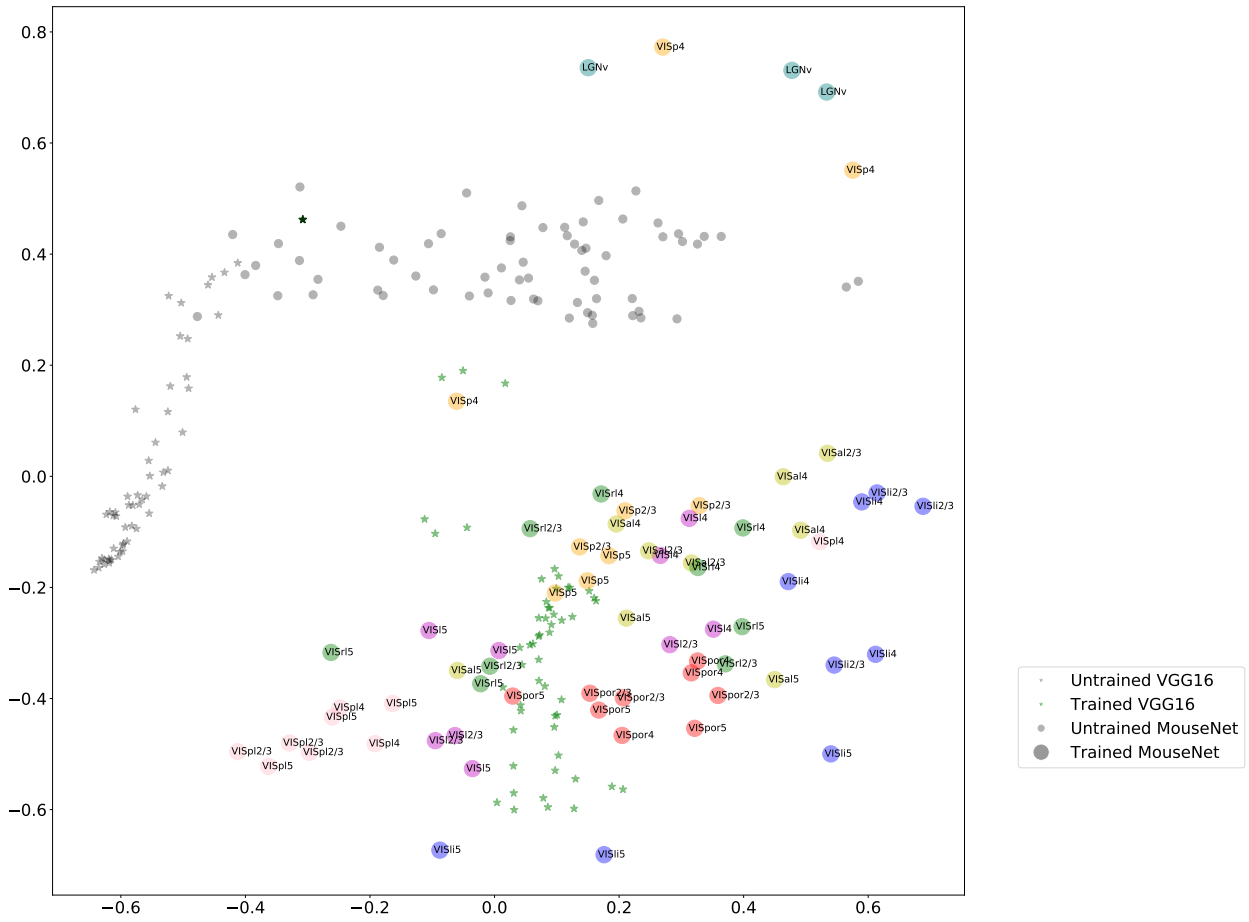

Figure 3.14: **Visualization of all layers of trained/untrained MouseNet and VGG16, for three instances (colored coded by areas).** Each dot represents a layer from a certain model instance. The position of the dots are the two-dimensional projection from the multidimensional scaling algorithm, with the distance measure defined as one minus the SSM value. The layers from three instances of trained MouseNet are color coded by their area names, and annotated with their region names. This result shows that different pathways in the MouseNet have learned distinct representations after training.

We have developed MouseNet, a deep network architecture that is consistent with a wide range of data on mouse visual cortex, including data from tract-tracing studies and

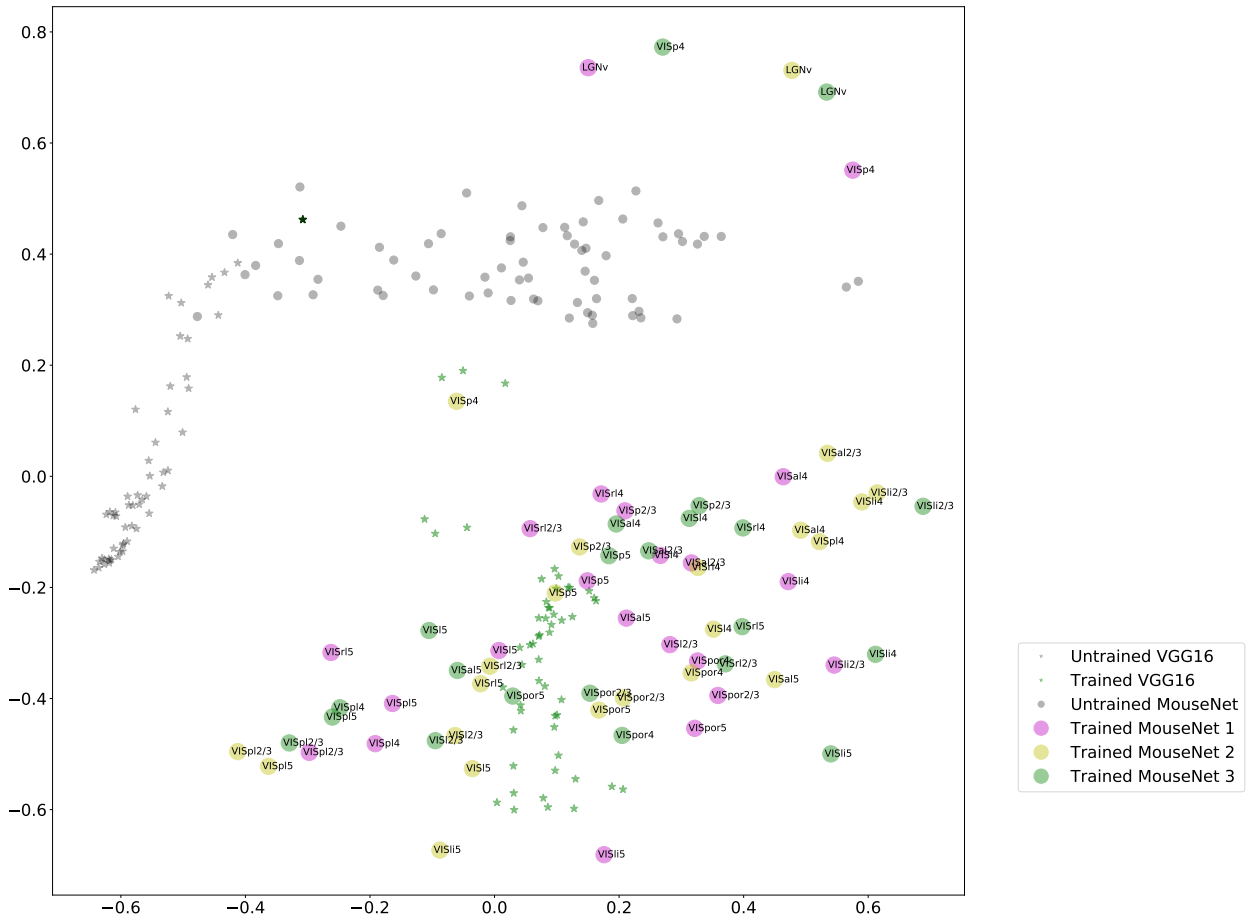

Figure 3.15: **Visualization of all layers of trained/untrained MouseNet and VGG16, for three instances (colored coded by instance).** Each dot represents a layer from a certain model instance. The position of the dots are the two-dimensional projection from the multidimensional scaling algorithm, with the distance measure defined as one minus the SSM value. The layers from three instances of trained MouseNet are color coded by their corresponding model instance. This result shows that training diversified the representations of all the three instances of MouseNet starting from different initialization states.

studies of local connection statistics. While standard deep networks have provided useful points of comparison with neurobiological systems, in the long term more biologically realistic deep networks may enable more specific comparisons with the brain, including comparisons between homologous groups of neurons, modeling of specific lesions, and analysis of functional differences between brain areas and pathways.

Using image classification as a working example, we use MouseNet to investigate using the task-training approach to model the functional representations in the mouse brain. An aspect of special interest is whether training on this task drives the representations in the model to be closer to those recorded from the real mouse brain, in comparison to representations in untrained versions of the MouseNet model or in generic deep networks. Using recordings from the large-scale Allen Brain Observatory survey, we find – consistent with the literature [Yamins et al., 2014, Yamins and DiCarlo, 2016] for other model species and systems – that training on an image classification task does drive MouseNet representations to better resemble those of the real data. However, this increase of functional similarity is not necessarily strictly monotonic with task performance. In our experiments we see the SSM correlation with the Brain Observatory responses saturating or even maximizing well before we achieve maximum accuracy on task performance. This is true for both MouseNet and VGG16.

Within the task-training paradigm, these results suggest that the specific image classification task we used, and perhaps image classification overall, is not the appropriate task for describing what the mouse visual cortex has learned and developed to compute. Nonetheless, MouseNet is an important reference to studies in more established species, which rely on comparisons of the ventral stream with architectures designed for object recognition. Although we know rodents are capable of performing tasks that require visual object discrimination, mouse ethology suggests that alternate computations are more important for the mouse visual system, such as motion tracking, predation, and predator avoidance. A promising future direction is to use task-training of the MouseNet model, together with the metrics tested here, to develop more realistic tasks and stimuli that may lead to more closely

matched representations.

In sum this work links anatomical and physiological data to task-driven CNN models, providing a road map for developing better task-driven models of the biological brain. It opens the door to building more detailed structures into the model, such as adding further brain areas as well as adding recurrence and using different inputs and readouts for different pathways. Incorporating new anatomical data is also easy within this framework. By making our code publicly available, and illustrating the model's success and failures in matching representations using well-studied metrics and tasks, we hope to facilitate future research along these lines.

Acknowledgments

We thank Tianqi Chen, Blake Richards, Shahab Bakhtiari, Graham Taylor for helpful discussions and suggestions on the manuscript. We thank Saskia de Vries, Hannah Choi, Kameron Decker Harris, Daniel Zdeblick, Timothy Lillicrap, Julie Harris, Severine Durand, and Jun Zhuang for helpful discussions. We thank the Allen Institute for Brain Science founder, Paul G. Allen, for his vision, encouragement, and support. We acknowledge the NIH Graduate training grant in neural computation and engineering (R90DA033461). Research reported in this publication was supported by the National Institute of Neurological Disorders and Stroke and the National Institute of Biomedical Imaging and Bioengineering at the National Institutes of Health under Award Number R01EB026908. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health.

INTERLUDE

In the previous chapter, we have constructed deep neural networks with anatomical constraints from the real mouse visual cortex. It is clear that the architecture of the mouse visual cortex has an interesting parallel structure rather than a single feed-forward pathway and we have shown that this parallel structure may lead to more diverse representations among the layers.

It is not easy to study what each pathway is doing in general for the complicated task, but we can get insights from studying settings with simpler network and task structures, where the learning process can be analytically quantified and analyzed.

In the following chapter, we use the simplest possible model with parallel pathways, and examine how the learning process distributes the task-related knowledge onto different pathways.

Chapter 4

KNOWLEDGE DISTRIBUTION IN DEEP LINEAR NETWORK WITH PARALLEL PATHWAYS

In this work, we study how knowledge—in terms of the singular values of the input-output correlation matrix—about a task is distributed through learning in networks with parallel pathways. We develop a systematic set of results in the setting of deep linear networks. We find that, for three layer networks: with a special initialization scheme, the final knowledge learned by each parallel pathway is exactly the same as at their initialization; For small random initialization scheme, the final learned knowledge by each pathway is proportional to its hidden layer size. For deeper networks with parallel pathways, we find that small differences in knowledge at the initialization of the pathways will be magnified across learning. This leads to large deviations in the final knowledge about task features among different pathways.

4.1 Introduction

In the primate visual system, it is well known that processing of visual information is coordinated by two parallel pathways, namely the "ventral" and the "dorsal" pathways [Felleman and van Essen, 1991, Goodale and Milner, 1992, Mishkin et al., 1983]. In the rodent visual system, anatomical analysis also suggests that there are parallel pathways [Harris et al., 2019, Wang et al., 2011, Wang et al., 2012], although the specific functional role of these pathways is less clear. Interestingly, recent research [Bakhtiari et al., 2021] has shown that self-supervised training in neural network architectures with parallel pathways can lead to the emergence of ventral-like and dorsal-like pathways. Another study shows that, with a similar self-supervised training objective, shallower architectures with parallel pathways

lead to closer matches with functional data in the Allen Brain Observatory than deep single pathway architectures [Nayebi et al., 2021]. Furthermore, an anatomically constrained deep neural network model for the mouse visual cortex has demonstrated that such parallel architectures have more diverse layer representations compared to single pathway architectures [Shi et al., 2021].

Additionally, a coincident observation has been made for the original AlexNet [Krizhevsky et al., 2012] deep neural network designed for image processing, which also shows the emergence of functional specialization across parallel pathways. Originally, the AlexNet was implemented on two separate Graphics Processing Units(GPUs) for extra computational power, which happened to mimic an architecture with two parallel pathways. It turned out that after learning, one of the pathway learned mostly monochrome filters while the other pathway learned mostly colored filters.

Overall, in the road map for developing future artificial intelligence systems, architectures with parallel pathways promise potential solutions for multi-tasking, multi-sensory, energy-efficient computing [Dean, 2021]. Despite the richness of information representation and the potential capabilities of architectures with parallel pathways, there is a lack of research in understanding the fundamental learning behavior that occurs in such architectures. Here we make an initial attempt to understand the behaviour of deep neural networks with parallel pathways, in the simple setting of deep linear networks.

Our starting point follows Saxe et al. [Saxe et al., 2019], which shows how, in such a linear network, the training data set for a given task can be fully described by a set of independent modes resulting from singular value decomposition (SVD) of the input-output correlation matrix. During gradient descent-based learning, a deep linear network will pick up the task related information from the data by acquiring knowledge — in terms of the singular values of the input-output correlation matrix— from each mode sequentially, from modes with largest singular values to the smallest.

In this work, we are interested in how the representations of the relevant features of the data that support the ability to make distinctions among examples are acquired by networks

with parallel pathways through learning. To answer that, we study how knowledge about the modes of input-output correlation matrix is learned by linear networks with parallel pathways. We begin with three layer networks, and find that, with a special initialization scheme, the final knowledge learned by each pathway is exactly the same as that at the initialization of network weights. For initialization schemes with small random weights, the final knowledge learned by each pathway is proportional to its hidden layer sizes. Interestingly, deeper networks with parallel pathways produce a different effect. Here, small differences of knowledge at the initialization of the pathways will be magnified, leading to large deviations in the final knowledge among different pathways.

4.2 Mathematical framework

In this section, we introduce the mathematical framework of this study.

4.2.1 Simplest network with parallel pathways

We start from the simplest possible network structure with parallel pathways: a three-layer (input, hidden, output) linear network with two parallel pathways. In mathematical form, the network is given by

$$\hat{y} = (W^{2a}W^{1a} + W^{2b}W^{1b})x \tag{4.1}$$

where $x \in R^{N_1}, \hat{y} \in R^{N_3}, h_a = W^{1a}x \in R^{N_{2a}}, h_b = W^{1b}x \in R^{N_{2b}}$. Figure 4.1 shows an illustration of the network. Note that this network is equivalent to a single pathway network with hidden layer size of $N_{2a} + N_{2b}$.

4.2.2 The gradient descent learning dynamics

Using the same setting as in Saxe et al. [Saxe et al., 2014], we train the network (Equation 4.1) to learn the input-output map given by a set of P examples $\{x^i, y^i\}, i = 1, 2, \ldots, P$ with gradient descent with the squared loss

$$L(W^{1a}, W^{2a}, W^{1b}, W^{2b}) = \frac{1}{2}\sum_{i=1}^{P} ||y^i - (W^{2a}W^{1a} + W^{2b}W^{1b})x^i||^2.$$

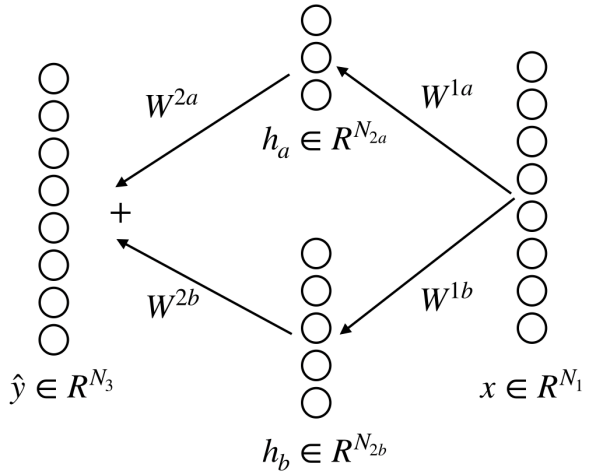

Figure 4.1: **Illustration of the three-layer two-pathway network.** The first layer input feeds into two hidden layers in parallel, whose outputs then feed into the third layer where they get summed up as the final output of the network.

For one training example, the gradient descent procedure yields the following update rule

$$\Delta W^{1a} = -\lambda \frac{\partial L}{\partial W^{1a}} = \lambda (W^{2a})^T \left(y^i - \hat{y}^i\right) x^{iT}$$

$$\Delta W^{2a} = -\lambda \frac{\partial L}{\partial W^{2a}} = \lambda \left(y^i - \hat{y}^i\right) (W^{1a} x^i)^T$$

$$\Delta W^{1b} = -\lambda \frac{\partial L}{\partial W^{1b}} = \lambda (W^{2b})^T \left(y^i - \hat{y}^i\right) x^{iT}$$

$$\Delta W^{2b} = -\lambda \frac{\partial L}{\partial W^{2b}} = \lambda \left(y^i - \hat{y}^i\right) (W^{1b} x^i)^T$$

For one epoch across all the examples, assuming the learning rate is small ($\lambda \ll 1$), the weights change minimally on each single example such that $W[i] \approx W(t)$ ($W[i]$ represents the weight on presentation of example i, $W(t)$ represents the weight within epoch t) for all examples within the t^{th} epoch. Denoting $E[yx^T] = \Sigma^{yx}, E[xx^T] = \Sigma^x$, the weight change

over one epoch is then given by

$$\begin{aligned}
\Delta W^{1a}(t) &= \sum_{i=1}^{P} \lambda (W^{2a}[i])^T \left(y^i - \hat{y}^i\right) x^{iT} \\
&= \sum_{i=1}^{P} \lambda (W^{2a}[i])^T \left(y^i - (W^{2a}[i]W^{1a}[i] + W^{2b}[i]W^{1b}[i])x^i\right) x^{iT} \\
&\approx \sum_{i=1}^{P} \lambda (W^{2a}(t))^T \left(y^i - (W^{2a}(t)W^{1a}(t) + W^{2b}(t)W^{1b}(t))x^i\right) x^{iT} \\
&= \lambda P (W^{2a}(t))^T (E[yx^T] - (W^{2a}(t)W^{1a}(t) + W^{2b}(t)W^{1b}(t))E[xx^T]) \\
&= \lambda P (W^{2a}(t))^T (\Sigma^{yx} - (W^{2a}(t)W^{1a}(t) + W^{2b}(t)W^{1b}(t))\Sigma^x)
\end{aligned}$$

$$\begin{aligned}
\Delta W^{2a}(t) &= \sum_{i=1}^{P} \lambda \left(y^i - \hat{y}^i\right) (W^{1a}[i]x^i)^T \\
&= \sum_{i=1}^{P} \lambda \left(y^i - (W^{2a}[i]W^{1a}[i] + W^{2b}[i]W^{1b}[i])x^i\right) (W^{1a}[i]x^i)^T \\
&\approx \sum_{i=1}^{P} \lambda \left(y^i - (W^{2a}(t)W^{1a}(t) + W^{2b}(t)W^{1b}(t))x^i\right) x^{iT} W^{1a}(t)^T \\
&= \lambda P (E[yx^T] - (W^{2a}(t)W^{1a}(t) + W^{2b}(t)W^{1b}(t))E[xx^T]) W^{1a}(t)^T \\
&= \lambda P (\Sigma^{yx} - (W^{2a}(t)W^{1a}(t) + W^{2b}(t)W^{1b}(t))\Sigma^x) W^{1a}(t)^T
\end{aligned}$$

Similarly

$$\begin{aligned}
\Delta W^{1b}(t) &\approx \lambda P (W^{2b}(t))^T (\Sigma^{yx} - (W^{2a}(t)W^{1a}(t) + W^{2b}(t)W^{1b}(t))\Sigma^x) \\
\Delta W^{2b}(t) &\approx \lambda P (\Sigma^{yx} - (W^{2a}(t)W^{1a}(t) + W^{2b}(t)W^{1b}(t))\Sigma^x) W^{1b}(t)^T
\end{aligned}$$

When the learning rate λ is sufficiently small, we can take the continuum limit of the

above difference equations to be

$$\tau \frac{d}{dt} W^{1a} = W^{2aT}(\Sigma^{yx} - \Omega\Sigma^{x}) \tag{4.2}$$

$$\tau \frac{d}{dt} W^{2a} = (\Sigma^{yx} - \Omega\Sigma^{x}) W^{1aT} \tag{4.3}$$

$$\tau \frac{d}{dt} W^{1b} = W^{2bT}(\Sigma^{yx} - \Omega\Sigma^{x}) \tag{4.4}$$

$$\tau \frac{d}{dt} W^{2b} = (\Sigma^{yx} - \Omega\Sigma^{x}) W^{1bT} \tag{4.5}$$

where

$$\Omega \equiv W^{2a} W^{1a} + W^{2b} W^{1b} \tag{4.6}$$

and the time constant

$$\tau \equiv \frac{1}{P\lambda} \tag{4.7}$$

4.2.3 Quantification of task related knowledge

For the ease of analysis, we assume the input correlation is the identity, i.e., $\Sigma^x = I$; one way to justify this, beyond the desire for simplicity, is to assume that upstream processing has "whitened" inputs. As outlined above, the task related information can be described by a set of independent modes of the input-output correlation matrix Σ^{yx}. We begin by taking the singular value decomposition of $\Sigma^{yx} = USV^T$, where U, V are orthogonal matrices satisfying $U^TU = V^TV = I$. The size of U is $N_3 \times N_3$. The size of V is $N_1 \times N_1$. S has size $N_3 \times N_1$ with decreasing non-zero elements s_α only on the diagonal. Its singular value on mode α represents the amount of input-output *knowledge* about mode α the network needs to learn for this task.

We now use the simple task introduced in Saxe et al. [Saxe et al., 2019] to further illustrate the idea (Figure 4.2). In this example, the inputs are the identity of a set of items, and the outputs are the properties of the items. Taking the SVD of the input output correlation matrix Σ^{yx}, we get four independent modes which in combination explain the whole data set. The first mode is shared by all the four items, representing things that can grow. The second mode distinguishes "plants/animals" by properties "move/roots". The third

mode distinguishes "bird/fish" by properties "fly/swim". The fourth mode distinguishes "flower/tree" by properties "leaves/petals". The singular values corresponding to each mode relate to the prevalence of that mode in the data. For example, the second mode describes properties for all of the four items (Canary, Salmon, Oak, Rose) while the third mode only describes properties for two items (Canary, Salmon).

Saxe et al. [Saxe et al., 2019] have shown that during gradient descent learning, a single pathway deep linear network will pick up the task related information from the data by acquiring knowledge from each mode sequentially, from modes with largest singular values to the smallest. In the current work, we investigate how knowledge about these modes is divided among multiple parallel pathways.

To demonstrate how knowledge can be distributed into parallel pathways, Figure 4.2 second row shows a possible split of the matrix S into the summation of two matrices K^a and K^b, with $K^a + K^b = S$. In this case, the knowledge of the first two modes is equally learned by the two pathways, while the third mode is only learned by pathway b and the fourth mode is only learned by pathway a. As a result, both pathways can differentiate plants/animals, while only pathway b can differentiate bird/fish and only pathway a can differentiate flower/tree.

To put this in mathematical form, let

$$\overline{W}^{1a} = W^{1a}V,\ \overline{W}^{2a} = U^T W^{2a},\ \overline{W}^{1b} = W^{1b}V,\ \overline{W}^{2b} = U^T W^{2b} \tag{4.8}$$

then we have

$$\hat{\Sigma}^{yx} = (W^{2a}W^{1a} + W^{2b}W^{1b})\Sigma^x = U(\overline{W}^{2a}\overline{W}^{1a} + \overline{W}^{2b}\overline{W}^{1b})V^T \tag{4.9}$$

and recall that

$$\Sigma^{yx} = USV^T \tag{4.10}$$

Since $\hat{\Sigma}^{yx}$ is tending towards Σ^{yx} through learning, the values of $\overline{W}^{2a}\overline{W}^{1a} + \overline{W}^{2b}\overline{W}^{1b}$ are tending towards S. We are interested in how S is split into $K^a = \overline{W}^{2a}\overline{W}^{1a}$ and $K^b = \overline{W}^{2b}\overline{W}^{1b}$ through the gradient descent dynamics.

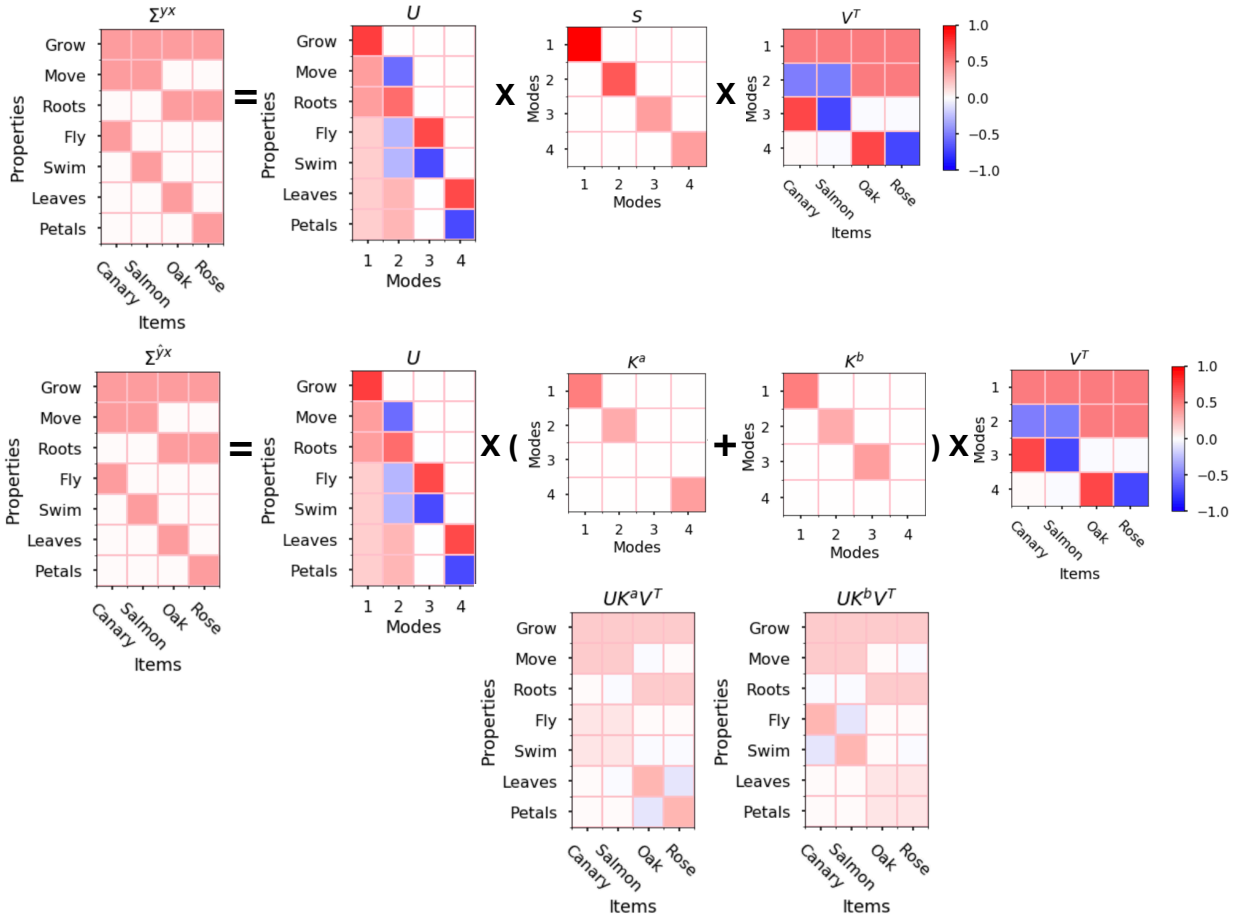

Figure 4.2: **Illustration of the general multiple pathway learning idea (see text).**

Although the summation of K^a and K^b will be diagonal after learning, K^a and K^b themselves are not diagonal in general. In this work, we concentrate on two special initialization schemes—namely diagonal knowledge initialization and small random initialization— where after learning, both K^a and K^b are about diagonal, so that the amount of knowledge in each mode is well represented by a single scalar. In those cases, the final knowledge learned by pathway a and pathway b can be analyzed theoretically and can be controlled accordingly through network initialization.

4.3 Learning in three-layer linear networks with parallel pathways

After projecting weight matrices via the singular vectors of the data matrix (Equation 4.8), the gradient descent dynamics is given by

$$\tau \frac{d}{dt} \overline{W}^{1a} = \overline{W}^{2aT} (S - \overline{\Omega}) \tag{4.11}$$

$$\tau \frac{d}{dt} \overline{W}^{2a} = (S - \overline{\Omega}) \overline{W}^{1aT} \tag{4.12}$$

$$\tau \frac{d}{dt} \overline{W}^{1b} = \overline{W}^{2bT} (S - \overline{\Omega}) \tag{4.13}$$

$$\tau \frac{d}{dt} \overline{W}^{2b} = (S - \overline{\Omega}) \overline{W}^{1bT} \tag{4.14}$$

where

$$\overline{\Omega} \equiv \overline{W}^{2a} \overline{W}^{1a} + \overline{W}^{2b} \overline{W}^{1b} \tag{4.15}$$

Denoting the α-th column vectors of $\overline{W}^{1a}, \overline{W}^{2aT}, \overline{W}^{1b}, \overline{W}^{2bT}$ as $m^\alpha \in R^{N_{2a}}, n^\alpha \in R^{N_{2a}}, p^\alpha \in R^{N_{2b}}, q^\alpha \in R^{N_{2b}}$ (see Figure 4.3, note that $\overline{W}^{1a}, \overline{W}^{1b}$ has N_1 columns whereas $\overline{W}^{2aT}, \overline{W}^{2bT}$ has N_3 columns), we can rewrite the above equations to be

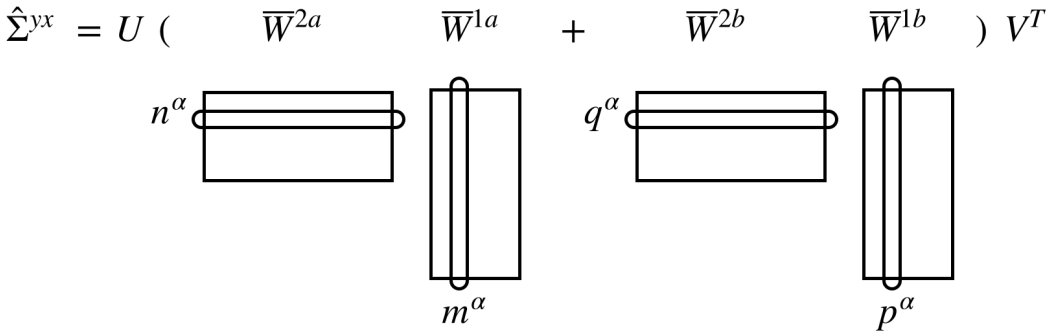

Figure 4.3: **Illustration of column and row notation for projected weight matrices.**

$$\tau \frac{d}{dt} m^\alpha = (s_\alpha - \omega^\alpha) n^\alpha - \sum_{\gamma \neq \alpha} n^\gamma (m^\alpha \cdot n^\gamma + p^\alpha \cdot q^\gamma) \tag{4.16}$$

$$\tau \frac{d}{dt} n^\alpha = (s_\alpha - \omega^\alpha) m^\alpha - \sum_{\gamma \neq \alpha} m^\gamma (n^\alpha \cdot m^\gamma + q^\alpha \cdot p^\gamma) \tag{4.17}$$

$$\tau \frac{d}{dt} p^\alpha = (s_\alpha - \omega^\alpha) q^\alpha - \sum_{\gamma \neq \alpha} q^\gamma (m^\alpha \cdot n^\gamma + p^\alpha \cdot q^\gamma) \tag{4.18}$$

$$\tau \frac{d}{dt} q^\alpha = (s_\alpha - \omega^\alpha) p^\alpha - \sum_{\gamma \neq \alpha} p^\gamma (n^\alpha \cdot m^\gamma + q^\alpha \cdot p^\gamma) \tag{4.19}$$

where

$$\omega^\alpha \equiv m^\alpha \cdot n^\alpha + p^\alpha \cdot q^\alpha \tag{4.20}$$

The dynamics has an energy function of the form

$$E = \frac{1}{2\tau} \sum_\alpha (s_\alpha - w^\alpha)^2 + \frac{1}{2\tau} \sum_{\substack{\alpha,\beta \\ \alpha \neq \beta}} (m^\alpha \cdot n^\beta + p^\alpha \cdot q^\beta)^2 \tag{4.21}$$

The first term in the energy function shows that the contributions from both pathways on the same mode $(m^\alpha \cdot n^\alpha, p^\alpha \cdot q^\alpha)$ will cooperatively learn the input-output mode strength s_α after learning. The second term shows that the interaction between different modes from the two pathways $(m^\alpha \cdot n^\beta, p^\alpha \cdot q^\beta)$ tend to cancel each other after learning.

In this section, we will analyze two special initialization schemes where the learned knowledge matrices $K^a = \overline{W}^{2a} \overline{W}^{1a}$ and $K^b = \overline{W}^{2b} \overline{W}^{1b}$ are diagonal. We call the first initialization scheme "diagonal knowledge initialization", where the knowledge matrices for each pathway is initialized as diagonal matrices and remain diagonal through the learning dynamics. We name the second initialization scheme "small random initialization", where the weight matrices are randomly initialized with sufficiently small values that will converge, as we show, to a version of the "diagonal knowledge initialization" during the beginning phase of the dynamics.

4.3.1 Diagonal knowledge initialization

For the diagonal knowledge initialization, we take the initial conditions of the form $m^\alpha \propto n^\alpha \propto r_a^\alpha (\alpha = 1, \ldots, \min(N_1, N_3)), p^\alpha \propto q^\alpha \propto r_b^\alpha (\alpha = 1, \ldots, \min(N_1, N_3))$, where $r_a^\alpha \cdot r_a^\beta = \delta_{\alpha\beta}, r_b^\alpha \cdot r_b^\beta = \delta_{\alpha\beta}$. With such an initialization, $m^\alpha(p^\alpha)$ will start off being parallel to $n^\alpha(q^\alpha)$ while orthogonal to $m^\beta, n^\beta(p^\beta, q^\beta, \beta \neq \alpha)$ and stay that way through time. In other words, the knowledge matrices $K^a = \overline{W}^{2a}\overline{W}^{1a}$ and $K^b = \overline{W}^{2b}\overline{W}^{1b}$ will start off as diagonal matrices and remain as diagonal matrices through time. Thus we can simply look at the evolution of the length of each vector.

Let $m = m^\alpha \cdot r_a^\alpha, n = n^\alpha \cdot r_a^\alpha, p = p^\alpha \cdot r_b^\alpha, q = q^\alpha \cdot r_b^\alpha$ and $s = s^\alpha$, then the dynamics of the scalar projections (m, n, p, q) becomes

$$\tau \frac{d}{dt} m = n(s - (mn + pq)) \tag{4.22}$$

$$\tau \frac{d}{dt} n = m(s - (mn + pq)) \tag{4.23}$$

$$\tau \frac{d}{dt} p = q(s - (mn + pq)) \tag{4.24}$$

$$\tau \frac{d}{dt} q = p(s - (mn + pq)) \tag{4.25}$$

with energy function

$$E = \frac{1}{2\tau}(s - (mn + pq))^2 \tag{4.26}$$

Note that

$$\frac{d}{dt}(m^2 - n^2) = 0, \quad \frac{d}{dt}(p^2 - q^2) = 0 \tag{4.27}$$

Thus the dynamics of both pathways follows hyperbolas of constant $m^2 - n^2$ and $p^2 - q^2$ and approaches the hyperbolic-shaped manifold of fixed points $mn + pq = s$. The origin is an unstable fixed point.

Further assuming $m = n, p = q$ will lead to a simple case (dropping this assumption leads to less straightforward expressions, but these can still be solved analytically, see Section

4.6.1). Letting $k_a = mn = m^\alpha \cdot n^\alpha, k_b = pq = p^\alpha \cdot q^\alpha$, we obtain

$$\tau \frac{dk_a}{dt} = 2k_a(s - (k_a + k_b)), \quad \tau \frac{dk_b}{dt} = 2k_b(s - (k_a + k_b)) \tag{4.28}$$

Notice that

$$\frac{d}{dt}\left(\frac{k_a}{k_b}\right) = 0 \tag{4.29}$$

In this case, the origin is an unstable fixed point, and $k_a + k_b = s$ is a stable line attractor (Figure 4.4).

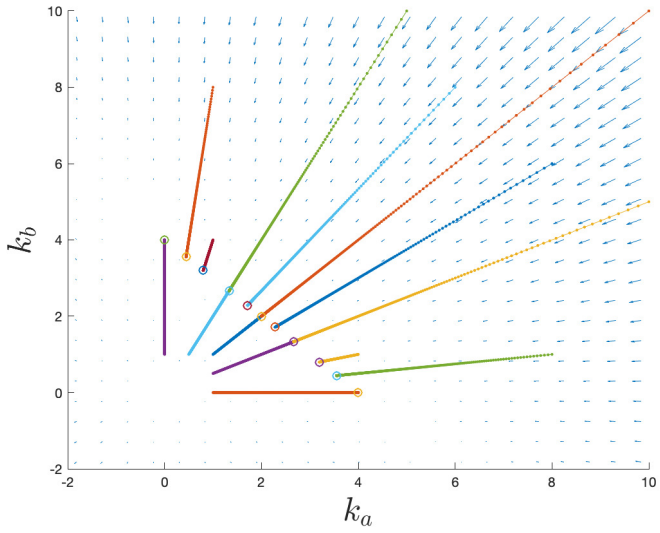

Figure 4.4: **Dynamics of acquired knowledge for a three-layer network with diagonal knowledge initialization.** The evolution of the acquired knowledge by the two pathways k_a, k_b with the simplifying assumption $m = n, p = q$, governed by Equation 4.28. The trajectories converge to $k_a + k_b = s$ and the ratio k_a/k_b remains constant through time.

Recall that k_a and k_b are the diagonal values of the knowledge matrices $K^a = \overline{W}^{2a}\overline{W}^{1a}$ and $K^b = \overline{W}^{2b}\overline{W}^{1b}$, so that they represent the amount of *knowledge* that is learned by each pathway about a certain input-output mode. Equation 4.29 shows that the ratio between

k_a and k_b does not change through time, which means that in this simple case the ratio of the final learned knowledge between the two pathways is the same as the ratio at their initialization. Since the final k_a and k_b after learning satisfy $k_a + k_b = s$, these final values can be directly obtained from the corresponding values at initialization.

Controlling knowledge distribution through diagonal knowledge initialization

Using the assumptions for the simple case in Section 4.3.1, we initialize the weight matrices with

$$W^{1a} = \overline{W}^{1a} V^T = R_a \cdot D_a \cdot V^T, \tag{4.30}$$

$$W^{2a} = U \overline{W}^{2a} = U \cdot D_a \cdot R_a^T, \tag{4.31}$$

$$W^{1b} = \overline{W}^{1b} V^T = R_b \cdot D_b \cdot V^T, \tag{4.32}$$

$$W^{2b} = U \overline{W}^{2b} = U \cdot D_b \cdot R_b^T \tag{4.33}$$

where $R_a(N_{2a} \times N_{2a})$, $R_b(N_{2b} \times N_{2b})$ are matrices with orthogonal columns $R_a^T R_a = I$, $R_b^T R_b = I$, and $D_a(N_{2a} \times N_1)$, $D_b(N_{2b} \times N_1)$ are diagonal matrices. From previous analysis, we know that when the training converges, we have

$$K^a / K^b = D_a^2 / D_b^2 \tag{4.34}$$

In other words, as above the ratios of the final learned knowledge between the two pathways is the same as the ratio at their initialization. Thus by specifying the ratio at the initialization, we can control the amount of knowledge each pathway carries at the end of learning.

In Fig.4.5, we illustrate two different learning dynamics resulting from two different initialization specifications. In the first case (top row), we initialize all the modes with a fixed knowledge ratio not far from one. As we have shown, this ratio will remain constant during the learning process. At the end of learning, both pathways learned knowledge about all the eight modes in the task. Correspondingly, a metric multidimensional scaling (MDS) visualization [Pedregosa et al., 2011, Borg and Groenen, 2005] on the hidden layers of both pathways reveal that they have learned a similar structure.

In the second case (bottom row), we initialize two pathways to have a dramatic difference in their knowledge of different input-output modes. Specifically, pathway a is initialized with large amount of knowledge for the first four modes and zero knowledge for the last four modes; pathway b is initialized with large amount of knowledge for the last four modes and zero knowledge for the first four modes. We can see that the corresponding knowledge ratios are maintained across learning. Moreover, the MDS plots show that, at the end of learning, pathway a has learned the general relation between the input data (eight inputs are separated into four categories, which can be further separated into two groups) but cannot differentiate all the 8 input examples. At the same time, pathway b can differentiate all the input examples but does not encode the relationship among the data.

4.3.2 Small random initialization of network weights

Small random initialization with large hidden layers leads to convergence to the special, diagonal cases

In this section, we consider random network initialization with small weights, i.e., the elements of $W^{1a}, W^{2a}, W^{1b}, W^{2b}$ are i.i.d. generated with $\mathcal{N}(0, \epsilon^2)$.

With those small initial weights, by taking the leading term of Equation 4.16-4.17, the beginning period of the dynamics for Equation 4.16 can be approximated by the following simplified equations (here we only write equations for m^α, n^α, since p^α, q^α governed by the same equations):

$$\tau \frac{d}{dt} m^\alpha = s_\alpha n^\alpha \tag{4.35}$$

$$\tau \frac{d}{dt} n^\alpha = s_\alpha m^\alpha \tag{4.36}$$

with solution

$$m^\alpha(t) = \cosh\left(\frac{s_\alpha}{\tau} t\right) m^\alpha(0) + \sinh\left(\frac{s_\alpha}{\tau} t\right) n^\alpha(0) \tag{4.37}$$

$$n^\alpha(t) = \sinh\left(\frac{s_\alpha}{\tau} t\right) m^\alpha(0) + \cosh\left(\frac{s_\alpha}{\tau} t\right) n^\alpha(0) \tag{4.38}$$

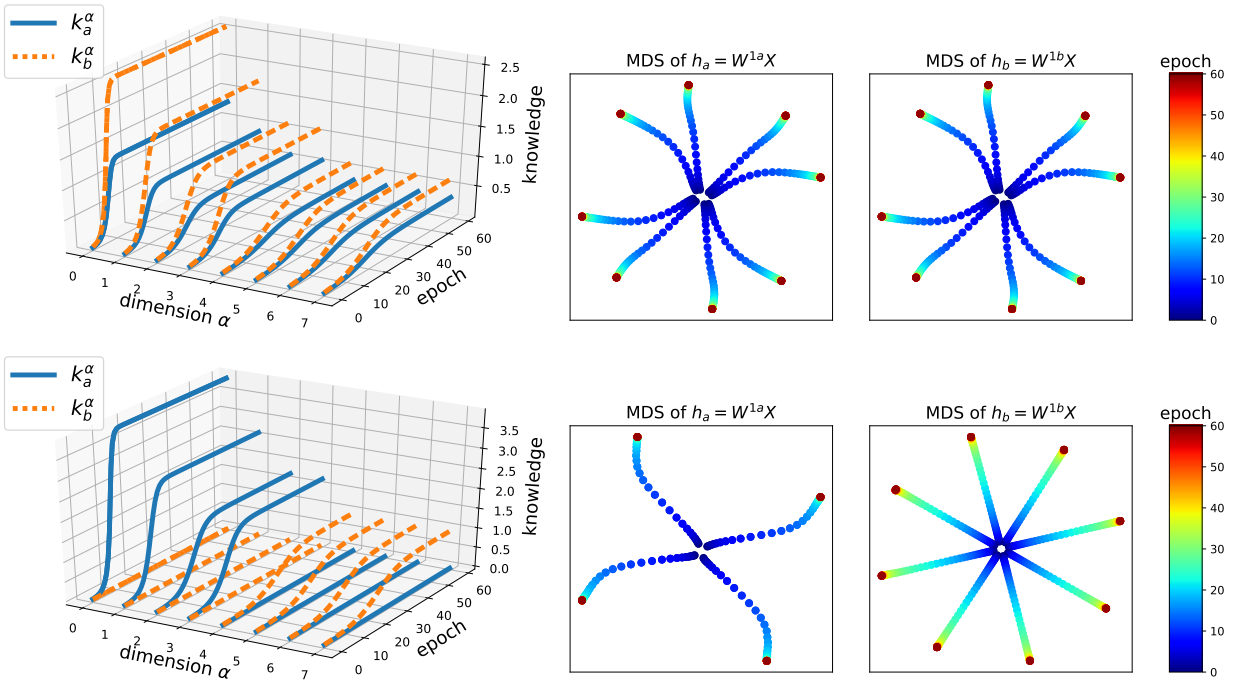

Figure 4.5: **Controlling the dynamics of knowledge distribution among parallel pathways through different diagonal initializations.** Two different specifications of initialization lead to shared (top) vs. distinct (bottom) knowledge distribution among the two pathways. The left panels show the increase of knowledge (k_a, k_b) at each mode α for each pathway through time. The right panels are the multidimensional scaling (MDS) visualization of the hidden layer representations of eight different inputs for pathway a (left) and pathway b (right) during learning. The curve show how the hidden representations of the different inputs diverge during the learning process.

Thus

$$m^\alpha(t), n^\alpha(t) \to \frac{e^{\frac{s_\alpha}{\tau}t}}{2}\left(m^\alpha(0) + n^\alpha(0)\right)(t \to \infty) \tag{4.39}$$

Notice that the speed with which $m^\alpha(t), n^\alpha(t)$ become aligned is independent of the scale of the initialization. Within time $T \sim O(\tau/s_\alpha)$, $m^\alpha(t)$ and $n^\alpha(t)$ can become close to parallel to each other, with the same length.

Within this beginning period $T \sim O(\tau/s_\alpha)$, if we want the approximation in Equation 4.35 to remain valid, we need

$$s_\alpha n^\alpha >> \sum_{\gamma \neq \alpha} (m^\alpha \cdot n^\gamma + p^\alpha \cdot q^\gamma) n^\gamma \tag{4.40}$$

and

$$s_\alpha >> w^\alpha = m^\alpha \cdot n^\alpha + p^\alpha \cdot q^\alpha \tag{4.41}$$

If the size of the hidden layers for both pathways are large enough such that $N\epsilon^2 >> 0$, then $m^\alpha(0)$ will be approximately perpendicular to $m^\beta(0)$:

$$E[m^\alpha(0) \cdot m^\alpha(0)] = N_{2a}\epsilon^2 \tag{4.42}$$

$$E[m^\alpha(0) \cdot m^\beta(0)] = 0 \tag{4.43}$$

Thus $m^\alpha(T)$ will remain perpendicular to $m^\beta(T)$ according to the solution (Equation 4.39), and the inequality in Equation 4.40 is guaranteed. For Equation 4.41, note that

$$m^\alpha(T) \cdot n^\alpha(T) \approx \frac{e^{\frac{2s_\alpha}{\tau}T}}{4} \left| m^\alpha(0) + n^\alpha(0) \right|^2 \approx \frac{e^{\frac{2s_\alpha}{\tau}T}}{2} \left| m^\alpha(0) \right|^2 \tag{4.44}$$

$$E[m^\alpha(T) \cdot n^\alpha(T)] \approx \frac{e^{\frac{2s_\alpha}{\tau}T}}{2} N_{2a}\epsilon^2 \tag{4.45}$$

thus we need

$$\frac{e^{\frac{2s_\alpha}{\tau}T}}{2}\epsilon^2(N_{2a} + N_{2b}) << s_\alpha, \;\; \epsilon << O\left(\sqrt{\frac{s_\alpha}{N_{2a} + N_{2b}}}\right) \tag{4.46}$$

In other words, for ϵ which satisfies

$$N_{2a}\epsilon^2 >> 0, \;\; N_{2b}\epsilon^2 >> 0, \;\; \epsilon << O\left(\sqrt{\frac{s_\alpha}{N_{2a} + N_{2b}}}\right) \tag{4.47}$$

m^α and n^α will become equal (parallel and with same length) within the beginning time period $T \sim O(\tau/s_\alpha)$.

Thus for large pathways with small random initialization, the beginning period of the dynamics will lead to the simple case in Section 4.3.1. An example of the dynamics in the beginning period is illustrated in Figure 4.6.

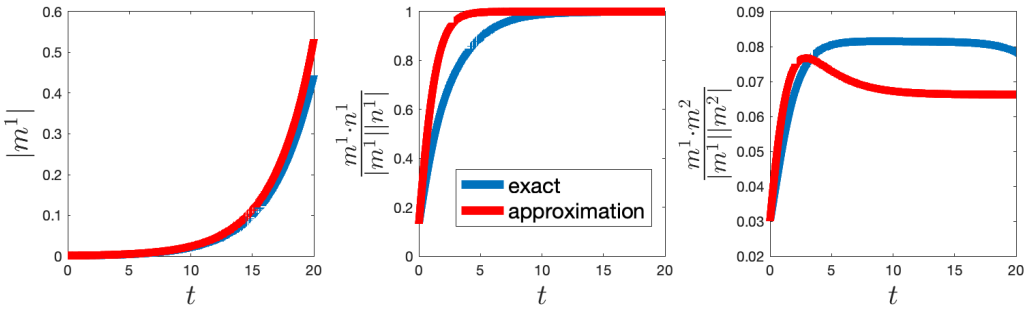

Figure 4.6: **Example dynamics in the beginning period.** Within the beginning period of time scale $O(\tau/s_\alpha)$, the approximation of Equation 4.35 remains valid and approximates the exact solution well (left); m^α and n^α become equal exponentially fast (middle); m^α and n^β start small and remain small for large hidden layers (right).

Thus the ratio between the knowledge learned by the two pathways is approximately their relative hidden layer sizes, since

$$k_a^\alpha = m^\alpha(t) \cdot n^\alpha(t) \approx m^\alpha(T) \cdot n^\alpha(T) \approx m^\alpha(T) \cdot m^\alpha(T) \tag{4.48}$$

$$k_a^\alpha / k_b^\alpha \approx m^\alpha(T) \cdot m^\alpha(T) / \left(p^\alpha(T) \cdot p^\alpha(T)\right) \tag{4.49}$$

$$\approx |m^\alpha(0) + n^\alpha(0)|^2 / |p^\alpha(0) + q^\alpha(0)|^2 \tag{4.50}$$

$$E[k_a^\alpha / k_b^\alpha] \approx E[|m^\alpha(0)|^2] / E[|p^\alpha(0)|^2] \approx N_{2a}/N_{2b} \tag{4.51}$$

In sum, we have shown that for networks with sufficiently small random initialization and large hidden layers such that $\epsilon << 1$ and $N\epsilon^2 >> 0$, the final learned knowledge by the two pathways is proportional to the sizes of their hidden layers, that is,

$$E[k_a^\alpha / k_b^\alpha] \approx N_{2a}/N_{2b} \tag{4.52}$$

Simulation result of small random initialization and large hidden layers

In the following, we show the results of gradient descent simulations for the more general case discussed in the previous section. We randomly sample networks with chosen scale of

initialization $\mathcal{N}(0, \epsilon^2)$ and different hidden layer sizes for the two pathways. We then use gradient descent to learn the task and examine the final learned knowledge ratio between the two pathways. We verify that, when the scales of ϵ and N_{2a}, N_{2b} meets the assumption that $\epsilon << 1$ and $N\epsilon^2 >> 0$, then the simulation matches well with the prediction of Equation 4.52. On the other hand, if the scale of ϵ and N_{2a}, N_{2b} deviate from the assumptions, then as expected the simulation results also deviate from the prediction (Figure 4.7). Overall, the trend that final knowledge ratios increase with relative hidden layer sizes is clearly visible, and becomes more precise as predicted for smaller ϵ and larger N.

4.4 Learning in deeper linear networks with parallel pathways

In this section, we look at the case where both of the pathways have multiple hidden layers, while still forming the final layer through concatenation, i.e.,

$$\hat{y} = (W^{N_w a} \cdots W^{2a} W^{1a} + W^{N_w b} \cdots W^{2b} W^{1b}) x \tag{4.53}$$

where each pathway has $N_l = N_w + 1$ layers with N_w weight matrices. The gradient descent dynamics can be written as follows

$$\tau \frac{d}{dt} W^{la} = \left(\prod_{i=l+1}^{N_w} W^{ia} \right)^T \left[\Sigma^{yx} - \left(\prod_{i=1}^{N_w} W^{ia} + \prod_{i=1}^{N_w} W^{ib} \right) \Sigma^x \right] \left(\prod_{i=1}^{l-1} W^{ia} \right)^T \tag{4.54}$$

$$\tau \frac{d}{dt} W^{lb} = \left(\prod_{i=l+1}^{N_w} W^{ib} \right)^T \left[\Sigma^{yx} - \left(\prod_{i=1}^{N_w} W^{ia} + \prod_{i=1}^{N_w} W^{ib} \right) \Sigma^x \right] \left(\prod_{i=1}^{l-1} W^{ib} \right)^T \tag{4.55}$$

where $\prod_{i=l_1}^{l_2} W^i = W^{l_2} W^{l_2 - 1} \cdots W^{l_1}$ when $l_1 \leq l_2$ and $\prod_{i=l_1}^{l_2} W^i = I$ when $l_1 > l_2$.

4.4.1 Diagonal knowledge initialization

As before, we consider the case where input correlation is the identity, i.e., $\Sigma^x = I$. As with the three layer case, a special initialization that can be analyzed analytically is when the weight matrices W^l at initialization can be diagonalized by a set of orthogonal matrices $R_l^T R_l = I$ such that $R_{l+1}^T W^l(0) R_l = D_l$ for both pathways, with $R_1 = V$ and $R_{N_w+1} = U$.

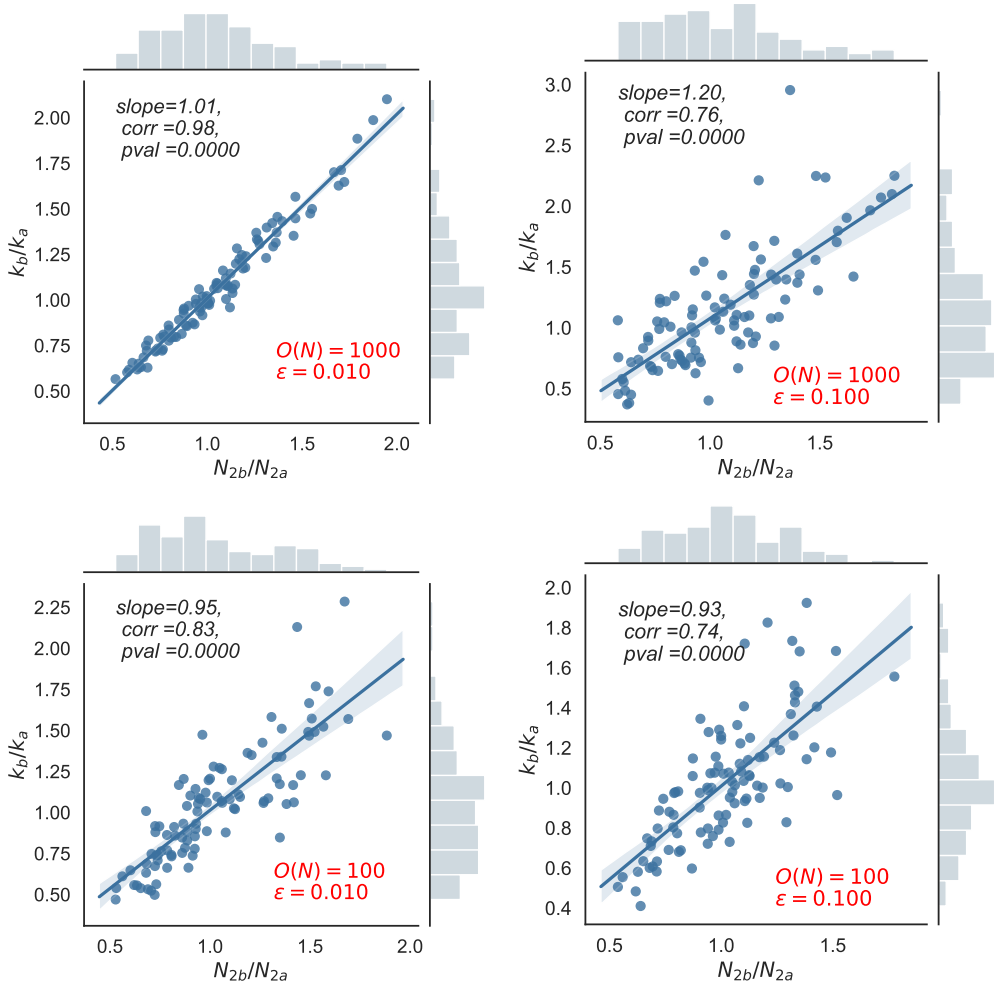

Figure 4.7: **Simulation results for small random initialization with large hidden layers.** Learned knowledge ratio versus hidden layer size ratio between two pathways. Each dot represents a simulation result with specified scale of initialization ϵ and hidden layer size N_{2a}, N_{2b}. We see that the simulations closely match the prediction of Equation 4.52 when $\epsilon << 1$ and $N\epsilon^2 >> 0$ (upper left). When ϵ increases (right) or N decreases (bottom), the simulation match becomes less precise.

Thus after change of the variables $W^{la} = R_{l+1,a}\overline{W}^{la}R_{l,a}^T, W^{lb} = R_{l+1,b}\overline{W}^{lb}R_{l,b}^T$, we get

$$\tau\frac{d}{dt}\overline{W}^{la} = \left(\prod_{i=l+1}^{N_w}\overline{W}^{ia}\right)^T\left[S - \left(\prod_{i=1}^{N_w}\overline{W}^{ia} + \prod_{i=1}^{N_w}\overline{W}^{ib}\right)\right]\left(\prod_{i=1}^{l-1}\overline{W}^{ia}\right)^T \tag{4.56}$$

$$\tau\frac{d}{dt}\overline{W}^{lb} = \left(\prod_{i=l+1}^{N_w}\overline{W}^{ib}\right)^T\left[S - \left(\prod_{i=1}^{N_w}\overline{W}^{ia} + \prod_{i=1}^{N_w}\overline{W}^{ib}\right)\right]\left(\prod_{i=1}^{l-1}\overline{W}^{ib}\right)^T \tag{4.57}$$

Since $\overline{W}^{la}$ and $\overline{W}^{lb}$ start as diagonal matrices as our assumption goes, the dynamics of both pathways can be decoupled into independently evolving modes, with each mode being a multiplication of N_w scalars $m^1, \ldots, m^{N_w}$ for pathway a and $p^1, \ldots, p^{N_w}$ for pathway b. In addition, we assume that the scalars are initialized equally within each pathway, such that all the scalars in the same pathway remain equal during the dynamics (i.e., $m^i = m, p^i = p$). Then from Equation 4.56-4.57 we obtain (for each mode α)

$$\tau \frac{dm}{dt} = m^{N_w - 1}(s - (m^{N_w} + p^{N_w})) \tag{4.58}$$

$$\tau \frac{dp}{dt} = p^{N_w - 1}(s - (m^{N_w} + p^{N_w})) \tag{4.59}$$

Thus the knowledge in each pathway, $k_a = \prod_{i=1}^{N_w} m^i = m^{N_w}, k_b = \prod_{i=1}^{N_w} p^i = p^{N_w}$, evolves as follows:

$$\tau \frac{dk_a}{dt} = N_w k_a^{2-2/N_w}(s - (k_a + k_b)) \tag{4.60}$$

$$\tau \frac{dk_b}{dt} = N_w k_b^{2-2/N_w}(s - (k_a + k_b)) \tag{4.61}$$

The special case of $N_w = 2$ reduces to Equation 4.28. For $N_w \geq 3$, we have

$$\frac{k_a'}{k_b'} = \frac{k_a^{2-2/N_w}}{k_b^{2-2/N_w}} \tag{4.62}$$

Integrating the above equation, we get

$$k_a^{2/N_w - 1} = k_b^{2/N_w - 1} + C, \quad C = k_a^{2/N_w - 1}(0) - k_b^{2/N_w - 1}(0) \tag{4.63}$$

When $N_w \to \infty$, we have

$$\frac{1}{k_a} = \frac{1}{k_b} + C, \quad C = \frac{1}{k_a(0)} - \frac{1}{k_b(0)} \tag{4.64}$$

Similar to Figure 4.4 for the three layer case, the evolution of acquired knowledge for both pathways of deeper networks is shown in Figure 4.8. We see that – in contrast to the case for three layer networks – for deeper networks, the knowledge ratio between the two pathways is not constant across learning. Rather, this ratio either increases or decreases, being magnified for small initialization below the line $k_a + k_b = s$ but reduced for large initialization above this line. These trends become more and more pronounced for deeper networks (greater N_w).

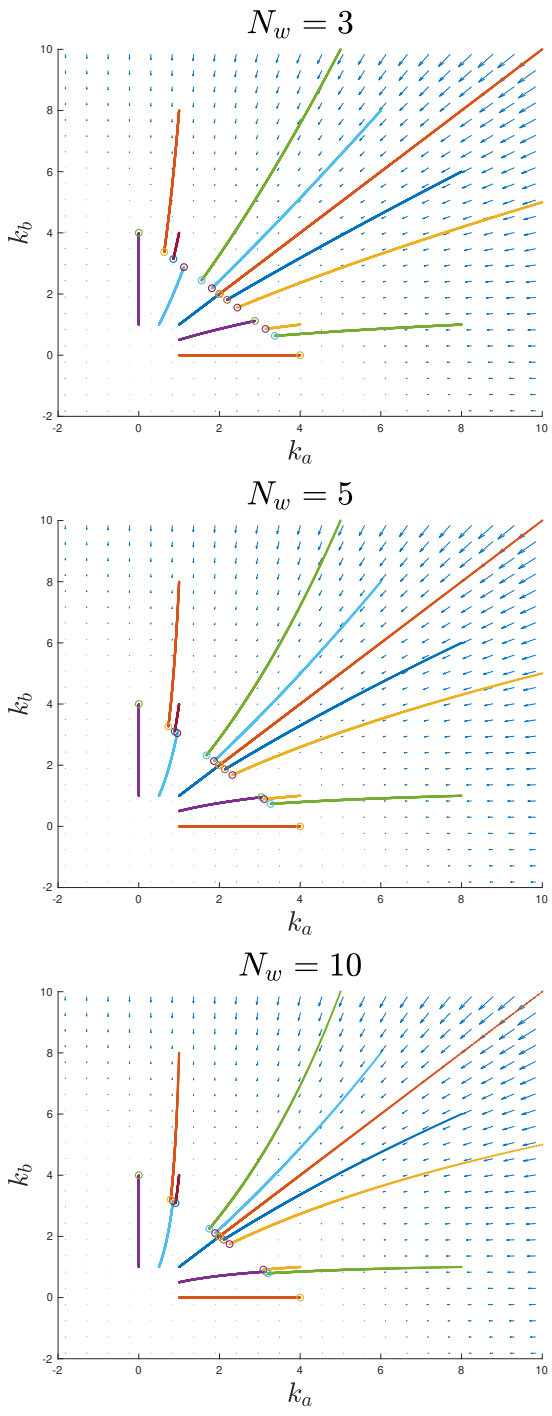

Figure 4.8: **Dynamics of acquired knowledge for deeper networks, for the special case initialization.** The evolution of the knowledge acquired by the two pathways k_a, k_b governed by Equation 4.60-4.61. The trajectories converge to $k_a + k_b = s$ and the ratio k_a/k_b changes more drastically through time for deeper networks.

4.4.2 Small random initialization

We now give a detailed analysis of multipathway learning for small random initialization, for "deeper" networks with four layers. We will conduct numerical simulations of networks with more layers further below, illustrating allied trends.

As with the three layer case, we will show that the final knowledge on each mode learned by two pathways can be predicted by a two stage process, namely

1. a beginning period where the dynamics evolves to the special case initialization;

2. a second stage where the knowledge evolves according to the special case initialization.

From Equation 4.54-4.55, the full dynamics for the weight matrices for the four layer network is given by (we only write the equations for pathway a, the analysis for pathway b is identical to pathway a)

$$\tau \frac{d}{dt} W^{1a} = W^{2aT} W^{3aT} (\Sigma^{yx} - \Omega) \tag{4.65}$$

$$\tau \frac{d}{dt} W^{2a} = W^{3aT} (\Sigma^{yx} - \Omega) W^{1aT} \tag{4.66}$$

$$\tau \frac{d}{dt} W^{3a} = (\Sigma^{yx} - \Omega) W^{1aT} W^{2aT} \tag{4.67}$$

where $\Omega = W^{3a} W^{2a} W^{1a} + W^{3b} W^{2b} W^{1b}$. Let

$$W^{1a} = \overline{W}^{1a} V^T, \;\; W^{2a} = \overline{W}^{2a}, \;\; W^{3a} = U \overline{W}^{3a} \tag{4.68}$$

$$W^{1b} = \overline{W}^{1b} V^T, \;\; W^{2b} = \overline{W}^{2b}, \;\; W^{3b} = U \overline{W}^{3b} \tag{4.69}$$

The dynamics of the weight matrices after projecting to the singular vectors of the input-output correlation matrix is then given by:

$$\tau \frac{d}{dt} \overline{W}^{1a} = \overline{W}^{2aT} \overline{W}^{3aT} (S - \overline{\Omega}) \tag{4.70}$$

$$\tau \frac{d}{dt} \overline{W}^{2a} = \overline{W}^{3aT} (S - \overline{\Omega}) \overline{W}^{1aT} \tag{4.71}$$

$$\tau \frac{d}{dt} \overline{W}^{3a} = (S - \overline{\Omega}) \overline{W}^{1aT} \overline{W}^{2aT} \tag{4.72}$$

where $\overline{\Omega} = \overline{W}^{3a}\overline{W}^{2a}\overline{W}^{1a} + \overline{W}^{3b}\overline{W}^{2b}\overline{W}^{1b}$.

Stage 1: small random weights converge to the diagonal knowledge initialization case. With small random initialization, during the beginning period when $\overline{\Omega} << S$, the dynamics for the projected weight matrices can be simplified as follows

$$\tau \frac{d}{dt}\overline{W}^{1a} = \overline{W}^{2aT}\overline{W}^{3aT}S \tag{4.73}$$

$$\tau \frac{d}{dt}\overline{W}^{2a} = \overline{W}^{3aT}S\overline{W}^{1aT} \tag{4.74}$$

$$\tau \frac{d}{dt}\overline{W}^{3a} = S\overline{W}^{1aT}\overline{W}^{2aT} \tag{4.75}$$

In the following, we are going to show that during the beginning period, the expectation (across realizations of weight matrices) of the "knowledge matrix"

$$K^a \triangleq \overline{W}^{3a}\overline{W}^{2a}\overline{W}^{1a} \tag{4.76}$$

evolves as a diagonal matrix, and we will describe the dynamics of its diagonal elements in the following. We will show that the derived expectation matches well with the simulations of exact dynamics of individual realizations with randomly generated weight matrices (Figure 4.9).

With small random initialization where W^{3a}, W^{2a}, W^{1a} are i.i.d. generated with $\mathcal{N}(0, \epsilon^2)$, at $t = 0$, we have

$$E[\overline{W}^{3a}\overline{W}^{2a}\overline{W}^{1a}] = 0, \tag{4.77}$$

$$E[\overline{W}^{1aT}\overline{W}^{1a}] = \epsilon^2 N_{2a} I_{N_1}, \tag{4.78}$$

$$E[\overline{W}^{2aT}\overline{W}^{2a}] = E[\overline{W}^{2a}\overline{W}^{2aT}] = \epsilon^2 N_{2a} I_{N_{2a}}, \tag{4.79}$$

$$E[\overline{W}^{3a}\overline{W}^{3aT}] = \epsilon^2 N_{2a} I_{N_3} \tag{4.80}$$

With the simplified dynamics (Equation 4.73-4.75), the dynamics of the knowledge matrix K^a evolves as

$$\begin{aligned} \frac{d}{dt}\left(\overline{W}^{3a}\overline{W}^{2a}\overline{W}^{1a}\right) &= \frac{d\overline{W}^{3a}}{dt}\overline{W}^{2a}\overline{W}^{1a} + \overline{W}^{3a}\frac{d\overline{W}^{2a}}{dt}\overline{W}^{1a} + \overline{W}^{3a}\overline{W}^{2a}\frac{d\overline{W}^{1a}}{dt} \\ = \frac{1}{\tau}\left(S\overline{W}^{1aT}\overline{W}^{2aT}\overline{W}^{2a}\overline{W}^{1a} + \overline{W}^{3a}\overline{W}^{3aT}S\overline{W}^{1aT}\overline{W}^{1a} + \overline{W}^{3a}\overline{W}^{2a}\overline{W}^{2aT}\overline{W}^{3aT}S\right) \end{aligned} \tag{4.81}$$

Plugging in Equation 4.77-4.80 and applying the independence between the matrices, we get

$$\begin{aligned} E\left[S\overline{W}^{1aT}\overline{W}^{2aT}\overline{W}^{2a}\overline{W}^{1a}\right] &= E\left[S\overline{W}^{1aT}E\left[\overline{W}^{2aT}\overline{W}^{2a}\right]\overline{W}^{1a}\right] \\ &= \epsilon^2 N_{2a} E\left[S\overline{W}^{1aT}\overline{W}^{1a}\right] = \epsilon^4 N_{2a}^2 S \end{aligned} \tag{4.82}$$

and likewise

$$E\left[\overline{W}^{3a}\overline{W}^{3aT}S\overline{W}^{1aT}\overline{W}^{1a}\right] = E\left[\overline{W}^{3a}\overline{W}^{2a}\overline{W}^{2aT}\overline{W}^{3aT}S\right] = \epsilon^4 N_{2a}^2 S \tag{4.83}$$

thus we have (at $t = 0$)

$$E\left[\frac{d}{dt}\left(\overline{W}^{3a}\overline{W}^{2a}\overline{W}^{1a}\right)\right] = \frac{3}{\tau}\epsilon^4 N_{2a}^2 S \tag{4.84}$$

Therefore we can see that the expectation of the knowledge matrix K^a starts from zero (Equation 4.77) and will increase only at the diagonal elements in the very beginning (Equation 4.84).

To further quantify the speed of knowledge increase during the beginning period, we examine the dynamics of the terms in Equation 4.81. From Equation 4.73-4.75, we have

$$\begin{aligned} \tau\frac{d}{dt}\left(\overline{W}^{1aT}\overline{W}^{1a}\right) &= \overline{W}^{1aT}\overline{W}^{2aT}\overline{W}^{3aT}S + S^T\overline{W}^{3a}\overline{W}^{2a}\overline{W}^{1a} \\ E\left[\tau\frac{d}{dt}\left(\overline{W}^{1aT}\overline{W}^{1a}\right)\right] &= 2S^T K^a \end{aligned} \tag{4.85}$$

and

$$\begin{aligned} \tau\frac{d}{dt}\left(\overline{W}^{3a}\overline{W}^{3aT}\right) &= S\overline{W}^{1aT}\overline{W}^{2aT}\overline{W}^{3aT} + \overline{W}^{3a}\overline{W}^{2a}\overline{W}^{1a}S^T \\ E\left[\tau\frac{d}{dt}\left(\overline{W}^{3a}\overline{W}^{3aT}\right)\right] &= 2K^a S^T \end{aligned} \tag{4.86}$$

and

$$\begin{aligned} \tau\overline{W}^{1aT}\frac{d}{dt}\left(\overline{W}^{2aT}\overline{W}^{2a}\right)\overline{W}^{1a} &= \overline{W}^{1aT}\overline{W}^{2aT}\overline{W}^{3aT}S\overline{W}^{1aT}\overline{W}^{1a} \\ &\quad + \overline{W}^{1aT}\overline{W}^{1a}S^T\overline{W}^{3a}\overline{W}^{2a}\overline{W}^{1a} \\ E\left[\tau\overline{W}^{1aT}\frac{d}{dt}\left(\overline{W}^{2aT}\overline{W}^{2a}\right)\overline{W}^{1a}\right] &= 2\overline{W}^{1aT}\overline{W}^{1a}S^T K^a \end{aligned} \tag{4.87}$$

and

$$\tau \overline{W}^{3a} \frac{d}{dt} \left(\overline{W}^{2a} \overline{W}^{2aT} \right) \overline{W}^{3aT} = \overline{W}^{3a} \overline{W}^{2a} \overline{W}^{1a} S^T \overline{W}^{3a} \overline{W}^{3aT} + \overline{W}^{3a} \overline{W}^{3aT} S \overline{W}^{1aT} \overline{W}^{2aT} \overline{W}^{3aT}$$

$$E \left[\tau \overline{W}^{3a} \frac{d}{dt} \left(\overline{W}^{2a} \overline{W}^{2aT} \right) \overline{W}^{3aT} \right] = 2K^a S^T \overline{W}^{3a} \overline{W}^{3aT} \tag{4.88}$$

Thus $\overline{W}^{1aT} \overline{W}^{1a}$ (Equation 4.85), $\overline{W}^{3a} \overline{W}^{3aT}$ (Equation 4.86), and the projections of $\overline{W}^{2a} \overline{W}^{2aT}$ (Equation 4.87), $\overline{W}^{2aT} \overline{W}^{2a}$ (Equation 4.88) presented in Equation 4.81 all grow in diagonal manner with the same speed in expectation. We denote their diagonal elements as $w_\alpha = E\left[(\overline{W}^{1aT} \overline{W}^{1a})_{\alpha,\alpha} \right]$ and let k_α denote $E\left[K^a_{\alpha,\alpha} \right]$. Then from Equation 4.81 for each mode α, the diagonal element of the knowledge matrix evolves as

$$\tau \frac{d}{dt} k_\alpha = 3 s_\alpha w_\alpha^2 \tag{4.89}$$

and diagonal elements of matrices in Equation 4.85 evolve as

$$\tau \frac{d}{dt} w_\alpha = 2 s_\alpha k_\alpha \tag{4.90}$$

These equations, together with $k_\alpha(0) = 0$ and $w_\alpha(0) = \epsilon^2 N_{2a}$, approximate the evolution of k_α in the beginning period.

For small random initialization, the above dynamics will converge to the following special case dynamics during the beginning period. These are the same dynamics as diagonal knowledge initialization with the decay term ignored for the beginning period (see a simulation example in Figure 4.9 and a perturbation analysis argument in Section 4.6.2):

$$k_\alpha = w_\alpha^{\frac{3}{2}} \tag{4.91}$$

$$\tau \frac{d}{dt} k_\alpha = 3 s_\alpha k_\alpha^{\frac{4}{3}} \tag{4.92}$$

where the evolution of knowledge for the two pathways can be written as (in the following we will omit the label α for specific mode and use a and b to denote the two pathways as

before)

$$\tau \frac{d}{dt} k_a = 3s k_a^{\frac{4}{3}} \tag{4.93}$$

$$\tau \frac{d}{dt} k_b = 3s k_b^{\frac{4}{3}} \tag{4.94}$$

with $\frac{dk_a}{dk_b} = (\frac{k_a}{k_b})^{\frac{4}{3}}$.

Stage 2: knowledge evolution according to diagonal knowledge initialization case. After the evolution of the knowledge has converged to the above dynamics, it will then follow the special case initialization discussed in the previous section (Equation 4.60-4.61 with $N_w = 3$), given by

$$\tau \frac{dk_a}{dt} = 3k_a^{\frac{4}{3}}(s - (k_a + k_b)) \tag{4.95}$$

$$\tau \frac{dk_b}{dt} = 3k_b^{\frac{4}{3}}(s - (k_a + k_b)) \tag{4.96}$$

again with $\frac{dk_a}{dk_b} = (\frac{k_a}{k_b})^{\frac{4}{3}}$.

In the $k_a - k_b$ phase plane, the solution trajectories to Equation 4.93-4.94 and Equation 4.95-4.96 follow the same paths, except that the dynamics for Equation 4.95-4.96 will finally converge to the line $k_a + k_b = s$. Thus if we run the first stage dynamics in the beginning period long enough so that they converge to 4.93-4.94, and then use the second stage dynamics, we can obtain the final learned knowledge k_a, k_b.

In summary, the two stage prediction of final knowledge k_a and k_b in one mode is calculated by the following procedure:

1. Follow first stage dynamics (Equation 4.89-4.90) with $k_\alpha(0) = 0$ and $w_\alpha(0) = \epsilon^2 N_{2a}(\epsilon^2 N_{2b}$ for pathway b) until convergence to Equation 4.91-4.92.

2. Follow the second stage dynamics (Equation 4.95-4.96) until converging.

An example of the combined two stage approximation is shown in Figure 4.10. It shows that the combined two-stage prediction is able to give good approximations to the final

learned knowledge in each pathway. This is true even if we take an aggressively long period for the first stage dynamics, since it converges to the special case dynamics.

These simulations, both of the full equations and based on the two stage approximation, clearly show that small differences in the initial knowledge between the two pathways will result in large differences in the final learned knowledge. An analytical argument for this consistent increase of the knowledge ratio during both stages is given in Section 4.6.3. Figure 4.11 presents further simulations with two stage prediction of the final learned knowledge for varied pathway sizes.

Moreover, we expect from this observation, together with the analysis for networks of arbitrary depth in Section 4.4.1, that for deeper networks the final knowledge learned by the two pathways will diverge even more dramatically than for the four layer case described above. We demonstrate this phenomenon through the simulations in Figure 4.12 and Figure 4.13. Note that the final learned knowledge matrices for the examples are not exactly diagonal; thus, the first stage did not converge fully to the special case.

In this section, our analysis and simulations tell us that for deeper networks, the small random differences in initialization can be amplified to lead to large differences in learned knowledge of different input-output modes among network pathways.

To summarize, we begin with three layer networks, and find that, with the diagonal knowledge initialization scheme, the final knowledge learned by each pathway is exactly the same as that at the initialization of network weights. For small random initialization scheme, the final knowledge learned by each pathway is proportional to its hidden layer sizes. Interestingly, deeper networks with parallel pathways produce a different effect. Here, small differences of knowledge at the initialization of the pathways will be magnified, leading to large deviations in the final knowledge among different pathways.

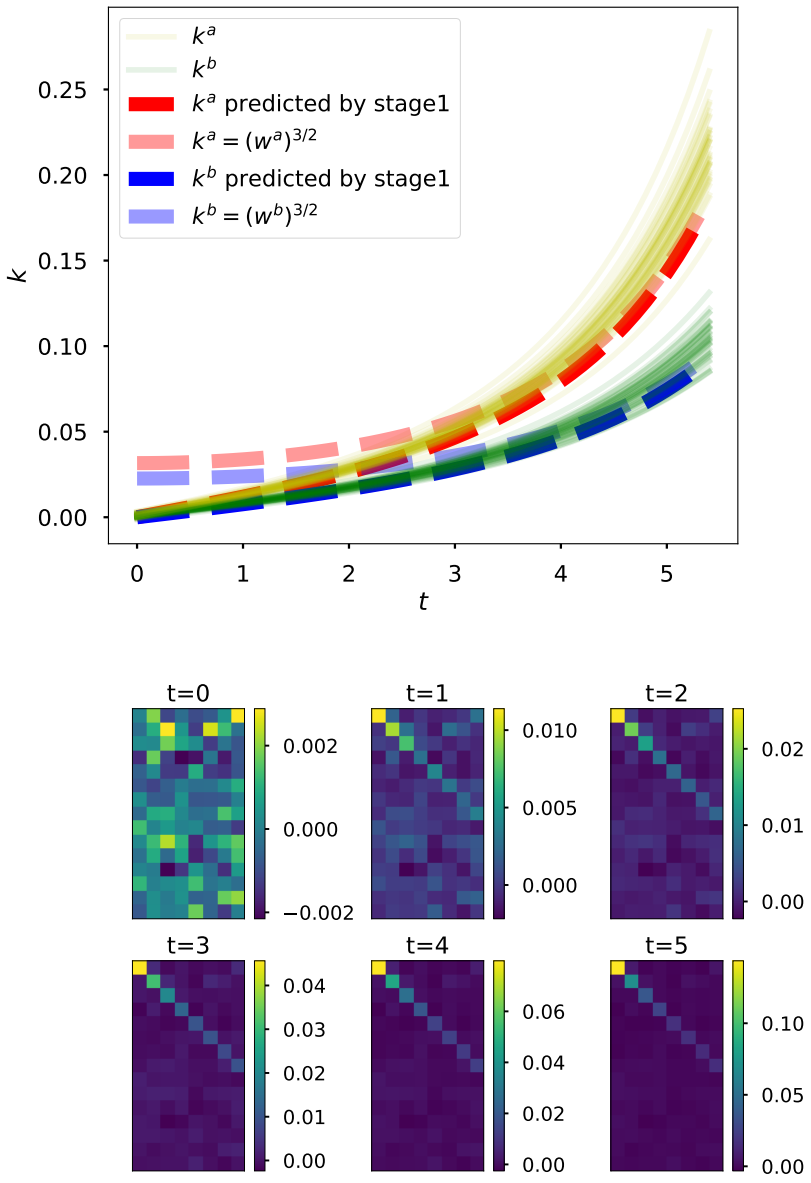

Figure 4.9: **Knowledge evolution during the beginning period.** (Top) The yellow and green lines are simulations of exact dynamics from multiple randomly generated weight matrices with fixed pathway sizes ($N_{2a} = 1000, N_{2b} = 800$) and $\epsilon = 0.01$. The dark dashed red and blue lines are predictions of k^a and k^b for the first mode from stage 1 (Equation 4.89-4.90). They converge to the special case dynamics (Equation 4.91-4.92) trajectories (light dashed red and blue) during the beginning period. (Bottom) An example realization of knowledge matrix evolution for pathway a at different time steps. The knowledge matrix starts as a random matrix and evolves into a diagonal matrix during the beginning period.

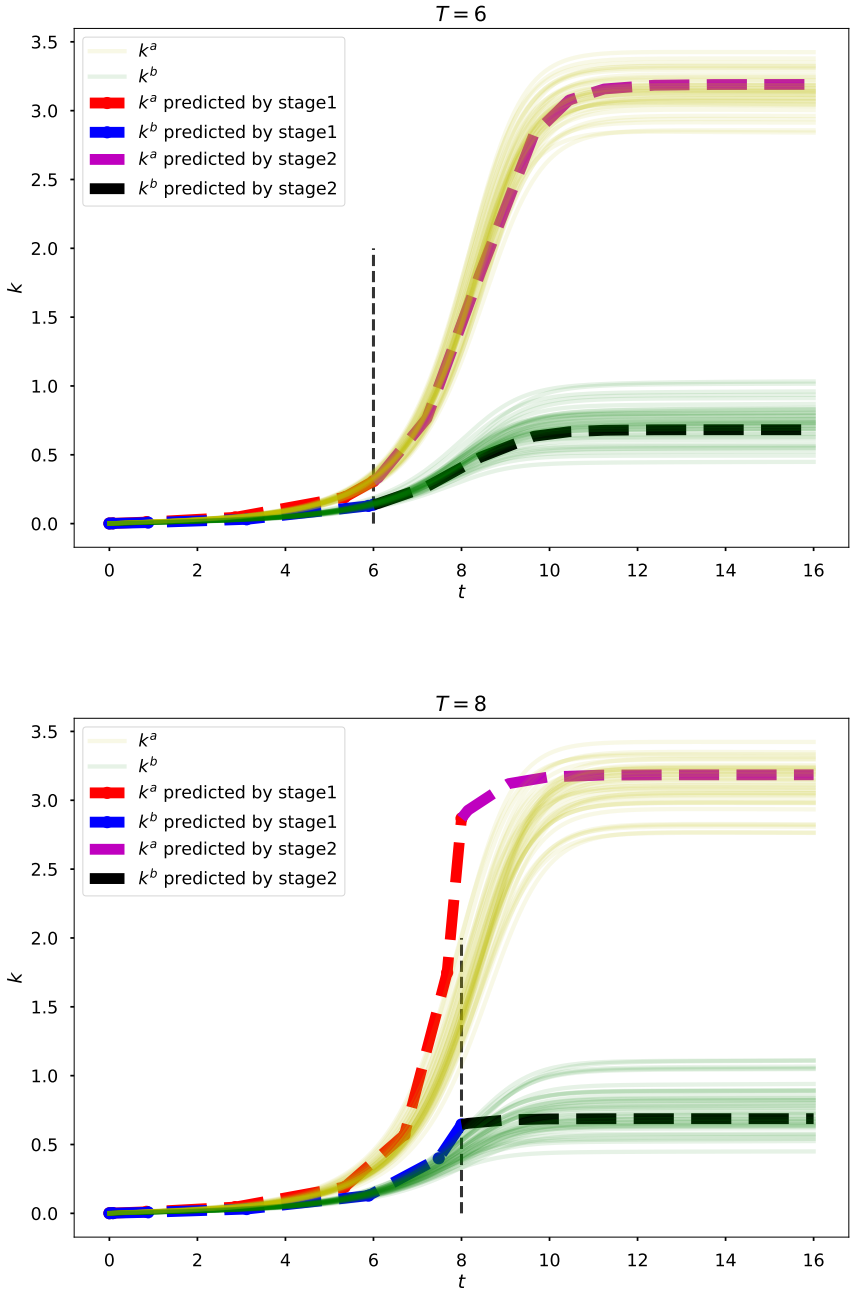

Figure 4.10: **Two-stage prediction of knowledge evolution.** The yellow and green lines are simulations of exact dynamics from multiple randomly generated weight matrices with fixed pathway sizes ($N_{2a} = 1000, N_{2b} = 800$) and $\epsilon = 0.01$. The dashed red and blue lines are predictions of k^a and k^b for the first mode from stage 1 (Equation 4.89-4.90). The dashed magenta and black lines are predictions of k^a and k^b for the first mode from stage 2 (Equation 4.60-4.61). Top and bottom are predictions from two different length of time for running the first stage dynamics.

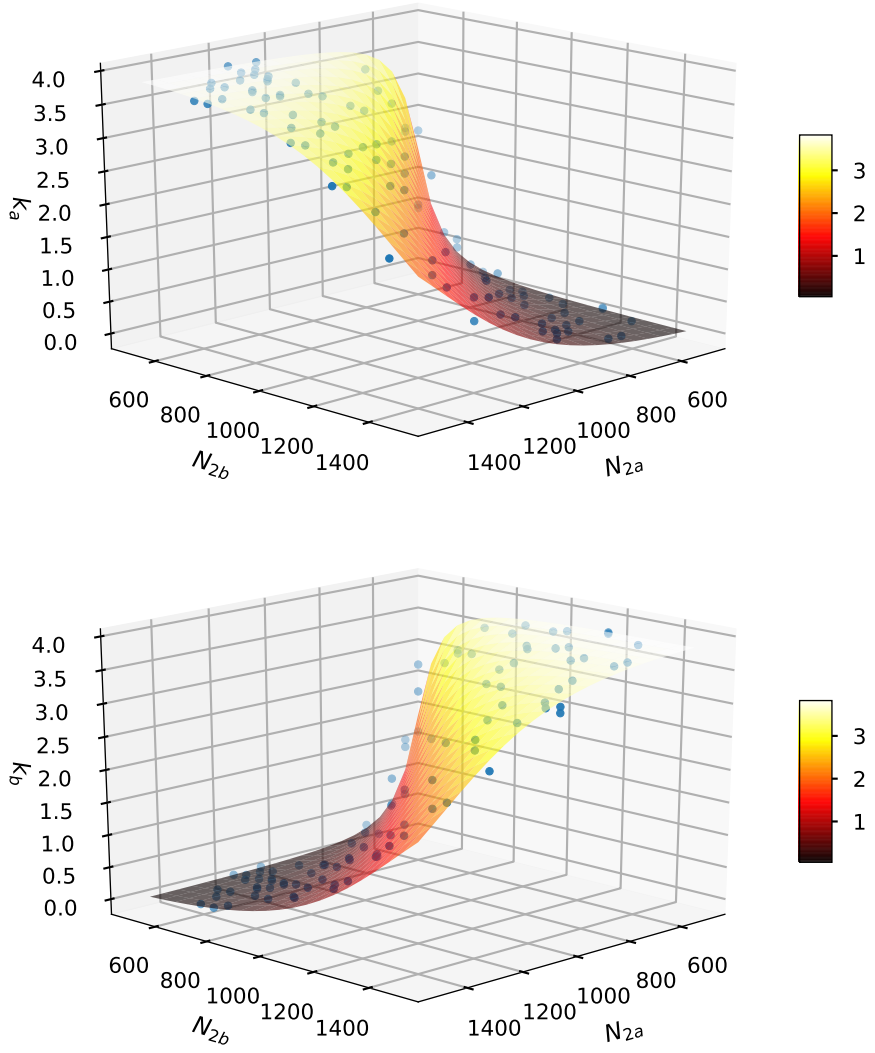

Figure 4.11: **Two-stage prediction of final learned knowledge along with simulation of exact dynamics for varied pathway sizes.** The surface plot is from the two stage prediction of the final knowledge learned by pathway a(top) and pathway b(bottom) with fixed $T = 6$ and $\epsilon = 0.01$. The blue dots are from simulations of exact dynamics from multiple randomly generated weight matrices with varied pathway sizes.

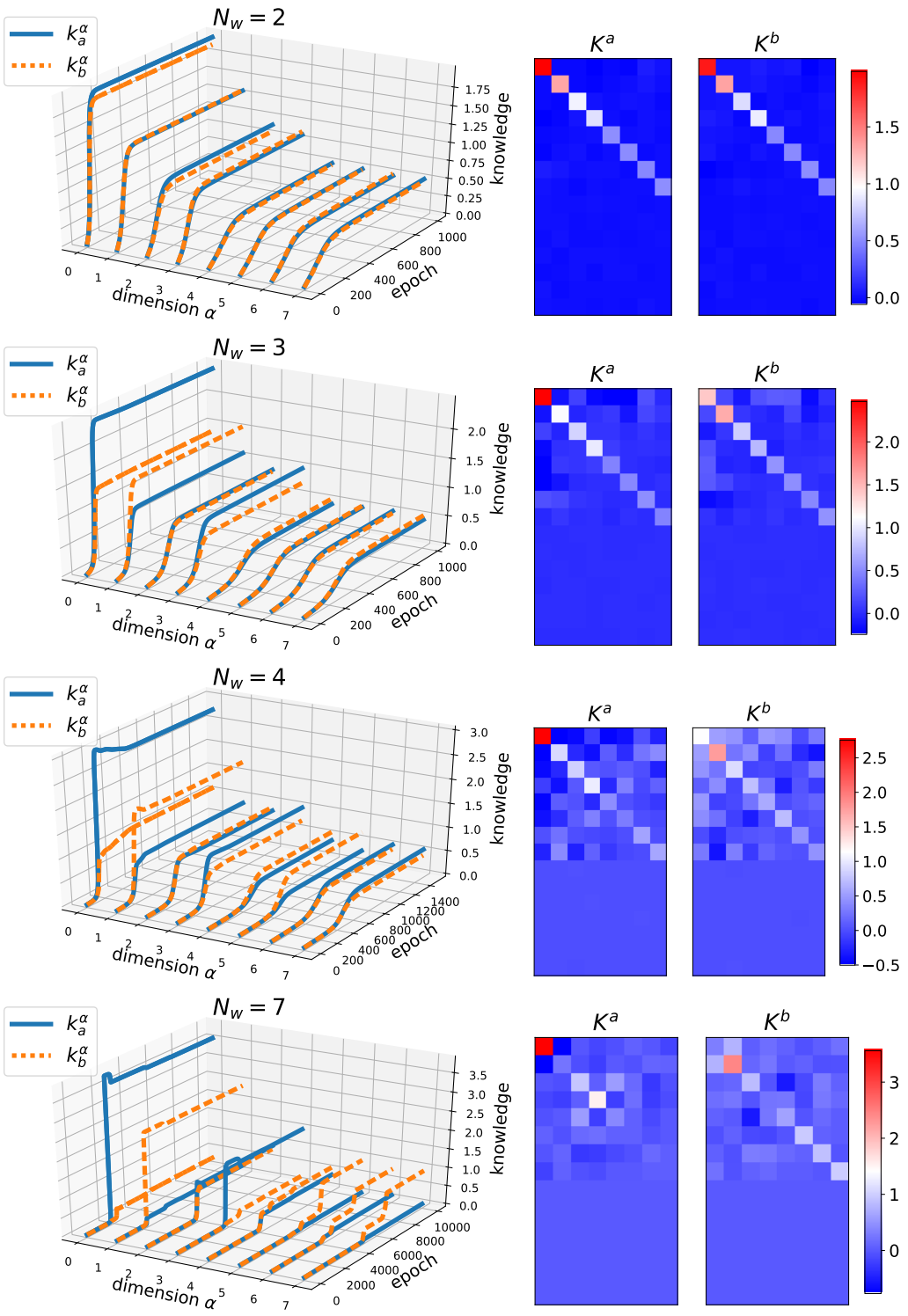

Figure 4.12: **Knowledge distribution in networks with increasing depth.** Knowledge evolution (left) and final learned knowledge matrices (right) for single realizations of networks with the same size ($N_a = N_b = 1000$) pathways for various depth, trained from small random initialization ($\epsilon = 0.01$). We can see that although the two pathways have the same relative size, the final knowledge learned by each pathways differs more dramatically for deeper networks.

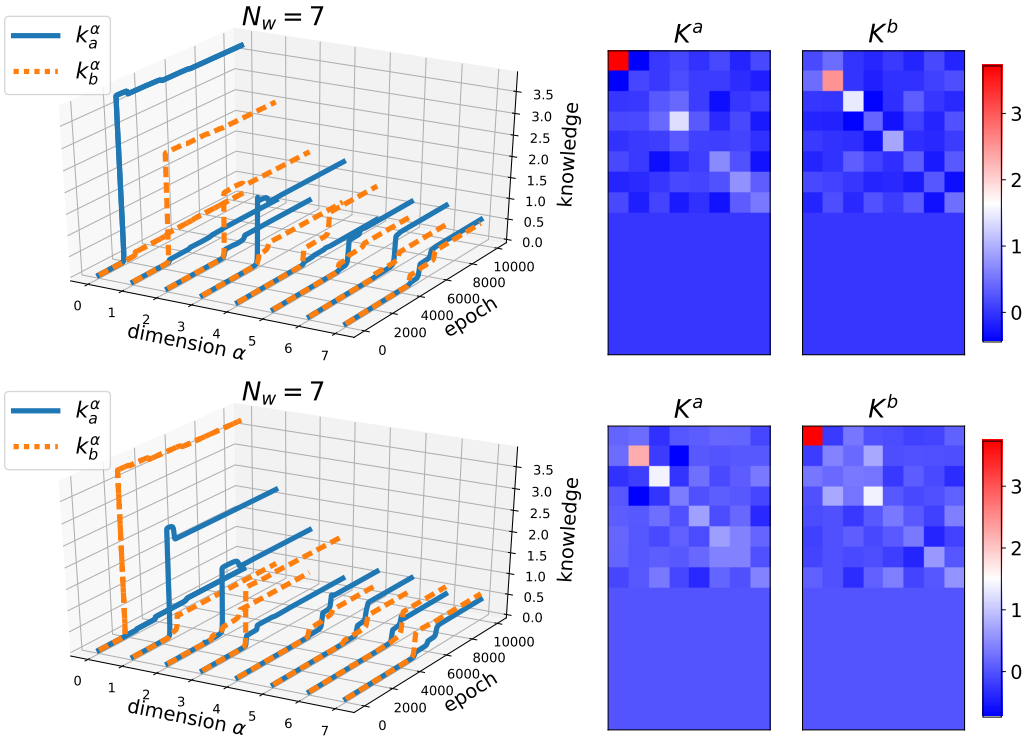

Figure 4.13: **Knowledge distribution in networks with same depth and different initialization.** Knowledge evolution (left) and final learned knowledge matrices (right) for single realizations of networks with the same size ($N_a = N_b = 1000$) pathways for the same depth trained from small random initialization ($\epsilon = 0.01$) with different random seeds. We can see that with different random seeds, the two pathways can learn knowledge corresponding to different modes.

4.5 Conclusion

In this work, we study how task-related knowledge is distributed in networks with parallel pathways during learning, in the setting of deep linear networks. We specifically examine two special initialization schemes, namely diagonal knowledge initialization and small random initialization. In both of these cases, the final knowledge learned by each pathway is strongly and predictably affected by their initialization. In particular, for small weight initialization, small differences in knowledge between the pathways at initialization can be magnified by the increasing depth of the networks, leading to more dramatically different knowledge distributions among pathways. Our model serves as a possible explanation of how diverse representations of parallel pathways emerge among biological and artificial models through learning.

Acknowledgments

We thank Tianqi Chen for helpful discussions. We thank Adrienne Fairhall for helpful comments and suggestions on the draft.

4.6 Appendix

4.6.1 General solution for diagonal knowledge initialization

Without assuming $m = n, p = q$, denote

$$m^2 - n^2 = c_0 \tag{4.97}$$

$$p^2 - q^2 = c_1 \tag{4.98}$$

W.O.L.G., assume $c_0, c_1 > 0, m^2 > n^2, p^2 > q^2$, let

$$m = \sqrt{c_0}\cosh\frac{\theta}{2}, \quad n = \sqrt{c_0}\sinh\frac{\theta}{2} \tag{4.99}$$

$$p = \sqrt{c_1}\cosh\frac{\phi}{2}, \quad q = \sqrt{c_1}\sinh\frac{\phi}{2} \tag{4.100}$$

We have

$$\tau\frac{d\theta}{dt} = 2(s - c_0\sinh\theta - c_1\sinh\phi) \tag{4.101}$$

$$\tau\frac{d\phi}{dt} = 2(s - c_0\sinh\theta - c_1\sinh\phi) \tag{4.102}$$

Thus

$$\frac{d}{dt}(\theta - \phi) = 0 \tag{4.103}$$

Setting $\theta - \phi = c_2$, the nonlinear dynamics of Equation 4.101 can be solved. In this case, the ratio of the final learned knowledge between the two pathways is not necessarily the same as at their initialization.

4.6.2 Perturbation analysis showing that the beginning dynamics of three layer networks with small random initialization converges to the special case

Here we provide an argument for the convergence of Equation 4.89-4.90 to Equation 4.91-4.92 by showing

1. Equation 4.91-4.92 is a solution to Equation 4.89-4.90.

2. The solution of Equation 4.91-4.92 is stable under small perturbations.

The first point is straightforward, by plugging Equation 4.91-4.92 into Equation 4.89-4.90.

We now show that the solution of Equation 4.91-4.92 is stable under small perturbations. Assume a small perturbation $x << 1$ from Equation 4.91

$$w_\alpha = k_\alpha^{2/3} + x \tag{4.104}$$

plugging in Equation 4.91-4.92, we have

$$\frac{d}{dt}k_\alpha = \frac{1}{\tau}3s_\alpha(k_\alpha^{2/3} + x)^2 \tag{4.105}$$

$$\begin{aligned}\frac{d}{dt}w_\alpha &= \frac{2}{3}k_\alpha^{-\frac{1}{3}}\frac{d}{dt}k_\alpha + \frac{d}{dt}x = \frac{2}{3}k_\alpha^{-\frac{1}{3}}\frac{1}{\tau}3s_\alpha(k_\alpha^{2/3} + x)^2 + \frac{d}{dt}x \\ &= \frac{2s_\alpha}{\tau}k_\alpha^{-\frac{1}{3}}(k_\alpha^{2/3} + x)^2 + \frac{d}{dt}x \\ &\approx \frac{2s_\alpha}{\tau}k_\alpha^{-\frac{1}{3}}(k_\alpha^{4/3} + 2k_\alpha^{2/3}x) + \frac{d}{dt}x \;\; (x << 0) \\ &= \frac{2s_\alpha}{\tau}(k_\alpha + 2k_\alpha^{1/3}x) + \frac{d}{dt}x\end{aligned} \tag{4.106}$$

We also have

$$\frac{d}{dt}w_\alpha = \frac{2s_\alpha}{\tau}k_\alpha \tag{4.107}$$

Combining Equation 4.106 and Equation 4.107, we get

$$\begin{aligned}\frac{2s_\alpha}{\tau}(k_\alpha + 2k_\alpha^{1/3}x) + \frac{d}{dt}x &= \frac{2s_\alpha}{\tau}k_\alpha \\ \frac{d}{dt}x &= -\frac{4s_\alpha}{\tau}k_\alpha^{1/3}x\end{aligned} \tag{4.108}$$

Thus given that $s_\alpha > 0, k_\alpha > 0$, x will decay to zero. Therefore the solution of Equation 4.91-4.92 is stable under small perturbations.

4.6.3 Small differences in knowledge between pathways at initialization lead to larger differences at the end of learning

In this section, we provide an analysis of how small differences in the initialization between two pathways will lead to large differences in their final learned knowledge, for a deeper linear network ($N_w = 3$). Specifically, we will show that

1. The knowledge difference between the two pathways will increase in stage 1.

2. The knowledge learned in stage 1 can be kept small for small initialization.

3. The knowledge difference between the two pathways will increase in stage 2.

We provide analytical or numerical arguments for the above statements as follows.

1. Recall that the stage 1 dynamics is as follows:

$$\tau \frac{d}{dt} k_\alpha = 3 s_\alpha w_\alpha^2 \tag{4.109}$$

$$\tau \frac{d}{dt} w_\alpha = 2 s_\alpha k_\alpha \tag{4.110}$$

with $k_\alpha^a(0) = k_\alpha^b(0) = 0, w_\alpha^a(0) = \epsilon^2 N_{2a}, w_\alpha^b(0) = \epsilon^2 N_{2b}$. Note that these starting values of k, w are in expectation, so there will be small difference between the two pathways in the realizations even when $N_{2a} = N_{2b}$. Let

$$k_\alpha^b(t) = k_\alpha^a(t) + \delta_k(t) \tag{4.111}$$

$$w_\alpha^b(t) = w_\alpha^a(t) + \delta_w(t) \tag{4.112}$$

We have

$$\tau \delta_k' = 3 s_\alpha ((w_\alpha^a + \delta_w)^2 - (w_\alpha^a)^2) = 6 s_\alpha w_\alpha^a \delta_w + O(\delta_w^2) \tag{4.113}$$

$$\tau \delta_w' = 2 s_\alpha (k_\alpha^b - k_\alpha^a) = 2 s_\alpha \delta_k \tag{4.114}$$

W.O.L.G., we assume $\delta_w(0) \geq 0$, and we also have $\delta_k(0) \approx 0$. Since s_α, w_α^a are both positive, δ_k, δ_w will grow together according to the above dynamics. Hence the knowledge difference between the two pathways will increase in stage 1.

2. Here we show numerically that after stage 1 converge to the following dynamics

$$k_\alpha = w_\alpha^{\frac{3}{2}} \tag{4.115}$$

$$\tau \frac{d}{dt} k_\alpha = 3 s_\alpha k_\alpha^{\frac{4}{3}} \tag{4.116}$$

the knowledge learned by each pathway can be kept small, given small enough initialization. As shown in Figure 4.14, smaller initialization leads to smaller k after the dynamics of stage 1. Thus as long as the initialization is small enough, we will have small k values for starting the stage 2 dynamics.

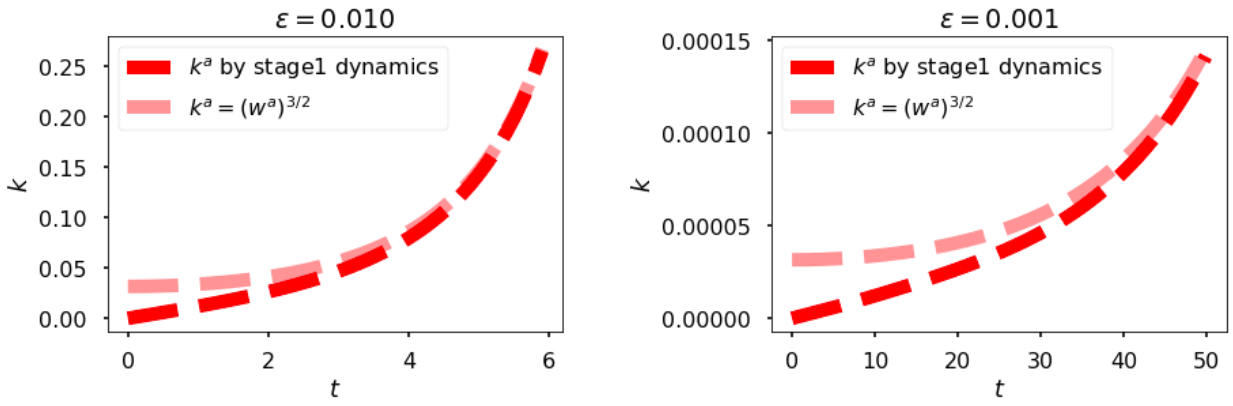

Figure 4.14: **Smaller initialization scale leads to smaller k after stage 1.** The solid red dashed lines are numerical solution of stage 1 dynamics (Equation 4.109-4.110). The faded red dashed lines are the solution that stage 1 dynamics converging to (Equation 4.115). We see that at the end of the convergence of stage 1 dynamics, the values of k can be kept small for small initialization.

3. Recall that the evolution of k during the second stage is as follows (in the following we omit the mode α for simplicity)

$$\tau \frac{dk_a}{dt} = 3k_a^{\frac{4}{3}}(s - (k_a + k_b)) \tag{4.117}$$

$$\tau \frac{dk_b}{dt} = 3k_b^{\frac{4}{3}}(s - (k_a + k_b)) \tag{4.118}$$

giving $\frac{dk_a}{dk_b} = (\frac{k_a}{k_b})^{\frac{4}{3}}$. From analysis above, we know that the initial values for the second stage dynamics are small, i.e., k_a, k_b starts at small values and increases during this stage until their sum equals to s. W.O.L.G, we assume $k_a > k_b$ at the start of the second stage

dynamics. Then we have

$$\frac{d}{dt}\left(\frac{k_a}{k_b}\right) = \frac{k_a' k_b - k_a k_b'}{k_b^2} = \frac{k_b'}{k_b}\left(\frac{k_a'}{k_b'} - \frac{k_a}{k_b}\right) = \frac{k_b'}{k_b}\left(\left(\frac{k_a}{k_b}\right)^{\frac{4}{3}} - \frac{k_a}{k_b}\right) > 0 \tag{4.119}$$

Therefore the knowledge difference between the two pathways will also increase in stage 2.

Chapter 5

CONCLUSION AND FUTURE DIRECTIONS

With the prevailing success of deep neural networks in tasks of all kinds, and a host of rapidly emerging accessible large scale biological data sets, establishing links between the two is more desirable than ever – and holds promise for benefiting both sides. In this thesis, we try to extend these links by three interconnected projects.

In Chapter 2, we take the first step, of comparing ANN models with large scale functional data sets from mouse visual cortex. To do that, we first need to assess the robustness of such a comparison under limited experimental conditions, such as a limited set of stimuli and number of neurons observed. Thanks to readily available ANN models, we can mimic the limited experimental conditions and assess the robustness of comparison metrics by comparing ANNs to themselves. Our empirical results show that metrics of pseudo-depth and similarity scores are indeed robust to choices of stimuli on the order of hundreds, and subsampling of neurons on the order of thousands – both of which characterize modern experimental datasets.

After we established the validity of the comparison metrics, we then used this methodology to investigate visual representations in the mouse visual cortex. By comparing the Allen Brain Observatory data with various ANN architectures, we show that mouse visual cortical areas are relatively high order representations. They have a broad, more parallel organization rather than a sequential hierarchy, with the primary area VISp being lower order relative to the other areas. This is consistent with the relatively flat hierarchy observed in [Harris et al., 2018].

This comparison paradigm is rather general and can be used for analyzing large scale data sets for other species and systems. The robust similarity metrics can also offer guidance in

searching for better models [Conwell et al., 2021] for these data sets. However, since the similarity metrics only provide a scalar indicating how similar two representations are, they do not reveal why two representations are similar (or not similar) and how one can improve their similarity. Without knowing the why, searching for better models can only rely on blind trial and error approaches. A possible future direction is to take a more detailed examination of the internal representations of the ANNs, using representation characteristics such as quantitative measures of the internal geometries [Chung and Abbott, 2021, Williams et al., 2021], and establish relations between these representation characteristics and the similarity measures studied above.

Although Chapter 2 has shown that ANNs can be compared to the mouse visual cortex data and can help understand underlying functional properties, they are still far from realistic models of the biological brain. In Chapter 3, we present the first (to our knowledge) ANN model of the mouse visual cortex (MouseNet) whose architecture is constrained by large scale mesoscopic anatomical data. With the MouseNet, we can demonstrate the computational power that corresponds to a given biological resource and architecture, make comparisons corresponding homologous groups of neurons in models and the mouse brain, model specific lesions of brain areas, and analyze functional differences between brain areas and pathways.

Combining MouseNet with the comparison methodologies introduced in Chapter 2, we further investigated the effect of image classification task-training on the functional representation of the model – in particular, whether training on this task drives the representations in the model to be closer to those recorded from the real mouse brain. Using recordings from the large-scale Allen Brain Observatory survey [de Vries et al., 2020], we find that training on an image classification task does drive MouseNet representations to better resemble those of the real data. However, this increase of functional similarity is not necessarily strictly monotonic with task performance. In our experiments we see the similarity score with the Brain Observatory responses saturating or even reaching a maximum well before we achieve maximum accuracy on task performance.

Within the task-training paradigm, these results suggest that the specific image classifi-

cation task we used, and perhaps image classification overall, is not the appropriate task for describing what the mouse visual cortex has learned and developed to compute. Nonetheless, MouseNet is an important reference to studies in more established species, which rely on comparisons of the ventral stream with architectures designed for object recognition. Although we know rodents are capable of performing tasks that require visual object discrimination, mouse ethology suggests that alternate computations are more important for the mouse visual system, such as motion tracking, predation, and predator avoidance. A promising future direction is to use task-training of the MouseNet model, together with the metrics tested here, to develop more realistic tasks and stimuli that may lead to more closely matched representations. Recent research [Bakhtiari et al., 2021, Nayebi et al., 2021] indicates that self-supervised training objectives may be better task choices.

The MouseNet architecture itself also has room for further developments, such as adding more brain areas, adding recurrence, using different inputs and readouts for different pathways, incorporating new anatomical data when available. Last but not least, the MouseNet can also benefit from developments of better learning algorithms, since the standard stochastic gradient descent used in training ANNs performs significantly slower when training the MouseNet compared to typical ANNs.

A key feature in our constructed MouseNet architecture is its parallel pathways. In Chapter 3 we have shown that the multiple pathways in the MouseNet possess diversified representations after training on the image classification task. One would wonder how these different pathways learn different aspects of a task and to what extent. In Chapter 4, we utilize the mathematical framework of linear ANNs' learning dynamics to study the simplest possible model with parallel pathways. We examine how the learning process distributes task-related knowledge onto different pathways and quantify the resulting amount in each pathway analytically.

To our surprise, we find that under two special initialization schemes, the final knowledge learned by each pathway is strongly affected by their initialization. In particular, for a deep linear network, a small difference between the pathways at their initialization will be

magnified by the increasing depth of the networks, leading to more dramatically different knowledge distribution among the pathways after learning. Our mathematical analysis of this simple model setting suggests that small differences in initialization between different pathways, combined with the depth of the network, can be the root cause of the diversified learned representations among different pathways.

This theoretical study can be extended in multiple future directions. First, one may study more general initialization schemes, starting from weak interactions between the different input-output dimensions, and quantify their effects on the final learned knowledge. Second, it would be interesting to define a reasonable quantification of the learned "knowledge" when the final learned knowledge matrix is not diagonal. Third, one can extend the analysis to nonlinear networks and more complicated tasks.

In summary, this thesis has extended the links between the ANNs and the real brain, showing strong evidence that studying them in parallel can cultivate fruitful findings and inspirations. We hope this thesis can contribute to the remarriage of the development of the artificial neural networks and the understanding of the biological brain.

BIBLIOGRAPHY

[Abbott et al., 2020] Abbott, L. F., Bock, D. D., Callaway, E. M., Denk, W., Dulac, C., Fairhall, A. L., Fiete, I., Harris, K. M., Helmstaedter, M., Jain, V., Kasthuri, N., LeCun, Y., Lichtman, J. W., Littlewood, P. B., Luo, L., Maunsell, J. H., Reid, R. C., Rosen, B. R., Rubin, G. M., Sejnowski, T. J., Seung, H. S., Svoboda, K., Tank, D. W., Tsao, D., and Van Essen, D. C. (2020). The mind of a mouse. *Cell*, 182(6):1372–1376.

[Amorim Da Costa and Martin, 2010] Amorim Da Costa, N. M. and Martin, K. (2010). Whose cortical column would that be? *Frontiers in Neuroanatomy*, 4:16.

[Andermann et al., 2011] Andermann, M., Kerlin, A., Roumis, D., Glickfeld, L., and Reid, R. (2011). Functional specialization of mouse higher visual cortical areas. *Neuron*, 72(6):1025 – 1039.

[Bakhtiari et al., 2021] Bakhtiari, S., Mineault, P., Lillicrap, T., Pack, C. C., and Richards, B. A. (2021). The functional specialization of visual cortex emerges from training parallel pathways with self-supervised predictive learning. *bioRxiv*.

[Bakker et al., 2012] Bakker, R., Wachtler, T., and Diesmann, M. (2012). Cocomac 2.0 and the future of tract-tracing databases. *Frontiers in Neuroinformatics*, 6:30.

[Barrett et al., 2019] Barrett, D. G., Morcos, A. S., and Macke, J. H. (2019). Analyzing biological and artificial neural networks: challenges with opportunities for synergy? *Current Opinion in Neurobiology*, 55:55–64. Machine Learning, Big Data, and Neuroscience.

[Billeh et al., 2020] Billeh, Y. N., Cai, B., Gratiy, S. L., Dai, K., Iyer, R., Gouwens, N. W., Abbasi-Asl, R., Jia, X., Siegle, J. H., Olsen, S. R., Koch, C., Mihalas, S., and Arkhipov, A. (2020). Systematic integration of structural and functional data into multi-scale models of mouse primary visual cortex. *Neuron*, 106(3):388 – 403.e18.

[Borg and Groenen, 2005] Borg, I. and Groenen, P. (2005). *Modern Multidimensional Scaling: Theory and Applications (Springer Series in Statistics)*.

[Cadena et al., 2019] Cadena, S. A., Sinz, F. H., Muhammad, T., Froudarakis, E., Cobos, E., Walker, E. Y., Reimer, J., Bethge, M., Tolias, A., and Ecker, A. S. (2019). How well do deep neural networks trained on object recognition characterize the mouse visual system? NeurIPS Workshop Neuro-AI.

[Cadieu et al., 2014] Cadieu, C. F., Hong, H., Yamins, D. L. K., Pinto, N., Ardila, D., Solomon, E. A., Majaj, N. J., and DiCarlo, J. J. (2014). Deep neural networks rival the representation of primate it cortex for core visual object recognition. *PLOS Computational Biology*, 10(12):1–18.

[Chung and Abbott, 2021] Chung, S. and Abbott, L. (2021). Neural population geometry: An approach for understanding biological and artificial neural networks. *Current Opinion in Neurobiology*, 70:137–144. Computational Neuroscience.

[Cichy et al., 2016] Cichy, R. M., Khosla, A., Pantazis, D., Torralba, A., and Oliva, A. (2016). Comparison of deep neural networks to spatio-temporal cortical dynamics of human visual object recognition reveals hierarchical correspondence. *Scientific Reports*, 6:27755 EP.

[Conwell et al., 2021] Conwell, C., Mayo, D., Buice, M. A., Katz, B., Alvarez, G. A., and Barbu, A. (2021). Neural regression, representational similarity, model zoology & neural taskonomy at scale in rodent visual cortex. *bioRxiv*.

[de Vries et al., 2020] de Vries, S. E. J., Lecoq, J. A., Buice, M. A., Groblewski, P. A., Ocker, G. K., Oliver, M., Feng, D., Cain, N., Ledochowitsch, P., Millman, D., Roll, K., Garrett, M., Keenan, T., Kuan, L., Mihalas, S., Olsen, S., Thompson, C., Wakeman, W., Waters, J., Williams, D., Barber, C., Berbesque, N., Blanchard, B., Bowles, N., Caldejon, S. D., Casal, L., Cho, A., Cross, S., Dang, C., Dolbeare, T., Edwards, M., Galbraith, J., Gaudreault, N., Gilbert, T. L., Griffin, F., Hargrave, P., Howard, R., Huang, L., Jewell, S., Keller, N., Knoblich, U., Larkin, J. D., Larsen, R., Lau, C., Lee, E., Lee, F., Leon, A., Li, L., Long, F., Luviano, J., Mace, K., Nguyen, T., Perkins, J., Robertson, M., Seid, S., Shea-Brown, E., Shi, J., Sjoquist, N., Slaughterbeck, C., Sullivan, D., Valenza, R., White, C., Williford, A., Witten, D. M., Zhuang, J., Zeng, H., Farrell, C., Ng, L., Bernard, A., Phillips, J. W., Reid, R. C., and Koch, C. (2020). A large-scale standardized physiological survey reveals functional organization of the mouse visual cortex. *Nature Neuroscience*, 23(1):138–151.

[Dean, 2021] Dean, J. (2021). Introducing pathways: A next-generation ai architecture. `https://blog.google/technology/ai/introducing-pathways-next-generation-ai-architecture/`.

[Deng et al., 2009] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*.

[DiCarlo et al., 2012] DiCarlo, J., Zoccolan, D., and Rust, N. (2012). How does the brain solve visual object recognition? *Neuron*, 73(3):415 – 434.

[Diedrichsen and Kriegeskorte, 2017] Diedrichsen, J. and Kriegeskorte, N. (2017). Representational models: A common framework for understanding encoding, pattern-component, and representational-similarity analysis. *PLOS Computational Biology*, 13(4):1–33.

[Durand et al., 2016] Durand, S., Iyer, R., Mizuseki, K., de Vries, S., Mihalas, S., and Reid, R. C. (2016). A comparison of visual response properties in the lateral geniculate nucleus and primary visual cortex of awake and anesthetized mice. *The Journal of neuroscience : the official journal of the Society for Neuroscience*, 36(48):12144–12156.

[Erhan et al., 2009] Erhan, D., Bengio, Y., Courville, A., and Vincent, P. (2009). Visualizing higher-layer features of a deep network. *Technical Report, Univeristé de Montréal*.

[Erö et al., 2018] Erö, C., Gewaltig, M.-O., Keller, D., and Markram, H. (2018). A cell atlas for the mouse brain. *Frontiers in Neuroinformatics*, 12:84.

[Evangelio et al., 2018] Evangelio, M., García-Amado, M., and Clascá, F. (2018). Thalamocortical projection neuron and interneuron numbers in the visual thalamic nuclei of the adult c57bl/6 mouse. *Frontiers in Neuroanatomy*, 12:27.

[Felleman and van Essen, 1991] Felleman, D. J. and van Essen, D. C. (1991). Distributed hierarchical processing in the primate cerebral cortex. *Cerebral cortex*, 1 1:1–47.

[Fukushima, 1988] Fukushima, K. (1988). Neocognitron: A hierarchical neural network capable of visual pattern recognition. *Neural Networks*, 1(2):119 – 130.

[Glickfeld and Olsen, 2017] Glickfeld, L. L. and Olsen, S. R. (2017). Higher-order areas of the mouse visual cortex. *Annual Review of Vision Science*, 3(1):251–273. PMID: 28746815.

[Goodale and Milner, 1992] Goodale, M. A. and Milner, A. (1992). Separate visual pathways for perception and action. *Trends in Neurosciences*, 15(1):20–25.

[Goodfellow et al., 2016] Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press.

[Graves et al., 2013] Graves, A., Mohamed, A.-r., and Hinton, G. (2013). Speech recognition with deep recurrent neural networks. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 6645–6649.

[Güçlü and van Gerven, 2015] Güçlü, U. and van Gerven, M. A. J. (2015). Deep neural networks reveal a gradient in the complexity of neural representations across the ventral stream. *Journal of Neuroscience*, 35(27):10005–10014.

[Harris et al., 2019] Harris, J. A., Mihalas, S., Hirokawa, K. E., Whitesell, J. D., Choi, H., Bernard, A., Bohn, P., Caldejon, S., Casal, L., Cho, A., Feiner, A., Feng, D., Gaudreault, N., Gerfen, C. R., Graddis, N., Groblewski, P. A., Henry, A. M., Ho, A., Howard, R., Knox, J. E., Kuan, L., Kuang, X., Lecoq, J., Lesnar, P., Li, Y., Luviano, J., McConoughey, S., Mortrud, M. T., Naeemi, M., Ng, L., Oh, S. W., Ouellette, B., Shen, E., Sorensen, S. A., Wakeman, W., Wang, Q., Wang, Y., Williford, A., Phillips, J. W., Jones, A. R., Koch, C., and Zeng, H. (2019). Hierarchical organization of cortical and thalamic connectivity. *Nature*, 575(7781):195–202.

[Harris et al., 2018] Harris, J. A., Mihalas, S., Hirokawa, K. E., Whitesell, J. D., Knox, J., Bernard, A., Bohn, P., Caldejon, S., Casal, L., Cho, A., Feng, D., Gaudreault, N., Graddis, N., Groblewski, P. A., Henry, A., Ho, A., Howard, R., Kuan, L., Lecoq, J., Luviano, J., McConoghy, S., Mortrud, M., Naeemi, M., Ng, L., Oh, S. W., Ouellette, B., Sorensen, S., Wakeman, W., Wang, Q., Williford, A., Phillips, J., Koch, C., and Zeng, H. (2018). The organization of intracortical connections by layer and cell class in the mouse brain. *bioRxiv*.

[Harris et al., 2016] Harris, K. D., Mihalas, S., and Shea-Brown, E. (2016). High resolution neural connectivity from incomplete tracing data using nonnegative spline regression. In Lee, D. D., Sugiyama, M., Luxburg, U. V., Guyon, I., and Garnett, R., editors, *Advances in Neural Information Processing Systems 29*, pages 3099–3107. Curran Associates, Inc.

[He et al., 2015] He, K., Zhang, X., Ren, S., and Sun, J. (2015). Deep residual learning for image recognition. *CoRR*, abs/1512.03385.

[He et al., 2016] He, K., Zhang, X., Ren, S., and Sun, J. (2016). Identity mappings in deep residual networks. *CoRR*, abs/1603.05027.

[Hénaff et al., 2019] Hénaff, O. J., Goris, R. L. T., and Simoncelli, E. P. (2019). Perceptual straightening of natural videos. *Nature Neuroscience*.

[Hornik et al., 1989] Hornik, K., Stinchcombe, M., and White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Netw.*, 2(5):359–366.

[Huberman and Niell, 2011] Huberman, A. D. and Niell, C. M. (2011). What can mice tell us about how vision works? *Trends in Neurosciences*, 34(9):464 – 473.

[Jewell and Witten, 2018] Jewell, S. and Witten, D. (2018). Exact spike train inference via ℓ_0 optimization. *Ann. Appl. Stat.*, 12(4):2457–2482.

[Jewell et al., 2019] Jewell, S. W., Hocking, T. D., Fearnhead, P., and Witten, D. M. (2019). Fast nonconvex deconvolution of calcium imaging data. *Biostatistics*.

[Kell et al., 2018] Kell, A. J., Yamins, D. L., Shook, E. N., Norman-Haignere, S. V., and McDermott, J. H. (2018). A task-optimized neural network replicates human auditory behavior, predicts brain responses, and reveals a cortical processing hierarchy. *Neuron*, 98(3):630 – 644.e16.

[Khaligh-Razavi and Kriegeskorte, 2014a] Khaligh-Razavi, S.-M. and Kriegeskorte, N. (2014a). Deep supervised, but not unsupervised, models may explain it cortical representation. *PLOS Computational Biology*, 10(11).

[Khaligh-Razavi and Kriegeskorte, 2014b] Khaligh-Razavi, S.-M. and Kriegeskorte, N. (2014b). Deep supervised, but not unsupervised, models may explain it cortical representation. *PLOS Computational Biology*, 10(11):1–29.

[Knox et al., 2019] Knox, J. E., Harris, K. D., Graddis, N., Whitesell, J. D., Zeng, H., Harris, J. A., Shea-Brown, E., and Mihalas, S. (2019). High-resolution data-driven model of the mouse connectome. *Network Neuroscience*, 3(1):217–236.

[Kornblith et al., 2019] Kornblith, S., Norouzi, M., Lee, H., and Hinton, G. (2019). Similarity of Neural Network Representations Revisited. In Chaudhuri, K. and Salakhutdinov, R., editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 3519–3529, Long Beach, California, USA.

[Kriegeskorte, 2015] Kriegeskorte, N. (2015). Deep neural networks: A new framework for modeling biological vision and brain information processing. *Annual Review of Vision Science*, 1(1):417–446. PMID: 28532370.

[Krizhevsky, 2012] Krizhevsky, A. (2012). Learning multiple layers of features from tiny images. *University of Toronto*.

[Krizhevsky et al., 2012] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In Pereira, F., Burges, C. J. C., Bottou, L., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc.

[Kubilius et al., 2019] Kubilius, J., Schrimpf, M., Kar, K., Rajalingham, R., Hong, H., Majaj, N., Issa, E., Bashivan, P., Prescott-Roy, J., Schmidt, K., Nayebi, A., Bear, D., Yamins, D. L., and DiCarlo, J. J. (2019). Brain-like object recognition with high-performing shallow recurrent anns. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems 32*, pages 12805–12816. Curran Associates, Inc.

[Kubilius et al., 2018] Kubilius, J., Schrimpf, M., Nayebi, A., Bear, D., Yamins, D. L. K., and DiCarlo, J. J. (2018). Cornet: Modeling the neural mechanisms of core object recognition. *bioRxiv*.

[Le and Yang, 2015] Le, Y. and Yang, X. S. (2015). Tiny imagenet visual recognition challenge.

[Levy and Reyes, 2012] Levy, R. B. and Reyes, A. D. (2012). Spatial profile of excitatory and inhibitory synaptic connectivity in mouse primary auditory cortex. *Journal of Neuroscience*, 32(16):5609–5619.

[Lindsay, 2020] Lindsay, G. W. (2020). Convolutional neural networks as a model of the visual system: Past, present, and future. *Journal of cognitive neuroscience*, page 1—15.

[Markram et al., 2015] Markram, H., Muller, E., Ramaswamy, S., Reimann, M., Abdellah, M., Sanchez, C., Ailamaki, A., Alonso-Nanclares, L., Antille, N., Arsever, S., Kahou, G., Berger, T., Bilgili, A., Buncic, N., Chalimourda, A., Chindemi, G., Courcol, J.-D., Delalondre, F., Delattre, V., Druckmann, S., Dumusc, R., Dynes, J., Eilemann, S., Gal, E., Gevaert, M., Ghobril, J.-P., Gidon, A., Graham, J., Gupta, A., Haenel, V., Hay, E., Heinis, T., Hernando, J., Hines, M., Kanari, L., Keller, D., Kenyon, J., Khazen, G., Kim, Y., King, J., Kisvarday, Z., Kumbhar, P., Lasserre, S., Le Bé, J.-V., Magalhães, B., Merchán-Pérez, A., Meystre, J., Morrice, B., Muller, J., Muñoz-Céspedes, A., Muralidhar, S., Muthurasa, K., Nachbaur, D., Newton, T., Nolte, M., Ovcharenko, A., Palacios, J., Pastor, L., Perin, R., Ranjan, R., Riachi, I., Rodríguez, J.-R., Riquelme, J., Rössert, C., Sfyrakis, K., Shi, Y., Shillcock, J., Silberberg, G., Silva, R., Tauheed, F., Telefont, M., Toledo-Rodriguez, M., Tränkler, T., Van Geit, W., Díaz, J., Walker, R., Wang, Y., Zaninetta, S., DeFelipe, J., Hill, S., Segev, I., and Schürmann, F. (2015). Reconstruction and simulation of neocortical microcircuitry. *Cell*, 163(2):456 – 492.

[Marshel et al., 2011] Marshel, J., Garrett, M., Nauhaus, I., and Callaway, E. (2011). Functional specialization of seven mouse visual cortical areas. *Neuron*, 72(6):1040 – 1054.

[Martin et al., 2001] Martin, D., Fowlkes, C., Tal, D., and Malik, J. (2001). A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, volume 2, pages 416–423 vol.2.

[Mcculloch and Pitts, 1943] Mcculloch, W. and Pitts, W. (1943). A logical calculus of ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5:127–147.

[Michaels et al., 2020] Michaels, J. A., Schaffelhofer, S., Agudelo-Toro, A., and Scherberger, H. (2020). A goal-driven modular neural network predicts parietofrontal neural dynamics during grasping. *Proceedings of the National Academy of Sciences*, 117(50):32124–32135.

[Milletari et al., 2016] Milletari, F., Navab, N., and Ahmadi, S.-A. (2016). V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *2016 Fourth International Conference on 3D Vision (3DV)*, pages 565–571.

[Mishkin et al., 1983] Mishkin, M., Ungerleider, L. G., and Macko, K. A. (1983). Object vision and spatial vision: two cortical pathways. *Trends in Neurosciences*, 6:414–417.

[Morcos et al., 2018] Morcos, A., Raghu, M., and Bengio, S. (2018). Insights on representational similarity in neural networks with canonical correlation. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems 31*, pages 5732–5741. Curran Associates, Inc.

[Nayebi et al., 2021] Nayebi, A., Kong, N. C. L., Zhuang, C., Gardner, J. L., Norcia, A. M., and Yamins, D. L. K. (2021). Unsupervised models of mouse visual cortex. *bioRxiv*.

[Oh et al., 2014] Oh, S. W., Harris, J. A., Ng, L., Winslow, B., Cain, N., Mihalas, S., Wang, Q., Lau, C., Kuan, L., Henry, A. M., Mortrud, M. T., Ouellette, B., Nguyen, T. N., Sorensen, S. A., Slaughterbeck, C. R., Wakeman, W., Li, Y., Feng, D., Ho, A., Nicholas, E., Hirokawa, K. E., Bohn, P., Joines, K. M., Peng, H., Hawrylycz, M. J., Phillips, J. W., Hohmann, J. G., Wohnoutka, P., Gerfen, C. R., Koch, C., Bernard, A., Dang, C., Jones, A. R., and Zeng, H. (2014). A mesoscale connectome of the mouse brain. *Nature*, 508(7495):207–214.

[Olah et al., 2017] Olah, C., Mordvintsev, A., and Schubert, L. (2017). Feature visualization. *Distill*. https://distill.pub/2017/feature-visualization.

[Olmos and Kingdom, 2004] Olmos, A. and Kingdom, F. A. A. (2004). A biologically inspired algorithm for the recovery of shading and reflectance images. *Perception*, 33(12):1463–1473.

[Ozbulak, 2019] Ozbulak, U. (2019). Pytorch cnn visualizations. `https://github.com/utkuozbulak/pytorch-cnn-visualizations`.

[Parisien et al., 2008] Parisien, C., Anderson, C. H., and Eliasmith, C. (2008). Solving the problem of negative synaptic weights in cortical models. *Neural computation*, 20(6):1473–1494.

[Paszke et al., 2019] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R.,

editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

[Pedregosa et al., 2011] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

[Pospisil et al., 2018] Pospisil, D. A., Pasupathy, A., and Bair, W. (2018). 'Artiphysiology' reveals v4-like shape tuning in a deep network trained for image classification. *eLife*, 7:e38242.

[Prusky et al., 2000] Prusky, G. T., West, P. W., and Douglas, R. M. (2000). Behavioral assessment of visual acuity in mice and rats. *Vision Research*, 40(16):2201 – 2209.

[Raghu et al., 2017] Raghu, M., Gilmer, J., Yosinski, J., and Sohl-Dickstein, J. (2017). Svcca: Singular vector canonical correlation analysis for deep learning dynamics and interpretability. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems 30*, pages 6076–6085. Curran Associates, Inc.

[Raghu and Schmidt, 2020] Raghu, M. and Schmidt, E. (2020). A survey of deep learning for scientific discovery. *ArXiv*, abs/2003.11755.

[Riesenhuber and Poggio, 1999] Riesenhuber, M. and Poggio, T. (1999). Hierarchical models of object recognition in cortex. *Nature Neuroscience*, 2(11):1019–1025.

[Ringach et al., 2002] Ringach, D. L., Shapley, R. M., and Hawken, M. J. (2002). Orientation selectivity in macaque v1: Diversity and laminar dependence. *Journal of Neuroscience*, 22(13):5639–5651.

[Russakovsky et al., 2015] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252.

[Sandbrink et al., 2020] Sandbrink, K. J., Mamidanna, P., Michaelis, C., Mathis, M. W., Bethge, M., and Mathis, A. (2020). Task-driven hierarchical deep neural network models of the proprioceptive pathway. *bioRxiv*.

[Saxe et al., 2014] Saxe, A. M., Mcclelland, J. L., and Ganguli, S. (2014). Exact solutions to the nonlinear dynamics of learning in deep linear neural network. In *In International Conference on Learning Representations*.

[Saxe et al., 2019] Saxe, A. M., McClelland, J. L., and Ganguli, S. (2019). A mathematical theory of semantic development in deep neural networks. *Proceedings of the National Academy of Sciences*, 116(23):11537–11546.

[Schrimpf et al., 2018] Schrimpf, M., Kubilius, J., Hong, H., Majaj, N. J., Rajalingham, R., Issa, E. B., Kar, K., Bashivan, P., Prescott-Roy, J., Schmidt, K., Yamins, D. L. K., and DiCarlo, J. J. (2018). Brain-score: Which artificial neural network for object recognition is most brain-like? *bioRxiv*.

[Senior et al., 2020] Senior, A., Evans, R., Jumper, J., Kirkpatrick, J., Sifre, L., Green, T., Qin, C., Žídek, A., Nelson, A., Bridgland, A., Penedones, H., Petersen, S., Simonyan, K., Crossan, S., Kohli, P., Jones, D., Silver, D., Kavukcuoglu, K., and Hassabis, D. (2020). Improved protein structure prediction using potentials from deep learning. *Nature*, 577:1–5.

[Serre et al., 2007] Serre, T., Oliva, A., and Poggio, T. (2007). A feedforward architecture accounts for rapid categorization. *Proceedings of the National Academy of Sciences*, 104(15):6424–6429.

[Shi et al., 2019] Shi, J., Shea-Brown, E., and Buice, M. (2019). Comparison against task driven artificial neural networks reveals functional properties in mouse visual cortex. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems 32*, pages 5764–5774. Curran Associates, Inc.

[Shi et al., 2021] Shi, J., Tripp, B., Shea-Brown, E., Mihalas, S., and Buice, M. (2021). Cnn mousenet: A biologically constrained convolutional neural network model for mouse visual cortex. *bioRxiv*.

[Silver et al., 2016] Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., van den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T., Leach, M., Kavukcuoglu, K., Graepel, T., and Hassabis, D. (2016). Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489.

[Simonyan and Zisserman, 2014] Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556.

[Springenberg et al., 2014] Springenberg, J. T., Dosovitskiy, A., Brox, T., and Riedmiller, M. (2014). Striving for simplicity: The all convolutional net.

[Stanford,] Stanford. http://cs231n.stanford.edu/.

[Stepanyants et al., 2007] Stepanyants, A., Hirsch, J. A., Martinez, L. M., Kisvárday, Z. F., Ferecskó, A. S., and Chklovskii, D. B. (2007). Local Potential Connectivity in Cat Primary Visual Cortex. *Cerebral Cortex*, 18(1):13–28.

[Tripp, 2019] Tripp, B. (2019). Approximating the architecture of visual cortex in a convolutional network. *Neural Computation*, 31(8):1551–1591. PMID: 31260392.

[Tripp and Eliasmith, 2016] Tripp, B. and Eliasmith, C. (2016). Function approximation in inhibitory networks. *Neural Networks*, 77:95–106.

[van Hateren and van der Schaaf, 1998] van Hateren, J. H. and van der Schaaf, A. (1998). Independent component filters of natural images compared with simple cells in primary visual cortex. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, 265(1394):359–366.

[Vinje and Gallant, 2000] Vinje, W. E. and Gallant, J. L. (2000). Sparse coding and decorrelation in primary visual cortex during natural vision. *Science*, 287(5456):1273–1276.

[Vinken and Op de Beeck, 2020] Vinken, K. and Op de Beeck, H. (2020). Deep neural networks point to mid-level complexity of rodent object vision. *Journal of Vision*, 20(11):417–417.

[Wang et al., 2020] Wang, Q., Ding, S.-L., Li, Y., Royall, J., Feng, D., Lesnar, P., Graddis, N., Naeemi, M., Facer, B., Ho, A., Dolbeare, T., Blanchard, B., Dee, N., Wakeman, W., Hirokawa, K. E., Szafer, A., Sunkin, S. M., Oh, S. W., Bernard, A., Phillips, J. W., Hawrylycz, M., Koch, C., Zeng, H., Harris, J. A., and Ng, L. (2020). The allen mouse brain common coordinate framework: A 3d reference atlas. *Cell*, 181(4):936–953.e20.

[Wang et al., 2011] Wang, Q., Gao, E., and Burkhalter, A. (2011). Gateways of ventral and dorsal streams in mouse visual cortex. *Journal of Neuroscience*, 31(5):1905–1918.

[Wang et al., 2012] Wang, Q., Sporns, O., and Burkhalter, A. (2012). Network analysis of corticocortical connections reveals ventral and dorsal processing streams in mouse visual cortex. *Journal of Neuroscience*, 32(13):4386–4399.

[Williams et al., 2021] Williams, A. H., Kunz, E., Kornblith, S., and Linderman, S. W. (2021). Generalized shape metrics on neural representations.

[Wu et al., 2016] Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., et al. (2016). Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

[Yamins and DiCarlo, 2016] Yamins, D. L. K. and DiCarlo, J. J. (2016). Using goal-driven deep learning models to understand sensory cortex. *Nature Neuroscience*, 19:356.

[Yamins et al., 2014] Yamins, D. L. K., Hong, H., Cadieu, C. F., Solomon, E. A., Seibert, D., and DiCarlo, J. J. (2014). Performance-optimized hierarchical models predict neural responses in higher visual cortex. *Proceedings of the National Academy of Sciences*, 111(23):8619–8624.

[Zhang et al., 2019] Zhang, A., Lipton, Z. C., Li, M., and Smola, A. J. (2019). *Dive into Deep Learning*. `http://www.d2l.ai`.

[Zhuang et al., 2017a] Zhuang, C., Kubilius, J., Hartmann, M. J., and Yamins, D. L. (2017a). Toward goal-driven neural network models for the rodent whisker-trigeminal system. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30, pages 2555–2565. Curran Associates, Inc.

[Zhuang et al., 2017b] Zhuang, J., Ng, L., Williams, D., Valley, M., Li, Y., Garrett, M., and Waters, J. (2017b). An extended retinotopic map of mouse cortex. *eLife*, 6:e18372.

[Zoccolan et al., 2009] Zoccolan, D., Oertelt, N., DiCarlo, J. J., and Cox, D. D. (2009). A rodent model for the study of invariant visual object recognition. *Proceedings of the National Academy of Sciences*, 106(21):8748–8753.