

A Firing-Rate Neural Network Model for the Primary Visual Cortex and Its Learning Study

Thesis submitted to
Tsinghua University
in partial fulfillment of the requirement
for the degree of
Master of Science
in
Biomedical Engineering

by

Huimiao Chen

Thesis Supervisor: Professor Sen Song

June, 2024

ABSTRACT

Computational models of the primary visual cortex (V1) play a crucial role in unravelling the neural processing mechanisms underlying visual perception. However, spiking neural network (SNN) models, which are biologically realistic, often suffer from poor computational efficiency. To address this challenge, this thesis uses biologically-inspired, computationally efficient firing-rate neural network models of V1 and investigates their learning capabilities in the context of visual processing, enabling the exploration of neural mechanisms at scale.

The study builds discrete form firing-rate models and applies them to the learning of V1 neural networks based on real data from the Allen Institute for Brain Science. The learning capabilities of the models are evaluated through three designed visual tasks: fine-orientation detection, image classification, and visual change detection. Multiple learning scenarios and settings are explored to assess the models' performance under different conditions.

Analysis of synaptic weight distributions after training suggests that a diverse range of weights is beneficial for learning complex patterns, and the observed log-normal distribution of weights indicates biological realism. Comparative analysis of 3k- and 6k-neuron models reveals that enhanced network connectivity and flexibility in synaptic weights significantly contribute to learning effectiveness, while neuronal diversity contributes slightly to the memory capability of a network. The incorporation of lateral geniculate nucleus (LGN) filters for visual input preprocessing is found to effectively enhance the model's ability to capture and utilize temporal information, which is crucial in visual change detection tasks.

The study investigates the impact of network size and LGN preprocessing on model performance, demonstrating that increasing the network size alone does not lead to significant improvements in performance without LGN preprocessing. LGN filters alone do not significantly contribute to model convergence. However, increasing the network size in conjunction with LGN preprocessing can better improve the model's performance and help the model converge.

The trained models exhibit robustness to internal and external noise to some extent. The effectiveness of LGN filters in improving the model's robustness to internal noise is

ABSTRACT

relatively modest compared to the benefits of increasing the network size. Networks with higher recurrency demonstrate strong resilience when confronted with internal noise. The internal representations and activation patterns of neurons in response to external noise reveal that the principal components capturing the most variance in the responses remain dominant despite the presence of noise, still following a power law distribution. This suggests that the model strikes a balance between achieving high precision in its representations and maintaining robustness to noise, reminiscent of the trade-offs observed in biological neural systems.

This thesis, by combining biological realism and computational efficiency to develop firing-rate models of V1, successfully simulates various visual tasks and provides a feasible framework for exploring large-scale neural networks more efficiently compared to existing SNN frameworks. The findings regarding the model's learning capabilities provide valuable insights into the computational aspects of neural modeling. The work offers a foundation for further research in biologically plausible visual neural modeling.

Keywords: firing-rate models; primary visual cortex; neural network learning; visual processing; LGN filters

TABLE OF CONTENTS

摘要.....	I
ABSTRACT	II
TABLE OF CONTENTS	IV
LIST OF FIGURES.....	VII
LIST OF TABLES	XI
LIST OF SYMBOLS AND ACRONYMS	XIII
CHAPTER 1 INTRODUCTION	1
1.1 Background and Objectives	1
1.1.1 Neural Modeling Framework	1
1.1.2 Visual Systems.....	4
1.2 Literature Review.....	6
1.2.1 Neural Modeling and Learning.....	6
1.2.2 V1 Modeling and Learning	9
1.3 Main Work.....	14
1.3.1 Research Motivation.....	14
1.3.2 Focus and Highlights	16
CHAPTER 2 DATA-BASED FIRING-RATE V1 MODEL	19
2.1 From Spiking Models to Firing-Rate Models	19
2.2 Discrete Form Firing-Rate Model.....	22
2.3 Data Description and Experiment Settings.....	25
2.4 Visual Tasks for Learning	29
2.5 Comparative Scenarios and Settings.....	32
2.6 Learning of Models	35
2.6.1 3k-Neuron Model Learning	35
2.6.2 6k-Neuron Model Learning	40
2.7 Comparison of 3k- and 6k-Neuron Models	43
2.8 Chapter Summary	45

TABLE OF CONTENTS

CHAPTER 3 APPLYING LGN FILTERS TO V1 MODEL	47
3.1 LGN Filters.....	47
3.2 Learning of Models with LGN	52
3.2.1 3k-Neuron Model Learning with LGN	52
3.2.2 6k-Neuron Model Learning with LGN	55
3.3 Comparison of 3k- and 6k-Neuron Models with LGN	58
3.4 Comparison of Models with and without LGN.....	60
3.5 Chapter Summary	63
CHAPTER 4 DISCUSSIONS.....	65
4.1 Weight Distributions	65
4.2 Firing Rates.....	68
4.3 Internal Noise	70
4.4 External Noise	75
4.5 Neuronal Responses.....	78
4.6 Comparison with SNNs and ANNs	80
4.7 Simple and Complex Cells	81
4.8 Chapter Summary	82
CHAPTER 5 CONCLUSIONS	85
REFERENCES.....	87
APPENDIX A EXAMPLES OF TRAINING AND TEST LOSS	96
APPENDIX B FIRST-ORDER EXPONENTIAL INTEGRATOR METHOD.....	100
APPENDIX C SUPPLEMENTARY LGN RESPONSES	102
APPENDIX D SUPPLEMENTARY WEIGHT DISTRIBUTIONS	106
APPENDIX E WEIGHT MEAN DISTRIBUTION OF NEURONAL CLASS PAIRS	110
APPENDIX F SUPPLEMENTARY NEURONAL RESPONSES RESULTS	116
ACKNOWLEDGEMENTS	122
声 明.....	123
RESUME.....	124
COMMENTS FROM THESIS SUPERVISOR.....	125

TABLE OF CONTENTS

RESOLUTION OF THESIS DEFENSE COMMITTEE	127
--	-----

LIST OF FIGURES

Figure 2.1	V1 neural network visualization and cell classes ^[76]	24
Figure 2.2	Fitted relative connection probability of each pair of neuron classes versus inter-neuron lateral distance ^[76-77]	25
Figure 2.3	Synaptic connectivity matrix for neuronal classes (3k neurons)	26
Figure 2.4	Synaptic connectivity probability matrix for neuronal classes (3k neurons)	27
Figure 2.5	Synaptic connectivity matrix for neuronal classes (6k neurons)	28
Figure 2.6	Synaptic connectivity probability matrix for neuronal classes (6k neurons)	28
Figure 2.7	The activation function for transformation from currents to rates	29
Figure 2.8	Temporal sequence of visual stimuli for the fine-orientation detection task	30
Figure 2.9	Temporal sequence of visual stimuli for the image classification task	30
Figure 2.10	Temporal sequence of visual stimuli for the visual change task	31
Figure 2.11	Epoch-wise test accuracy for the default and other three scenarios: average and task-specific performance across three tasks (3k neurons)	35
Figure 2.12	Epoch-wise test accuracy for the default scenario and other three output layer settings: average and task-specific performance across three tasks (3k neurons)	38
Figure 2.13	Epoch-wise test accuracy for the default scenario and random output: average and task-specific performance across three tasks (3k neurons).....	38
Figure 2.14	Epoch-wise test accuracy for the default scenario and hybrid learning: average and task-specific performance across three tasks (3k neurons).....	39
Figure 2.15	Epoch-wise test accuracy for the default and other three scenarios: average and task-specific performance across three tasks (6k neurons)	40
Figure 2.16	Epoch-wise test accuracy for the default scenario and other three output layer settings: average and task-specific performance across three tasks (6k neurons)	41
Figure 2.17	Epoch-wise test accuracy for the default scenario and random output: average and task-specific performance across three tasks (6k neurons).....	42
Figure 2.18	Epoch-wise test accuracy for the default scenario and hybrid learning: average and task-specific performance across three tasks (6k neurons).....	43
Figure 2.19	Average performance of all tasks by scenario and neuron count	44

LIST OF FIGURES

Figure 2.20	Average performance of all tasks by output layer and neuron count	45
Figure 3.1	Schematic of visual input processing of LGN filters ^[54]	47
Figure 3.2	Examples of spatial and temporal filters for LGN cells ^[55]	48
Figure 3.3	Schematic diagram of the data preprocessing from images to rates of both no LGN and with LGN settings	48
Figure 3.4	Rate responses of OFF cells in our LGN model to visual stimuli from the fine-orientation detection task	49
Figure 3.5	Rate responses of all types of ON cells in our LGN model to visual stimuli from the fine-orientation detection task	50
Figure 3.6	Spike responses of all cells in our LGN model to visual stimuli from the fine-orientation detection task	51
Figure 3.7	Epoch-wise test accuracy for the default and other three scenarios: average and task-specific performance across three tasks (3k neurons, with LGN)	52
Figure 3.8	Epoch-wise test accuracy for the default scenario and other three output layer settings: average and task-specific performance across three tasks (3k neu- rons, with LGN)	53
Figure 3.9	Epoch-wise test accuracy for the default scenario and random output: av- erage and task-specific performance across three tasks (3k neurons, with LGN)	54
Figure 3.10	Epoch-wise test accuracy for the default scenario and hybrid learning: av- erage and task-specific performance across three tasks (3k neurons, with LGN)	55
Figure 3.11	Epoch-wise test accuracy for the default and other three scenarios: average and task-specific performance across three tasks (6k neurons, with LGN)	56
Figure 3.12	Epoch-wise test accuracy for the default scenario and other three output layer settings: average and task-specific performance across three tasks (6k neurons, with LGN)	57
Figure 3.13	Epoch-wise test accuracy for the default scenario and random output: av- erage and task-specific performance across three tasks (6k neurons, with LGN)	57
Figure 3.14	Epoch-wise test accuracy for the default scenario and hybrid learning: av- erage and task-specific performance across three tasks (6k neurons, with LGN)	58

LIST OF FIGURES

Figure 3.15	Average performance of all tasks by scenario and neuron count (with LGN).....	59
Figure 3.16	Average performance of all tasks by output layer and neuron count (with LGN).....	60
Figure 3.17	Average performance of all tasks by scenario and LGN (3k neurons)....	61
Figure 3.18	Average performance of all tasks by output layer and LGN (3k neurons)62	
Figure 3.19	Average performance of all tasks by scenario and LGN (6k neurons)....	63
Figure 3.20	Average performance of all tasks by output layer and LGN (6k neurons)64	
Figure 4.1	Weight distributions of initial values, multiple scenarios, and the hybrid learning (3k neurons).....	66
Figure 4.2	Log-transformed weight distributions of initial values, multiple scenarios, and the hybrid learning (3k neurons).....	67
Figure 4.3	Firing rate distribution of all excitatory and inhibitory neuron classes in different layers (3k neurons).....	68
Figure 4.4	Firing rate distribution of all excitatory and inhibitory neuron classes in different layers (6k neurons).....	68
Figure 4.5	Firing rate distribution of all excitatory and inhibitory neuron classes in different layers (3k neurons, with LGN).....	69
Figure 4.6	Firing rate distribution of all excitatory and inhibitory neuron classes in different layers (6k neurons, with LGN).....	69
Figure 4.7	Impact of slow and quick noise on model performance under the default scenario	71
Figure 4.8	Impact of slow and quick noise on model performance under the “uniform neurons” scenario.....	72
Figure 4.9	Impact of slow and quick noise on model performance under the “random synapses” scenario.....	73
Figure 4.10	Impact of slow and quick noise on model performance under the “no Dale’s law” scenario.....	74
Figure 4.11	Impact of slow and quick noise on model performance under the hybrid learning	75
Figure 4.12	Examples of noisy MNIST images used to evaluate the model.....	76
Figure 4.13	Classification accuracy of the model on noisy MNIST images under the default scenario.....	76

LIST OF FIGURES

Figure 4.14	Classification accuracy of the model on noisy MNIST images under the “uniform neurons” scenario.....	77
Figure 4.15	Classification accuracy of the model on noisy MNIST images under the “random synapses” scenario	77
Figure 4.16	Classification accuracy of the model on noisy MNIST images under the “no Dale’s law” scenario	78
Figure 4.17	Classification accuracy of the model on noisy MNIST images under the hybrid learning.....	78
Figure 4.18	PCA results of neuronal responses to clean and noisy MNIST images and of the response difference (default scenario)	83
Figure 4.19	Significance analysis of power law exponents of three PCA fit curves in all models	84
Figure 4.20	Significance analysis of dot product of between top 5 principal components of neuronal response to original and noisy input Significance analysis using dot product between principal components of neuronal responses to original and noisy inputs (default scenario)	84

LIST OF TABLES

Table 2.1	Distribution of excitatory and inhibitory neurons in network layers (3k neurons)	26
Table 2.2	Distribution of excitatory and inhibitory neurons in network layers (6k neurons)	27
Table 2.3	Numerical test accuracy for the default and other three scenarios: average and task-specific performance across three tasks (3k neurons).....	36
Table 2.4	Numerical test accuracy for the default scenario and other three output layer settings: average and task-specific performance across three tasks (3k neurons)	37
Table 2.5	Numerical test accuracy for the default scenario and hybrid learning: average and task-specific performance across three tasks (3k neurons).....	39
Table 2.6	Numerical test accuracy for the default and other three scenarios: average and task-specific performance across three tasks (6k neurons).....	40
Table 2.7	Numerical test accuracy for the default scenario and other three output layer settings: average and task-specific performance across three tasks (6k neurons)	42
Table 2.8	Numerical test accuracy for the default scenario and hybrid learning: average and task-specific performance across three tasks (6k neurons).....	43
Table 3.1	Numerical test accuracy for the default and other three scenarios: average and task-specific performance across three tasks (3k neurons, with LGN)	53
Table 3.2	Numerical test accuracy for the default scenario and other three output layer settings: average and task-specific performance across three tasks (3k neurons, with LGN).....	54
Table 3.3	Numerical test accuracy for the default scenario and hybrid learning: average and task-specific performance across three tasks (3k neurons, with LGN) .	55
Table 3.4	Numerical test accuracy for the default and other three scenarios: average and task-specific performance across three tasks (6k neurons, with LGN)	56
Table 3.5	Numerical test accuracy for the default scenario and other three output layer settings: average and task-specific performance across three tasks (6k neurons, with LGN).....	58

LIST OF TABLES

Table 3.6 Numerical test accuracy for the default scenario and hybrid learning: average and task-specific performance across three tasks (6k neurons, with LGN). 58

LIST OF SYMBOLS AND ACRONYMS

ADAM	Adaptive moment estimation
AI	Artificial intelligence
ANN	Artificial neural network
BMTK	Brain modeling toolkit
BPTT	Backpropagation through time
CIFAR-10	Canadian Institute for Advanced Research 10 dataset
CNN	Convolutional neural network
GLIF	Generalized leaky integrate-and-fire
GPU	Graphics processing unit
HH	Hodgkin-Huxley
ImageNet	ImageNet dataset
IoT	Internet of things
LGN	Lateral geniculate nucleus
LIF	Leaky integrate-and-fire
LNP	Linear-nonlinear-Poisson
MNIST	Modified National Institute of Standards and Technology dataset
ODE	Ordinary differential equation
ON	Light increment
OFF	Light decrement
PCA	Principal component analysis
ReLU	Rectified linear unit
ResNet	Residual network
RNN	Recurrent neural network
SCN	Suprachiasmatic nucleus
SGD	Stochastic gradient descent
SNN	Spiking neural network
sON	Sustained ON (light increment)
sOFF	Sustained OFF (light decrement)
STDP	Spike-timing-dependent plasticity
TF	Temporal frequency
tON	Transient ON (light increment)
tOFF	Transient OFF (light decrement)

LIST OF SYMBOLS AND ACRONYMS

V1	Primary visual cortex
VGG	Visual Geometry Group network

CHAPTER 1 INTRODUCTION

1.1 Background and Objectives

1.1.1 Neural Modeling Framework

Neural network models can be broadly categorized into two types: spiking models and firing-rate models. They differ primarily in the level of abstraction and biological detail they incorporate.

Spiking models attempt to recreate the actual electrical activity of neurons by modeling the timing of each individual spike. They are more biologically realistic, incorporating the intricate timing of action potentials and the effect of each spike on postsynaptic neurons^[1-2]. This allows spiking models to offer a nuanced representation of neuronal activity, emphasizing the precise spike timing believed to be crucial for various neural computations^[3-4]. Additionally, these models are capable of exhibiting complex temporal dynamics such as bursting and synchronization, commonly observed in actual neural systems^[5]. The capacity to use spike timing for encoding information adds a significant dimension to these models, a feature may be absent in firing-rate models^[6-7].

However, spiking models come with notable disadvantages. Due to the fine temporal resolution needed to accurately simulate individual spikes and their synaptic interactions, these models are computationally intensive^[8]. This demand for high computational resources or new neuromorphic hardware makes spiking models less practical for certain applications^[9-10]. Furthermore, the inherent complexity of these models poses challenges in theoretical analysis and makes their integration with standard learning algorithms more difficult^[11-15]. This analytical complexity often limits their usability in broader computational neuroscience and neural network training contexts^[13-16].

Firing-rate models, on the other hand, represent neuronal activity by abstracting the complex temporal dynamics of individual spikes into a continuous average firing rate^[3,17-18]. This simplification translates the neuron's spiking behavior into a smooth variable over time, reflecting the average rate of action potentials per unit time. Such models are particularly advantageous due to their computational efficiency^[3,19-21]; by focusing on averaged rates rather than individual spikes, they require significantly less computational power and time. This efficiency facilitates the simulation of larger neural networks and the handling of more complex computational

tasks^[22-25], making these models highly suitable for exploring extensive systems where the precise timing of spikes is less critical.

Moreover, the smooth, continuous nature of the firing rate output in these models lends itself to mathematical tractability, simplifying the process of leveraging well-established learning algorithms in conjunction with the existing ones like backpropagation^[26-28]. This feature enables the analytical calculation of network dynamics, which might be cumbersome with spiking models due to their inclusion of rapid, detailed dynamics and a broader range of timescales. Additionally, firing-rate models often involve fewer adjustable parameters compared to spiking models, reducing the requirements for the memory of computation resources, such as graphics processing units (GPUs), and simplifying the process of parameter tuning^[25,29-31]. They also have the capability to produce stochastic sequences of spikes based on deterministically computed rates^[32-35], which can be crucial for certain types of neural modeling tasks. In scenarios where neural responses are effectively described by firing rates, the output of a model unit directly corresponds to the average firing rate of a group of represented neurons, offering a detailed and thorough explanation of the neural mechanisms and computations underlying particular cognitive abilities and processes orchestrated by the brain.

Firing-rate models may not fully capture the complexities of spike timing and inter-neuronal correlations^[36-40], aspects that are subtly important for a nuanced understanding of the nervous system. They are generally suitable for situations where neurons operate relatively independently, and precise spike timing is not crucial, which accounts for a significant portion of the modeling scenarios in computational neuroscience. This is largely due to the inherent robustness and redundancy of the brain, which allows it to maintain stable performance even when individual neurons exhibit variability in their firing patterns. The brain's resilience to noise and perturbations enables it to function effectively without relying on the precise timing of every single spike^[41]. Moreover, many cognitive processes, such as perception, decision-making, and memory, can be adequately modeled using firing-rate models, as they often depend on the collective activity of neural populations rather than the intricate dynamics of individual spikes^[42-43]. Therefore, while the abstraction to average rates facilitates computational efficiency and mathematical tractability^[44-45], it does mean that some of the finer temporal dynamics and the rich variability in neuronal behavior might not be fully represented^[25]. Basically, this simplification allows for broader applicational ease, though it occasionally overlooks some

detailed aspects of neural function^[24,46-47].

In practice, firing-rate models have been instrumental in advancing our understanding of complex neural processes. For instance, when investigating the neural circuits in the spinal cord that are responsible for processing pain signals, with a specific emphasis on allodynia, by employing firing-rate models, researchers aim to gain insights into the underlying mechanisms that contribute to the development of chronic pain when these circuits are disrupted^[24]; the application of firing-rate models to the suprachiasmatic nucleus (SCN) contributes to exploring how environmental light influences the activity of neurons in the SCN and the subsequent effects on circadian rhythms^[24]. The two examples use firing rate models to understand and develop treatments for conditions related to pain and circadian disruptions, highlighting the versatility of these models. Also, firing-rate models facilitate the exploration of how neural populations encode visual stimuli and adapt to changes in visual inputs^[48]. Similarly, firing-rate models have played a pivotal role in studies of memory and learning, modeling phenomena such as synaptic plasticity and the stability of short-term memory networks^[49-50]. Firing-rate models have proven to be invaluable in unraveling the intricacies of gamma rhythms in the brain^[25,51]. These models are particularly adept at shedding light on the complex interplay between various neuronal components. By leveraging the simplicity and efficiency of firing-rate models, researchers can delve deeper into the fundamental principles governing the synchronization of neural networks and the emergence of cross-frequency interactions. These models also serve as a foundation for the development of novel data-driven approaches, allowing scientists to bridge the gap between theoretical understanding and empirical observations. These applications demonstrate the model's ability to capture essential neural dynamics in a computationally efficient manner, making them valuable tools for neuroscience research.

In the future, the integration of firing-rate models with advanced machine learning techniques would present a promising avenue for enhancing their utility and application scope^[24]. This integration could lead to improved models that are capable of complex pattern recognition and predictive modeling, which are integral to both neuroscience research and the development of neural-inspired computing systems^[52]. Additionally, methodological improvements in parameter estimation could further refine these models, enabling them to better mimic observed neural behaviors and respond dynamically to changes in network conditions^[17]. Considering the development of hybrid models that

blend the detailed temporal resolution of spiking models with the computational efficiency of firing-rate models could also offer a balanced approach for modeling large-scale neural systems^[53]. Such advancements will not only expand the capabilities of firing-rate models but also enhance their precision and relevance in contemporary neuroscience research.

1.1.2 Visual Systems

Visual systems are among the most complex and highly studied systems in neuroscience, primarily responsible for the reception, processing, and interpretation of visual information. These systems span multiple levels of neural architecture, from the retina to the lateral geniculate nucleus (LGN) and through to various cortical areas, including the primary visual cortex (V1). Each layer processes different aspects of visual information, such as color, motion, and spatial relationships, before integrating this data to form a coherent visual perception. Neural models, particularly firing-rate models, play a crucial role in simulating and understanding these intricate processes, providing insights into both the normal functioning and pathological states of visual cognition.

Firing-rate models are pivotal in simulating the dynamics of visual processing, particularly in areas like the LGN and V1. In the LGN, these models help to understand how visual signals are filtered and relayed to the cortex^[54-55], mimicking how certain shapes or patterns are emphasized over others through retinal input. The V1 is known for its role in processing complex visual stimuli. Here, firing-rate models abstract the neuronal interactions into simplified rates of firing that correlate with various visual features such as orientation, motion, and spatial frequency^[56]. These models help elucidate how external stimuli influence neural firing rates, leading to perceptual decisions. For example, they have been instrumental in understanding phenomena like visual salience and the adaptive changes that occur in neuronal firing in response to varying visual contexts^[57-58]. This abstraction is crucial for studying how the cortex encodes visual information and for modeling phenomena like cortical response properties and the normalization processes that regulate neuronal activity^[59]. These processes are essential for understanding how neurons compete and cooperate in response to visual inputs, influencing everything from basic perception to complex decision-making processes.

On the other hand, the study of brain visual systems has profound implications for computer vision^[60-62], particularly in pattern recognition and image processing. One notable example is the Gabor filter, which is modeled after the receptive fields of simple cells in the V1^[63-64]. Gabor filters are widely used in computer vision for texture analysis,

edge detection, and feature extraction^[65-66]. Another technique, convolutional neural networks (CNNs), drawing inspiration from the architectural and functional principles governing the visual cortex, employs a series of interconnected processing layers that reflect the brain's hierarchical strategy for analyzing and interpreting visual stimuli.^[67-68]. The understanding of how the brain processes visual information through models like firing-rate models is thereby expected to provide a template that can enhance and guide the development of more efficient and robust computer vision systems^[60,69].

Despite significant advancements in our understanding of biological vision and its applications in computer vision, many problems remain unsolved. The visual system is incredibly complex, and there are still numerous aspects of visual processing that are not yet fully understood. As new discoveries continue to be made in biological vision systems, they will continue to contribute to the development of more advanced and biologically plausible computer vision techniques. Neural modeling, in particular, will play a crucial role in bridging the gap between biological vision and artificial vision systems. By creating computational models that more efficiently simulate the functioning of the visual cortex and other neural structures involved in visual processing, researchers can gain deeper insights into the underlying mechanisms of vision and develop more efficient and robust computer vision algorithms.

Firing-rate models are invaluable not only for their ability to simplify complex neural interactions but also for their utility in predicting and modifying system behavior under different sensory conditions. They are essential tools for dissecting the function of neural circuits without the computational cost required for more detailed spiking models. Additionally, these models facilitate a deeper understanding of how the brain adapts to changes in visual environments, impacting areas such as the development of adaptive visual prosthetics and therapies for visual impairments^[70-71]. Current models may oversimplify the diverse contributions of various neuron types and their interconnected dynamics^[1,72-75]. Future research should aim to integrate more biophysical properties and real data into firing-rate models and explore advanced hybrid models that combine the temporal resolution of spiking models with the computational efficiency of firing-rate models^[54,76-77]. Such advances may improve the accuracy of these models and bring new paradigms for understanding unsolved mysteries in visual neuroscience and enhancing computer vision technologies.

1.2 Literature Review

1.2.1 Neural Modeling and Learning

Replicating the intricate behaviors and information processing capabilities of neurons and neural networks found in living organisms has been a fundamental goal driving the field of computational neuroscience. Various approaches have been proposed, ranging from detailed biophysical models that capture the intricate mechanisms of individual neurons to more abstract models that aim to reproduce the essential computational properties of neural systems.

Spiking models strive to accurately simulate the dynamics of biological neurons by depicting neural activity through discrete, temporally precise events termed spikes, closely resembling the brain's natural computational principles. The most biophysically realistic spiking model is the Hodgkin-Huxley (HH) model, which describes the dynamics of ion channels and membrane potentials^[78-79]. Although the HH model provides a high degree of biophysical realism and accuracy in describing neural dynamics, its computational demands can be substantial, limiting its efficiency in large-scale simulations. To simplify and reduce computational complexity, researchers have developed some reduced HH models^[6,20,80]. These models retain key features of the original HH model but reduce the number of parameters and equations, making simulations more manageable. Reduced models are valuable for large-scale simulations and studies of neural networks. The leaky integrate-and-fire (LIF) model^[19,81] and the Izhikevich model^[82] have emerged as two prominent simplified spiking models, gaining extensive application in computational neuroscience research. Generalized leaky integrate-and-fire (GLIF) models extend the capabilities of the LIF framework to capture a broader range of neural dynamics. Researchers have developed an array of GLIF models with escalating complexity, discovering that the more advanced models were superior in predicting neuron responses to novel stimuli^[19,80,83-84]. A significant contribution of GLIF model research is the establishment of a simplified model database that enhances the understanding of neuronal behavior and supports the development of expansive network computational models. Some GLIF models extend the firing mechanism to a more statistical approach, incorporating probabilistic elements into the spike generation process. Instead of relying on a fixed threshold for spike initiation, these models introduce stochasticity by considering the probability of firing based on the membrane potential and other factors. This probabilistic approach allows for a more realistic representation of the variability and noise observed in biological neu-

rons. Specifically, the authors^[84] created a database of point models linked to the Allen Cell Types Database, designed to streamline the construction of system models incorporating multiple cell types. These models capture the essential spiking dynamics while being computationally more tractable compared to the original HH models.

Spiking neural networks (SNNs) extend spiking models to network-level descriptions, allowing the study of emergent properties and computations arising from the interactions between spiking neurons. SNNs have been used to model various aspects of neural processing, such as sensory coding^[85], memory^[86], and decision-making^[87]. Maass^[11] proposed that SNNs represent the 3rd generation of neural network architectures. SNNs offer a promising avenue for enhancing the future of artificial intelligence (AI), with several advantages that position them as efficient and capable models^[88-90]. First, SNNs are inherently energy-efficient due to their event-driven processing nature. This aligns well with the need for more sustainable and energy-conscious AI systems, making them suitable for edge devices and the Internet of things (IoT) applications. Second, its biological fidelity can enhance our understanding of neural computation and potentially lead to more biologically plausible AI systems. Third, SNNs excel at processing information over time. Their spike-timing-dependent plasticity (STDP) learning method^[91] enables them to adapt and respond to temporal patterns^[92], making them valuable for tasks that require temporal understanding, such as speech recognition and motion detection. Forth, SNNs naturally produce sparse spiking patterns, which can be advantageous for information encoding, reducing redundancy in data, and improving memory efficiency.

However, SNNs, although as energy-efficient alternatives to traditional artificial neural networks (ANNs), are not well compatible with existing computation devices and call for advanced neuromorphic hardware^[9-10,14-15,90,93-97], such as the memristors. Neuromorphic hardware, which is specifically designed to support SNN computations, is still in its early stages of development and requires further advancements to fully exploit the capabilities of SNNs^[94-95,98]. The complex dynamics and asynchronous nature of SNNs pose challenges for their efficient implementation on conventional computing architectures, which are optimized for synchronous and sequential processing^[90,99]. As a result, the development of SNNs has been limited by the availability of specialized neuromorphic hardware^[100-101], which is currently not as mature and accessible as the hardware used for ANNs^[93,96].

For the learning algorithms of SNNs, currently error backpropagation with strategi-

cally crafted surrogate gradients^[77,102-103] and transformation of ANNs into SNNs^[14,104] are two main approaches used by researchers for SNN training to tackle the non-differentiability challenge. In terms of the surrogate gradients, while this approach has shown promise in achieving results comparable to deep learning, it introduces inherent inaccuracies in the descent direction and sensitivity to function scale choices^[105-106], also posing compatibility challenges with leading neuromorphic devices^[9,13-14]. Although ANN-SNN transformation has been successfully applied to renowned architectures like VGG and ResNet, progressively closing the conversion gap on ImageNet^[107-111], constraints from ANNs affect SNN performance, such as longer inference times, limited suitability for streaming data, and prioritizing performance over exploring SNN dynamics for brain-inspired intelligence^[104,112].

In contrast to the gradient descent optimization techniques commonly employed in ANNs, such as stochastic gradient descent (SGD) and adaptive moment estimation (ADAM)^[113-114], along with normalization and distributed training for large-scale models in practical AI, SNNs lack universally recognized core learning algorithms.

In this context, firing-rate models offer several advantages over SNNs and other biologically detailed models. Firing-rate models provide a tractable framework for studying neural computation and learning in large-scale networks. This computational efficiency enables the exploration of network-level phenomena, such as the emergence of receptive fields, attractor dynamics, and effective learning in recurrent networks^[115-116].

Firing-rate models have also been successful in capturing a wide range of neural phenomena and behavioral functions. Moreover, firing-rate models have been applied to explain the dynamics of neural populations during cognitive tasks, such as attention^[59], and to decode neural activity for brain-machine interfaces. Firing-rate models that are precisely formulated have the capability to capture the average behavior of diverse neuronal populations interconnected through both chemical and electrical synapses. These models have been shown to be consistent with computational simulations conducted on extensive networks comprising a wide variety of spiking neurons that communicate via both electrical gap junctions and chemical neurotransmitter-mediated connections^[117]. Firing-rate models have also been used as a novel technique to train recurrent SNNs. By creating supplementary targets that guide the training process, firing-rate models enable the performance enhancement of SNNs and this capability offers fresh insights into the importance of these models and brings a valuable tool for constructing SNNs or hy-

brid networks^[118]. Moreover, by introducing elements such as coordinated spike timing, interconnected neuronal properties, and the inherent variability of limited-size neural populations, the shortcomings of firing-rate models can be mitigated to a certain degree. These enhancements bring the models closer in line with the actual behavior of biological neural networks, ultimately leading to a more faithful representation of the complex computations occurring within the cortex^[119]. Some functions and applications of firing-rate models have already been mentioned in subsection 1.1.1. The ability of these models to capture a broad range of neural and behavioral phenomena highlights their explanatory power and practical utility.

Importantly, firing-rate models provide a bridge between biologically inspired neural networks and machine learning. By focusing on the computational properties of neural circuits rather than their detailed biophysical mechanisms, firing-rate models can inform the design of ANNs and learning algorithms^[67,120-121]. Insights from these models have led to the development of powerful machine learning techniques, which leverage the dynamics of recurrent neural networks for temporal processing and learning. Furthermore, firing-rate models have been used to study the properties of deep neural networks and to analyze their similarities and differences with biological neural networks^[122-123].

1.2.2 V1 Modeling and Learning

The study of brain modeling aims to understand the intricate workings of the brain by developing computational models that simulate its neural activity and functional properties. These models offer crucial understanding of the fundamental processes involved in neural computation and the emergence of cognitive abilities. One area of particular interest is the modeling of the V1, a key region responsible for early visual processing.

This part will highlight the progress made in brain modeling, particularly in the study of V1. Pioneering models in computational neuroscience introduced the groundbreaking concepts of receptive fields and orientation selectivity, which significantly advanced our comprehension of how visual information is processed and interpreted by the brain. Subsequent models incorporated excitatory-inhibitory interactions to capture the complex dynamics observed in V1 neurons. The introduction of STDP emphasized the role of precise temporal coding in shaping V1 connectivity and information processing. Recent advancements have focused on constructing large-scale and multiscale models that integrate anatomical, physiological, and functional data. These models provide a more comprehensive understanding of V1's structure-function relationships and have revealed insights into

the emergent properties of the brain. Overall, these advancements in brain modeling have significantly advanced our understanding of V1 and paved the way for further investigations into neural computation and the development of brain-inspired technologies.

1.2.2.1 Early Models of Orientation Selectivity

The pioneering work of Hubel and Wiesel in 1959^[124] laid the foundation for V1 modeling. This groundbreaking work uncovered the existence of orientation-selective neurons in the V1 and introduced the influential concept of receptive fields, providing pivotal insights into the early stages of visual processing in the brain. Utilizing meticulous receptive field mapping of individual neurons in cat striate cortex, they uncovered orientation selectivity as a key feature encoded among these cells. Specifically, V1 neurons respond optimally to bar or edge stimuli at particular angles, representing early orientation coding. Furthermore, a significant portion of neurons in V1 demonstrate directional selectivity, exhibiting heightened responses not only to contours of a specific preferred orientation but also to those moving through their receptive fields in a favored direction. Hubel and Wiesel also characterized the antagonistic center-surround structure, elucidating how orientation-tuned subregions interact to enhance contour detection. Altogether, their study demonstrated that even the earliest cortical visual area contains specialized detectors for elementary features, not merely undifferentiated light inputs. These foundational ideas revolutionized our understanding of how visual information is encoded and processed within the cerebral cortex. The receptive field mappings and selectivity discovered in this seminal work established foundations for models of hierarchical feature integration underlying visual perception.

Inspired by these findings, Fukushima in 1980^[125] introduced the classic “simple cell” model, which incorporated orientation tuning and spatial filters to simulate the responses of V1 neurons to visual stimuli. The Neocognitron, a self-organizing neural network model, was introduced to recognize patterns invariant to shifts in position, enabling robust pattern recognition capabilities. The author addressed a common challenge in pattern recognition, where objects might appear in different positions within an image. The Neocognitron model incorporated hierarchical layers of interconnected neurons with receptive fields that responded to specific features, such as edges or lines, at different spatial positions. Through a process of unsupervised learning and self-organization, the model could learn and recognize patterns even when they were shifted or partially occluded. The Neocognitron demonstrated promising results in pattern recognition tasks, showcasing its

potential for building robust and position-invariant recognition systems. This paper laid the foundation for subsequent developments in CNNs^[67-68], which are widely used today in various applications, including image and video recognition, as aforementioned.

1.2.2.2 Incorporating Excitatory-Inhibitory Interactions

To capture the complex dynamics observed in V1, subsequent models integrated excitatory and inhibitory interactions among neurons. Carandini et al. in 1997^[126] proposed a model that incorporated both feedforward and recurrent connections, enabling the simulation of spatial organization and contextual modulation observed in experimental studies. The proposed model shed light on the neural mechanisms responsible for essential response characteristics, including invariance to contrast variations and selectivity to specific orientations. Specifically, the paper investigated the response properties of simple cells in the macaque V1 and explored the role of linearity and normalization in their functioning. The authors conducted experiments to measure the responses of simple cells to visual stimuli of varying contrasts and sizes. They found that the responses of these cells could be accurately described by a linear model, where the output is a weighted sum of the inputs. Additionally, they observed that the responses of these cells were modulated by a divisive normalization process, which scaled down the responses based on the overall level of input activity. The findings suggest that linearity and normalization are essential mechanisms in the processing of visual information in the V1.

The incorporation of excitatory-inhibitory interactions in V1 models, as exemplified by Carandini et al.'s work, marked a significant advancement in understanding the complex dynamics of visual processing. The interplay between excitation and inhibition has been shown to play a crucial role in shaping the response properties of V1 neurons, such as orientation selectivity and contrast invariance^[127]. Moreover, the balance between excitation and inhibition has been implicated in the emergence of sparse coding and efficient representation of visual information^[128]. The inclusion of these interactions in computational models has facilitated the exploration of the underlying mechanisms and has provided valuable insights into the functioning of the primary visual cortex. Furthermore, these models have paved the way for more sophisticated architectures, such as hierarchical and recurrent networks, which have been instrumental in advancing our understanding of higher-level visual processing and perception^[129-130].

1.2.2.3 Spiking Neural Network Models

Advancements in computational power and increased understanding of neural dynamics led to the development of SNN models for V1. Many aspects and characteristics of SNNs have been discussed in the preceding sections.

Gerstner et al. in 1996^[131] used STDP^[91], which allowed for the learning and adaptation of receptive fields based on visual input. This work demonstrated the importance of precise spike timing in shaping the connectivity and activity patterns in V1. Specifically, the work used STDP neuronal learning rule to achieve precise temporal coding at the sub-millisecond timescale. The authors proposed a mathematical model that incorporates STDP. Through theoretical analysis and simulations, they demonstrated that this learning rule allows neurons to selectively respond to specific temporal patterns of inputs. By exploiting the temporal order of spikes, the proposed learning rule enables neurons to encode and process time-varying information with high temporal precision. SNNs and STDP provide fundamental insights into the mechanisms underlying precise temporal coding in the brain and has implications for understanding neural computation and information processing.

The development of SNN models for V1 has been a significant advancement in understanding the complex dynamics of visual processing. Incorporating the temporal aspects of neuronal firing and providing a more biologically realistic representation, SNNs have shed light on the mechanisms underlying the emergent properties of V1, such as orientation selectivity, spatial receptive fields, and contextual modulation. By leveraging the sparse and asynchronous nature of spiking activity, SNNs have the potential to reveal the computational principles employed by the brain for efficient visual information processing^[70,128,132].

1.2.2.4 Large-Scale Models and Multiscale Approaches

Efforts have focused on constructing large-scale models of V1 that incorporate detailed anatomical and physiological data and advances in neuroimaging techniques and the availability of multimodal data have facilitated the development of multiscale models that integrate information across different levels of brain organization^[133-134]. These models combine data from molecular, cellular, and systems levels to provide a more comprehensive understanding of V1's structure-function relationships.

The Blue Brain Project, for instance, concentrates its efforts on studying the neo-

cortex, the brain area associated with advanced cognitive capabilities, aiming to unravel the intricate neural circuitry underlying these functions^[135]. Through a combination of experimental data, neuroinformatics, and computational modeling, the project aims to reconstruct the intricate network of neurons and synapses in the neocortex. This detailed simulation allows researchers to explore the emergent properties of the brain, investigate how information is processed, and gain insights into neurological disorders. Markram et al.^[8] showcase the pioneering work of the Blue Brain Project in their paper, which details the remarkable endeavors undertaken to reconstruct and simulate the intricate microcircuitry of the neocortex. The authors describe their multiscale approach, combining experimental data from gene expression, cellular morphology, and electrophysiology to create a detailed model of the neocortical circuitry. Through a combination of data-driven reconstruction and computational simulations, they successfully reproduce the emergent properties of the neocortex at various scales. The paper highlights the complexity of the neocortical microcircuitry, showcasing the intricate network of neurons and synapses involved in information processing. This research signifies a major leap forward in comprehending the organization and operation of the neocortex, establishing a foundation for deeper explorations into brain function, neurological conditions, and the creation of technologies inspired by the brain's architecture.

Another work by Ritter et al.^[133] integrates computational modeling and multimodal neuroimaging to simulate and infer neurophysiological mechanisms underlying macroscopic brain signals, enabling the prediction of unobservable states and parameters, facilitating the formulation of new hypotheses about brain function and connectivity.

The development of large-scale and multiscale models of V1 provides a more comprehensive understanding of its structure-function relationships. These advancements integrate data across different levels of brain organization, unveiling the emergent properties of the brain and paving the way into complex brain functions.

1.2.2.5 Recent Large-Scale Data-Driven Models

Billeh et al. in 2020^[76,136] presents a comprehensive approach to developing multiscale models of the V1 in mice. The authors integrate structural and functional data from experimental studies to build a detailed and biologically realistic model of V1. They incorporate information about the connectivity patterns, neural morphology, and electrophysiological properties of individual neurons within the model. By systematically integrating data from different spatial and temporal scales, the authors create a multi-scale

model that captures the hierarchical organization and functional properties of V1. The resulting model provides insights into the complex dynamics and information processing capabilities of the visual cortex. This work represents a significant step towards understanding the principles underlying V1 function and lays the foundation for future studies on the neural mechanisms of visual perception.

Building upon the aforementioned research and employing a specialized variant of the GLIF neuron model^[84], Chen, Scherr, and Maass’s paper^[77] introduces a comprehensive computational model of V1 that incorporates data derived from the mammalian visual system. The authors developed a model that captures the robust and versatile visual processing capabilities observed in the brain. The model incorporates a hierarchical structure and simulates the activity of millions of neurons, capturing the receptive field properties and connectivity patterns observed in V1. Through extensive simulations, the authors demonstrate that their model exhibits brain-like properties, including robustness to noise, tolerance to occlusions, and the ability to generalize across different visual tasks. This work represents a significant step towards building computational models that closely resemble the functioning of the V1, and it has implications for understanding visual processing and developing more brain-inspired artificial vision systems.

These works showcase the progress in developing biologically realistic and large-scale models of V1. By integrating structural and functional data across different scales, these models capture the hierarchical organization and complex dynamics of the visual cortex. The incorporation of brain-like properties, such as representation for robustness and generalization abilities, brings us closer to understanding the principles underlying visual systems.

1.3 Main Work

1.3.1 Research Motivation

The V1 is a critical component of the visual system, responsible for processing and interpreting complex visual information. Developing accurate computational models of V1 is essential for understanding how visual perception and cognition emerge from underlying neural dynamics. While spiking models can capture the detailed temporal dynamics of individual neurons, their computational complexity limits their applicability to large-scale simulations and integration with standard learning algorithms. In aforementioned recent work of modeling of V1, a great number of computation resources are required for

simulating the spiking models. In contrast, firing-rate models provide a more tractable and efficient framework for modeling V1 function. By abstracting neuronal activity into average firing rates or other continuous variables, these models facilitate the simulation of larger neural networks and enable easier integration with machine learning techniques. However, existing firing-rate models of V1 often lack biological realism or they do not incorporate detailed connectivity patterns and cell-type specific properties derived from experimental data.

The main motivation of this research is to advance our understanding of V1 function through the development of a biologically-constrained, data-driven firing-rate model that balances biological realism with computational efficiency and learning capabilities. By developing an accurate computational model of V1, we can gain deeper insights into how visual perception and cognition emerge from the underlying neural dynamics. A firing-rate model that incorporates detailed biological constraints and connectivity patterns derived from experimental data can provide a more realistic representation of V1 function compared to existing models.

Another key motivation is to bridge the gap between experimental data and computational models. Recent advances in experimental techniques have provided a wealth of data on the connectivity patterns and cell-type specific properties of V1 neurons. However, many existing computational models of V1 do not fully incorporate this detailed experimental data. The motivation is to develop a data-driven firing-rate model that integrates these experimental findings, providing a more accurate representation of V1 function and enabling the validation of computational predictions against empirical observations.

Furthermore, developing a biologically-constrained firing-rate model of V1 can provide a foundation for studying higher-level visual processing. The V1 serves as the first stage of cortical visual processing, providing inputs to higher-level visual areas. By developing an accurate firing-rate model of V1, we can establish a solid foundation for modeling and understanding the function of downstream visual areas. This can facilitate the investigation of how higher-level visual features and object representations emerge from the hierarchical processing of visual information.

Moreover, firing-rate models help to explore the potential of bridging AI and neuroscience by leveraging insights from biological neural networks to inform the development of novel learning rules and hybrid learning approaches. By incorporating biologically-inspired learning mechanisms, such as Hebbian learning, into the firing-rate model, we

can investigate how these learning rules contribute to the emergence of robust and efficient visual representations. Additionally, by combining these biologically-inspired learning rules with traditional machine learning techniques, such as backpropagation, we can develop hybrid learning approaches that leverage the strengths of both biological and artificial neural networks. This interdisciplinary approach can lead to the development of more powerful and biologically-plausible AI systems for visual processing tasks.

Also, accurate computational models of V1 can inform the design of neural prosthetics and brain-machine interfaces for restoring or augmenting visual function in individuals with visual impairments. A biologically-constrained firing-rate model can provide insights into the neural coding principles and connectivity patterns necessary for effectively stimulating or decoding visual information in the brain.

Overall, the main motivation for developing a biologically-constrained, data-driven firing-rate model of V1 is to advance our understanding of visual processing, bridge the gap between experimental data and computational models, and provide a foundation for studying higher-level visual function and developing neural applications, etc. By leveraging the computational efficiency of firing-rate models while incorporating detailed biological constraints, this research aims to create a powerful tool for investigating the neural basis of visual perception and cognition.

1.3.2 Focus and Highlights

This thesis focuses on developing a computationally efficient and biologically plausible firing-rate neural network model for the V1 and investigating its learning capabilities. The main objectives and highlights of the work include:

1. Transitioning from spiking models to firing-rate models: The thesis discusses the motivation behind the shift from computationally intensive spiking models to more efficient firing-rate models. It develops discrete form firing-rate models that enable the study of biologically plausible visual processing in V1.

2. Utilizing real data for model development: The V1 model is constructed using data from the Allen Institute for Brain Science, ensuring a biologically realistic representation of neuronal distributions and connectivity. This data-driven approach enhances the model's biological plausibility.

3. Evaluating learning capabilities through visual tasks and multiple scenarios and settings: The learning performance of the V1 model is assessed using three carefully designed visual tasks: fine-orientation detection, image classification, and visual change

detection. Various learning scenarios and settings are explored to comprehensively analyze the model's capabilities under different conditions.

4. Incorporating LGN filters: The thesis investigates the impact of preprocessing the visual input using biologically-inspired LGN filters. The incorporation of these filters aims to enhance the model's ability to capture and utilize temporal information, which is crucial for visual processing.

5. Analyzing the impact of network size and architecture: The thesis compares the performance of models with different network sizes (3k and 6k neurons) and examines the role of network architecture in learning effectiveness. This analysis provides insights into the relationship between network complexity and model performance.

6. Analyzing synaptic weight distributions and neuronal responses: The thesis examines the distribution of synaptic weights after training under different scenarios and learning approaches. It also analyzes the internal representations and activation patterns of neurons in response to external stimuli and noise. These analyses provide insights into the underlying mechanisms of learning and information processing in the V1 model.

7. Investigating the robustness to internal and external noise: The thesis explores the model's resilience to internal noise, such as slow and quick noise, and external noise, such as noisy input images. The analysis of the model's performance under noisy conditions provides insights into its robustness and the role of LGN filters and network size in mitigating the impact of noise.

8. Exploring hybrid learning approaches: The thesis investigates the potential of hybrid learning approaches that combine different learning mechanisms. By designing and evaluating hybrid learning strategies, the work aims to identify effective methods for enhancing the model's learning capabilities and convergence properties, or potential challenges.

9. Exploring biological realism: The thesis tries comparing the model's properties and behaviors with some characteristics of biological neural systems. This includes examining the distribution of synaptic weights, the power law distribution of processed neuronal responses, and other potential properties. By drawing parallels with biological observations, the work wants to evaluate the model's biological plausibility and identify areas for further refinement.

By focusing on these key aspects, the thesis aims to contribute to the understanding of neural processes underlying visual perception and provide valuable insights into the

CHAPTER 1 INTRODUCTION

computational aspects of neural modeling. The developed firing-rate model of V1 serves as a foundation for further investigations into biologically plausible visual processing, learning, and robustness in neural networks.

CHAPTER 2 DATA-BASED FIRING-RATE V1 MODEL

2.1 From Spiking Models to Firing-Rate Models

A firing-rate model is fundamentally a continuous-variable model. In this context, the term “rate” can extend beyond merely quantifying frequency in Hz (cycles per second), although people try to link it to the real firing rates; it represents a continuous measure of neural activity over time. This abstraction in firing-rate models encapsulates the averaged output of a neuron’s spiking activity, without necessarily detailing the exact spikes at each time points.

In many rate models, the “rate” typically reflects the average activity level of a neuron over a predetermined time window. This measure may correlate with, but is not strictly equivalent to, the actual firing frequency. Crucially, the model employs a continuous variable to depict neural activity. This continuous representation serves as an input to other neurons or as a mechanism to interpret the computational outcomes of the network. The inherent continuity of these models allows for the application of linear and nonlinear systems theory, thereby enhancing their analytical accessibility and applicability across a range of cognitive and computational tasks. In contrast, spiking models represent discrete-variable models, which focus on individual spikes as the fundamental unit of neural representation.

The validity of a neuronal firing-rate model crucially depends on how accurately the average firing frequency of neurons, computed across multiple trials, captures the influence that the actual spike sequences exert on the dynamic behavior of the neural network. In other words, the firing-rate model’s credibility hinges on whether the mean firing rates serve as a reliable proxy for the real-world effects of the precise timing patterns of neuronal spike trains on the network’s temporal dynamics. The supposition that the neural response function can be substituted by the corresponding firing rate generally relies on the observation that each neuron in the network receives a multitude of inputs, thereby rendering this approximation viable^[3].

Researchers usually measure firing rate response curves by directly injecting current into the cell body of a neuron. This methodology enables the researchers to define the input-output relationship of the neuron, specifically the correlation between the input current and the resulting firing rate. The essential step in creating a firing-rate model is to

substitute the neural response function with the neuron's firing rate. This important replacement, as the key step in formulating firing-rate models, simplifies the intricate temporal dynamics of individual spikes into a more manageable representation of the neuron's activity^[3].

Now, we demonstrate the derivation of a firing-rate model from a spiking model, following the approaches outlined by Dayan and Abbott in their seminal work "Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems"^[3].

To begin with, the synaptic current of a postsynaptic neuron can be determined by aggregating the individual currents elicited by the activity of presynaptic neurons.

$$I_s = \sum_{b=1}^{N_{\text{pre}}} w_b \int_{-\infty}^t d\tau K_s(t - \tau) \rho_b(\tau) \quad (2.1)$$

where w_b is the weight and $\rho_b(\tau) = \sum_i \delta(\tau - t_i)$ is a spike train, both of presynaptic neuron b ; $K_s(t) = \frac{e^{-t/\tau_s}}{\tau_s}$ is the normalized synaptic current kernel, whose integral from 0 to infinity is 1; N_{pre} is the number of presynaptic neurons. Note that here it is assumed postsynaptic currents from same synapse's multiple spikes can add up linearly, corresponding to the integral sign, and postsynaptic currents from different presynapses can also add up linearly, corresponding to the summation sign.

By replacing $\rho_b(\tau)$ by the rate of presynaptic neurons r_b , then we can introduce the rate into the calculation.

$$I_s = \sum_{b=1}^{N_{\text{pre}}} w_b \int_{-\infty}^t d\tau K_s(t - \tau) r_b(\tau) \quad (2.2)$$

where $\rho_b(\tau)$ is directly replaced by $r_b(\tau)$ without any other changes because we can conceptualize the firing rate $r_b(\tau)$ as representing the probability of a spike occurring. While $\rho_b(\tau)$ represents the actual firing events, there exists an underlying consistency between these two quantities. In other words, we can consider the firing rate as an accurate reflection of the likelihood of observing spikes.

Due to the exponential nature of the current kernel $K_s(t) = \frac{e^{-t/\tau_s}}{\tau_s}$, we can derive a particularly convenient form for the derivative of Equation (2.2). This advantageous property simplifies the mathematical manipulations involved in analyzing the dynamics of the postsynaptic current.

$$\tau_s \frac{dI_s}{dt} = -I_s + \sum_{b=1}^{N_{\text{pre}}} w_b r_b = -I_s + \mathbf{w} \cdot \mathbf{r} \quad (2.3)$$

where \mathbf{w} and \mathbf{r} are vectors of weights and rates, respectively. Note that if the kernel $K_s(t)$

is in other forms like bi-exponential terms, then we cannot have the above differential equation but a more complicated result.

Furthermore, it is assumed that the steady-state firing rate and synaptic current can be related through an activation function, denoted as $F(\cdot)$. This function maps the synaptic current to the corresponding firing rate of the postsynaptic neuron, providing a concise mathematical representation of the neuron's input-output relationship:

$$r = F(I_s) \quad (2.4)$$

The activation function $F(\cdot)$ can assume different forms, including a sigmoid function or a rectified linear unit (ReLU), contingent upon the particular attributes of the neuron being modeled. The choice of the activation function is crucial in capturing the nonlinear dynamics of neuronal firing and plays a significant role in shaping the behavior of the firing-rate model.

Nevertheless, the activation function is constrained to operate within steady-state conditions. When the firing rate and synaptic current are varying over time, a time constant τ_r should be introduced to capture the temporal delay between changes in the current and the corresponding response in the firing rate. This time constant represents the time scale over which the firing rate adapts to variations in the synaptic input.

To comprehensively describe the relationship between the firing rate and synaptic current of a neuron, we can formulate three different versions of the governing equations.

Instantaneous version: This version directly relates the firing rate to the synaptic current using the activation function, assuming an instantaneous response, i.e., $\tau_r \ll \tau_s$. This means the change of firing rate follows the change of current instantaneously without any significant delay.

$$\begin{aligned} \tau_s \frac{dI_s}{dt} &= -I_s + \mathbf{w} \cdot \mathbf{r} \\ r(t) &= F(I_s(t)) \end{aligned} \quad (2.5)$$

Low-pass filtered version: This version incorporates the time constant τ_r to account for the temporal delay in the firing rate response, which cannot be ignored, i.e., $\tau_r \approx \tau_s$. This means the change of firing rate and the change of current need some time to be synchronized so that the high frequency components will be filtered when the change of

current goes to the rate and vice versa.

$$\begin{aligned}\tau_s \frac{dI_s}{dt} &= -I_s + \mathbf{w} \cdot \mathbf{r} \\ \tau_r \frac{dr(t)}{dt} &= -r(t) + F(I_s(t))\end{aligned}\tag{2.6}$$

Current-equilibrium low-pass filtered version: This version combines the dynamics of both the synaptic current and the firing rate, providing a complete description of the neuron's behavior, i.e., $\tau_r \gg \tau_s$. This means the change of current follows the change of firing rate instantaneously without any significant delay.

$$\tau_r \frac{dr(t)}{dt} = -r(t) + F(\mathbf{w} \cdot \mathbf{r})\tag{2.7}$$

These three variations provide an all-encompassing foundation for simulating the interaction between synaptic currents and firing rates in neuronal dynamics, considering the temporal facets of neuronal response characteristics. The selection of the appropriate version hinges on the comparative magnitudes of the time constants τ_r and τ_s . Note that both the instantaneous version and the current-equilibrium low-pass filtered version are reduced forms or special cases of the more general low-pass filtered version. The reduced two versions are used for simplifying the computation.

Note that although firing-rate models are not as biologically realistic as SNNs, they can be derived approximately from the underlying neural dynamics^[3]. Incorporating parameters from real biological data, such as network connectivity and time constants, is crucial for constructing firing-rate neural networks. By utilizing these data to inform the design and parameterization of the firing-rate models, we can create networks with key biological constraints, benefiting from the computational efficiency. This approach allows us to get a balance between biological plausibility and tractability.

2.2 Discrete Form Firing-Rate Model

To emulate a firing-rate network, one must convert the continuous-time differential equations describing the firing-rate dynamics into discrete-time difference equations. This conversion is essential for training and inference in standard machine learning frameworks and computer architectures.

Discretization entails approximating the continuous-time derivatives using finite differences across small time intervals. This allows for the iterative updating of the neural network's state at each time step, effectively capturing the temporal evolution of the fir-

ing rates. By doing so, learning algorithms like backpropagation through time (BPTT) for recurrent neural networks (RNNs) can be employed, and hardware accelerators such as GPUs can be leveraged.

To obtain more accurate difference equations compared to the Euler method, which can cause large errors as time evolves during simulations, we can employ the first-order exponential integrator method^[137-138]. This method provides a better approximation for the continuous-time dynamics in a certain differential equation form.

By applying the exponential integrator method to the three versions of the firing-rate differential equations given in last section, we arrive at the following set of difference equations:

Instantaneous version ($\tau_r \ll \tau_s$):

$$\begin{aligned} I_s(t + \Delta t) &= e^{-\frac{1}{\tau_s} \Delta t} I_s(t) + \left(1 - e^{-\frac{1}{\tau_s} \Delta t}\right) \mathbf{w} \cdot \mathbf{r}(t) \\ r(t + \Delta t) &= F(I_s(t + \Delta t)) \end{aligned} \quad (2.8)$$

Low-pass filtered version ($\tau_r \approx \tau_s$):

$$\begin{aligned} I_s(t + \Delta t) &= e^{-\frac{1}{\tau_s} \Delta t} I_s(t) + \left(1 - e^{-\frac{1}{\tau_s} \Delta t}\right) \mathbf{w} \cdot \mathbf{r}(t) \\ r(t + \Delta t) &= e^{-\frac{1}{\tau_r} \Delta t} r(t) + \left(1 - e^{-\frac{1}{\tau_r} \Delta t}\right) F(I_s(t)) \end{aligned} \quad (2.9)$$

Current-equilibrium low-pass filtered version ($\tau_r \gg \tau_s$):

$$r(t + \Delta t) = e^{-\frac{1}{\tau_r} \Delta t} r(t) + \left(1 - e^{-\frac{1}{\tau_r} \Delta t}\right) F(\mathbf{w} \cdot \mathbf{r}(t)) \quad (2.10)$$

In the above difference equations, Δt represents the discrete time step size (e.g., 1 ms), and the exponential terms capture the decay dynamics of the synaptic current and firing rate over time. These difference equations provide a more accurate approximation of the continuous-time dynamics compared to the Euler method, with which no exponential terms will be remained and the dynamics will be destroyed more. They take into account the exponential nature of the firing-rate equations and provide a stable and efficient way to simulate the temporal evolution of the network. See Appendix B for the method to derive the above difference equations.

By using the network connection information to get the upstream neurons of all neurons and compiling the difference equations for each neuron in the network together, we can create a comprehensive mathematical framework that describes the dynamics of the entire system. This framework serves as the foundation for simulating the network's be-

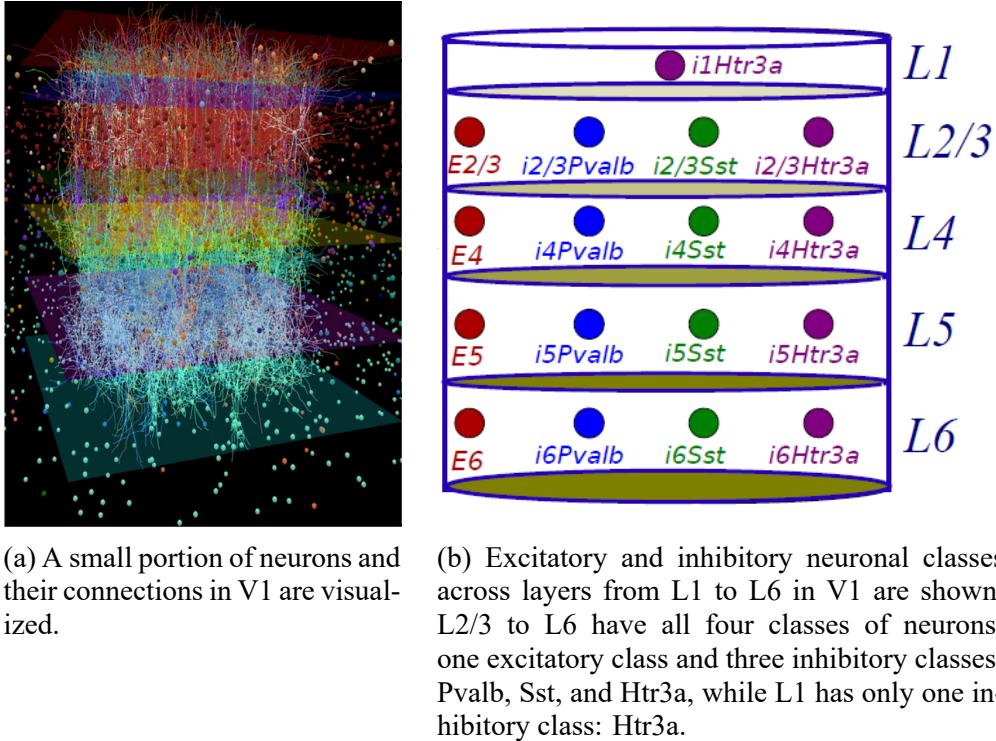
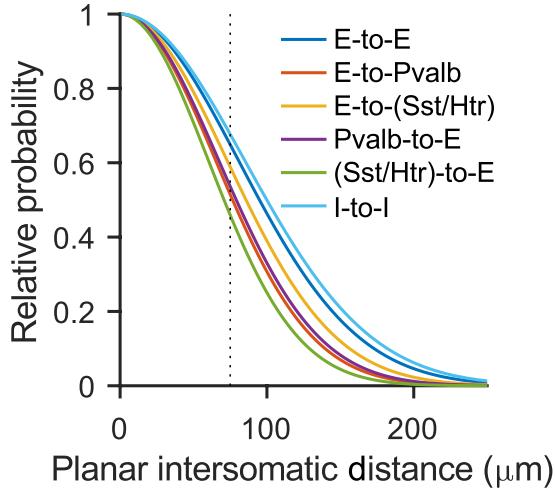


Figure 2.1 V1 neural network visualization and cell classes^[76]

havior and applying learning algorithms to optimize the synaptic weights for specific tasks under different scenarios and settings.

The firing-rate neural network model described above is computationally more efficient compared to a spiking network model. Take a GLIF model-based network model^[77] as the example for comparison (see Equations (1) to (4) in the reference^[77]). Our firing-rate model, except for the weights, uses only two variables for each neuron: the firing rate $r(t)$ and the synaptic current $I_s(t)$. In contrast, the GLIF model requires six variables per neuron: a postsynaptic current, two after-spike currents, a membrane potential, a binary spiking variable, and an auxiliary variable for calculating the postsynaptic current. By reducing the number of variables per neuron, the firing-rate model significantly reduces the computational burden and memory requirements. Furthermore, the absence of discrete variables in the firing-rate model makes it more computationally friendly. Note that the external current is not considered in this comparison, as it can be regarded as a background input for both models.



The relative connection probability between pairs of neuron classes is shown as a function of the lateral distance separating them. The y-axis represents the relative probability of a synaptic connection forming between two neurons. The x-axis denotes the lateral distance between the two neurons. The vertical dashed line is drawn at 75 μm as a reference distance.

Figure 2.2 Fitted relative connection probability of each pair of neuron classes versus interneuron lateral distance^[76-77]

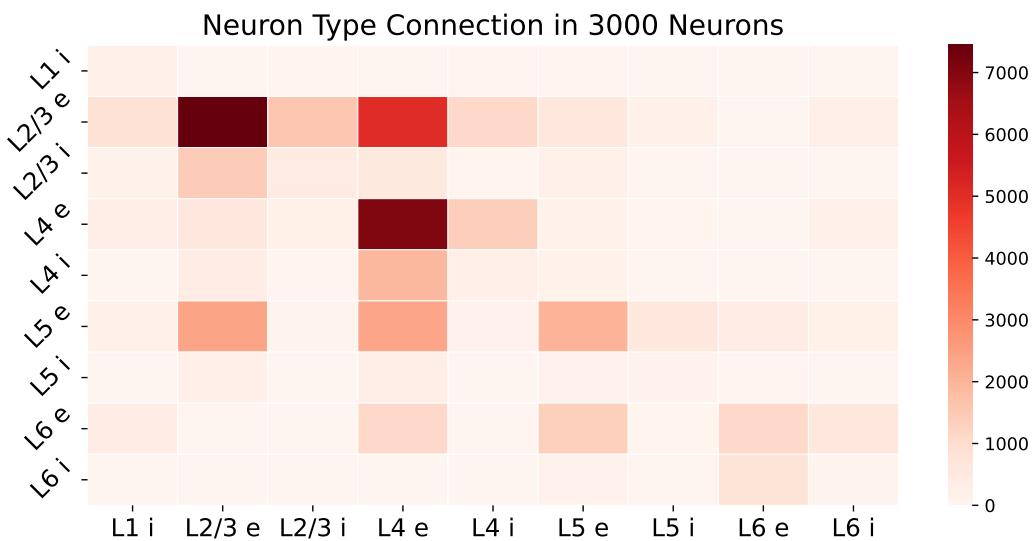
2.3 Data Description and Experiment Settings

We construct our firing-rate models using data from the Allen Institute for Brain Science^[76,136]. The dataset contains a total of 230,924 neurons and 70,139,111 synapses. The dataset includes a total of 111 types of neuron models, which are categorized into 17 classes, illustrated in Figure 2.1. For the 17 different types of neurons shown in the figure, they can have different properties, such as targeting area in a neuron which can be the soma or dendrites. But from the perspective of a point neuron model, we focus on the connection probability and the synaptic strengths, as shown in Figure 4 in the work of Billeh et al.^[76], and the time constants which are determined by the type of connected two neurons^[76,136]. The fitted curves of relative connection probability with respect to lateral distance between neuron classes are shown in Figure 2.2.

Due to computational resource limitations, we build our models using either 3,000 or 6,000 neurons from the dataset. The dataset provides the coordinates (x, y, z) for all neurons. To select a specific number of neurons (e.g., 3,000), we identify the smallest cylinder at the center of the cortex patch that encompasses exactly 3,000 neurons. We then retain the connections between these neurons and remove any connections from neurons outside the cylinder to create our network. The resulting network for 3,000 neurons contains 48,705 synapses, while the network for 6,000 neurons has 198,174 synapses. On average, each neuron in the network with 3,000 neurons has approximately 16 synapses,

Table 2.1 Distribution of excitatory and inhibitory neurons in network layers (3k neurons)

Neuron class	Neuron number
L1 i	59
L2/3 e	753
L2/3 i	129
L4 e	593
L4 i	94
L5 e	455
L5 i	64
L6 e	725
L6 i	128

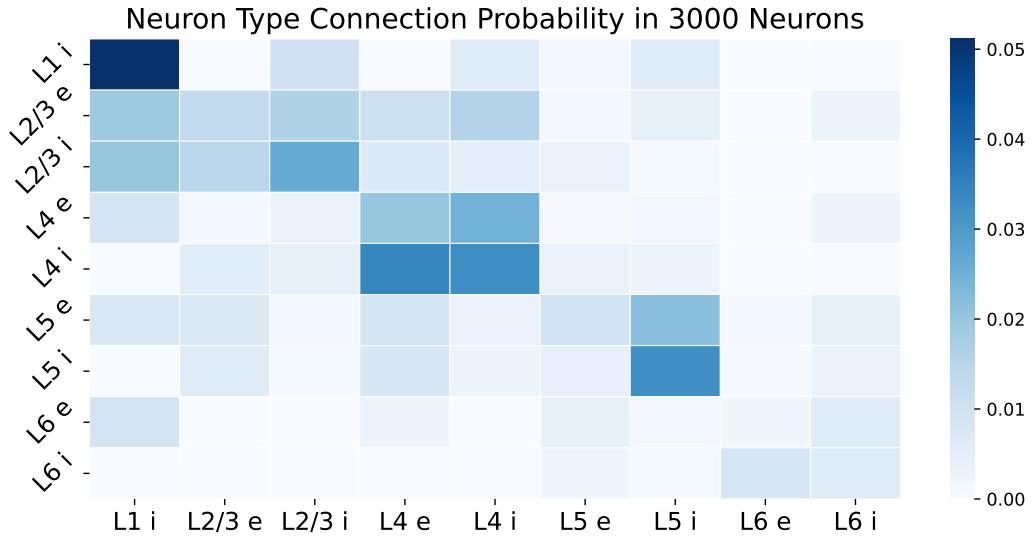


This 9×9 colored table visualizes the number of synapses between pairs of source and target neuron classes. The horizontal axis (x-axis) represents the source neuron classes, while the vertical axis (y-axis) denotes the target neuron classes.

Figure 2.3 Synaptic connectivity matrix for neuronal classes (3k neurons)

whereas each neuron in the network with 6,000 neurons has approximately 33 synapses. Notably, when the number of neurons is doubled, the number of synapses increases by a factor of approximately four, indicating that the connections between neurons have quadrupled as the network size doubled.

The neuron and synapse distributions for the 3,000-neuron and 6,000-neuron networks can be found in Table 2.1, Figure 2.3, Figure 2.4, Table 2.2, Figure 2.5, and Figure 2.6. Basically, people think the connections between layers facilitate a robust forward flow of visual data from L4 to L2/3 and then to L5 and L6, supplemented by numerous



The layout and labeling are same as the description in Figure 2.3.

Figure 2.4 Synaptic connectivity probability matrix for neuronal classes (3k neurons)

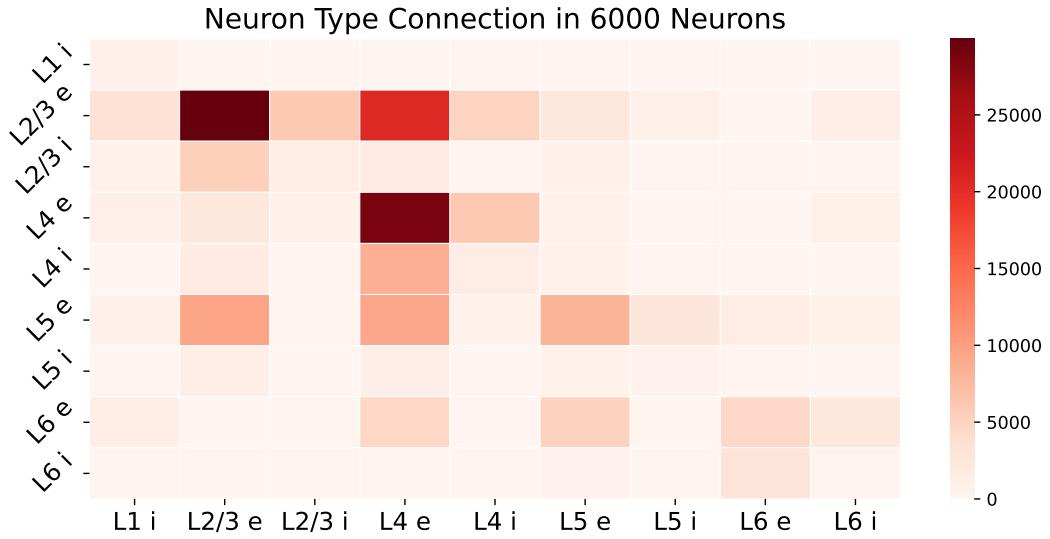
Table 2.2 Distribution of excitatory and inhibitory neurons in network layers (6k neurons)

Neuron class	Neuron number
L1 i	113
L2/3 e	1508
L2/3 i	250
L4 e	1189
L4 i	213
L5 e	907
L5 i	133
L6 e	1425
L6 i	262

recurrent pathways^[77].

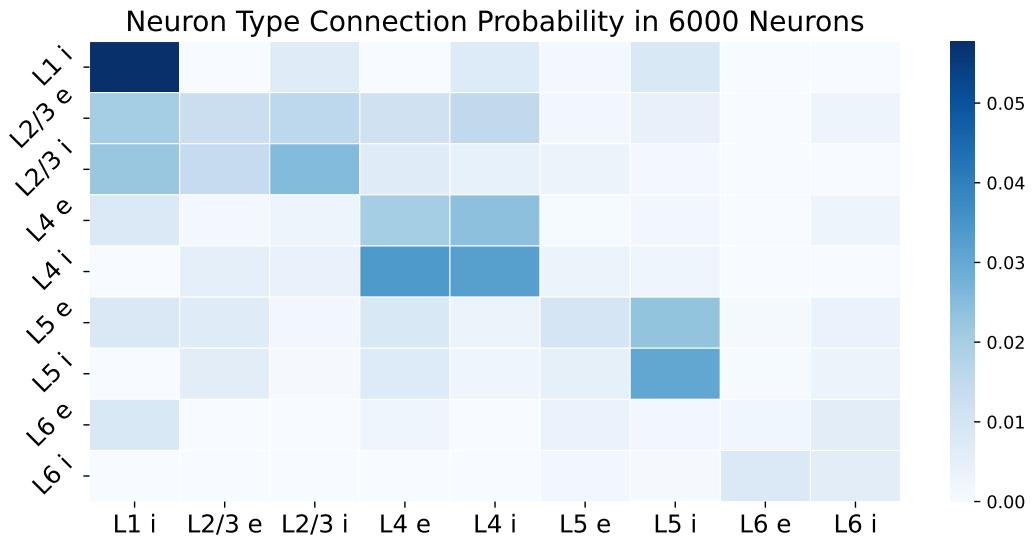
For the inputs to the network, half of them are delivered to L4 and the rest are randomly delivered to other layers. The outputs are extracted from L5 by default.

Regarding neuron properties, the firing-rate model is a simplified version of the spiking model, and as such, we only require two key parameters: the synaptic time constant (τ_s) and the firing rate time constant (τ_r). It is important to note that while the Allen Institute dataset provides individual synaptic time constants for each synapse, our firing-rate model simplifies this parameter to be neuron-specific. To achieve this, we calculate the neuron's τ_s as the weighted average of the τ_s values of its immediate upstream synapses.



The layout and labeling are same as the description in Figure 2.3.

Figure 2.5 Synaptic connectivity matrix for neuronal classes (6k neurons)



The layout and labeling are same as the description in Figure 2.3.

Figure 2.6 Synaptic connectivity probability matrix for neuronal classes (6k neurons)

However, the dataset does not include information about τ_r , which is not a typical property of a neuron. To address this, we sample the τ_r parameter based on an assumed distribution, using τ_s as the mean, because τ_s and τ_r are most likely to be close values. The sampling could introduce the diversity for τ_r . The relative magnitude between τ_s and τ_r determines which version of the firing-rate difference equations derived in previous sections we employ. Alternatively, we can opt to use the most general version, which is the low-pass filtered version, to account for the dynamics in all situations. The activation

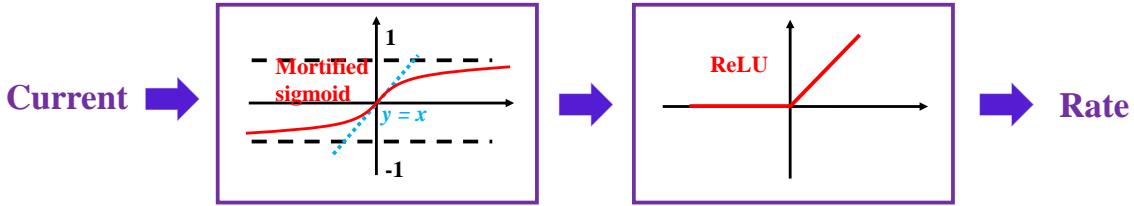


Figure 2.7 The activation function for transformation from currents to rates

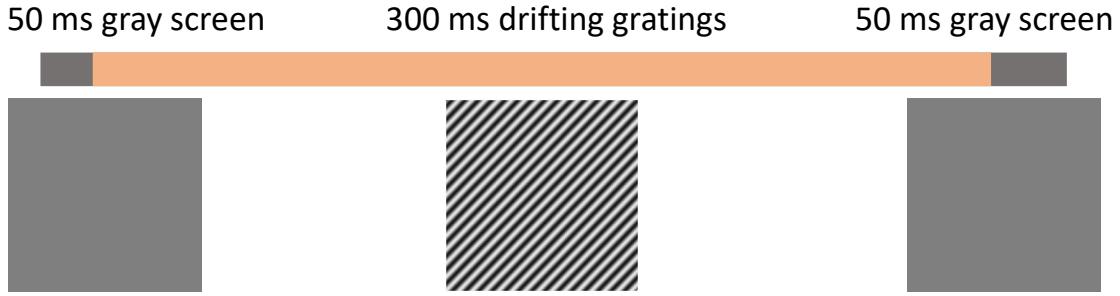
function we employed to map the synaptic current to the corresponding firing rate is a composite function comprising a modified sigmoid function, where the values along the x -axis are scaled such that the derivative at $x = 0$ is 1 and the y range of the function is linearly transformed to $[-1, 1]$, same as $\tanh(\cdot)$ which is often considered as an alternative activation function for rate models, from $[0, 1]$, combined with a ReLU after it to ensure the rate is always non-negative. The schematic diagram of the activation process is given in Figure 2.7. Note that if considering the rate as a generalized concept of a continuous variable, negative rates can be acceptable and no ReLU is needed.

Recall that for each neuron in the network Equations 2.8, 2.9 or 2.10 will be applied to describe the dynamics. Which to use depends on the relative magnitude between τ_s and τ_r , as mentioned in the 2.2. For variables in the equations, whether a weight between two neurons can be non-zero, i.e., the weight number, depends on the connection matrix, which is extracted from the dataset, because we want to use the real biological brain connections; τ_s is calculated based on data from the dataset and τ_r is sampled based on a distribution with τ_s being the mean. The weights w for each neuron need to be learned.

2.4 Visual Tasks for Learning

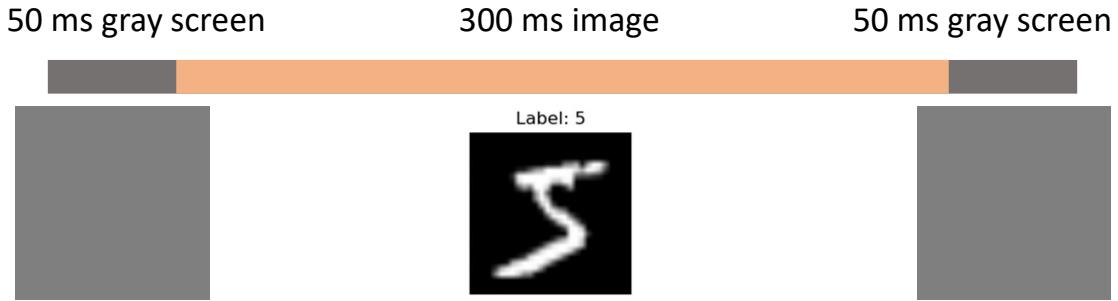
We designed three visual tasks to evaluate the learning capabilities of the model: the fine-orientation detection task^[139-140], the image classification task, and the visual change detection task^[141-142]. Each task evaluates distinct facets of the model's capabilities. The fine-orientation discrimination and visual change detection tasks are commonly employed in mouse experimental studies with established paradigms, and the image classification task is adapted from well-known computer vision benchmarks.

1. **Fine-orientation detection task** (see Figure 2.8): This task assesses the model's capacity to discern nuanced variations within a movie. The model is presented with a sequence of frames containing oriented gratings, and it must accurately identify the orientation of each grating. The task requires the model to discriminate between finely varied



The temporal sequence of stimuli initiates with a neutral gray screen displayed for 50 ms, providing a baseline visual field. This is followed by the core stimulus phase, consisting of drifting gratings presented for 300 ms. These gratings are sinusoidal waves characterized by a temporal frequency of 2 Hz and a spatial frequency of 0.05 cycles per degree. The contrast of the gratings varies randomly between 0.8 and 1, ensuring a range of visual intensity. Each grating's initial phase is randomized, starting anywhere within a full 0 to 360-degree cycle. Crucially, the orientation of the gratings is confined between 43 and 47 degrees, excluding the exact median of 45 degrees, with precision up to 0.2 degrees (e.g., 43.2, 47.0 degrees but not 45 degrees). Following the drifting gratings, another 50 ms gray screen period concludes the sequence, signalling the end of the stimulus presentation. At the end of the trial, the model is tasked with predicting whether the orientation of the gratings exceeded 45 degrees.

Figure 2.8 Temporal sequence of visual stimuli for the fine-orientation detection task

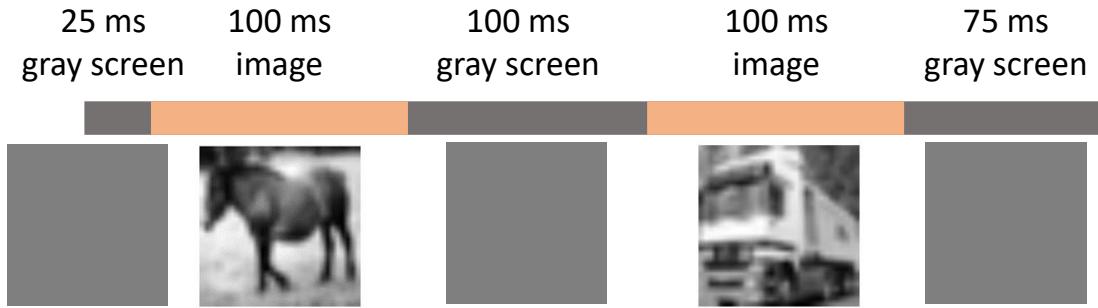


The temporal sequence of stimuli initiates with a neutral gray screen displayed for 50 ms, providing a baseline visual field. This is followed by the core stimulus phase which is an image from the MNIST dataset, containing a handwritten digit ranging from 0 to 9, presented for 300 ms. Following the image presentation, another gray screen is shown for 50 ms. At the end of the trial, the model is required to perform a 10-class classification task, predicting the digit displayed in the movie.

Figure 2.9 Temporal sequence of visual stimuli for the image classification task

orientations, testing its sensitivity to detailed visual information. The chance level accuracy for this task is 50%.

2. Image classification task (see Figure 2.9): In this task, the model must precisely categorize images into various classes. The model is presented with a diverse set of images, each belonging to a specific class, and it must assign the correct label to each image. This task assesses the model's ability to learn and recognize complex visual patterns and to generalize its knowledge across different instances within each class. The chance level



The temporal sequence of stimuli initiates with a neutral gray screen displayed for 25 ms, providing a baseline visual field. This is followed by the core stimulus phase which is an image from the CIFAR-10 dataset, which consists of 10 different classes, is presented for 100 ms. A gray screen is then shown for 100 ms. A second image from the same dataset is shown for 100 ms. This image has a 50% chance of being the same as the first image and a 50% chance of being a different image from a different class. The trial ends with a 75 ms presentation of a gray screen. Upon trial completion, the model must decide if the second image differs from the initial one. The longer duration of the final gray screen (75 ms) compared to the initial gray screen (25 ms) is designed to test the memory ability of the model. For this task, images from the CIFAR-10 dataset are converted to grayscale.

Figure 2.10 Temporal sequence of visual stimuli for the visual change task

accuracy for this task is 10%.

3. Visual change detection task (see Figure 2.10): The visual change task challenges the model’s capacity to remember and integrate previous information. The model is presented with a sequence of images, and it must detect when a change occurs in the visual input. This task requires the model to maintain a representation of the previous images and compare it with the current input to identify any changes. It assesses the model’s capability to encode and recall visual data over time. The chance level accuracy for this task is 50%.

By evaluating the model’s performance on these tasks, we can gain insights into its ability to learn and adapt to different visual challenges.

It’s important to note that we regulated the duration of the visual stimuli across the three tasks due to the limited GPU memory available for model training. Our training dataset comprised 18,000 samples, with 6,000 samples per task. The test dataset consisted of 1,800 samples, with 600 samples per task. The temporal resolution was set to 1 ms. The computational resources employed were NVIDIA GeForce RTX 4090 GPUs with 24 GB memory and NVIDIA A800 GPUs with 80 GB memory. For each experiment, we conducted training over 200 epochs.

The movie data comprises frames with a resolution of 200 by 200 pixels. At each time point, every frame, represented as a two-dimensional matrix, undergoes a transfor-

mation process to be pooled and reshaped into a one-dimensional array, which serves as the input to the computational model. The length of the one-dimensional array is 832 in our experiments, which is equal to the LGN cell number in chapter 3. So, the input to the V1 model is 832 time sequences, which are fed to 832 input neurons. See Figure 3.3 for comparison with the data preprocessing with LGN in chapter 3.

The fine-orientation detection and visual change detection tasks are 2-class classification tasks, and the image classification task is a 10-class classification task. To generate each class output, we aggregate data from a group of 20 neurons. So we have in total 280 readout neurons, which by default are randomly selected from L5, to output the predictions. The cross-entropy loss is employed for all three tasks to direct the backpropagation process.

2.5 Comparative Scenarios and Settings

For the performance evaluation of our model, we established four distinct scenarios: the default scenario, the “uniform neurons” scenario, the “random synapses” scenario, and the “no Dale’s law” scenario.

Default scenario: in the default scenario, the following conditions are met. (1) Neurons exhibit diversity; specifically, the synaptic time constant (τ_s) is determined by connections from a neuron’s immediate upstream neurons, and the firing rate time constant (τ_r) is sampled based on its τ_s . (2) Synaptic connections are informed by the dataset from the Allen Institute. (3) Dale’s law is applied, guaranteeing that neurons preserve their roles as either consistently excitatory or inhibitory. Altering any of the above three conditions—neuronal diversity, synaptic connection structure, or the enforcement of Dale’s law—result in the creation of the three alternative scenarios, as described below.

“Uniform neurons” scenario: this scenario reduces neuronal variability by making the parameters of all excitatory neurons identical to each other, and doing the same for all inhibitory neurons. This means that all excitatory neurons share the same values for the synaptic time constant (τ_s) and the firing rate time constant (τ_r), while all inhibitory neurons have their own set of uniform τ_s and τ_r values. All other aspects mirror the default scenario.

“Random synapses” scenario: In this scenario, our objective was to construct a connected graph. We initiated this process by organizing all n neurons into a foundational tree-like structure, achieved by forming $n - 1$ synaptic connections: the second neuron

links directly to the first, the third neuron establishes a connection with either the first or second neuron, the fourth neuron connects to one of the first three, and so on. This approach ensures basic connectivity. The total synapse count is kept consistent with the default scenario. Beyond the $n - 1$ synapses required for the tree structure, the remaining synaptic connections are generated entirely at random among all neurons. This design results in a network that is not purely random but is potentially well-connected, providing a balance between structure and randomness.

“No Dale’s law” scenario: In this scenario, the traditional constraints of the Dale’s law are deactivated. This allows synaptic weights to shift signs during the learning process, allowing neurons to deliver positive, negative or a mix of positive and negative weights to downstream neurons. It is crucial to emphasize, however, that despite these changes in synaptic signs, the intrinsic parameters of the neurons, their time constants, remain fixed throughout the learning phase, consistent with the settings in the other scenarios. That is to say, if an excitatory neuron becomes inhibitory during learning, its time constants are not allowed to be changed but keep the original constants of the excitatory parameters.

Moreover, to explore how various output layer configurations affect the model’s learning ability, we alter the output layer from the default L5 to L2/3, L4, and L6. L1 is excluded from consideration because it contains too few neurons. Additionally, we compare these settings to a random output setting.

“Output: L2/3” setting: the readout neurons are randomly selected from neurons in L2/3.

“Output: L4” setting: the readout neurons are randomly selected from neurons in L4.

“Output: L6” setting: the readout neurons are randomly selected from neurons in L6.

“Output: random” setting: the readout neurons are randomly selected from neurons in all layers.

Furthermore, to explore the potential of hybrid learning, which combines the backpropagation and Hebbian learning, we apply this approach to the visual tasks.

“Hybrid learning”: In this setting, synapses connecting neurons within the same layer are selected to undergo Hebbian learning, while the learning of the resting connections still use backpropagation. The Hebbian learning of a synapse is determined by the

product of the firing rates of the source and target neurons, as well as a learning rate. To maintain biological plausibility, the weights are constrained to maintain their sign during Hebbian learning. In other words, the Dale's law always holds in Hebbian learning. For these synapses adopting Hebbian learning, the backpropagation no longer updates their weights. In our experiments, this hybrid learning setting serves as an initial exploration into integrating biologically plausible learning mechanisms into firing-rate network models. The pseudo code of the hybrid learning is given in Algorithm 2.1. Investigating hybrid learning that incorporates Hebbian plasticity is meaningful for several reasons. First, Hebbian learning introduces a biologically plausible learning mechanism into the firing-rate network models. Studying how the integration of this unsupervised, local learning rule affects the network's learning capabilities can provide insights into the computational principles of biological neural systems. Furthermore, the hybrid approach allows us to explore the potential benefits and trade-offs of combining supervised backpropagation with unsupervised Hebbian learning.

Algorithm 2.1 Hybrid Learning in Firing-Rate Network Models

Input: Firing rates of neurons, initial synaptic weights, Hebbian learning rate, backpropagation learning rate

Output: Updated synaptic weights

Initialize synaptic weights

while not converged **do**

- Forward pass: Compute firing rates of neurons
- for** each synapse **do**

 - if** synapse connects neurons within the same layer **then**

 - Compute the product of source and target neuron firing rates:
 - $\Delta w = \eta_{\text{Heb}} \cdot r_i \cdot r_j$
 - Update the synaptic weight using the Hebbian learning rule:
 - $w_{ij} = w_{ij} + \Delta w$
 - Constrain the weight to maintain its sign (Dale's law):
 - $$w_{ij} = \begin{cases} \max(0, w_{ij}), & \text{if } w_{ij} \geq 0 \\ \min(0, w_{ij}), & \text{if } w_{ij} < 0 \end{cases}$$

 - else**

 - Backward pass: Get the gradient for the synapse using backpropagation
 - Update the synaptic weight using gradient descent G :
 - $w_{ij} = w_{ij} - \eta_{\text{BP}} * G$

 - end if**

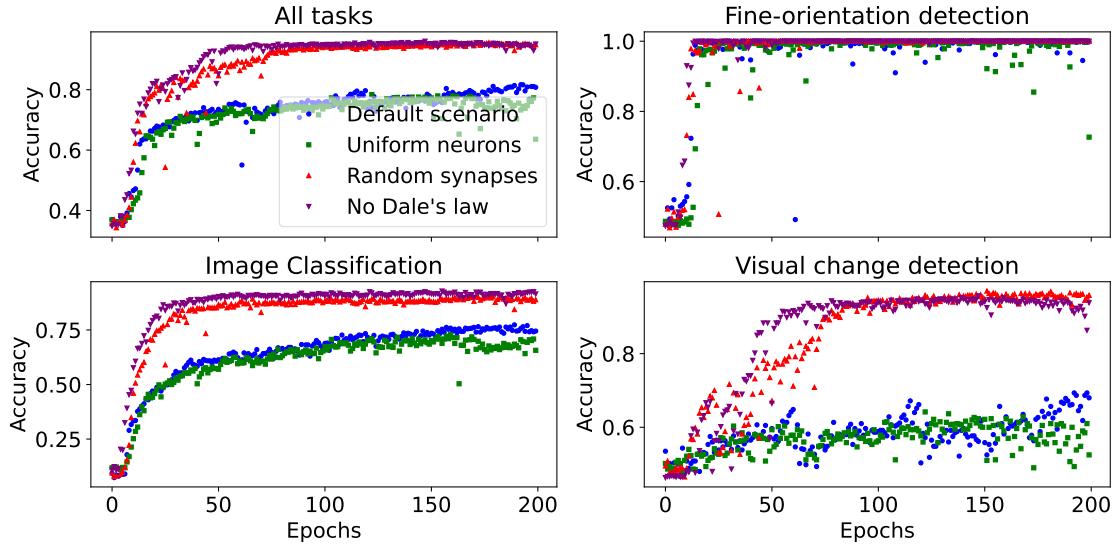
- end for**
- Compute the loss function
- if** convergence criteria met **then**

 - Break

- end if**

end while

In our built framework, any combination of the aforementioned scenarios and set-



The top-left subplot illustrates the average accuracy across all three tasks, providing a general view of the model performance. The remaining subplots depict the accuracy specific to each task: fine-orientation detection, image classification, and visual change detection. The scenarios here include a default one, one with uniform neurons, another with random synapses, and a final scenario where the Dale’s law is not applied, as aforementioned in section 2.5. Each data point marks the corresponding accuracy at a particular epoch, showcasing the learning progression and stabilization over time.

Figure 2.11 Epoch-wise test accuracy for the default and other three scenarios: average and task-specific performance across three tasks (3k neurons)

tings can be realized for the learning and testing, with the exception that only one output setting can be employed at a time. For instance, we can construct a control model that incorporates “uniform neurons,” “no Dale’s law,” and “hybrid learning” simultaneously. However, within the scope of this thesis, we present only a subset of the possible combinations, focusing on some certain analysis.

2.6 Learning of Models

2.6.1 3k-Neuron Model Learning

This subsection presents the learning of a 3k-neuron firing-rate model under different scenarios, output settings, and the hybrid learning.

2.6.1.1 Comparison of Multiple Scenarios

Figure 2.11 presents a comparison of average and task-specific performance across three tasks.

In terms of average accuracy, as shown in Figure 2.11, the “random synapses” and “no

Table 2.3 Numerical test accuracy for the default and other three scenarios: average and task-specific performance across three tasks (3k neurons)

	All tasks	Fine-orientation detection	Image classification	Visual change detection
Default scenario	81.13%	100.00%	76.48%	68.07%
Uniform neurons	77.71%	100.00%	71.67%	62.67%
Random synapses	95.35%	100.00%	90.05%	96.48%
No Dale’s law	95.66%	100.00%	92.43%	94.98%

“Dale’s law” scenarios clearly outperform both the default and “uniform neurons” scenarios. The superior performance of the “random synapses” scenario, intentionally designed to form a well-connected graph, clearly demonstrates that enhanced connectivity significantly contributes to learning effectiveness. In contrast, the default connection scheme may suffer due to the exclusion of synapses from neurons outside the designated cortex cylinder, which contains all 3,000 neurons in our simulation. We observe that the number of synapses per neuron decreases as the network size is reduced, based on the reported numbers of neurons and synapses for the full cortical patch, the 3k-neuron network, and the 6k-neuron network (see the values provided in section 2.3). This reduction in network size diminishes the recurrent connectivity within the network, despite the introduction of background inputs to emulate signals from external neurons. Also, we evaluated the connectivity of both the default and the “random synapses” networks. Although each is structured as a directed graph, the analysis reveals that the “random synapses” network is strongly connected, ensuring that every neuron is reachable from any other neuron. Conversely, the default network is weakly connected, indicating that connectivity exists only when the directionality of the synapses is disregarded. The “no Dale’s law” scenario, which allows for synaptic weight sign changes, provides the network with enhanced flexibility for learning. This scenario achieves marginally better performance than the “random synapses” scenario. Similarly, the default scenario slightly outperforms the “uniform neurons” scenario. This modest difference between the default and “uniform neurons” scenarios can be attributed to the limited diversity in neuron parameters—specifically, the two time constants—which is less rich compared to spiking models, as discussed in section 2.2 that the firing-rate model remains fewer variables for each neuron, thus probably reducing the observable performance differences between these scenarios.

In the fine-orientation detection task, as shown in Figure 2.11, all scenarios demonstrate good performance, but the default and “uniform neurons” scenarios exhibit less

Table 2.4 Numerical test accuracy for the default scenario and other three output layer settings: average and task-specific performance across three tasks (3k neurons)

	All tasks	Fine-orientation detection	Image classification	Visual change detection
Default scenario	81.13%	100.00%	76.48%	68.07%
Output: L2/3	81.82%	100.00%	80.82%	66.33%
Output: L4	77.58%	100.00%	70.48%	64.22%
Output: L6	82.54%	100.00%	78.48%	70.35%
Output: random	79.45%	100.00%	78.07%	61.23%

stability compared to the other two scenarios. The image classification task and the visual change detection task appear to be more challenging, as the “random synapses” and “no Dale’s law” scenarios significantly outperform the default and “uniform neurons” scenarios. When comparing the “random synapses” and “no Dale’s law” scenarios, it is observed that the former converges somewhat more slowly across all three tasks, which may be caused by the more recurrent connections.

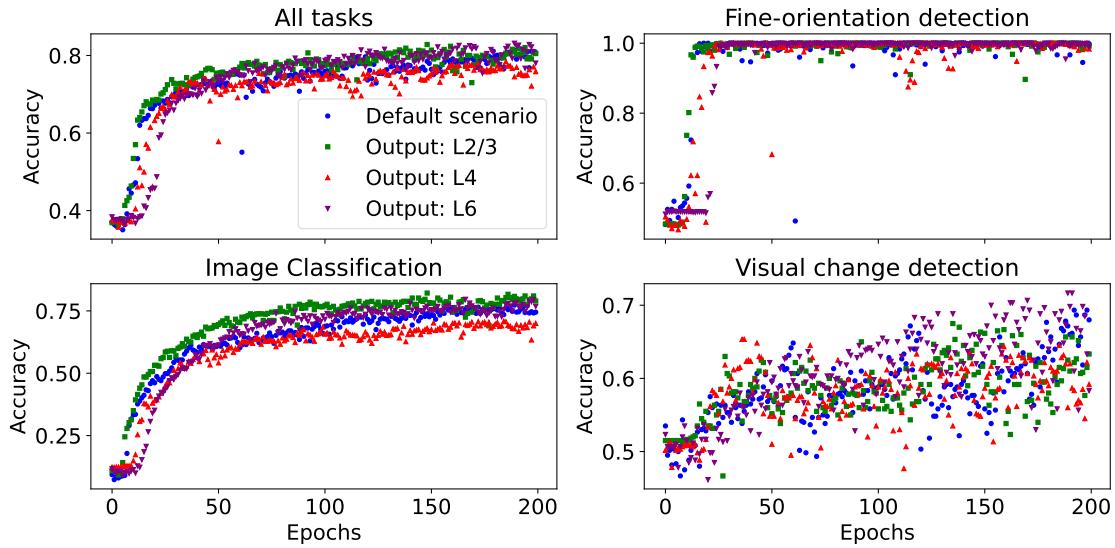
Apart from the graphical results, we use the best 10 epochs’ average test accuracy to evaluate the numerical performance for all experiments, which correspond to results shown in the tables presenting accuracy, e.g., Table 2.3 and Table 2.4.

According to the numerical results presented in Table 2.3, all scenarios achieve a perfect accuracy of 100% in the fine-orientation detection task. However, in the other two tasks, the default scenario surpasses the “uniform neurons” scenario by approximately 5%. The “random synapses” scenario outperforms the ”no Dale’s law” scenario in the visual change detection task, probably due to its higher number of recurrent connections, which enhances the model’s memory capacity. In contrast, the “no Dale’s law” scenario demonstrates superior performance on the image classification task, potentially because the increased flexibility in weights allows for improved distinguishing ability.

Some examples of training and test loss figures are presented in Appendix A, which show similarity and consistency with the corresponding accuracy figures.

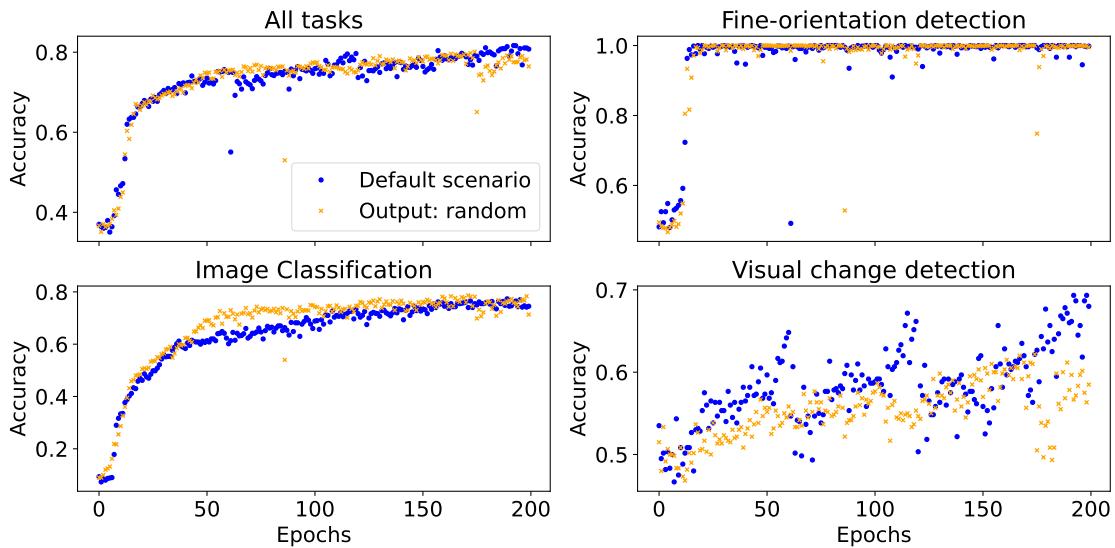
2.6.1.2 Comparison of Different Output Layers

The graphical and numerical results presented in Figure 2.12, Figure 2.13, and Table 2.4 reveal that outputting from L4, the same layer as the input, yields the worst performance. This is reasonable, as the recurrent nature of the network is not effectively utilized to enhance learning when the output is taken directly from the input layer.



The layout and labeling are same as the description in Figure 2.11.

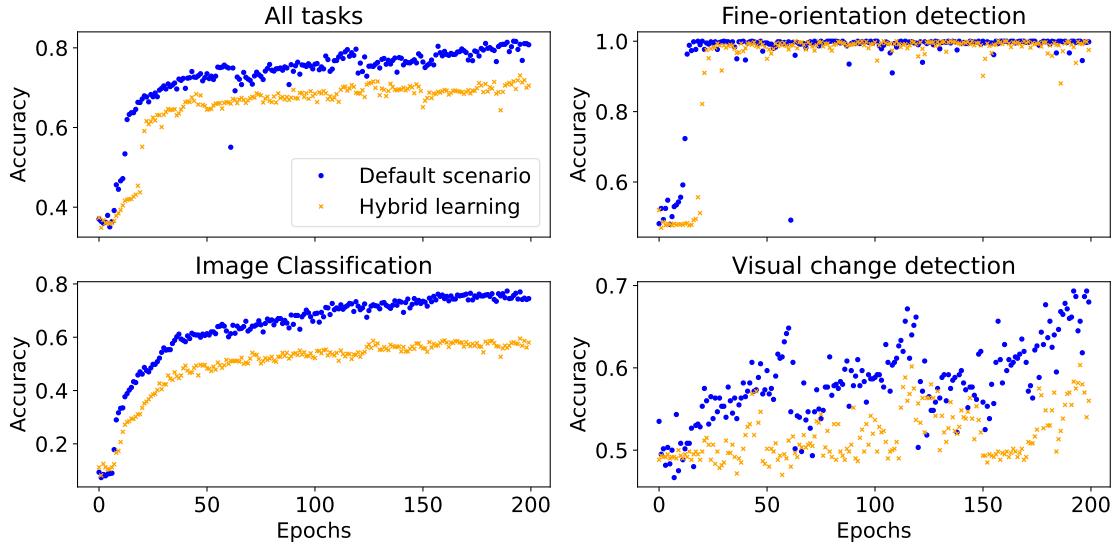
Figure 2.12 Epoch-wise test accuracy for the default scenario and other three output layer settings: average and task-specific performance across three tasks (3k neurons)



The layout and labeling are same as the description in Figure 2.11.

Figure 2.13 Epoch-wise test accuracy for the default scenario and random output: average and task-specific performance across three tasks (3k neurons)

Among the remaining output settings, outputting from L2/3, L5 (default scenario), and L6 demonstrate comparable performance, with the L6 output setting achieving the highest average accuracy across all tasks and the L5 output setting (default scenario) achieving slightly worse performance than the L2/3 and L6 output settings, which is to some extent consistent with the connection information, shown in Figure 2.3, of our selected network indicating the L6 is deeper than L5 and L2/3 from the input L4. The



The layout and labeling are same as the description in Figure 2.11.

Figure 2.14 Epoch-wise test accuracy for the default scenario and hybrid learning: average and task-specific performance across three tasks (3k neurons)

Table 2.5 Numerical test accuracy for the default scenario and hybrid learning: average and task-specific performance across three tasks (3k neurons)

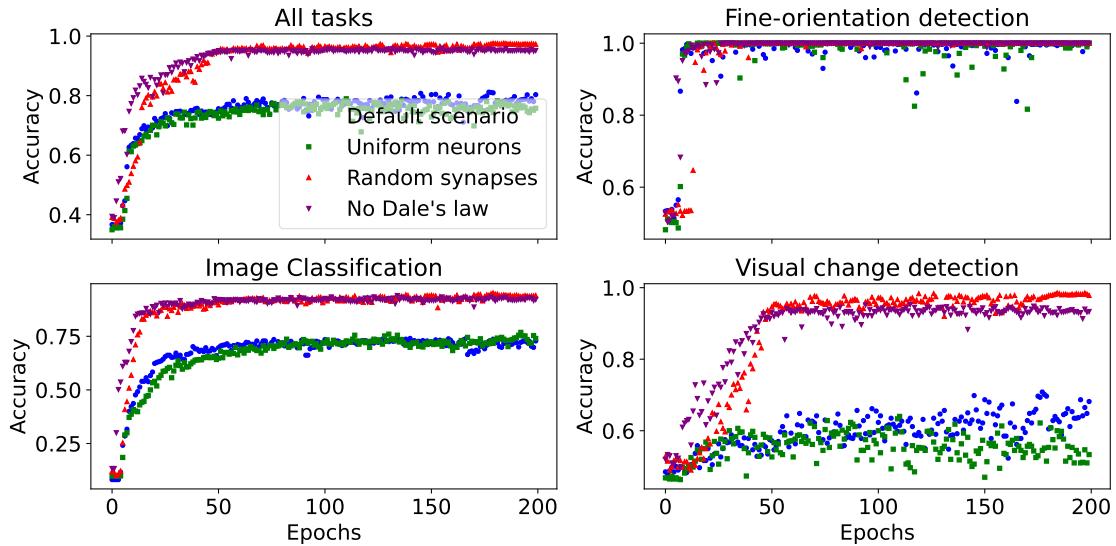
	All tasks	Fine-orientation detection	Image classification	Visual change detection
Default scenario	81.13%	100.00%	76.48%	68.07%
Hybrid learning	71.67%	100.00%	58.67%	58.45%

random output setting's performance falls between the others, suggesting that the specific choice of output layer does influence the model's learning capabilities.

It is noteworthy that although outputting from L6 provides the best results, it exhibits a slightly slower convergence compared to the other settings. This indicates a potential trade-off between the depth of the output layer and the speed of learning. Outputting from deeper layers, such as L6, may allow the model to capture more complex features and representations, leading to improved accuracy. However, this increased depth may also require more iterations for the model to converge, as the information needs to propagate through a larger portion of the network.

2.6.1.3 Comparison of Backpropagation and Hybrid Learning

The graphical and numerical results presented in Figure 2.14 and Table 2.5 reveal that the hybrid learning scenario performs worse than the default scenario, particularly in the image classification and visual change detection tasks, as the fine-orientation detection



The layout and labeling are same as the description in Figure 2.11.

Figure 2.15 Epoch-wise test accuracy for the default and other three scenarios: average and task-specific performance across three tasks (6k neurons)

Table 2.6 Numerical test accuracy for the default and other three scenarios: average and task-specific performance across three tasks (6k neurons)

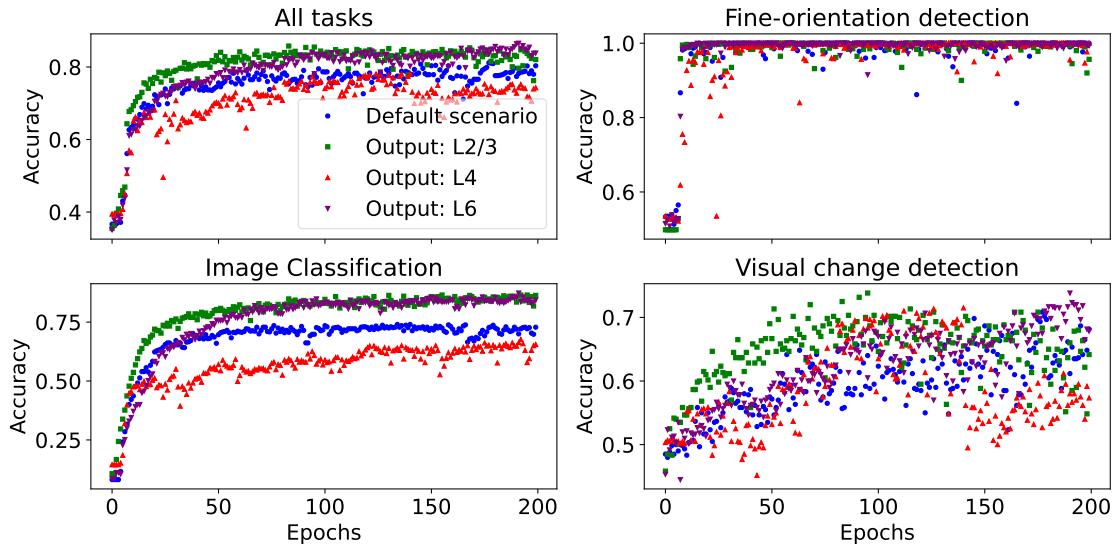
	All tasks	Fine-orientation detection	Image classification	Visual change detection
Default scenario	80.56%	100.00%	73.72%	69.30%
Uniform neurons	78.08%	100.00%	75.47%	62.30%
Random synapses	97.48%	100.00%	94.57%	98.30%
No Dale's law	95.76%	100.00%	93.28%	94.73%

task has been the least challenging task.

This shows designing an effective hybrid learning approach that combines Hebbian learning with other learning algorithms, such as backpropagation, presents significant challenges. The results indicate that the current implementation of Hebbian learning in the hybrid scenario may not be optimal and can hinder the network's performance, especially in tasks that rely on memory retention.

2.6.2 6k-Neuron Model Learning

This subsection presents the learning of a 6k-neuron firing-rate model under different scenarios, output settings, and the hybrid learning.



The layout and labeling are same as the description in Figure 2.11.

Figure 2.16 Epoch-wise test accuracy for the default scenario and other three output layer settings: average and task-specific performance across three tasks (6k neurons)

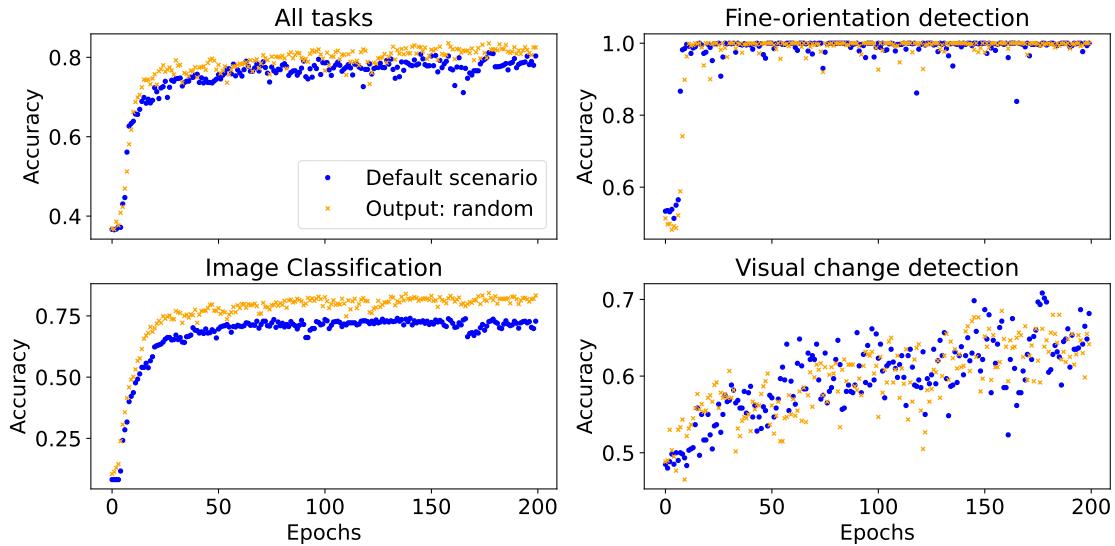
2.6.2.1 Comparison of Multiple Scenarios

The graphical and numerical results are presented in Figure 2.15 and Table 2.6. Similar to the previous findings, the “random synapses” and “no Dale’s law” scenarios outperform both the default and “uniform neurons” scenarios. Additionally, the “random synapses” scenario converges slightly more slowly than the “no Dale’s law” scenario in all three tasks.

The “random synapses” scenario exhibits the best performance across all three tasks, marginally surpassing the “no Dale’s law” scenario. For the rest two scenarios, the default scenario achieves a higher average performance compared to the “uniform neurons” scenario, with a 7% improvement in the visual change detection task. However, the default scenario performs nearly 2% worse than the “uniform neurons” scenario in the image classification task.

2.6.2.2 Comparison of Different Output Layers

According to results shown in Figure 2.16, Figure 2.17, and Table 2.7, it can be observed that similarly outputting from L4 leads to the worst performance. Outputting from L6 has the best performance, following by the results of outputting from L2/3 and L5 (default scenario). From Figure 2.5, it can be observed that in our case, L6 are located at deeper locations than L2/3 and L5, which accounts for their better accuracy.



The layout and labeling are same as the description in Figure 2.11.

Figure 2.17 Epoch-wise test accuracy for the default scenario and random output: average and task-specific performance across three tasks (6k neurons)

Table 2.7 Numerical test accuracy for the default scenario and other three output layer settings: average and task-specific performance across three tasks (6k neurons)

	All tasks	Fine-orientation detection	Image classification	Visual change detection
Default scenario	80.56%	100.00%	73.72%	69.30%
Output: L2/3	85.09%	100.00%	86.27%	71.75%
Output: L4	77.76%	100.00%	65.90%	70.50%
Output: L6	85.63%	100.00%	85.65%	72.00%
Output: random	83.03%	100.00%	83.30%	67.23%

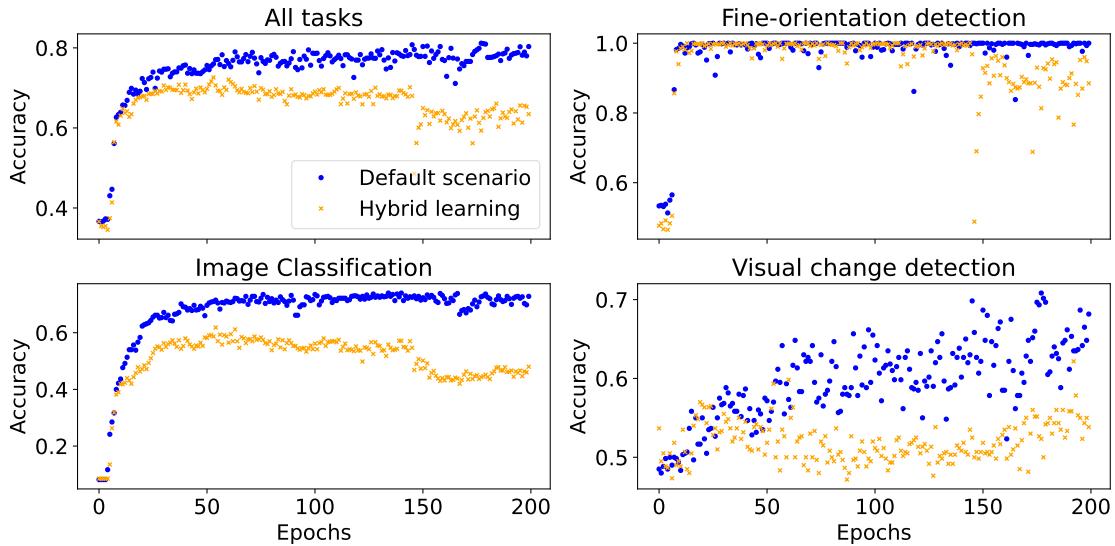
2.6.2.3 Comparison of Backpropagation and Hybrid Learning

The graphical and numerical results of hybrid learning are presented in Figure 2.18 and Table 2.8.

Similar to the results presented in subsubsection 2.6.1.3, the hybrid learning scenario, which combines backpropagation and Hebbian learning, performs worse than the pure backpropagation setting in the larger-scale network.

An intriguing observation is that the hybrid learning approach appears to require an “early stopping” mechanism to prevent the accuracy from deteriorating after reaching a plateau. Uncovering the underlying mechanisms could contribute to the development of more biologically plausible learning algorithms.

The need for “early stopping” in the hybrid learning scenario suggests that the current



The layout and labeling are same as the description in Figure 2.11.

Figure 2.18 Epoch-wise test accuracy for the default scenario and hybrid learning: average and task-specific performance across three tasks (6k neurons)

Table 2.8 Numerical test accuracy for the default scenario and hybrid learning: average and task-specific performance across three tasks (6k neurons)

	All tasks	Fine-orientation detection	Image classification	Visual change detection
Default scenario	80.56%	100.00%	74.72%	69.30%
Hybrid learning	71.24%	100.00%	59.52%	57.83%

implementation of Hebbian learning may introduce instabilities or overfitting in the network over prolonged training. As the learning progresses, the Hebbian component may cause the weights to deviate from their optimal values, leading to a decline in performance. This further shows that achieving good-performance hybrid learning is challenging.

2.7 Comparison of 3k- and 6k-Neuron Models

Comparing the results of the 3k-neuron and 6k-neuron models reveals several findings. Graphical results based on numerical results in previous sections in this chapter are given in Figure 2.19 and Figure 2.20.

One interesting observation is that the “random synapses” scenario starts to outperform the “no Dale’s law” scenario in both the image classification and visual change detection tasks when the network scale is increased from 3k to 6k neurons. This suggests that as the network grows larger, the importance of recurrent connectivity becomes more

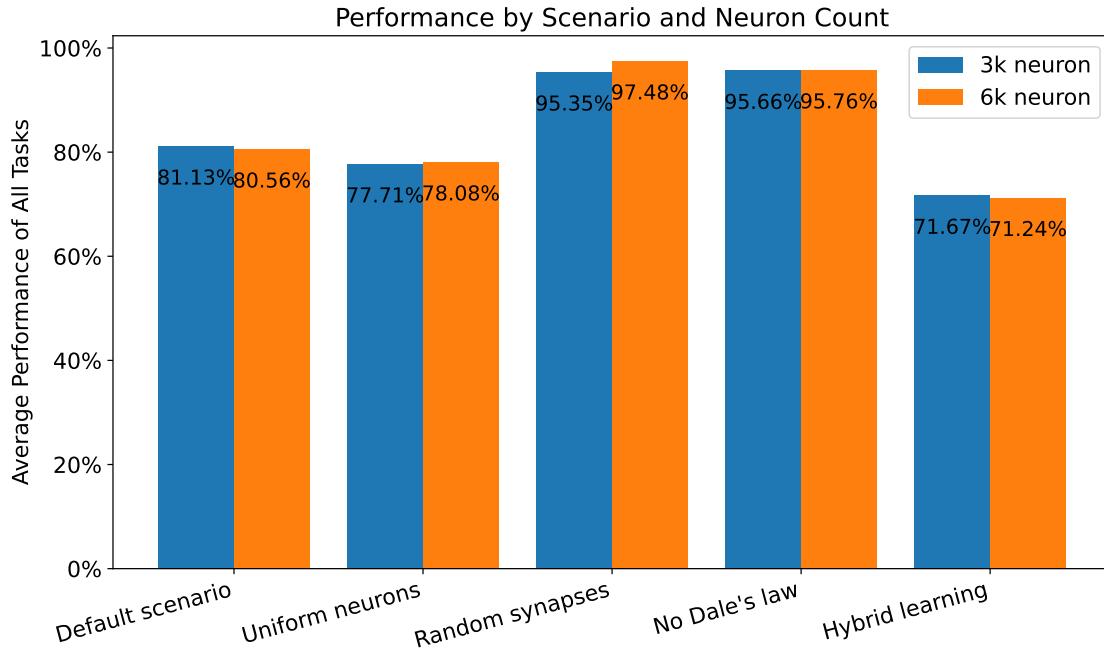


Figure 2.19 Average performance of all tasks by scenario and neuron count

pronounced than the elimination of Dale’s law. This finding offers a potential explanation for the prevalence of Dale’s law in most neurons in the brain, as the benefits of recurrent connectivity may outweigh the advantages of removing this constraint in larger-scale networks. At least, good recurrent connectivity can be an effective alternative of “no Dale’s law.”

Moreover, we observe that the “uniform neurons” scenario slightly surpasses the default scenario in the image classification task when the network scale is increased to 6k neurons. However, it still performs considerably worse in the visual change detection task. This indicates that neuronal diversity may play a vital role in storing and retrieving visual information over time, especially in larger-scale networks. The heterogeneity of neuronal properties may enable the network to encode and maintain information more effectively, facilitating the detection of changes in visual input.

We also discover that for all scenarios and different output settings, increasing the network scale from 3k to 6k neurons does not lead to a significant improvement in performance. The accuracy of the 6k-neuron network is very similar to that of the 3k-neuron network. This suggests that the model scale is not the primary bottleneck in enhancing performance, and other factors such as data preprocessing, network architecture, learning algorithms, dataset sizes, and task complexity may play a more crucial role.

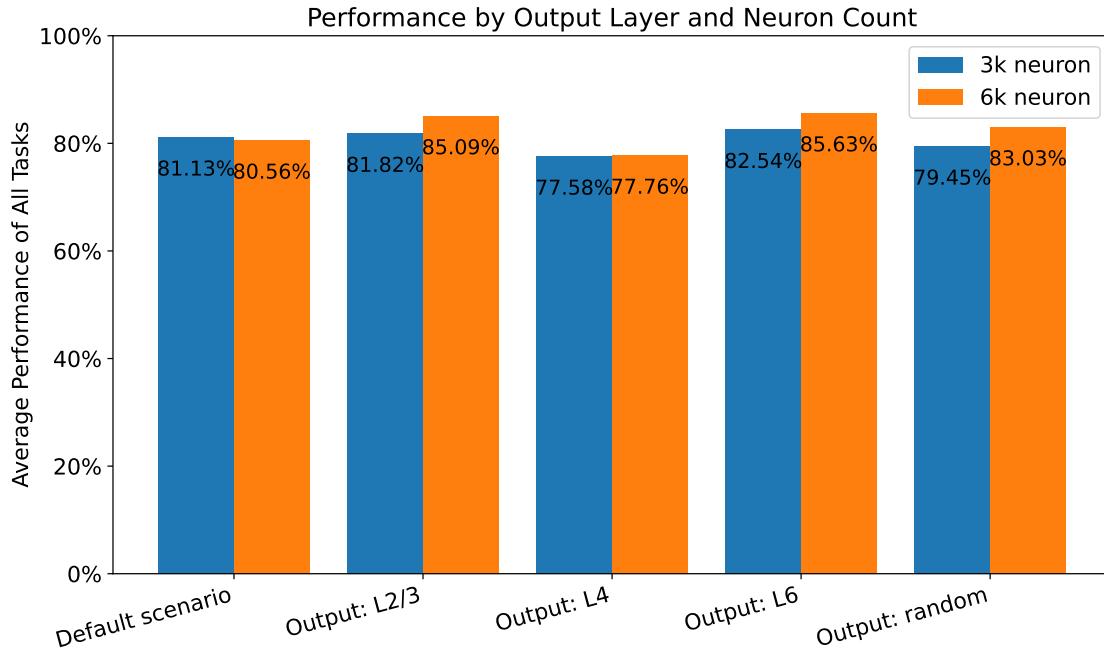


Figure 2.20 Average performance of all tasks by output layer and neuron count

2.8 Chapter Summary

This chapter focuses on the development and evaluation of a firing-rate neural network model for the V1. The chapter presents the transition from spiking models to firing-rate models, the formulation of the discrete form firing-rate model, and the application of this model to various visual tasks for learning capabilities assessment. The key points covered in the chapter are listed as follows.

- 1. Transition from spiking models to firing-rate models:** The chapter discusses the shift from detailed spiking models to more computationally friendly and efficient firing-rate models. This transition is essential for simulating large-scale neural networks and handling complex computational tasks.
- 2. Discrete form firing-rate model:** The development of a discrete form firing-rate model is described, which is crucial for training and inference within traditional machine learning frameworks and computer architectures.
- 3. Data description and experiment settings:** The chapter details the dataset from the Allen Institute for Brain Science, which contains information on neurons and synapses. The model utilizes a subset of this data to simulate networks with either 3,000 or 6,000 neurons.
- 4. Visual tasks for learning:** Three designed visual tasks are used to evaluate the model's learning capabilities: fine-orientation detection, image classification, and visual

change detection.

5. Comparative scenarios and settings: The chapter provides different scenarios for experiments, such as “uniform neurons,” “random synapses,” and “no Dale’s law,” along with various output layer settings and hybrid learning approaches.

6. Learning of models: The learning process of the 3k- and 6k-neuron models is presented, comparing performance across different scenarios, output settings, and learning methods. The results indicate that enhanced network connectivity and flexibility in synaptic weights significantly contribute to learning effectiveness. Neuronal diversity contributes to the memory capability of a network. Designing a good hybrid learning is challenging.

7. Comparison of 3k- and 6k-neuron models: The chapter concludes with a comparison of the model performance between the 3k-neuron and 6k-neuron networks. Increasing the neuron count from 3k to 6k does not lead to a significant improvement in performance, suggesting that other factors like data preprocessing and network architecture may be more impactful.

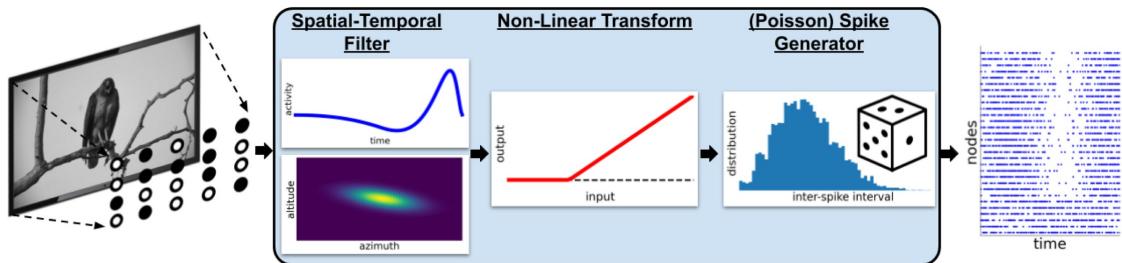
CHAPTER 3 APPLYING LGN FILTERS TO V1 MODEL

In this chapter, we add LGN filters to the V1 model to see how they, as an important factor, can affect the model learning. The comparison of multiple scenarios, of different output layers, and of backpropagation and hybrid learning, as well as the comparison of 3k- and 6k-neuron models with LGN in this chapter show the results and analysis in similar way to the ones in chapter 2. Then based on the results both in chapter 2 and this chapter, we conduct the analysis of models with and without LGN in section 3.4, which is an important part of this chapter.

3.1 LGN Filters

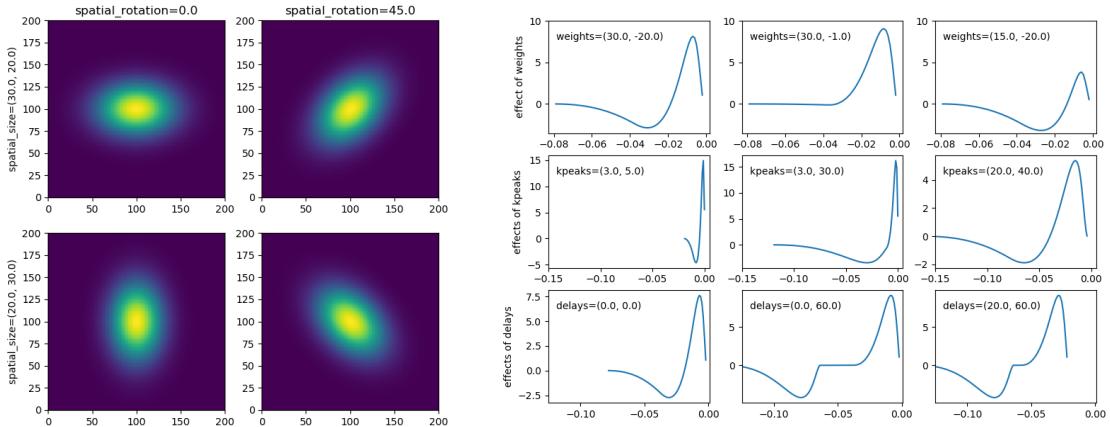
In the brain modeling toolkit (BMTK)^[54-55] developed by the Allen Institute for Brain Science, a linear-nonlinear-Poisson (LNP) model^[143-145] is employed to describe the functions of the LGN. The LNP model is composed of three key elements: a spatiotemporal linear filter, a nonlinear transformation, and a Poisson spike generator, as shown in Figure 3.1. For our firing-rate model, we focus on the filters and non-linearity, as spike generation is not required.

There are three types of cells in the BMTK LGN module: simple ON/OFF filter cells, spatial mixture filter cells, and dual filter cells.



The general workflow of LGN filters for processing visual stimuli is depicted in the schematic. When a visual input, such as a movie, is encountered, it is spatially and temporally analyzed by a distributed array of filters. Each filter performs a sequence of computations on the movie frames: it begins by convolving the visual input with predefined spatial and temporal kernels to capture the dynamic patterns of activity. This is followed by a non-linear rectification process that translates the convolved signals into a time-varying positive firing rate. The firing rate effectively captures the filter's reaction to the dynamic visual input. For further computational modeling or analysis, these firing rates can be transformed into sequences of spikes, or spike trains, employing a Poisson spike generator, which replicates the probabilistic character of neuronal spike generation.

Figure 3.1 Schematic of visual input processing of LGN filters^[54]



(a) Spatial filters with different size and rotations are visualized.

(b) Temporal filters with different peak magnitudes, spreads, and delays are visualized.

Figure 3.2 Examples of spatial and temporal filters for LGN cells^[55]

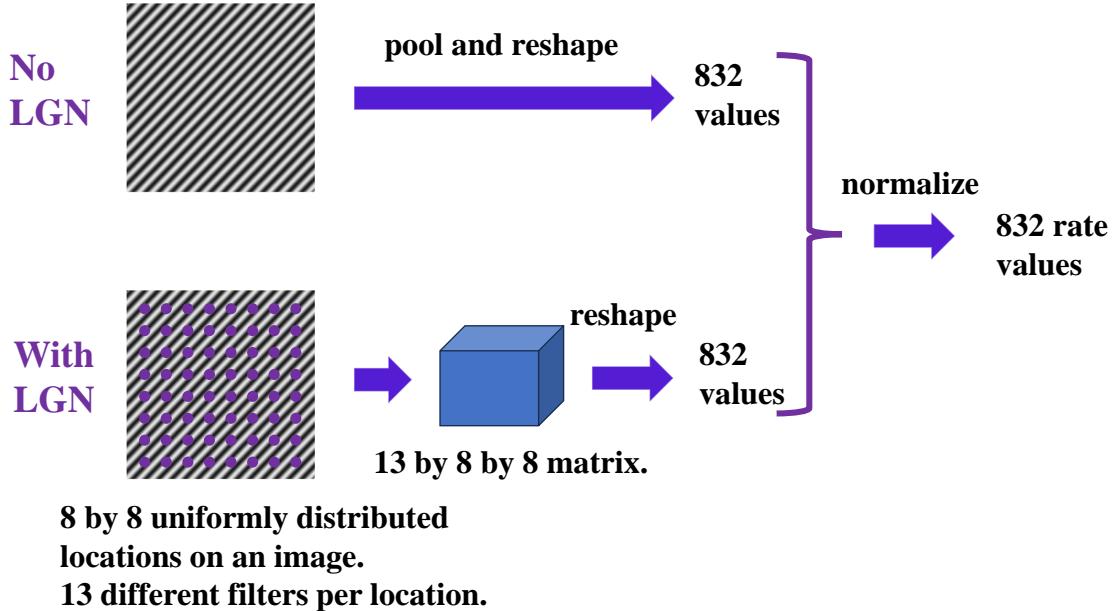


Figure 3.3 Schematic diagram of the data preprocessing from images to rates of both no LGN and with LGN settings

A simple ON/OFF filter, such as sON_TF8 and tOFF_TF15, is constructed by combining a Gaussian spatial filter and a double cosine temporal filter^[76,146]. In these examples, the prefixes “s” and “t” denote the sustainedness of the cell’s response, with “s” representing sustained (slow response) and “t” representing transient (quick response). The suffixes “ON” and “OFF” indicate the polarity of the cell, signifying whether it responds to an increase (ON) or decrease (OFF) in light intensity. The labels “TF8” and “TF15” represent the temporal frequency of the filter in Hz.

The Gaussian spatial filter determines the spatial extent and shape of the cell’s recep-

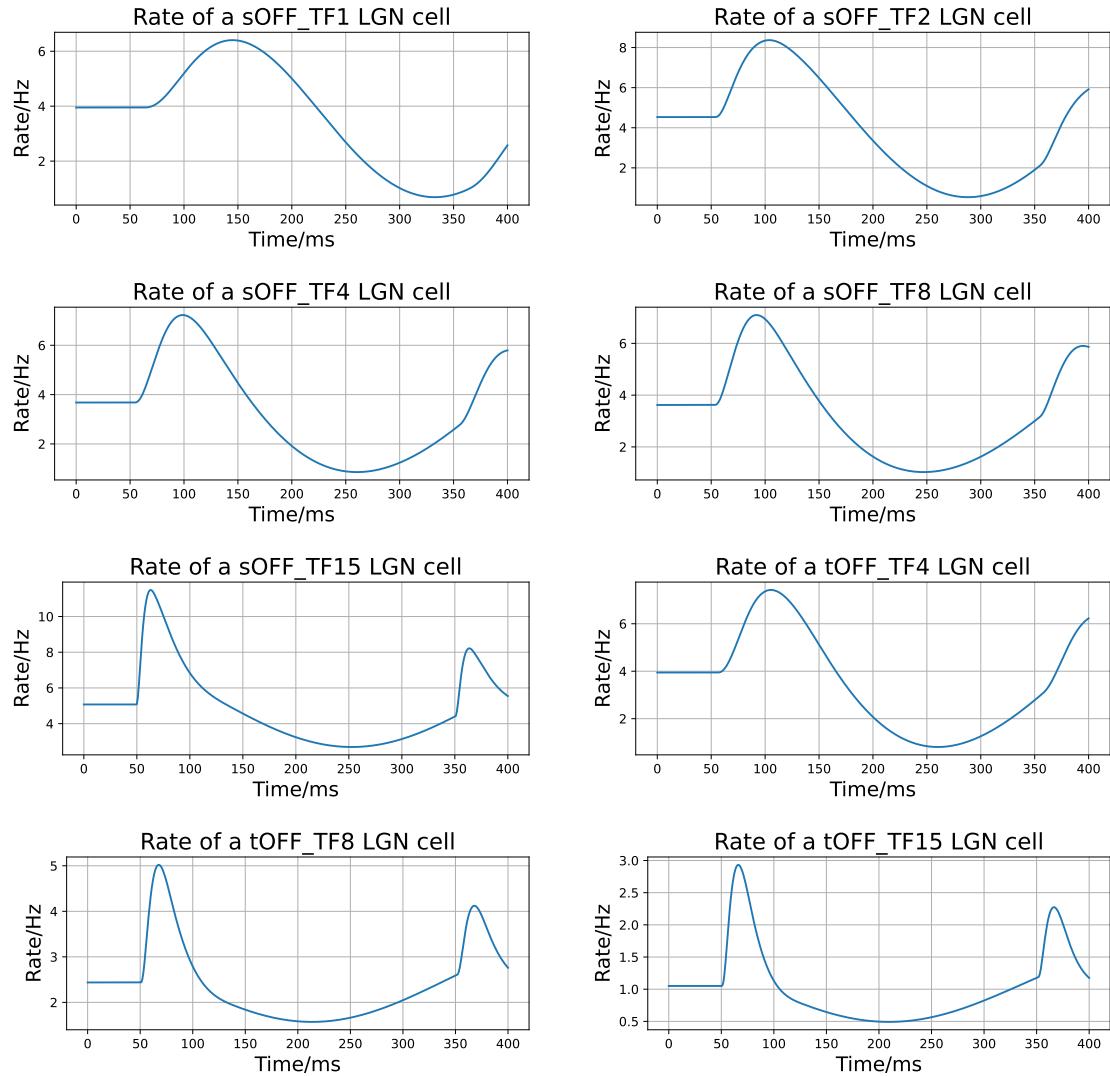


Figure 3.4 Rate responses of OFF cells in our LGN model to visual stimuli from the fine-orientation detection task

tive field, while the double cosine temporal filter captures the temporal dynamics of the cell’s response to visual stimuli. The combination of these filters allows the LGN cells to selectively respond to specific spatial and temporal patterns of light intensity changes. See Figure 3.2 for examples of spatial and temporal filters with different parameters.

Spatial mixture cells in the LGN possess unique receptive fields that result from the combination of two distinct spatial filters. These filters work together to form a structure reminiscent of a Mexican hat^[147], which allows the cells to respond preferentially to specific patterns of light intensity in their receptive fields. Specifically, spatial mixture cells are highly sensitive to areas of contrast, such as a bright region encircled by darkness or a dark region surrounded by brightness.

Dual filter cells are distinguished by their integration of two separate simple filters

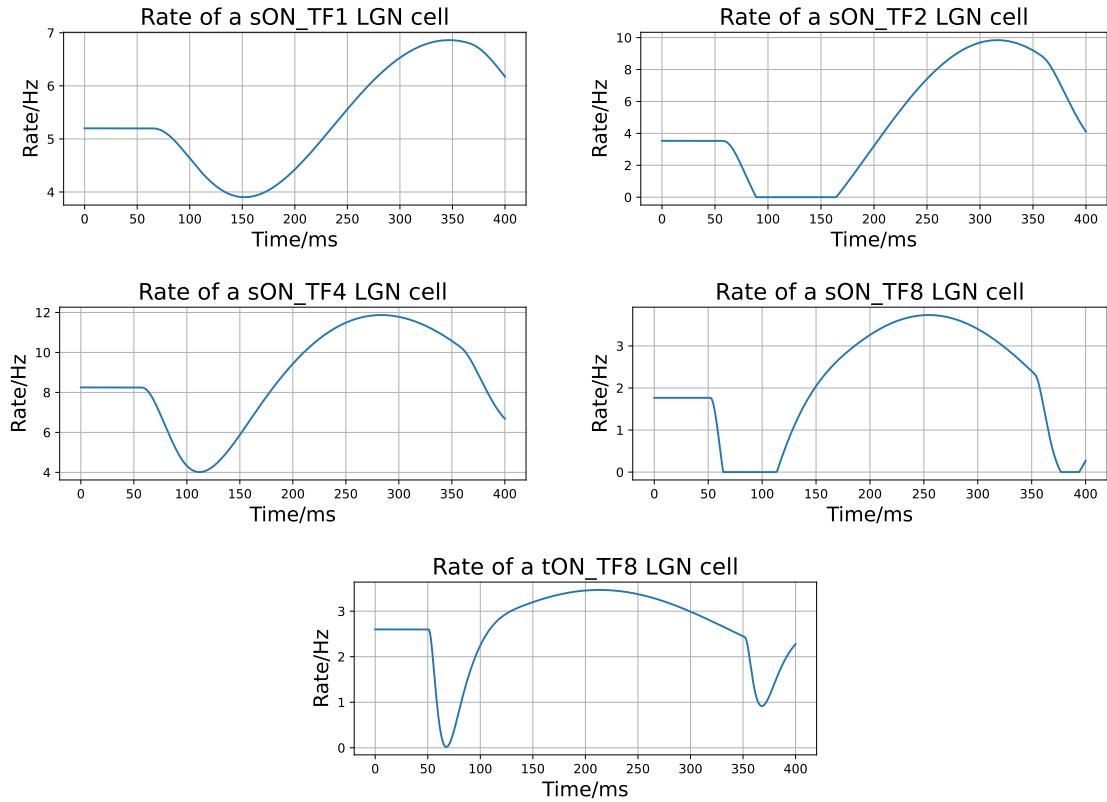
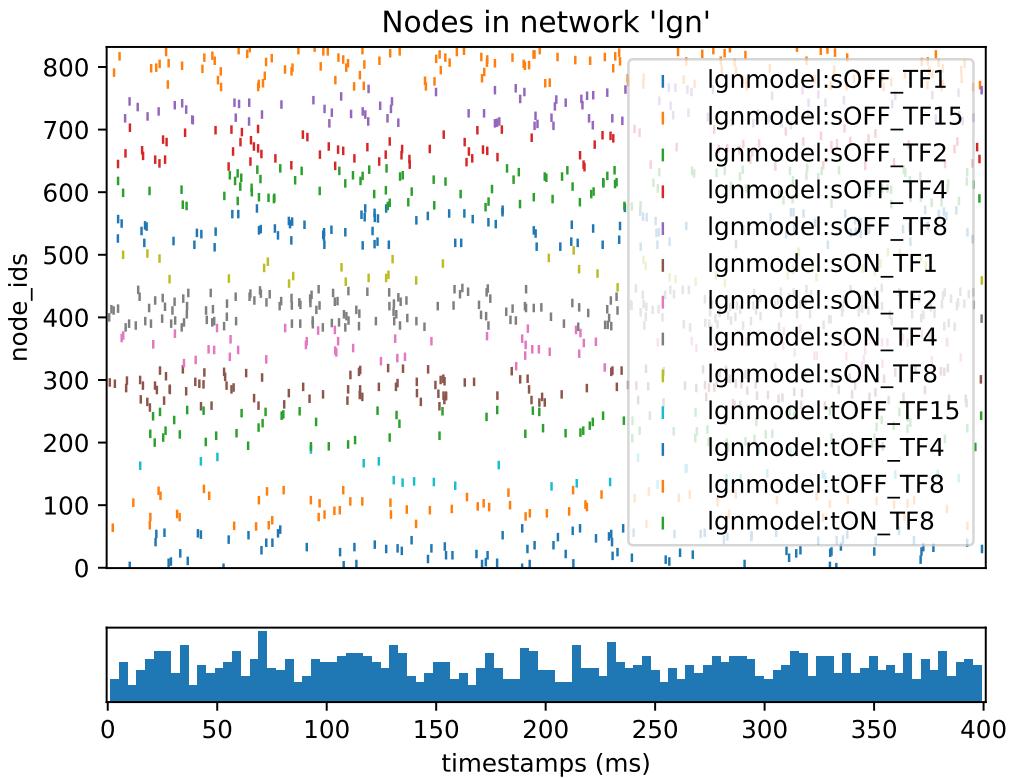


Figure 3.5 Rate responses of all types of ON cells in our LGN model to visual stimuli from the fine-orientation detection task

into a single functional unit.

To model the LGN, we utilize a set of 13 different types of simple filter cells, rather than incorporating spatial mixture cells or dual filter cells. This approach is based on the understanding that the effects of spatial mixture and dual filter cells can be achieved through the combination of simple filter cells. By using an array of simple filter cells with diverse properties, we aim to capture the essential features of LGN processing while maintaining a more tractable and efficient model.

For each of the 13 simple filter cell types, we distribute 64 cells with a spatial size of 15 (arranged in an 8 by 8 grid) uniformly across the input image (200 by 200), which represents a single frame in a movie. This spatial arrangement ensures that the cells collectively cover the entire visual field and can extract features from all regions of the input. As a result, our LGN model generates an output consisting of 832 channels, corresponding to the responses of the 832 simple filter cells. This high-dimensional output serves as the input to the subsequent processing stage, where each channel can be fed to a corresponding neuron in the V1 model. The schematic diagram of the data preprocessing of both no LGN and with LGN settings is given in Figure 3.3 for comparison.



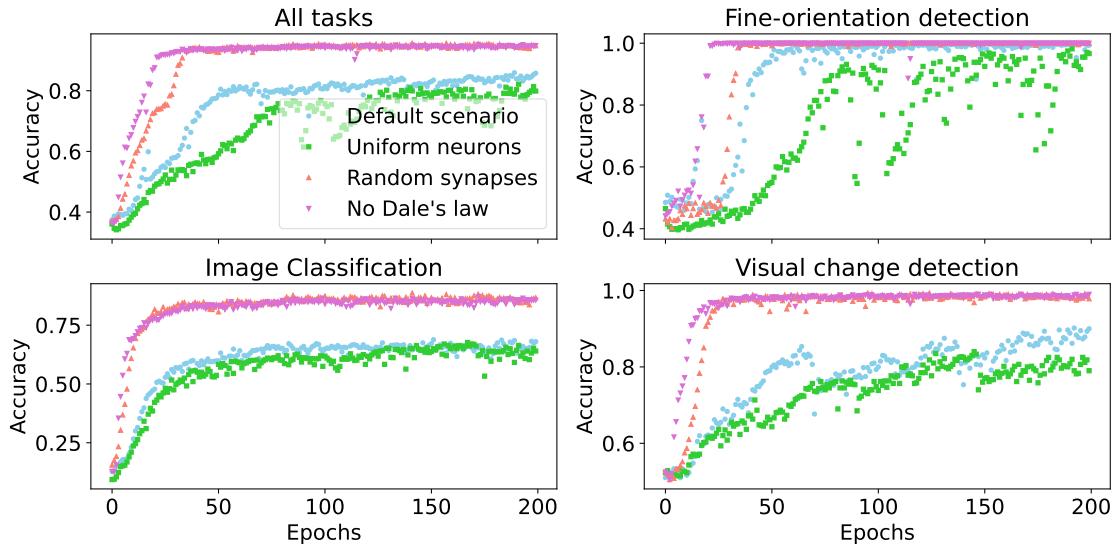
The raster plot includes all 832 cells of 13 types in our LGN models.

Figure 3.6 Spike responses of all cells in our LGN model to visual stimuli from the fine-orientation detection task

To evaluate the rate generation capabilities of our LGN model, here we employ a random movie of visual stimuli derived from a fine-orientation detection task. The results of this evaluation are presented in Figure 3.4 and Figure 3.5, which showcase the response of a single representative cell for each cell type. These figures reveal distinct changes in the firing rate at approximately 50 ms and 350 ms, coinciding with the onset and offset of the drifting gratings (see Figure 2.8), respectively. These rate modulations well demonstrate the ability of the LGN model to capture the temporal dynamics of certain receptive fields in the visual input.

We also generate spikes based on the computed firing rates. Figure 3.6 displays the spiking responses of all 832 cells in our LGN model.

The responses of the LGN model to visual stimuli from the the image classification and visual change detection tasks are provided in Appendix C, where richer dynamics of the LGN cells can be observed.



The layout and labeling are same as the description in Figure 2.11.

Figure 3.7 Epoch-wise test accuracy for the default and other three scenarios: average and task-specific performance across three tasks (3k neurons, with LGN)

3.2 Learning of Models with LGN

In this section, we employ the preprocessed data using the LGN model as input for training our V1 models. This approach aims to enhance the biological plausibility of our simulations by incorporating the early stages of visual processing that occur in the LGN. We here just provide simple analysis of the results in this section. We later focus more on the comparison of 3k- and 6k- Models with LGN and the comparisons of models with and without LGN in the sections immediately after this section.

3.2.1 3k-Neuron Model Learning with LGN

This subsection presents the learning of a 3k-neuron firing-rate model with LGN under different scenarios, output settings, and the hybrid learning.

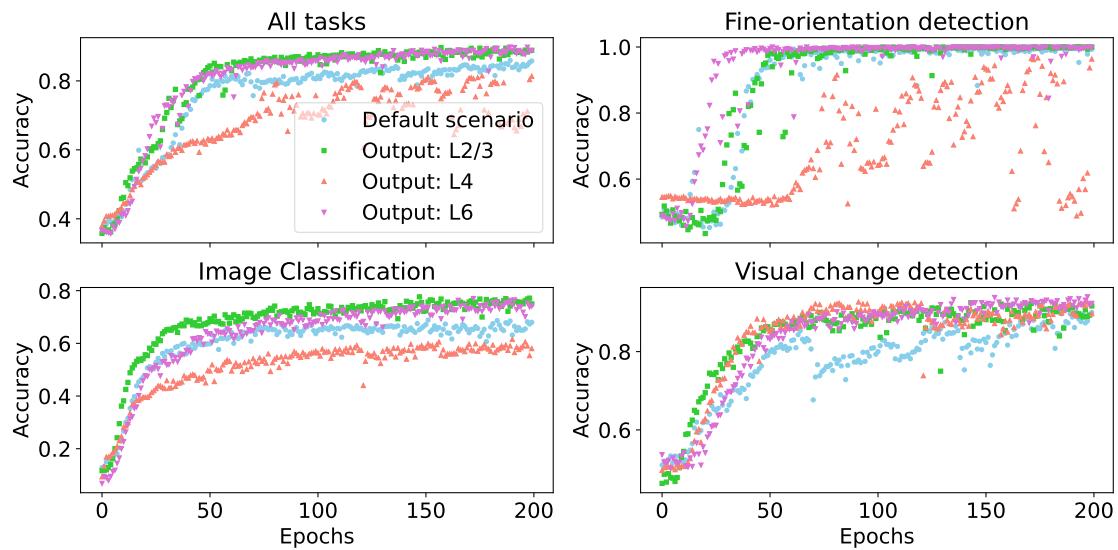
3.2.1.1 Comparison of Multiple Scenarios

The graphical and numerical results of the performance of the different scenarios in our V1 models when trained with LGN-preprocessed data are presented in Figure 3.7 and Table 3.1.

The “random synapses” and “no Dale’s law” scenarios continue to demonstrate superior performance compared to both the default and “uniform neurons” scenarios. The four scenarios exhibit distinct convergence speeds during the training process. The “no Dale’s law” scenario converges the fastest. The “random synapses” scenario follows closely. The

Table 3.1 Numerical test accuracy for the default and other three scenarios: average and task-specific performance across three tasks (3k neurons, with LGN)

	All tasks	Fine-orientation detection	Image classification	Visual change detection
Default scenario	85.07%	99.82%	68.00%	89.40%
Uniform neurons	80.84%	96.73%	66.62%	82.90%
Random synapses	95.45%	100.00%	87.75%	99.17%
No Dale's law	95.01%	100.00%	86.37%	98.95%



The layout and labeling are same as the description in Figure 2.11.

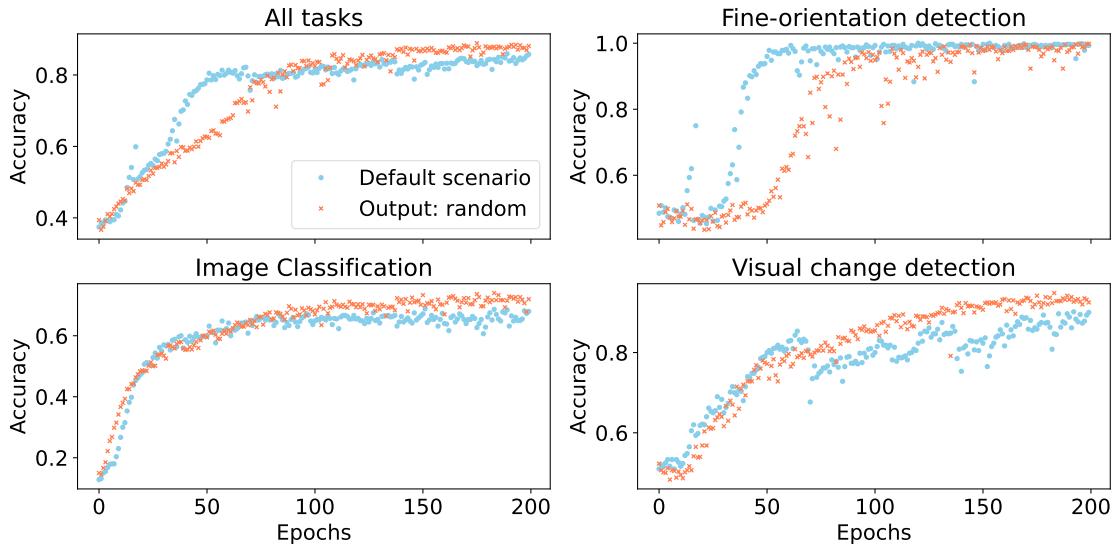
Figure 3.8 Epoch-wise test accuracy for the default scenario and other three output layer settings: average and task-specific performance across three tasks (3k neurons, with LGN)

default scenario converges more slowly than the “no Dale’s law” and “random synapses” scenarios, but still outpaces the “uniform neurons” scenario. The “uniform neurons” scenario struggles to converge for the fine-orientation detection task, although it has good performance at some epochs.

All four scenarios achieve a good average accuracy exceeding 80% when trained with LGN-preprocessed data. This high level of performance demonstrates the effectiveness of using the LGN-preprocessed input for training our V1 models.

3.2.1.2 Comparison of Different Output Layers

The graphical and numerical results of the performance of the different output settings in our V1 models when trained with LGN-preprocessed data are presented in Figure 3.8, Figure 3.9, and Table 3.2.



The layout and labeling are same as the description in Figure 2.11.

Figure 3.9 Epoch-wise test accuracy for the default scenario and random output: average and task-specific performance across three tasks (3k neurons, with LGN)

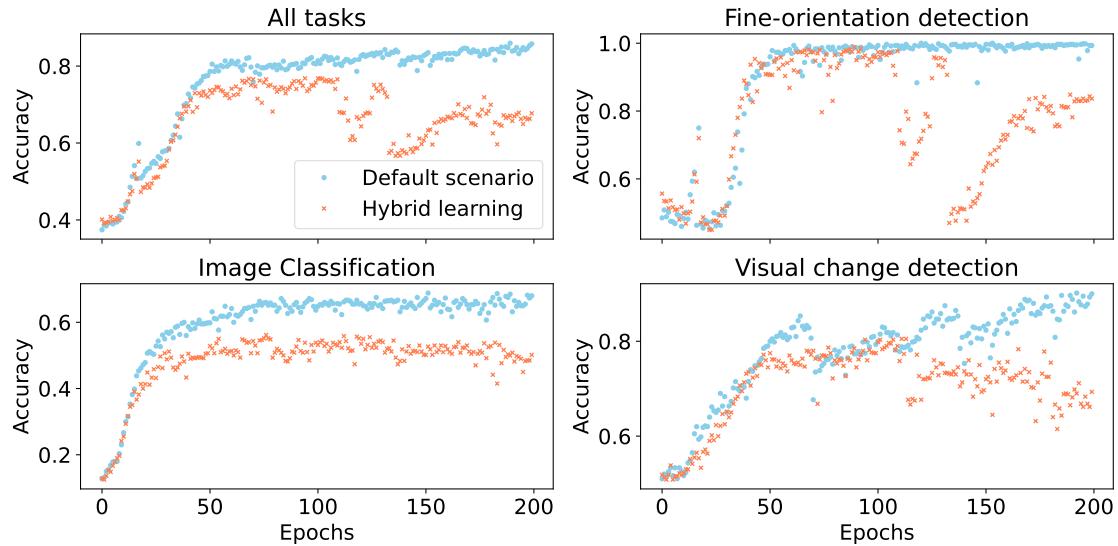
Table 3.2 Numerical test accuracy for the default scenario and other three output layer settings: average and task-specific performance across three tasks (3k neurons, with LGN)

	All tasks	Fine-orientation detection	Image classification	Visual change detection
Default scenario	85.07%	99.82%	68.00%	89.40%
Output: L2/3	89.24%	100.00%	76.83%	91.90%
Output: L4	80.79%	94.82%	59.98%	92.40%
Output: L6	89.37%	100.00%	75.82%	93.40%
Output: random	88.24%	99.55%	73.08%	93.87%

The results show that outputting from L6 yields the best performance, and outputting from L4 leads to the poorest performance. When the output is taken from L4, the model fails to converge for the fine-orientation detection task, resulting in an accuracy lower than 60%. Despite the differences in performance across the output settings, all output settings achieve a good average accuracy exceeding 80% when trained with LGN-preprocessed data.

3.2.1.3 Comparison of Backpropagation and Hybrid Learning

The graphical and numerical results of hybrid learning when trained with LGN-preprocessed data are presented in Figure 3.10 and Table 3.3. The backpropagation algorithm outperforms the hybrid learning approach in terms of both accuracy and conver-



The layout and labeling are same as the description in Figure 2.11.

Figure 3.10 Epoch-wise test accuracy for the default scenario and hybrid learning: average and task-specific performance across three tasks (3k neurons, with LGN)

Table 3.3 Numerical test accuracy for the default scenario and hybrid learning: average and task-specific performance across three tasks (3k neurons, with LGN)

	All tasks	Fine-orientation detection	Image classification	Visual change detection
Default scenario	85.07%	99.82%	68.00%	89.40%
Hybrid learning	76.61%	98.02%	55.35%	79.90%

gence. “Early stop” for hybrid learning could be a effective measure for it to maintain its accuracy.

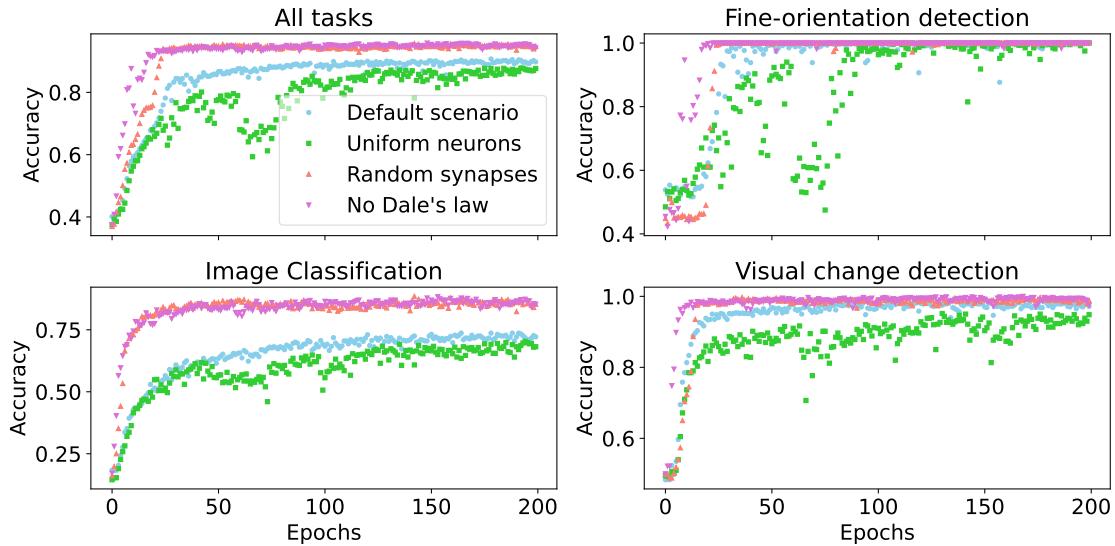
3.2.2 6k-Neuron Model Learning with LGN

This subsection presents the learning of a 6k-neuron firing-rate model with LGN under different scenarios, output settings, and the hybrid learning.

3.2.2.1 Comparison of Multiple Scenarios

The graphical and numerical results of the performance of the different scenarios when trained with LGN-preprocessed data are presented in Figure 3.11 and Table 3.4.

The “random synapses” and “no Dale’s law” scenarios still show best performance compared to both the default and “uniform neurons” scenarios. The default scenario outperforms the “uniform neurons” scenario in both the image classification and visual change tasks. All four scenarios achieve an average accuracy exceeding 85% when trained



The layout and labeling are same as the description in Figure 2.11.

Figure 3.11 Epoch-wise test accuracy for the default and other three scenarios: average and task-specific performance across three tasks (6k neurons, with LGN)

Table 3.4 Numerical test accuracy for the default and other three scenarios: average and task-specific performance across three tasks (6k neurons, with LGN)

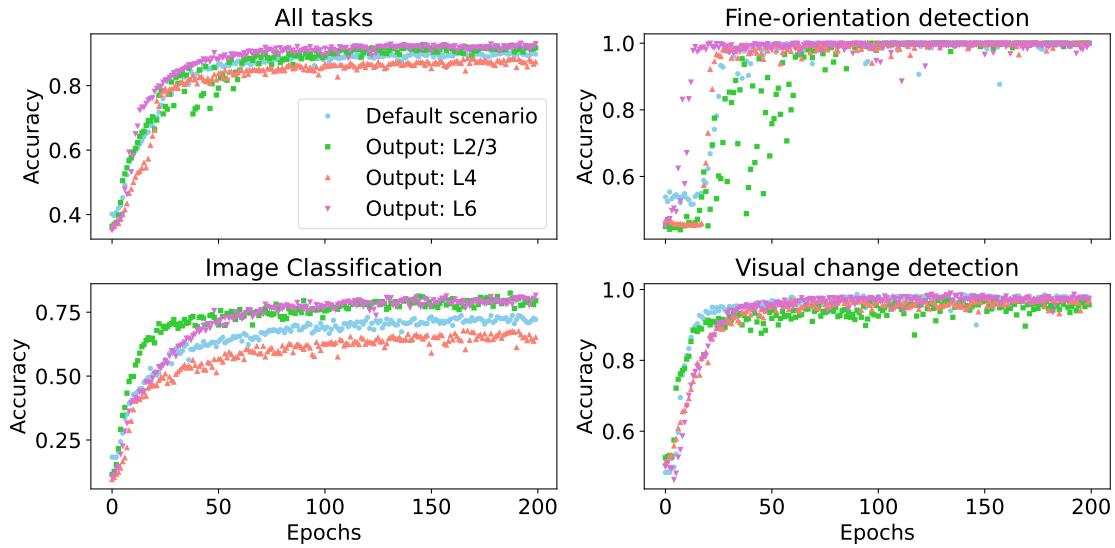
	All tasks	Fine-orientation detection	Image classification	Visual change detection
Default scenario	90.33%	100.00%	73.30%	98.28%
Uniform neurons	87.58%	100.00%	69.57%	94.77%
Random synapses	95.44%	100.00%	87.40%	99.50%
No Dale's law	95.64%	100.00%	87.58%	99.80%

with LGN-preprocessed data.

3.2.2.2 Comparison of Different Output Layers

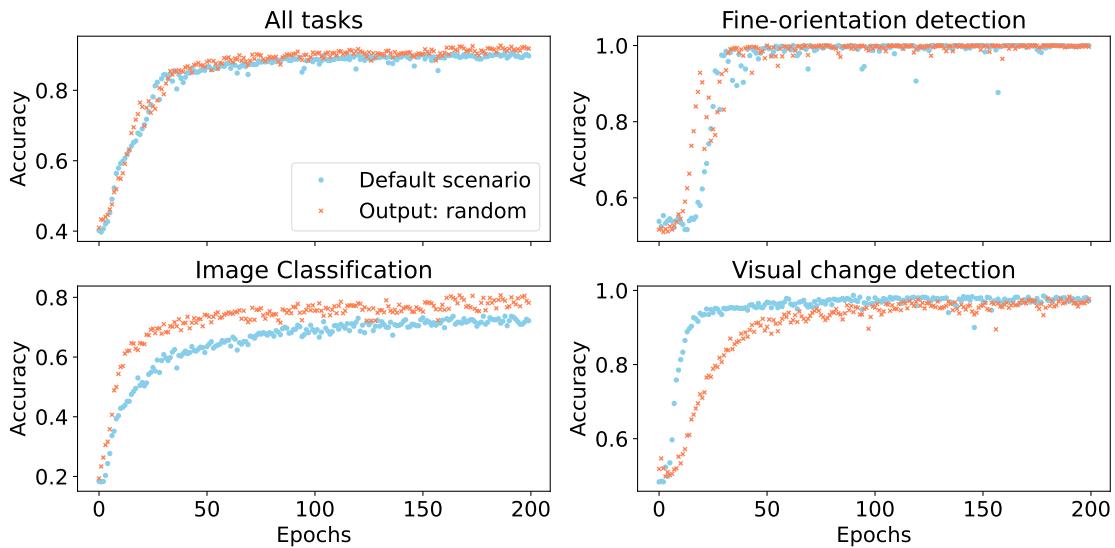
The graphical and numerical results of the performance of the different output settings in our V1 models when trained with LGN-preprocessed data are presented in Figure 3.12, Figure 3.13, and Table 3.5.

The results show that except for the random output, outputting from L6 yields the best performance, followed by outputting from L2/3, and L5, which is the default scenario. Outputting from L4 performs worse than others. All output settings achieve an average accuracy exceeding 85% when trained with LGN-preprocessed data.



The layout and labeling are same as the description in Figure 2.11.

Figure 3.12 Epoch-wise test accuracy for the default scenario and other three output layer settings: average and task-specific performance across three tasks (6k neurons, with LGN)



The layout and labeling are same as the description in Figure 2.11.

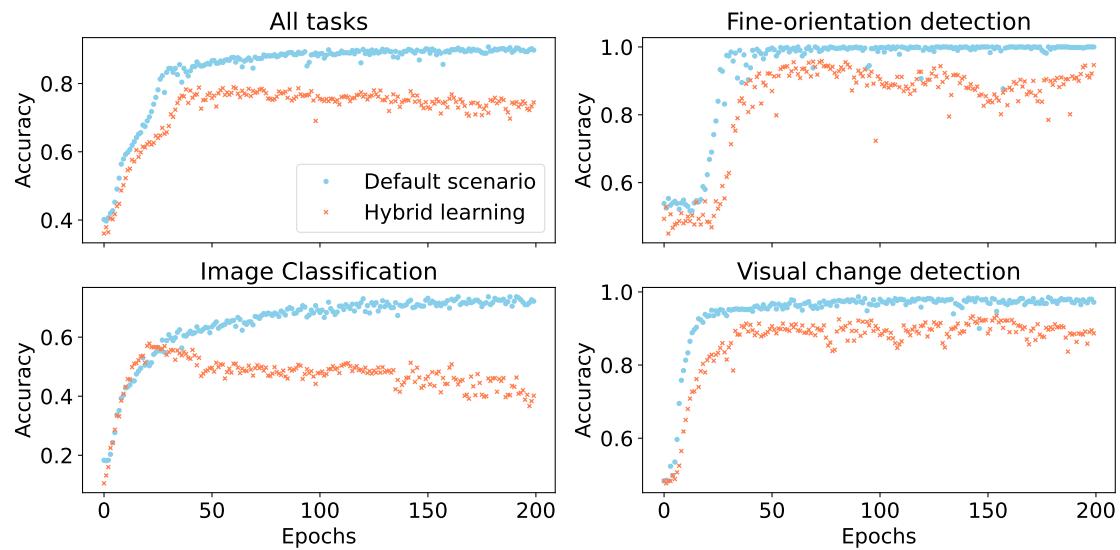
Figure 3.13 Epoch-wise test accuracy for the default scenario and random output: average and task-specific performance across three tasks (6k neurons, with LGN)

3.2.2.3 Comparison of Backpropagation and Hybrid Learning

The graphical and numerical results of hybrid learning when trained with LGN-preprocessed data are presented in Figure 3.14 and Table 3.6. The backpropagation algorithm outperforms the hybrid learning approach in terms of both accuracy and convergence. “Early stop” for hybrid learning would still be helpful. But a notable observation is that the hybrid learning’s accuracy of the visual change detection task reaches to higher

Table 3.5 Numerical test accuracy for the default scenario and other three output layer settings: average and task-specific performance across three tasks (6k neurons, with LGN)

	All tasks	Fine-orientation detection	Image classification	Visual change detection
Default scenario	90.33%	100.00%	73.30%	98.28%
Output: L2/3	92.29%	100.00%	80.88%	97.07%
Output: L4	88.16%	100.00%	67.28%	97.72%
Output: L6	92.78%	100.00%	80.97%	98.48%
Output: random	92.33%	100.00%	80.23%	97.58%



The layout and labeling are same as the description in Figure 2.11.

Figure 3.14 Epoch-wise test accuracy for the default scenario and hybrid learning: average and task-specific performance across three tasks (6k neurons, with LGN)

than 90%.

3.3 Comparison of 3k- and 6k-Neuron Models with LGN

A comparison of the graphical and numerical results presented in the previous sections of this chapter reveals interesting insights into the performance of our V1 models

Table 3.6 Numerical test accuracy for the default scenario and hybrid learning: average and task-specific performance across three tasks (6k neurons, with LGN)

	All tasks	Fine-orientation detection	Image classification	Visual change detection
Default scenario	90.33%	100.00%	73.30%	98.28%
Hybrid learning	78.53%	95.18%	56.58%	92.67%

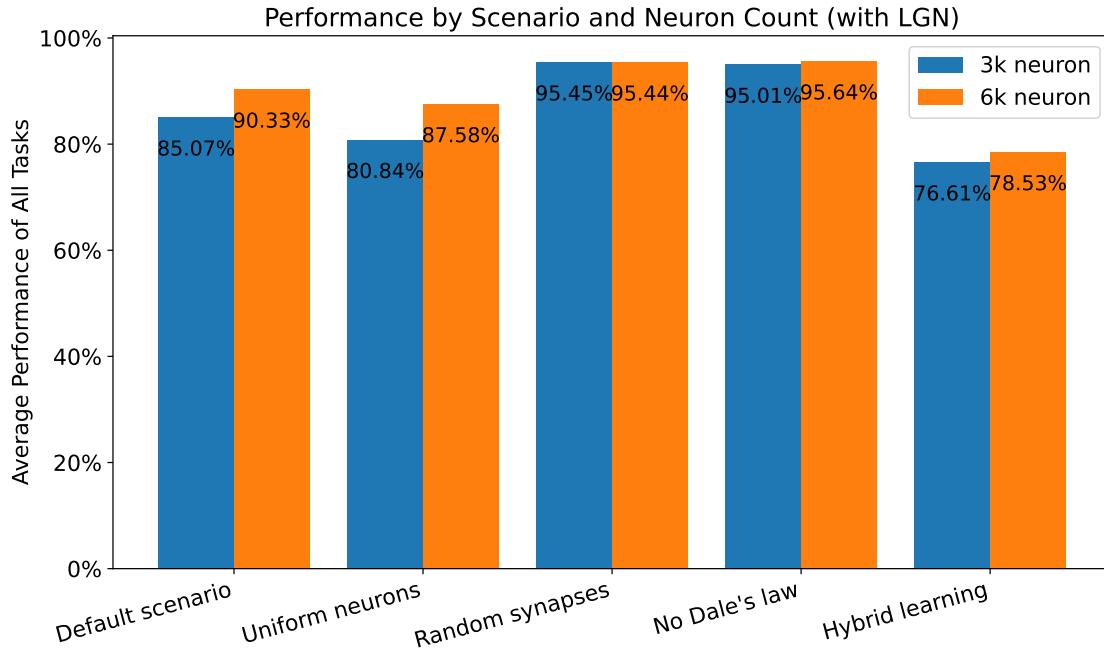


Figure 3.15 Average performance of all tasks by scenario and neuron count (with LGN)

with LGN as the network scale increases from 3k to 6k neurons.

New graphical results based on numerical results in previous sections in this chapter are given in Figure 3.15 and Figure 3.16 for comparison.

The performance of the “random synapses” and “no Dale’s law” scenarios remains stable when the network size is increased. On the other hand, the performance of the default and “uniform neurons” scenarios improves significantly with the increase in network size. As the number of neurons is doubled from 3k to 6k, these scenarios exhibit performance levels that become closer to those of the “random synapses” and “no Dale’s law” scenarios. This indicates that the additional computational capacity and representational power provided by the larger network allow the default and “uniform neurons” scenarios to better learn and capture the underlying patterns in the LGN-preprocessed data. The performance improvement associated with the increase in network size is also observed across all output settings and the hybrid learning. All three tasks’ performance is improved when the network size increases, and the improvement in the visual change detection is the most significant.

It is worth noting that in the 3k-neuron network, the “uniform neurons” scenario struggles to converge, outputting from L4 leads to unsuccessful convergence, and the hybrid learning approach converges poorly. However, when the network size is increased to 6k neurons, these issues are substantially mitigated. The larger network appears to pro-

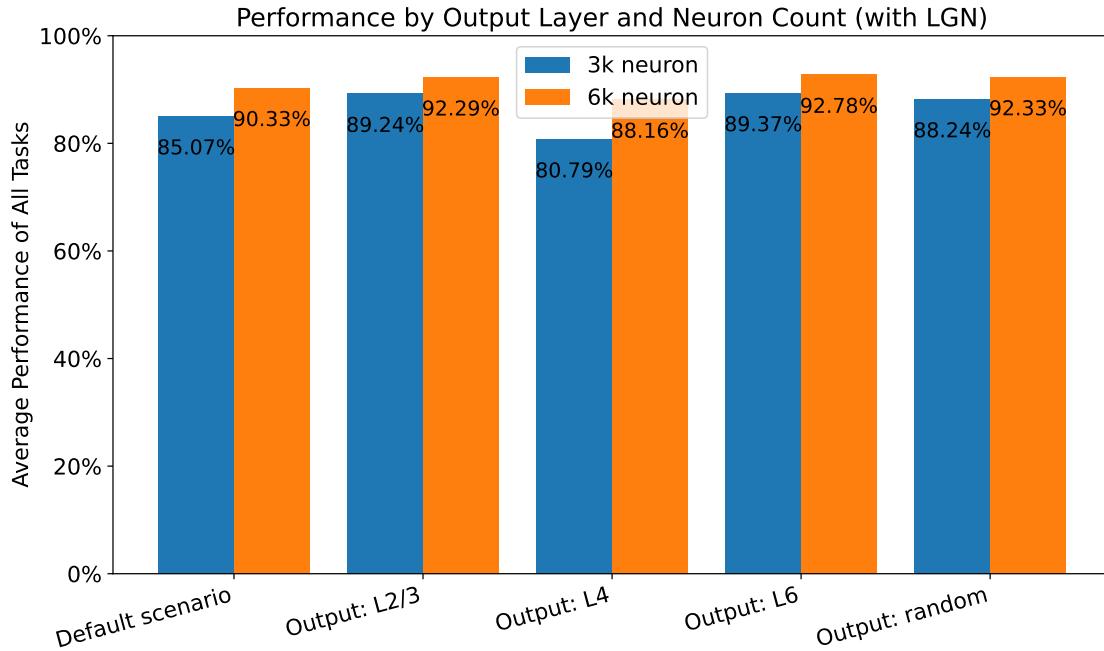


Figure 3.16 Average performance of all tasks by output layer and neuron count (with LGN)

vide the necessary capacity and flexibility to overcome the convergence difficulties and improve overall performance.

The hybrid learning approach demonstrates a notable improvement in accuracy when applied to a 6k-neuron model with LGN, particularly in the visual change detection task. This observation suggests that Hebbian learning may have the potential to be even more effective in significantly larger-scale networks. The increased network size, combined with the preprocessing capabilities of the LGN, seems to provide a more suitable environment for the Hebbian learning mechanism to exploit the inherent structure and dynamics of the visual input. This finding highlights the importance of exploring the scalability of biologically inspired learning algorithms, such as Hebbian learning, in the context of large-scale neural networks.

3.4 Comparison of Models with and without LGN

In this section, we compare the results presented in this chapter, which focuses on models incorporating LGN preprocessing, with the results from chapter 2, which covers models without LGN preprocessing. This comparison reveals several interesting insights and conclusions.

New graphical results based on numerical results in previous sections in this chapter and chapter 2 are given in Figure 3.17, Figure 3.18, Figure 3.19 and Figure 3.20 for

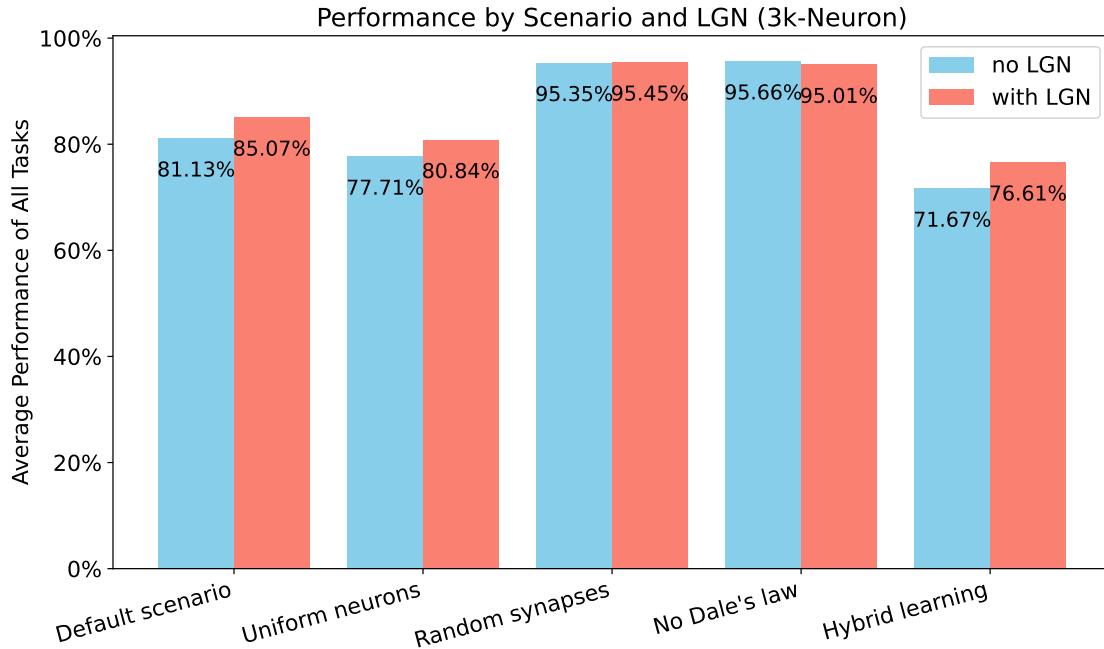


Figure 3.17 Average performance of all tasks by scenario and LGN (3k neurons)

comparison.

Firstly, for both the 3k- and 6k-neuron networks, the incorporation of LGN preprocessing leads to a significant improvement in the average performance across all scenarios, output settings, and hybrid learning. Here take the 3k-neuron network as the example for analysis, and the 6k-neuron network has the similar conclusions. This enhancement is particularly pronounced in the visual change detection task, where the performance sees a substantial boost. Comparing the results in Table 2.3 and Table 3.1, we observe that the accuracy percentages for the default and “uniform neurons” scenarios in the visual change detection task increase by approximately 20% when LGN preprocessing is employed. Similarly, comparing the results in Table 2.4 and Table 3.2, we find that the accuracy percentages across all output settings increase by more than 20%, with the performance of outputting from random neurons experiencing an impressive more than 30% improvement when using LGN-preprocessed data. The LGN also contributes to the improvement of hybrid learning, according to results in Table 2.5 and Table 3.3. But when comparing all the figures of models with and without LGN, it can be observed the introduction of LGN does not significantly help the model to better converge during learning.

However, it is important to note that for both the 3k- and 6k-neuron networks, the performance on the image classification task experiences a decline when LGN preprocessing is used. We hypothesize that the incorporation of LGN filters enhances the model’s

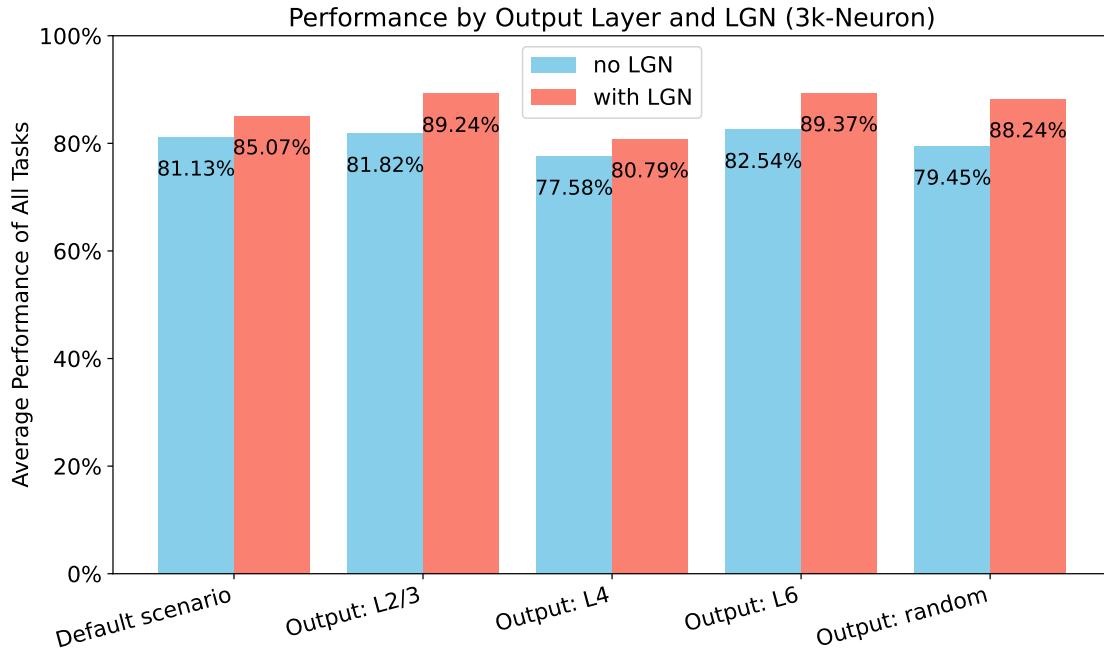


Figure 3.18 Average performance of all tasks by output layer and LGN (3k neurons)

ability to detect temporal differences, but this enhancement comes at the cost of compromising its multi-class classification capabilities. Additionally, the performance on the fine-orientation detection task, which appears to be the least challenging task, slightly decreases in some scenarios, output settings, and hybrid learning when LGN preprocessing is adopted.

Based on these observations, we can conclude that LGN filters play a crucial role in enabling the model to learn and retain historical visual information, a capability that cannot be achieved solely by increasing the network size, as analyzed in section 2.7.

Furthermore, we observe that after incorporating LGN preprocessing, increasing the network scale becomes beneficial for further improving the model's performance. In contrast, without LGN preprocessing, the model fails to exhibit performance gains when its size is increased, as shown in section 2.7 and section 3.3.

These findings highlight the significance of LGN preprocessing in enhancing the model's ability to capture and utilize temporal information, particularly in the context of visual change detection tasks. However, it may be important in some situations to consider the trade-offs associated with LGN preprocessing, such as the potential decrease in performance on other tasks.

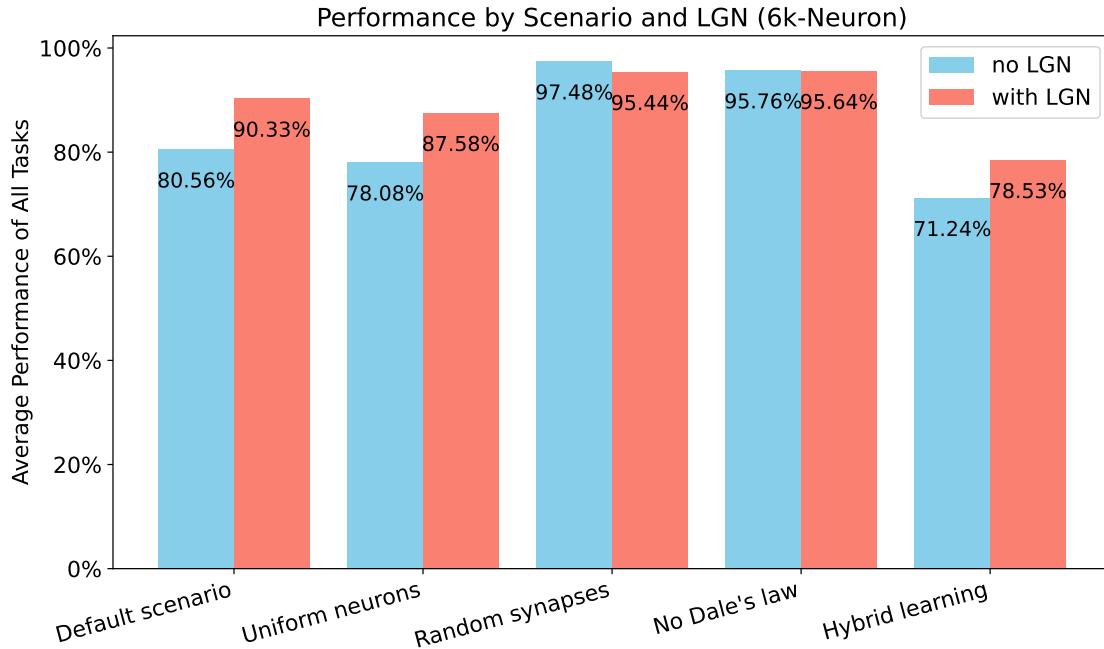


Figure 3.19 Average performance of all tasks by scenario and LGN (6k neurons)

3.5 Chapter Summary

This chapter applies LGN filters to the firing-rate neural network model of the V1 and evaluates their impact on the model’s performance. The key points of this chapter are listed as follows.

1. LGN filters: The chapter introduces LGN filters, which are able to capture the temporal dynamics in receptive fields, and use them to preprocess visual information before it is fed into the V1 model.

2. Learning of models with LGN: The results of training the V1 model with LGN-preprocessed data are presented. The performance of the model is evaluated under different scenarios, output settings, and hybrid learning. The findings show that LGN preprocessing significantly improves the model’s performance, particularly in visual change detection tasks.

3. Comparison of 3k- and 6k-neuron models with LGN: This chapter compares the performance of V1 models with 3,000 neurons and 6,000 neurons when using the LGN filters. The results indicate that increasing the network size, in conjunction with LGN preprocessing, further improves the model’s performance and help the model to converge.

4. Comparison of models with and without LGN: This chapter contrasts the performance of models with LGN preprocessing against those without. The incorporation of

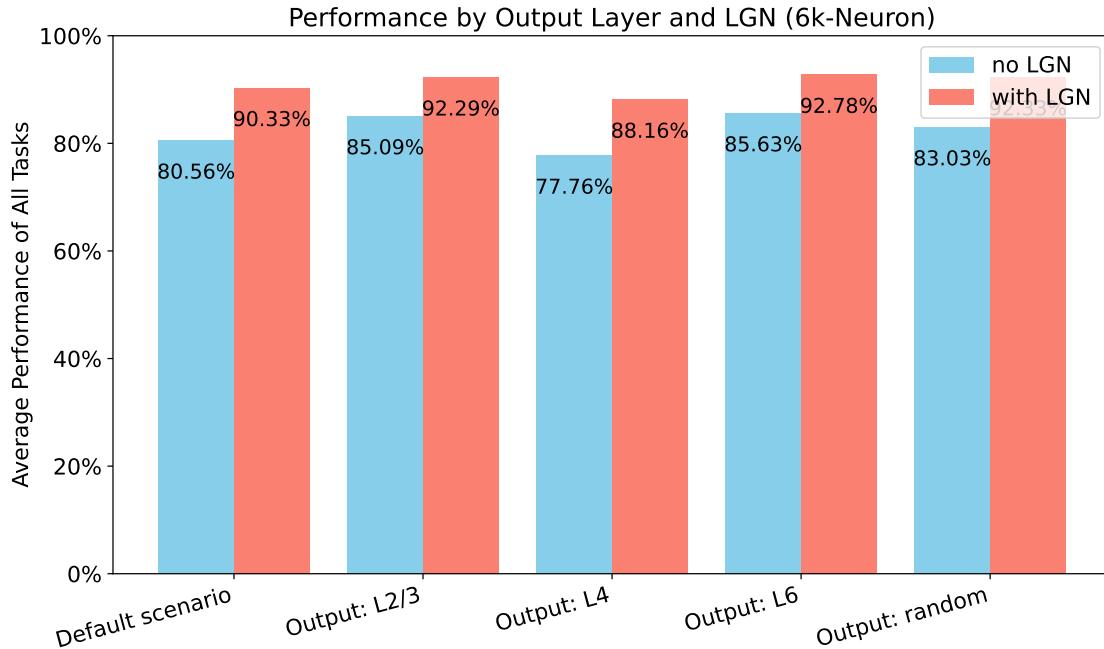


Figure 3.20 Average performance of all tasks by output layer and LGN (6k neurons)

LGN filters is shown to hugely enhance the model's ability to capture and utilize temporal information, which is crucial for visual change detection, while the performance of other tasks are slightly compromised as a trade-off. LGN filters do not significantly contribute to the model convergence, although they improve the performance. LGN filters enhance the hybird learning. These results provide evidence for the importance of LGN's role in early visual processing.

CHAPTER 4 DISCUSSIONS

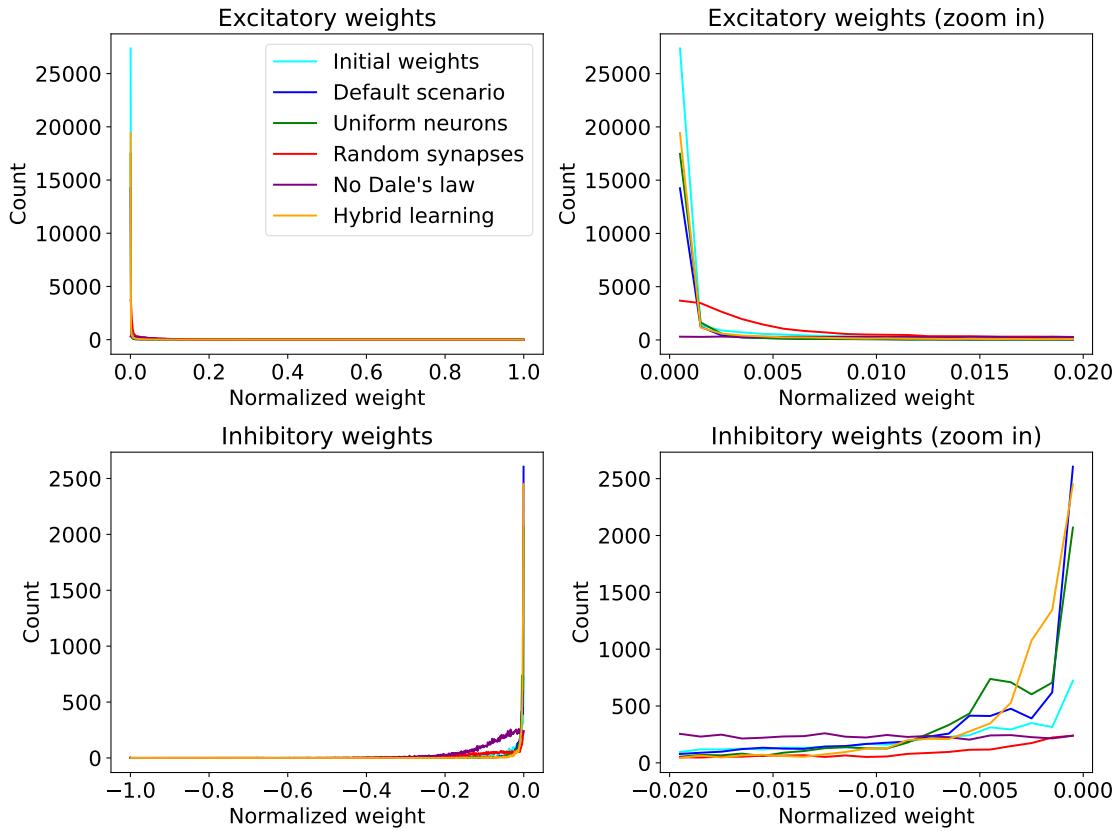
4.1 Weight Distributions

To investigate how synapse weights change after training under the four scenarios (default, “uniform neurons”, “random synapses”, “no Dale’s law”) and hybrid learning, we plot and compare the weight distributions of initial values, multiple scenarios, and hybrid learning. For this analysis, we use the 3k-neuron network models (no LGN) as examples. Figure 4.1 presents the normalized weight distributions, while Figure 4.2 illustrates the log-transformed weight distributions.

The analysis of other models, including the 6k-neuron model, the 3k-neuron model with LGN, and the 6k-neuron model with LGN, yields similar insights. The weight distribution plots for these models can be found in Appendix D.

In Figure 4.1, we observe that both excitatory and inhibitory weights are densely concentrated near zero, with a small portion dispersed away from zero. The zoomed-in subfigures reveal that the scenarios with the best performance, such as the “random synapses” and “no Dale’s law” scenarios, tend to have more weights spread further away from zero, as evidenced by their flatter weight distributions near zero compared to other curves. It is notable that the “no Dale’s law” scenario has a smaller number of excitatory weights compared to others because it can change the weight sign in the learning phase, which means there are more inhibitory weights, as shown in the two subfigures at the bottom in Figure 4.1. The remaining curves, including those of initial values, the default scenario, the “uniform neurons” scenario, and hybrid learning, exhibit steeper slopes near zero, indicating a higher concentration of weights close to zero and fewer large weights. Here, large weights mean more positive values for excitatory weights and more negative values for inhibitory weights.

This observation is further supported by Figure 4.2, which presents the log-transformed normalized weight distributions. The log-transformation allows for a clearer visualization of the weight distribution away from zero by mapping the near-zero weights to a large range that accounts for only a small portion of the x-axis after the transformation. In the log-transformed distribution, the “no Dale’s law” scenario exhibits the highest proportion of large inhibitory weights, which may play a crucial role in maintaining good dynamics for effective learning. In the “no Dale’s law” scenario, the model

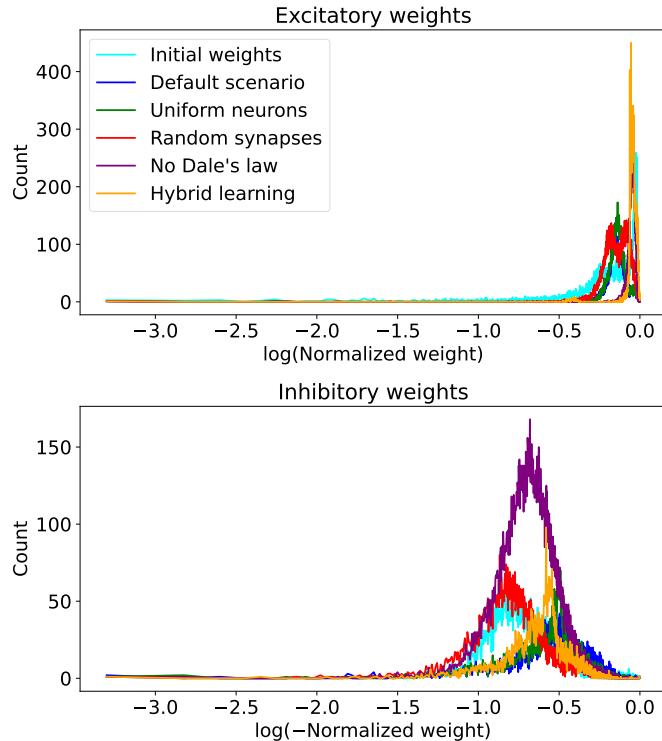


To facilitate plotting and comparison, for each model, the excitatory weights are normalized to the range [0, 1], while the inhibitory weights are normalized to the range [-1, 0]. The two subfigures on the left provide a comprehensive overview of the weight distributions for both excitatory and inhibitory neurons. These subfigures showcase the entire range of normalized weights, allowing for a broad understanding of the distribution shapes. To offer a more detailed examination of the weight distributions within the most densely populated regions, the two subfigures on the right present zoomed-in versions. For excitatory weights, the zoomed-in subfigure focuses on the interval [0, 0.02], while for inhibitory weights, it focuses on the interval [-0.02, 0].

Figure 4.1 Weight distributions of initial values, multiple scenarios, and the hybrid learning (3k neurons)

then can have whatever ratio of positive weights to negative weights. The model in the “random synapses” scenario, on the other hand, has a fixed number of excitatory weights, but it makes more large excitatory weights. Both the “random synapses” and “no Dale’s law” scenarios have more large weights: the former has more large excitatory weights, while the latter has more large inhibitory weights. It is worth noting that the log-normal distribution of weights can be interpreted as an indication of biological realism.

Interestingly, hybrid learning’s large excitatory weights are with values very close to 1. This observation may be a results of its difficulty in achieving convergence. These extreme large excitatory weights may have potentially led to instability and hindered convergence during training. Scenarios with a smaller proportion of extreme large excitatory



A logarithmic transformation is applied to the original normalized weights to better visualize and compare the normalized weights that are close to 1 or -1. In the log-transformed space, a value of 0 corresponds to a normalized weight of 1 for excitatory weights and -1 for inhibitory weights. Negative values in the log-transformed space indicate that the original normalized weights lie between 0 and 1 for excitatory weights and between -1 and 0 for inhibitory weights. The more negative the log-transformed value, the closer the original normalized weight is to 0.

Figure 4.2 Log-transformed weight distributions of initial values, multiple scenarios, and the hybrid learning (3k neurons)

weights may be more resistant to such instabilities, resulting in a smoother and more reliable convergence process.

Therefore, on the one hand, large weights play a crucial role in maintaining effective learning dynamics in neural networks, enabling the model to capture complex patterns and achieve good performance. On the other hand, excessively large weights can also lead to instability during the learning process, potentially hindering convergence. Striking the right balance between the benefits and risks associated with large weights is essential for optimal model training.

The analysis of weight distributions provides valuable insights into the characteristics of different scenarios and learning approaches. The presence of large weights, particularly in the best-performing scenarios, suggests their importance in maintaining good network dynamics and facilitating effective learning. The log-transformed distributions further verify the differences in weight distributions across scenarios, emphasizing the roles of

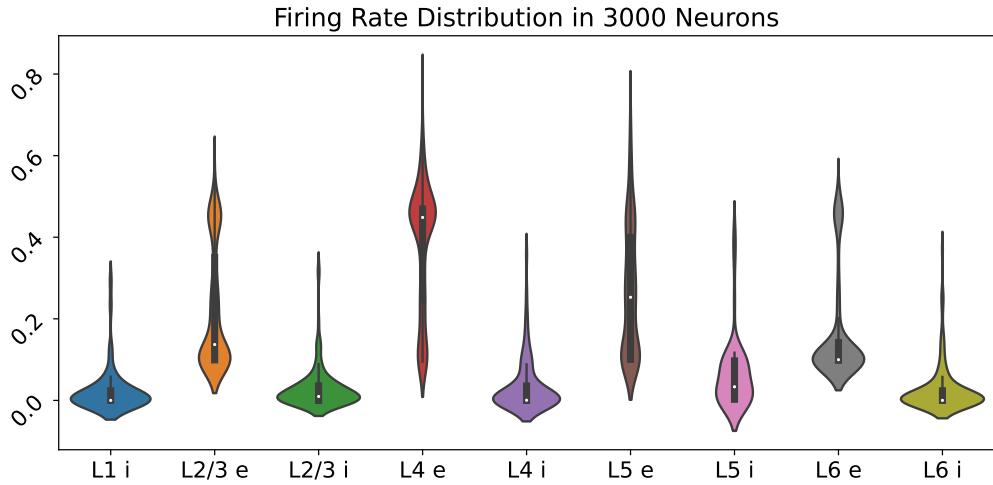


Figure 4.3 Firing rate distribution of all excitatory and inhibitory neuron classes in different layers (3k neurons)

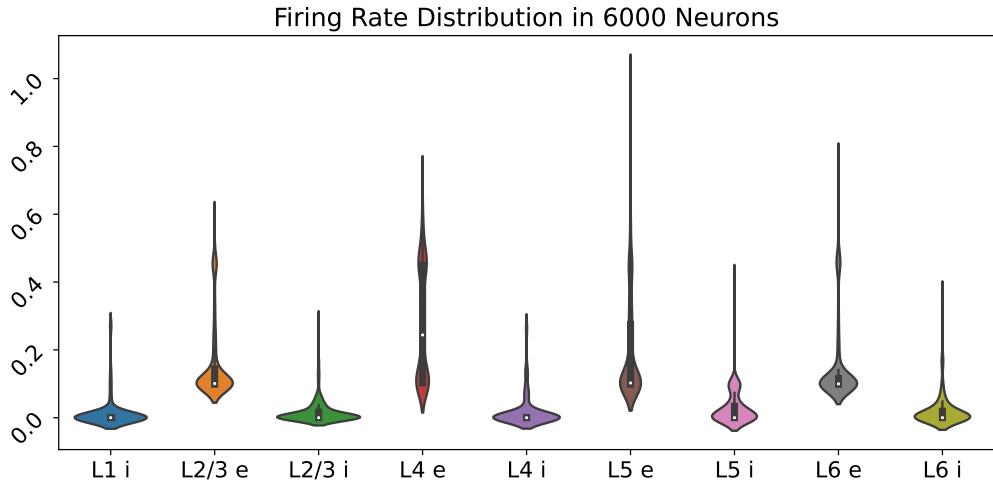


Figure 4.4 Firing rate distribution of all excitatory and inhibitory neuron classes in different layers (6k neurons)

large excitatory and inhibitory weights in shaping network behavior and performance.

Weight distribution of neuronal class pairs across layers is supplemented in Appendix E.

4.2 Firing Rates

To gain insights into the activity patterns of the neuronal populations in our multi-layer model, we analyzed the firing rate distributions of different neuronal classes in the last millisecond, i.e., the time step for output, of the 400-ms simulation. The firing rate distributions are visualized using violin plots to provide a comprehensive representation of the data distribution.

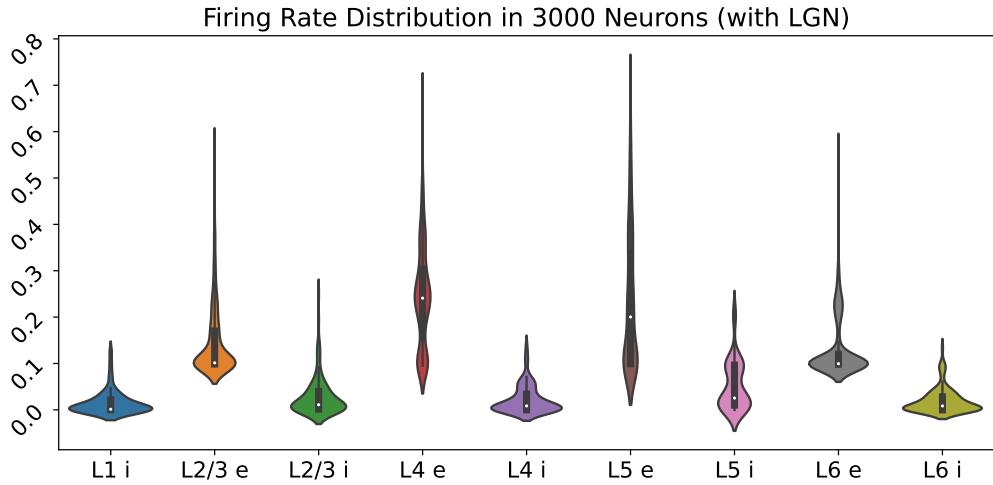


Figure 4.5 Firing rate distribution of all excitatory and inhibitory neuron classes in different layers (3k neurons, with LGN)

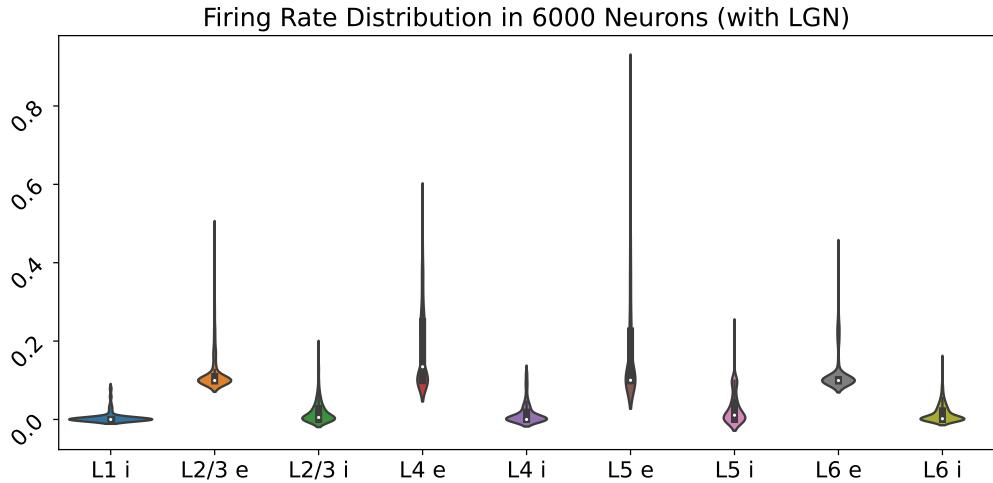


Figure 4.6 Firing rate distribution of all excitatory and inhibitory neuron classes in different layers (6k neurons, with LGN)

We compared the firing rate distributions for models with 3k- and 6k-neurons, both with and without LGN. The neuronal classes considered in this analysis include inhibitory and excitatory populations in all layers. All the results presented here correspond to the default scenario.

Effect of LGN: Comparing the models with and without LGN, the presence of LGN appears to have a concentrating effect on the firing rate distributions. In both the 3k-neuron and 6k-neuron models, the firing rate distributions become more compact and less dispersed when LGN is included. This is particularly noticeable in the excitatory populations of layers L2/3, L4, and L5. The narrower distribution suggests that LGN may play a role in regulating and stabilizing the firing rates of these populations, reducing the

variability in their activity patterns. Furthermore, it can be observed that the median firing rates and overall distributions are shifted downwards when LGN is present.

Impact of neuronal population size: Increasing the neuronal population size has a similar reduction and concentration effect on the firing rates of excitatory cells. In the 6k-neuron model, the firing rates of the excitatory populations tend to be lower and more compact compared to their counterparts in the 3k-neuron model.

One possible explanation for this phenomenon is that the increased number of neurons in the 6k-neuron model leads to more extensive and complex connectivity patterns. The larger population size may result in increased inhibitory feedback or competition among excitatory neurons, effectively suppressing and centralizing their firing rates. This could be a mechanism for maintaining balance and preventing excessive excitation in the larger neuronal network. The change in firing rate distribution can be considered as an indication that both LGN and larger population size help the model to learn more effectively.

4.3 Internal Noise

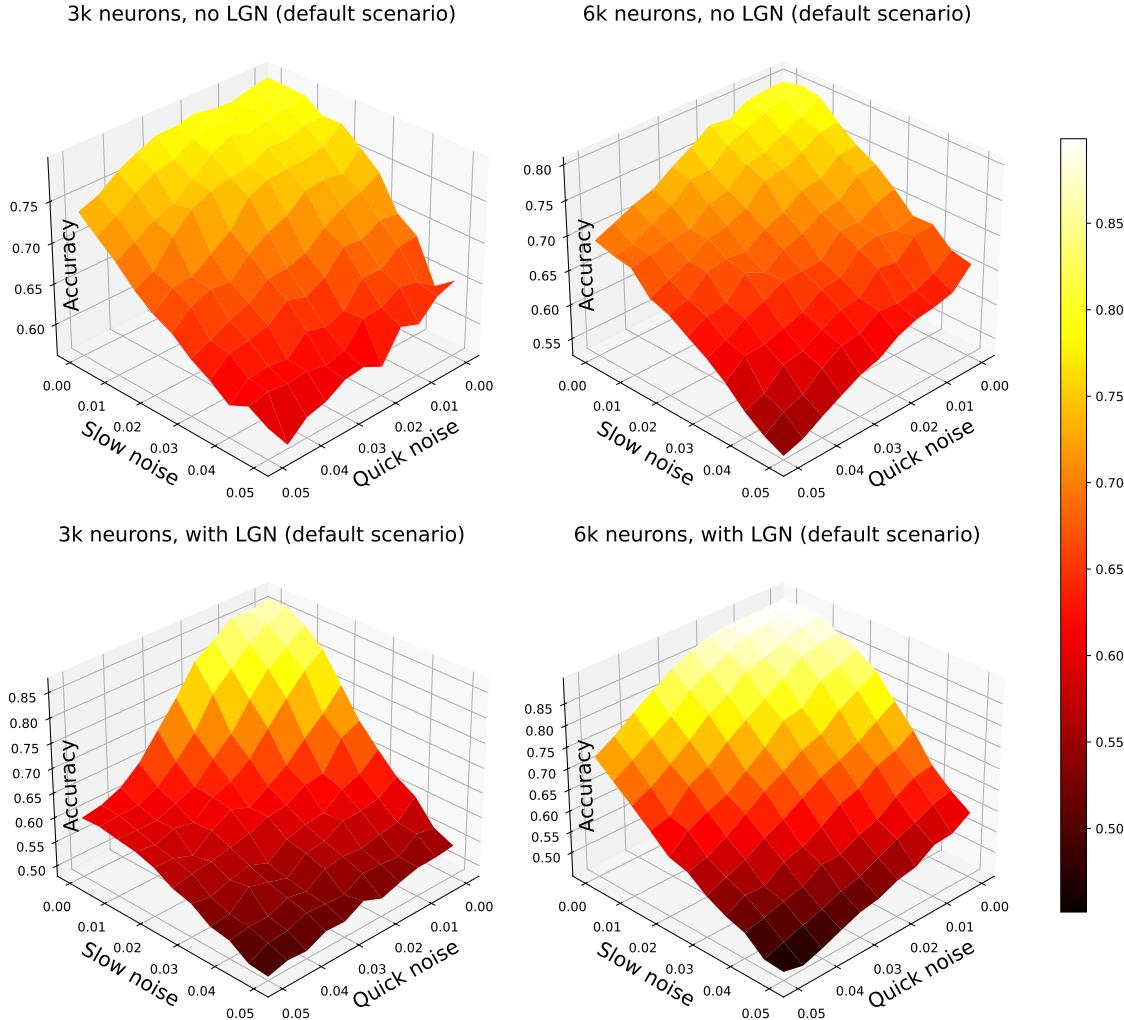
In this analysis, we investigate the impact of internal noise on the performance of models under the four scenario (default, “uniform neurons”, “random synapses”, “no Dale’s law”) and hybrid learning.

Two types of internal noise are considered: slow noise and quick noise. Both noise types are drawn from the same long-tailed distribution to mimic biological features^[77]. Slow noise is identical for all neurons at a given time point, while quick noise varies for each neuron. During simulation, quick noise is drawn n times at each time point, where n is the number of neurons, while slow noise is drawn only once per time point.

We tested the effect of internal noise on the four different network models: the 3k-neuron network, the 6k-neuron network, the 3k-neuron network with LGN, and the 6k-neuron network with LGN. The results are presented in Figure 4.7, Figure 4.8, Figure 4.9, Figure 4.10 and Figure 4.11.

First, we analyze the default scenario.

For the 3k-neuron network model (upper left corner subfigure in Figure 4.7), the accuracy drops from slightly above 80% to roughly 60% as both slow noise and quick noise increase from 0 to 0.05. This demonstrates that the model is quite robust to internal noise, because the positive noise is widely added to all neurons without considering a neuron’s state. Here, slow noise has a more significant negative impact on the model’s performance

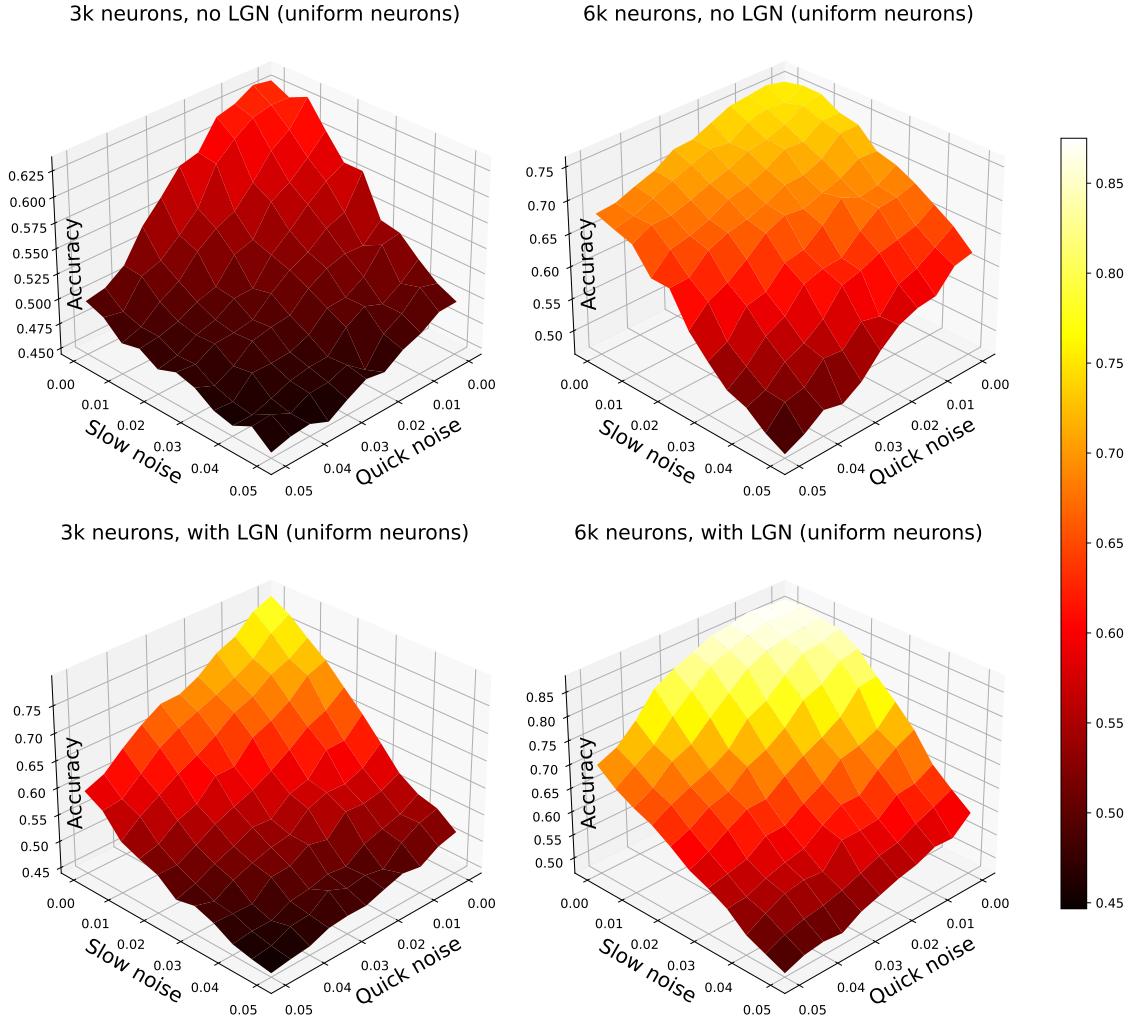


The average accuracy of three tasks are used to evaluate the performance of the model when the slow noise and quick noise change. Both slow noise and quick noise are varied from 0 to 0.05. The value of noise, e.g., 0.02, means the ratio of the mean of the noise distribution to the neuron's maximum rate.

Figure 4.7 Impact of slow and quick noise on model performance under the default scenario

compared to quick noise. This can be observed by comparing the performance decrease when slow noise is 0 and quick noise changes, to the performance decrease when quick noise is 0 and slow noise changes. The model's performance is much better when slow noise is 0 and quick noise is 0.05, compared to when slow noise is 0.05 and quick noise is 0. This observation may be because slow noise affects all neurons uniformly, causing a more substantial impact on network dynamics, while the effects of quick noise on different neurons may tend to cancel out, reducing its negative influence. But this bias is not significant in other plots, so it can also be an coincidence.

In the case of the 6k-neuron network model (upper right corner subfigure in Figure 4.7), the accuracy drops from slightly above 80% to roughly 55% as both noise types

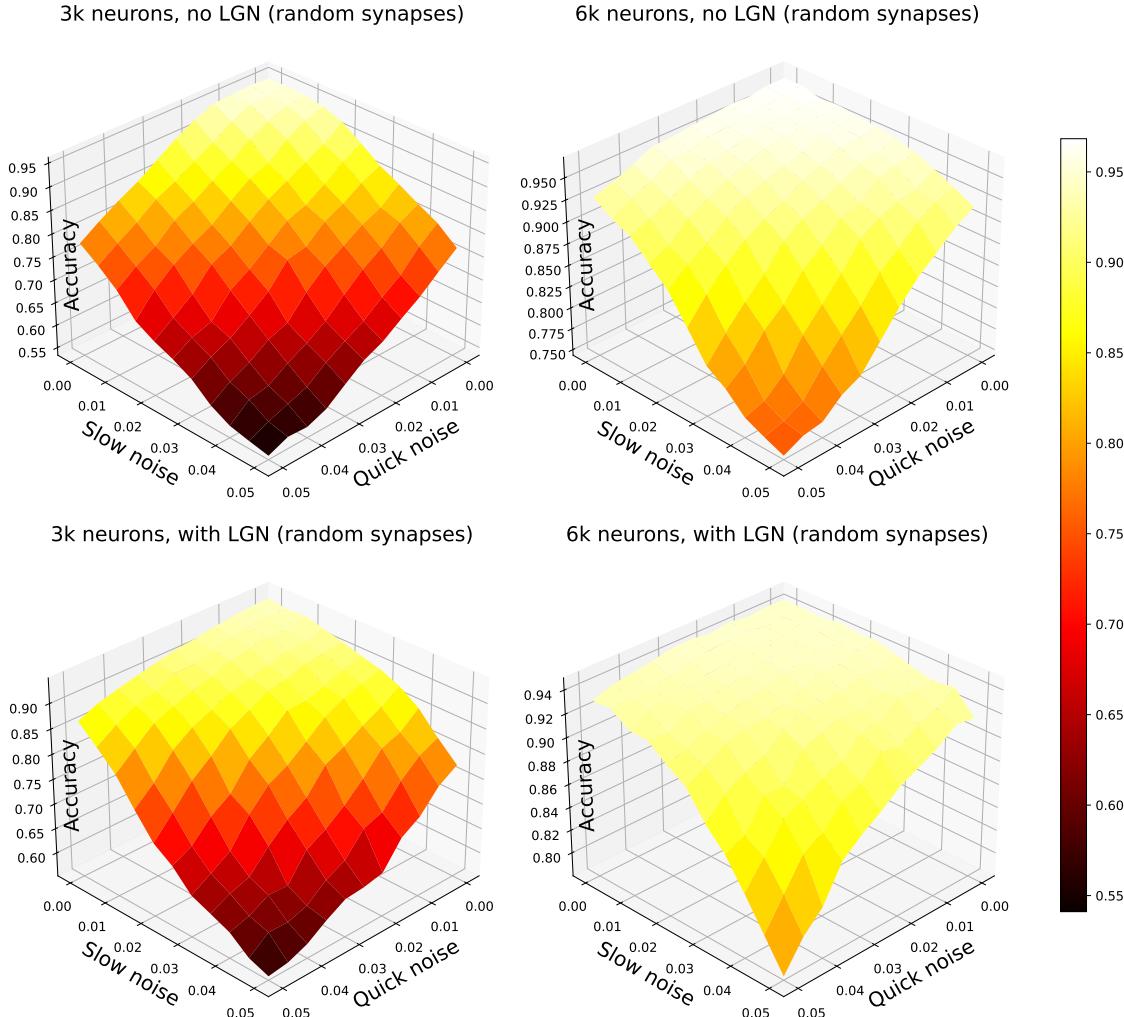


The layout and labeling are same as the description in Figure 4.7

Figure 4.8 Impact of slow and quick noise on model performance under the “uniform neurons” scenario

increase from 0 to 0.05. However, the bias between quick noise and slow noise is reduced, as evidenced by the more symmetric surface compared to the upper left corner subfigure in Figure 4.7. The surface exhibits a saddle point in the middle, suggesting that the model’s performance undergoes a quick-slow-quick decrease as the slow and quick noise levels increase. This indicates that the bigger network size does not increase the robustness when noise is too big but can increase the accuracy when the slow noise and quick noise are relatively small.

For the 3k-neuron network model with LGN (lower left corner subfigure in Figure 4.7), the accuracy drops from slightly above 85% to roughly 50% as both noise types increase from 0 to 0.05. Although the performance is higher than the model without LGN (upper left corner subfigure in Figure 4.7) when both noise types are zero, there is an ac-

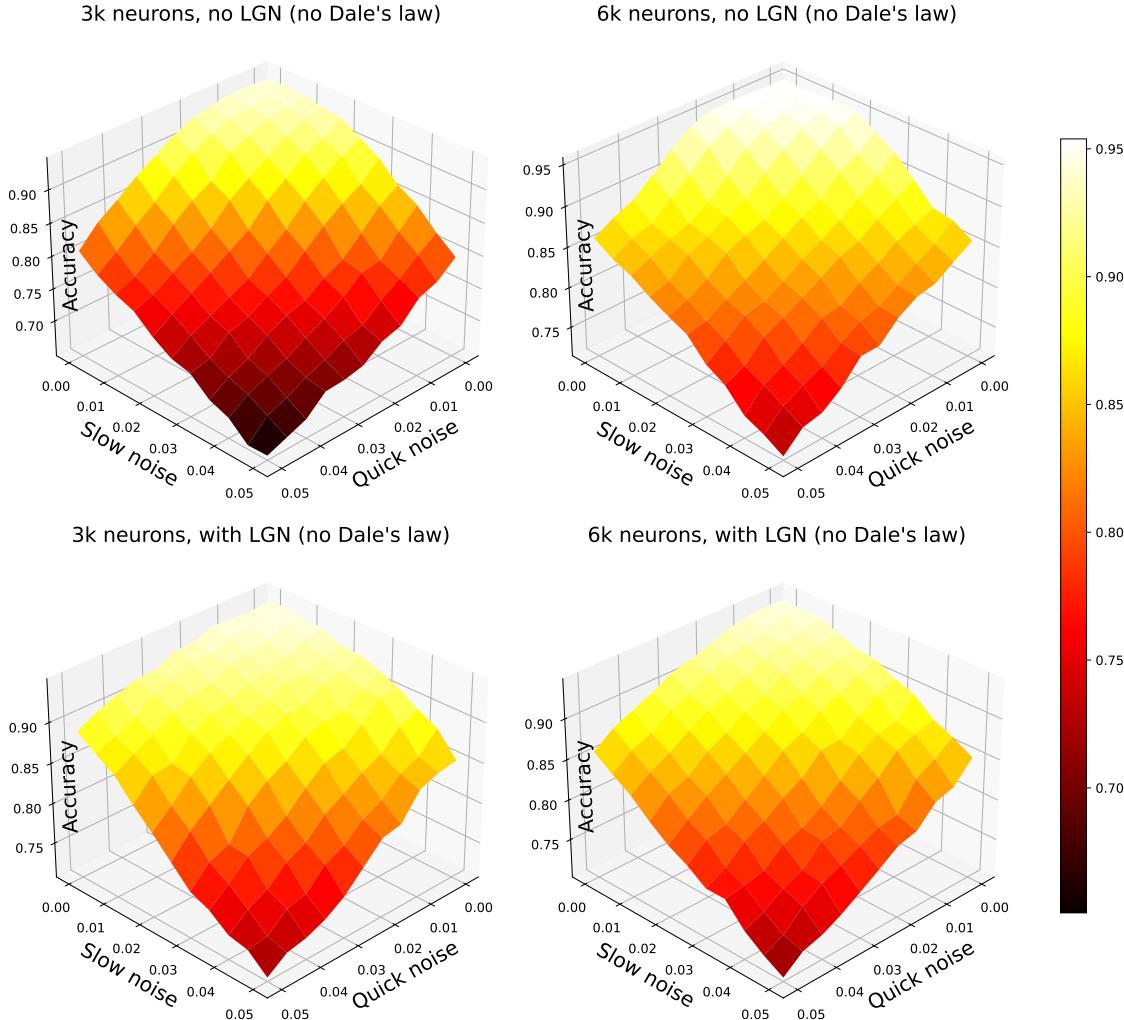


The layout and labeling are same as the description in Figure 4.7

Figure 4.9 Impact of slow and quick noise on model performance under the “random synapses” scenario

celerating drop in performance as the noise levels begin to increase. This indicates that the LGN filters do not enhance the robustness of the model for big noise and its ability of maintaining the accuracy as noise goes larger is weaker than the bigger network size, because the decreasing is faster. The reason may be that the LGN filters serve as an external factor to improve the performance and the noise is added directly to the network, affecting the neuronal states, while the model size is an intrinsic property.

Similarly, for the 6k-neuron network model with LGN (lower right corner subfigure in Figure 4.7), the accuracy drops from slightly above 90% to roughly 50% as both types of noise increase from 0 to 0.05. Despite the higher performance compared to the model without LGN (upper right corner subfigure in Figure 4.7) when both noise types are zero, there is a faster drop in performance when either noise type exceeds 0.02. This further

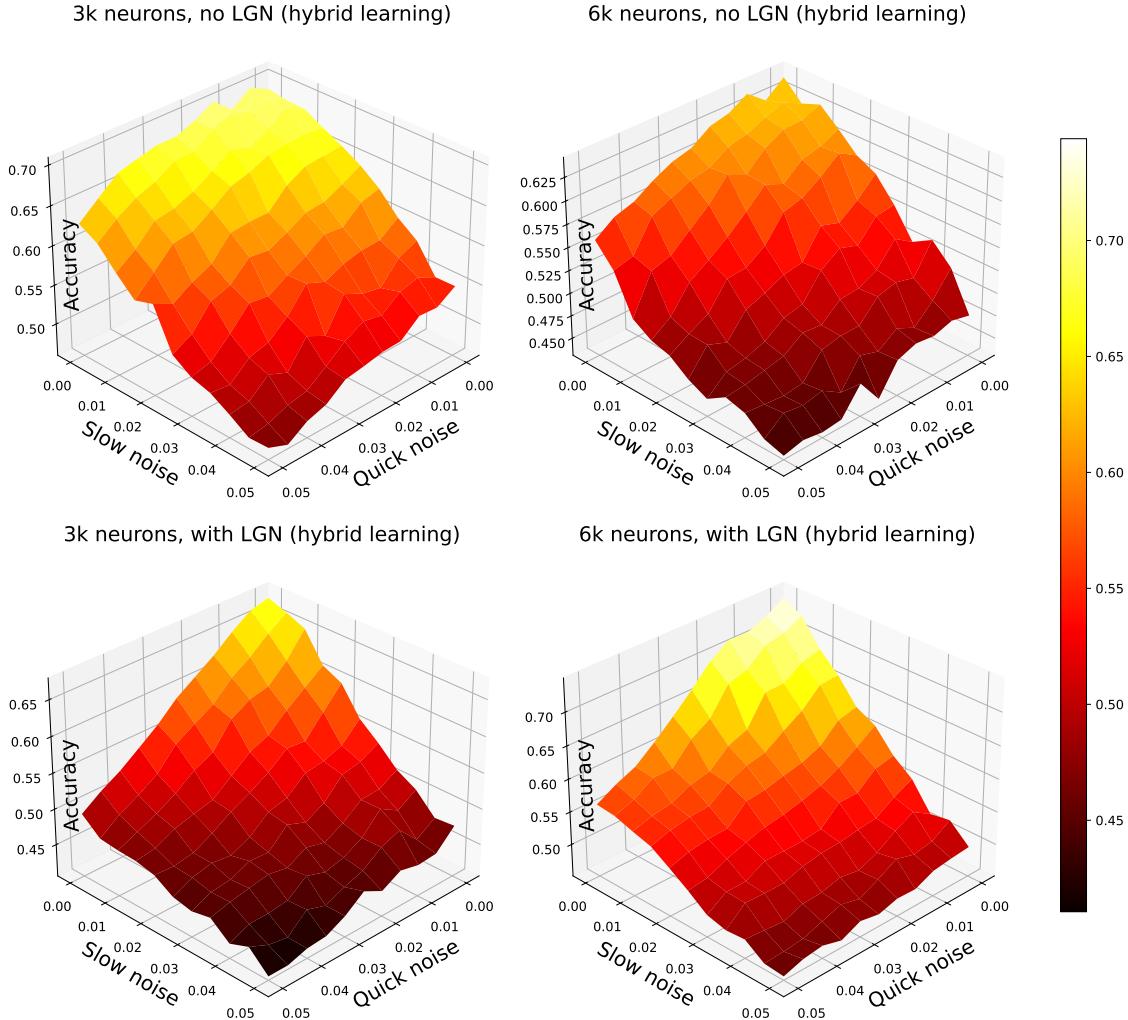


The layout and labeling are same as the description in Figure 4.7

Figure 4.10 Impact of slow and quick noise on model performance under the “no Dale’s law” scenario

verifies that the capability of the LGN filters to enhance the robustness of the model is limited. If comparing with the 3k-neuron network model with LGN (lower left corner subfigure in Figure 4.7), it shows the network size has a better ability to help the model be robust, even though the accuracy drops to the same level when slow noise and quick noise both reach to 0.05.

For results in the “uniform neurons”, “no Dale’s law” scenarios and hybrid learning (Figure 4.8, Figure 4.10, and Figure 4.11), we can have nearly the same conclusion. The LGN filters have a limited impact on enhancing the model’s robustness compared to increasing the network size. However, when the noise level becomes sufficiently high, the model’s performance converges to the same level, regardless of the presence of LGN filters or a larger network size. However, in the “random synapses” scenario, the larger



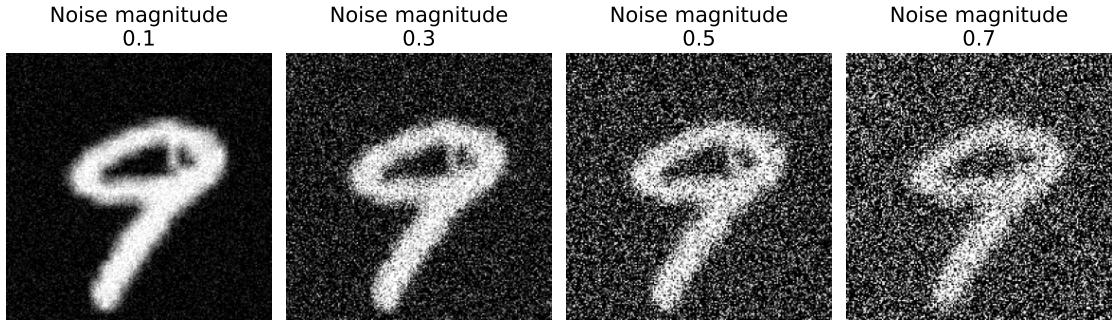
The layout and labeling are same as the description in Figure 4.7

Figure 4.11 Impact of slow and quick noise on model performance under the hybrid learning network size can significantly enhance the robustness of the model, as shown in Figure 4.9. This indicates that a more recurrent network is much stronger when facing internal noise.

4.4 External Noise

In this section, we investigate the performance of models when exposed to external noisy input, specifically focusing on movies consisting of noisy images. This test is conducted only for the image classification task, as adding noise to drifting gratings in the fine-orientation detection task is not a traditional operation, and the stimuli of the visual change detection task are not sensitive to noise. We introduce Gaussian noise to the images from the MNIST dataset to simulate noisy input.

The noisy input is tested under the four scenarios (default, “uniform neurons”, “ran-



Four images are derived from the same original MNIST test image, with Gaussian noise of increasing magnitude added to each one. The noise magnitudes are 0.1, 0.3, 0.5, and 0.7, corresponding to the levels used in the accuracy curve analysis in Figure 4.13, Figure 4.14, Figure 4.15, Figure 4.16, and Figure 4.17.

Figure 4.12 Examples of noisy MNIST images used to evaluate the model

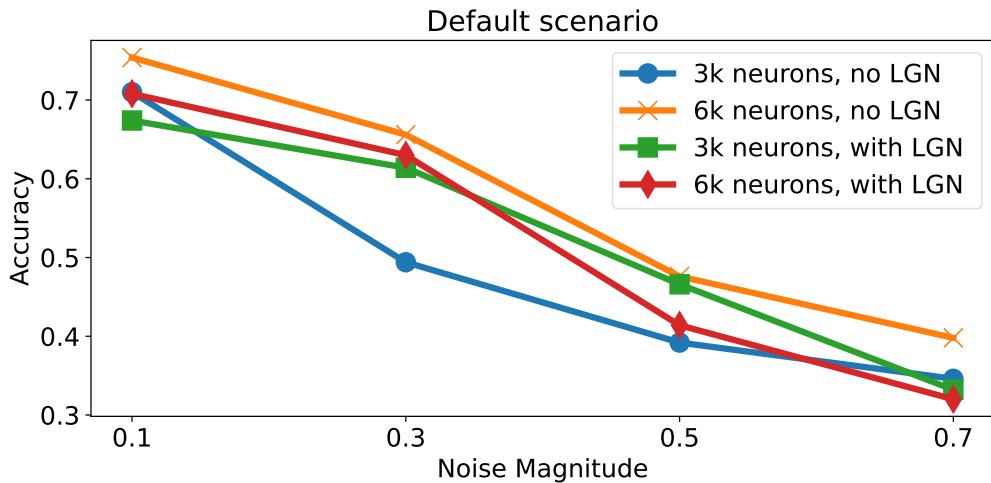


Figure 4.13 Classification accuracy of the model on noisy MNIST images under the default scenario

dom synapses”, “no Dale’s law”) and hybrid learning for four different network models: the 3k-neuron network model, the 6k-neuron network model, the 3k-neuron network model with LGN, and the 6k-neuron network model with LGN. Figure 4.12 presents examples of noisy images at four different noise levels, where the noise magnitude represents the standard deviation of the Gaussian noise added to a grayscale image with values ranging from 0 to 1.

Figure 4.13, Figure 4.14, Figure 4.15, Figure 4.16, and Figure 4.17 illustrate the impact of increasing noise levels in the MNIST images on the accuracy of image classification. From section 3.4, we know that LGN preprocessing does not improve accuracy for the image classification task. Consequently, the initial accuracy values for the four curves in all figures are similar. Take the default scenario as the example. As the noise

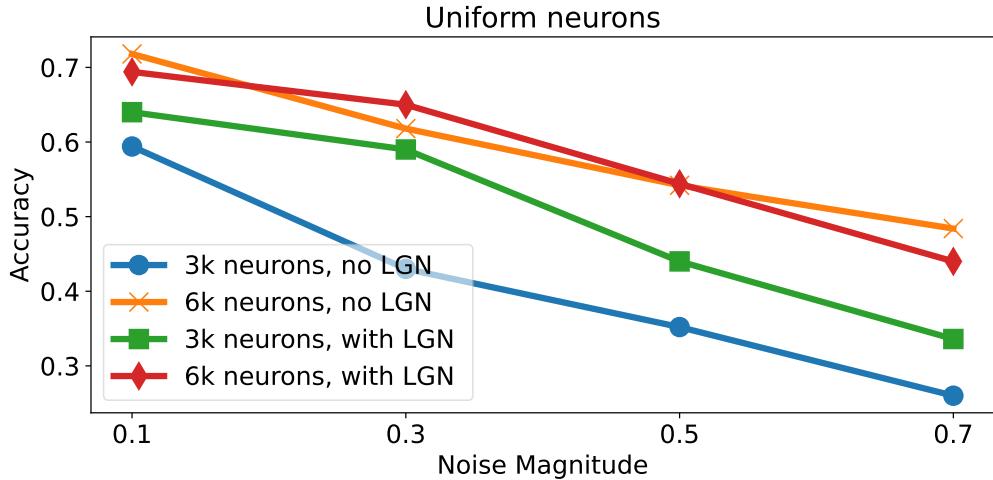


Figure 4.14 Classification accuracy of the model on noisy MNIST images under the “uniform neurons” scenario

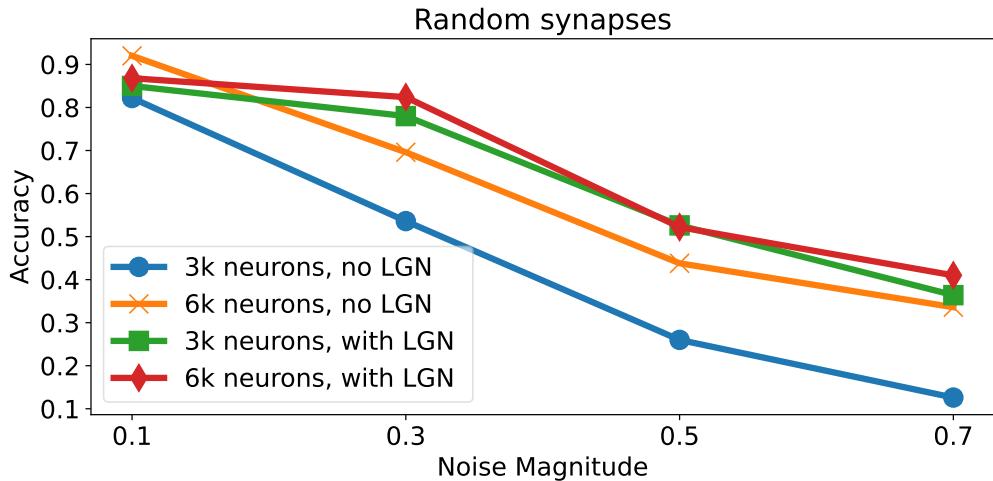


Figure 4.15 Classification accuracy of the model on noisy MNIST images under the “random synapses” scenario

magnitude increases, the reduction tendencies of the four curves follow a similar pattern. When the noise magnitude reaches 0.7, the accuracy drops from approximately 70% to 40%, which is still significantly higher than the chance level accuracy of 10%. So, the models exhibit a certain level of robustness to noisy input.

Comparing the performance of models with and without LGN preprocessing, we observe that LGN does not provide a significant advantage in handling external noisy input for the image classification task. This suggests that the benefits of LGN preprocessing may be task-specific and more pronounced in scenarios where the input is inherently noisy or the task requires enhanced sensitivity to certain features.

For the “random synapses” scenario, which performs better when facing internal

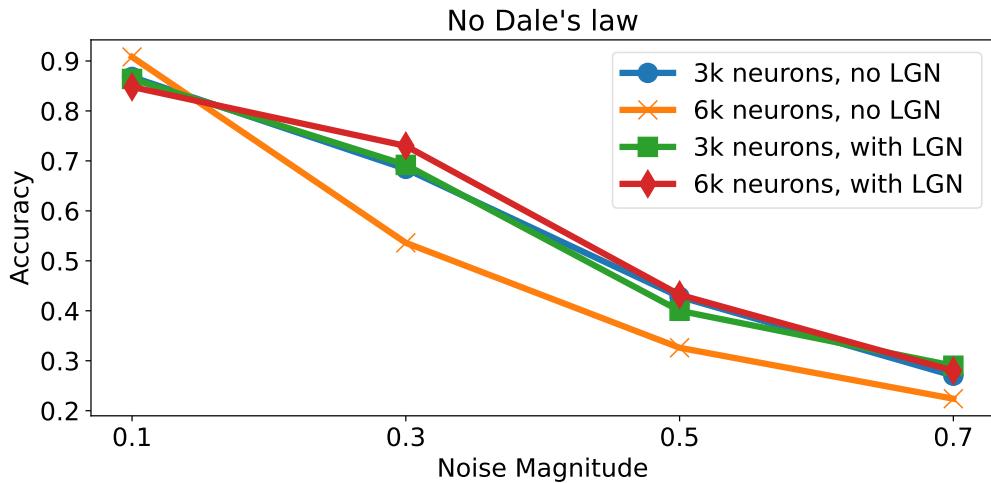


Figure 4.16 Classification accuracy of the model on noisy MNIST images under the “no Dale’s law” scenario

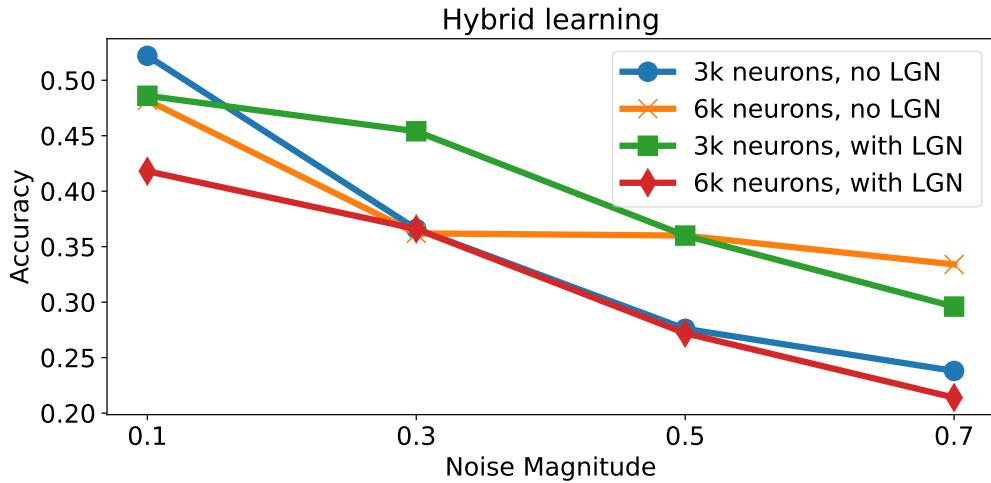


Figure 4.17 Classification accuracy of the model on noisy MNIST images under the hybrid learning

noise, its model doesn’t show significantly better robustness to external noise compared to other models.

4.5 Neuronal Responses

To further analyze the impact of noisy input on model performance, it would be informative to examine the internal representations and activation patterns of the neurons in response to increasing noise levels. This could provide insights into how the models adapt and maintain their discriminative power in the presence of noise.

We performed principal component analysis (PCA) on the neuronal responses to both clean and noisy input images from the MNIST dataset, corresponding to the image clas-

sification task. The analysis was conducted on a randomly selected subset of 50 neurons from the network, which are same for all analysis.

First, we presented 500 test images to the network and recorded the neuronal responses, resulting in a 500 by 50 matrix of neuron states. PCA was then applied to this matrix to identify the principal components that capture the most variance in the neuronal responses. To further explore the network's robustness to input noise, we repeated the analysis using noisy versions of the same 500 test images. Gaussian noise with five different levels of magnitude (0.1, 0.3, 0.5, and 0.7) was added to the original grayscale images. This process yielded a total of 2,000 noisy images (500 images \times 4 noise levels). The neuronal responses to these noisy inputs were recorded, resulting in a 2,500 by 50 matrix of neuron states. PCA was then applied to this matrix, and the explanatory ratios of the principal components were plotted in the same manner as described above. Moreover, we did the PCA for a 2,500 by 50 difference matrix, which is the difference by subtracting the neuronal responses to original, clear images from the neuronal responses to noisy inputs, and plotted the figures also in the same manner. The graphical results are presented in Figure 4.18. Due to the space limit, only the results default scenario are included here. Other results are provided in Appendix F. For all the PCA plots, we fit a exponential function $y = ae^{bx}$, as show in the figures.

According the fit function given in Figure 4.18 and those in Appendix F, we can observe that the bar plots of principal components well follow the power law, which is a good balance between high-precision and robustness and can be interpreted as an indication of biological realism. Comparing the exponents in subfigures of responses to original inputs and noisy inputs and response difference, we can find a clear decreasing tendency of the exponents from the original to the noisy to the difference. This indicates that the noise slightly reduces the ratio of explained variance of top principal components. The smallest exponents of the difference response PCA in most cases verify that the noise components are flatter so that their negative effect on the feature representation is reduced, which accounts for the model's robustness. The significance of the relationship between the exponents are tested in Figure 4.19, where the exponents of all models (the 3k-neuron network, the 6k-neuron network, the 3k-neuron network with LGN, and the 6k-neuron network with LGN) under the four scenarios (default, “uniform neurons”, “random synapses”, “no Dale’s law”) and hybrid learning are used to do the test, showing the highest level of significance and clear tendency.

We also conducted significance for the dot product between the principal components from neuronal responses to original and noisy inputs to show their difference from random dot product values and demonstrate the principal components of responses to noisy inputs keep very similar directions to the ones of responses to original inputs, as shown in Figure 4.20 and the similar figures in Appendix F. Most have good significance with the best-performing scenarios “random synapses” and “no Dale’s law” showing better significance.

4.6 Comparison with SNNs and ANNs

The performance of SNNs and ANNs on tasks like MNIST classification have been done by other works^[77,148]. SNNs typically have more variables per neuron compared to firing-rate models, requiring significantly more computational resources and GPU memory. For instance, the research by Chen et al.^[77] utilized 160 pieces of Nvidia A100 Tensor Core GPUs to simulate an SNN with about 50,000 neurons. Our focus is on investigating the performance and properties of firing-rate models, which have the advantage of reduced dependency on computational resources, and what aspects contribute to improvement of various performance of rate models. This difference in computational requirements makes a direct comparison difficult, as we do not have access to the same level of resources.

Comparison with ANNs is also non-trivial due to the inherent differences in network architectures. The brain-like network we modeled is sparse, making it challenging to determine an appropriate ANN architecture for comparison. ANNs have numerous variants with vastly different performances^[149-152], and selecting a suitable candidate for specific tasks is not straightforward. In existing work, for example, Chen et al.^[77] compared their model with some typical ANNs, but the choice of these networks may not be sufficiently convincing.

If comparing with the results of Chen et al. (SNNs one magnitude larger than our rate models and some ANNs; SNNs use a time window to make output but our rate models only use one time point), our models’ performance on the fine-orientation detection is better than SNNs even they are about 10 times larger than ours in scale, demonstrating the power of rate model in capturing nuance in data. For default scenario, our models’ performance on MNIST classification (the image classification task) is worse than SNNs and some ANNs; for random synapses and no Dale’s law scenarios, our performance is

similar to SNNs and ANNs. The robustness of our models to MNIST inputs is between SNNs and ANNs (both feedforward CNNs and recurrent CNNs) (see Figure 6 in the paper of Chen et al.). For the visual change detection task, when LGN is equipped, our model’s performance is comparable with SNNs (also with LGN) and ANNs.

4.7 Simple and Complex Cells

Classical studies of the primary visual cortex have revealed important functional cell types, most notably simple cells and complex cells^[56,153-154]. Simple cells exhibit linear spatial summation, responding maximally to a preferred stimulus orientation. In contrast, complex cells have nonlinear spatial summation and exhibit some degree of spatial invariance in their responses^[155].

While our V1 model does not explicitly designate certain neurons as simple or complex cells, it is likely that these cell types would emerge naturally in the network after training, given the model’s biological plausibility and the large network size. Simple cells are often characterized by their connection to LGN cells^[156], but this is not a necessary condition. Rather, the defining feature of a simple cell is its unimodal response function^[157].

In our model, the incorporation of LGN filters serves to provide biologically realistic input preprocessing, but it does not predetermine the existence of simple cells in the network. The emergence of simple and complex cells would be a result of the learning process and the network’s self-organization based on the visual tasks and input statistics.

Future work could involve analyzing the trained network to identify and characterize the emergent simple and complex cells. This could be done by examining the response properties of individual neurons to various visual stimuli and comparing them to the canonical properties of simple and complex cells described in the literature. Such an analysis would provide insights into how well the model captures these important functional cell types and would further validate its biological plausibility.

The presence of both simple and complex cells in the trained network would demonstrate that the model can self-organize to develop these key functional specializations, even without explicit architectural constraints, thereby highlighting the power of data-driven learning in capturing biologically relevant features.

4.8 Chapter Summary

This chapter delves into various aspects of the research findings, including weight distributions, internal noise, external noise, and neuronal responses. Here's a summary of the key points discussed in this chapter.

1. **Weight distributions:** The chapter examines the distribution of synaptic weights after training under different scenarios and learning approaches. It is observed that scenarios with more widespread weight distributions tend to perform better, suggesting that a diverse range of weights is beneficial for learning complex patterns. However, too extreme weights can reduce the performance of the models. The log-normal distribution of weights shows an indication of biological realism.
2. **Firing rates:** LGN and increased neuronal population size lead to more concentrated and reduced firing rates, particularly in the excitatory populations, suggesting their role in regulating and stabilizing the network's activity and potentially contributing to more effective learning.
3. **Internal noise:** The research investigates the impact of internal slow and quick noise on model performance. The models exhibit robustness to internal noise to some extent. The effectiveness of LGN filters in improving the model's robustness to internal noise is relatively modest compared to the benefits of increasing the network size. A network with higher recurrency demonstrates superior resilience when confronted with internal noise.
4. **External noise:** The chapter explores the model's performance when exposed to external noisy input, specifically focusing on image classification tasks with noisy MNIST images. The models maintain a significant level of accuracy even with high noise magnitudes. The incorporation of LGN preprocessing offers limited benefits in terms of the model's ability to cope with external noise.
5. **Neuronal responses:** PCA is used to analyze the internal representations and activation patterns of neurons in response to external noise. The PCA results show that despite the addition of noise, the principal components capturing the most variance in the responses remain dominant. The results also show that the principal components of the neuronal responses follow a power law distribution, indicating a balance between high-precision and robustness, and suggesting biological realism.

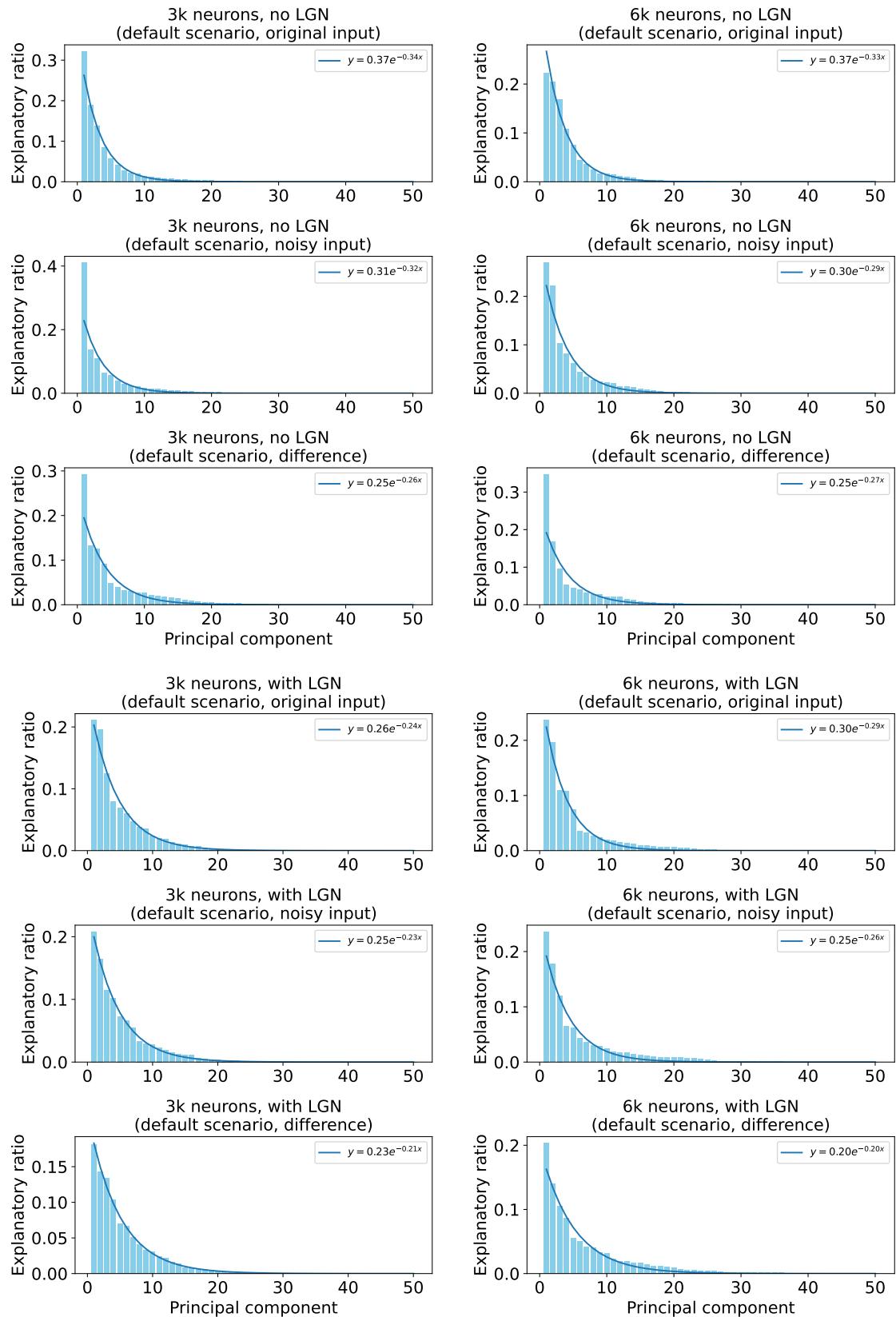
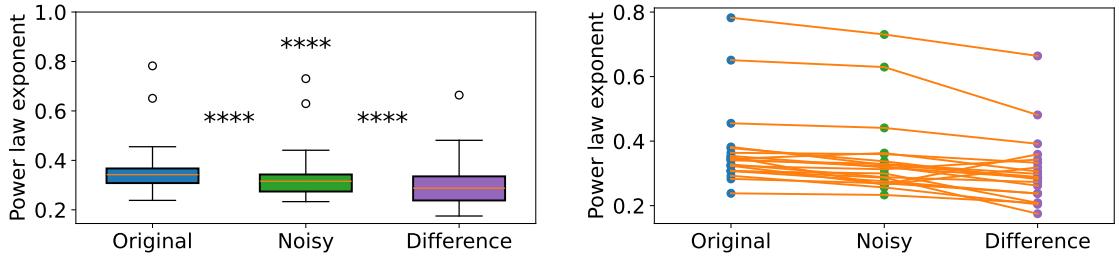


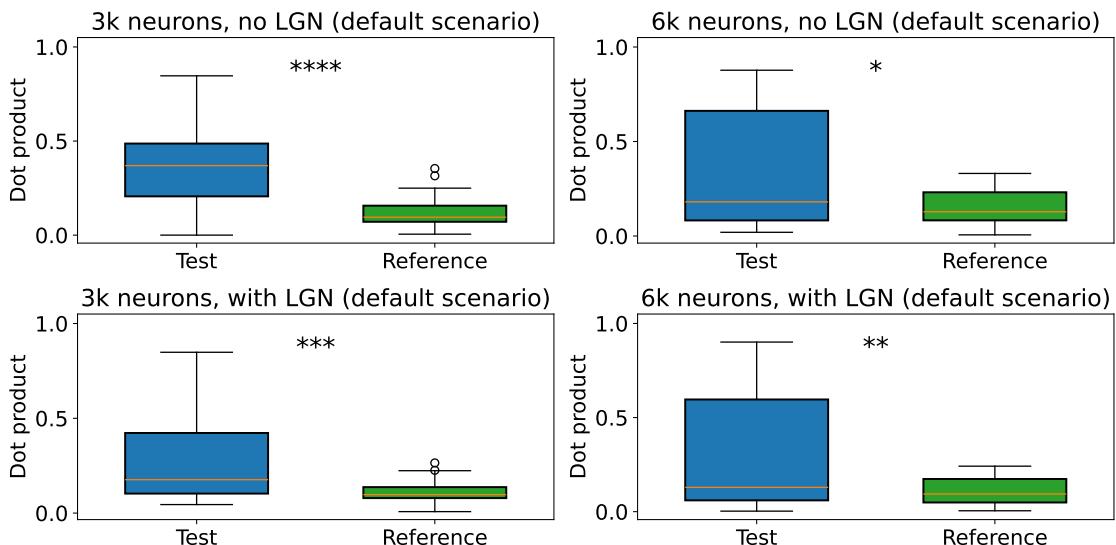
Figure 4.18 PCA results of neuronal responses to clean and noisy MNIST images and of the response difference (default scenario)



(a) T-test analysis is conducted to show the significance of the difference between the fit power law exponents of the curves of neuronal responses to the original input and the noisy input and the response difference. Four asterisks represent the highest level of significance of each pair of exponent groups.

(b) The scatter plot of all exponents used for the T-test analysis shows a clear decreasing tendency from original to noisy then to difference.

Figure 4.19 Significance analysis of power law exponents of three PCA fit curves in all models



The figure presents T-test analysis of a comparison of dot products calculated between principal component vectors derived from neuronal responses to original and noisy inputs. Two sets of dot products were computed for the analysis. (1) Test: between each pair of vectors from the top 5 principal components of original inputs and the top 5 principal components of noisy inputs, resulting in 25 values; and (2) Reference: between each pair of vectors from the top 5 principal components of original inputs and a set of 5 randomly generated orthogonal vectors, also yielding 25 values.

Figure 4.20 Significance analysis of dot product of between top 5 principal components of neuronal response to original and noisy input Significance analysis using dot product between principal components of neuronal responses to original and noisy inputs (default scenario)

CHAPTER 5 CONCLUSIONS

Motivated by the need for computationally efficient and biologically plausible models, this thesis develops a firing-rate neural network model for the V1 and investigates its learning capabilities. The study contributes to the understanding of neural processes underlying visual perception and provides valuable insights into the computational aspects of neural modeling.

The key contributions and findings of this work are as follows.

First, the transition from spiking models to firing-rate models addresses the need for computationally efficient yet biologically plausible models, which cannot be achieved by existing SNNs. The discrete form firing-rate models developed in this thesis provide a framework for large-scale simulation. Equipped with the simulation tool, based on real data from the Allen Institute for Brain Science, the evaluation of the models' learning capabilities through representative visual tasks (fine-orientation detection, image classification, and visual change detection) and various scenarios / settings (such as "uniform neurons," "random synapses," "no Dale's law," different output layers, and hybrid learning) offers a comprehensive analysis of the factors influencing learning effectiveness.

Further, we compare 3k- and 6k-neuron models, and the models with LGN filters, which provides valuable insights into the impact of network size and early visual processing on model performance. The study reveals that enhanced network connectivity and flexibility in synaptic weights significantly contribute to learning effectiveness, while neuronal diversity plays a role in the memory capability of the networks, which aligns with conclusions in some existing literature. The incorporation of LGN significantly enhances the model's ability to capture and utilize temporal information, which is consistent with LGN's function in many other works, improving the performance of visual change detection, although it slightly compromises the performance of other tasks. LGN filters are also shown to enhance hybrid learning, verifying the importance of LGN's role in early visual processing. Interestingly, increasing the network size alone does not lead to significant improvements in performance without input preprocessing from LGN, and LGN filters alone do not significantly contribute to model convergence, but the combination of increased network size and LGN preprocessing can further improve the model's performance and convergence, highlighting the interplay between network scale and input

preprocessing in shaping neural computation, which to the best of our knowledge is not well studied by existing work.

At last, we discuss the biological realism of our models through analysis from multiple aspects. The observed log-normal distribution of synaptic weights after training under different scenarios and learning approaches is consistent with biological findings. Moreover, the study reveals that more widespread weight distributions lead to better performance, indicating that a diverse range of weights is beneficial for learning complex patterns, although extremely positive or negative weights can reduce the model’s performance. In terms of robustness, the model exhibits resilience to both internal and external noise to some extent. Increasing the network size offers more benefits in improving the model’s robustness to internal noise compared to LGN filters, and networks with higher recurrency demonstrate even stronger resilience when confronted with internal noise. This is a new perspective of analyzing robustness. It differs from the traditional focus on the laminar structure of the brain. Additionally, the analysis of the internal representations and activation patterns of neurons in response to external noise reveals that the principal components capturing the most variance in the responses remain dominant and follow a power law distribution, suggesting that the model strikes a balance between achieving high precision in its representations and maintaining robustness to noise. This balance is reminiscent of the trade-offs observed in biological neural systems, also supporting the biological plausibility.

Future work should focus on refining the hybrid learning algorithms, exploring the role of network architecture in more depth, and extending these models to new learning rules, more cognitive tasks, and other brain regions.

REFERENCES

- [1] Izhikevich E M. Which model to use for cortical spiking neurons?[J]. IEEE transactions on neural networks, 2004, 15(5): 1063-1070.
- [2] Gerstner W, Lehmann M, Liakoni V, et al. Eligibility traces and plasticity on behavioral time scales: experimental support of neohebbian three-factor learning rules[J]. Frontiers in neural circuits, 2018, 12: 53.
- [3] Dayan P, Abbott L F. Theoretical neuroscience: computational and mathematical modeling of neural systems[M]. MIT press, 2005.
- [4] Payeur A, Guerguiev J, Zenke F, et al. Burst-dependent synaptic plasticity can coordinate learning in hierarchical circuits[J]. Nature neuroscience, 2021, 24(7): 1010-1019.
- [5] Buzsaki G, Draguhn A. Neuronal oscillations in cortical networks[J]. science, 2004, 304(5679): 1926-1929.
- [6] Gerstner W, Kistler W M, Naud R, et al. Neuronal dynamics: From single neurons to networks and models of cognition[M]. Cambridge University Press, 2014.
- [7] Illing B, Gerstner W, Brea J. Biologically plausible deep learning—but how far can we go with shallow networks?[J]. Neural Networks, 2019, 118: 90-101.
- [8] Markram H, Muller E, Ramaswamy S, et al. Reconstruction and simulation of neocortical microcircuitry[J]. Cell, 2015, 163(2): 456-492.
- [9] Roy K, Jaiswal A, Panda P. Towards spike-based machine intelligence with neuromorphic computing[J]. Nature, 2019, 575(7784): 607-617.
- [10] Tang J, Yuan F, Shen X, et al. Bridging biological and artificial neural networks with emerging neuromorphic devices: fundamentals, progress, and challenges[J]. Advanced Materials, 2019, 31(49): 1902761.
- [11] Maass W. Networks of spiking neurons: the third generation of neural network models[J]. Neural networks, 1997, 10(9): 1659-1671.
- [12] Tavanaei A, Ghodrati M, Kheradpisheh S R, et al. Deep learning in spiking neural networks[J]. Neural networks, 2019, 111: 47-63.
- [13] Pei J, Deng L, Song S, et al. Towards artificial general intelligence with hybrid tianjic chip architecture[J]. Nature, 2019, 572(7767): 106-111.
- [14] Rathi N, Chakraborty I, Kosta A, et al. Exploring neuromorphic computing based on spiking neural networks: Algorithms to hardware[J]. ACM Computing Surveys, 2023, 55(12): 1-49.
- [15] Maass W. How can neuromorphic hardware attain brain-like functional capabilities?[J]. National Science Review, 2024, 11(5): nwad301.
- [16] Abbott L F, DePasquale B, Memmesheimer R M. Building functional networks of spiking model neurons[J]. Nature neuroscience, 2016, 19(3): 350-355.
- [17] Paninski L. Maximum likelihood estimation of cascade point-process neural encoding models [J]. Network: Computation in Neural Systems, 2004, 15(4): 243.

REFERENCES

- [18] Pillow J W, Shlens J, Paninski L, et al. Spatio-temporal correlations and visual signalling in a complete neuronal population[J]. *Nature*, 2008, 454(7207): 995-999.
- [19] Gerstner W, Kistler W M. Spiking neuron models: Single neurons, populations, plasticity[M]. Cambridge university press, 2002.
- [20] Izhikevich E M. Dynamical systems in neuroscience[M]. MIT press, 2007.
- [21] Ostojic S. Two types of asynchronous activity in networks of excitatory and inhibitory spiking neurons[J]. *Nature neuroscience*, 2014, 17(4): 594-600.
- [22] Ermentrout B, Terman D H. Mathematical foundations of neuroscience: volume 35[M]. Springer, 2010.
- [23] Barak O. Recurrent neural networks as versatile tools of neuroscience research[J]. *Current opinion in neurobiology*, 2017, 46: 1-6.
- [24] Ginsberg A. Firing-rate models in computational neuroscience: New applications and methodologies[D]. 2023.
- [25] Liu Z, Han F, Wang Q. A review of computational models for gamma oscillation dynamics: from spiking neurons to neural masses[J]. *Nonlinear Dynamics*, 2022, 108(3): 1849-1866.
- [26] Rumelhart D E, Hinton G E, Williams R J. Learning representations by back-propagating errors [J]. *nature*, 1986, 323(6088): 533-536.
- [27] Hertz J, Krogh A, Palmer R G, et al. Introduction to the theory of neural computation[M]. American Institute of Physics, 1991.
- [28] Lillicrap T P, Cownden D, Tweed D B, et al. Random synaptic feedback weights support error backpropagation for deep learning[J]. *Nature communications*, 2016, 7(1): 13276.
- [29] Shadlen M N, Newsome W T. The variable discharge of cortical neurons: implications for connectivity, computation, and information coding[J]. *Journal of neuroscience*, 1998, 18(10): 3870-3896.
- [30] Renart A, De La Rocha J, Bartho P, et al. The asynchronous state in cortical circuits[J]. *science*, 2010, 327(5965): 587-590.
- [31] Pietras B, Gallice N, Schwalger T. Low-dimensional firing-rate dynamics for populations of renewal-type spiking neurons[J]. *Physical Review E*, 2020, 102(2): 022407.
- [32] Softky W R, Koch C. The highly irregular firing of cortical cells is inconsistent with temporal integration of random epsps[J]. *Journal of neuroscience*, 1993, 13(1): 334-350.
- [33] Moreno-Bote R, Parga N. Response of integrate-and-fire neurons to noisy inputs filtered by synapses with arbitrary timescales: Firing rate and correlations[J]. *Neural Computation*, 2010, 22(6): 1528-1572.
- [34] Schwalger T, Deger M, Gerstner W. Towards a theory of cortical columns: From spiking neurons to interacting neural populations of finite size[J]. *PLoS computational biology*, 2017, 13 (4): e1005507.
- [35] Keeley S L, Zoltowski D M, Aoi M C, et al. Modeling statistical dependencies in multi-region spike train data[J]. *Current opinion in neurobiology*, 2020, 65: 194-202.
- [36] Mainen Z F, Sejnowski T J. Reliability of spike timing in neocortical neurons[J]. *Science*, 1995, 268(5216): 1503-1506.

REFERENCES

- [37] Brette R. Philosophy of the spike: rate-based vs. spike-based theories of the brain[J]. *Frontiers in systems neuroscience*, 2015, 9: 151.
- [38] London M, Häusser M. Dendritic computation[J]. *Annu. Rev. Neurosci.*, 2005, 28: 503-532.
- [39] Truccolo W, Hochberg L R, Donoghue J P. Collective dynamics in human and monkey sensorimotor cortex: predicting single neuron spikes[J]. *Nature neuroscience*, 2010, 13(1): 105-111.
- [40] Kraynyukova N, Tchumatchenko T. Stabilized supralinear network can give rise to bistable, oscillatory, and persistent activity[J]. *Proceedings of the National Academy of Sciences*, 2018, 115(13): 3464-3469.
- [41] Izhikevich E M. Polychronization: computation with spikes[J]. *Neural computation*, 2006, 18 (2): 245-282.
- [42] Miller K D. Canonical computations of cerebral cortex[J]. *Current opinion in neurobiology*, 2016, 37: 75-84.
- [43] Yang G R, Joglekar M R, Song H F, et al. Task representations in neural networks trained to perform many cognitive tasks[J]. *Nature neuroscience*, 2019, 22(2): 297-306.
- [44] Tuckwell H C. Introduction to theoretical neurobiology: linear cable theory and dendritic structure: volume 1[M]. Cambridge University Press, 1988.
- [45] Naud R, Gerstner W. Coding and decoding with adapting neurons: a population approach to the peri-stimulus time histogram[M]. Public Library of Science San Francisco, USA, 2012.
- [46] Coombes S. Next generation neural population models[J]. *Frontiers in Applied Mathematics and Statistics*, 2023, 9: 1128224.
- [47] Tian Y, Murphy M J, Steiner L A, et al. Modeling instantaneous firing rate of deep brain stimulation target neuronal ensembles in the basal ganglia and thalamus[J]. *Neuromodulation: Technology at the Neural Interface*, 2024, 27(3): 464-475.
- [48] Butts D A. Data-driven approaches to understanding visual neuron activity[J]. *Annual review of vision science*, 2019, 5: 451-477.
- [49] Mongillo G, Barak O, Tsodyks M. Synaptic theory of working memory[J]. *Science*, 2008, 319 (5869): 1543-1546.
- [50] Carandini M, Heeger D J. Normalization as a canonical neural computation[J]. *Nature reviews neuroscience*, 2012, 13(1): 51-62.
- [51] Keeley S, Byrne Á, Fenton A, et al. Firing rate models for gamma oscillations[J]. *Journal of neurophysiology*, 2019, 121(6): 2181-2190.
- [52] Marblestone A H, Wayne G, Kording K P. Toward an integration of deep learning and neuroscience[J]. *Frontiers in computational neuroscience*, 2016, 10: 94.
- [53] Brette R, Gerstner W. Adaptive exponential integrate-and-fire model as an effective description of neuronal activity[J]. *Journal of neurophysiology*, 2005, 94(5): 3637-3642.
- [54] Dai K, Gratton S L, Billeh Y N, et al. Brain modeling toolkit: An open source software suite for multiscale modeling of brain circuits[J]. *PLOS Computational Biology*, 2020, 16(11): e1008386.
- [55] The brain modeling toolkit[EB/OL]. Allen Institute for Brain Science. <https://alleninstitute.git.hub.io/bmtk/>.

REFERENCES

- [56] Hubel D H, Wiesel T N. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex[J]. *The Journal of physiology*, 1962, 160(1): 106.
- [57] Krasovskaya S, MacInnes W J. Salience models: A computational cognitive neuroscience review[J]. *Vision*, 2019, 3(4): 56.
- [58] Fecteau J H, Munoz D P. Salience, relevance, and firing: a priority map for target selection[J]. *Trends in cognitive sciences*, 2006, 10(8): 382-390.
- [59] Reynolds J H, Heeger D J. The normalization model of attention[J]. *Neuron*, 2009, 61(2): 168-185.
- [60] Cox D D, Dean T. Neural networks and neuroscience-inspired computer vision[J]. *Current Biology*, 2014, 24(18): R921-R929.
- [61] Mathis M W, Mathis A. Deep learning tools for the measurement of animal behavior in neuroscience[J]. *Current opinion in neurobiology*, 2020, 60: 1-11.
- [62] Lindsay G W. Convolutional neural networks as a model of the visual system: Past, present, and future[J]. *Journal of cognitive neuroscience*, 2021, 33(10): 2017-2031.
- [63] Daugman J G. Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters[J]. *JOSA A*, 1985, 2(7): 1160-1169.
- [64] Jones J P, Palmer L A. An evaluation of the two-dimensional gabor filter model of simple receptive fields in cat striate cortex[J]. *Journal of neurophysiology*, 1987, 58(6): 1233-1258.
- [65] Mehrotra R, Namuduri K R, Ranganathan N. Gabor filter-based edge detection[J]. *Pattern recognition*, 1992, 25(12): 1479-1494.
- [66] Kamarainen J K, Kyrki V, Kalviainen H. Invariance properties of gabor filter-based features—overview and applications[J]. *IEEE Transactions on image processing*, 2006, 15(5): 1088-1099.
- [67] LeCun Y, Bengio Y, Hinton G. Deep learning[J]. *nature*, 2015, 521(7553): 436-444.
- [68] Yamins D L, DiCarlo J J. Using goal-driven deep learning models to understand sensory cortex [J]. *Nature neuroscience*, 2016, 19(3): 356-365.
- [69] Wolpaw J R, Birbaumer N, McFarland D J, et al. Brain–computer interfaces for communication and control[J]. *Clinical neurophysiology*, 2002, 113(6): 767-791.
- [70] Nirenberg S, Pandarinath C. Retinal prosthetic strategy with the capacity to restore normal vision[J]. *Proceedings of the National Academy of Sciences*, 2012, 109(37): 15012-15017.
- [71] Lewis P M, Ackland H M, Lowery A J, et al. Restoration of vision in blind individuals using bionic devices: a review with a focus on cortical visual prostheses[J]. *Brain research*, 2015, 1595: 51-73.
- [72] Marder E, Goaillard J M. Variability, compensation and homeostasis in neuron and network function[J]. *Nature Reviews Neuroscience*, 2006, 7(7): 563-574.
- [73] Herz A V, Gollisch T, Machens C K, et al. Modeling single-neuron dynamics and computations: a balance of detail and abstraction[J]. *science*, 2006, 314(5796): 80-85.
- [74] Gjorgjieva J, Drion G, Marder E. Computational implications of biophysical diversity and multiple timescales in neurons and synapses for circuit performance[J]. *Current opinion in neurobiology*, 2016, 37: 44-52.

REFERENCES

- [75] Fairhall A L, Lewen G D, Bialek W, et al. Efficiency and ambiguity in an adaptive neural code [J]. *Nature*, 2001, 412(6849): 787-792.
- [76] Billeh Y N, Cai B, Gratiy S L, et al. Systematic integration of structural and functional data into multi-scale models of mouse primary visual cortex[J]. *Neuron*, 2020, 106(3): 388-403.
- [77] Chen G, Scherr F, Maass W. A data-based large-scale model for primary visual cortex enables brain-like robust and versatile visual processing[J]. *science advances*, 2022, 8(44): eabq7592.
- [78] Hodgkin A L, Huxley A F, Katz B. Measurement of current-voltage relations in the membrane of the giant axon of loligo[J]. *The Journal of physiology*, 1952, 116(4): 424.
- [79] Hodgkin A L, Huxley A F. A quantitative description of membrane current and its application to conduction and excitation in nerve[J]. *The Journal of physiology*, 1952, 117(4): 500.
- [80] Gerstner W, Naud R. How good are neuron models?[J]. *Science*, 2009, 326(5951): 379-380.
- [81] Abbott L F. Lapicque's introduction of the integrate-and-fire model neuron (1907)[J]. *Brain research bulletin*, 1999, 50(5-6): 303-304.
- [82] Izhikevich E M. Simple model of spiking neurons[J]. *IEEE Transactions on neural networks*, 2003, 14(6): 1569-1572.
- [83] Mensi S, Naud R, Pozzorini C, et al. Parameter extraction and classification of three cortical neuron types reveals two distinct adaptation mechanisms[J]. *Journal of neurophysiology*, 2012, 107(6): 1756-1775.
- [84] Teeter C, Iyer R, Menon V, et al. Generalized leaky integrate-and-fire models classify multiple neuron types[J]. *Nature communications*, 2018, 9(1): 709.
- [85] Thorpe S, Delorme A, Van Rullen R. Spike-based strategies for rapid processing[J]. *Neural networks*, 2001, 14(6-7): 715-725.
- [86] Lundqvist M, Rehn M, Djurfeldt M, et al. Attractor dynamics in a modular network model of neocortex[J]. *Network: Computation in Neural Systems*, 2006, 17(3): 253-276.
- [87] Wang X J. Probabilistic decision making by slow reverberation in cortical circuits[J]. *Neuron*, 2002, 36(5): 955-968.
- [88] Bohte S M, Kok J N, La Poutre H. Error-backpropagation in temporally encoded networks of spiking neurons[J]. *Neurocomputing*, 2002, 48(1-4): 17-37.
- [89] Maass W, Natschläger T, Markram H. Real-time computing without stable states: A new framework for neural computation based on perturbations[J]. *Neural computation*, 2002, 14(11): 2531-2560.
- [90] Furber S. Large-scale neuromorphic computing systems[J]. *Journal of neural engineering*, 2016, 13(5): 051001.
- [91] Song S, Miller K D, Abbott L F. Competitive hebbian learning through spike-timing-dependent synaptic plasticity[J]. *Nature neuroscience*, 2000, 3(9): 919-926.
- [92] Guyonneau R, VanRullen R, Thorpe S J. Neurons tune to the earliest spikes through stdp[J]. *Neural Computation*, 2005, 17(4): 859-879.
- [93] Pfeiffer M, Pfeil T. Deep learning with spiking neurons: Opportunities and challenges[J]. *Frontiers in neuroscience*, 2018, 12: 409662.

REFERENCES

- [94] Thakur C S, Molin J L, Cauwenberghs G, et al. Large-scale neuromorphic spiking array processors: A quest to mimic the brain[J]. *Frontiers in neuroscience*, 2018, 12: 891.
- [95] Davies M, Srinivasa N, Lin T H, et al. Loihi: A neuromorphic manycore processor with on-chip learning[J]. *Ieee Micro*, 2018, 38(1): 82-99.
- [96] Schuman C D, Potok T E, Patton R M, et al. A survey of neuromorphic computing and neural networks in hardware[A]. 2017.
- [97] Zhu K, Pazos S, Aguirre F, et al. Hybrid 2d–cmos microchips for memristive applications[J]. *Nature*, 2023, 618(7963): 57-62.
- [98] Li X, Zhong Y, Chen H, et al. A memristors-based dendritic neuron for high-efficiency spatial-temporal information processing[J]. *Advanced Materials*, 2023, 35(37): 2203684.
- [99] Bouvier M, Valentian A, Mesquida T, et al. Spiking neural networks hardware implementations and challenges: A survey[J]. *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, 2019, 15(2): 1-35.
- [100] Zhang W, Yao P, Gao B, et al. Edge learning using a fully integrated neuro-inspired memristor chip[J]. *Science*, 2023, 381(6663): 1205-1211.
- [101] Li Y, Tang J, Gao B, et al. Monolithic three-dimensional integration of rram-based hybrid memory architecture for one-shot learning[J]. *Nature Communications*, 2023, 14(1): 7140.
- [102] Wu Y, Deng L, Li G, et al. Spatio-temporal backpropagation for training high-performance spiking neural networks[J]. *Frontiers in neuroscience*, 2018, 12: 331.
- [103] Shrestha S B, Orchard G. Slayer: Spike layer error reassignment in time[J]. *Advances in neural information processing systems*, 2018, 31.
- [104] Yi Z, Lian J, Liu Q, et al. Learning rules in spiking neural networks: A survey[J]. *Neurocomputing*, 2023, 531: 163-179.
- [105] Zenke F, Vogels T P. The remarkable robustness of surrogate gradient learning for instilling complex function in spiking neural networks[J]. *Neural computation*, 2021, 33(4): 899-925.
- [106] Li Y, Guo Y, Zhang S, et al. Differentiable spike: Rethinking gradient-descent for training spiking neural networks[J]. *Advances in Neural Information Processing Systems*, 2021, 34: 23426-23439.
- [107] Stockl C, Maass W. Optimized spiking neurons can classify images with high accuracy through temporal coding with two spikes[J]. *Nature Machine Intelligence*, 2021, 3(3): 230-238.
- [108] Rueckauer B, Lungu I A, Hu Y, et al. Conversion of continuous-valued deep networks to efficient event-driven networks for image classification[J]. *Frontiers in Neuroscience*, 2017, 11: 682.
- [109] Sengupta A, Ye Y, Wang R, et al. Going deeper in spiking neural networks: Vgg and residual architectures[J]. *Frontiers in Neuroscience*, 2019, 13.
- [110] Hu Y, Tang H, Pan G. Spiking deep residual network[J]. *IEEE Transactions on Neural Networks and Learning Systems*, 2021.
- [111] Han B, Roy K. Deep spiking neural network: Energy efficiency through time based coding[C]// European Conference on Computer Vision. 2020: 388-404.

REFERENCES

- [112] Li G, Deng L, Tang H, et al. Brain inspired computing: A systematic survey and future trends [M]. TechRxiv, 2023.
- [113] Bottou L. Stochastic gradient descent tricks[M]//Neural networks: Tricks of the trade. Springer, 2012: 421-436.
- [114] Kingma D P, Ba J. Adam: A method for stochastic optimization[A]. 2014.
- [115] Mante V, Sussillo D, Shenoy K V, et al. Context-dependent computation by recurrent dynamics in prefrontal cortex[J]. nature, 2013, 503(7474): 78-84.
- [116] Barak O, Sussillo D, Romo R, et al. From fixed points to chaos: three models of delayed discrimination[J]. Progress in neurobiology, 2013, 103: 214-222.
- [117] Pietras B, Devalle F, Roxin A, et al. Exact firing rate model reveals the differential effects of chemical versus electrical synapses in spiking networks[J]. Physical Review E, 2019, 100(4): 042412.
- [118] DePasquale B, Churchland M M, Abbott L. Using firing-rate dynamics to train recurrent networks of spiking model neurons[A]. 2016.
- [119] Schwalger T, Chizhov A V. Mind the last spike—firing rate models for mesoscopic populations of spiking neurons[J]. Current opinion in neurobiology, 2019, 58: 155-166.
- [120] Sussillo D, Abbott L F. Generating coherent patterns of activity from chaotic neural networks [J]. Neuron, 2009, 63(4): 544-557.
- [121] Eliasmith C, Anderson C H. Neural engineering: Computation, representation, and dynamics in neurobiological systems[M]. MIT press, 2003.
- [122] Cadieu C F, Hong H, Yamins D L, et al. Deep neural networks rival the representation of primate it cortex for core visual object recognition[J]. PLoS computational biology, 2014, 10 (12): e1003963.
- [123] Yamins D L, Hong H, Cadieu C F, et al. Performance-optimized hierarchical models predict neural responses in higher visual cortex[J]. Proceedings of the national academy of sciences, 2014, 111(23): 8619-8624.
- [124] Hubel D H, Wiesel T N. Receptive fields of single neurones in the cat's striate cortex[J]. The Journal of physiology, 1959, 148(3): 574.
- [125] Fukushima K. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position[J]. Biological cybernetics, 1980, 36(4): 193-202.
- [126] Carandini M, Heeger D J, Movshon J A. Linearity and normalization in simple cells of the macaque primary visual cortex[J]. Journal of Neuroscience, 1997, 17(21): 8621-8644.
- [127] Shapley R, Hawken M J. Color in the cortex: single-and double-opponent cells[J]. Vision research, 2011, 51(7): 701-717.
- [128] Olshausen B A, Field D J. Emergence of simple-cell receptive field properties by learning a sparse code for natural images[J]. Nature, 1996, 381(6583): 607-609.
- [129] Serre T, Oliva A, Poggio T. A feedforward architecture accounts for rapid categorization[J]. Proceedings of the national academy of sciences, 2007, 104(15): 6424-6429.

REFERENCES

- [130] Spoerer C J, McClure P, Kriegeskorte N. Recurrent convolutional neural networks: a better model of biological object recognition[J]. *Frontiers in psychology*, 2017, 8: 278016.
- [131] Gerstner W, Kempter R, Van Hemmen J L, et al. A neuronal learning rule for sub-millisecond temporal coding[J]. *Nature*, 1996, 383(6595): 76-78.
- [132] Izhikevich E M, Edelman G M. Large-scale model of mammalian thalamocortical systems[J]. *Proceedings of the national academy of sciences*, 2008, 105(9): 3593-3598.
- [133] Ritter P, Schirner M, McIntosh A R, et al. The virtual brain integrates computational modeling and multimodal neuroimaging[J]. *Brain connectivity*, 2013, 3(2): 121-145.
- [134] D'Angelo E, Jirsa V. The quest for multiscale brain modeling[J]. *Trends in neurosciences*, 2022, 45(10): 777-790.
- [135] Arachchige A S P M. The blue brain project: pioneering the frontier of brain simulation[J]. *AIMS neuroscience*, 2023, 10(4): 315.
- [136] Models of the mouse primary visual cortex[EB/OL]. Allen Institute for Brain Science. <https://portal.brain-map.org/explore/models/mv1-all-layers>.
- [137] Numerical methods for ordinary differential equations[EB/OL]. Wikipedia. https://en.wikipedia.org/wiki/Numerical_methods_for_ordinary_differential_equations.
- [138] Exponential integrator[EB/OL]. Wikipedia. https://en.wikipedia.org/wiki/Exponential_integrator.
- [139] Rumyantsev O I, Lecoq J A, Hernandez O, et al. Fundamental bounds on the fidelity of sensory cortical coding[J]. *Nature*, 2020, 580(7801): 100-105.
- [140] Stringer C, Michaelos M, Tsyboulski D, et al. High-precision coding in visual cortex[J]. *Cell*, 2021, 184(10): 2767-2778.
- [141] Garrett M, Manavi S, Roll K, et al. Experience shapes activity dynamics and stimulus coding of vip inhibitory cells[J]. *elife*, 2020, 9: e50340.
- [142] Siegle J H, Jia X, Durand S, et al. Survey of spiking in the mouse visual system reveals functional hierarchy[J]. *Nature*, 2021, 592(7852): 86-92.
- [143] Chichilnisky E. A simple white noise analysis of neuronal light responses[J]. *Network: computation in neural systems*, 2001, 12(2): 199.
- [144] Simoncelli E P, Paninski L, Pillow J, et al. Characterization of neural responses with stochastic stimuli[J]. *The cognitive neurosciences*, 2004, 3(327-338): 1.
- [145] Schwartz O, Pillow J W, Rust N C, et al. Spike-triggered neural characterization[J]. *Journal of vision*, 2006, 6(4): 13-13.
- [146] Durand S, Iyer R, Mizuseki K, et al. A comparison of visual response properties in the lateral geniculate nucleus and primary visual cortex of awake and anesthetized mice[J]. *Journal of Neuroscience*, 2016, 36(48): 12144-12156.
- [147] Ricker wavelet[EB/OL]. Wikipedia. https://en.wikipedia.org/wiki/Ricker_wavelet.
- [148] He W, Wu Y, Deng L, et al. Comparing snns and rnns on neuromorphic vision datasets: Similarities and differences[J]. *Neural Networks*, 2020, 132: 108-120.

REFERENCES

- [149] Ren S, He K, Girshick R, et al. Faster r-cnn: Towards real-time object detection with region proposal networks[J]. Advances in neural information processing systems, 2015, 28.
- [150] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[J]. Advances in neural information processing systems, 2017, 30.
- [151] Song J, Meng C, Ermon S. Denoising diffusion implicit models[A]. 2020.
- [152] Gu A, Dao T. Mamba: Linear-time sequence modeling with selective state spaces[A]. 2023.
- [153] Hubel D H, Wiesel T N. Receptive fields and functional architecture of monkey striate cortex [J]. The Journal of physiology, 1968, 195(1): 215-243.
- [154] De Valois R L, Albrecht D G, Thorell L G. Spatial frequency selectivity of cells in macaque visual cortex[J]. Vision research, 1982, 22(5): 545-559.
- [155] Movshon J A, Thompson I, Tolhurst D. Spatial and temporal contrast sensitivity of neurones in areas 17 and 18 of the cat's visual cortex.[J]. The Journal of physiology, 1978, 283(1): 101-120.
- [156] Alonso J M, Martinez L M. Functional connectivity between simple cells and complex cells in cat striate cortex[J]. Nature neuroscience, 1998, 1(5): 395-403.
- [157] Martinez L M, Alonso J M. Complex receptive fields in primary visual cortex[J]. The neuroscientist, 2003, 9(5): 317-331.

APPENDIX A EXAMPLES OF TRAINING AND TEST LOSS

Here are examples of training and test loss during the learning phase. We show the training and corresponding test loss figures of the 3k-neuron networks under four scenarios (default, “uniform neurons”, “random synapses”, and “no Dale’s law” scenarios) both with and without LGN, and of the 6k-neuron networks under different output settings (L5, L2/3, L4, L6) both with and without LGN as the examples. See Figure A.1, Figure A.2, Figure A.3, and Figure A.4. Note that for the subfigure for all tasks in these figures, the loss is the sum of loss of all three tasks.

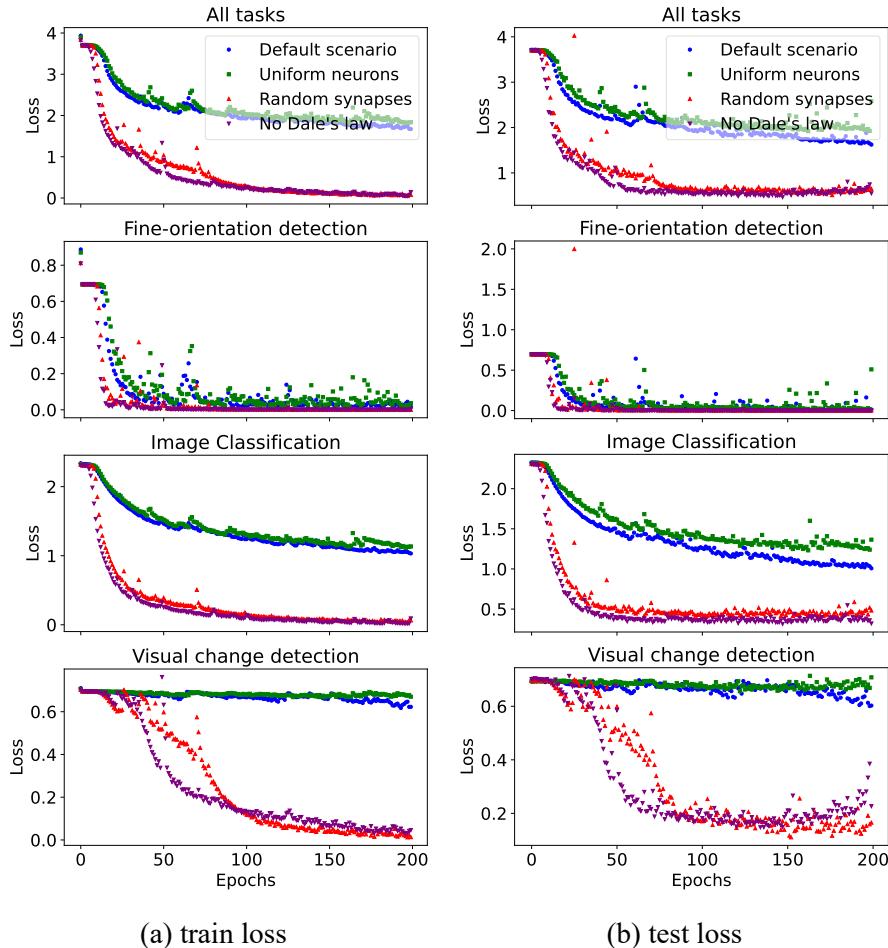


Figure A.1 Epoch-wise loss for the default and other three scenarios: sum and task-specific performance across three tasks (3k neurons)

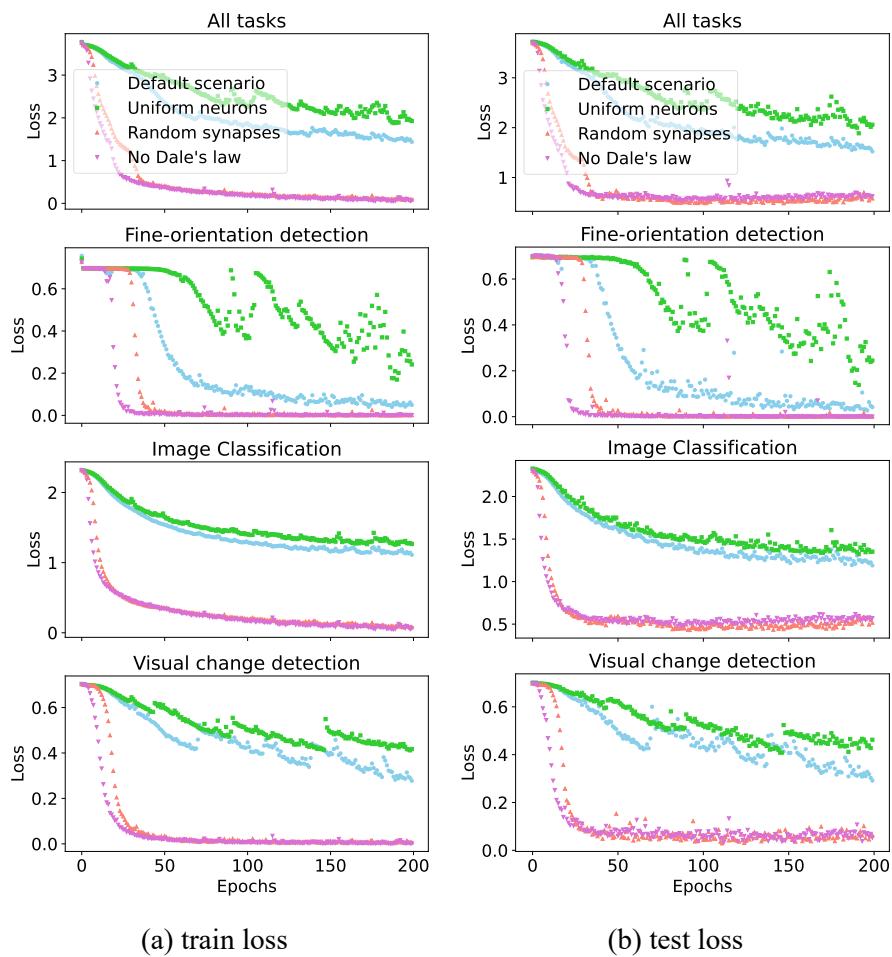


Figure A.2 Epoch-wise loss for the default and other three scenarios: sum and task-specific performance across three tasks (3k neurons, with LGN)

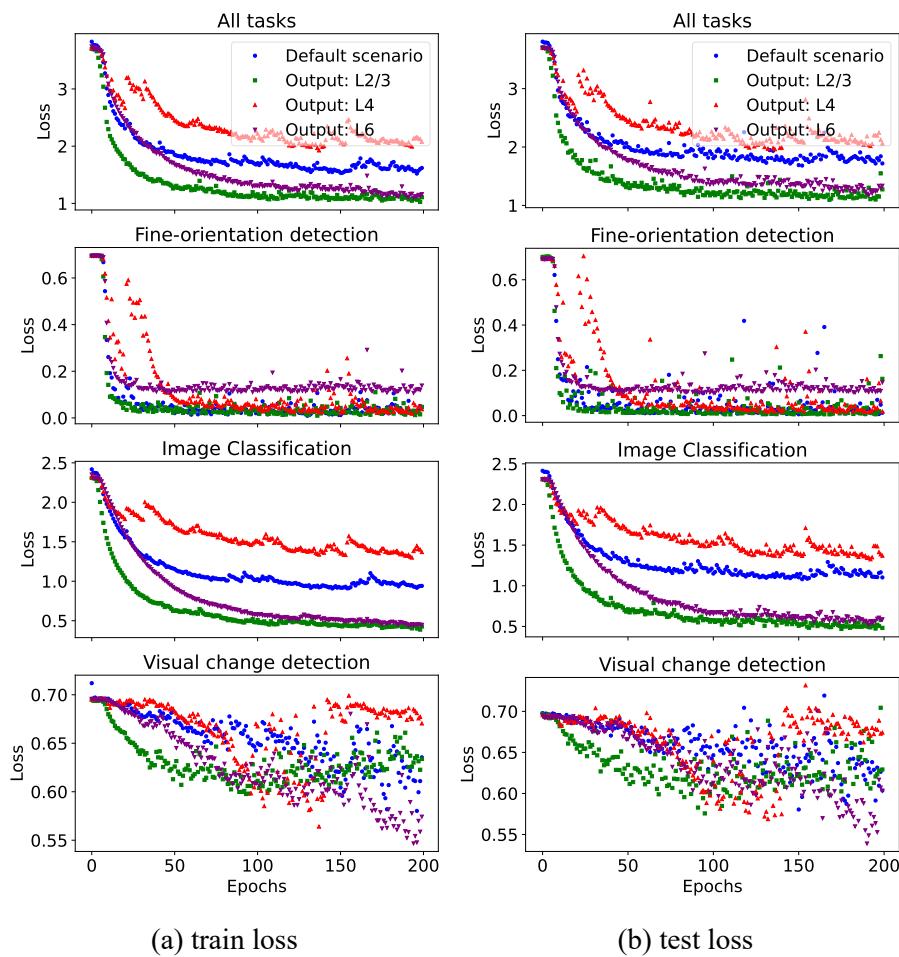


Figure A.3 Epoch-wise loss for the default scenario and other three output layer settings: sum and task-specific performance across three tasks (6k neurons)

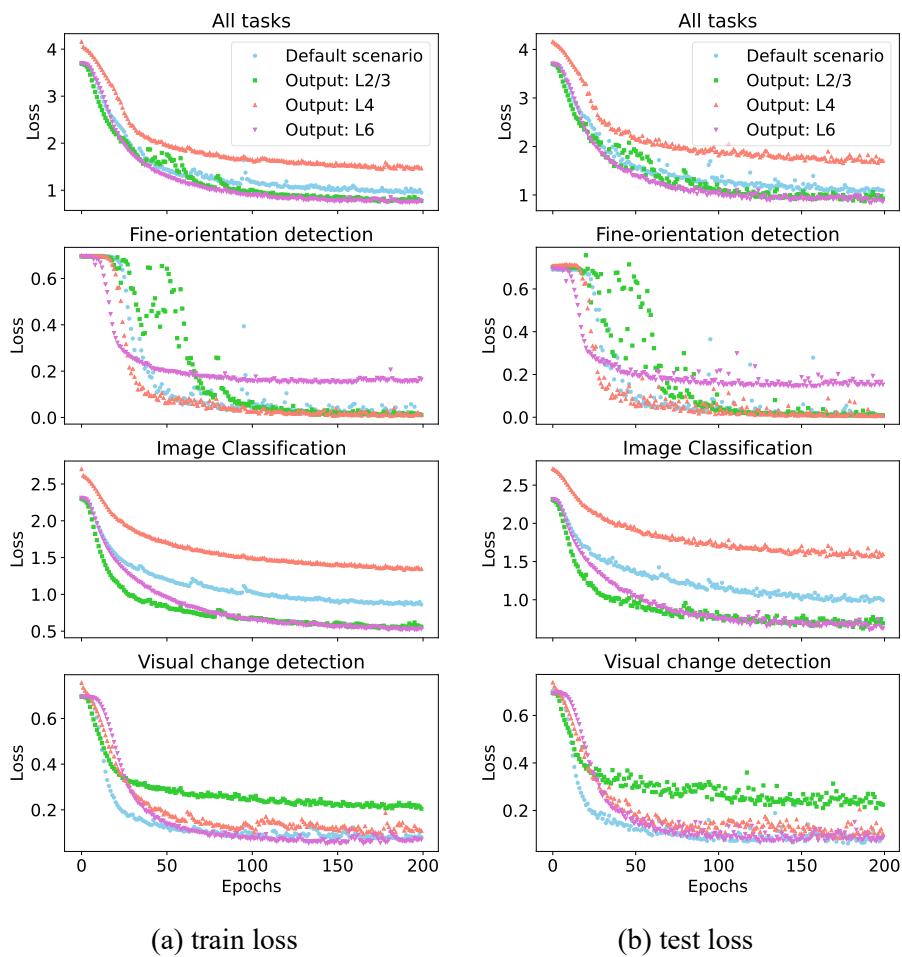


Figure A.4 Epoch-wise loss for the default scenario and other three output layer settings: sum and task-specific performance across three tasks (6k neurons, with LGN)

APPENDIX B FIRST-ORDER EXPONENTIAL INTEGRATOR METHOD

The first-order exponential integrator method is a numerical technique used to solve ordinary differential equations (ODEs) that involve exponential terms or have solutions with exponential behavior. This method is particularly well-suited for problems where the solution exhibits rapid decay or growth, as it can capture the exponential dynamics more accurately than traditional methods like the Euler method or the Runge-Kutta methods.

The key idea behind the exponential integrator method is to split the ODE into a linear part and a nonlinear part, and then treat the linear part exactly while approximating the nonlinear part. This splitting allows for the exact integration of the linear term, which often represents the dominant exponential behavior of the solution.

For a first-order linear differential equation $\frac{dy}{dx} + p(x)y = Q(x)$, the general solution is given by:

$$y = e^{-\int p(x)dx} \left(\int Q(x)e^{\int p(x)dx} dx + C \right) \quad (\text{B.1})$$

Applying this solution to a differential equation of the form $y'(t) = -Ay + \mathcal{N}(t)$, which can be rewritten as $\frac{dy}{dt} + Ay = \mathcal{N}(t)$, we have:

$$y = e^{-\int Adt} \left(\int \mathcal{N}(y(t))e^{\int Adt} dt + C \right) \quad (\text{B.2})$$

Simplifying the exponential terms, we obtain:

$$y = e^{-At} \left(\int \mathcal{N}(y(t))e^{At} dt + C \right) \quad (\text{B.3})$$

Let $t_{n+1} = t_n + h$. We have the following equations.

$$y(t_n) = e^{-At_n} \left(\int \mathcal{N}(y(t)) e^{At} dt \Big|_{t=t_n} + C \right) \quad (\text{B.4})$$

$$y(t_{n+1}) = e^{-A(t_n+h)} \left(\int \mathcal{N}(y(t)) e^{At} dt \Big|_{t=t_n+h} + C \right) \quad (\text{B.5})$$

Subtracting Equation (B.4) times e^{-Ah} from Equation (B.5), we have:

$$\begin{aligned}
 y(t_n + h) - e^{-Ah}y(t_n) &= e^{-A(t_n+h)} \left(\int \mathcal{N}(y(t))e^{At} dt \Big|_{t=t_n}^{t=t_n+h} \right) \\
 &= e^{-A(t_n+h)} \int_0^h \mathcal{N}(y(t_n + \tau))e^{A(t_n+\tau)} d\tau \\
 &= \int_0^h e^{-A(h-\tau)} \mathcal{N}(y(t_n + \tau)) d\tau
 \end{aligned} \tag{B.6}$$

Assuming $\mathcal{N}(y(t_n + \tau))$ is constant from $\tau = 0$ to $\tau = h$ and letting it be $\mathcal{N}(y(t_n))$, we have:

$$\begin{aligned}
 y(t_n + h) &\approx e^{-Ah}y(t_n) + \mathcal{N}(y(t_n)) \int_0^h e^{-A(h-\tau)} d\tau \\
 &= e^{-Ah}y(t_n) + A^{-1} (1 - e^{-Ah}) \mathcal{N}(y(t_n))
 \end{aligned} \tag{B.7}$$

Equation (B.7) is the final equation that we use to derive the difference equations for the discrete form firing-rate model.

APPENDIX C SUPPLEMENTARY LGN RESPONSES

The rate responses of the LGN model to the visual stimuli from the image classification and visual change detection tasks as well as the corresponding raster plots are supplemented here.

For the image classification task, the timings for the core stimulus are same as the fine-orientation detection task (see Figure 2.9). For the visual change detection task, the core stimulus exists from 25 ms to 125 ms and from 225 ms to 325 ms (see Figure 2.10).

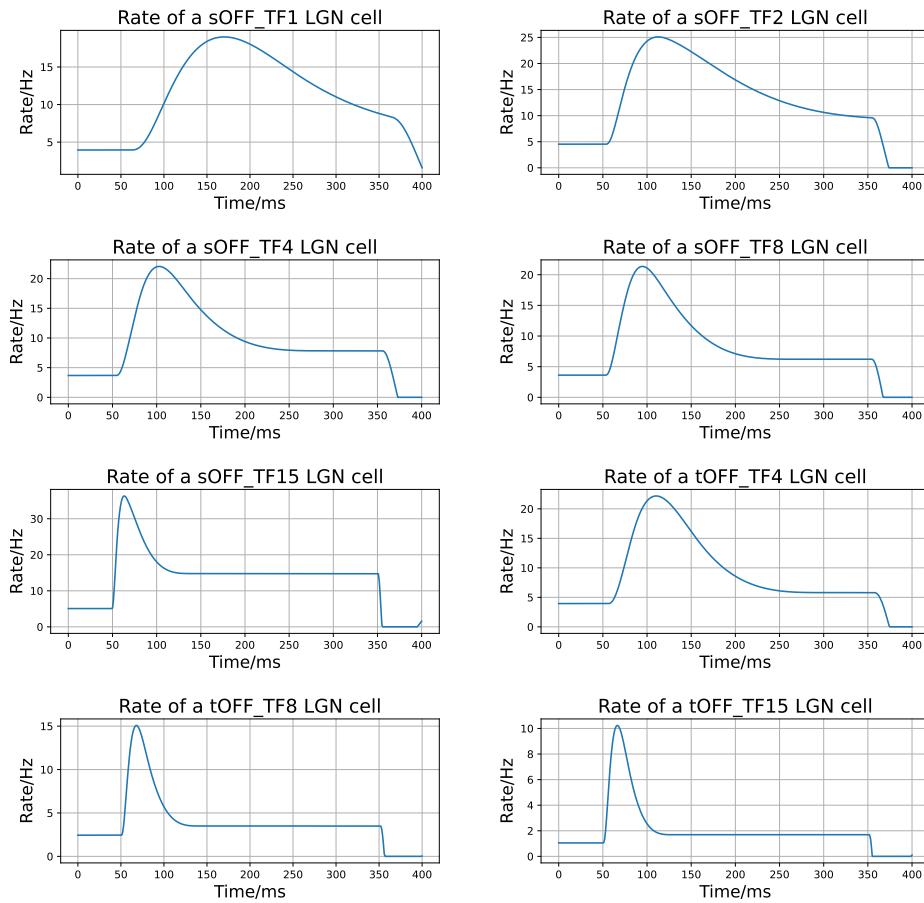


Figure C.1 Rate responses of OFF cells in our LGN model to visual stimuli from the image classification task

APPENDIX C SUPPLEMENTARY LGN RESPONSES

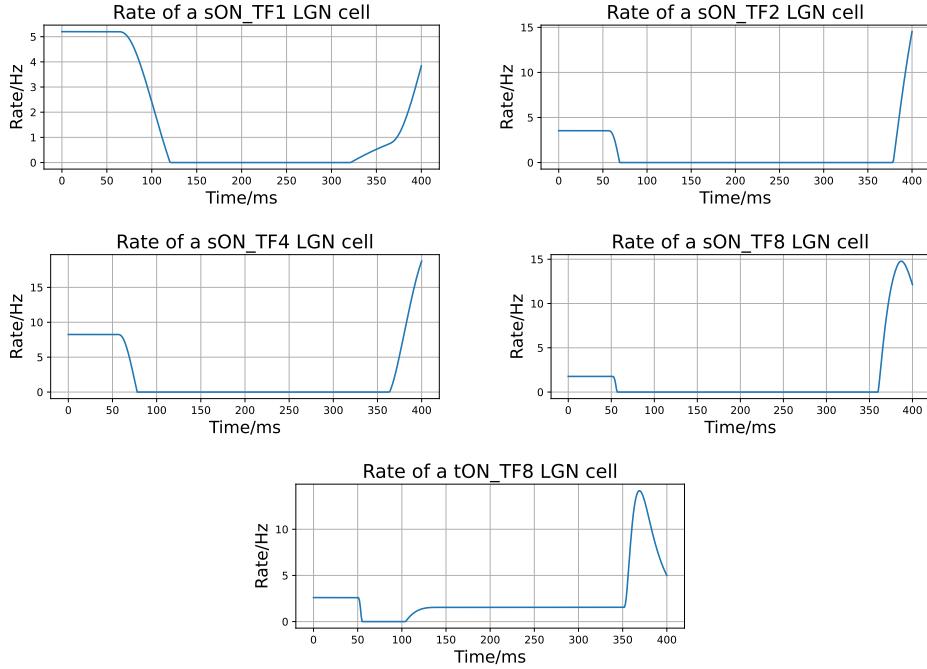
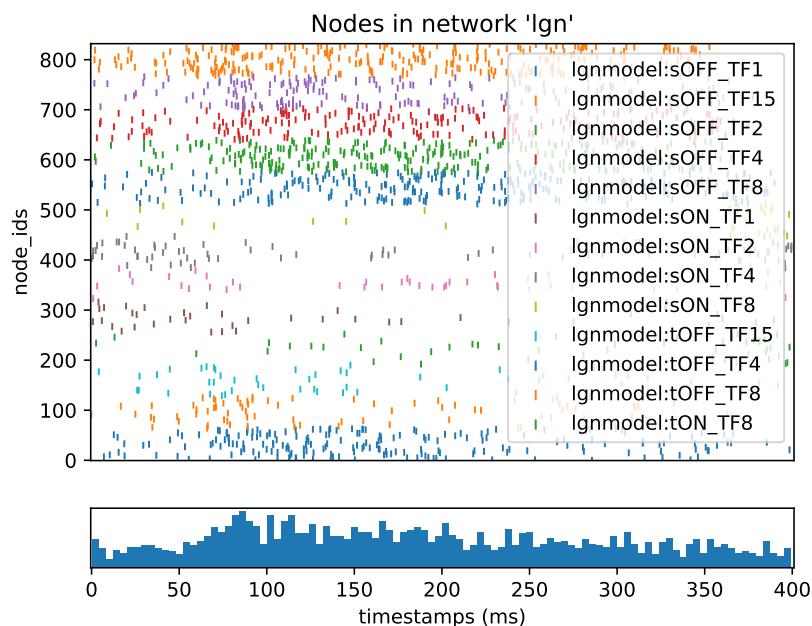


Figure C.2 Rate responses of all types of ON cells in our LGN model to visual stimuli from the image classification task



The raster plot includes all 832 cells of 13 types in our LGN models.

Figure C.3 Spike responses of all cells in our LGN model to visual stimuli from the image classification task

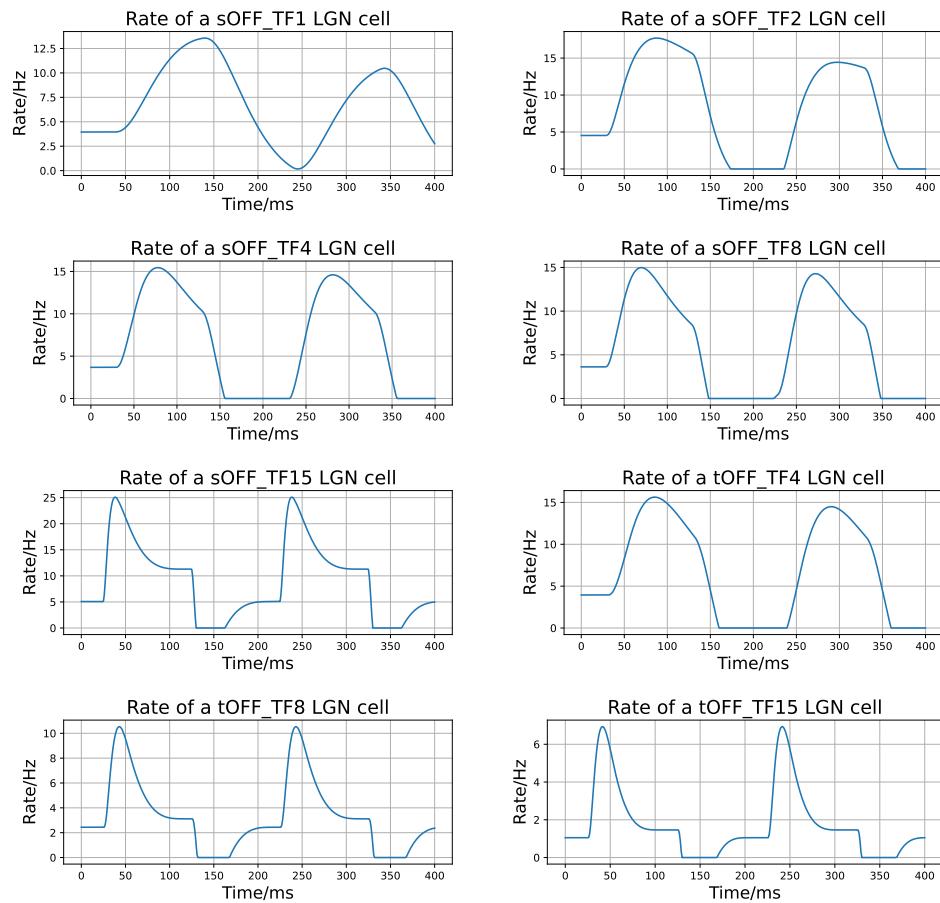


Figure C.4 Rate responses of OFF cells in our LGN model to visual stimuli from the visual change detection task

APPENDIX C SUPPLEMENTARY LGN RESPONSES

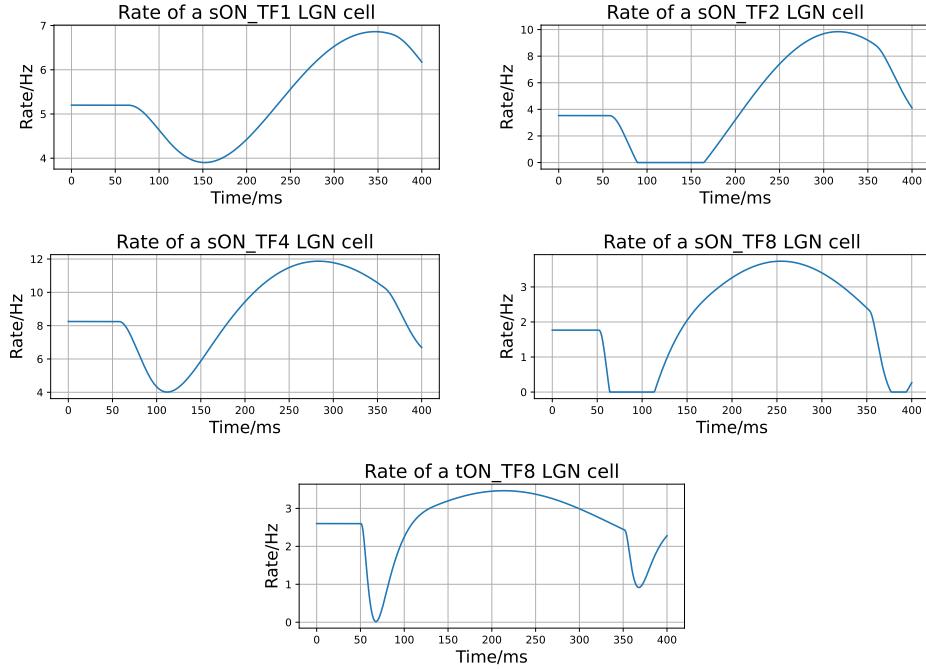
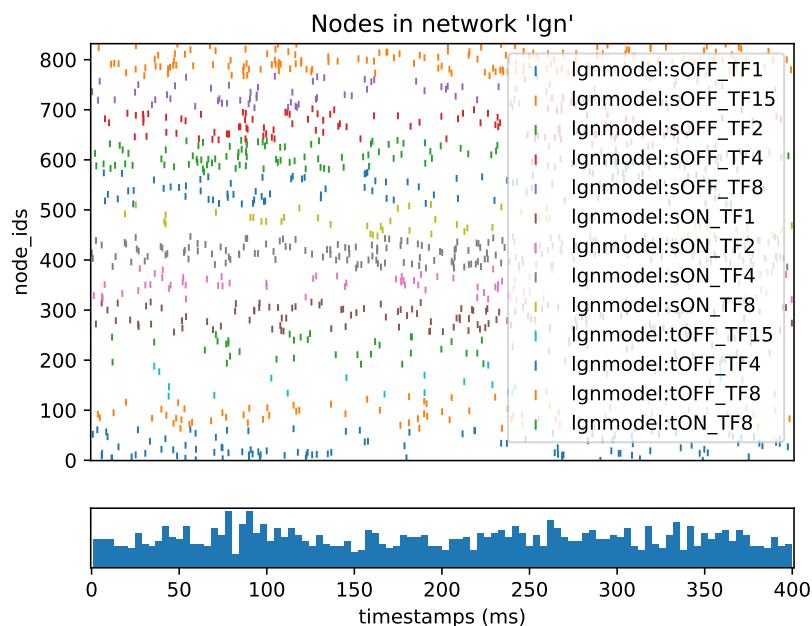


Figure C.5 Rate responses of all types of ON cells in our LGN model to visual stimuli from the visual change detection task

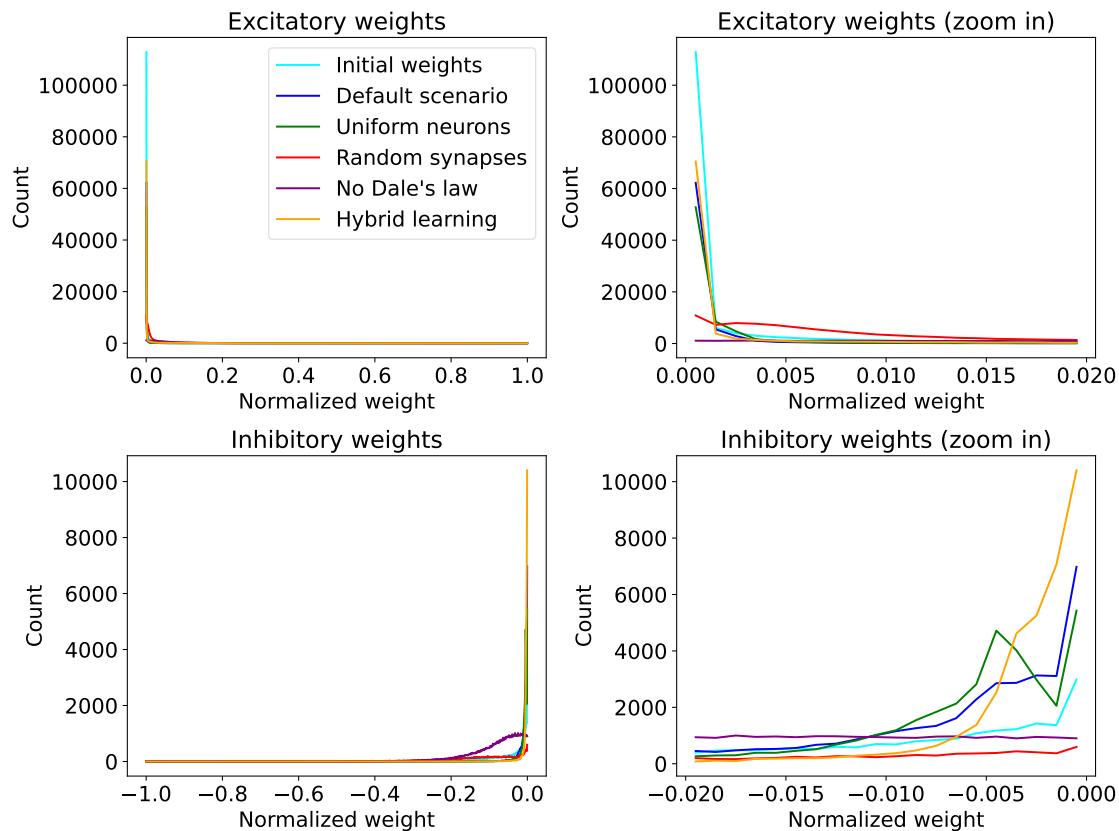


The raster plot includes all 832 cells of 13 types in our LGN models.

Figure C.6 Spike responses of all cells in our LGN model to visual stimuli from the visual change task

APPENDIX D SUPPLEMENTARY WEIGHT DISTRIBUTIONS

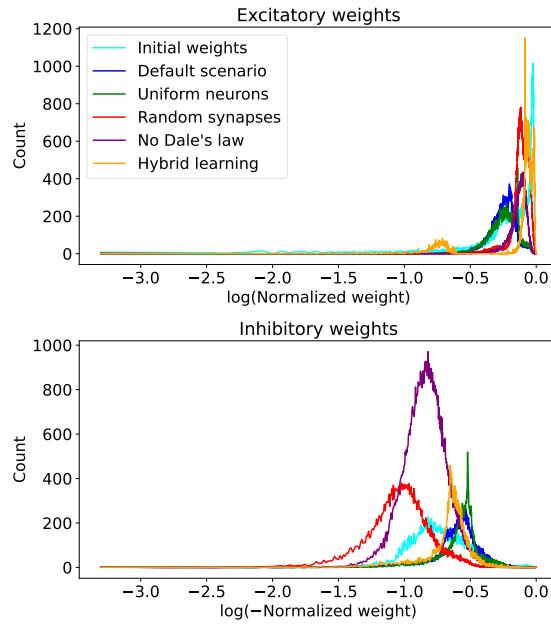
The weight distributions of the 6k-neuron network models, the 3k-neuron network models with LGN, and the 6k-neuron network models with LGN are provided here, as shown in Figure D.1, Figure D.2, Figure D.3, Figure D.4, Figure D.5, and Figure D.6. The results are similar to those of the 3k-neuron model, as discussed in section 4.1.



The layout and labeling are same as the description in Figure 4.1

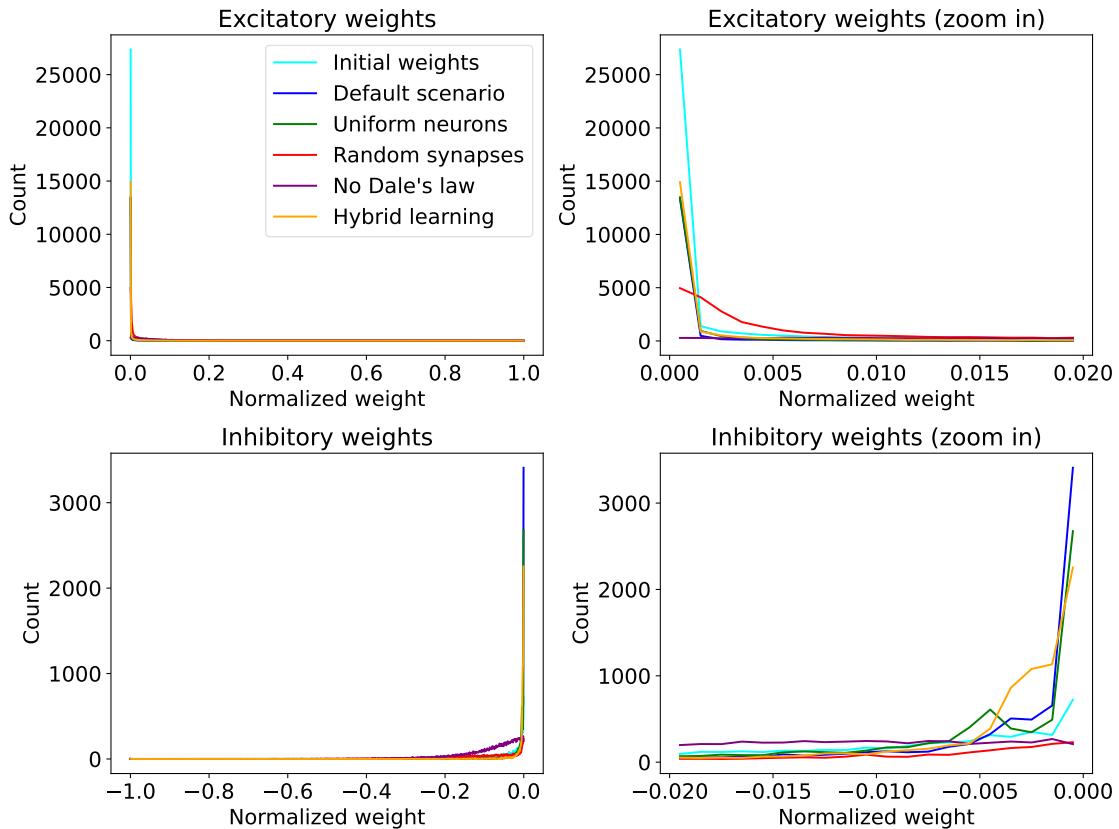
Figure D.1 Weight distributions of initial values, multiple scenarios, and the hybrid learning (6k neurons)

APPENDIX D SUPPLEMENTARY WEIGHT DISTRIBUTIONS



The layout and labeling are same as the description in Figure 4.2

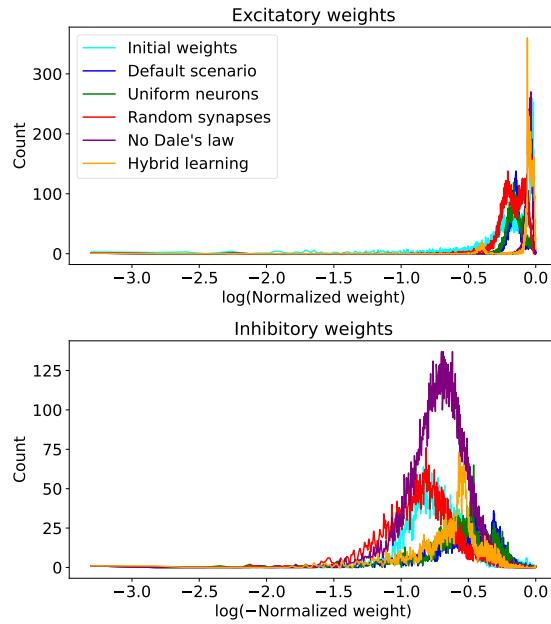
Figure D.2 Log-transformed weight distributions of initial values, multiple scenarios, and the hybrid learning (6k neurons)



The layout and labeling are same as the description in Figure 4.1

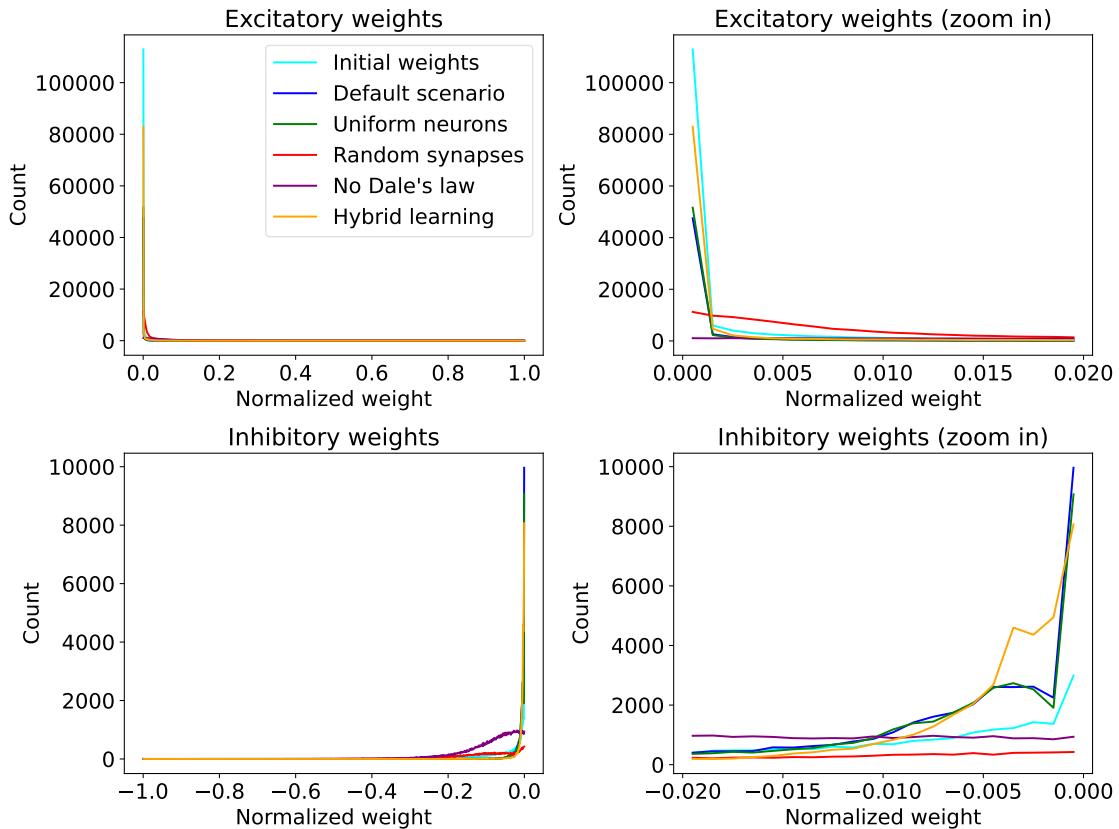
Figure D.3 Weight distributions of initial values, multiple scenarios, and the hybrid learning (3k neurons, with LGN)

APPENDIX D SUPPLEMENTARY WEIGHT DISTRIBUTIONS



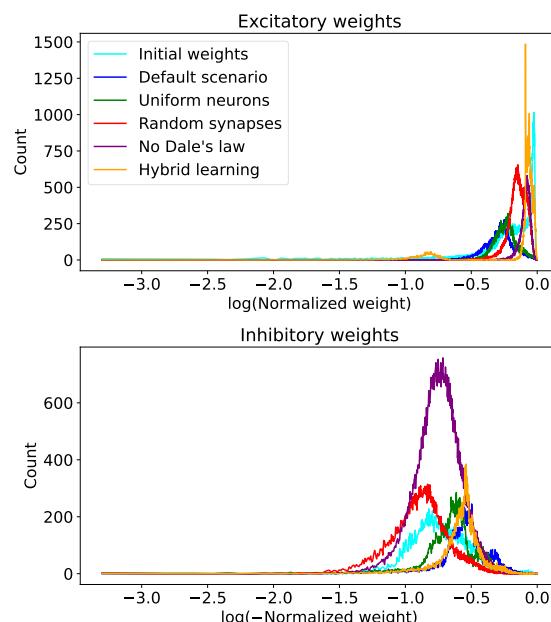
The layout and labeling are same as the description in Figure 4.2

Figure D.4 Log-transformed weight distributions of initial values, multiple scenarios, and the hybrid learning (3k neurons, with LGN)



The layout and labeling are same as the description in Figure 4.1

Figure D.5 Weight distributions of initial values, multiple scenarios, and the hybrid learning (6k neurons, with LGN)



The layout and labeling are same as the description in Figure 4.2

Figure D.6 Log-transformed weight distributions of initial values, multiple scenarios, and the hybrid learning (6k neurons, with LGN)

APPENDIX E WEIGHT MEAN DISTRIBUTION OF NEURONAL CLASS PAIRS

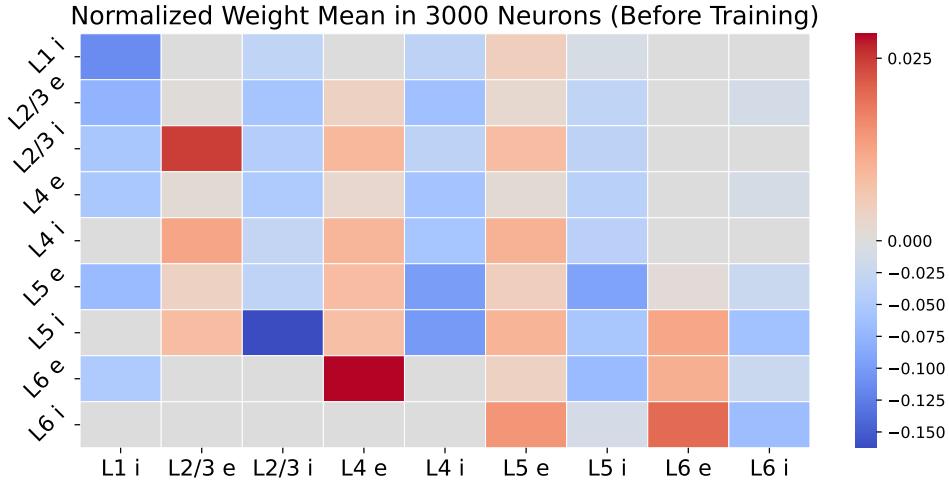
The visual representation of the normalized weight mean matrices for different neuronal class pairs is presented in this appendix. The figures (Figure E.1, Figure E.2, Figure E.3, Figure E.4, Figure E.5, Figure E.6, Figure E.5, and Figure E.6) compare the weight distributions before and after training, with and without LGN, and for 3k- and 6k-neurons, corresponding to the default scenario.

Analysis of weight changes in neuronal class pairs:

1. Before training (Figure E.1, Figure E.3, Figure E.5, and Figure E.5), the normalized weight means for most neuronal class pairs are relatively low and homogeneous, ranging from -0.025 to 0.15. This suggests that the initial synaptic weights tend to be randomly distributed and do not exhibit any significant patterns or preferences for specific neuronal class connections.
2. After training (Figure E.2, Figure E.4, Figure E.6, and Figure E.6), the normalized weight means show more pronounced variations and patterns. Some neuronal class pairs exhibit stronger positive or negative weights, indicating that the training process has modified the synaptic strengths to reflect the learned features and dependencies in the data.
3. Comparing the weight distributions before and after training, it is evident that the training process has a significant impact on the synaptic weights. The weights become more differentiated and specific to certain neuronal class connections, suggesting that the network has learned to prioritize or suppress certain pathways based on the input data and desired output.

Analysis of LGN presence:

1. Without LGN (Figure E.1, Figure E.2, Figure E.3, and Figure E.4), the normalized weight means after training (Figures E.2 and E.4) show a relatively balanced distribution of positive and negative weights across different neuronal class pairs. This indicates that the network has learned to capture both excitatory and inhibitory connections to process the input data effectively.
2. With LGN (Figure E.5, Figure E.6, Figure E.5, and Figure E.6), the normalized weight means after training (Figures E.6 and E.8) exhibit a more skewed distribution



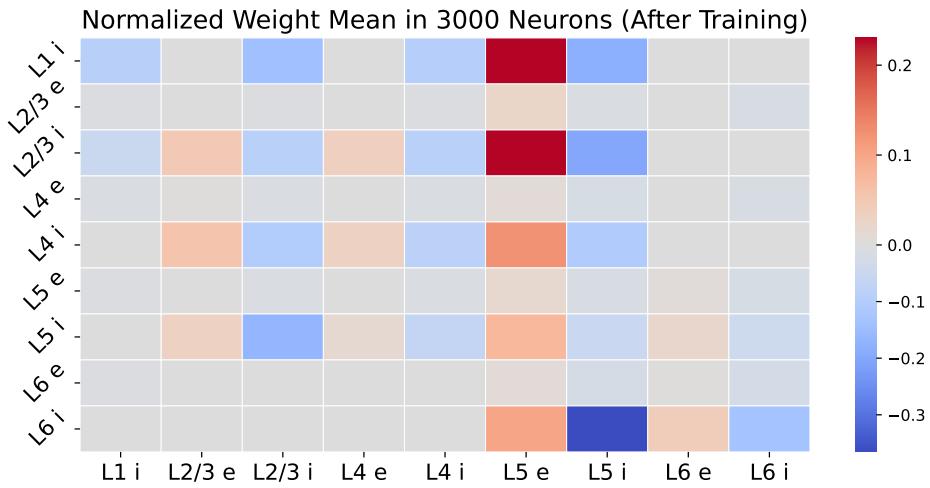
This 9×9 colored table visualizes the number of synapses between pairs of source and target neuron classes. The horizontal axis (x-axis) represents the source neuron classes, while the vertical axis (y-axis) denotes the target neuron classes.

Figure E.1 Normalized weight mean matrix for neuronal classes (3k neurons, before training)

towards positive weights, particularly for connections involving the LGN. This suggests that the presence of the LGN enhances the excitatory connections and may facilitate the propagation of visual information from the input layer to higher cortical layers.

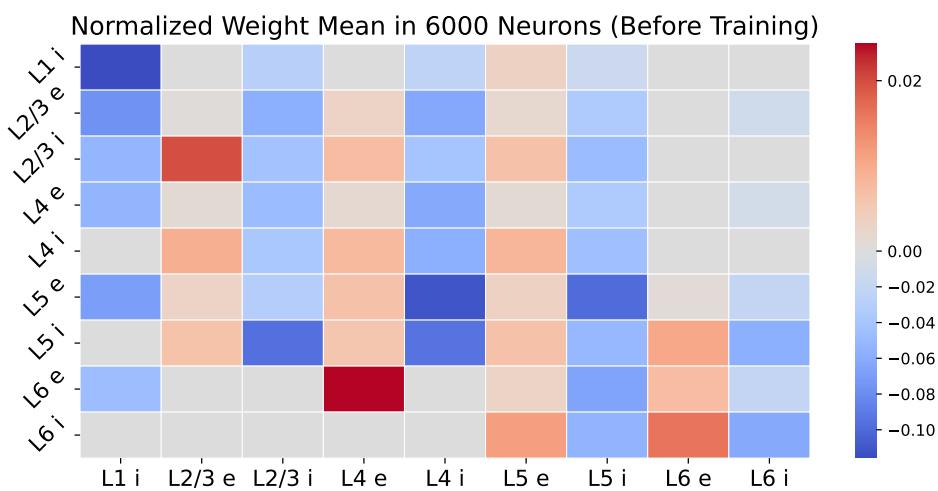
Analysis of network size (3k vs. 6k neurons):

1. The normalized weight mean matrices for 3,000 neurons (Figure E.1, Figure E.2, Figure E.5, and Figure E.6) and 6,000 neurons (Figure E.3, Figure E.4, Figure E.7, and Figure E.8) show similar overall patterns and trends in terms of weight distribution before and after training, and with or without the presence of the LGN.
2. However, the weight distributions for the 6,000-neuron network appear to be more fine-grained and have a smaller range compared to the 3,000-neuron network. This suggests that increasing the network size allows for more precise and localized weight adjustments, potentially enabling the network to capture more intricate features and patterns in the data.



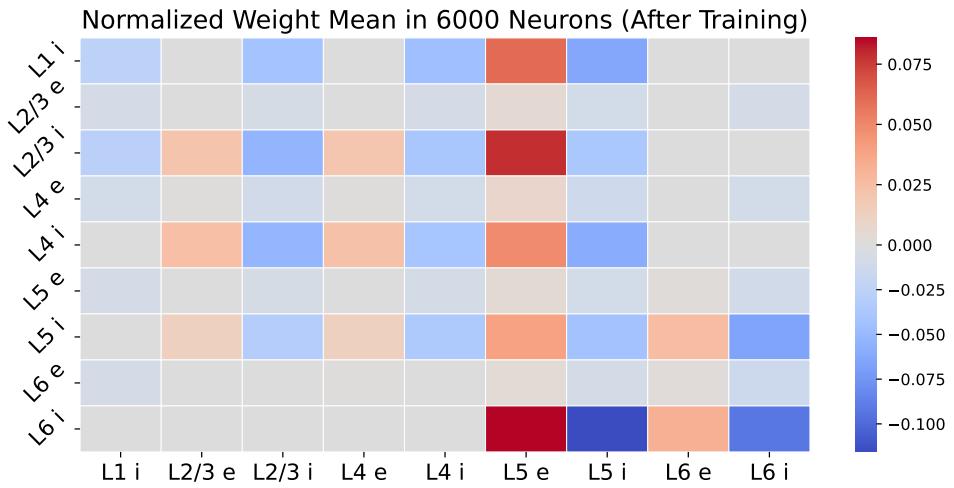
The layout and labeling are same as the description in Figure E.1.

Figure E.2 Synaptic connectivity probability matrix for neuronal classes (3k neurons, after training)



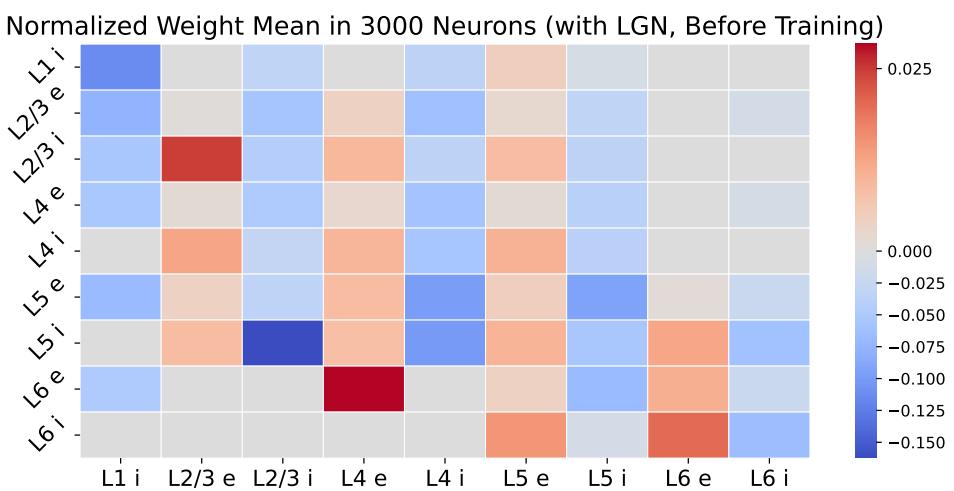
The layout and labeling are same as the description in Figure E.1.

Figure E.3 Synaptic connectivity probability matrix for neuronal classes (6k neurons, after training)



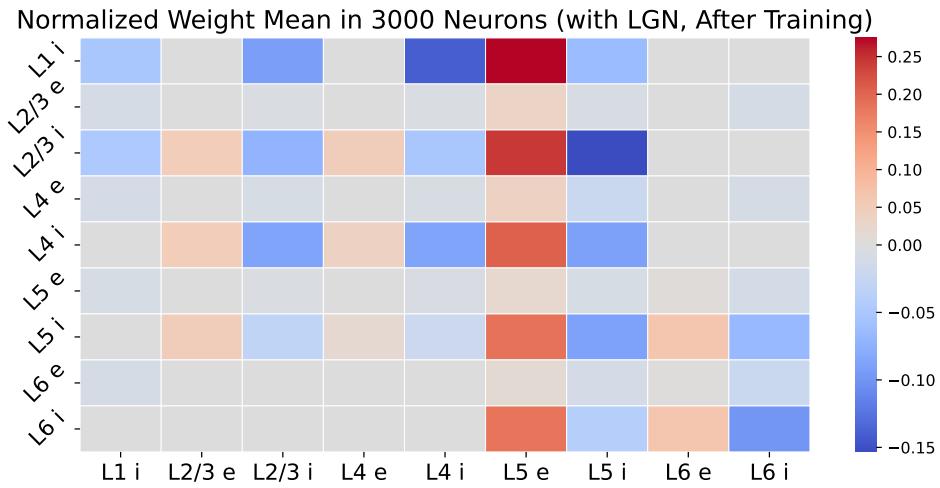
The layout and labeling are same as the description in Figure E.1.

Figure E.4 Synaptic connectivity probability matrix for neuronal classes (6k neurons, after training)



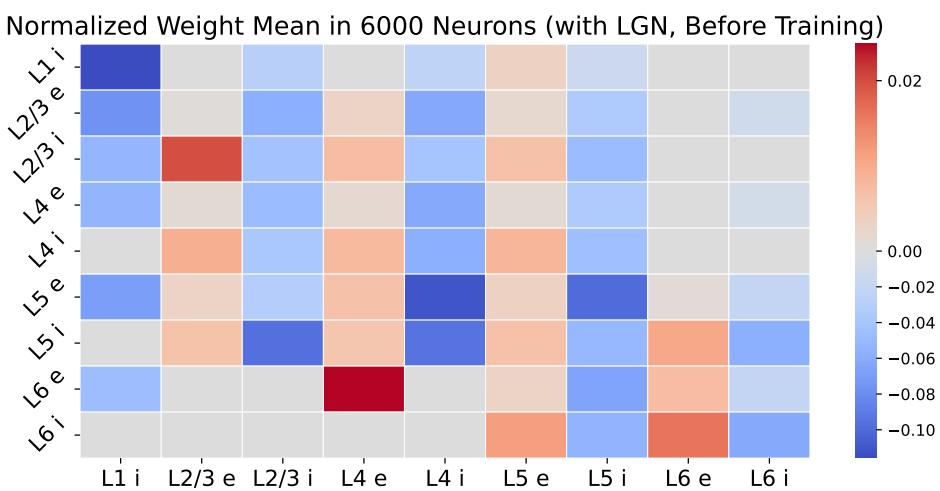
The layout and labeling are same as the description in Figure E.1.

Figure E.5 Normalized weight mean matrix for neuronal classes (3k neurons, with LGN, before training)



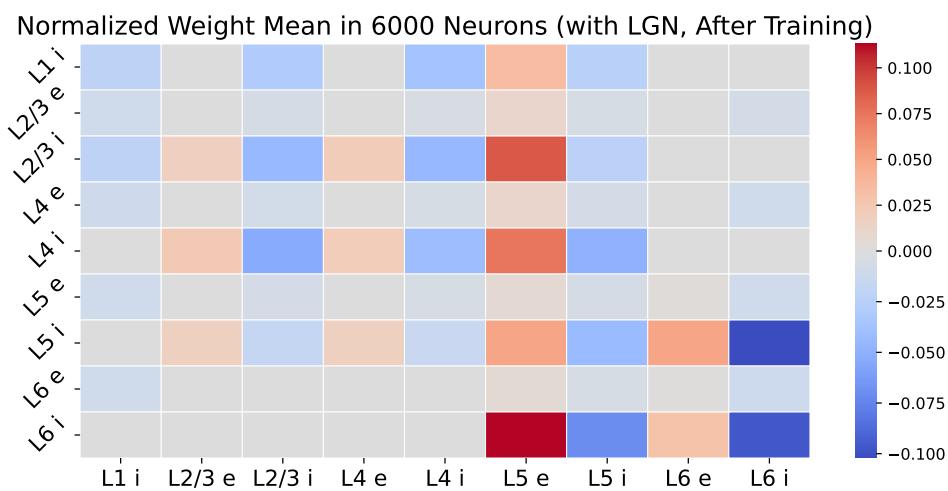
The layout and labeling are same as the description in Figure E.1.

Figure E.6 Synaptic connectivity probability matrix for neuronal classes (3k neurons, with LGN, after training)



The layout and labeling are same as the description in Figure E.1.

Figure E.7 Synaptic connectivity probability matrix for neuronal classes (6k neurons, with LGN, before training)



The layout and labeling are same as the description in Figure E.1.

Figure E.8 Synaptic connectivity probability matrix for neuronal classes (6k neurons, with LGN, after training)

APPENDIX F SUPPLEMENTARY NEURONAL RESPONSES RESULTS

The supplementary graphical results for the section 4.5 are provided here.

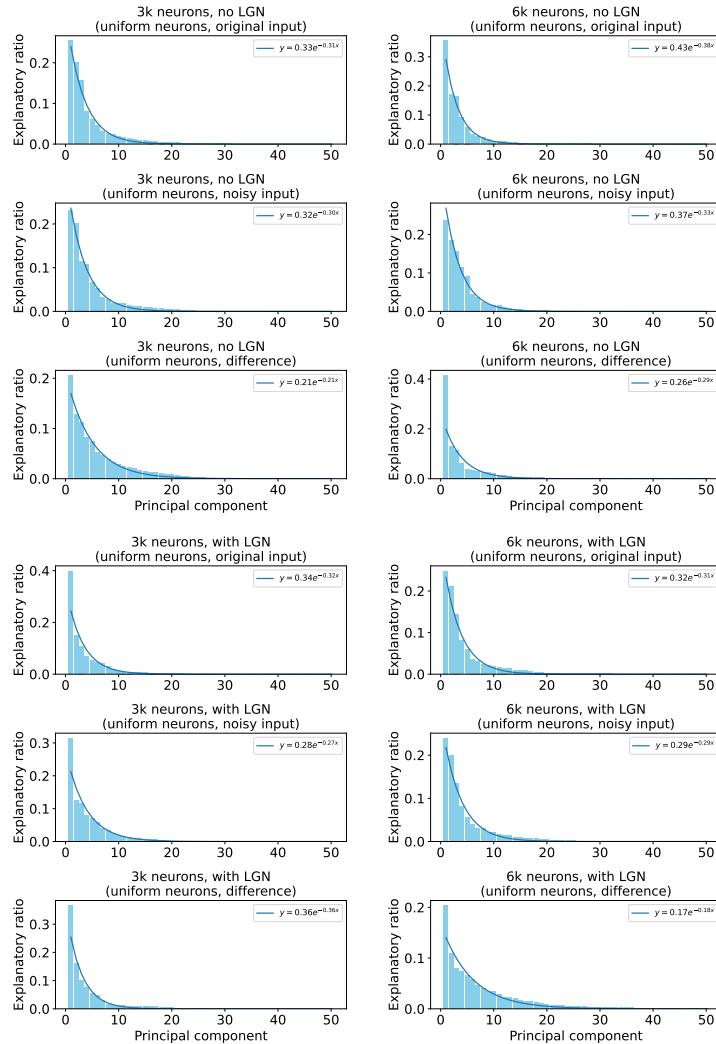


Figure F.1 PCA results of neuronal responses to clean and noisy MNIST images and of the response difference (uniform neurons)

APPENDIX F SUPPLEMENTARY NEURONAL RESPONSES RESULTS

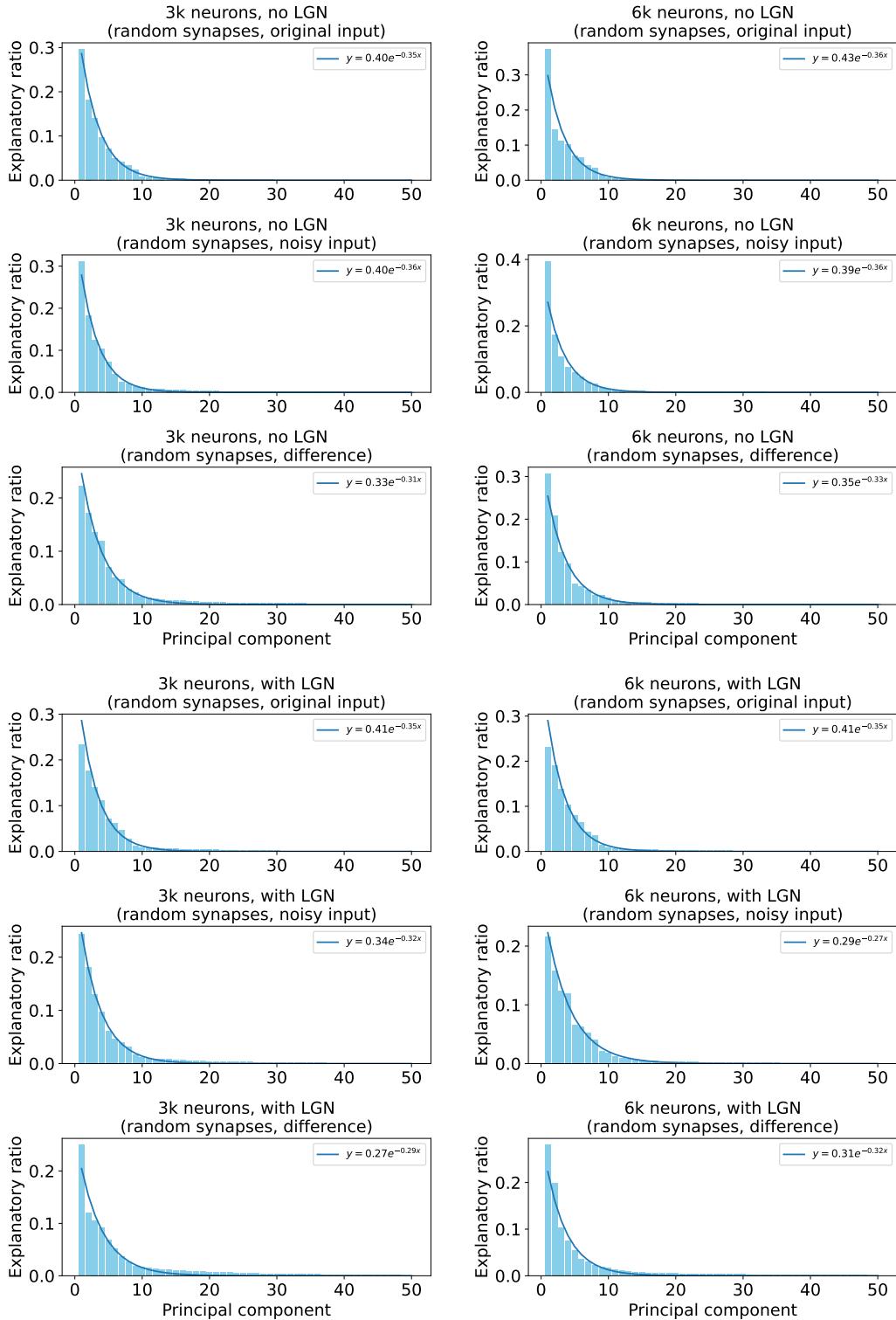


Figure F.2 PCA results of neuronal responses to clean and noisy MNIST images and of the response difference (random synapses)

APPENDIX F SUPPLEMENTARY NEURONAL RESPONSES RESULTS

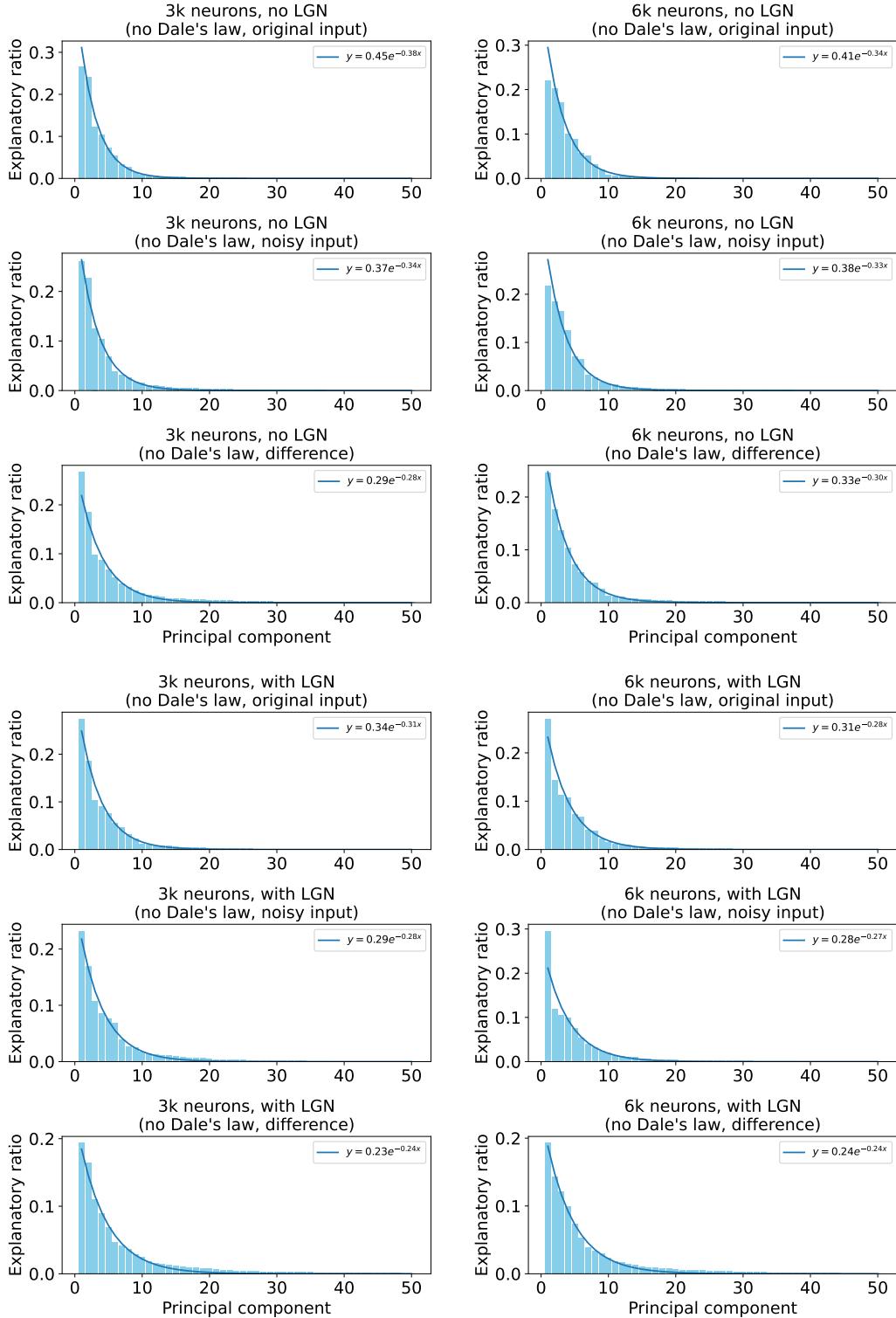


Figure F.3 PCA results of neuronal responses to clean and noisy MNIST images and of the response difference (no Dale's rule)

APPENDIX F SUPPLEMENTARY NEURONAL RESPONSES RESULTS

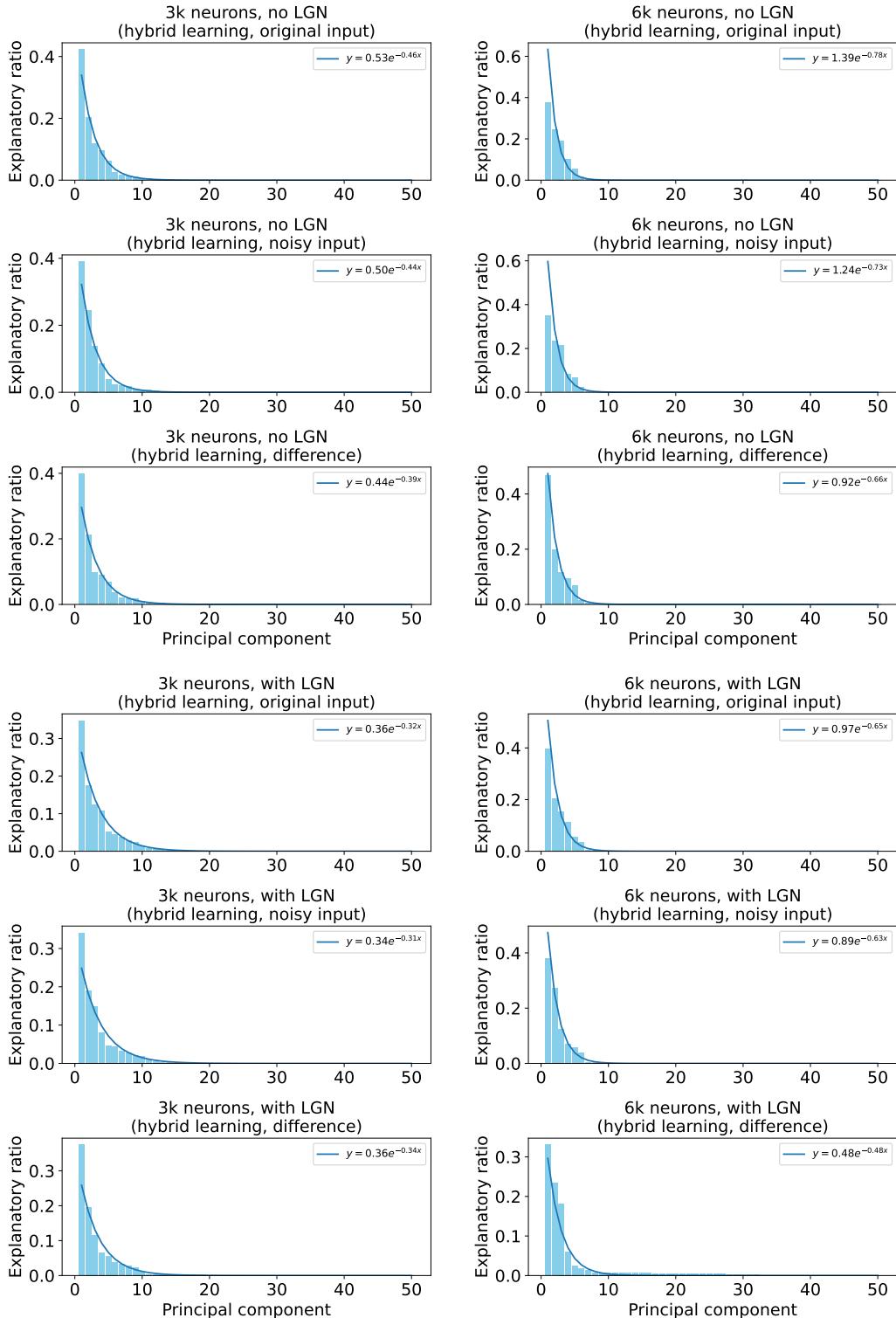
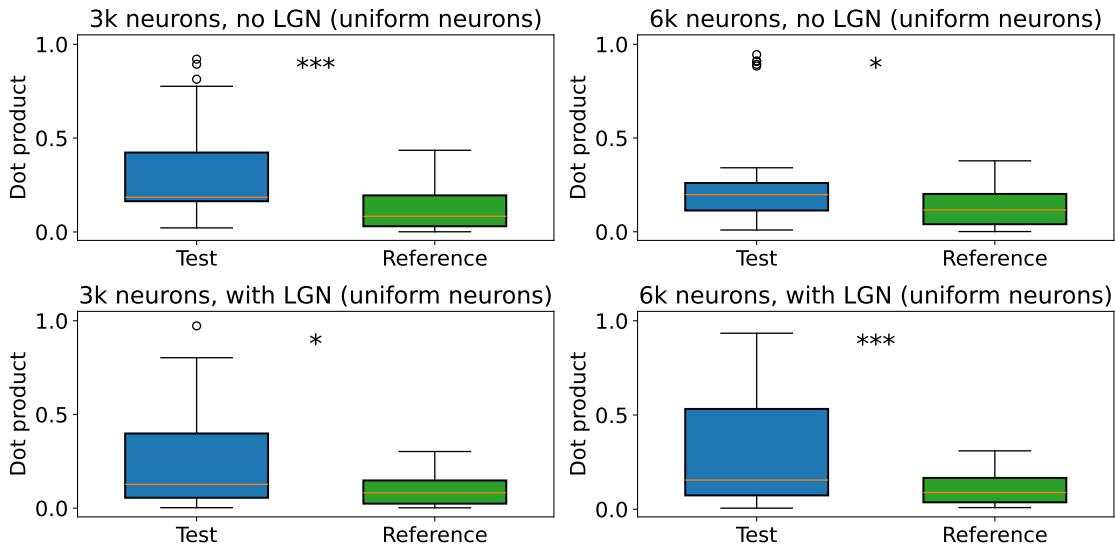


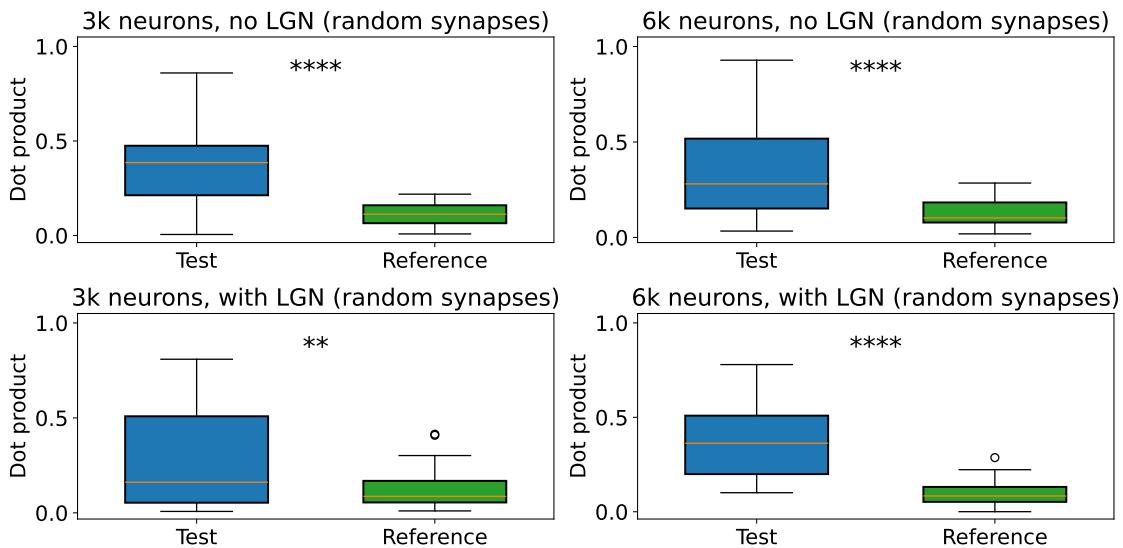
Figure F.4 PCA results of neuronal responses to clean and noisy MNIST images and of the response difference (hybrid learning)

APPENDIX F SUPPLEMENTARY NEURONAL RESPONSES RESULTS



The layout and labeling are same as the description in Figure 4.20

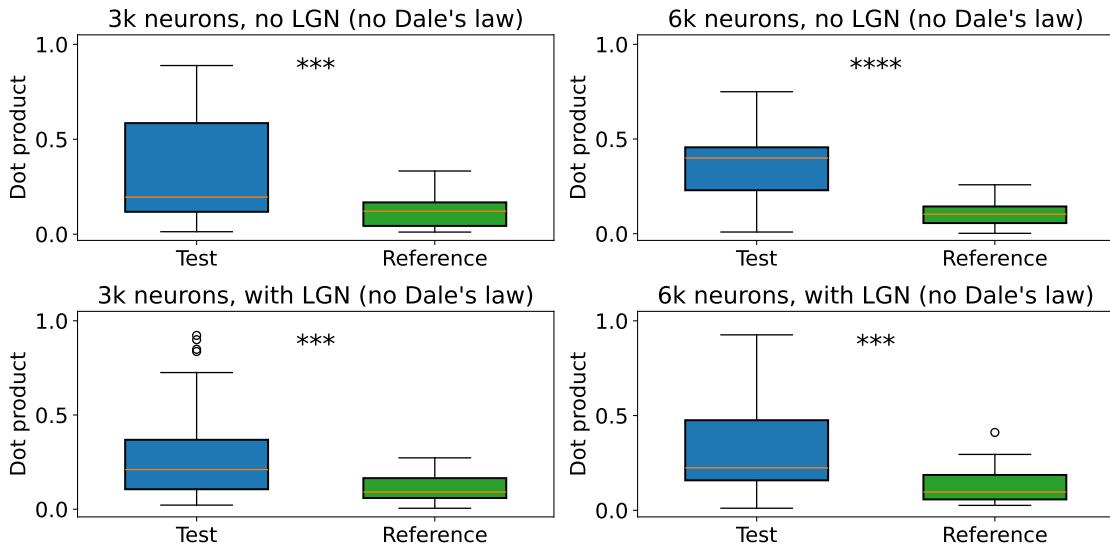
Figure F.5 Significance analysis of dot product of between top 5 principal components of neuronal response to original and noisy input Significance analysis using dot product between principal components of neuronal responses to original and noisy inputs (uniform neurons)



The layout and labeling are same as the description in Figure 4.20

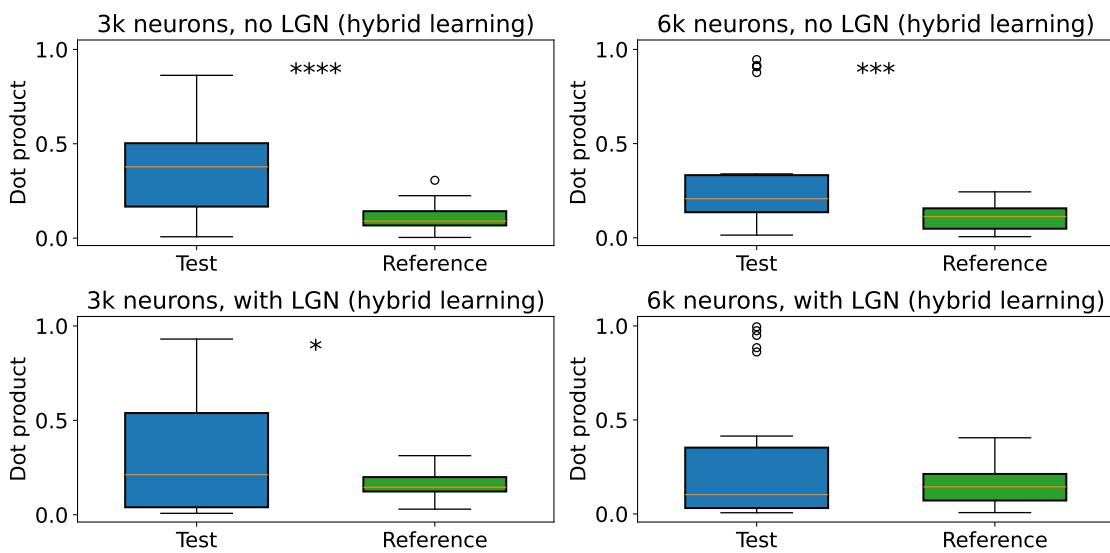
Figure F.6 Significance analysis of dot product of between top 5 principal components of neuronal response to original and noisy input Significance analysis using dot product between principal components of neuronal responses to original and noisy inputs (random synapses)

APPENDIX F SUPPLEMENTARY NEURONAL RESPONSES RESULTS



The layout and labeling are same as the description in Figure 4.20

Figure F.7 Significance analysis of dot product of between top 5 principal components of neuronal response to original and noisy input Significance analysis using dot product between principal components of neuronal responses to original and noisy inputs (no Dale's law)



The layout and labeling are same as the description in Figure 4.20

Figure F.8 Significance analysis of dot product of between top 5 principal components of neuronal response to original and noisy input Significance analysis using dot product between principal components of neuronal responses to original and noisy inputs (hybrid learning)

ACKNOWLEDGEMENTS

First, I would like to express my sincere gratitude to my advisor, Prof. Sen Song, for his invaluable guidance, insightful ideas, and unwavering support throughout my research journey in computational neuroscience. His expertise, patience, and encouragement have been instrumental in shaping this work.

I am also deeply grateful to my friend, Jinyu Zang, for the countless fruitful discussions, valuable feedback, and assistance he provided during the course of this research. His willingness to lend a helping hand and share his knowledge has been truly appreciated.

I would like to thank the Tsinghua-Johns Hopkins BME dual degree program for providing me this unique opportunity to study and research with so many talented minds.

Furthermore, I would like to acknowledge the contributions of all those who have directly or indirectly supported me in this endeavor, be it through their advice, resources, or moral support.

COMMENTS FROM THESIS SUPERVISOR

This thesis presents a comprehensive and insightful study on firing-rate neural network models for the primary visual cortex (V1) and their learning capabilities in the context of visual processing tasks. The research is well-motivated, aiming to develop computationally efficient and biologically plausible models to explore the neural processes underlying visual perception.

Huimiao Chen has done an excellent job in building the discrete form firing-rate models and applying them to the learning of V1 neural networks using real experimental data. The designed visual tasks - fine-orientation detection, image classification, and visual change detection - provide a solid framework for evaluating the learning capabilities of the models under various scenarios and settings.

The analysis of synaptic weight distributions, comparison between 3k- and 6k-neuron models, and investigation of the role of lateral geniculate nucleus (LGN) filters demonstrate Huimiao's strong analytical skills and deep understanding of the subject matter. The findings regarding the benefits of diverse synaptic weights, enhanced network connectivity, neuronal diversity, and LGN preprocessing offer valuable insights into the workings of V1 and its role in visual processing.

Furthermore, the discussions on model robustness to internal and external noise, the effectiveness of LGN filters, and the balance between representational precision and noise resilience showcase Huimiao's ability to interpret results in a broader context and draw meaningful conclusions. These insights contribute to our understanding of the trade-offs and adaptations in biological neural systems.

To further expand on this work, several future directions can be considered. First, exploring more biologically realistic neuronal dynamics could enhance the rate models' resemblance to their biological counterparts. Second, a detailed study of network recurrence and its role in robustness could provide additional insights into the stability and resilience of neural systems. Finally, investigating the impact of different learning rules on the performance of rate models could offer a more comprehensive understanding of the learning mechanisms in V1.

Overall, this thesis is a valuable work in the field of computational neuroscience and neural modeling. Huimiao has demonstrated a strong command of the subject, excellent

COMMENTS FROM THESIS SUPERVISOR

research skills, and the ability to conduct independent and original work. The thesis is well-structured, clearly written, and thoroughly referenced.

I recommend Huimiao Chen for the Master's degree in Biomedical Engineering. Huimiao has met all the requirements for the degree, and I believe this work lays a solid foundation for further research in biologically plausible visual processing and neural modeling.