UNIVERSITY OF CALIFORNIA,
IRVINE

A Computational Investigation of Cortical Motion Perception and Hippocampal Spatial Memory

DISSERTATION

submitted in partial satisfaction of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

in Cognitive Sciences

by

Kexin Chen

Dissertation Committee:
Professor Jeffrey Krichmar, Chair
Professor Mimi Liljeholm
Professor Aaron Bornstein

2023

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Page

LIST OF TERMS

ANN Artificial Neural Network. A type of computational model that is modeled after the structure of the biological neural network, which can perform learning through modifications of the connection weights between neurons..

SNN Spiking Neural Network. A type of artificial neural network that models the network activity using spiking events. In addition to the number of spikes, it also incorporates temporal information of the neuronal activity..

Sparse coding An encoding scheme that efficiently represents a set of data by achieving a balance between minimal population activation towards a stimulus, and adequate coverage of the stimulus space by individual units..

Dimensionality reduction A computational process that reduces the number of features in a representation space while preserving the variance in the original data. In neural networks, this can be achieved by projecting the input layer into a layer of neurons with a smaller size, and using the activity of these neurons to reconstruct the original input data..

NSC Nonnegative Sparse Coding. A computational method that achieves dimensionality reduction by decomposing the data into a set of sparsely activated nonnegative components. NSC is suggested to be employed by many sensory areas in the brain to encode external stimuli..

Evolutionary computation A type of computational approach inspired by the Darwinian evolutionary theory for solving complex problem..

Evolutionary algorithm A specific type of evolutionary computation algorithm that follows a process in which the highest-performing members of a population survive and reproduce the next generation population in order to find optimal solutions to a given problem..

STDP Spike-timing-dependent plasticity. A type of synaptic plasticity which modifies connection weights between a pair of neurons based on the relative timing of spikes between pre- and post-synaptic neurons..

STP Short-term plasticity. A type of synaptic plasticity observed in the brain which induces temporary changes in synaptic strength over a short time scale, ranging from hundres to thousands of milliseconds..

Plaid A type of visual stimulus that consists of two or more superimposed sinusoidal gratings. As the plaid stimulus drifts in one direction, the component gratings may appear to move in other directions, causing ambiguity in perceiving the motion of the plaid stimulus..

Optic flow Changes of structured light on the retina caused by the relative motion between an observer and the scene during locomotion of the observer. Optic flows contain important self-motion information including the heading direction and the moving velocity of the observer..

Triple-T maze A track environment that can be used to conduct complex spatial working memory tasks. In a Triple-T navigation task, the animal starts at the bottom part of the maze and is required to traverse through four partially overlapping routes sequentially with no repeats..

Cognitive map A mental representation of the spatial structure of the environment, and the spatial relationship among the agent, the navigation space, and objects within the environment..

Place cell A type of hippocampal neurons that have an activity bump only when the animal was in certain locations in the environment. Spatial locations in the environment can be mapped by the population activity of these place cells..

Analogy cell A sub-population of subiculum neurons that exhibit multiple firing fields which correspond to locations that are spatially separated, but share structural or functional similarities, such as head direction, axis of travel, and progression through a route..

Axis-tuned cell A subset of subiculum neurons that have bimodal activity bumps that correspond to two heading directions 180°apart, encoding the axis of travel. Axis-tuned responses were observed in a Triple-T maze, and may be a special type of analogous representations found in the subiculum..

CA1 A sub-region of the hippocampus which sends major hippocampal outputs to cortical areas. CA1 is heavily involved in the formation, consolidation, and retrieval of hippocampus-dependent memories..

SUB Subiculum. A brain region that is located in the hippocampal formation that receives direct synaptic inputs from the hippocampal sub-region CA1 and projects to various cortical and subcortical areas involved in spatial navigation and memory..

GPU Graphics Processing Units. A type of processor that is widely used in machine learning applications because of their efficiency in performing parallel computations in neural networks..

ACKNOWLEDGMENTS

I am deeply grateful to the following individuals and organizations for their contribution to the completion of my dissertation:

First and foremost, I would like to express my sincere appreciation to my advisor, Professor Jeffrey Krichmar, for his endless patience and continual support throughout my graduate studies. His dedication to research and compassion for people have been a source of inspiration for me. I am also grateful to my committee members, Professor Aaron Bornstein and Professor Mimi Liljeholm, for their invaluable insights and expertise. Furthermore, I extend my thanks to Professor Douglas Nitz, with whom I have had the privilege of collaborating since my undergraduate years.

I would also like to acknowledge the significant contributions made by my co-authors to the research presented in this dissertation. In particular, Lars Niedermeier's development and improvement efforts on CARLsim 6, which are described in Chapter 3, and Professor Michael Beyeler's foundational work on the MSTd, which forms the basis of the research project presented in Chapter 4. I am also grateful for the data collected and analysed by Dr. Alexander Johnson and Dr. Jacob Olson, which were fundamental to the modeling study described in Chapter 5. I am thankful for the support provided by Professor Kenneth De Jong and Dr. Eric Scott in the application of evolutionary algorithms, which were utilized throughout the modeling studies in Chapters 4 and 5.

I also express my gratitude to the members of the Cognitive Anteater Robotics Laboratory (CARL) - Dr. Tiffany Hwu, Dr. Hirak Kashyap, Dr. Xinyun Zou, Jinwei Xing, Amirhosein Mohaddesi, Tim Lui, Harrison Espino, Robert Bain, and Nickolas Alonso, for their camaraderie and helpful discussions about my research.

I am deeply appreciative of my cohort members, including but not limited to Xiaojue Zhou, Priyam Das, Paulina Silva, Nidhi Banavar, and Aakriti Kumar, for creating a supportive community and sharing the ups and downs of graduate school.

I thank IEEE for permission to include Chapters 3 and 5 of my dissertation, which were originally published in the Proceedings of the International Joint Conference on Neural Networks (IJCNN 2022 and IJCNN 2021). I also thank Society for Neuroscience for permission to include Chapter 4 of my dissertation, which was originally published in the Journal of Neuroscience. I am grateful to the National Science Foundation (NSF) and the Air Force Scientific Office of Research (AFOSR) for their financial support, which made this research possible.

I would also like to thank my friends, Jingyi Huang, Debbie Chen, Catty Du, Phoebe Cui, Yizi Lin, Changzhe Xu, and Ziye Wu, for their encouragement and support. I am fortunate to have them in my life, and I look forward to cherishing our friendship for years to come.

Finally, I want to express my heartfelt appreciation to my parents, Zida Chen and Qixian Li, and my brother, Haoyao Chen, for their unconditional love and support, and for always believing in me. I also thank my significant other, Yung Yang Lan, for his love, understanding, and patience, which sustained me throughout this journey.

VITA

Kexin Chen

EDUCATION

Doctor of Philosophy in Cognitive Neuroscience **2018-2023**
University of California, Irvine *Irvine, CA*

Bachelor of Science in Cognitive Science **2014-2017**
University of California, San Diego *San Diego, CA*

PUBLICATIONS

Chen K, Beyeler M, Krichmar JL (2022). Cortical motion perception emerges from dimensionality reduction with evolved spike-timing dependent plasticity rules. *Journal of Neuroscience*. DOI: 10.1523/JNEUROSCI.0384-22.2022

Niedermeier, L, **Chen, K**, Xing, J, Das, A, Kopsick, JD, Scott, EO, Sutton, N, Weber, K, Dutt, N, Krichmar, JL (2022). CARLsim 6: An Open Source Library for Large-Scale, Biologically Detailed Spiking Neural Network Simulation. *IJCNN 2022*.

Kopsick, JD, Tecuatl, C, Moradi, K, Attilli, SM, Kashyap, HJ, Xing, J, **Chen, K**, Krichmar JL, Ascoli, GA (2022). Robust Resting-State Dynamics in a Large-Scale Spiking Neural Network Model of Area CA3 in the Mouse Hippocampus. *Cogn Comput*. https://doi.org/10.1007/s12559-021-09954-2

Chen K, Johnson A, Scott, EO, Zou X, De Jong KA, Nitz DA, Krichmar JL (2021). Differential Spatial Representations in Hippocampal CA1 and Subiculum Emerge in Evolved Spiking Neural Networks. *IJCNN 2021*.

Zou X, Scott, EO, Johnson A, **Chen K**, Nitz DA, De Jong KA, Krichmar JL (2021). Neuroevolution of a recurrent neural network for spatial and working memory in a simulated robotic environment. *GECCO 2021*.

Xing J, Nagata T, **Chen K**, Neftci E, Krichmar, JL. (2021) Domain Adaptation In Reinforcement Learning Via Latent Unified State Representation. *AAAI 2021*.

Chen K, Hwu T, Kashyap HJ, Krichmar JL, Stewart K, Xing J and Zou X (2020) Neurorobots as a Means Toward Neuroethology and Explainable AI. *Front. Neurorobot*. 14:570308. doi: 10.3389/fnbot.2020.570308

WORK AND RESEARCH EXPERIENCE

Graduate Student Researcher **2018-2023**
University of California, Irvine *Irvine, California*

Software Engineer Intern, Machine Learning **2021, 2022**
Facebook/ Meta *Menlo Park, California*

Software Engineer **2017-2018**
DeepRadiology *Remote, California*

Undergraduate Research Assistant **2016-2017**
University of California, San Diego *San Diego, California*

ACADEMIC SERVICES

Peer Reviewer **2023**
2023 International Joint Conference on Neural Networks (IJCNN 2023)

Co-Organizer & Program Committee **2022**
NeuroVision Workshop at CVPR 2022

Guest Editor **2022**
Biological Cybernetics Special Issue:
What can Computer Vision learn from Visual Neuroscience?

Coordinator **2020**
UC Irvine Cognitive Sciences Colloquium Committee

AWARDS

John I. Yellott Scholar Award - Honorable Mention **2020, 2019**
Google PhD Fellowship - Campus Nomination **2019**

ABSTRACT OF THE DISSERTATION

A Computational Investigation of Cortical Motion Perception and Hippocampal Spatial Memory

By

Kexin Chen

Doctor of Philosophy in Cognitive Sciences

University of California, Irvine, 2023

Professor Jeffrey Krichmar, Chair

This dissertation explores the use of computational models, specifically spiking neural network (SNN) models combined with evolutionary computation, as a versatile approach to investigate neural representations and underlying neural mechanisms involved in navigation. The aim of this thesis is to understand how information is processed and represented in the cortical motion stream for visual motion perception and in the hippocampus for spatial cognition.

The first study introduces CARLsim, an efficient SNN simulator with biologically detailed implementation of spiking neurons, synapses, and various synaptic plasticity rules. Additionally, CARLsim is integrated with evolutionary computation libraries, providing flexible and powerful modeling capabilities. This tool is utilized in the modeling studies presented in this dissertation.

In the second study, an SNN model is developed to understand the complex response properties of neurons in the dorsal sub-region of the Medial Superior Temporal area (MSTd) area and how they support heading estimation. The resulting model showed receptive fields that are suitable for processing self-motion-induced optic flows, and accurately encoded the heading direction in the neuron population. This study explores how synaptic plasticity rules could implement nonnegative sparse coding (NSC) for efficient coding, and supports a previous theory which suggests that receptive fields observed in MSTd may emerge through dimensionality reduction on its input (Beyeler et al., 2016).

The third study investigates how spatial representations that are important for navigation emerge with self-motion cues and external cues that signal the relationship between the agent and the environment. SNN models of hippocampal sub-region CA1 and the subiculum (SUB) were developed using neuronal data recorded in a working memory navigational task. The models exhibited differential spatial representations that reflect their functional distinctions and match experimental observations. This study shows that distinct representations of spatial features can be formed by different weightings in the integration of navigational variables.

This dissertation provides insights into how the brain processes information during navigation, shedding light on the neural mechanisms underlying visual motion perception and spatial cognition. The presented studies highlight the potential of SNNs and evolutionary computation as powerful tools for computational modeling of the brain, which offer the advantage of biological plausibility and minimal assumptions on the model parameters. This research may inspire the development of other novel modeling approaches. The models developed in this research could have practical applications in robotics and artificial intelligence.

Chapter 1

Introduction

Efficient navigation is an essential task for humans and other animals. This is a complex process that involves integrating various sensory modalities and cognitive processes. In particular, visual motion processing and spatial memory are critical components of navigation. Optic flow, generated as an individual moves through an environment, enables the perception of motion and tracking of movement relative to the surroundings. On the other hand, spatial memory may contain a mental map that represents the structure and components of the spatial environment. These components allow individuals to make informed decisions about their own movement, enabling successful navigation through complex environments. As a result, individuals can perform tasks such as path planning, finding shortcuts, and adjusting movements based on their perceived self-motion. Extensive data have been collected to investigate these two brain functions. The cortical motion stream is responsible for processing visual motion information, and the hippocampal area plays a critical role in spatial memory and navigation. However, the mechanisms underlying visual motion processing and spatial memory in navigation are not yet fully understood.

In recent years, computational modeling has become an increasingly powerful tool for understanding the neural mechanisms underlying complex behaviors, such as navigation. By developing

models that are structurally and/ or functionally similar to the brain, researchers can test various hypotheses and predict behavior under different conditions. This approach enables understanding of data, identification of patterns, verification of hypotheses, and generation of testable predictions about the brain. Notably, Spiking neural network (SNN) models offer an advantage in describing biological details of the biological neural network by representing neural activity with spike trains, which contain information about the timing and frequency of spiking activity, rather than the average firing rate of the neuron. SNNs also allow for the investigation of spike-based neural plasticity, which has been observed in the brain and suggested to play important roles in learning and memory. To optimize the performance of SNN models, Evolutionary computation provides a flexible approach which selects solutions that lead to a higher score in the defined fitness function. The combination of evolutionary computation with SNNs and synaptic plasticity rules has been demonstrated to be a successful method for modeling the brain (Rounds et al., 2018; Elbrecht et al., 2020).

The aim of this dissertation is to explore neural mechanisms of visual motion processing and spatial memory in navigation using computational modeling, specifically, a combination of evolutionary computation with SNNs and synaptic plasticity rules. This research focuses on investigating how the brain efficiently represents visual motion information, encodes self-motion parameters for navigation, and integrates navigational variables to develop different forms of spatial representations that are relevant to navigation.

To achieve these goals, this research begins with enhancing the features of CARLsim, an SNN simulator, to allow for a user-friendly modeling framework and an efficient implementation of neural dynamics, as well as flexible implementations of biologically observed synaptic plasticity mechanisms. This dissertation also describes modeling studies of cortical motion perception and spatial representations in the hippocampal area, which reproduced neuronal activities observed in the brain and made novel predictions of the modeled areas.

The contributions of this research include: 1) enhancement of an open-source modeling tool, 2) the development of biologically detailed models of cortical motion perception and spatial representations, which captured key response properties of neurons in the target brain regions, 3) a computational demonstration of how spike-timing-dependent plasticity (STDP) could perform dimensionality reduction and sparse coding to achieve an efficient encoding of the input space, and 4) a prediction that STDP may also perform feature selection on locomotion and external cues to form different spatial representations important for navigation.

1.1 Organization

The dissertation is organized as follows:

Chapter 2 provides background information by reviewing relevant literature. First, it gives a general overview of computational modeling, SNN, and evolutionary computation, highlighting their application in neuroscience research. Secondly, it briefly reviews the dorsal visual stream and the motion processing pathway, discussing computational efforts towards understanding response properties in these regions. Finally, it briefly reviews evidence of spatial cognition in the rodent brain, particularly those found in the hippocampal formation, and presents computational and theoretical models that were developed to understand them.

Chapter 3 introduces CARLsim, which is an open-source spiking neural network simulator that was used to create and simulate models presented in Chapter 4 and Chapter 5. This chapter provides an overview of the sixth release of CARLsim, introduces new features, presents simulation results, and discusses related work.

Chapter 4 focuses on a computational investigation of optic flow processing in MSTd using SNNs with evolved neural plasticity rules. Neurons in this model developed receptive fields that preferred a mixture of motion patterns, similar to those observed in the MSTd. This chapter demonstrates

that neural plasticity, specifically STDP, may allow the brain to perform sparse coding and dimensionality reduction on high-dimensional input to represent information efficiently.

Chapter 5 presents SNN models of two hippocampal regions, CA1 and subiculum, which showed differing encoding schemes of navigational features. The same modeling approach as in Chapter 4 was utilized in developing this model. Although the SNN models of these two hippocampal regions shared the same network architecture and optimization method, they evolved to integrate behavioral variables differently, leading to different spatial representations which resembled those observed in their biological counterparts.

Chapter 6 presents the main conclusions of the dissertation, including a discussion of the implications of the research studies in relation to existing literature. It also provides a discussion of future directions for extending the studies.

Chapter 2

Background

2.1 Computational modeling of the brain

Computational methods have become increasingly important in the field of neuroscience for modeling the complex dynamics of the brain. In particular, mathematical expressions can be used to describe relationships between variables, and computational models enable simulations that help to understand the interactions between components of the neural system. By manipulating elements within these computational models, we can observe how the alteration of components lead to changes in the system, enabling us to draw useful conclusions that advance our understanding of the brain. Furthermore, computational models that closely replicate certain brain structures and exhibit functionalities of certain brain areas can serve as good representations of the biological neural network. These models offer the flexibility that allows researchers to have full control over the design of network architecture, and the ability to record the entire state of the network, including the activity of neurons and their connections.

Increasingly powerful and complex machine learning methods and model architectures have been proposed in recent years, which demonstrate the ability to perform various tasks, including object

recognition in an image, or learning to understand and generate natural languages. Neuroscientists have also recognized the potential of using these models as a way to better understand brain mechanisms (Richards et al., 2019; Vu et al., 2018). Computational models of the brain can be developed through fitting to neural data directly, or by defining an optimization goal for the model and observing properties that emerged through learning. With the advancement of deep learning, the second approach has become increasingly popular. One example of this approach involves training a deep neural network to perform path integration (Banino et al., 2018). This model helps explain that grid-like representations of an environment may emerge from the learning process of performing vector navigation. Another example of this modeling approach involves training a deep neural network to estimate self-motion velocities. This model helps explain that response properties of the dorsal visual stream, which is responsible for processing visual motion, may emerge as a result of estimating self-motion parameters (Mineault et al., 2021).

Although deep learning models provide new opportunities for studying functionalities of the brain, they often lack explainability, leading them to be regarded as “black boxes" just like the biological brain. In contrast, models with simpler architecture offer the advantage that allows us to learn precisely how model elements, for example, how each neuron and connection contributes to the final output of the network. In a recent computational study, Rideaux et al. (2021) presented a model of the motion processing stream which contains only a few layers of neurons in a feed-forward network model. This model received visual motion sequences and velocity information as input and learned to estimate velocities of self-motion and scene motion, as well as whether the visual and vestibular signals came from the same event. After the model successfully completed the task, it was tested with stimuli similar to those used in neurophysiological studies, and the activity of neurons in the MSTd layer was recorded. As an emergent property of the model, some neurons in the model developed a preference for the same visual and vestibular heading directions, while other neurons preferred opposite visual and vestibular directions, similar to the brain’s congruent and opposite cells (Gu et al., 2008). With a thorough analysis of the connection weight values and experiments involving the lesion of congruent of opposite cells, the authors confirmed congruent

cell's role in estimating self-motion, and made a novel prediction about opposite cell's role in estimating scene motion.

In Chapters 4 and 5, I present computational models of a cortical area that is involved in visual motion perception, and regions in the hippocampal formation that play critical roles in spatial cognition. I show that these models accurately replicate key response properties of neurons in these brain areas. Additionally, I test these models using tasks similar to those used in physiological studies, and perform lesion studies to investigate how certain input variables contribute to the emergent properties of the model. These experiments demonstrate the explainability of these computational models, and show their potential to generate helpful predictions about the mechanisms of the brain.

2.1.1 Artificial neural networks (ANNs) and Spiking neural networks (SNNs)

Artificial neural networks (ANNs) are a widely used tool for computational modeling due to their ability to represent information processing in biological neural networks. ANNs are generally composed of interconnected neurons that can transmit information through their connections with modifiable weight values. One form of representing the activity of neurons in ANNs is through rate-based encoding, which approximates the neuron activity by their average firing rates. However, a more biologically plausible representation of real neurons uses spiking events to represent the neuron activities.

Using spiking neurons, Spiking neural networks (SNNs) capture more details of the biological neurons than rate-based networks. In SNNs, a spiking neuron accumulates incoming currents, and when the membrane potential reaches the threshold, the neuron fires a spike. The size and shape of a spike are not affected by the input, but the timing of a spike depends on the input. SNNs encode information not only in the number of spikes but also in the temporal aspect of the spikes, allowing for more sophisticated non-linear computations than rate-based networks. SNNs have been used

to investigate sensory processing in the brain (Xu et al., 2020), to model brain activities (Kasabov, 2014), as well as to replicate neuronal activities (Rounds et al., 2018).

Different spiking neuron models have been proposed to capture the complex dynamics of real neurons. The simpler integrate-and-fire neuron model computes the membrane potential by integrating all input information including synaptic input and injected currents. Once the membrane potential reaches a threshold, a spike is generated, and the membrane potential is reset. Variants of the integrate-and-fire neuron include the leaky integrate-and-fire (LIF) neuron, where an exponential decay of membrane potential is incorporated (Dutta et al., 2017). The Hodgkin-Huxley model represents a more complex neuron model, which uses voltage-gated ion channels to describe the state of a neuron (Hodgkin and Huxley, 1952). The Hodgkin-Huxley model describes the biological mechanisms in more detail, but is also more computationally demanding. The Izhikevich model combines the computational efficiency of integrate-and-fire neurons with the biological plausibility of the Hodgkin-Huxley model. This model describes the state of a neuron using 4 or 9 parameters and is able to reproduce spiking and bursting behavior of cortical neurons (Izhikevich, 2004).

Connections in ANNs mimic the axon and dendrites of the biological neurons. The strength of the connection between a pre-synaptic neuron and a post-synaptic neuron determines the amount of information received by the post-synaptic neuron. The changes in connection strength or “weights" happen through learning. Back-propagation is a common learning method used in modern ANNs, which updates connection weights based on the difference between the predicted output and the desired output, and the error signals are transmitted back through the network to neurons that are several synapses away. However, current biological evidence suggests that backpropagation may not be implemented in the brain. To better understand the network dynamics during learning and to have a more direct comparison between computational models and recorded neural data, it is beneficial to employ learning rules that have been observed in the brain. Two such synaptic plasticity rules are short-term plasticity (STP) and spike-timing dependent plasticity (STDP).

STP is a type of synaptic plasticity that induces temporary changes in synaptic strength over a short time scale, typically on the order of hundreds to thousands of milliseconds (Tsodyks and Wu, 2013). Short-Term Depression (STD) decreases synaptic strength through depletion of neurotransmitters, and Short-Term Facilitation (STF) increases synaptic strength through an increase release probability of subsequent neurotransmitters due to the influx of calcium into the axon terminal after spike generation. STDP is another type of synaptic plasticity, which modifies connection weights based on the relative timing of spikes between pre- and post-synaptic neurons (Sjöström and Gerstner, 2010). If a pre-synaptic neuron fires shortly before a post-synaptic neuron, the connection between them is strengthened. Conversely, if the pre-synaptic neuron fires shortly after the post-synaptic neuron, the connection is weakened. Both STP and STDP have been observed in multiple brain regions (Markram et al., 1998; Wang et al., 2006; Sjöström et al., 2008; Caporale and Dan, 2008). Incorporating STP and STDP into SNNs can help to better understand the mechanisms of learning and plasticity in the brain, and can lead to the development of more biologically realistic and computationally sophisticated models of the brain.

Simulating SNNs that capture a considerable amount of biological detail involves lengthy and complex computational processes that require careful network state update and memory allocation. To address these challenges, a number of SNN simulators have been developed, including CARLsim (Chou et al., 2018; Balaji et al., 2020), NEURON (Carnevale and Hines, 2006), Brian (Stimberg et al., 2019), and NEST (de Schepper et al., 2022). Compared to other SNN simulators, CARLsim is advantageous because it supports parallel simulations with multiple GPUs and CPUs, allowing for fast computation in large-scale SNN simulations. Additionally, CARLsim features an automated parameter tuning interface (PTI), which can be bridged to external libraries for evolutionary computation such as Evolutionary Computation in JAVA (ECJ) (Luke, 2017) and Library for Evolutionary Algorithms in Python (LEAP) (Coletti et al., 2020). Hyper-parameter tuning in SNNs with evolutionary algorithms usually involves simultaneous evaluation of multiple individual networks, which requires considerable computing resources. CARLsim's ability to allocate individual networks or parts of the networks to multiple CPUs and GPUs significantly

accelerates the simulation process. This is especially beneficial when running SNN simulations on high-performance computing clusters and laboratory workstations, which contain arrays of CPUs and GPUs (Chou et al., 2018). Chapter 3 introduces the new features integrated into the latest release of CARLsim, which further extends the biological details of SNN simulations in the setup of the STP and STDP plasticity update rules, and provides more flexibility for SNN simulations that involve various neuromodulation setups.

2.1.2 Evolutionary algorithms and their applications on neural network optimization

Evolutionary algorithms were developed upon the Darwinian evolutionary theory and describe a process in which the highest-performing members of a population survive and reproduce the next generation population (De Jong, 2006). Evolutionary algorithms follow a general framework: a population of size m evolves over time, with a number of individuals selected from the current population to act as parents and produce n offsprings. This expanded population of size $m+n$ is then reduced to m individuals to keep the size of the population constant. The values for the parent population size m and offspring population size n should be carefully chosen (Jansen et al., 2005). In general, larger values of m should be chosen for more complex optimization problems. The selection of values for n reflects a balance between exploration and exploitation. As the offsprings were generated from the parent population, the larger the value of n, the more influence the parent population has on the next generation of population, and thus the less explorative the algorithm will be. Apart from choosing appropriate values for m and n, it is also important to choose effective methods for the following steps in the process of evolving a population: selection and reproduction.

The selection stage is involved in both the process of selecting the parent individuals as candidates to produce offsprings, and the process of selecting m individuals among the $m+n$ population to survive for the current generation. There are two commonly-used selection techniques: tournament

selection and fitness proportionate selection. In tournament selection, k individuals are randomly selected to compete with each other, and the best performing individual in these k competitors is then selected for reproduction (Fang and Li, 2010). This process is repeated until the desired number of offsprings have been produced. In fitness proportionate selection, the probability of an individual being selected for reproduction is proportional to its fitness value relative to the fitness values of other individuals in the population (Blickle and Thiele, 1996). Tournament selection generally has a stronger selection pressure than fitness proportionate selection, as only the best individuals are selected in the tournament competitions. On the other hand, in fitness proportionate selection, a lower-fitness individual still has a probability of being selected for reproduction.

Reproductive mechanisms can be achieved through mutation or recombination (Eiben and Smith, 2015). Mutation is a single-parent reproductive mechanism that introduces small variations in the parent's genes to produce offspring. These variations can be implemented through mutation operators such as bit-flip or Gaussian mutation. Recombination, typically through crossover operators, produces offspring by combining genes from two or more parents. Mutation facilitates exploration by introducing new genes, while recombination exploits existing knowledge and searches in the space defined by the gene values of the parent individuals.

Evolutionary algorithms represent a versatile approach with very few assumptions about the task structure and has various applications in optimizing neural networks, a process known as neuroevolution (Lehman and Miikkulainen, 2013; Miikkulainen, 2010). The optimization could be done through evolving network parameters (i.e. connection weights), or even network structures, such as the number of neuron layers, the number of neurons per layer, or the connection topology between neurons. These network features could be mapped directly to the genetic encoding of the evolutionary algorithm. However, this direct encoding approach suffers from scalability issues. When the network becomes more complex, the number of variables to be encoded can increase, making it difficult to search through the parameter space thoroughly. An alternative approach is indirect encoding, in which the genetic encoding defines a set of parameters that can be used to

generate a solution, instead of mapping the genes directly to network components. This approach provides an efficient and compact representation of the process to construct a network (Lehman and Miikkulainen, 2013).

A number of previous studies have demonstrated that evolutionary algorithms provide a powerful tool for constructing and optimizing SNNs. Pavlidis et al. (2005) showed that by directly evolving the weight vectors, they could generate SNNs with a generalization ability comparable to that of a standard multi-layer perceptrons trained using gradient descent-based algorithms. For understanding and processing spatio-temporal brain data, Kasabov (2014) proposed an SNN model architecture called NeuCube, which was based on a 3D evolving SNN that mapped structural and functional areas of the brain concerining spatial-temporal brain data. Evolutionary algorithms were applied to evolve for new neurons and new connections in the SNN based on the input data. Results of this study suggest that this modeling approach led to a better accuracy for classification and pattern recognition tasks, and can be applied to model various spatio-temporal brain data. Instead of directly evolving the connection weights, Rounds et al. (2018) proposed a method of evolving the parameters of the spike-timing dependent plasticity (STDP) rule in SNNs. This approach greatly reduces the search space for network parameters, and was shown to be effective for modeling the activity of individual neurons recorded in rodents while performing a navigational task. Elbrecht et al. (2020) explored the relationship between STDP and evolutionary algorithms as optimization approaches for training SNNs, and found that the combination of STDP and the evolution of network structure led to a reduced network size and an improved generalization ability, both of which are desirable characteristics in deployed SNNs. To explore interpretable plasticity rules for SNNs, Jordan et al. (2021) proposed an evolutionary approach where mathematical expressions were encoded as genotypes and evolved to complete tasks in various learning situations, including reinforcement learning, supervised learning, and error-driven learning. The evolved learning rules were interpretable and some of them closely resembled the plasticity rules observed in the brain, such as STDP. Additionally, the approach also recovered rules similar to gradient-based meth-

ods for learning from target signals, and uncovered previously unknown mechanisms for efficient reward learning.

In Chapters 4 and 5, I demonstrate that evolving synaptic learning rules in SNNs is a versatile modeling approach that could be applied to various forms of neural data, with diverse optimization objectives. The approach was successfully employed to model cortical motion perception in the macaque brain and spatial cognition in the rodent brain, resulting in models that accurately replicated crucial neuron properties in the target regions. Furthermore, these models can generate novel hypotheses for future neurophysiological investigations.

2.2 Visual motion perception

The ability to perceive motion, including one's own movement relative to the environment, and the movement of objects in the environment, allows individuals to make informed decisions about their actions during navigation. Vision is an important source for motion perception. When an observer moves through an environment, the relative motion between the observer and the scene causes changes of structured light on the retina, creating an an optic flow (Raudies, 2013). From the optic flow, it is possible to derive several key properties of the observer's self-motion, including their speed and direction of travel, as well as the rate of rotation. The estimation of optic flow has been extensively studied in computer vision, and a number of computer vision techniques have been proposed for estimating optic flow from moving images, such as the Lucas-Kanade method (Lucas and Kanade, 1981) and the Horn-Shunck method (Horn and Schunck, 1981), both of which compute spatial and temporal image derivatives to estimate optic flow components. It is intriguing how the biological brain perceives optic flows and translates them into behavioral variables relevant to navigation.

Originating in the visual cortex, two distinct visual pathways have been characterized as the “what" and “where" pathway, respectively. The “what" pathway, also known as the ventral pathway, consists of a hierarchically organized set of cortical regions, beginning with the primary visual cortex (V1) and extending to the inferior temporal (IT) cortex (Ungerleider and Pessoa, 2008; Issa et al., 2018). This pathway is specialized in the recognition of objects. The “where" pathway, also referred to as the dorsal pathway, involves areas V1, the middle temporal cortex (MT), and the medial superior temporal cortex (MST), which connects to intraparietal regions including the lateral intraparietal area (LIP), the ventral intraparietal area (VIP), and the middle intraparietal area (MIP). This pathway is specialized in the localization of objects. Despite extensive research focused on the “what" pathway, the “where" pathway has received comparatively less attention.

The visual motion processing stream is contained in the dorsal pathway. Similar to the ventral pathway, this processing stream follows a hierarchical structure, where neurons in the downstream regions have larger receptive fields and exhibit more elaborate response properties (Andersen et al., 1997). These properties allow the downstream areas to perform more complex analyses of their input signals, enabling more global and complex assessments of motion.

2.2.1 Direction selective neurons in V1

The cortical motion stream begins in V1, where direction-selective neurons were discovered (Hubel and Wiesel, 1962; Emerson et al., 1992). This subset of V1 neurons respond to different directions of movements, and have relatively small receptive fields. A simple hypothetical neural circuit known as the Reichardt detector was proposed for motion detection (Reichardt, 1961). In this model, two photoreceptors are positioned in two separate locations. The location difference results in a time difference for these two receptors to receive the light intensity input. The response signal of the first receptor is delayed by the model mechanism, and compared to the response of the second receptor through multiplication. The direction of motion can thus be derived if the delayed

response from the first receptor matches with the instantaneous response of the second receptor (Figure 2.1A) (Reichardt, 1961; Borst, 2000). The Reichardt detector demonstrates the ability to detect local motion as it is sensitive to changes in both space and time. Adelson and Bergen (1985) proposed that the problem of motion detection can be expressed as detecting oriented lines in a spatio-temporal three-dimensional space. For example, as illustrated in Figure 2.1B, the state of a stimulus that moves from left to right in a space defined by the x and y spatial coordinates can be expanded to the temporal dimension t. In this example, the movement does not involve the y axis. Therefore, the movement of the stimulus can be expressed as a slanted bar spanning the x and t axes, and the orientation of this slanted bar tells the motion of the stimulus. Analogous to how V1 neurons detect edges in static images, the selectivity of V1 neurons to the direction of stimulus movement can be regarded as detecting orientation in a spatio-temporal space (Carandini et al., 1999). Based on the concept of processing motion in the spatio-temporal frequency domain, Adelson and Bergen (1985) elaborated the Reichardt model and introduced the “motion energy" models. The basic units of this model are space-time separable filters whose impulse responses are given by the joint responses of spatial and temporal responses. Local motion energy can be computed by squaring and summing the the output of a pair of these spatio-temporal filters (Figure 2.1C). An opponent energy detector can be constructed by combining leftward and rightward energy detectors. As a result, an (R-L) detector would respond positively to rightward motion, and negatively to leftward motion. The spatio-temporal filters used in this class of models are physiologically plausible, which is supported by recordings from V1 neurons in the monkey or cat brain (Emerson et al., 1992; Rust et al., 2005; Heess and Bair, 2010).

2.2.2 Velocity selective neurons in MT

The next stage of visual motion processing involves the MT area, which receives afferent input from V1 and has been shown to play a critical role in motion perception (Britten et al., 1996). The causal link between MT neuronal activity and psychophysical judgement of motion was observed

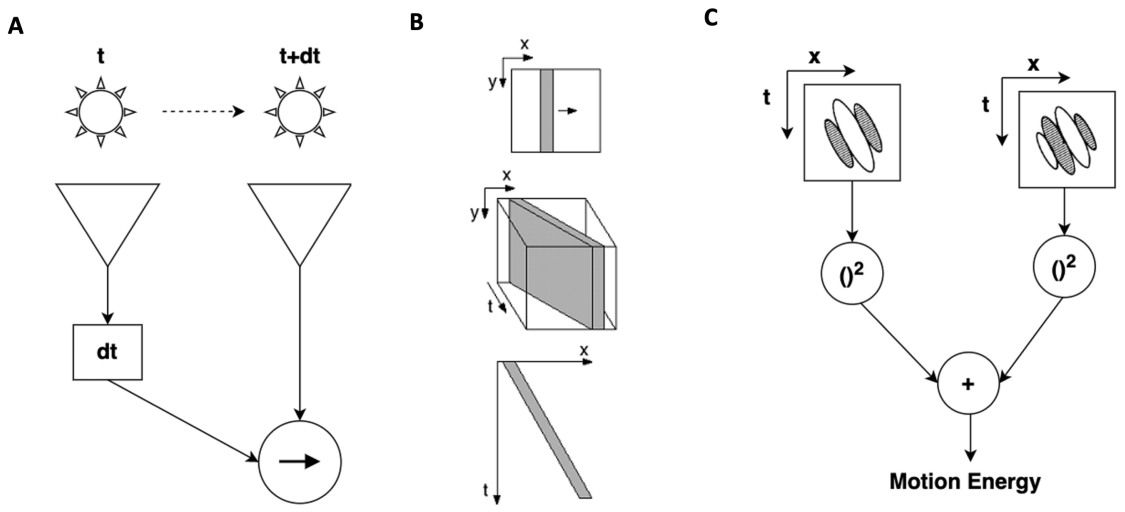


Figure 2.1: **A.** The Reichardt detector. As the object moves from left to right, the two photoreceptors receive light intensity signals with a time difference. The response signal of the first receptor is applied with a delay and compared with the response of the second receptor. A rightward motion will activate the output neuron in this model. **B.** Figure adapted from Carandini (2012). Top: a stimulus moves to the right. Middle: the motion represented in a three-dimensional spatio-temporal space. Bottom: the motion represented in a spatio-temporal space, omitting the invariant dimension. **C.** Redrawn from Adelson and Bergen (1985). The motion energy model. The response of a pair of spatio-temporal filters being squared and summed which results in a phase-independent measure of local motion energy.

in behavioral studies. For instance, by recording the performance of monkeys and neural activities of MT neurons while the animal were engaged in a psychophysical task, Newsome et al. (1989) showed that perceptual decisions regarding motion can be determined based on a relatively small number of neurons. Additionally, (Celebrini and Newsome, 1995) showed that the perceptual decisions of monkeys during psychophysical tasks can be influenced by stimulation of the MT area during the presentation of visual stimuli.

Compared to V1 direction-selective neurons, MT neurons have a more global view of the motion pattern, and are selective to the velocity of motion, including the direction and speed (Albright, 1984; Celebrini and Newsome, 1995). MT neurons have been demonstrated to effectively resolve ambiguity in motion perception. The use of plaid stimuli has enabled researchers to examine the response properties of cortical neurons to both local and global motion directions. A plaid stimulus can be constructed by superimposing two gratings with different orientations. As illustrated in Figure 2.2 A, when the plaid stimulus moves in one direction, which indicates the global motion of the plaid pattern, the component gratings appear to move in two different directions, corresponding to the local motion that conflicts with the global motion pattern. In contrast to most V1 neurons, which respond only to a single component direction, ignoring the global motion of the plaid pattern, two types of motion-selective neurons have been discovered in the MT area: component direction selective (CDS) and pattern direction selective (PDS) neurons (Celebrini and Newsome, 1995). The CDS neurons exhibit tuning curves for plaid stimuli with two distinct lobes, corresponding to the directions of motion of the two component gratings. On the other hand, the PDS neurons display a single-lobed tuning curve for plaid stimuli, which is invariant to different formation of the two components of the stimulus. Additionally, unlike V1 neurons, MT neurons were found to be strongly suppressed if two motions in different direction are present in their receptive fields (Snowden et al., 1991). This mechanism is thought to be important for removing noise in the motion stimulus, and showed that MT neurons have more complex response properties than V1 neurons (Andersen et al., 1997).

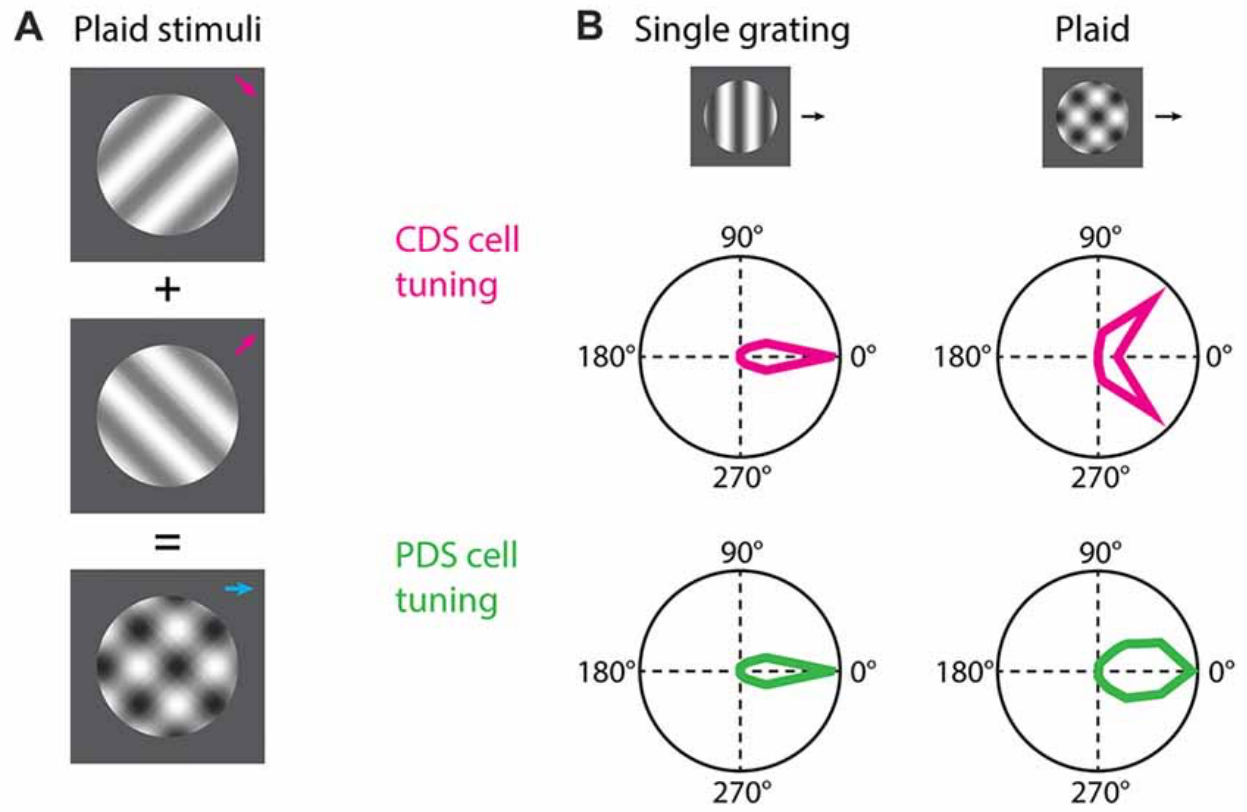


Figure 2.2: Figure adapted from Rasmussen and Yonehara (2017). **A.** Plaid stimuli. A plaid stimulus can be formed by superimposing two gratings with different orientations. As a plaid stimulus moves in a particular direction, its two component gratings appear to move in two other directions. **B.** Both CDS and PDS cells in MT respond to the direction of a drifting grating. However, for a drifting plaid stimulus, CDS cells respond to the drifting direction of the component gratings, and PDS cells respond to the "pattern direction", which corresponds to the driting direction of the plaid stimulus.

The PDS neurons appear to be key to understanding how the brain perceives coherent motion patterns despite ambiguity in local motion. Adelson and Movshon (1982) suggested that the brain utilizes the intersection-of-constraints (IOC) principle to determine global motion. Under this principle, when a plaid stimulus moves in one direction, the lines representing the motion direction of the two component gratings would intersect at a single point, which corresponds to the moving direction of the plaid stimulus. Simoncelli and Heeger (1998) proposed a computational model of area MT which incorporated a neural implementation of the IOC principle and accounted for the pattern selectivity of MT neurons. This model is based on a two-stage processing framework that involves the V1 and MT cortical areas. Both processing stages share the same basic form of computation, including the summing of all input responses, rectification, squaring, and response normalization. In the first stage, V1 neurons are modeled as spatio-temporal filters which led to direction selectivity, but no velocity selectivity. In the next stage, each MT neuron integrates the responses of a particular set of V1 neurons that have spatio-temporal tuning profiles that form certain patterns in the two-dimensional spatio-temporal frequency domain. This linear integration process is in agreement with the IOC hypothesis, and allows the MT neurons to develop velocity tuning and pattern motion selectivity. Additionally, each MT neuron sums the responses of V1 neurons whose receptive field positions fall in a local spatial neighborhood, allowing the MT neurons to have a larger receptive field than V1 neurons. Finally, the responses of the MT neurons are pooled together to provide a population code for the motion of the stimulus. This model successfully captured the pattern motion selectivity of MT neurons and accounted for much of the physiology of MT neurons, such as the suppression of response by non-preferred motions. In addition to PDS, MT CDS neurons may also be constructed in a similar framework, using a broad spatial frequency bandwidth.

A more elaborated computational model of area MT that incorporated surround suppression of V1 neurons to account for nonlinearity in visual processing was introduced by (Rust et al., 2006). In this cascade linear-nonlinear model, MT neurons linearly integrate nonlinear response of a population of direction-selective V1 cells. The direction selectivity of V1 neurons is described as von

Mises functions. Unlike the Simoncelli-Heeger model, which operates on spatio-temporal image intensities, the cascade model is restricted to inputs that are mixtures of 12 sinusoidal gratings of fixed spatial and temporal frequency, which was sufficient to capture the response properties of V1 neurons. This study compared two forms of divisive normalization in the V1 stage: 1) an “untuned" normalization that weights all stimuli equally, as implemented in the Simoncelli-Heeger model, and 2) a “tuned" normalization that gives less weight to non-preferred stimuli. The latter normalization method is consistent with surround suppression in V1 neurons, which has been shown to be a divisive operation that is selective for both orientation and direction (Cavanaugh et al., 2002). Results in this study showed that the pattern motion selectivity in the modeled MT neurons could be largely attributed to the tuned normalization component. The tuned normalization component provides a simple and biologically plausible way of modeling a wide range of the complex response properties of MT neurons. This study also suggests that, for the formation of pattern motion selectivity, pooling inputs from a broad range of preferred directions and incorporating strong motion opponent suppression may be essential.

2.2.3 Optic flow processing and heading perception in MST

Computational models based on V1-MT physiology provide an effective framework for low-level visual motion processing, allowing for the computation of maps of local motion vectors from retinal images (Nishida et al., 2018). However, the small size of receptive fields in these two brain areas suggests that they are not sufficient for processing optic flows, which span the entire visual field. On the other hand, the medial superior temporal area (MST) located in the higher levels of the dorsal stream, contains neurons with large receptive fields. In particular, neurons in the dorsal part of MST (MSTd) have receptive fields that can cover most of the visual field (Komatsu and Wurtz, 1988). Area MST receives primary inputs from MT (Raiguel et al., 1997), and is believed to play a crucial role in processing optic flow and perceiving self-motion.

In order to study the response properties of MSTd neurons to visual motion patterns, various stimuli paradigms have been employed. During these experiments, MSTd neuronal activity was recorded as animals viewed visual stimuli that simulated scenes one would encounter during locomotion. Results of early studies showed that MSTd neurons were sensitive to optic flow patterns that are related to self-motion, including expansion, contraction, translation, and rotation movements (Tanaka and Saito, 1989; Tanaka et al., 1989). Some MSTd neurons were sensitive to only one type of motion, but most neurons were sensitive to two or three components of optic flow motion patterns (Duffy and Wurtz, 1991). Graziano et al. (1994) introduced a class of stimuli that spanned a spiral space which allowed for the characterization of the continuous tuning property of MSTd neurons. The spiral coordinates represented expansion and contraction motion on the vertical axis, clockwise and counterclockwise rotation on the horizontal axis, and spiral motion such as expanding clockwise spiral along the intermediate directions. This study, which recorded 57 MSTd neurons, found a large proportion of these neurons showed a preference for expanding motion. Additionally, this study revealed that MSTd neurons showed some degree of position invariance in terms of preferred spiral motion for stimuli located in two different locations separated by a vertical displacement of 8.5°.

The evidence of MSTd neurons responding to optic flow patterns led to investigations into the role of area MSTd in visual navigation, particularly with regards to heading perception. By using electrical microstimulation to perturb the activity of MSTd neurons, Britten and Van Wezel (1998) demonstrated that the heading perception of monkeys was biased by the stimulation, indicating a direct involvement of MSTd in the perception of heading from optic flow. It might be natural to hypothesize that MSTd neurons each encodes the heading direction by using the focus of expansion (FOE) in the preferred optic flow pattern. However, the property of position invariance of MSTd tuning to motion patterns suggests that individual MSTd neurons cannot precisely encode the heading direction. Instead, it has been proposed that heading direction is encoded in broadly tuned populations of MSTd neurons (Graziano et al., 1994). Supporting this hypothesis, Hamed et al. (2003) suggested that MSTd neurons may act as basis functions to encode heading direc-

tion. Results from this study showed that the FOE in optic flow could be accurately estimated using optimal linear estimators from single-trial responses of 144 MSTd neurons, with an average error of 2-3°, which was consistent with the discrimination thresholds measured in humans and monkeys. This study suggests that downstream neurons could compute nonlinear functions of perceptual variables, such as the FOE, by linearly combining the activities of MSTd neurons. The heading preference of MSTd neurons was further characterized for both translational and rotational motion in a three-dimensional (3D) space (Gu et al., 2006; Takahashi et al., 2007). These studies revealed that MSTd neurons exhibited broad, approximately sinusoidal tuning, and that most of them had a preferred heading direction for azimuth at 0°or 180°, which corresponds to leftward or rightward movement during translation, or pitch or yaw motion for rotational movements. The overabundance of neurons preferring lateral motion was shown to be correlated with better heading discrimination around the straight forward direction (Britten, 2008; Gu et al., 2010).

Computational models have been developed to understand the visual response properties of MST neurons, which follow the visual motion processing hierarchy of V1-MT-MST. In these models, MT neurons are modeled as speed- and direction-tuned units with tuning curves similar to those observed in the brain. The response of MT units is then integrated by MST units to account for the complex response of MST neurons. Using a simple three-layer feed-forward model, Sereno and Sereno (1991) demonstrated that Hebbian learning could capture the rotation and dilation motion tuning and position invariance property of MSTd neurons. In a follow-up study, Zhang et al. (1993) modified this model by using broader cosine tuning curves in the input layer, which allowed MST units to have stable responses to their preferred motion components regardless of the magnitudes of additional complex motion components in the stimulus. These results were consistent with the decomposition hypothesis, which proposes that MST neurons decompose complex motion stimuli into their component parts. However, MSTd neurons were found to be tuned to a continuum of motion patterns in the spiral space, thus not supporting the idea that these neurons decompose optic flows into component motions (Graziano et al., 1994). To understand the underlying structure of MST neuron receptive fields, Yu et al. (2010) recorded MSTd responses to full-field optic flow

stimuli and local patches of these optic flow patterns. They modeled the local motion responses with Gaussian mixture models and found that a linear combination of the local responses did not account for global responses well. Instead, adding response gain modulation significantly improved the fit to the data, suggesting that complex interactions may occur among MST inputs. Similarly, Mineault et al. (2012) argued that nonlinearity is necessary to fully explain the complex visual selectivity of MST neurons. Specifically, they observed that a linear model was effective in capturing translational selectivity but not for spiral stimuli. By adding a parameter that accounts for compressive responses of the MT output, they found that the MST units had a strong tuning to both translation and spiral stimuli. They suggested that this compressive mechanism could be implemented through synaptic depression at the MT-MST connections, or could also be a form of normalization that resembles surround suppression in MT.

Beyeler et al. (2016) proposed a model that accounted for the 3D translation and rotation selectivity of MSTd neurons. They applied a dimensionality reduction method known as nonnegative matrix factorization (NMF) to a matrix of MT responses, and decomposed this matrix into two smaller matrices, which represented the MT-MST connection weights and the activation of MSTd units. Through this method, MSTd-like receptive fields emerged from the model, showing preferences to mixtures of translational, rotational, and deformational flow components. Furthermore, the MSTd units displayed 3D translational and rotational selectivity similar to those observed neurophysiological studies (Takahashi et al., 2007). Results in this study suggest that the complex response properties of MSTd neurons may be a by-product of MSTd neurons performing dimensionality reduction on their inputs. In a different modeling study by Mineault et al. (2021), deep neural network models that were optimized to predict self-motion parameters, including walking speed and head rotation, could learn motion representations that closely resembled those found in the dorsal visual stream. The networks were able to account for neuronal responses in areas V1, MT and MST. These findings suggest that the visual motion properties in the dorsal stream may be explained by the brain's optimization towards an ecologically relevant cost function of orienting oneself during self-motion. Correspondingly, in Chapter 4, I demonstrate that MSTd-like response

properties, including selectivity to spiral motion stimuli, and 3D translational and rotational heading selectivity, emerge in a neural network model that is optimized to reconstruct input optic flows. This study suggests that the complex visual response properties observed in MSTd neurons may be relevant to the brain's efficient representation of the visual space for achieving effective visual motion perception.

Computational models of MSTd have employed different ways of decoding heading direction. Perrone and colleagues proposed a class of template-based models in which heading was directly encoded by individual neurons as heading detectors (Perrone, 1994; Perrone and Stone, 1998). Similarly, in the model proposed by Layton and Browning (2014), MSTd subpopulations each encode a heading direction, with different units within subpopulations tuned to different motion patterns in a spiral space, and the maximally active subpopulation signaling the perceived heading direction. Another proposed model by Grossberg et al. (1999) implemented log polar cortical magnification on retinal images before they were processed by the visual motion stream. The modeled MSTd units developed representations that encoded preferred heading direction, allowing for accurate heading perception without using complex templates. In line with neurophysiological evidence (Hamed et al., 2003), the heading/ FOE was decoded from the population activity of MSTd units in the sparse decomposition model (Beyeler et al., 2016). This population decoding method is also used in the SNN model presented in Chapter 4 of this dissertation.

In summary, the visual processing pathway follows a hierarchical scheme, where each subsequent level integrates the output from the previous level, resolving conflicts and extracting global information. In Chapter 4, an (SNN) model of MSTd is introduced to provide insight into how the brain processes optic flow. This model integrates the local motion response of MT-like speed and direction selective neurons, capturing a range of response properties observed in macaque MSTd neurons. The model demonstrates a neurobiologically plausible implementation of the sparse coding and dimensionality reduction processing of optic flows proposed in Beyeler et al. (2016), and accurately encodes heading using a population code.

2.3 Cognitive maps

Efficient representations of environmental features is crucial for effective navigation. Behavioral experiments on maze learning in rats have demonstrated that rodents possess remarkable abilities to learn and infer about the spatial structure of their environment, creating an integrated map of the environment in their brain that they can utilize for navigation. These "cognitive maps" allow animals to locate themselves and other items in an environment and move between places through available routes, even when familiar routes to a goal are blocked (Tolman, 1948). The response of specific cells in the brain may represent certain components of episodic memory, which maps out features in the environment useful for navigation.

The hippocampus has long been known for its role in memory, particularly in the context of spatial navigation. For instance, rats with lesions in the hippocampus showed significant impairments in a task where they were required to locate and arrive at a previously found platform hidden in a large circular pool of water (Morris et al., 1982). The rodent hippocampal formation, consisting of the Cornu Ammonis (CA) fields, dentate gyrus (DG), the subicular region (SUB), and the entorhinal cortex (EC), was suggested to represent important components of the cognitive maps (O'Keefe and Nadel, 1979). During navigation, representations of location and orientation are essential for path planning. For example, when positional information about the animal's current location and destination, as well as the head direction of the animal the is known, a vector corresponding to the most efficient path to the target location can be computed (McNaughton et al., 2006).

The first piece of evidence that revealed how cognitive maps may be represented in the brain was the discovery of place cells in the rat hippocampal sub-region CA1. These neurons show an activity bump only when the animal is in certain locations in the environment (O'Keefe and Dostrovsky, 1971). Spatial locations in the entire environment can thus be mapped by the population activity of these place cells. Similar to place cells which show sharply tuned spatial firing, grid cells in the entorhinal cortex have multiple firing fields and these firing fields form a grid that tiled the entire

environment (Hafting et al., 2005). Representations of the animal's head direction in relation to the environment were found in the rat post-subiculum. These neurons are known as head direction cells that encode the animal's head direction independent of the animal's position (Taube et al., 1990). Furthermore, spatial representations in many other forms have been observed in the hippocampal formation, including object vector cells that signal the animal's distance and direction to objects in the environment (Høydal et al., 2019), boundary vector cells that fire when the animal is in close to the borders of the environment (Lever et al., 2009), and so on. These representations provide important insights into how the brain encodes spatial relationships in different contexts (Figure 2.3).

This dissertation focuses on the hippocampal sub-region CA1, and the subiculum (SUB), which has received less attention compared to the CA1. SUB receives direct synaptic inputs from CA1, and projects to various cortical and subcortical areas (Matsumoto et al., 2019; Sun et al., 2019; Olson et al., 2021). Neurons in the SUB conjunctively encode position, speed, and direction (Ledergerber et al., 2021). Compared to CA1 place cells, SUB place cells showed larger and less specific place fields (Sharp and Green, 1994; Potvin et al., 2007), and these representations were suggested to be the convergence of a number of CA1 place cells (Barnes et al., 1990). In addition to exhibiting direction selectivity, SUB neurons also respond to boundaries of the environment (Lever et al., 2009) and some neurons are tuned to the animal's axis of travel (Olson et al., 2017). SUB neurons were characterized to be encoding "kinds" of places; for example, on a Triple-T maze, SUB neurons are often active for locations that are analogous with respect to the maze structures (Olson et al., 2021).

Both CA1 and SUB are major hippocampal output regions, and they are reciprocally connected. SUB receives a large projection from CA1, and sends a sparser backward projection to CA1 (O'Mara, 2005; Xu et al., 2016). The mammillary bodies project strongly to SUB, which may be the source of head direction information along with the anterior thalamic nuclei (ATN) (Allen and Hopkins, 1989). It has been suggested that SUB processes input information with a mixed

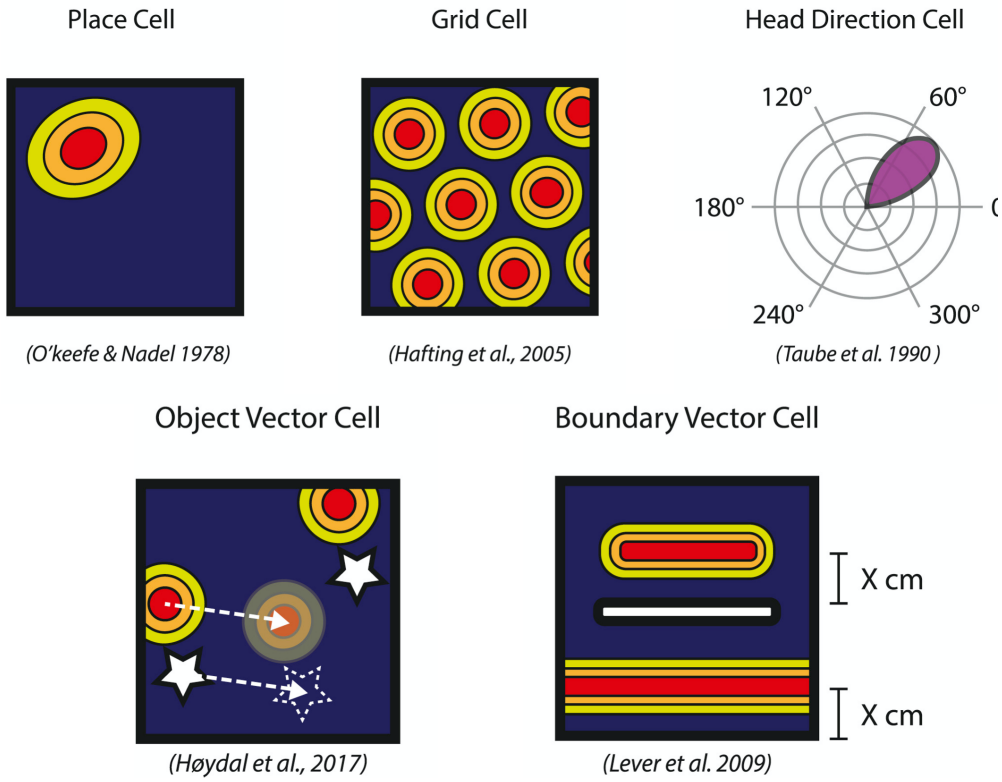


Figure 2.3: Schematic drawings of spatial representations observed in the rodent brain. **A.** Place cells that are active when the animal is in certain locations in the environment. Initially discovered in the hippocampal sub-region CA1 (O'Keefe and Dostrovsky, 1971). **B.** Grid cells that have multiple firing fields that form grids tiling the spatial environment. Discovered in the medial entorhinal cortex (MEC) (Hafting et al., 2005). **C.** Head direction cells that are active when the animal is facing a particular direction. Initially discovered in the postsubiculum (Taube et al., 1990). **D.** Object vector cells that are active when the animal is in a particular distance and direction to objects in the environment. Discovered in MEC (Høydal et al., 2019). **E.** Boundary vector cells that are active when the animal is close to the boundary of the environment. Discovered in the subiculum (Lever et al., 2009). Figure adapted from Behrens et al. (2018).

selectivity and broadcasts the hippocampal signal for more efficient readout in distant brain areas (Ungerleider and Pessoa, 2008).

Remarkably, spatial representations in the hippocampus involve more than the current experience. For example, place cells in CA1 were shown to have trajectory-dependent activity. Using a plus-shaped maze where the rat navigated from one arm to another, with a decision point at the intersection, Ferbinteanu and Shapiro (2003) observed that place cell activity on the shared segment of two routes varied with different route selection. This differential activity was dependent on the rat's prior trajectory, preceded the navigation decision and provided a prospective coding decisions in the immediate future. Neurons that exhibited this property were also termed "splitter cells," and the quality of the trajectory-dependent information, measured by the consistency of firing, and the discriminability between left and right turns, was found to be correlated with task performance (Kinsky et al., 2020).

The emergence of spatial representations in the brain and the integration of different signals that contribute to these representations have been investigated through various computational models. Burgess et al. (1994) suggested that hippocampal place fields can be generated through associative learning with sensory inputs. In this model, sensory inputs were modeled using a set of "sensory cells" that respond broadly to the distance of discrete sensory cues such as visual, olfactory, and auditory cues. Learning occured through exploration of the environment, and the model demonstrated that unsupervised competitive learning led to the development of stable place fields. Redish and Touretzky (1997) proposed a system-level model of rodent navigation based on path integration using a place code and a representation of head direction. This model incorporated self-motion signals that updated the place code, and suggested a mechanism for aligning place code with local view information upon re-entry into familiar environments. Using a brain-based device with a simulated nervous system that modeled the hippocampus and its surrounding regions following anatomy, Krichmar et al. (2005) demonstrated that place fields emerged from the by integrating visual and self-movement cues during exploration in a real environment. Additionally, based on

the evidence that the EC provides strong cortical inputs into the hippocampus and the discovery of grid cells as well as boundary vector cells in the EC, computational models have been proposed to describe the emergence of place fields from EC inputs. Solstad et al. (2006) showed that place fields can be formed by summing the inputs from a population of grid cells with similar spatial phases, diverse grid orientations, and a biologically plausible range of grid spacings. In addition, Hartley et al. (2000) demonstrated that inputs sensitive to the distance and allocentric direction of boundaries in the environment, which is what boundary vector cells respond to, can drive place-specific responses.

To investigate how these forms of spatial representations can guide navigation, reward learning has been incorporated in studies of hippocampal-dependent navigation. In these studies, the activity of a population of place cells signals the location of the agent, and the agent takes actions which were represented by the activity of head direction cells to receive rewards. Place fields were described as either Gaussian tuning curves that tiled the entire environment (Foster et al., 2000; Frémaux et al., 2013), or as an integration of allothetic information provided by the visual system and idiothetic information such as internal movement-related signals (Arleo and Gerstner, 2000). These studies modeled place cells in the CA1 and CA3, which have small and specific place fields. In contrast, Burgess et al. (1994) used SUB cells with large place fields built from CA1/ CA3 projections, allowing information to be passed from positions far from the goal. To account for the representation of future occupancy and decisions in place cells, Stachenfeld et al. (2017) suggested that the predictive nature of hippocampal place codes could be realized by a successor representation (SR) in a reinforcement learning framework. The SR serves as a representation of the expected discounted future state occupancy, and can be combined with the reward function to form the value function (Dayan, 1993). Their SR model showed that a predictive code in the hippocampus could flexibly support adaptive behavior.

In Chapter 5, I propose biologically plausible SNN models of hippocampal sub-region CA1 and SUB. I show that self-motion variables, including linear and angular velocity, head direction, and

position information can be integrated differently within the same network architecture, through separately evolved STDP learning curves. This integration results in differing spatial representations as observed in CA1 and SUB, respectively.

Chapter 3

CARLsim 6: An Open Source Library for Large-Scale, Biologically Detailed Spiking Neural Network Simulation

3.1 Introduction

Spiking Neural Networks (SNNs) provide a powerful method for modeling the dynamics of biological neural networks due to their ability to encode and process data in the temporal dimension, making them an important tool for both the neuroscience and the machine learning communities. As research in these fields continues to advance, there is a growing need for more flexible and accessible tools for building and simulating complex SNNs. Such tools would provide an easy-to-use

This chapter is a slightly modified version of the paper titled "CARLsim 6: An Open Source Library for Large-Scale, Biologically Detailed Spiking Neural Network Simulation" published in the 2022 International Joint Conference on Neural Networks (IJCNN) Proceedings (Niedermeier et al., 2022). The inclusion of this paper is in compliance with the permission policy of the original publisher.

user interface, efficient and scalable network processing ability, and a rich set of native features that supports a wide variety of simulation needs.

CARLsim is a biologically detailed and large-scale simulator for SNNs that meets these criteria. It supports various types of spiking neuron models, including the Izhikevich model with four or nine parameters, as well as the Leaky Integrate-and-Fire (LIF) model. CARLsim also supports the simulation of spiking neurons with multiple compartments, such as spiking neurons that include a somatic compartment and a number of dendritic compartments, whose membrane capacitance can be separately simulated with the Izhikevich model (Izhikevich and Edelman, 2008). In addition, spiking neurons in a group can be arranged in a three-dimensional grid to capture the spatial structure of the network. For example, this feature is used in the SNN model presented in Chen et al. (2022), where the neuron groups of two visual motion processing layers were organized into grids to preserve the retinotopic information in these two layers. On the synapse level, CARLsim supports simulations in the current-based (CUBA) mode, where the synaptic current into the postsynaptic neuron is directly proportional to the synaptic weight, and in the conductance-based (COBA) mode, where the input synaptic current is dependent on conductances with exponential decays. Synaptic connections between neurons can have fixed weight values or be updated with synaptic plasticity, such as spike-timing-dependent plasticity (STDP) or short-term plasticity (STP). Various forms of synaptic plasticity have been implemented in CARLsim, including STDP that can be applied to either glutamatergic synapses (E-STDP) or to GABAergic synapses (I-STDP), Dopamine-modulated STDP (DA-STDP), STP, and homeostatic plasticity (Beyeler et al., 2015).

To enhance its usability, CARLsim provides real-time and offline data analysis tools and an automated parameter tuning interface (PTI) which accelerates the creation of SNN models and facilitates the analysis of simulation data (Beyeler et al., 2015). CARLsim leverages the power of diverse computing tools by supporting the use of multiple graphics processing units (GPUs) and multiple central processing units (CPUs) concurrently in a heterogeneous computing cluster (Chou

et al., 2018). With the advantage of energy efficiency and computing speed, neuromorphic chips are also a promising option for SNN applications, such as cloud edge devices and autonomous robots. Recently, a Python interface was introduced to CARLsim, which preserves its computing efficiency realized by the C++ core developments, and also further increases its accessibility to the computational neuroscience and machine learning communities (Balaji et al., 2020). PyCARL facilitates the construction of SNN models for neuromorphic hardware, and supports the estimation of hardware latencies to allow for performance optimization (Balaji et al., 2020).

This chapter describes the development of CARLsim's sixth generation, which involves both system maintenance and the introduction of new features (Niedermeier et al., 2022). System maintenance efforts focus on ensuring sustainability and improving accessibility to users. These efforts include addressing defects, extending documentation, and integrating CMake which ensures CARLsim to be compatible to all relevant target platforms (i.e. Linux, Windows, and MacOS). In addition to the analysis toolbox already implemented in MATLAB, CARLsim 6 now provides a Python implementation of the Offline Analysis Tool (OAT). Apart from these, new features introduced in CARLsim 6 include the support for multiple neuromodulatory features that enable the implementation of the mammalian neuromodulatory system as defined by Krichmar (2008); Avery and Krichmar (2017). Furthermore, the configuration of STDP and STP is now applied at the connection level, rather than being restricted to the post-synaptic group level. The new structural features and neuromodulation implementations are illustrated in Figure 3.1.

Overall, the system maintenance effort and the feature enhancement in CARLsim 6 allow users to utilize a variety of hardware configurations and incorporate greater biological fidelity into their models. In the remainder of this chapter, we describe the new system maintenance and functional features in detail. Continuing the trend from previous releases, CARLsim 6 remains open-source, extensible, backwards compatible with prior versions. The source code is publicly available on GitHub at https://github.com/UCI-CARL/CARLsim6.

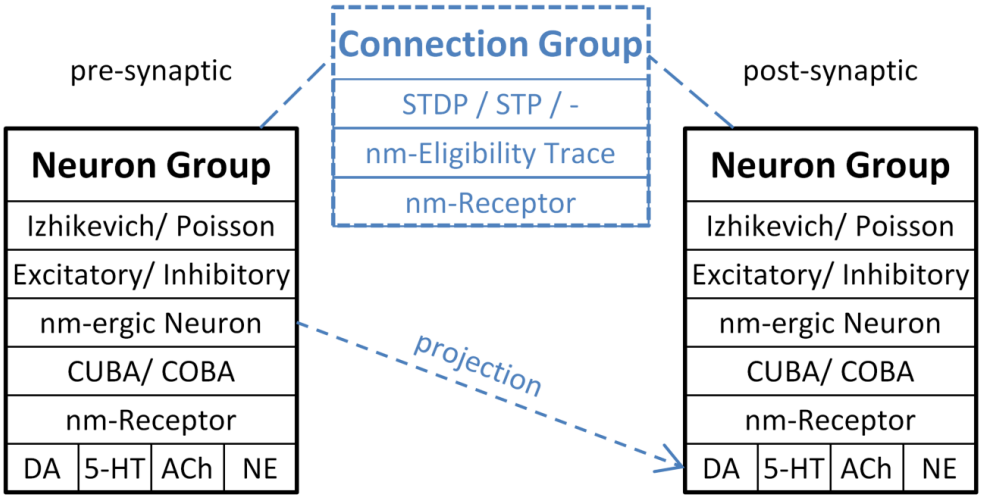


Figure 3.1: Configuration of CARLsim 6 structural and neuromodulation features. CUBA/COBA can be configured at group level. Neuromodulatory neurons can project to other neuron groups. Plasticity is defined on the connection level. Receptors can be defined on both the pre- and post-synapctic groups.

3.2 New features in CARLsim 6

3.2.1 System maintenance

The Integration of CMake and the Multi-Operating System Support

As a highly efficient software framework for simulating large-scale SNNs, CARLsim is developed in C++ and leverages the computational power of CUDA. However, due to the complexity of its compilation and build process, as well as the constantly evolving nature of its software and hardware dependencies, maintaining the software across different platforms has remained a challenge.

To address this issue, CARLsim 6 now integrates CMake into the build process. CMake is recognized as the best practice build system for C++ projects by the open source community. It allows developers to define the build process in a platform-independent way, and generates build files that are specific to the target platform. With CMake, CARLsim 6 can be built using the same CMake scripts across all supported platforms, including Linux, Mac OS, and Windows. Users can

now configure the build process to compile and link only the desired features using the CMake Graphical User Interface (GUI) or the command line interface (CLI).

CARLsim 6 restores the previously removed support for the Windows platform, as CMake enables the generation of project files for all current versions of Microsoft Visual Studio. On Windows, CARLsim now deploys as an optimized Dynamic Link Library (DLL) that can be integrated as a plug-in, which further improves the flexibility and maintainability of the software. In addition, CARLsim 6 provides extensions to PyNN, enabling PyCARL (Balaji et al., 2020), the Python interface to CARLsim, to run on Windows, as well as enabling interactive analyses with the network simulations in JupyterLab.

The improvement in CUDA and GPU support

The support of GPU and CUDA is key to the high performance of CARLsim. It is thus important to ensure that CARLsim 6 can support the latest CUDA versions and GPU architectures. CARLsim 6 has been successfully tested on Linux and Windows up to CUDA version 11.5. Additionally, CARLsim 6 provides support for GPUs with the Ampere architecture, which is used by NVIDIA GForce RTX 3060 up to the NVIDIA DGX-A100. The CARLsim website lists recommended systems and tested configurations, and also includes performance benchmarks to serve as a guideline.

Binaries, Docker support, and community contributions

To increase usability of the software, CARLsim 6 provides binaries for commonly used platforms, such as Ubuntu 20.04 LTS and Windows 10 /11. This allows users to evaluate the software performance without the need to compile it from source.

Additionally, to increase consistency and reproducibility of CARLsim programs in different software environments, different machines, and different platform versions, CARLsim 6 now offers

Docker images of Ubuntu 20.04 with CARLsim 6 pre-installed and configured. This simplifies the process of using CARLsim 6 for users, eliminates compatibility issues on different machines, and removes the need for installation and configuration from the user side. This is also beneficial for deploying CARLsim 6 projects on cloud computing platforms, such as Kubernetes, which run containerized workloads and services.

As an open-source software, contributions to CARLsim by the community follow the GitHub workflow. New features and fixes are organized in Git feature branches and are committed by the contributor as a GitHub Pull Request (PR). After the PR is reviewed by the CARLsim development team, changes can be merged into the master branch, triggering the continuous integration and build pipeline (CI/CD) for quality testing. If all quality gates have been passed, the binaries are created under the release tagged “LATEST" and will be available for download.

Python Offline Analysis Toolbox (OAT)

CARLsim provides convenient tools for monitoring network activity, such as spike monitors, neuron monitors, and group monitors for recording neuronal activity, and connection monitors for recording synaptic changes. The outputs of these monitors can be analyzed with the Offline Analysis Toolbox (OAT) integrated in CARLsim. However, in earlier versions of CARLsim, only the MATLAB OAT is provided, which posed a challenge for users that have limited access and familiarity with MATLAB. In response to this, CARLsim 6 now introduces the Python OAT, which offers a more accessible and flexible option for analyzing network simulations. This is also beneficial to users of PyCARL (Balaji et al., 2020), as the Python OAT allows them to simulate and analyze data generated in the simulation without the need to switch between different software or tools.

Parameter tuning with Evolutionary Computation packages

CARLsim 6 provides an automated parameter tuning interface (PTI), which is integrated with two powerful evolutionary computation packages written in Java (ECJ) (Luke, 2017) and in Python (LEAP) (Coletti et al., 2020). The PTI assists in finding optimal parameters for network simulations at different levels, from the level of single neurons to the level of the entire network, using a fitness function defined by the user that describes the need of their task. To perform a parameter tuning process in CARLsim with the PTI, the user creates an SNN model in CARLsim and specifies the open parameters to optimize. The evolutionary computation settings, such as the population size of a generation or reproduction methods, can then be defined in ECJ or LEAP. During the parameter search process, the PTI communicates between CARLsim and the EC package, passes parameter values provided by the EC to simulate networks in CARLsim, and obtains fitness scores from the network simulation for the EC to generate a new set of parameter values. The evolutionary search process is accelerated with the support of parallel execution of multiple SNN instances in CARLsim. CARLsim in combination with ECJ has demonstrated success in reproducing neural dynamics observed in a number of brain regions (Carlson et al., 2014; Rounds et al., 2018; Chen et al., 2021, 2022). As a newly integrated package in CARLsim 6, LEAP provides easy-to-use syntax and powerful visualization features.

While CARLsim 6 introduces several functional enhancements that allow for SNN simulations with high biological fidelity, the additional parameters may add complexity to the parameter tuning process. The automated PTI provides an efficient solution to this challenge, allowing users to search for appropriate parameter values with ease.

3.2.2 Functional enhancements

Connection-level configuration of synaptic plasticity

CARLsim implements spike-timing-dependent plasticity STDP, and short-term plasticity STP, both of which are synaptic plasticity mechanisms observed in the brain, and have also been demonstrated to be powerful tools for encoding and processing information in SNNs through synaptic updates.

In previous releases of CARLsim, STDP parameters were specified on the post-synaptic group, and all connections to this group with the same connection type (i.e., E-STDP or I-STDP) shared the same STDP parameters. The present release offers improved flexibility in STDP setting and allows users to assign separate STDP parameters for each inter-group connection. This feature enhances the biological plausibility of the SNN model, and increases the learning capacity of the model by introducing additional parameters.

To illustrate this feature, the code snippet below demonstrates how to define different STDP configurations for the two incoming excitatory projections into the same output neuron group. Note the different variables used in the two “setESTDP" functions.

```
// two separate connections into the output group
sim.connect(gIn, gOut, ...
sim.connect(gOut, gOut, ...

// set E-STDP for the input connection
sim.setESTDP(gIn, gOut, true, STANDARD,
             ExpCurve(alpha_LTP_in, tau_LTP_in,
             -alpha_LTD_in, tau_LTP_in)
             );

// set E-STDP for the recurrent connection
```

```
sim.setESTDP(gOut, gOut, true, STANDARD,
             ExpCurve(alpha_LTP_out, tau_LTP_out,
             -alpha_LTD_out, tau_LTP_out)
             );
```

In addition to STDP, CARLsim 6 now allows users to configure STP for each inter-group connection. Various forms of STP have been observed in neural circuits for distinct connection types (Moradi and Ascoli, 2019). The updated feature in CARLsim 6 enables further variability to be included in an SNN by allowing STP variables u, τ_u, τ_v, and τ_d to be set for each connection.

The code snippet below demonstrates how STP can be configured on an inter-group connection between Basket (inhibitory) and Pyramidal (excitatory) cells, two well documented neuron types in the Hippocampus:

```
// configure STP on an inter-group connection between
// CA3 Basket and CA3 Pyramidal groups
sim.setSTP(CA3_Basket, CA3_Pyramidal,
           true,
           STPu(0.23f, 0.04f),
           STPtauU(16.74f, 2.0f),
           STPtauX(384.34f, 50.0f),
           STPtdAMPA(5.0f, 0.0f),
           STPtdNMDA(150.0f, 0.0f),
           STPtdGABAa(7.64f, 1.5f),
           STPtdGABAb(150.0f, 0.0f),
           STPtrNMDA(0.0f, 0.0f),
           STPtrGABAb(0.0f, 0.0f)
           );
```

As illustrated in the example above, each STP variable, including u, τ_u, τ_v, and τ_d for *AMPA* and $GABA_A$, can be assigned a mean and standard deviation. This implementation allows for the

incorporation of biologically realistic parameter estimates based on receptors with fast synaptic properties.

Group-level configuration of input current

The implementation of current-based (CUBA) and conductance-based (COBA) synaptic models in CARLsim gives users the flexibility to choose between a less computationally extensive input current integration approach and a more realistic conductance based integration approach.

In previous versions of CARLsim, the choice of CUBA and COBA was applied to the entire simulation. In CARLsim 6 the calculation method of the input current can be individually defined for each neuron group to provide greater flexibility. The interface is backward compatible and applies default network settings, so that models developed with earlier versions of CARLsim can still be simulated without changes. Furthermore, the group-based configuration was expanded to include metabotropic receptors, also known as G-Protein Coupled Receptors (Bucher and Marder, 2013; Pándy-Szekeres et al., 2022; Kooistra et al., 2020), which inherently rely on neuromodulators. Examples of these receptors include dopaminergic D1/2, norepinephrine alpha1, and muscarinic for acetylcholine ACh. The neuromodulation features will be described in more detail below.

Neuromodulatory effects on target neuron groups

Neuromodulators can have broad, long-lasting effects on downstream neurons (Lakna, 2019). In addition to the well-studied single pathways of distinct neuromodulators (Rosenzweig et al., 2007), it has been observed that neuromodulators also heavily influence each other (Krichmar, 2008; Avery and Krichmar, 2017). To facilitate SNN simulations that incorporate the effects of neuromodulators, CARLsim 6 provides support for four major neuromodulators (NM4): dopamine (DA), serotonin (5-HT), acetylcholine (ACh), and norepinephrine (NE). The concentration of these neuromodulators are captured as an NM4 molarity vector in the neuron group, which is targeted by

the projection of neuromodulatory neurons. Accompanying the implementation of these features, the CARLsim offline analysis toolkit (OAT) was also extended to visualize the molarity in the distinct target groups. Figure 3.3 shows an example visualization of the molarity of different neuromodulators. In the current implementation, a target neuron group accumulates the molarity of all contained postsynaptic neurons of neuromodulatory neurons. The NM4 molarity vector is utilized as input for multivariate functions to implement neuromodulated excitability and plasticity.

In an SNN model of the insular cortex developed using CARLsim, dopamine modulated postsynaptic facilitation (DA-PSF) was implemented with custom code (Chou et al., 2015). This implementation is limited to only the effect of dopamine. However, multiple modulators can act on the same synapse to modify its strength depending on the behavioral needs. Different neuromodulators can lead to drastically different effects on the synapse. For example, 5-HT can functionally silence synapses, and dopamine can unmask synapses that are normally silent. The combined action of multiple neuromodulators on synapses can be more complex than simply additive, and the same neuromodulator can have opposing actions on the synaptic strength (Nadim and Bucher, 2014). With the group-level configuration of input current implemented in CARLsim 6, the calculation method of the input current can be individually set for a neuron group and parameters being incorporated using an extended form of linear combination. This implementation extends and generalizes the DA-PSF method described in Chou et al. (2015).

The following code snippet demonstrates the configuration of synergistic and antagonistic receptors on the target neuron group. Note that the inclusion or removal of these receptor effects can be realized with the same function “setNM4weighted", which takes in different parameter values that correspond to the concentrations of each type of receptor.

```
// target neuron group (g_nm)
sim.setNeuromodulator(g_nm,
                      baseDA, tauDA, releaseDA, true,
                     base5HT, tau5HT, release5HT, true,
                     0.001f, 1.f, 0.f, false,
```

```
                        baseNE, tauNE, releaseNE, true
                        );

// synergistic receptor
sim.setNM4weighted(g_nm, NM4W_LN21,
                    1.0f/1.5f,  // DA normalize and weight
                    0.0f,       // 5-HT
                    0.0f,       // ACh
                    2.0f/1.5f,  // NE normalize and weight
                    4.0f / (1.0f + 2.0f), // normalize all and boost
                    1.0f       // unmodulated baseline
                    );

// antagonistic receptor
sim.setNM4weighted(g_nm, NM4W_LN21,
                     0.0f,       // DA
                     -1.0f/1.5f,  // 5-HT normalize and weight
                     0.0f,       // ACh
                     1.0f/1.5f,  // NE normalize and weight
                     4.0f/2.0f,  // normalize all and boost
                     2.0f        // unmodulated baseline
                     );
```

Figures 3.3 demonstrates the effects of synergistic and antagonistic receptors on the defined target neuron group, where the concentrations of different neuromodulators separately strengthen or weaken the input current based on the individually defined parameters, as shown in the code snippet above. In this simulation, the simulated Izhikevich neurons exhibited non-linear excitability with crossover from tonic to phasic expression. In the control group, with increasing input current, the crossover happened when the current reached about 10 μA. With synergistic receptors added, DA lowers the crossover down to 6 μA, and NE lowers the crossover to 3 μA. When both neuromodulators were present, the phasic firing is almost instantaneous (Figure 3.3 (a)). In contrast, the

antagonistic receptor 5-HT can silence the neuron if presented alone, or neutralize the effect of NE if the concentration level of both receptors are the same (Figure 3.3 (b)).

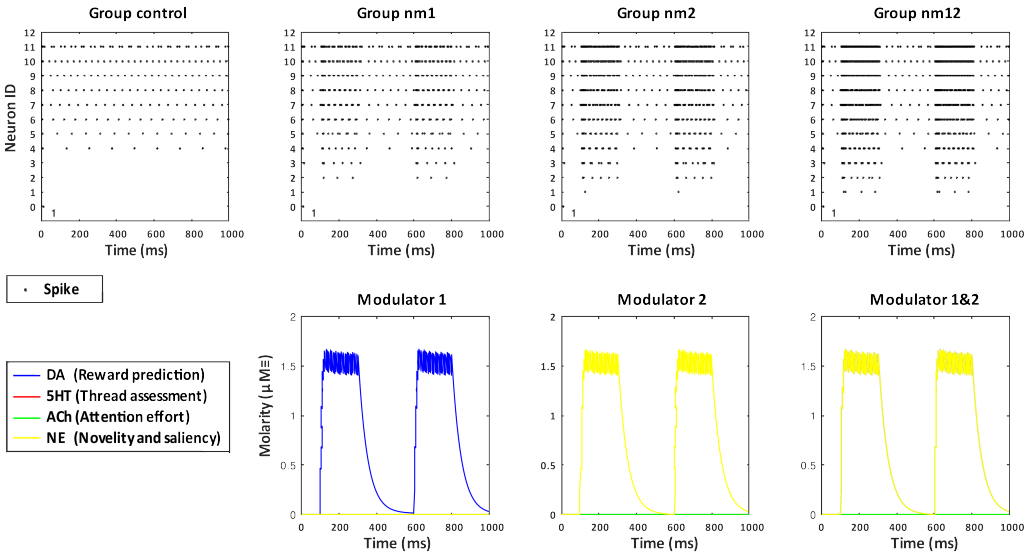
Conductance modulation of receptors

CARLsim 6 also implements the neuromodulated $\alpha 1$/ $\alpha 2A$ NE receptors, and the $D1$/ $D2$ DA receptors, the concentration of which were shown to impact the level of arousal and decision-making performance in visuospatial working memory tasks (Avery et al., 2013; Avery and Krichmar, 2015).

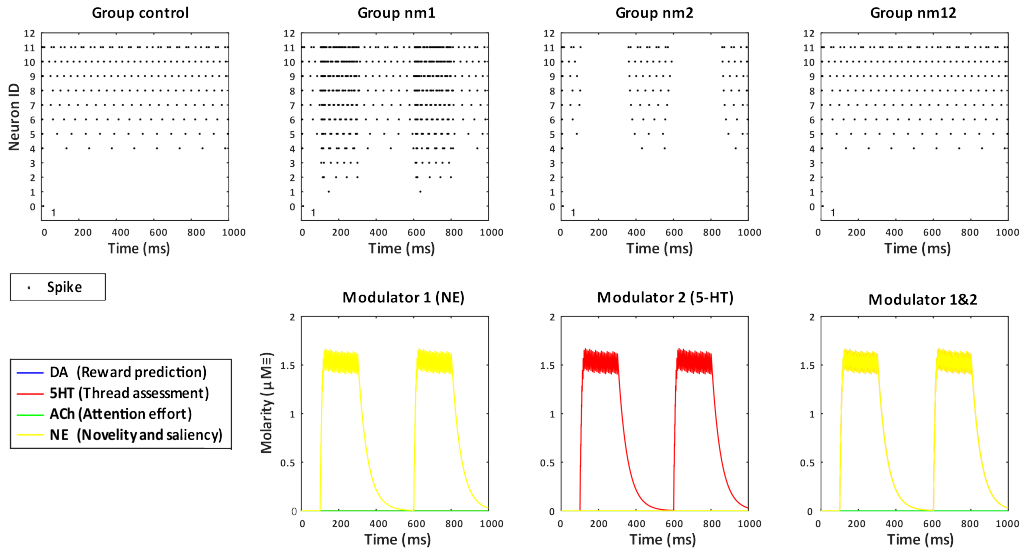
As shown in the code snippet below, the receptors can be configured using the generic interface function “setConnectionModulation" with parameter values defined for each type of receptor. In this function, the first two arguments correspond to the pre- and post-synaptic groups, which are used to identify the connection group and assign the receptor parameters.

```
// noadrenergic alpha2A receptor
sim.setConnectionModulation(g_L3e[c], g_L3e[c], alpha2A_ADK13);

// dopaminergic D1 receptor
sim.setConnectionModulation(g_L3e_npref[d], g_L3e[c], D1_ADK13);

// dopaminergic D2 receptors
sim.setConnectionModulation(g_L5e[c], g_L5e[c], D2_AK15);

sim.setConnectionModulation(g_MD_SC, g_L5e[c], D2_AK15);

// alpha1 receptor (DA, NE, lambda)
sim.setNM4weighted(g_L3e[c], alpha1_ADK13,
                   1.f, 0.f, 0.f, 1.0f, 1.f,
                   -1.0f / 6.0f / log(1.0f / 3.0f)
                   );
```

Figure 3.4 (a) shows the underlying continuous mapping of DA to the connection-based neuromodulatory factor μ_c of the D2 receptor, and figure 3.4 (b) shows the continuous bivariate mapping of



(a) Synergistic impact of NE on DA resulting in tonic to phasic excitability



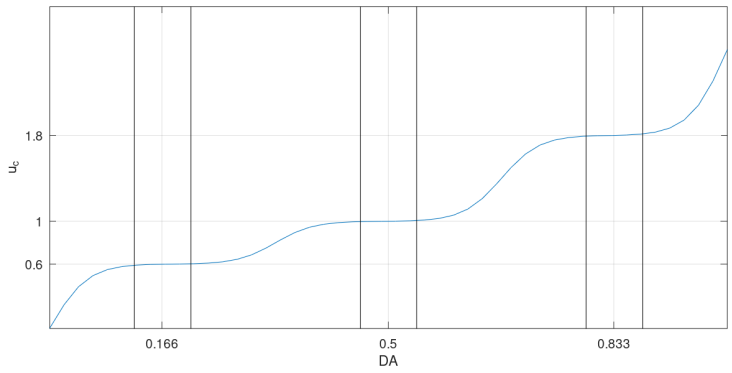
(b) Antagonistic impact of 5-HT on NE with suppression and equilibrium

Figure 3.3: CARLsim 6 allows for the configuration of Izhikevich neurons with neuromodulated synergistic and antagonistic receptors, that exhibit non-linear excitability with an arbitrary crossover from tonic to phasic firing. The left column shows the firing in CARLsim's SpikeMonitor of such neurons without neuromodulation, labeled as control group. Each neuron was mapped to an increasing input current, displayed on the y-axis, with a designed crossover at about 10 μA ($\mapsto$ NeuronID 10). The simulation time is displayed on the x-axis in ms for each neuron group. (a) Synergistic effect on the neurons of the neuromodulators NE and DA. The molarity of the target groups are displayed in the CARLsim's GroupMonitor in the second row, which shows consistent values over time to demonstrate the synergistic effect on the receptor that is the changed crossover point of the firing from tonic to phasic. DA lowers the crossover down to 6 μA, while NE lowers it even more down to 3 μA, and with both neuromodulators present, the phasic firing is almost instantaneous. (b) Antagonistic effects of 5-HT on NE, which is suppression for 5-HT and neutralization at equilibrium. The SpikeMonitor of group nm1 shows that the antagonistic receptor has the same excitability for NE as above. However, when 5-HT is present, it acts as antagonist, and the nm2 group displays gaps of spikes, when the neuron is silenced by 5-HT. The right column shows that when both NM are present at the same level, the NM-effect is neutralized and group nm12 exhibits the same behaviour as the control group.

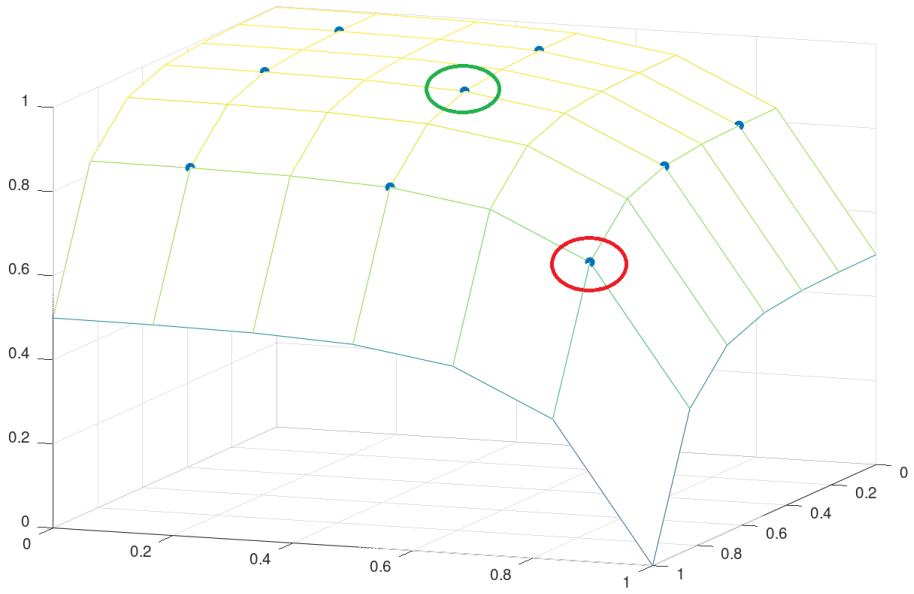
DA and NE to the group-based neuromodulatory factor μ of the α1 receptor. These results show the sensitivity dependency of the interaction between NE and DA, and confirm the successful replication of the neuromodulatory levels described in Avery et al. (2013) and Avery and Krichmar (2015). In addition to the already implemented receptor types, the implementation of configurable neuromodulatory effects using a generic interface function is easily extensible to other G protein-coupled receptors.

Neuromodulated short-term plasticity (STP) and spike-timing-dependent plasticity (STDP)

CARLsim 6 introduces the capability to modulate the short-term plasticity (STP) applied to a neuron group to be influenced by neuromodulators. As a result, STP can act as a gain-control mechanism that modifies synaptic strength as a function of the frequency of presynaptic activity. In some cases, it can even switch the sign of synaptic dynamics from depression to facilitation and vice versa (Nadim and Bucher, 2014).



(a) Neuromodulatory factor $\mu_c(DA)$ of receptor D2



(b) Neuromodulatory factor $\mu(DA, NE)$ of receptor $\alpha 1$

Figure 3.4: A well-balanced concentration of different types of neuromodulators has a profound impact on the working memory. CARLsim 6 provides a continuous mapping of NE, DA to the neuromodulatory factor μ that matches the discrete levels investigated in Avery et al. (2013) and Avery and Krichmar (2015). (a) Continous mapping of DA to the connection based neuromodulatory factor μ_c of the D2 receptor fitting the discrete levels (inside columns) given by Avery and Krichmar (2015). (b) Continuous bivariate mapping of DA, NE to the group based neuromodulatory factor μ of the $\alpha 1$ receptor fitting the discrete bivariate levels (marked by blue dots) given by Avery et al. (2013).

The following code snippet shows how the STP variables u, τ_u, τ_v can be configured to be modulated by a multivariate function on the NM4 concentration, thus enabling the expression of facilitation as standard STP at a certain level. To ensure consistency with previous implementations, the modulated STP was validated to yield the same plasticity as standard STP. This new feature can be applied to elucidate synaptic gain and plasticity operations in the hippocampal subregion CA3 (Kopsick et al., 2022).

```
// facilitative STP
sim.setSTP(g1,true, 0.15f, 750.0f, 50.0f);

// modulated STP by serotonin
float u[] = { 0.0f, 1.0f, 0.0f, 0.0f, 0.30f / 0.15f, 1.0f };
float tau_u[] = { 0.0f, 1.0f, 0.0f, 0.0f, -700.0f / 750.0f, 1.0f };
float tau_x[] = { 0.0f, 1.0f, 0.0f, 0.0f, 700.0f / 50.0f, 1.0f };
sim.setNM4STP(g1, u, tau_u, tau_x);
sim.setNeuromodulator(g1,
                        0.001f, 1.f, 0.f, false,
                        1.0f; 1.f, 0.f, true, // base5HT
                        0.001f, 1.f, 0.f, false,
                        0.001f, 1.f, 0.f, false
                       );
```

On top of the eligibility trace-based spike-timing-dependent plasticity (STDP) for dopamine (DA-STDP) implemented in previous versions, CARLsim 6 now extends this feature to other neuromodulators such as 5-HT, ACh, and NE. Furthermore, the offline analysis tool (OAT) has been updated to monitor the neuromodulator level of the eligibility trace-based STDP, facilitating the analysis of the effect of neuromodulation on STDP and synaptic dynamics.

The connection-level configuration of STDP in CARLsim 6 allows for the creation of multiple connection groups with different neuromodulators. This capability facilitates the implementation of complex neuromodulatory systems such as the one described in Krichmar (2008). The code

snippet below demonstrates the configuration of STDP modulated by different neuromodulators, which can be realized by the same function “setSTDP" with different neuromodulator flags.

```
// set up dopamine modulated STDP
sim.setSTDP(gin, gPFC, true, DA_MOD,
            alphaPlus,tauPlus,alphaMinus,tauMinus
            );
// or for other modulators
sim.setSTDP(gin2, gPFC, true, SE_MOD,..
sim.setSTDP(gin3, gPFC, true, AC_MOD,..
sim.setSTDP(gin4, gPFC, true, NE_MOD,..
```

In addition to modeling the effect of neuromodulators in eligibility trace-based STDP, CARLsim 6 also introduces the ability to model the effect of neuromodulators on long-term potentiation (LTP) and long-term depression (LTD) of synapses. The balance of LTP and LTD can be altered by different neuromodulators. For example, the activation of the protein kinase A (PKA) pathway promotes and gates LTP, and the activation of the phospholipase C (PLC) pathway promotes LTD. Interestingly, the activation of one pathway also supresses the activation of the other, leading to a push-pull rule for neuromodulation of long-term synaptic plasticity that is independent of the underlying mechanisms of LTP and LTD (Nadim and Bucher, 2014).

CARLsim 6 extends the PKA/ PLC modulation effect to any arbitrary neuromodulator pair, such as NE and ACh. Figure 3.5 shows how the phenomenological STDP curves, which can take different shapes depending on the positively or negatively defined amplitude parameters, can be altered by the push-pull effect of PKA and PLC.

This implementation enables SNNs to have adaptive learning rates depending on the neuromodulators, and also may lead to unlearning in some cases, which may be particularly relevant for learning by rewiring as described in Krichmar (2012), Bucher and Marder (2013), and Krichmar

et al. (2019). The following code snippet demonstrates the configuration of PKA/PLC-modulated LTP/LTD of STDP.

```
float ALPHA_LTP = 0.08f;
float ALPHA_LTD = -0.12f;
float TAU_LTP = 8.0f;
float TAU_LTD = 8.0f;
//enum FigNr  { _a,   _b,  _c,  ..
//float ne []={ 1.0f,  1.0f, 0.0f, ..
//float ach[]={ 1.0f,  0.0f, 1.0f, ..
//float a_p[]={ 0.08f, 0.16f, -0.08f, ..
//float a_m[]={-0.12f, 0.12f, -0.24f, ..

// set STDP group
g1 = sim->createGroup("excit", 1, EXCITATORY_NEURON);
sim->setNeuronParameters(g1, 0.02f, 0.2f, -65.0f, 8.0f);

// set modulated STDP group
//g2 = sim->createGroup(..
//sim->setNeuronParameters(g2,..
sim->setNeuromodulator(g2,
                       1.0f, 1.0f, 1.0f, false, // DA
                       1.0f, 1.0f, 1.0f, false, // 5HT
                       ach[fig], 1000000, 0.000001f, true,
                       ne[fig],  1000000, 0.000001f, true
                       );

// set ESTDP with reference values
sim->setConductances(false);
sim->setESTDP(gex2, g1, true, STANDARD,
              ExpCurve(a_p[fig], TAU_LTP, a_m[fig], TAU_LTP)
              );
```

```
// set PKA_PLC modulated ESTDP
sim->setConductances(false);
sim->setESTDP(gex2, g2, true, PKA_PLC_MOD,
              ExpCurve(ALPHA_LTP, TAU_LTP, ALPHA_LTD, TAU_LTP),
              PkaPlcModulation(NM_NE,  1.0f,   // pka
                               NM_ACh, 1.0f    // plc
                              )
              );
```

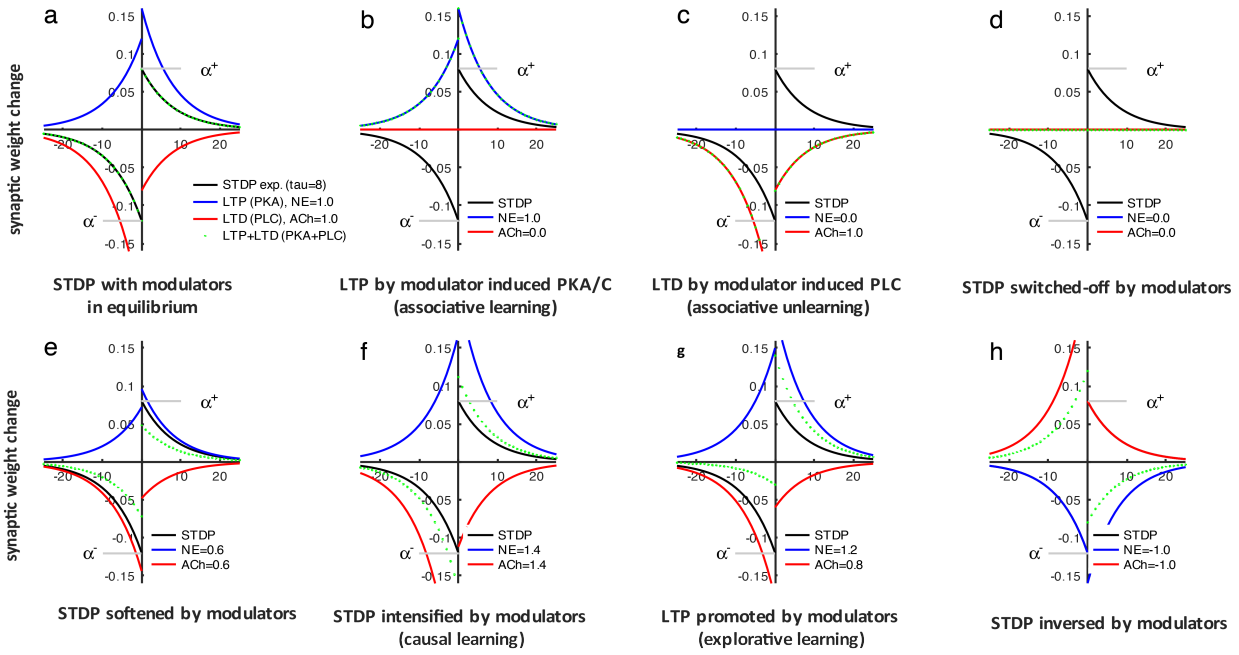


Figure 3.5: LTP/LTD is modulated by NE and ACh and can adapt STDP based learning to environmental needs.

3.3 Benchmarks and example simulations

3.3.1 Computational performance

A primary goal during the development of CARLsim 6 was to maintain comparable performance with its previous versions, despite the kernel changes that made the implementation of various new

features possible. To evaluate the performance of CARLsim 6 against its predecessor, CARLsim 5, we conducted multi-GPU benchmark tests similar to those performed in Chou et al. (2018). This set of benchmark tests used an SNN model with 2 subnetworks, each consisting of 80 % excitatory neurons and 20 % inhibitory neurons, and an input Poisson neuron group to drive the activity of the network. Each subnetwork had four inter-group connections: the input connection to the excitatory group, the recurrent connection within the excitatory group, and the reciprocal connections between the excitatory and inhibitory groups, with each neuron having 100 synapses.

In the 1-GPU and 1-CPU tests, all neuron groups were allocated to the same GPU or CPU. In the 2-GPU and 2-CPU tests, each subnetwork is partitioned among 2 GPUs or CPUs, and 2 out of the 4 connections are allocated to different GPUs or CPUs.

To test the scalability of CARLsim, we varied the total number of excitatory and inhibitory neurons from 10^3 to 10^5. Figure 3.6 shows the simulation time of each benchmark setup. The results show that CARLsim 6 achieves similar computing efficiency compared to the previous release, CARLsim 5, and retains the capability of utilizing multiple GPUs for larger-scale SNN simulations.

3.3.2 Large-scale model of a hippocampal subregion

In the past, models of the neural circuit were usually limited to modeling only a few neuron types and a simple form of their connections for studying a particular function of the circuit. To increase biological accuracy and scale of these models, additional biological details can be incorporated, such as the number of neurons for each neuron type as observed in the brain, connection probability between neuron types, synaptic plasticity that induces long and short term synaptic changes, neuromodulatory effects on synaptic dynamics, and spatial structure of neuron groups and their connections.

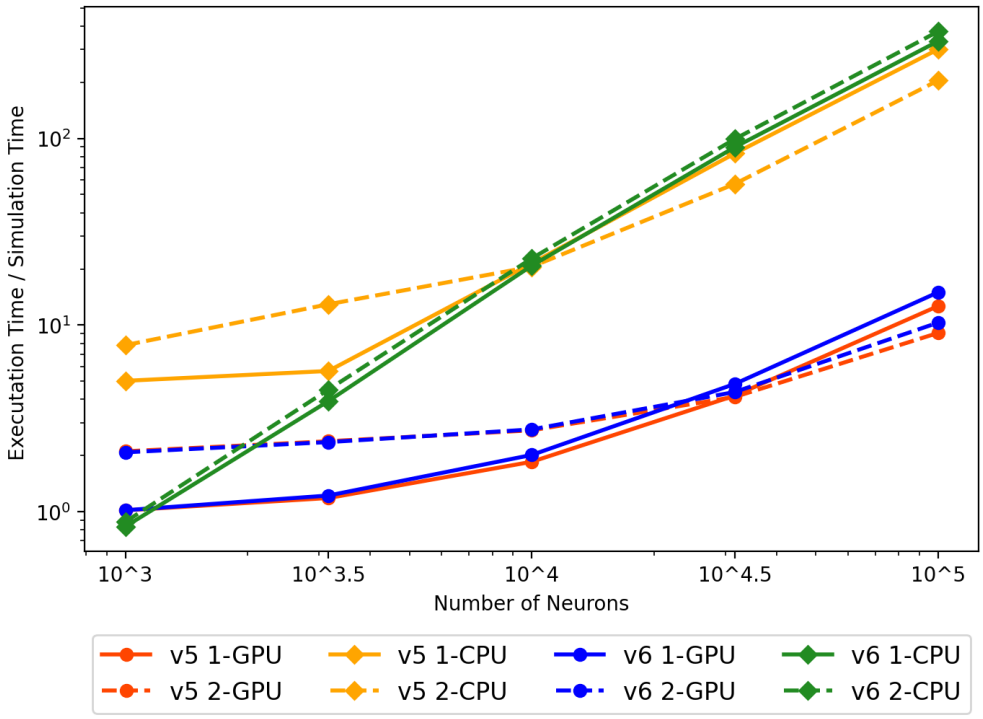


Figure 3.6: Benchmark performance of CARLsim 5 (v5) and CARLsim 6 (v6). The vertical axis represents the execution time, where a value of 1 indicates real-time performance.

CARLsim 6 provides support for these implementations, as demonstrated in a large-scale neural circuit model of mouse hippocampal subregion CA3. This model consists of 8 neuron types, 90,000 neurons, 51 neuron-type specific connections, and 250 million synapses. Parameters for simulating the neuron groups, their connections, as well as the synaptic plasticity dynamics were derived from published data of the hippocampus hosted in the Hippocampome.org knowledge base (Kopsick et al., 2022; Moradi and Ascoli, 2019; Tecuatl et al., 2021; Komendantov et al., 2019).

CARLsim 6 allows for a fast simulation of this large-scale network, with one second of simulation time taking 4 minutes of real time. The simulated network produced beta oscillations that are similar to those observed in the hippocampal subregion CA3 during stationary behaviors. Additionally, the simulated neuron groups exhibited firing patterns that are consistent with existing knowledge of cell type-specific activity in vivo, such as the local field potentials (LFPs) for each neuron type (Figure 3.7 (a)), and the distribution of firing phase of each neuron type relative to the beta oscillation (Figure 3.7 (b)).

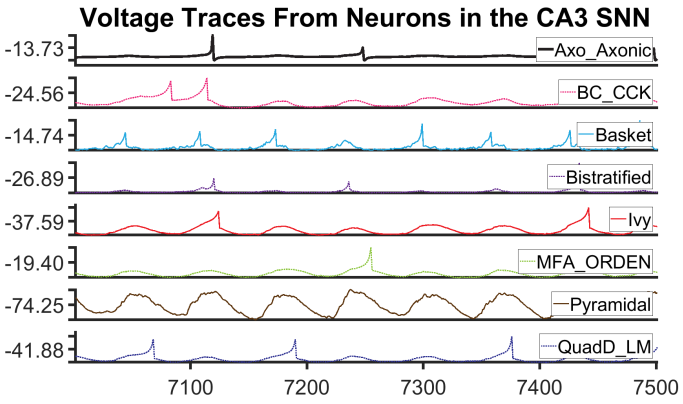
This simulation shows that CARLsim 6 has the potential to be adapted and extended to support biologically realistic, data-driven neural circuit models of various brain areas. Its support for efficient simulations of large and complex network models makes it an appealing simulation environment for modeling at the neural circuit level.

3.4 Related work

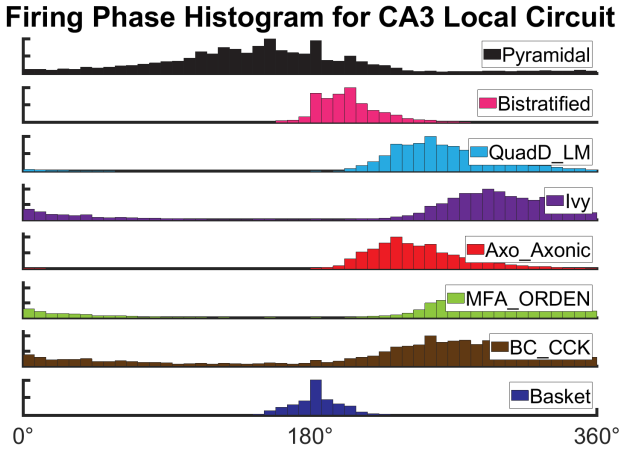
Compared to Artificial Neural Networks (ANNs) that encode neural activity with a rate-based approach, SNN models encodes the timing of the spikes, which allows them to process temporal information. SNNs also offer the advantage of incorporating biologically realistic features into neural network simulations, allowing for the modeling of the biological neural systems in more details, which can lead to a better understanding of the neural circuits and their functions.

To facilitate the development of SNN models, various SNN simulators have been developed. We compare CARLsim 6 with the latest versions of several of these open source SNN simulators that support parallel execution of SNN simulations, conductance based synapses, and synaptic plasticity, which are the key features of CARLsim 6 that allows for accurate and efficient SNN simulations. Table 3.1 presents a comparison of supported features and software/ hardware requirements of different SNN simulation platforms. These simulators include: Brian2 (Stimberg et al., 2019), GeNN (Yavuz et al., 2016), Nengo (Bekolay et al., 2014), NEST (de Schepper et al., 2022), NEURON (Carnevale and Hines, 2006), NeuronGPU (Golosio et al., 2021).

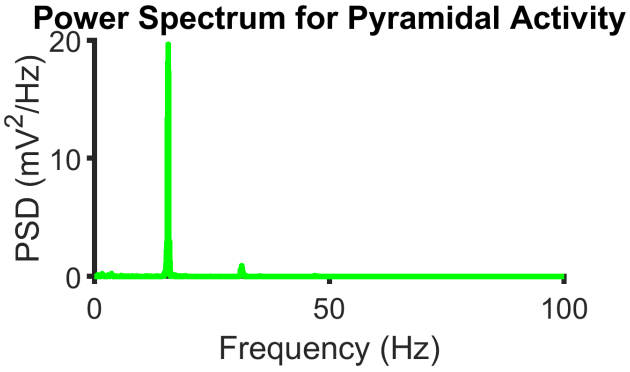
Similar to CARLsim 6, NeuronGPU utilizes GPUs as a backend and has an optimized spike delivery algorithm (Golosio et al., 2021) which enables high computing performance comparable to CARLsim. However, NeuronGPU's feature development remains at an early stage, as it currently only has nearest-neighbor STDP implemented as synaptic plasticity and lacks neuromodulation.



(a)



(b)



(c)

Figure 3.7: (a) Representative voltage traces of individual neurons from each neuron type in the SNN model. The horizontal axis represents time (ms), and the vertical axis represents the voltage (mV). (b) Firing phase histograms of each neuron type relative to the beta oscillation. (c) The power spectrum of CA3 Pyramidal neurons.

We also compare the support of neuromodulation in different SNN simulators, which is an important feature introduced in CARLsim 6. NEST, for instance, can generate code for DA-STDP utilizing NESTML (de Schepper et al., 2022). Brian provides an improved generator syntax for synaptic plasticity. Nengo applies BMC as unsupervised learning as equivalent for tripplet based STDP. The other simulators including NCS or GeNN, have incorporated notable changes in this functional area.

Overall, compared to other simulators, CARLsim 6 is advantageous in that it implements a wide variety of synaptic learning rules and biologically plausible model features, and is also highly optimized for large scale SNN simulations, utilizing parallel execution with different computing hardware.

3.5 Conclusion

CARLsim 6 offers enhanced flexibility for the implementation of biorealistic features in SNNs by supporting the configuration of STDP and STP at the connection level, instead of being restricted to the post-synaptic group level. Additionally, CARLsim 6 now supports the implementation of multiple neuromodulators and their effects on neuronal excitability and plasticity. The introduction of the Python OAT and the Python-based parameter tuning with LEAP makes modeling and analysis tools more accessible to users. With the system maintenance efforts, CARLsim 6 remains computationally efficient on multiple compute platforms. With these improvements, CARLsim 6 is well positioned to be a valuable tool for constructing biologically detailed models of the brain, and developing highly efficient neurobiologically-inspired algorithms that are amenable to neuromorphic implementations (Krichmar et al., 2019).

As an open-source tool licensed under the MIT License, CARLsim 6 provides a matured core library with a powerful support system for new projects, making it well suited for building a new

Table 3.1: Comparison of SNN simulators by recent features. An ‘X‘ denotes that the feature is directly supported by the simulator, while a ‘/‘ means that the user has to implement custom code, respectively that the feature is only partially implemented, and a blank ‘ ‘ that feature is not available in general. Gray colored cells highlight features new in CARLsim 6 compared to previous versions.

	CARLsim 6	**Brian 2.5**	**NEURON**	**GeNN 4.6**	**NCS 6**	**Nengo 3.2**	**NEST 3.2**	**NeuronGPU**
Synaptic plasticity								
DA-STDP	X	/	X			X	X	
5HT-/ ACh-/ NE-STDP	X	/	/			X		
Modulated LTP/ LTD	X							
Connection specific STDP	X	X	X				X	X
Neuromodulated STP	X	/	/					
Connection specific STP	X	X	X				X	
Synapse model								
Group level CUBA	X	X			X		X	
Group level COBA	X	X			X		X	
Neuromodulation	X	X	X	/		X	X	
Tools								
Parameter tuning (JAVA)	X					X		
Parameter tuning (Python)	X	/				X		
Analysis/ visualization	X	/	X		/	X		
Front-ends								
Python/ PyNN	X	X	X	X	X	X	X	X
C/ C++	X		X	X	X			X
Back-ends								
Single-threaded CPU	X	X	X	X	X	X	X	X
Multi-threaded CPU	X	X	X		X	X	X	X
Distributed		/	X		X	X	X	X
Single GPU	X	/		X	X	X		X
Multi-GPU	X				X			X
Hybrid (Multi-CPU/ GPU)	X				X			

ecosystem of neuroscience and engineering applications. CARLsim 6 is available on GitHub at https://github.com/UCI-CARL/CARLsim6.

Acknowledgment

This work was supported by NSF award IIS-1813785, and by the United States Air Force Award FA9550-19-1-0306. The authors are thankful for the computing resources provided by CHASE-CI under NSF Grant CNS-1730158.

Chapter 4

Cortical motion perception emerges from dimensionality reduction with evolved spike-timing dependent plasticity rules

4.1 Introduction

As an observer moves through the environment, the motion between oneself and the scene causes a change in the structure of light, which is reflected on the retina as optic flow. Optic flows can be used to perceive the heading direction of self-motion and to guide locomotion (Warren and Hannon, 1988). Neurons in the dorsal visual pathway of the primate brain are selective for motion (Britten, 2008). The Middle Temporal (MT) area contains neurons that are tuned to the speed and

This chapter is a reprint of the article titled “Cortical motion perception emerges from dimensionality reduction with evolved spike-timing dependent plasticity rules", which was originally published in the Journal of Neuroscience in 2022 (Chen et al., 2022). This chapter includes minor revisions to the original article. The inclusion of this article is in compliance with the permission policy of the original publisher.

direction of motion (Albright, 1984). Receiving primary input from MT (Raiguel et al., 1997), neurons in the dorsal sub-region of the medial superior temporal (MSTd) area respond to large and complex optic flow patterns, including translation, rotation, expansion, contraction, and the intermediates of these motions (Graziano et al., 1994). MSTd was suggested to play an important role in heading perception. Physiological studies showed that microstimulation of the MSTd area biased the monkey's perception of heading directions based on optic flow (Britten and Van Wezel, 1998; Gu et al., 2012). The causal role for self-motion perception was also established in the human functional equivalent of macaque area MST (hMST) (Schmitt et al., 2020). Furthermore, heading direction can be decoded from the population activity of MSTd neurons (Hamed et al., 2003). MST was also suggested to participate in 3D velocity estimation (Mineault et al., 2012). These studies provide evidence for MSTd's role in self-motion perception based on optic flow. However, it is not well understood how MSTd integrates input from MT to form complex selectivity to motion patterns and to perceive self-motion.

The brain is under tight metabolic constraints and uses numerous strategies to achieve efficient representations of information that allow for high performance and information transfer (Krichmar, 2019). Nonnegative sparse coding (NSC) is an efficient population coding scheme that combines dimensionality reduction with sparsity constraints to reduce the number and activity of neurons to represent environmental features (Beyeler et al., 2019). A prior study described a computational model that can account for a wide range of MSTd visual response properties through applying a dimensionality reduction algorithm known as Nonnegative Matrix Factorization (NMF) with sparsity constraints, which implements a form of NSC, to MT input activity (Beyeler et al., 2016). This MSTd model exhibited sparse, 'parts-based' representations of the optic flow patterns that resembled the receptive fields observed in MSTd neurons. This model provided evidence that the seemingly complex response properties may emerge from MSTd neurons performing a biological equivalent of dimensionality reduction and sparse coding on their input. However, it remains to be shown how these coding schemes can be implemented in neurobiological circuits.

This chapter, which is based on published work (Chen et al., 2022), describes how a neurobiologically plausible synaptic plasticity could implement a form of NSC. Learning processes in the brain are believed to occur in the synaptic changes among neurons. Hebb's rule describes a mechanism that leads to long-term-potentiation (LTP) between excitatory neurons given their consistent coactivity (Hebb, 1949). Oja's learning rule adds constrained synaptic modification on top of the Hebbian model to prevent saturation of synaptic weights (Oja, 1982). In the spiking domain, the spike-timing dependent plasticity (STDP) learning rule modifies synaptic weights according to the relative timing of spikes of the pre-synaptic and post-synaptic neurons (Sjöström and Gerstner, 2010). These synaptic learning rules perform statistical inferences on the input data. Oja's rule was shown to reduce the dimensionality of datasets similar to performing principal component analysis (PCA) (Oja, 2008). Analogously, through mathematical derivation, STDP in combination with homeostatic scaling (STDP-H) was suggested to have the same effect on the input data as NMF (Carlson et al., 2013). We propose that STDP-H could have a similar effect on MSTd receptive fields.

In the present study, we implement a Spiking Neural Network (SNN) model of monkey MSTd based on evolved STDP-H parameters. We show that by learning to reconstruct the input stimuli with STDP-H, the network extracts representations of the input in a form that resembles the receptive fields of MSTd neurons. The spiking neurons in the network show selectivity to spiral motions and 3D heading direction similar to the selectivity observed in the brain. The present results suggest a neurobiologically plausible method for producing sparse and reduced receptive fields in the dorsal visual system, and possibly other brain regions (Beyeler et al., 2019).

4.2 Materials and methods

A spiking neural network model (SNN) of visual cortical areas MT and MSTd was created to reconstruct optic flow patterns resulting from self-movement. Figure 4.1 shows the overall archi-

tecture of the SNN model. In this model, input optic flow stimulus was processed by an array of MT neurons, whose responses were then converted to Poisson spike trains and projected to a group of excitatory spiking neurons that simulated MSTd. The MSTd group projected to a group of inhibitory neurons, and the inhibitory neurons provided feedback inhibition to the MSTd group to regulate the network activity. Connection weights in the network were updated with an unsupervised learning rule which implemented STDP and homeostatic scaling (STDP-H). The parameters of the learning rule were optimized through evolutionary algorithms, with an objective function measuring how well the network reconstructed the input stimuli. Neural activity was modeled with Izhikevich neurons (Izhikevich, 2004), with regular spiking for excitatory neurons and fast spiking for inhibitory neurons.

SNN simulations were performed using CARLsim 5 (Chou et al., 2018; Balaji et al., 2020). Parameter optimization was done with the parameter tuning interface (PTI) in CARLsim 5, which utilized the Evolutionary Computations in JAVA (ECJ) library (Luke, 2017). The evolutionary process which included network training and validation was performed with 1 GPU node and 15 CPU cores on the computing clusters supported by the Cognitive Hardware and Software Ecosystem Community Infrastructure (CHASE-CI) (Altintas et al., 2019). We utilized parallel execution enabled by CARLsim and PTI to distribute computations among the GPU and CPU cores, which greatly accelerated the simulation processes. In each generation of evolutionary computation, PTI launched a total of 50 separate SNN instances that were simulated in parallel and evaluated concurrently.

4.2.1 Code accessibility

Custom code used in this study including neural network simulations and data analysis scripts will be publicly shared upon publication of this article.

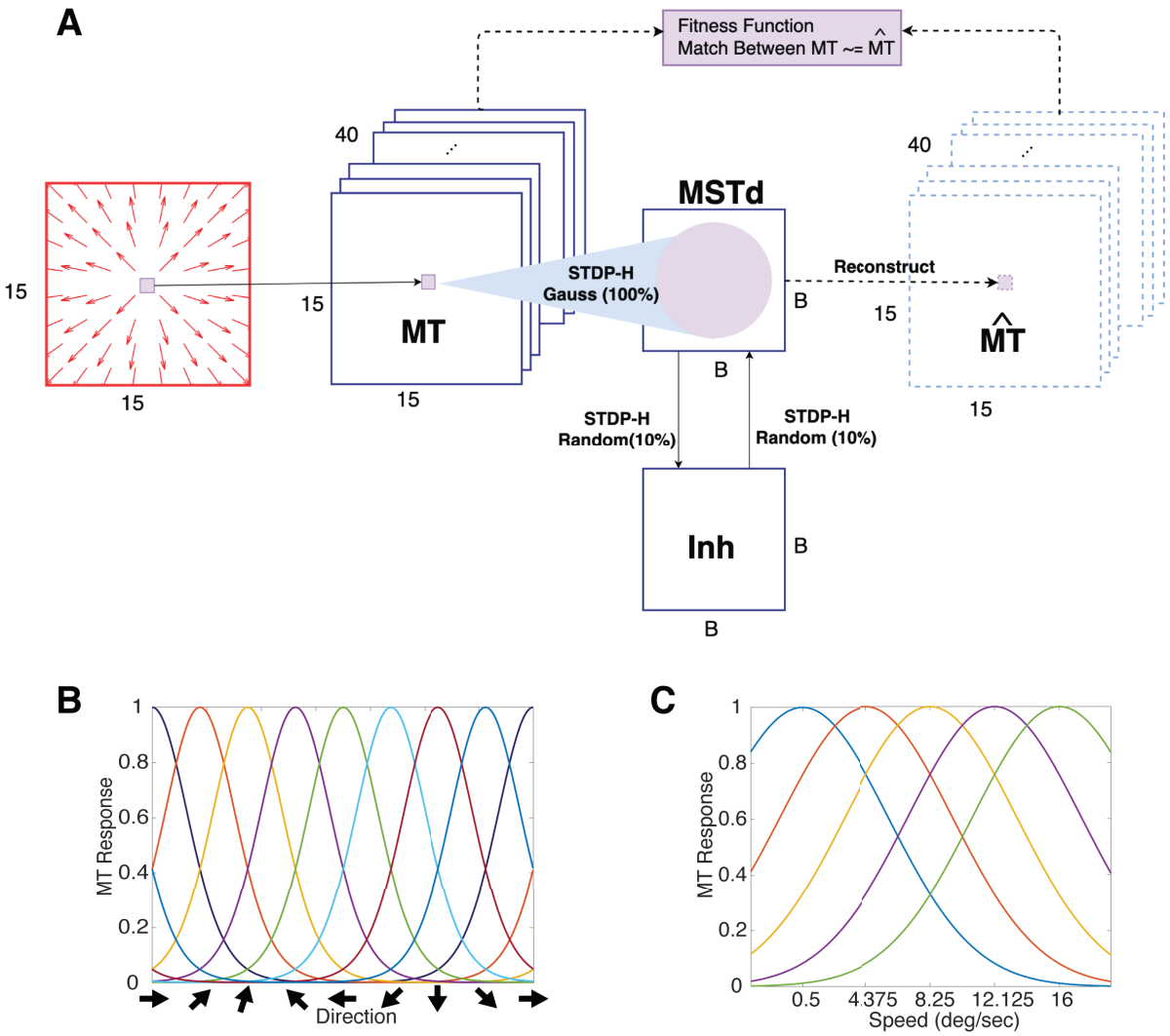


Figure 4.1: Model deisgn. **A.** Architecture of the SNN model. Optic flow stimulus was first processed by an array of MT neurons tuned to speed and direction of motion. The MT activity was converted to Poisson spike trains as the input to the network. The MT neuron group was connected to the MSTd group with a Gaussian projection, which allowed the MSTd neurons to receive input from MT neurons that locate in different locations of the visual field and were tuned to different direction and speed. The MSTd group was reciprocally connected to a group of inhibitory neurons, which regulated the activity of the network. All connections in the network were modulated by STDP-H. The MT $\rightarrow$ MSTd connection weights and the MSTd group activity were used to reconstruct the input. A fitness function measured the network performance based on the reconstruction accuracy, and STDP-H parameters were evolved with evolutionary computation to optimize the fitness function. **B.** Direction tuning curves of the simulated MT neurons. **C.** Speed tuning curves of the simulated MT neurons. The horizontal axis is plotted on a log_2 scale.

4.2.2 Input stimuli

Input stimuli to the network were computer-generated arrays of optic flows that resembled those used in physiological experiments. To simulate the apparent motion on the retina that would be caused by an observer moving in a 3D environment, we used the model of the motion field proposed by Longuet-Higgins and Prazdny (1980), where a pinhole camera with focal length $f = 1$ cm was pointed to a frontoparallel plane located at a distance d, which simulated 3D real world points at a reference depth Z. The pinhole camera was used to project the 3D real world points, $\vec{P} = [X,Y,Z]^t$ onto a 2D image plane, $\vec{p} = [x,y]^t = f/Z[X,Y]^t$ (i.e. the retina). When the camera moved in the 3D environment, the local motion of points on the 2D image plane was described as a function of the camera's motion and the location of the points. In our simulation, the camera moved with a translational velocity $\vec{v} = (v_x, v_y, v_z)$ and a rotational velocity $\vec{\omega} = (\omega_x, \omega_y, \omega_z)$. The optic flow component $(\dot{x}, \dot{y})$ on the 2D image plane at location (x,y) could then be computed as following:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \underbrace{\frac{1}{d}\begin{pmatrix} -f & 0 & x \\ 0 & -f & y \end{pmatrix}\begin{pmatrix} v_x \\ v_y \\ v_z \end{pmatrix}}_{\text{translational flow}} + \underbrace{\frac{1}{f}\begin{pmatrix} x\cdot y & -\left(f^2+x^2\right) & f\cdot y \\ \left(f^2+y^2\right) & -x\cdot y & -f\cdot x \end{pmatrix}\begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix}}_{\text{rotational flow}} \tag{4.1}$$

The first component in the right side of the equation computed the translational flow, and the second component computed the rotational flow. These two components superimposed linearly, and the rotational flow was independent of the depth d. Simulated flow fields were represented as 15×15 pixel arrays, subtending a visual angle of 90° x 90°.

For the training and validation of the network, we generated a dataset containing 1280 flow field stimuli. Each optic flow stimulus was presented to the network for 500 ms, interleaved by 500 ms of a blank stimulus. To simulate motion patterns resulting from locomotion, the linear velocity of the pinhole camera was set to 1 m/s for stimuli that included a translational flow and 0 m/s for the pure rotation motion. The angular velocity was set to 1 radian/s) for stimuli that included a rotational flow, which was within the range of natural head rotation movements (Cullen, 2019), and 0 radian/s for the pure translation motion. The depth was set to 1 m from the observer. Stimuli in the training and validation dataset were sampled uniformly from a laminar motion space and a spiral motion space. The laminar motion space contained unidirectional motion patterns that simulated eight directions (45° intervals in 360°) of translation. For these stimuli, the rotational component in Equation 4.1 was 0, and in the translational component, $v_x = \{0, \pm\sqrt{\frac{1}{2}}, \pm 1\}$, $v_y = \{0, \pm\sqrt{\frac{1}{2}}, \pm 1\}$, and $v_z = 0$. The spiral motion space contained the following motion patterns: expansion/ contraction, clockwise/ counter-clockwise rotation, and intermediate spiral motions, namely clockwise-expanding, clockwise-contracting, counter-clockwise-expanding, and counter-clockwise contracting motions. The center of motion (COM) of these stimuli was kept near the center of the visual field, with $(v_x, v_y) \approx 0$ and $v_z = \pm 1$ for stimuli that had a translational component, and $(v_x, v_y) \approx 0$ and $v_z = \pm 1$ for stimuli that had a rotational component. In order to introduce a slight variability in the COM, We converted the velocities to spherical coordinates and applied a Gaussian noise with standard deviation of 10° to the x and y velocity components, before converting them back into Cartisian coordinates and using Equation 4.1 to compute for optic flow components. Schematic drawings of these motion patterns are shown in Figure 4.2.

We also generated three other datasets, which were used for testing the SNNs after training, by following the protocols used in neurophysiological studies (Gu et al., 2006; Takahashi et al., 2007; Hamed et al., 2003; Graziano et al., 1994). We used these datasets to quantify the response properties, including the Gaussian tuning in spiral space, the 3D translation and rotation heading selectivity, as well as the population encoding of heading, of our simulated MSTd neurons. These measurements allowed us to compare the selectivity of our simulated MSTd neurons to the one

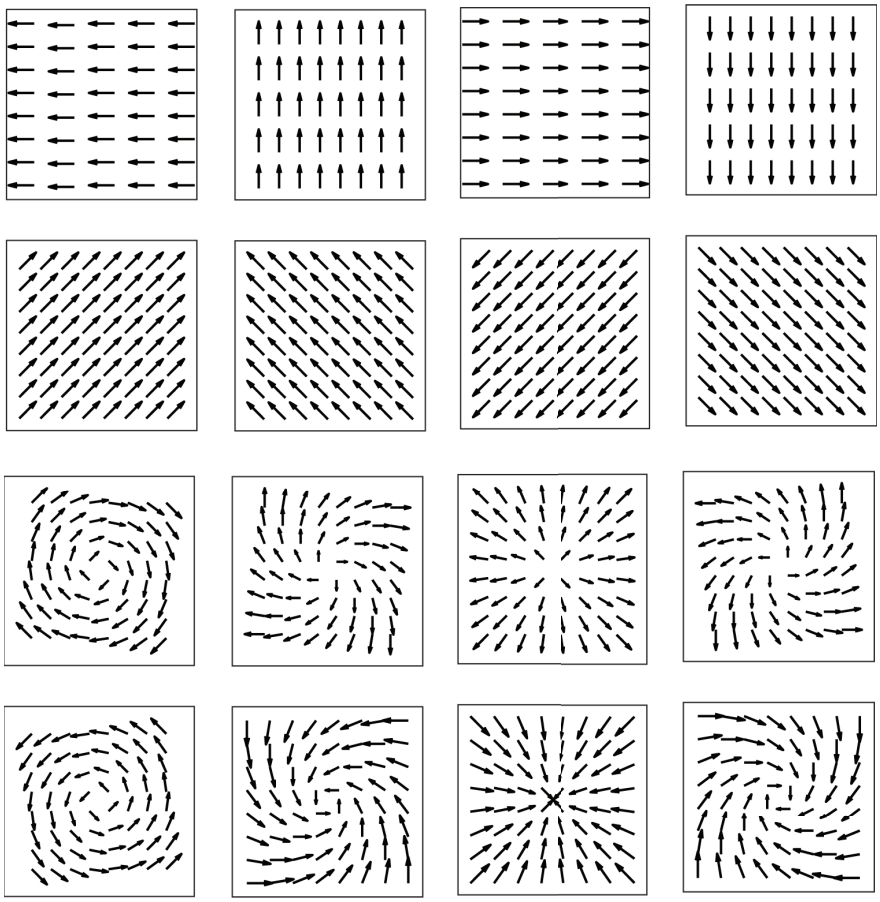


Figure 4.2: Schematic drawings of motion patterns that the training and validation dataset was sampled from. The first two rows depict motion patterns in the laminar motion space, and the last two rows depict motion patterns in the spiral motion space.

observed in the real brain. It is worth noting that these three separate datasets contained stimuli not present in the training and validation dataset. We computed the fitness scores achieved by the evolved and trained models on these three datasets, which allowed us to measure the generalization ability of the model and ensure that the performance was not limited to the validation dataset.

4.2.3 Model design

The SNN model consisted of three neuron groups: MT, MSTd, and an inhibitory group (Figure 4.1). The network model processed flow field stimuli in three steps: 1) Input flow fields were first processed by the MT group, 2) The MSTd group received Poisson spike trains from the MT group, and 3) The input was reconstructed with a dot product of the MT→MSTd connection weights and the MSTd neuronal activation.

The MT neuron group was organized into 3D grids with dimensions $15 \times 15 \times 40$. The first two dimensions corresponded to the pixels of the input flow fields that subtended $90^\circ \times 90^\circ$ of the visual angle. The third dimension corresponded to the direction and speed tuning profile of the neurons. At each spatial location, there were a total of 40 MT-like model units (selective for eight directions times five speeds of motion). This organization can be seen as having multiple layers of MT neurons covering the visual field, with each layer of neurons tuned to the same direction and speed of motion but responding to different locations (Figure 4.1). Neurons in this group were modeled as idealized MT neurons selective to particular combinations of speed and direction of motion. Each MT neuron had a receptive field of 1 pixel area of the simulated flow fields, corresponding to 3° of visual angle. In this study, the firing rate of a MT neuron R_{MT} responding to the motion flow at the location (x,y) was given by:

$$R_{MT}(x,y;\theta_{pref},\rho_{pref}) = d(x,y;\theta_{pref})s(x,y;\rho_{pref}), \tag{4.2}$$

where the component $d(x,y;\theta_{pref})$ computed the direction response of the neuron, and $s(x,y;\rho_{pref})$ computed the speed response of the neuron.

The direction response $d(x,y;\theta_{pref})$ was described by a circular von Mises function (Figure 4.1B):

$$d(x,y;\theta_{pref}) = exp(\sigma_{theta}(cos(\theta(x,y) - \theta_{pref}) - 1)), \tag{4.3}$$

where the direction response was a function of the difference between the stimulus direction $\theta(x,y)$ and the preferred direction θ_{pref} of the neuron. The tuning bandwidth parameter σ_{theta} was set to 3, corresponding to a tuning width of approximately 90°.

The speed response $s(x,y;\rho_{pref})$ was described by a log Gaussian function (Figure 4.1C), which approximated MT tuning curves observed in the macaque brain (Nover et al., 2005). The speed response of a MT neuron was given by:

$$s(x,y;\rho_{pref}) = exp(-\frac{log(\frac{\rho(x,y)+s_0}{\rho_{pref}+s_0})^2}{2\sigma^2}), \tag{4.4}$$

where the speed response was a function of the difference between the stimulus speed $\rho(x,y)$ and the preferred speed ρ_{pref} of the neuron. The free parameter σ determined the width of the curve and was set to 1.16. The speed offset parameter s_0 was set to 0.33, which prevented undefined logarithm values when stimulus speed approaches zero. These parameter settings were consistent with the median values across the MT population (Nover et al., 2005).

The MT response R_{MT} computed from Equation 4.2 represented the normalized firing rate of the neuron ranging from 0 to 1, and was used to generate spike trains following a Poisson distribution with a maximum firing rate of 20 Hz. Neurons in the MT group were connected to the MSTd group following a Gaussian distribution, with which neurons that were spatially closer to each other had a higher probability of being connected and had higher initial connection weights. We

defined a 2D Gaussian connectivity along the first two dimensions of the MT neuron group, and all layers of the MT neurons shared the same connectivity scheme, such that each MSTd neuron had a circular receptive field covering MT neurons at different locations and tuned to different direction and speed of motion. The width of the Gaussian curve, which corresponded to the radius r of this circular receptive field was an open parameter that was evolved through the evolutionary computation process. Activity of the MSTd group was regulated by an inhibitory neuron group, which contained the same number of neurons as the MSTd group. The MSTd neurons provided feedforward excitation to the inhibitory neuron group, and the inhibitory group provided feedback inhibition to the MSTd neurons. Both connections followed a uniform random connectivity with a 10 % probability. All inter-group connections were modulated by STDP-H plasticity during training, which include the projection from MT to MSTd (MT $\rightarrow$ MSTd), from MSTd to the inhibitory group (MSTd $\rightarrow$ Inh), and from the inhibitory group back to the MSTd group (Inh $\rightarrow$ MSTd). As discussed in Section 4.2.6, the parameter values for STDP-H were selected through evolutionary computation.

The resulting MSTd activity was used to reconstruct the input stimuli along with the MT $\rightarrow$ MSTd connection weights. Analogous to the sparse decomposition model proposed by Beyeler et al. (2016), the MT activity was represented as a multivariate matrix $\mathbf{V}$, where each column within the matrix represented an instance of input stimulus $\vec{v}_i$, namely the population activity of the MT group responding to a flow field stimulus. The connection weights between the MT and MSTd groups were represented as a matrix $\mathbf{W}$, where w_{ij} was the connection weight between MT neuron i and MSTd neuron j. The activity of the MSTd neurons was represented as the coefficients in the matrix $\mathbf{H}$, where each column in the matrix $\vec{h}_i$ denoted the MSTd population activity responding to the input stimulus $\vec{v}_i$. To test the hypothesis of whether STDP-H allows the network to learn a reduced representation of the input $\mathbf{V}$ by decomposing it into two lower rank matrices $\mathbf{W}$ and $\mathbf{H}$, we took a dot product of the two matrices to reconstruct the input matrix $\mathbf{V}$. Effectively, a particular instance of input stimulus $\vec{v}_i$ was reconstructed by taking the inner product of the connection weights $\mathbf{W}$ and the MSTd activation $\vec{h}_i$ (see Figure 4.3).

The second column in Figure 4.3 shows a visualization of the connection weights between the MT and MSTd neuron groups. Each grid shows the receptive field of a MSTd neuron, determined by calculating a population vector for direction and speed of motion from **W** (Georgopoulos et al., 1982). These receptive fields acted as basis flow fields. With the MSTd neuron activity indicating the degree of activation, a linear superposition of the basis flow fields reconstructed the input flow field pattern.

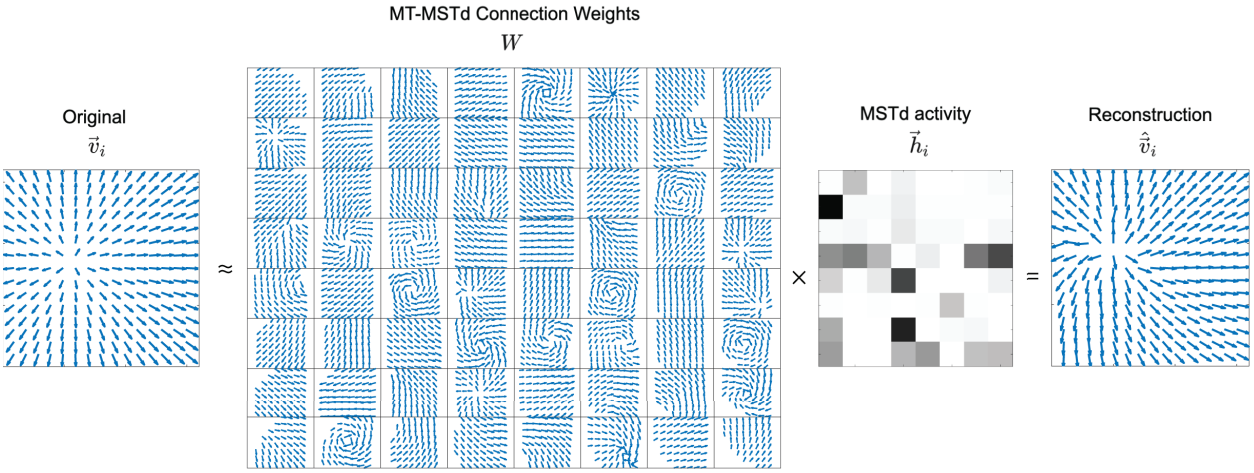


Figure 4.3: Stimulus reconstruction. STDP-H performed dimensionality reduction on the input matrix **V** and decomposed it into two smaller matrices **W**, the MT→MSTd weights, and **H**, the MSTd activity. Each column of the input matrix represented an input instance $\vec{v}_i$, which was visualized as the original flow field. In the figure, the MT → MSTd connection weights **W** were visualized as a group of basis flow fields. The MSTd activation to this particular stimulus $\vec{h}_i$ was represented as a column in the matrix **H**, which denoted the degree of activation of the corresponding basis flow field. Darker color in this visualization corresponded to a higher level of activity. The reconstructed input was shown on the right. The correlation score between the original and the reconstruction of this particular input instance was 0.84.

4.2.4 Spike-timing dependent plasticity and homeostatic scaling (STDP-H)

All three inter-group connections (i.e. MT→MSTd, MSTd→Inh, and Inh→MSTd) were plastic, whose weight values were modulated by the STDP learning rule in combination with homeostatic synaptic scaling (STDP-H). Homeostatic scaling was applied in a multiplicative manner, which modified synaptic weights based on the average post-synaptic firing rate R (Turrigiano et al., 1998;

Turrigiano and Nelson, 2004; Carlson et al., 2013). As illustrated in Figure 4.4, homeostatic scaling adjusted synaptic properties (i.e. connection weight values) to keep the activity of the neurons close to the target firing rate. The total effect of STDP-H on a particular synapse $w_{i,j}$ connecting pre-synaptic neuron i and post-synaptic neuron j can be described as below:

$$\frac{dw_{i,j}}{dt} = [\underbrace{\alpha \cdot w_{i,j}(1-\bar{R}/R_{target})}_{\text{homeostasis}} + \underbrace{\beta(LTP_{i,j}+LTD_{i,j})}_{STDP}] \cdot K \tag{4.5}$$

The α and β terms controlled the strength of STDP and homeostasis. We fixed the value of β to be 1, and evolved the value of α for the optimal relative strength of the two learning components. The parameter K was a term that damped oscillation in the weight updates and sped up learning, which was defined as:

$$K = \frac{\bar{R}}{T \cdot (1+\left|1-\bar{R}/R_{target}\right| \cdot \gamma)} \tag{4.6}$$

Here, the parameter T was the time scale over which the firing rate of the post-synaptic neuron was averaged and γ was a tuning factor, which were set to 10 and 50, respectively, in this study.

In Equation 4.5, the first component described the homeostatic scaling, which was a function of the ratio between the mean firing rate $\bar{R}$ and the target firing rate R_{target} of the neuron j. With homeostatic scaling, the rate of weight changes decreased if the neuron was over-excited, and increased if the neuron was overly quiescent. The second component of Equation 4.5 described STDP, which was composed of long-term-potentiation (LTP) and long-term-depression (LTD). STDP strengthened or weakened the synaptic connection according to the timing of the pre- and post-synaptic spikes. LTP and LTD were described as decaying exponential functions, with a

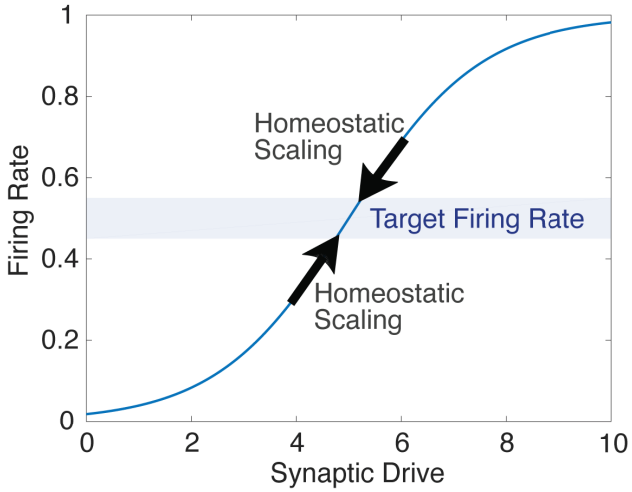


Figure 4.4: Illustration of the effect of homeostatic synaptic scaling on an individual neuron. As the synaptic drive (i.e., input synaptic weights) increases, the firing rate of the neuron increases and exceeds the target firing rate. Homeostatic scaling brings the activity down to the target zone by decreasing the input strength. If the synaptic drive is too low and the activity of the neuron is below the target firing rate, homeostatic scaling raises the input strength and brings the activity of the neuron back into the target firing zone. Synaptic drive is in arbitrary units and firing rate is normalized. Adapted with permission from Turrigiano and Nelson (2004).

magnitude parameter A^+ or A^-, and a decay constant τ^+ or τ^-:

$$LTP_{i,j} = A^+ exp\left(\frac{-(t_j - t_i)}{\tau^+}\right) \text{ for } t_j > t_i \tag{4.7}$$

$$LTD_{i,j} = -A^- exp\left(\frac{t_j - t_i}{\tau^-}\right) \text{ for } t_j \leq t_i \tag{4.8}$$

Parameters A^+, A^-, τ^+, and τ^- were evolved through evolutionary computation.

4.2.5 Training and validation of the model

From the training and validation dataset with 1280 flow fields, we randomly selected 160 stimuli as validation samples and the rest as training samples. In both the training and validation processes, each stimulus was presented to the network for 500 ms, followed by a blank stimulus presented for the same duration of time. During training, connection weights in the network were updated with STDP-H in an unsupervised manner. Parameters of the STDP-H learning rule were optimized through evolutionary computation for each iteration of the training and validation process.

To validate whether the network had learned representations that allowed the reconstruction of unseen stimuli, we used the validation phase where we froze the connection weights and stopped STDP-H from modulating the connections. Flow fields from the validation set were presented to the network in the same way as during the training phase. Here we recorded the activity of the MSTd neurons during the 500 ms of stimulus presentation. As described in section 4.2.3, we reconstructed the input by multiplying $\mathbf{W}$, the connection weights between the MT and MSTd neuron groups, with $\vec{h}_i$, the corresponding MSTd neuronal response to the stimulus. After obtaining a reconstruction for all flow fields from the validation set, we assembled them into a matrix $\hat{V}$, where each column contained the reconstructed MT activity pattern corresponding to an input flow field. We defined a fitness function which accounted for both the reconstruction accuracy and network stability, as follows:

$$y = corr(V, \hat{V}) - \lambda L \tag{4.9}$$

In the first term of the fitness function, we calculated the Pearson correlation between the input matrix V and the reconstructed matrix $\hat{V}$. To prevent instability in the network, we added the

second term L, which described a penalty for high firing rates of the MSTd neuron population and was weighted by a scaling factor $\lambda = 0.001$. The penalty term was defined as:

$$L = \begin{cases} max(\bar{R}_{MSTd}) - R_{MSTd;t} & \text{, if } max(\bar{R}_{MSTd}) > R_{MSTd;t} \\ 0 & \text{, otherwise} \end{cases} \tag{4.10}$$

Here, $\bar{R}_{MSTd}$ was the mean firing rate of individual MSTd neurons, and $R_{MSTd;t} = 250\,\text{Hz}$ was the firing rate threshold. When the maximum mean firing rate of MSTd neurons exceeded the firing rate threshold, the network received a penalty score.

4.2.6 Optimization of the model via evolutionary computation

Parameters of the STDP-H learning rule were optimized through evolutionary computation. A total of 19 parameters divided into 5 groups were evolved: 1) STDP amplitude parameters A^+ and A^- and decay constants τ^+ and τ^- for all three inter-group connections (12 parameters), 2) target firing rates R_{MSTd} of the MSTd group and R_{Inh} of the inhibitory neuron group (2 parameters), 3) maximum connection weights for all three inter-group connections (3 parameters), 4) the radius r of the Gaussian connection between MT and MSTd (1 parameter), and 5) the parameter α that adjusted the relative strengths between STDP and homeostatic scaling for each of the three inter-group connections (1 parameter). Table 4.1 shows the ranges of values for these evolved parameters.

Parameter	A^+	A^-	τ^+	τ^-	**Target FR**	**Max Wt**	**Radius (r)**	**alpha**
Minimum	0	-4e-3	5.0	5.0	5.0	0.001	0.75*sqrt(B)	0.1
Maximum	4e-3	0	100.0	100.0	20.0	0.5	sqrt(B)	1.0

Table 4.1: Parameters evolved with Evolutionary Computation

We used an Evolution Strategy (ES) that took the form of ES-(μ, λ), in which $\mu = 5$ specified the size of the parent population and $\lambda = 50$ specified the size of the offspring population (De Jong, 2006). In the first generation of the evolutionary process, the ES initialized the parameters for 50 individual networks. The network instances were then trained and validated with simulated flow fields in the training and validation dataset, and were evaluated with the fitness function defined in section 4.2.5. These fitness scores were then transferred to the ES. The ES performed a binary tournament selection to select the best performing $\mu = 5$ individual networks as the parent networks for the next generation, and modified their parameters through replication and mutation to produce a new population of $\lambda = 50$ network individuals. This optimization process continued for 30 generations.

4.3 Results

The SNN model went through 30 generations of an evolutionary process which searched for optimal parameters of the STDP-H synaptic learning rule in order to maximize a fitness function that measured the accuracy of input reconstruction. Each evolutionary generation consisted of a training phase and a validation phase. Optic flow stimuli were generated by projecting from a pinhole camera moving at different combinations of linear and angular velocity to a frontoparallel plane. A total of 1280 computer-generated flow field stimuli that simulated different motion patterns were used in the training and validation phases. In the training phase, 1120 stimuli were randomly selected and presented to the network, and connection weights in the network were updated by STDP-H in an unsupervised manner. In the validation phase, 160 optic flow stimuli, which were not used in the training phase, were presented to the network with STDP-H disabled, and the fitness function was computed over the validation set of stimuli. After the evolutionary process, we presented several sets of novel stimuli used in physiological studies to examine the selectivity and response properties of the simulated MSTd neurons in the models.

4.3.1 Model performance and evolved STDP parameters

To investigate whether the SNN model was able to accurately reconstruct the input stimuli, and to understand the effect of model size on the performance of the model, we experimented with different sized neuron populations for the MSTd group ($B = \{16, 36, 64, 100, 144\}$). All configurations shared the same network setup as described in section 4.2.3. The differences among different configurations were the size of the MSTd group, B, and the range of values for the Gaussian projection width, r, which was adjusted such that the MSTd neurons in all configurations covered similar portions of the visual field. To ensure reliable results from the parameter optimization process and to obtain a sizable pool of simulated MSTd neurons for subsequent analyses (a minimum of 144 MSTd neurons were required for the analysis described in section 4.3.5), we completed 10 separate runs for the $B = 16$ configuration, so it had more variability, and 5 separate runs for all other configurations. In each run of the evolutionary process, a total of 1500 SNN instances were simulated (30 generations and 50 network individuals in each generation).

With a population of 50 individual networks in each generation, all five network configurations converged to optimal solutions within 30 generations (Figure 4.5). For all configurations, fitness scores reached $\sim$0.5 at the first generation of the evolutionary process, and gradually increased over generations. At generation 30, all five configurations reached fitness scores greater than 0.65 (see Table 4.2). As shown in Table 4.2, the total time for completing 30 generations of optimization process and the simulation of 1500 network individuals increased with the size of the MSTd group. The evolutionary process of the smallest network with $B = 16$ MSTd neurons required 3.89 $\pm$ 0.75 days to complete, while the largest network with $B = 144$ MSTd neurons required 13.60 $\pm$ 2.19 days to complete. Networks with a larger MSTd group showed a better reconstruction ability and reached higher fitness scores. Among the evolved networks, configurations with $B = \{64, 100, 144\}$ MSTd neurons were able to approximate the input data with $R^2 > 0.7$, indicating an accurate reconstruction of the input stimuli. Although network configurations with more than $B = 144$ MSTd neurons may lead to an even better reconstruction accuracy, we showed in the

sparseness analysis (see section 4.3.2) that the level of sparseness may not improve beyond the $B = 144$ configuration. To ensure that the reconstruction ability of the network models was not limited to the 160 samples randomly sampled from the training and validation datasets, we computed fitness scores that the evolved and trained network models achieved on the three separate datasets, which were composed of samples that were not present in the training and validation dataset (see Table 4.3). The results showed that the models were able to generalize to other motion patterns. These datasets, which are from previous neurophysiological studies (Graziano et al., 1994; Gu et al., 2006; Takahashi et al., 2007; Hamed et al., 2003), will be used in our subsequent analyses for sections 4.3.3, 4.3.4, and 4.3.5.

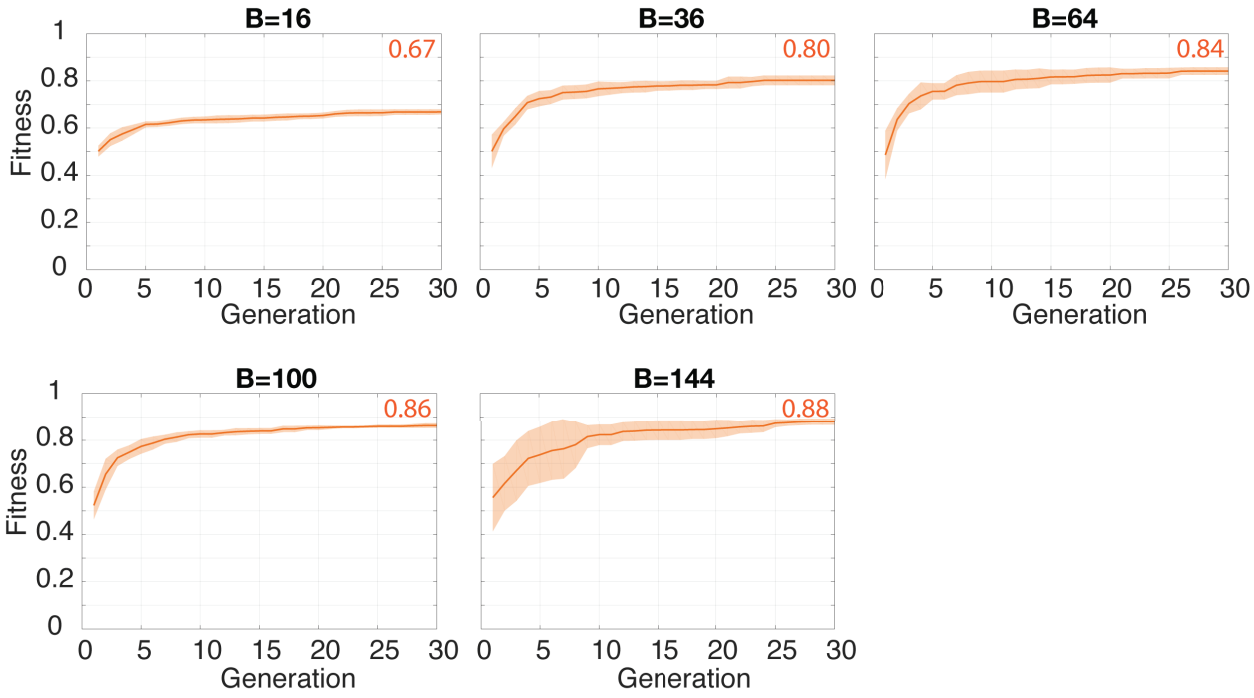


Figure 4.5: Best-so-far (BSF) fitness curve across 30 generations of evolutionary process for each network configuration. Solid lines denote the mean fitness scores of all individual runs of the network configuration. Shaded area **denotes** the standard deviation of the fitness scores.

STDP, as a biologically observed process, modifies synaptic strengths based on the time differences between pre- and post-synaptic spikes. The largest weight changes induced by STDP occur when the time difference between the pre- and post-synaptic spikes is small. As the time difference increases, the weight changes decrease exponentially. Typically, if a presynaptic neuron fires prior

Number of MSTd neurons	**Execution time (days)**	**Fitness (BSF after 30 generations)**	R^2
B=16	3.89 ± 0.75	0.67 ± 0.02	0.45 ± 0.02
B=36	5.28 ± 0.30	0.80 ± 0.02	0.64 ± 0.04
B=64	7.39 ± 0.85	0.84 ± 0.01	0.71 ± 0.03
B=100	10.40 ± 1.52	0.86 ± 0.01	0.74 ± 0.02
B=144	13.60 ± 2.19	0.88 ± 0.02	0.77 ± 0.03

Table 4.2: Execution time and performance of all network configurations. Values are the mean±standard deviation. BSF = best so far of an evolutionary computation run. R^2 is the coefficient of determination.

Number of MSTd neurons	**Spiral selectivity dataset**	**3D translation/ rotation dataset**	**Population encoding dataset**
B=16	0.54 ± 0.06	0.57 ± 0.05	0.56 ± 0.05
B=36	0.74 ± 0.03	0.75 ± 0.03	0.70 ± 0.03
B=64	0.81 ± 0.02	0.79 ± 0.01	0.72 ± 0.03
B=100	0.81 ± 0.05	0.78 ± 0.05	0.71 ± 0.02
B=144	0.87 ± 0.02	0.82 ± 0.01	0.76 ± 0.02

Table 4.3: Fitness scores achieved by the evolved and trained networks on the three datasets used in subsequent analyses: Spiral selectivity dataset (section 4.3.3), 3D translation/ rotation dataset (section 4.3.4), and Population encoding dataset (section 4.3.5). Values are the mean±standard deviation.

to the postsynaptic neuron, the connection is strengthened because intuitively, the presynaptic neuron *caused* the postsynaptic neuron to fire. If the order is reversed, the relationship between these neurons is uncorrelated and that connection would typically be weakened. In the case of STDP for inhibitory neurons, the order might be reversed since if there is causality, the inhibition should be released (i.e., weaken the synapse). However it should be noted that many variations of timing and order have been observed experimentally.

We examined the evolved STDP parameters for the best performing network individuals with each configuration. In the evolutionary computation process, we searched the parameter space for the amplitude and time constant components of the STDP curves for LTD and LTP. Through the optimization towards an accurate reconstruction of the input spiking patterns, the STDP parameters evolved to show similar patterns across all five network model configurations as measured by the area over the LTP or LTD part of the STDP curves (Figure 4.6). The MT $\rightarrow$ MSTd input connection showed stronger LTD effects than LTP. STDP for the inhibitory connections evolved the opposite pattern, the MSTd $\rightarrow$Inh and the Inh $\rightarrow$ MSTd connections showed stronger LTP effects than LTD. Such biases agree with the hand-designed STDP learning curves used in computational studies, which demonstrated the effect of these types of curves in balancing synaptic strength and encouraging competitive learning in the synapses (Gerstner et al., 1997; Song et al., 2000; Abbott and Nelson, 2000).

These learning curves also contributed to sparseness in the learned representations. The MT $\rightarrow$ MSTd input connection was biased toward LTD, which led to a net effect of depression of all synapses and stabilized network activity by reducing the firing rate of network neurons. With a reduced overall activity in the network, correlated spikes between a particular pair of pre- and postsynaptic neurons are more easily differentiated, which triggered LTP consistently and strengthened the synapse. As a result, after learning through STDP, some synapses were selectively driven to be close to the maximum weight values, and other synapses close to the minimum values. On the other hand, the inter-connection between the MSTd group and the inhibitory neuron group were

biased toward LTP, which tended to strengthen the synapses. Stronger synapses in the feedforward MSTd-Inh connection led to a higher level of activity in the inhibitory neuron group, which when combined with the strong synapses in the feedback Inh-MSTd connection, provided a powerful inhibition on the MSTd neurons. The STDP curves in all three connections encouraged a sparser activity in the MSTd neuron group. Taken together, it suggests that the evolutionary process discovered STDP curves, similar to those observed experimentally and proposed theoretically, that stabilize activity and produce sparse, reduced representations.

4.3.2 Sparse representations in simulated MSTd neurons

Sparse coding describes a coding scheme in which a stimulus could be represented by a small set of neurons, and each stimulus activates different subsets of neurons (Foldiak and Endres, 2008). This is in contrast to dense codes, where every stimulus activates all neurons, and local codes, where each neuron is highly selective to a particular stimulus (Arbib, 1995). Sparse codes are a favorable compromise between the dense and local codes, which **reduce** the overall neural activity required to represent the stimulus space (Vinje and Gallant, 2000; Beyeler et al., 2019).

Beyeler et al. (2016) showed that the MSTd units in the sparse decomposition model formed a sparse population code of the input, and suggested that sparse codes may be employed by MSTd to learn compact and multifaceted representations of the input. In the current study, we also investigated whether the simulated MSTd neurons in the SNN model formed sparse representations of the input activity patterns. We quantified the level of sparseness of the simulated MSTd neurons using the sparseness metrics defined in Vinje and Gallant (2000). The lifetime sparseness metric measured how selective a neuron was to different stimuli, and the population sparseness metric measured how many neurons were activated by any given stimulus. The sparseness values range from 0 to 1, with 1 indicating very selective responses and maximum sparseness, and 0 indicating a dense code. We computed both metrics for the MSTd activity over the validation trials after the

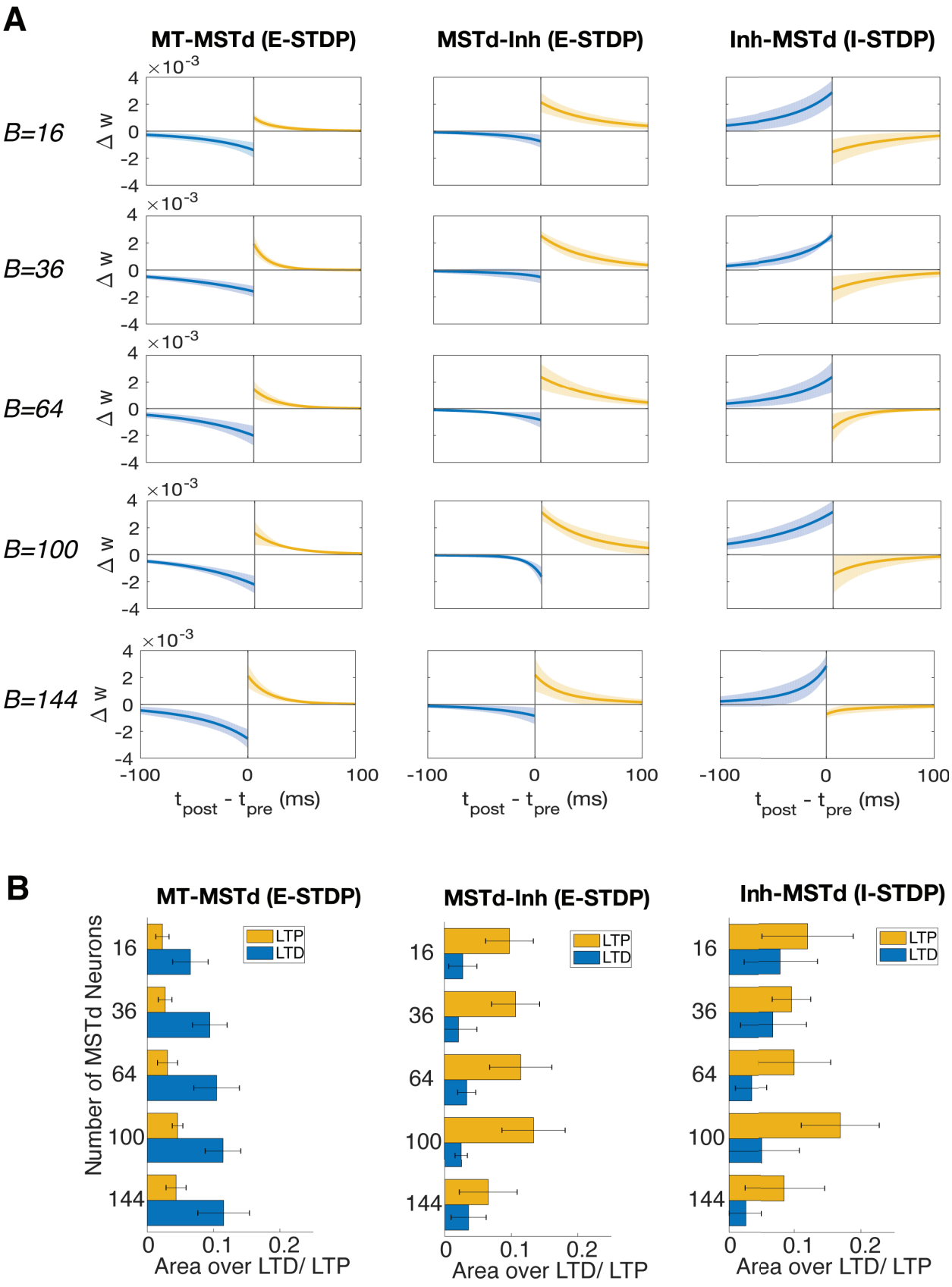

A
MT-MSTd (E-STDP)
MSTd-Inh (E-STDP)
Inh-MSTd (I-STDP)
B=16
B=36
B=64
B=100
B=144
Δ w
×10^-3
4
2
0
-2
-4
-100
0
100
t_post - t_pre (ms)
B
MT-MSTd (E-STDP)
MSTd-Inh (E-STDP)
Inh-MSTd (I-STDP)
LTP
LTD
Number of MSTd Neurons
16
36
64
100
144
0
0.1
0.2
Area over LTD/ LTP

Figure 4.6: Evolved STDP parameters. **A.** Evolved STDP curves visualized by plotting the changes of synaptic weight (Δw) against the time difference between the pre-synaptic (t_{pre}) and post-synaptic (t_{post}) spikes. The blue curves correspond to the LTP component of STDP (Equation 4.7), and the yellow curvescorrespond to the LTD component (Equation 4.8). Solid lines denote the mean values calculated from all evolved network instances (10 network instances for $B = 16$, 5 network instances for $B = \{36, 64, 100, 144\}$), and the shaded areas denote the standard deviation. **B.** Area over the LTD or LTP component of the STDP curves.

models were fully evolved and trained. The same procedure was applied to all configurations of the SNN model ($B = \{16, 36, 64, 100, 144\}$).

As shown in Figure 4.7, network models with a larger MSTd group tended to have sparser representations of the input stimuli both in lifetime sparseness and population sparseness. Sparseness values in both metrics ranged from $\sim$0.5 for the $B = 16$ configuration to $\sim$0.8 for the $B = 144$ configuration. These results indicated that, similar to the sparse decomposition model (Beyeler et al., 2016), the simulated MSTd neurons in all configurations of the SNN model learned a sparse population code of the input activity patterns. The results also indicated that the sparseness values may saturate at the $B = 144$ configuration and not improve for networks with a larger number of MSTd neurons, as the fitness scores increased minimally from the $B = 100$ configuration to the $B = 144$ configuration.

4.3.3 MSTd-like receptive fields and spiral selectivity

We visualized the receptive fields of simulated MSTd neurons by applying population decoding on the connection weight values in the networks evolved with $B = \{16, 36, 64, 100, 144\}$ MSTd neurons (Georgopoulos et al., 1982). The 2D motion selectivity of many simulated MSTd neurons may be predicted from the visualized weight patterns. Figure 4.3 shows the visualized weight patterns of an evolved and trained network individual with the $B = 64$ configuration as an example. Simulated MSTd neurons in the network had large receptive fields and preferred a mixture of translational and rotational flow. The receptive fields depicted a wide range of optic flow pat-

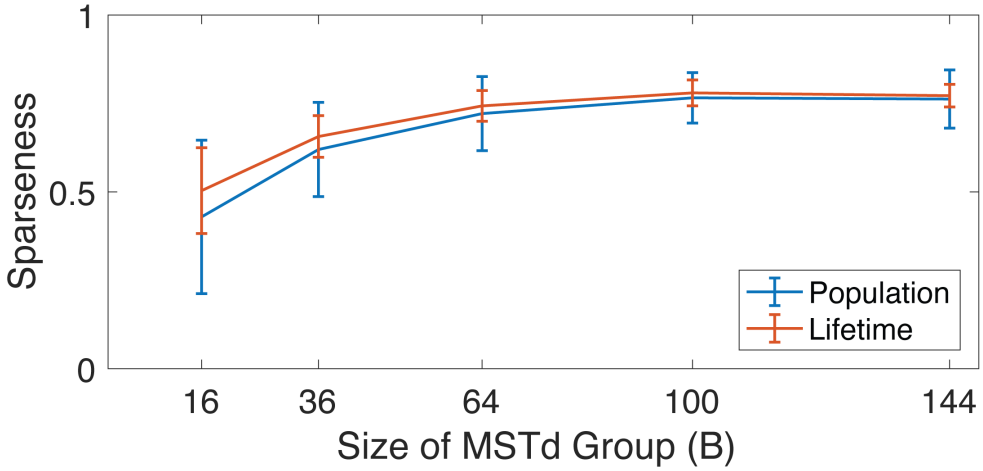


Figure 4.7: Sparseness measurements of different configurations of the SNN model. Population sparseness measured how many neurons were activated by any stimulus, and lifetime sparseness measured how many stimuli any given neuron responds to. The level of sparseness increases with the increased number of MSTd neurons in the network.

terns, including translation, rotation, expansion, contraction, and intermediate of these basic flow patterns. Networks with other configurations ($B = \{16, 36, 100, 144\}$) showed similar weight patterns.

To quantify the selectivity of the simulated MSTd neurons to a continuum of spiral motions, we followed the procedure described in Graziano et al. (1994), where the tuning of MSTd neurons was determined by a Gaussian curve fit (Figure4.8 A and B). We generated a set of eight “basic stimuli" as described in Graziano et al. (1994), which contained expansion, contraction, clockwise rotation, counter-clockwise rotation, and the four cardinal directions of translation (up, down, right, and left). We presented these stimuli to the fully evolved and trained networks, and obtained the response of the simulated MSTd neurons to each stimulus. We first used a criterion that the neuronal response should be greater than 10 % of the maximum response in the population for any of the stimuli to filter out simulated neurons that were not responsive to the basic stimuli. We then fit the tuning curve of each neuron with a Gaussian function. The peak of the Gaussian function corresponded to the preferred spiral motion of the neuron. These analyses were performed on all configurations of the network with $B = \{16, 36, 64, 100, 144\}$ MSTd neurons. In Figure

4.8 we present the results from the networks with $B = 64$ MSTd neurons (results of the other four configurations are shown in Figure 4.9). From five separately evolved and trained network individuals with the $B = 64$ configuration, we obtained the response of 320 MSTd neurons, 317 of which responded significantly to at least one stimulus. We plotted the tuning curve of each MSTd neuron as a function of their response to the spiral motion stimuli, and fit the tuning curve with a Gaussian function. We measured the goodness-of-fit of the Gaussian function by computing the correlation r between the Gaussian function and the tuning curve. We also recorded the width σ of the Gaussian curves and compared them to the real neurons. A large proportion of simulated MSTd neurons had smooth tuning curves and good Gaussian fits: the mean r score for the Gaussian fit was 0.97 $\pm$ 0.04 SD, and 278 out of 317 neurons (87.70 %) had r scores greater than 0.9. The average tuning width for these neurons was 74.09°, with SE of 4.44°. These measurements were comparable to the tuning widths of MST neurons in Graziano et al. (1994) (average width σ =61°, SE=5.9°).

The modeled MSTd neurons showed similar responses to spiral flow fields as those observed experimentally. We estimated the preferred spiral motion of the neuron by finding the peak of the Gaussian curve. Figure 4.8C and Figure 4.8D show the preferred spiral motion of the simulated MSTd population. In Figure 4.8C, each arrow represents the preferred motion of one simulated MSTd neuron. The population of simulated MSTd neurons showed a wide range of preferred spiral motion patterns. We categorized these neurons by their tuning to the eight basic motion types (expansion, contraction, both rotations, and four intermediate directions of spiral motion). Figure 4.8D shows the distribution of simulated MSTd neurons tuned to different spiral motions. Unlike the MST neurons reported in Graziano et al. (1994), where the population showed a bias to expanding stimulus (Figure 4.8A, B), the simulated MSTd population in the SNN model had their preferred spiral motion distributed more evenly for each motion type.

Previous studies (Gu et al., 2006; Takahashi et al., 2007; Gu et al., 2010; Beyeler et al., 2016) suggested that the predominance of expansion-tuned MST neurons may be due to selection bias

in the neuron screening process. Earlier neurophysiological studies of MSTd tended to use expansion stimuli to locate visually responsive neurons in MSTd, and therefore may have led to a biased population of neurons more likely to be tuned to expanding optic flows. We applied the same selection bias and sampled a sub-population of simulated MSTd neurons. In pre-screening process, neurons that had a strong response to the expansion stimulus were more likely to be included in this sub-population. With this simulation of selection bias, we selected 39 neurons from the pool of simulated MSTd neurons in networks with the $B = 64$ configuration. After filtering out neurons that were not significantly tuned to any of the basic motion stimuli, we had 36 simulated neurons for the subsequent analysis. In the sub-population that was pre-screened for expansion, 47 % were expansion-tuned neurons, 50 % spiral-tuned neurons (42 % expanding spiral and 8 % contracting spiral), 3 % rotation-tuned neurons, and 0 % contraction-tuned neurons. The distribution of preferred spiral motion in Graziano et al. (1994) was expansion (40 %), spiral [37 % (expanding spiral 28 %, contracting spiral 9 %)], rotation (16 %), and contraction (7 %). Similar to the data in Graziano et al. (1994), a large proportion of simulated MSTd neurons in the SNN model were tuned to expansion (Figure 4.8E, F). The pre-screened sub-population had tuning curves with good Gaussian fits, with a mean r score of 0.98 ± 0.05 SD. The average tuning width was 64.90°, with SE of 10.82°, which was also comparable to the MST data in Graziano et al. (1994).

4.3.4 3D translation and rotation heading selectivity

Neurophysiological studies characterized the 3D translation and rotation heading tuning of MSTd neurons using visual stimuli that simulated the observer's movement through a 3D cloud of "stars" (Gu et al., 2006; Takahashi et al., 2007). These stimuli were generated following a "translation protocol", which consisted of stimuli that depicted straight translational movements along 26 directions corresponding to all combinations of azimuth and elevation angles in increments of 45°, and a "rotation protocol", which consisted of stimuli that depicted rotational movements along the same 26 directions, with the direction corresponding to the axis of rotation according to the right-

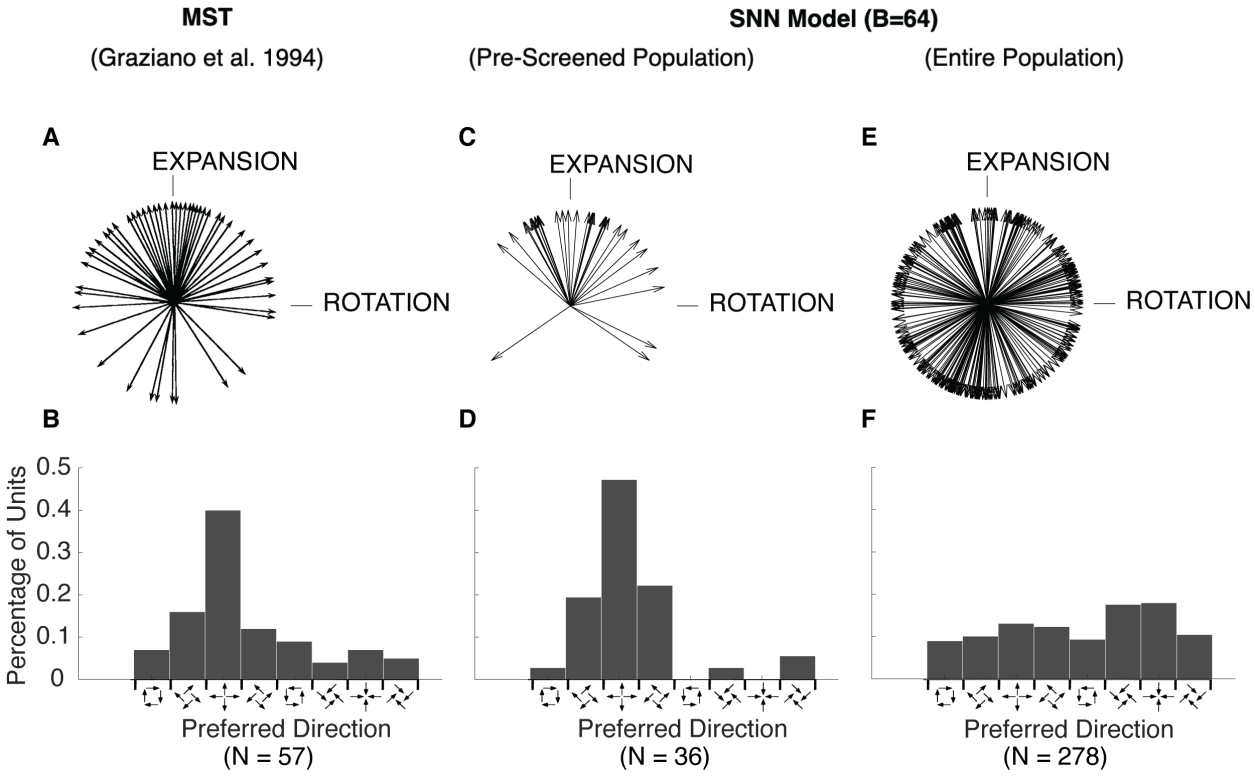


Figure 4.8: MSTd response to spiral flow fields. Plots in the first row visualize the distributions of tuning with arrows spanning the spiral motion space. Each arrow represents one neuron. Histograms in second row visualize the distributions of tuning by showing the percentage of simulated neurons tuned to each type of motion. **A,B.** Reprinted with permission from Graziano et al. (1994). In a population of 57 neurons recorded from the MSTd area, the tuning of MSTd neurons spanned the entire spiral space, with a large proportion of neurons tuned to expanding motions. **C-F.** Analyses of the entire population of simulated MSTd neurons obtained from 5 separately evolved and trained network instances with the $B = 64$ configuration. **C,D.** In a pre-screened for expansion subpopulation of the simulated MSTd neurons, a large proportion of simulated neurons were tuned to expanding motions. **E,F.** In the entire simulated MSTd neuron population, 278 simulated MSTd neurons had significant tuning to spiral stimuli. The preferred spiral directions distributed evenly.

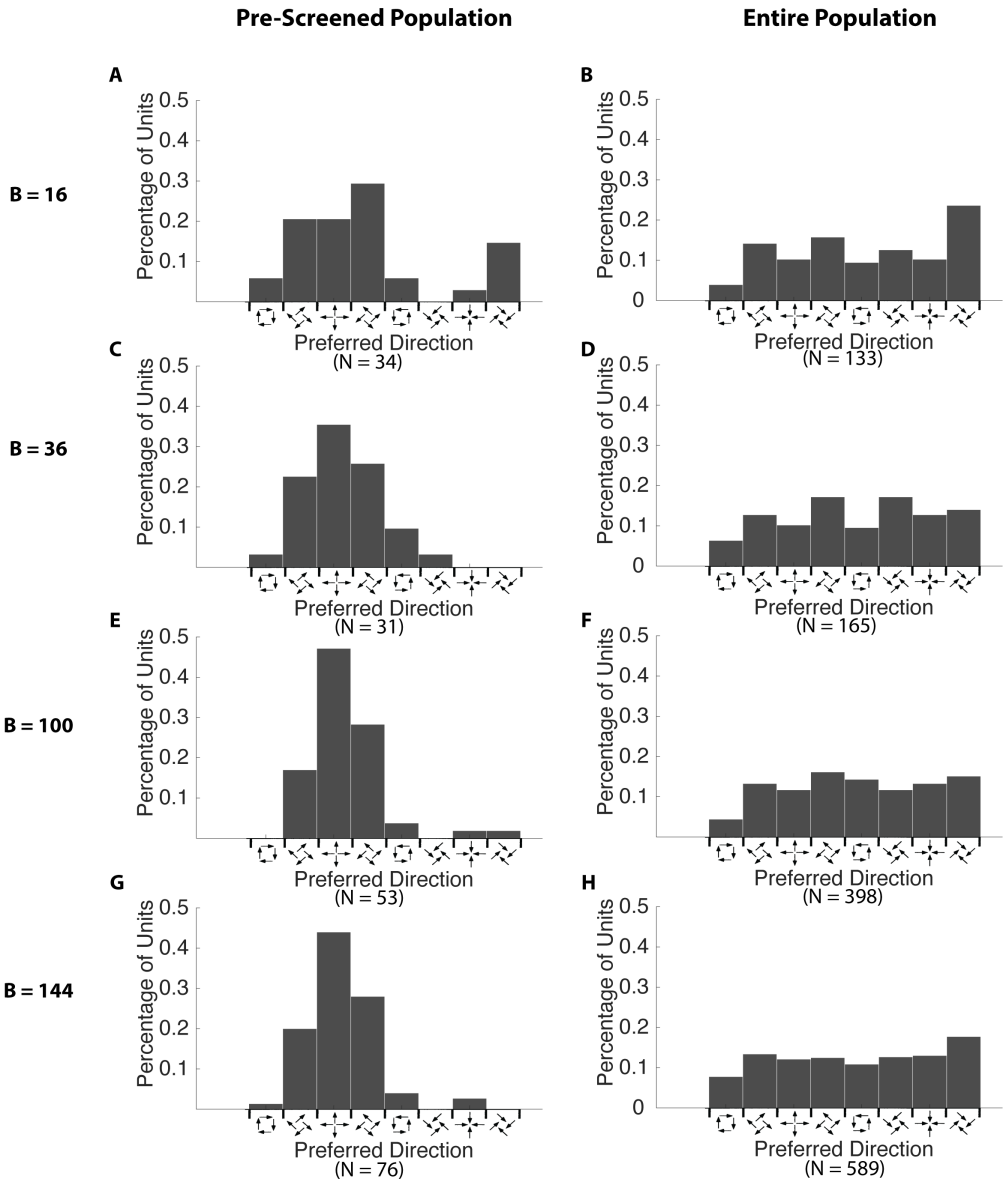


Figure 4.9: MSTd response to spiral flow fields ($B = \{16, 36, 10, 144\}$). The left column shows the distribution of spiral tuning of the pre-screened population. Similar to the $B = 64$ configuration as shown in Figure 4.8D, a large proportion of simulated neurons were tuned to expanding motions. The right column shows the distribution of spiral tuning of the entire population of simulated MSTd neurons obtained in 10 separately evolved and trained network instances with the $B = 16$ configuration, and 5 network instances with the $B = \{36, 10, 144\}$. Similar to the $B = 64$ configuration as shown in Figure 4.8F, the preferred spiral directions distributed evenly.

hand rule. These 26 directions were described by 26 movement vectors spaced 45° apart in both azimuth and elevation on the sphere.

We tested whether the simulated MSTd neurons in the SNN models had similar translation and rotation tuning as experimental observations. We followed the same protocols as the neurophysiological studies and generated visual motion stimuli to test the SNN models (Gu et al., 2006; Takahashi et al., 2007). The translational stimuli were generated with a peak velocity v_z =0.3 m/s, and the rotational stimuli were generated with a peak velocity ω_z =20 °/s. We recorded the activity of the simulated MSTd neurons during the stimulus presentation, and obtained 3D tuning curves for each MSTd neuron by plotting the neuron activity as a function of the azimuth and elevation of the stimuli. As the heading directions were sampled around the sphere, the heading tuning function was spherical. We transformed the data using the Lambert cylindrical equal-area projection, in order to plot the 3D tuning curves on Cartesian axes (Snyder, 1987). Figure 4.10 shows a comparison of 3D translation and rotation tuning in an example macaque MSTd neuron and in an example simulated MSTd neuron from the SNN model ($B = 64$). The tuning curves of the neurons were illustrated as contour maps, where the x-axis represented the azimuth angle, the y-axis represented the sinusoidally transformed version of the elevation angle, and the mean firing rate of the neuron to each stimulus was plotted as a function of azimuth and elevation angles. Similar to MSTd neurons (Takahashi et al., 2007), our simulated MSTd neurons showed broad tuning curves for translational and rotational heading directions. We computed the vector sum of the unit's response to all 26 stimuli to determine the preferred translation and rotation directions for the simulated neurons. The experimentally observed MST neuron shown in Figure 4.10A and B was tuned to 291° azimuth and −18° elevation under the rotation condition, and was tuned to 190° azimuth and −50° elevation under the translation condition (Takahashi et al., 2007). The simulated MSTd neuron also showed strong spatial tuning under both the rotation and translation condition. This particular simulated MSTd neuron shown in Figure 4.10C and D was tuned to 35° azimuth and −27° elevation for rotational heading, and 216° azimuth and −65° elevation for translational heading.

We examined the distribution of preferred directions of the simulated MSTd neurons. Figure 4.11 shows the distribution of translation and rotation direction preferences of MST neurons (Takahashi et al., 2007) and of all 320 simulated MSTd neurons from 5 individually evolved and trained SNN models with the $B = 64$ configuration (results of the other four configurations are shown in Figure 4.12). Each data point in these plots represents the 3D preferred direction of a single neuron. Note that in these plots, 360° in the azimuth angle was equivalent to 0° in the azimuth angle. Similar to the distribution reported in Takahashi et al. (2007), the simulated MSTd neurons also showed significantly non-uniform distributions in azimuth and elevation for both translation and rotation heading preferences (uniformity test, $p < 0.05$). Rotational preferences showed a bimodal distribution in azimuth, clustering at 0° and 180°, corresponding to pitch-up and pitch-down rotations, respectively. Translational preferences had bimodal distributions in both azimuth and elevation, clustering at 0° and 180° in azimuth, corresponding to leftward and rightward movement directions, and ± 90° in elevation, corresponding to upward and downward movement directions. We categorized the simulated neurons with their preferred motion type by finding neurons with preferred directions falling within ± 30° of each of the cardinal axes of the motion type. Similar to the recordings in Takahashi et al. (2007), the majority of the simulated MSTd neurons were tuned to yaw (± 90° in elevation) or pitch (0° and 180° in azimuth) directions for rotational headings, and a smaller percentage of simulated MSTd neurons were tuned to roll motions (90° and 270° in azimuth) (Table 4.4). Under the translation condition, the majority of the simulated MSTd neurons were tuned to lateral (0° and 180° in azimuth) or vertical (± 90° in elevation) translational motions (see Table 4.5). To quantify the strength of heading tuning of single MSTd neurons, we computed the heading tuning index (HTI) value defined in Gu et al. (2006). A HTI ranges from 0 to 1, with larger values indicating stronger heading tuning. Gu et al. (2006) reported that the average HTI value for their sampled MSTd neurons was 0.48 ± 0.16. The simulated MSTd neurons in the networks with the $B = 64$ configuration showed a stronger tuning, averaging 0.66 ± 0.14 and 0.68 ± 0.12 for rotation and translation respectively.

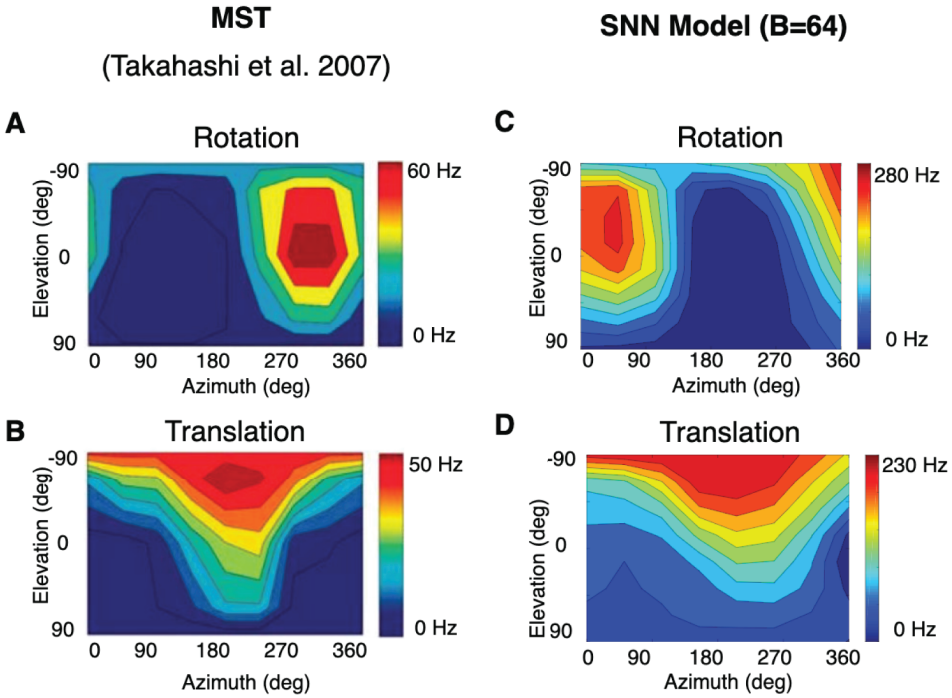


Figure 4.10: 3D translation and rotation tuning of example MSTd neurons. **A,B.** Reprinted with permission from Takahashi et al. (2007). Translation and rotation tuning of an example MST neuron recorded under the visual rotation and translation conditions. **C,D.** Translation and rotation tuning of an example MSTd neuron from a fully evolved and trained SNN model with the $B = 64$ configuration. Each contour map shows the Lambert cylindrical equal-area projection of the original data (Snyder, 1987).

	Yaw	**Pitch**	**Roll**
Takahashi et al., 2007	36/127 (28%)	27/127 (21%)	1/127 (1%)
SNN Model ($B = 64$)	67/320 (20.94%)	95/320 (29.69%)	25/320 (7.81%)

Table 4.4: Number of neurons tuned to each motion type under the visual rotation condition

	Lateral	**Fore-aft**	**Vertical**
Takahashi et al., 2007	57/307 (19%)	20/307 (7%)	76/307 (25%)
SNN Model ($B = 64$)	75/320 (23.44%)	29/320 (9.06%)	73/320 (22.81%)

Table 4.5: Number of neurons tuned to each motion type under the visual translation condition

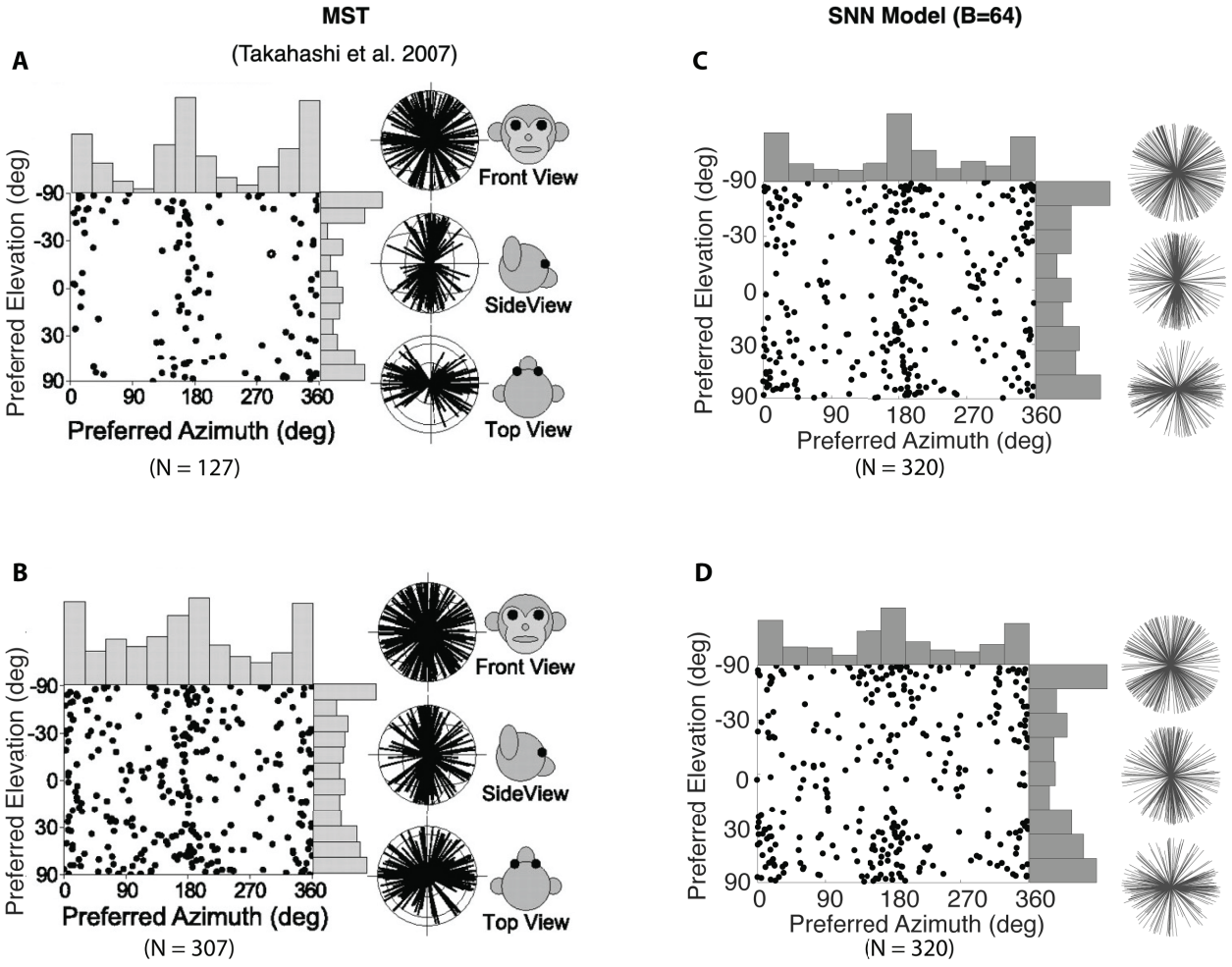


Figure 4.11: Distribution of 3D translation and rotation tuning of MSTd neurons. Each data point in the scatter plot represents the preferred azimuth and elevation angles of one neuron. The histograms along the sides of the scatter plot show the marginal distributions. The 2D projections of unit-length 3D preferred direction vectors were shown in the radial plots, including the front view, the side view, and the top view. Each radial line in these plots represents one neuron. **A,B.** Reprinted with permission from Takahashi et al. (2007). MST neurons recorded under the visual rotation and translation conditions. **C,D.** Simulated MSTd neurons from 5 separately evolved and trained SNN model with the $B = 64$ configuration.

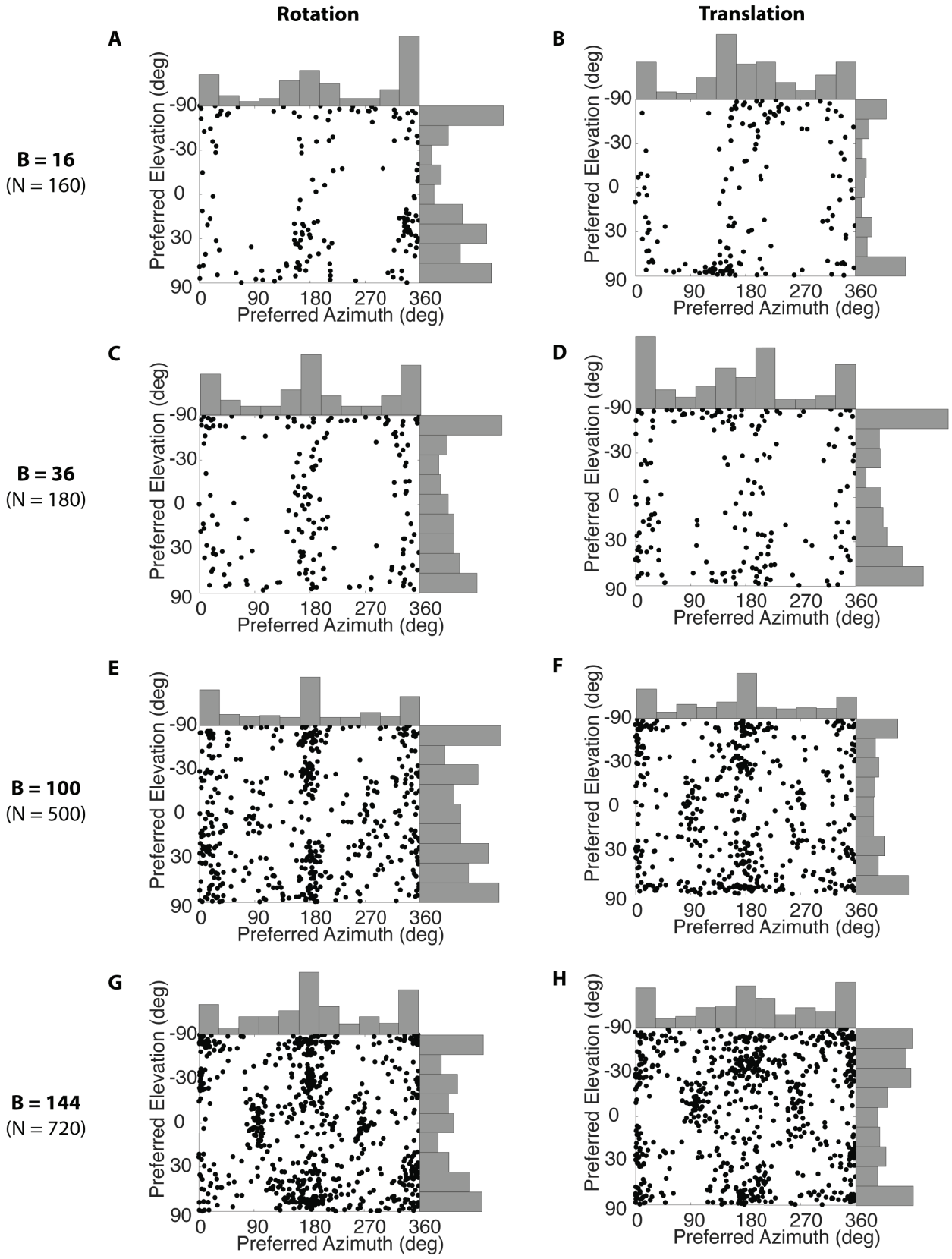

Rotation
Translation
A
B
C
D
E
F
G
H
B = 16
(N = 160)
B = 36
(N = 180)
B = 100
(N = 500)
B = 144
(N = 720)
Preferred Elevation (deg)
-90
-30
0
30
90
0
90
180
270
360
Preferred Azimuth (deg)

Figure 4.12: Distribution of 3D translation and rotation tuning of simulated MSTd neurons ($B = \{16, 36, 10, 144\}$). The left column shows the rotational heading preferences of all simulated neurons in each network configuration, and the right column shows the translational heading preferences. Each data point in the scatter plot represents the preferred azimuth and elevation angles of one simulated neuron. The histograms along the sides of the scatter plot show the marginal distributions.

What might be the source of the non-uniform distribution of heading preferences, especially the over-representation of lateral motion (azimuth of 0° and 180°), and how might these characteristics of the MSTd neuron population contribute to self-motion perception? Gu et al. (2010) suggested that an abundance of MSTd neurons preferring lateral heading directions may account for the observation that primates are most accurate at judging small variations in heading references directly in front of the animal, and less precise for heading around an eccentric reference. Most of their recorded MSTd neurons have broad, cosine-like tuning curves, and the preference to lateral headings caused the peak discriminability of these neurons to lie near straight ahead.

In the evolved and trained models, the simulated MSTd neurons also had broad, cosine-like tuning curves (see Figure 4.10). We tested whether these simulated MSTd neuron populations also maximal discriminability for heading directions near the straight-ahead. We presented motion patterns that simulated eight directions of translation in the horizontal plane (0°, $\pm$ 45°, $\pm$ 90°, $\pm$ 135°, 180°, relative to straight-ahead) to the models and recorded the activity of the simulated MSTd neurons. Following the experimental protocol of Gu et al. (2010), we computed the peak discriminability of individual simulated MSTd neurons, and estimated the precision of heading discrimination by computing population Fisher information from the tuning curves of the simulated MSTd neurons. To obtain smooth tuning curve of the simulated MSTd neurons, we fitted a spline function with 0.01° resolution to the coarsely sampled data (45° spacing). We then calculated the spatial derivative of the spline fit and obtained the tuning curve slope. Peak discriminability was achieved at the steepest slope of the tuning curves. For each heading direction, we computed the population Fisher information based on the derivative of the tuning curve for each simulated neuron, and the variance of the response of each simulated neuron to each heading direction, as

described in Gu et al. (2010). Larger values of Fisher information indicate higher precision in heading discrimination.

In order to make a direct comparison with the neurophysiological data, we adopted the coordinate system used in Gu et al. (2010), where 0° azimuth corresponded to the straight-ahead direction. Figure 4.13 compares the distribution of peak discriminability of recorded MSTd neurons and simulated MSTd neurons in the models with different network configurations. Consistent with the neurophysiologically recorded data, most simulated MSTd neurons had their peak discriminability clustered around forward (0°) and backward (180°) headings. Figure 4.14 shows population Fisher information computed from the heading tuning curves for recorded MSTd neurons and simulated MSTd neurons in the network models. Consistent with the data reported in Gu et al. (2010), there was a clear dependence of Fisher information on reference heading for the simulated MSTd neuron populations, with a peak for forward headings (0°) and a minimum for lateral headings ($\pm$ 90°). The magnitudes of Fisher information differed between the recorded and simulated MSTd neurons, and also across different network configurations, due to the different sample sizes and differences in signal-to-noise ratio. The simulated MSTd neurons diverged from the recorded neurons in that the former population showed higher Fisher information for backward headings (180°).

4.3.5 Population encoding of heading direction

MSTd neurons were suggested to encode perceptual variables such as the focus of expansion (FOE) in optic flow, eye position, and pursuit to support self-motion. Perceptual variables could be decoded with an average accuracy of 2° to 3° based on single-trial response of a population of 144 MSTd neurons (Hamed et al., 2003).

In the current study we focus on investigating whether the simulated MSTd neuron population in the SNN model also encoded FOE robustly. We generated a dataset with 10,000 optic flow fields that simulated various translational motions as described in the physiological study (Hamed

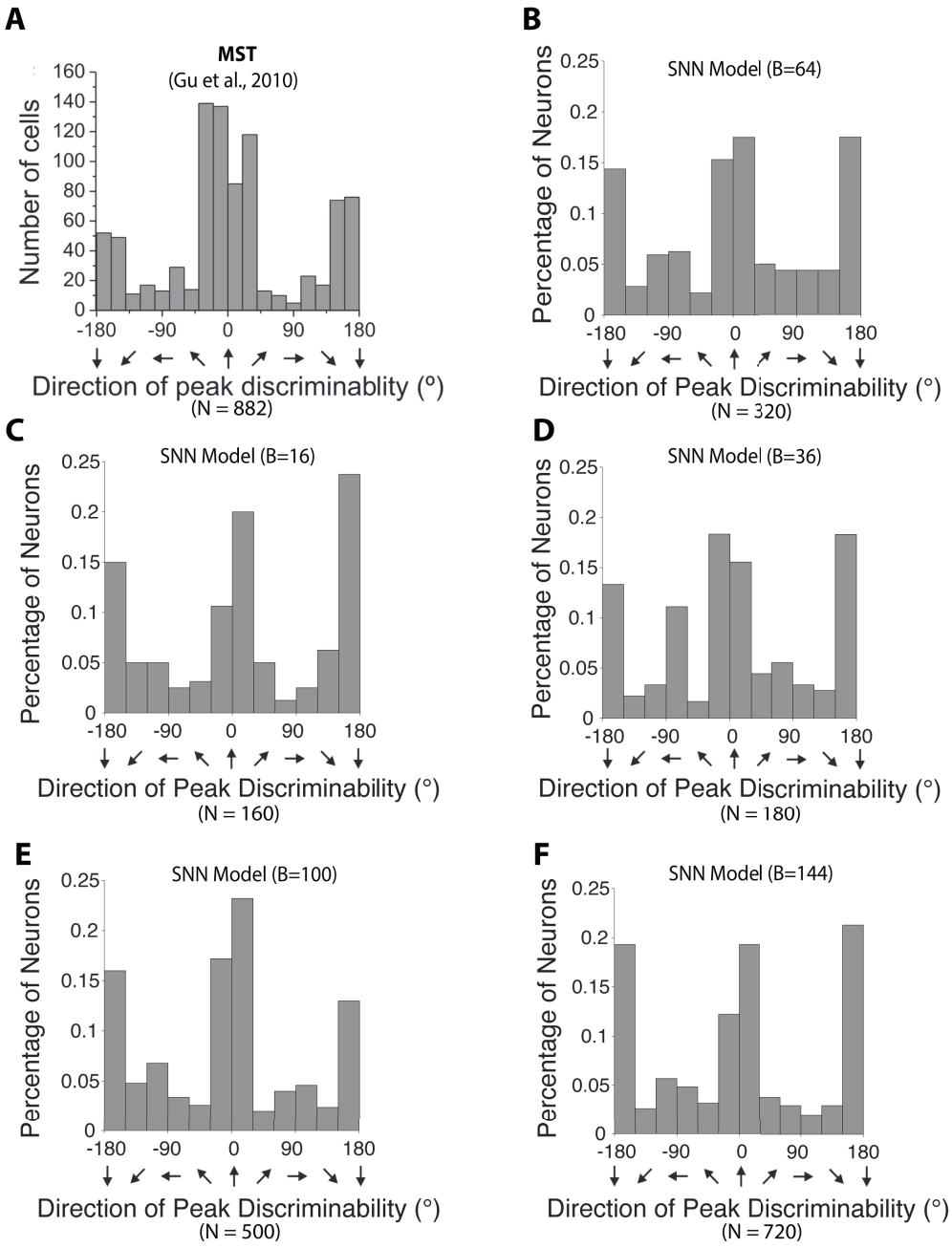


Figure 4.13: Distribution of the direction of maximal discriminability for neurons recorded in macaque MSTd (**A**, reprinted with permission from Gu et al. (2010)) and simulated MSTd neurons in the SNN models (**B-F**). Both the recorded and the simulated neuron populations showed a bimodal distribution with peaks around the forward (0°) and backward (180°) headings.

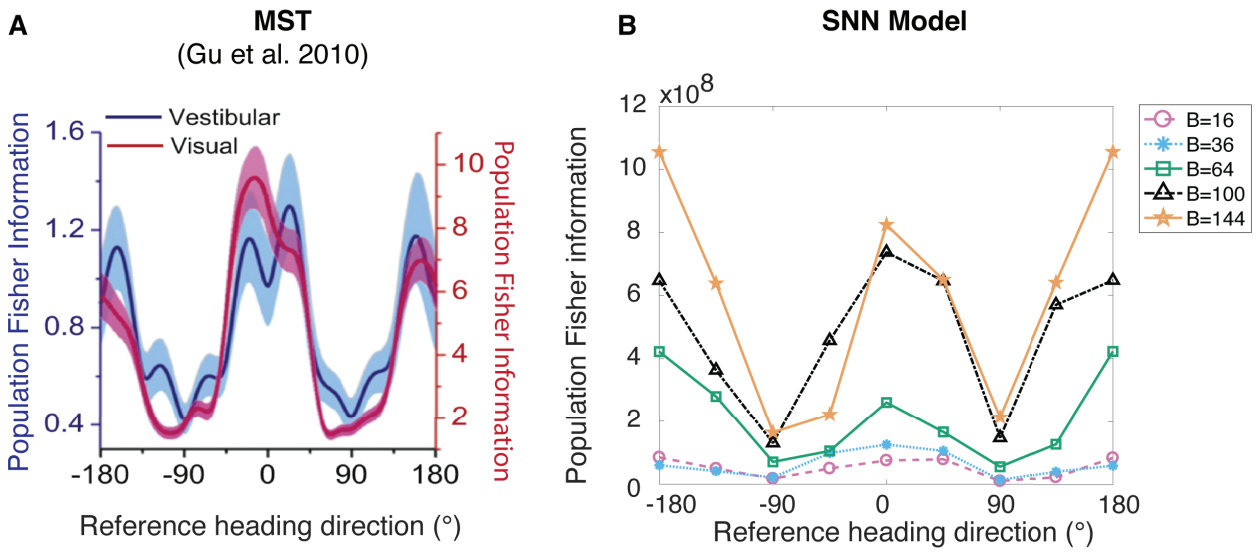


Figure 4.14: Population Fisher information computed from neurons recorded in macaque MSTd (**A**, reprinted with permission from Gu et al. (2010)) and from simulated MSTd neurons in the SNN models (**B**). The error bands in **A** show (95 % confidence intervals derived from a bootstrap procedure.

et al., 2003). The “heading" condition contained flow fields that depicted randomly selected heading directions, with azimuth between 60° and 120°, and elevation between -30° and 30°. These flow fields simulated linear movements toward a back plane located at various distances in meters, $d = \{2, 4, 8, 16, 32\}$ from the observer, with the observer moving at a velocity in m/s, $v_z = \{0.5, 0.75, 1, 1.25, 1.5\}$, without any rotational movements. The goal was to decode the MSTd population activity to predict the location of FOE, which in this case, was the same as predicting the heading direction of the observer. With the fully evolved and trained network models configured with different number of MSTd neurons ($B = \{16, 36, 64, 100, 144\}$), we obtained MSTd group activity responding to the “heading" stimuli. For each configuration, we randomly selected 144 units from the pool of simulated MSTd neuron collected from all separately evolved and trained networks, and used optimal linear estimators with a 10-fold cross-validation procedure to find the linear weights that could decode heading from the activity of this sub-population of simulated MSTd neurons. Table 4.6 shows the prediction error of FOE of the MSTd neurons and simulated MSTd neurons from the network instances with the $B = 64$ configuration. FOE as a key perceptual variable could be decoded with a slightly higher degree of precision than the experimentally

observed MSTd neurons (Hamed et al., 2003). As shown in Figure 4.15, FOE could be decoded with a similar level of accuracy from simulated MSTd neurons in all 5 configurations of the SNN model (Wilcoxon's rank sum test, p = 1 for all pair-wise comparisons).

	FOE (x, y)
Hamed et al., 2003	($3.62° \pm 6.78°$, $3.87° \pm 4.96°$)
SNN Model ($B = 64$)	($0.98° \pm 0.77°$, $1.13° \pm 0.87°$)

Table 4.6: Error in predicting FOE using the activity a population of 144 MSTd neurons sampled from evolved and trained models with the $B = 16$ configuration.

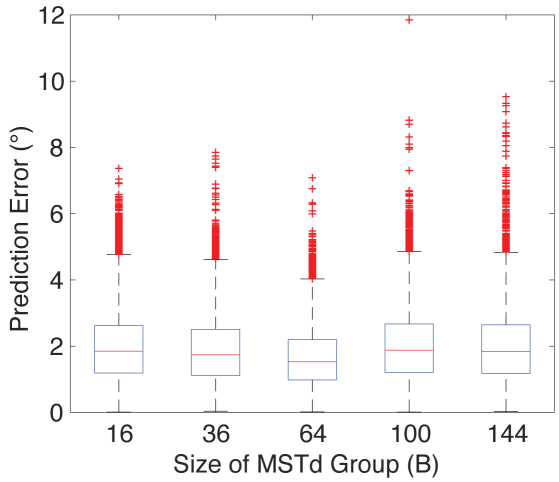


Figure 4.15: Error in predicting FOE (Heading) using the activity a population of 144 MSTd neurons from each configuration ($B = \{16, 36, 64, 100, 144\}$) of the network model. Simulated MSTd neurons from different configurations of the network exhibited similar levels of accuracy in these two tasks. The interquartile range box represents the middle 50% of the values. The line within the box indicates the median. The red data points mark outliers that are 1.5 times bigger than the interquartile range, and the whiskers extend to the most extreme data points that are not considered outliers.

4.4 Discussion

By evolving STDP-H parameters of SNNs, we demonstrated that a model of MSTd was able to account for many experimentally response properties observed in monkeys: 1) Tuning to spiral motions, 2) 3D translation and rotation selectivity, and 3) Encoding of perceptual variables such as

heading. The simulated MSTd neurons learned compressed and efficient representations of input activity patterns with a population that was an order of magnitude smaller than its MT neuron input group. These representations, which accurately reconstructed input stimuli, efficiently encoded complex self-motion that could be used by downstream neural areas. The present results provide a linkage from prior machine learning studies that suggested nonnegative sparse coding can produce similarities to responses observed in the cortex (Hoyer, 2002; Beyeler et al., 2016, 2019), to neurobiologically plausible computations. Specifically, our results suggest that neurobiological plasticity, like STDP-H, may be contributing to dimensionality reduction and sparse coding observed in the brain.

4.4.1 Response properties of simulated MSTd neurons and implications for downstream processing

Simulated MSTd neurons in our SNN model exhibited a variety of response properties observed in neurophysiological studies. Firstly, simulated MSTd neurons had Gaussian tuning spanning the entire spiral space, and many of these neurons showed a strong representation of expanding motions (Graziano et al., 1994). Secondly, similar to the MSTd neurons recorded in visual translational and rotational heading experiments (Gu et al., 2006; Takahashi et al., 2007), the simulated MSTd neurons showed strong tuning to 3D translation and rotation heading directions, with more neurons preferring lateral and vertical translational movements, and pitch and yaw rotational movements. The simulated MSTd neuron had maximal heading discriminability clustered around forward (0°) and backward (180°) headings. Such properties may arise from the SNN models learning to reconstruct the input optic flow dataset, which sampled headings near the center of the visual field for expanding and contracting motion patterns. It's interesting to note these biases in heading preferences and heading discriminability were also observed in macaque MSTd neurons (Gu et al., 2006; Takahashi et al., 2007; Gu et al., 2010), which may suggest that in natural scenes, optic flow patterns with headings near the center of the visual field are more commonly experienced during

forward and backward movements. In the training and validation dataset, we sampled forward and backward movements with the same probability, while backward locomotion is an uncommon activity in natural movements (Perrone, 1986, 1994). This discrepancy may explain the higher precision in discriminating backward headings with the simulated MSTd neurons and the neurons recorded in the macaque MSTd. In future studies, by varying the distribution of the training data and observing how this affects the tuning curves of the MSTd neurons, we will have a better understanding of how observed response properties of MSTd neurons may be an adaptation for efficient self-motion perception. Lastly, FOE as one of the key perceptual variables could be decoded with high accuracy from our simulated MSTd neurons (Hamed et al., 2003). These results suggest that the simulated MSTd neurons encoded information essential for perceiving and guiding self-motion.

These properties emerged as the model evolved unsupervised learning parameters (STDP-H) to reconstruct optic flow patterns. This led to the efficient coding of perceptual features that may be decoded by downstream neurons for motor behaviors. For example, our SNN model showed a continuum of tuning in the spiral space that has been suggested to simultaneously encode the curvature of a path and heading (Layton and Browning, 2014). This allows MSTd to perform trajectory estimation and guide self-motion along complex trajectories. Heading tuning that spanned the entire translational and rotational movement space allows for accurate heading direction perception. The sparse, reduced representations of the global motion pattern allows for efficient encoding of key locomotion parameters including the heading direction. These observations are consistent with the hypothesis that MSTd transforms stimulus information into representations more directly linked to perception and action, supported by the connections between MSTd to many cortical and subcortical areas (Boussaoud et al., 1990; Felleman and Van Essen, 1991 Jan-Feb; Wild and Treue, 2021).

A limitation of the current study is that heading perception with the simulated MSTd neurons was only studied under the locomotion condition without eye movements. Previous studies have

shown that although eye rotations during smooth pursuit distort the optic flow patterns on the retina, humans and nonhuman primates are able to perceive their heading direction correctly (Royden et al., 1992; Britten and Van Wezel, 2002). It will be interesting to investigate whether the simulated MSTd neurons could predict heading in the presence of eye movements, and whether retinal mechanisms are sufficient to discount distortions to the optic flow fields, as suggested in previous computational and neurophysiological studies (Royden, 1997; Perrone and Stone, 1998; Beyeler et al., 2016; Manning and Britten, 2019). The stimulus paradigm used in this study was not well equipped for such an analysis, as the stimuli did not provide meaningful distinction between head rotation and eye rotation.

4.4.2 Relating STDP-H to Nonnegative Sparse Coding in the brain

It has been suggested that STDP-H may lead to sparse, reduced representations in the brain (Carlson et al., 2013; Beyeler et al., 2019). Indeed, our model showed the ability to accurately reconstruct the input stimuli in a neuron population by activating a small subset of neurons (population sparseness), which were active on only a subset of stimuli (lifetime sparseness). In our SNN model, the high-dimensional visual motion input, represented by the population activity of 9000 MT neurons, projected to a small group of MSTd neurons ($B = \{16, 36, 64, 100, 144\}$). The simulated MSTd neurons were pressured to reduce the dimensions of the input and find efficient representations that retained relevant features in the input space. In addition, sparse coding was encouraged by the synaptic competition induced by STDP at the synapse level, and by feedback inhibition (Finelli et al., 2008; Chistiakova et al., 2015). Also, the STDP curves in our SNN model that modulated the MT→MSTd connections evolved to have stronger LTD than LTP, which led to a pruning effect on the connection weights by depressing uncorrelated spiking activity that led to sparser activity (Feldman, 2000, 2012). Finally, homeostatic synaptic scaling preserved the relative weight differences between synapses on the same postsynaptic neuron and stabilized STDP learning (Carlson et al., 2013).

The biological brain operates under tight metabolic constraints and is often pressured to find strategies for accurate information encoding which also allows for efficient transfer, while minimizing the energy costs (Krichmar, 2019). Sparse coding is one such strategy that reduces neural activity without sacrificing performance (Graham and Field, 2007). The sparse coding strategy was demonstrated to be effective in encoding sensory input and was suggested to be ubiquitous throughout the brain (Olshausen and Field, 2004). Dimensionality reduction is a strategy to transform high-dimensional and complex information into a compressed form that is easier to transfer and read out at subsequent levels of processing. The sparse coding scheme implemented in Olshausen and Field (2004) used a greater number of neurons than the dimensions of the input space to represent the input, suggesting “dimensionality expansion". This may seem contradictory to the dimensionality reduction strategy. However, Hoyer (2002) combined the two strategies to form the Nonnegative Sparse Coding (NSC) method, which decomposes multivariate data into non-negative sparse components. Hoyer (2004) further showed that explicitly incorporating sparseness constraints is important for learning nonnegative representations that match the underlying elements of the data. NSC has been demonstrated to reproduce receptive fields of neurons in the primary visual cortex (V1) by decomposing the activities of ON- and OFF-channels in response to natural image patches (Hoyer, 2002). NMF with sparsity constraints implements a form of NSC, which when applied to MT-like input, resulted in sparse, parts-based representations of the optic flow resembling the receptive fields of experimentally observed MSTd neurons (Beyeler et al., 2016). Outside of the visual cortex, NSC was suggested to be employed by many other sensory areas in the brain to encode external stimuli (Beyeler et al., 2019).

The scheme of dimensionality reduction and sparse coding induced by STDP-H in our SNN model is consistent with NSC. NSC describes the input data as a superposition of a set of sparsely activated basis functions and requires that the basis functions as well as the activation values to be nonnegative (Hoyer, 2002). The input optic flow patterns resulting from locomotion, despite their high dimensionality encoded in the population activity of idealized MT neurons, had some statistical regularities. For example, Beyeler et al. (2016) showed that these optic flow patterns could

be reconstructed with a linear combination of a set of "basis flow fields". These basis flow fields captured the key features within the data and represented the data in a lower dimensional space. In the present study, we demonstrated that STDP-H allowed the spiking neurons to detect and extract these statistical regularities among the spike trains, and developed representations of the input optic flow fields similar to those obtained through NSC. Together with the sparse decomposition model proposed in Beyeler et al. (2016), the SNN model presented in the present study challenges the prevailing views about the MST area in macaque cortex, by showing that prominent neuronal responses traditionally attributed to specialized self-motion "templates" might instead be a by-product of neurons performing dimensionality reduction on their inputs. Our work agrees with the hypothesis that STDP-H may be a biologically plausible neural mechanism that implements NSC in the brain (Carlson et al., 2013), and to our knowledge, may be the first computational study to demonstrate the link between the STDP-H synaptic learning rules and the NSC neural encoding scheme.

4.4.3 Model alternatives

A number of computational models have been proposed to reproduce the motion selectivity of MSTd neurons. Examples of these models include a model trained with unsupervised Hebbian learning (Zhang et al., 1993), the "multi-cause" model that took the form of an autoencoder (Zemel and Sejnowski, 1998), and a two-layer back-propagation network that was trained in a supervised manner to approximate tuning curves of MST neurons (Beardsley and Vaina, 1998). Alternatively, Perrone (1994); Perrone and Stone (1998) proposed the "template model" which constructed MST units that each encoded a specific combination of heading and rotation through combining the input from specifically arranged MT-like units. By modeling the MSTd neurons' responses to a set of novel continuous optic flow stimuli, Mineault et al. (2012) showed that nonlinear integration, especially compressive integration of MT input led to the complex selectivity observed in MST neurons. Layton and Browning (2014) defined MSTd hypercolumns that had receptive fields cen-

tered at different locations of the visual field and possessed selectivities across a spiral space of motions. Mineault et al. (2021) showed that a deep neural network trained to predict self-motion parameters could account for the selectivity of neurons in the dorsal visual stream.

Here, we employed a novel modeling approach of evolving synaptic learning rules in a SNN, which posed minimal assumptions on the selection of model parameters. Selectivity and receptive fields of the simulated MSTd neurons in our models were emergent properties without hand-crafted templates, which makes our modeled neurons flexible and generalizing to other complex tasks. This approach is generalizable and has been used to replicate neural recordings in the retrosplenial cortex (Rounds et al., 2018) and the medial temporal lobe (Chen et al., 2021).

In summary, we showed that MSTd-like response properties emerge from a SNN model that was trained with STDP-H and evolutionary computation, with an objective of reconstructing the input optic flow fields. Simulated MSTd neurons in this model formed a sparse population code of the input activity pattern, and encoded perceptual variables important for self-motion perception. This model demonstrated that STDP-H reduced the dimensions of input stimuli and allowed spiking neurons to learn sparse and efficient representations, in a similar way that NSC performs decomposition of complex input. It suggests a biologically plausible method for how these and other representations in the brain emerge.

Chapter 5

Differential Spatial Representations in Hippocampal CA1 and Subiculum Emerge in Evolved Spiking Neural Networks

5.1 Introduction

Rodents flexibly and reliably navigate in the world by using a variety of available spatial information. Effective spatial navigation is supported by localizing oneself in the environment, knowing the current direction of movement, and inferring about the progress along routes leading to the destination. Several regions in the rodent brain have been identified as important for spatial navigation, including the hippocampal sub-region CA1 and the subiculum (SUB). Lesions of both

The contents of this chapter were originally published as a paper titled “Differential Spatial Representations in Hippocampal CA1 and Subiculum Emerge in Evolved Spiking Neural Networks" in the 2021 International Joint Conference on Neural Networks (IJCNN) Proceedings (Chen et al., 2021). The inclusion of this paper is in compliance with the permission policy of the original publisher.

regions cause deficits in navigational abilities and a loss of accurate localization ability (Morris et al., 1990). On the single cell level, place cells have been identified in both regions (O'Keefe and Dostrovsky, 1971; Sharp and Green, 1994). Place cells were shown to signal the allocentric position of the animal during navigational tasks. These cells fire selectively in specific locations in the environment, and the population ensemble activity can be used to decode the animal's movement (Wilson and McNaughton, 1993). However, compared to CA1, SUB place cells showed larger, less specific place fields (Potvin et al., 2007), and exhibit more directional modulation for activity within those fields (Sharp and Green, 1994; Olson et al., 2017).

A number of computational models have been proposed to explain the emergence of CA1 place cells (Arleo and Gerstner, 2000; Burgess and O'Keefe, 2011; Tsodyks and Sejnowski, 1995), but fewer computational studies concern the spatial representations in SUB neurons. As biologically detailed models, Spiking Neural Networks (SNNs) have been used to investigate sensory processing in the brain (Xu et al., 2020) and to model brain activities (Kasabov, 2014). Different methods of optimizing SNNs have also been proposed, such as unsupervised learning with spike timing dependent plasticity (STDP) (Diehl and Cook, 2015) and evolving the structure of the network (Kasabov, 2014). Our method is unique that we evolve the synaptic learning rule in SNNs to map recorded behavioral data to neuronal data.

In this chapter, we extend the method of evolving SNNs introduced in (Rounds et al., 2018) to replicate neural dynamics of CA1 and SUB neurons. We use datasets recorded from the CA1 and SUB while rats performed a complex navigational task on a Triple-T maze (Fig. 5.1) (Olson et al., 2021). Neural circuits of both regions are modeled with SNNs optimized by evolving spike timing dependent plasticity with homeostatic scaling (STDP-H) parameters using evolutionary algorithms (EAs). In the SNN models, behavioral variables including the allocentric position of the animal and self-motion related variables such as the head direction and linear/angular velocity of the animal serve as the input to the network, and a recurrently connected group of excitatory spiking neurons is tuned to replicate the CA1 and SUB neural activity. Our results suggest that the same spiking

neural network modeling framework can be used to model different brain regions related to spatial navigation. In addition, an analysis of the connection weights and the results from ablation studies are suggestive of how CA1 and SUB integrate sensory information differently to form spatial representations. The CA1 region fires sparsely and less actively with higher spatial information, whereas the SUB is more active and directionally selective. These differences are captured in the connection weights and the evolved STDP-H parameters of these regions.

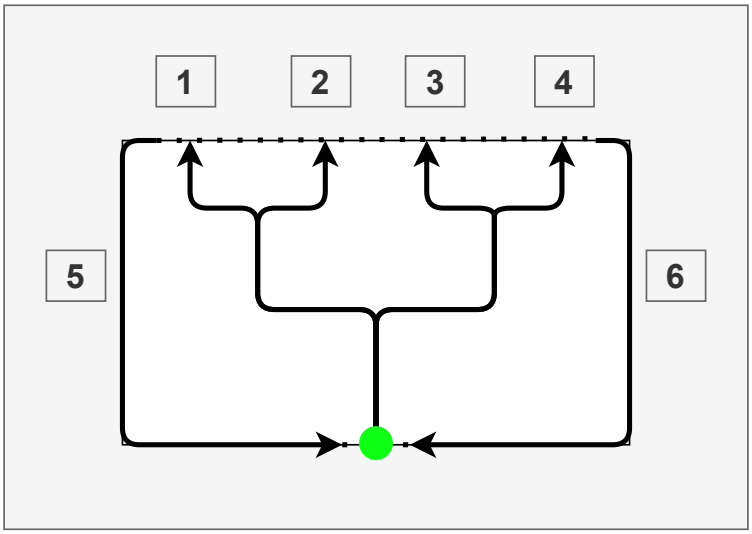


Figure 5.1: Schematic drawing of the navigational task. The rat would start from the bottom (indicated by a green dot), take one of the four internal routes (i.e., Routes 1-4) to get to a reward site, and return to the starting point via one of the two return routes (i.e., Routes 5-6). The rats demonstrated remarkable navigation ability and working memory capacity by visiting all four reward sites with minimal repeats.

5.2 Methods

We took a unique modeling approach of evolving hyper-parameters in spiking neural networks. Spiking neural networks are preferable in this study as they accurately captured the nature of our datasets, which included spike trains that were distributed into spatial bins and processed with a smoothing function (Olson et al., 2017). We processed the spike trains in our spiking models following the same procedure to ensure a direct comparison with the recorded data. The models

were optimized with an approach combining evolutionary algorithms and unsupervised learning using STDP and homeostatic synaptic scaling (STDP-H) (Rounds et al., 2018). The evolutionary algorithms searched for optimal hyper-parameters for the STDP-H learning rule, without directly updating the connection weights in the network. To ensure the reliability of our modeling and parameter search process, we carried out 5 independent evolutionary runs for each modeled region, each with a population size of 15 individual networks that underwent 50 generations of evolutionary algorithms.

All simulations were performed with the CARLsim 4 spiking neural network simulator (Chou et al., 2018). CARLsim 4 includes a parameter tuning interface (PTI) that links to an evolutionary computation library called ECJ (Luke, 2017).

5.2.1 Network model

The network model contained 1282 neurons in total: 640 excitatory neurons modeled as regular spiking (RS) Izhikevich spiking neurons, 160 inhibitory neurons modeled as fast spiking (FS) Izhikevich spiking neurons (Izhikevich, 2003), and 482 input neurons modeled as Poisson spike generators. The input layer contained four types of behavioral inputs: 450 neurons for allocentric position (Pos), 12 neurons for angular velocity (AV), 12 neurons for linear velocity (LV), and 8 neurons for head direction (HD). Each input group was connected to both the excitatory and inhibitory groups. The inhibitory neuron group provided feed-forward inhibition to the excitatory neuron group, while the excitatory neuron group had recurrent excitatory connections within its own group. Neuron groups were sparsely connected with a probability of 0.1 (Fig. 5.2).

Input streams to the SNNs represented the kinds of information processed by connected regions. For example the hippocampal sub-region CA3 provides positional information (Leutgeb et al., 2007), the medial entorhinal cortex (MEC) provides velocity related information (Kropff et al., 2015), and the anterior thalamic nucleus (ATN) provides movement and head direction related

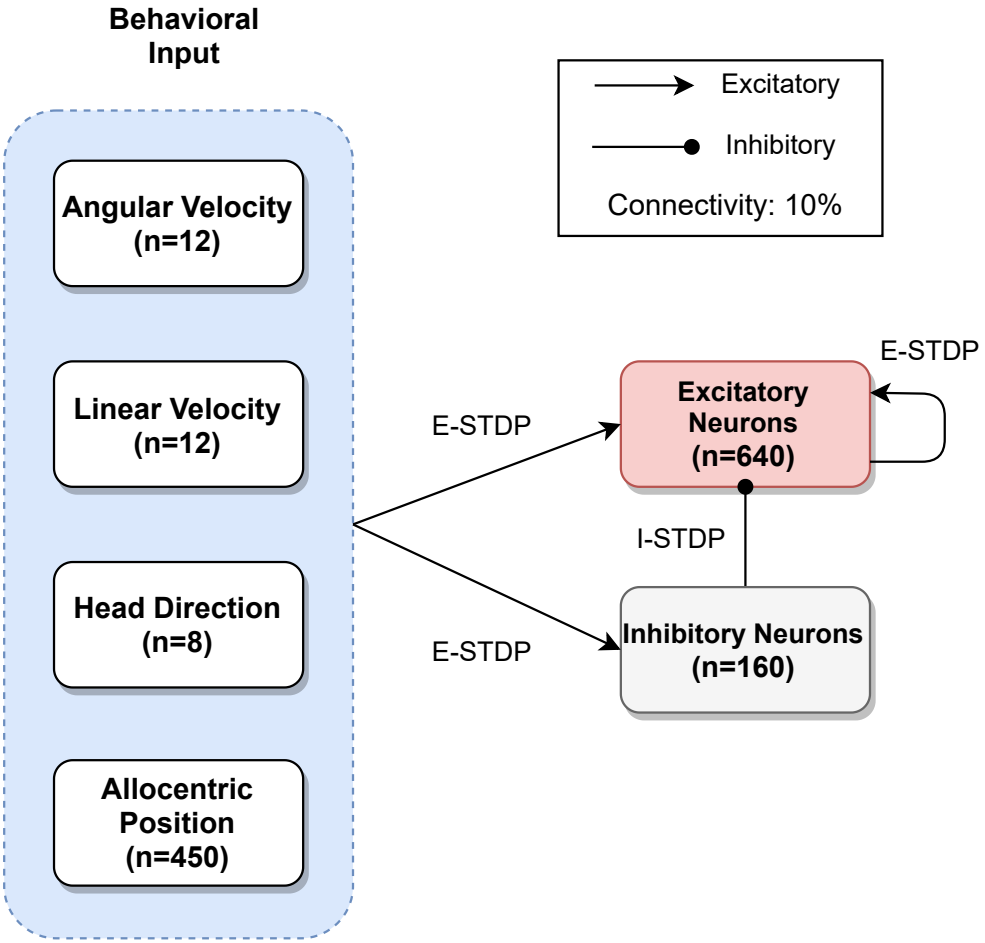


Figure 5.2: Network architecture. This network contains four input neuron groups, representing four types of behavioral variables: angular velocity (AV), linear velocity (LV), head direction (HD), and allocentric position (Pos) of the rat. The input neuron groups are connected to an excitatory neuron group and an inhibitory neuron group. Neurons within and between groups have a connection probability of 0.1.

information (O'Mara, 2005; Frost et al., 2020) to CA1 and SUB. We created the tuning curves for each type of input streams following the same fashion as in (Rounds et al., 2018). Parameter values were set to allow the tuning curves to cover the entire value range in our datasets and elicit varying neuronal responses. The analog response from the tuning curves was converted into spike trains using a Poisson spike generator.

Tuning curves of AV and LV neurons were modeled as Gaussian functions of the rat's angular/-linear velocity. If the rat was moving at a velocity s, the activity of neurons $i = 1...N$ ($N = 12$) in these two groups was given by:

$$f_i(s) = r_{\max} \times exp\left(-\frac{1}{2}\left(\frac{s - s_i}{\sigma}\right)^2\right) \tag{5.1}$$

where $r_{max} = 40$ is the maximum firing rate of the neuron population, and s_i is the preferred velocity of neuron i. Standard deviation of the curve σ was set to be 6 and 20 for AV and LV respectively.

Tuning curves of HD neurons were modeled with cosine functions. With an input head direction , the activity of neurons $i = 1...N$ ($N = 8$) in the head direction input group was given by:

$$f_i(\theta) = \begin{cases} r_{max} \times \cos(\theta - \theta_i) & ,\text{if} \quad |\theta - \theta_i| <= \frac{\pi}{4} \\ 0 & ,otherwise \end{cases}, \tag{5.2}$$

where $r_{max} = 60$, and θ_i represented the preferred direction of neuron i.

Tuning curves of Pos neurons were modeled as 2D-Gaussian functions of the rat's position in the maze. For an input position p, which represented the (x, y) values in the maze in Cartesian coordinates, the activity of neurons $i = 1...N$ ($N = 450$) in the position input group was given by:

$$f_i(p) = r_{\max} \times exp\left(-\frac{1}{2}\left(\frac{||p - p_i||}{\sigma}\right)^2\right) \tag{5.3}$$

where $r_{max} = 40$, and $||\cdot||$ denotes the Euclidean norm. p_i is the preferred position of neuron i, and σ as the width of the tuning curve took the value 25 in this study.

5.2.2 Evolutionary computation

Instead of directly evolving the connection weights, we employed a "learning to learn" paradigm and evolved a total of 20 hyper-parameters in our network. The hyper-parameters in the network fell into 3 categories: (1) Parameters of the STDP learning rule, which included the amplitude parameters $A+$ and $A-$, and the time decay constants $\tau+$ and $\tau-$. Optimal values for these parameters were searched for the Excitatory-STDP that projected into both the excitatory (EE) and inhibitory (EI) neuron groups, and the Inhibitory-STDP that projected into the excitatory (IE) neuron group. (2) Parameters of the homeostatic synaptic scaling rule that were applied to the excitatory and inhibitory neuron groups, which included the target firing rate R_{target} and the time scale parameter T. (3) The maximum connection weights for the inter-group connections (i.e., Inp $\rightarrow$ Exc, Inp $\rightarrow$ Inh, Exc $\rightarrow$ Exc, Inh $\rightarrow$ Exc).

Models were optimized through 50 generations of evolutionary algorithms. In the first generation, the evolutionary algorithm initialized 15 networks, each of which went through a training and testing phase and obtained a fitness score. Parameters of the best performing 3 networks were then used to generate a new generation of 15 network individuals using the (μ, λ) Evolutionary Strategy (ES) (Luke, 2017).

We used a multi-parent $(\mu + \lambda)$ Evolution Strategy (ES) model implemented in ECJ to optimize the network (Luke, 2017). In each generation, the fitness scores from $(\mu + \lambda) = 18$ individual networks were collected, and the top $\mu = 3$ individuals were selected as the parent population to produce the next generation of $\lambda = 15$ individuals. For the reproductive process, we adopted a combination of mutation and cross-over, which added 1-dimensional Gaussian noise to each hyper-parameter with a mutation probability of 0.5 and a cross-over rate of 0.6. This mechanism allowed us to get a good balance between exploring the parameter space and keeping the useful gene values already found.

5.2.3 Training and testing of the model

We used a subset of the data in (Olson et al., 2021), which included 5430 trials from 32 recording sessions for CA1, and 5908 trials from 49 recordings for SUB. Each trial contained neuronal activity recorded from one brain region and behavioral variables of the animal as it traversed one of the six routes of the maze shown in Fig. 5.1. For both CA1 and SUB datasets, we shuffled the data based on trials and split them in half to create a pool of training data and a pool of testing data.

In each evolutionary algorithm generation, a population of networks with different hyper-parameters went through a training and a testing phase, and the fitness scores of the networks were evaluated by the evolutionary algorithm to generate the hyper-parameters for the next generation. The training and testing datasets each consisting of 600 trials (i.e., 100 trials for each route) were re-sampled every generation from the training and testing data pools respectively. In the training phase, behavioral data were fed into the network and STDP-H learned associations between neurons and stabilized network activity. STDP updated the connection weights based on the temporal distance of pre- and post-synaptic spikes, and homeostatic scaling modified the weights in a multiplicative manner based on the post-synaptic firing rate (Carlson et al., 2013). During testing, we disabled synaptic plasticity and froze the connection weights. We presented the network with different behavioral and neurophysiological data from the training phase. To determine how well the simulated neurons resembled the experimentally observed neurons, a Pearson correlation coefficient, ρ, was computed between the mean firing rate of excitatory neuron activity in the SNN and experimentally observed neurons. Using a greedy approach, we determined a match between a simulated neuron and an experimentally observed neuron based on the highest correlation value and each neuron could only be matched once. After all experimentally observed neurons found a match, a fitness value of the network was calculated by summing the ρ values of all matched neuron pairs:

$$y = \sum_{i}^{N} \rho(\bar{R}^{i}_{real}, \bar{R}^{i}_{match}) - L \tag{5.4}$$

where L was a penalty for unrealistically high firing rates, which only applied when the maximum mean firing rate of one of the excitatory neurons R_{exc} exceeded the threshold firing rate $R_t = 100$ Hz:

$$L = \begin{cases} max(\bar{R}_{exc}) - R_t & \text{, if} \quad max(\bar{R}_{exc}) > R_t \\ 0 & \text{, otherwise} \end{cases}, \tag{5.5}$$

Although only a subset of neurons in the excitatory neuron group were matched to the recorded neurons, the entire neuron group was intended to model a larger population of CA1 and SUB neurons. Neurons that were not explicitly matched to the recorded neurons were expected to have similar response properties as those that were explicitly matched.

5.2.4 Positional reconstruction matrix

Population analysis of the neuronal activity was conducted by comparing the positional reconstruction matrices of the simulated neurons in the models and experimentally observed neurons in our datasets (Cowen and Nitz, 2014) . We concatenated the neuronal activity in every positional bin of all six routes and computed mean firing rate vectors for every neuron based on odd and even trials. We obtained mean rate matrices $R_{odd} \in \mathbb{R}^{n \times m}$ and $R_{even} \in \mathbb{R}^{n \times m}$ for the entire neuron population, where n represented the number of neurons in the population, and m represented the number of positional bins in the maze. Each column in the matrix represented the population ac-

tivity in a specific positional bin. We then computed the positional reconstruction matrix based on the columns of the odd trial ensemble rate matrix and the even trial one:

$$M = \rho(R_{odd}, R_{even}), \quad M \in \mathbb{R}^{m \times m} \tag{5.6}$$

where $\rho(\cdot)$ computes the Pearson correlation coefficient.

Comparing the positional reconstruction matrix of simulated neurons and experimentally recorded neurons allowed us to gauge how well the response properties of simulated neurons resembled those of recorded neurons. A similarity score was obtained by converting the matrices into column vectors and correlating the two vectors:

$$g = \rho(M_{simulated}, M_{recorded}), g \in \mathbb{R}^{1 \times 1} \tag{5.7}$$

Different from the fitness function (Equation 5.4), this measurement took into account neurons that were not explicitly matched to the recorded neurons, and thus also tested whether the learned response properties generalized to the entire simulated neuron population.

5.3 Results

5.3.1 Evolved networks captured properties of both regions

SNN models were optimized such that a subset of the excitatory neurons had firing patterns aligned with those of experimentally recorded neurons (295 neurons in the CA1 dataset, and 382 neurons in the SUB dataset). As described in Section 5.2.3, fitness function of the network was defined to be the sum of pairwise correlation values of all simulated-recorded neuron pairs, with a penalty for high firing rates of the simulated neurons (Equation 5.4). The highest fitness scores that could be achieved by the CA1 and SUB models were 295 and 382 respectively. These scores correspond to a perfect correlation value of 1 between the modeled and experimentally recorded neuron firing rates for all neurons in each dataset. To compare the performance of the two models, the fitness values were normalized by the number of neurons in their corresponding brain regions.

We conducted 5 evolutionary runs for each model. With a population of 15 individual networks, the best fitness score in the first generation averaged to be 140.44 for the CA1 model and 136.83 for the SUB model. As shown in Fig. 5.3, by 50 generations, the networks achieved an averaged fitness score of 186.97 for the CA1 model, and 213.48 for the SUB model, corresponding to a mean Pearson's ρ value of 0.63 and 0.56 respectively (normalized to the number of recorded neurons in each brain region). The fitness scores were comparable to those reported in (Rounds et al., 2018). These scores showed that the firing patterns of experimentally observed neurons were captured by a subset of neurons in the excitatory neuron group. The networks also showed a generalization ability, as excitatory neurons that were not explicitly matched to the recorded neurons also showed response properties similar to those observed in CA1 and SUB.

Similar to rodent recordings(Olson et al., 2017, 2021), we observed spatially selective place cell responses in the simulated CA1 and directionally modulated responses along maze axes in the simulated SUB (Fig. 5.4). Simulated neurons in the CA1 model were mostly quiet in other positions,

and had lower firing rates than the simulated SUB. Simulated neurons in the SUB had higher firing rates and responded to multiple locations. As has been observed in the rat, some of the SUB neurons encoded analogous spaces and were sensitive to the direction of travel.

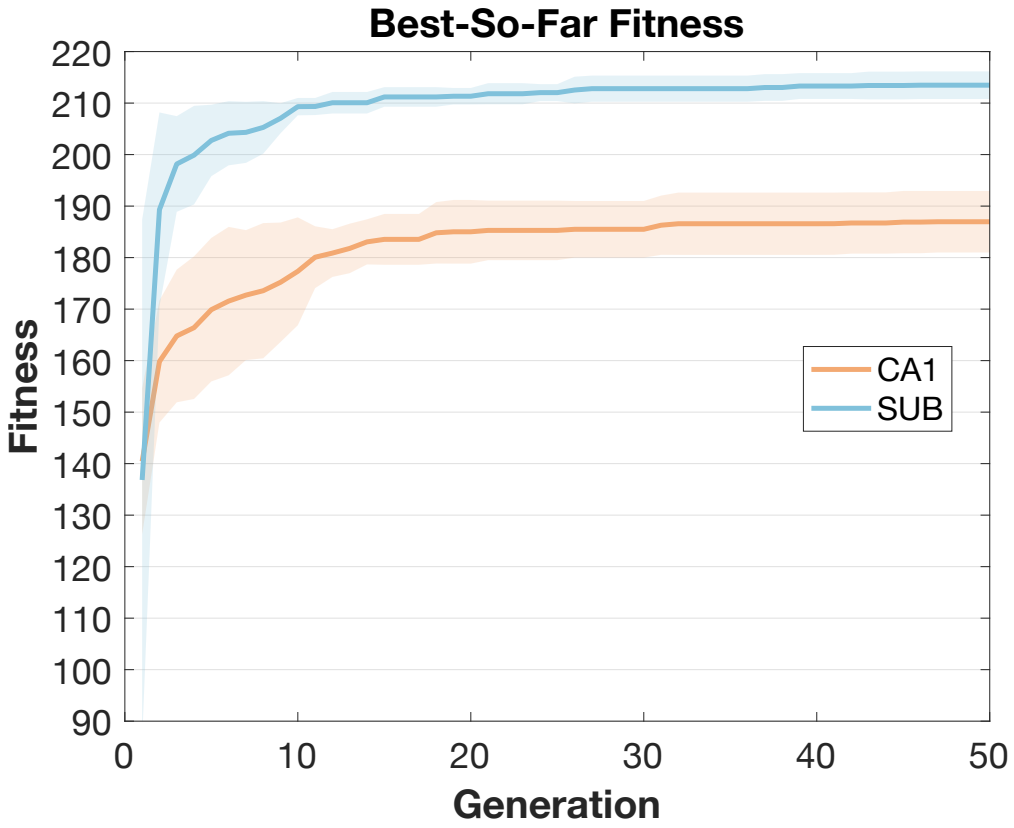


Figure 5.3: Best-so-far fitness scores over 50 generations for the CA1 and SUB models. Solid lines show the mean and shaded areas show the standard deviation of 5 runs. At generation 50, the CA1 model reached a mean fitness score of 186.97 and the SUB model had a mean score of 213.48, corresponding to a mean Pearson's ρ value of 0.63 and 0.56 respectively.

After training, the distribution of weights reflected the function of the brain region. Fig. 5.5 shows the histogram of connection weights from the input variables to the excitatory neuron groups. In the CA1 model (Fig. 5.5, top row), connection weights showed a U-shape distribution pattern. The weight values clustered at the limiting values (i.e. 0 and maximum weight), with more values near 0 for AV, LV, and HD, and more values near the maximum weight for Pos. This reflects the place encoding observed in this region. In contrast, a large proportion of connection weights in the SUB model (Fig. 5.5, bottom row) clustered near the maximum weight, with all four types of input variables showing a similar distribution. The responsiveness of SUB neurons, on average, to

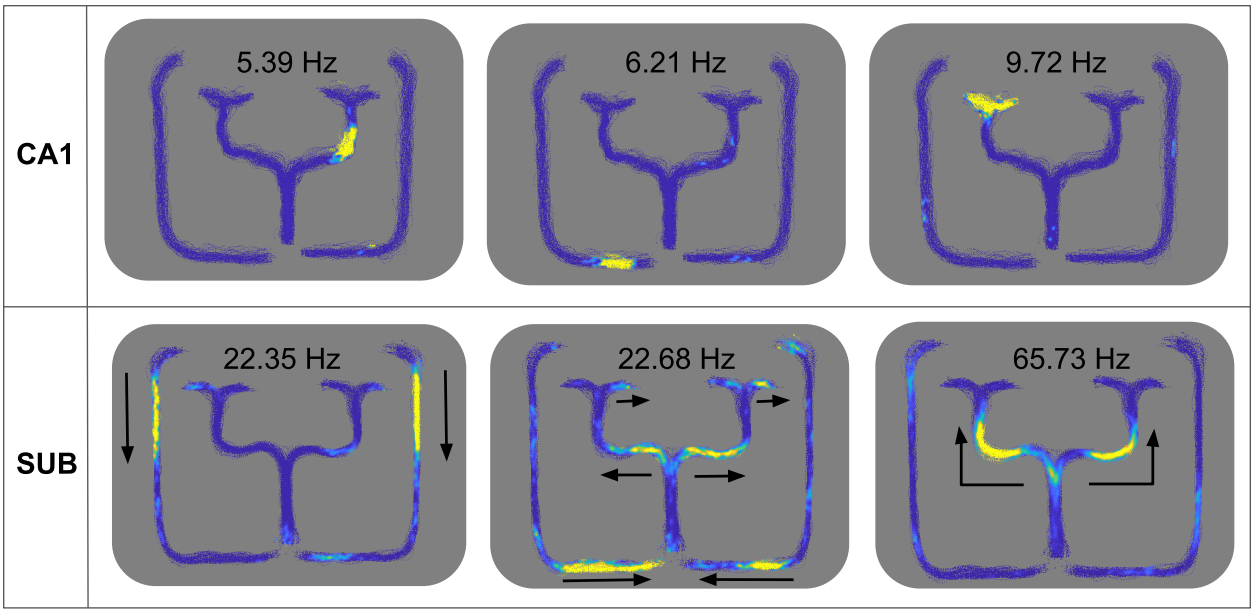


Figure 5.4: Examples of representative excitatory neurons in the CA1 and SUB models. Each firing rate map is labeled with the maximum firing rate of the neuron. Model units exhibit differential spatial representations: CA1 model units show single place fields, while the SUB model units respond to multiple locations that are analogous with respect to maze structures. The first and third SUB units show examples of analogy cells, and the second SUB unit shows an example of an axis-tuned cell. Arrows denote the direction of travel.

a broader set of positional, directional, and self-motion input types may be key to generation of firing fields in multiple locations that are analogous in terms of the direction of travel and location within topologically similar routes

Interestingly, the STDP parameters evolved to support these differential responses. Fig. 5.6 shows the evolved STDP curves for the two modeled brain regions. Compared to the SUB model, the CA1 model showed stronger long-term-depression (LTD) for E-STDP on both excitatory and inhibitory neurons, and weaker long-term-potentiation (LTP) for E-STDP on the inhibitory neurons. For the other evolved hyper-parameters, the maximum weight for the Inh $\rightarrow$ Exc connection in the CA1 model is stronger than that in the SUB model (CA1: 0.54 ± 0.05, SUB: 0.16 ± 0.11, Wilcoxon's rank sum test, $p < 0.01$). The CA1 model also showed a trend of having smaller values for the time scale parameter T (CA1: 0.78 ± 0.35 s, SUB: 4.5 ± 3.29 s, Wilcoxon's rank sum test, $p = 0.056$)

and the mean firing rate for the excitatory group (CA1: 2.28 ± 0.36 Hz, SUB: 5.19 ± 2.38 Hz, Wilcoxon's rank sum test, $p = 0.056$).

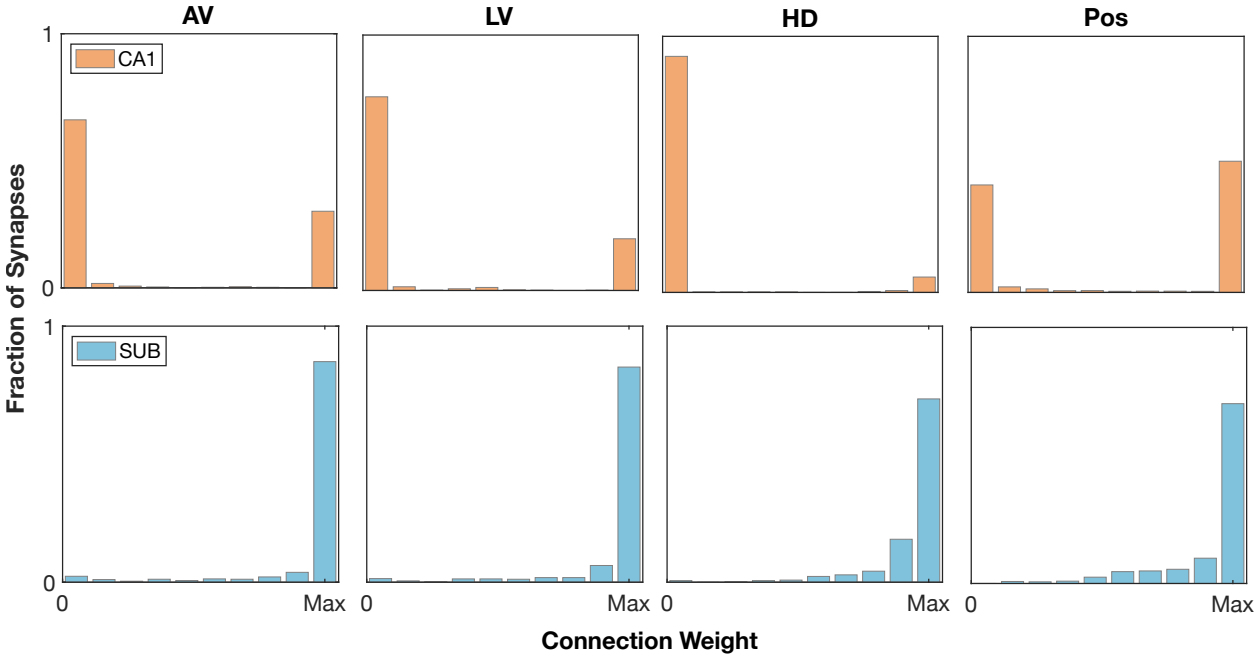


Figure 5.5: Histograms of the connection weights from the input groups to the excitatory neuron groups. Weight values range from 0 to the maximum weight value, which was evolved by the evolutionary algorithm. CA1 weights show more weights near 0 for AV, LV, and HD, and more weights near the maximum weight for Pos. In the SUB model, weight values of all four types of input all cluster near the maximum weight value.

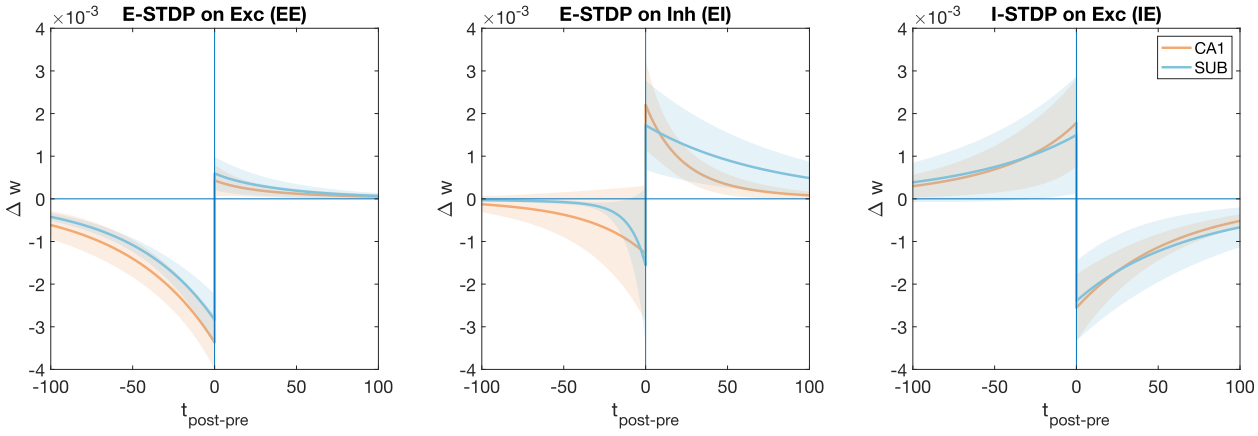


Figure 5.6: Visualization of STDP curves used in the evolved and trained CA1 and SUB models. Solid lines show the mean, and shaded areas show the standard deviation of 5 evolutionary runs. CA1 model had slightly stronger LTD on the EE and EI connections, and weaker LTP on the EI connections. The I-STDP curve in both models had similar amplitude and time constant.

To make a quantitative comparison between the spatial representations emerged from the two modeled regions, we conducted spatial analyses on the excitatory neurons of both modeled region (Table 5.1). These analyses were developed to interpret neural correlates of rodent navigation. We computed spatial information per spike (Skaggs et al., 1993), spatial sparsity (Skaggs et al., 1996), spatial selectivity (Skaggs et al., 1996), and spatial coherence (Kubie et al., 1990). As the excitatory neuron group was intended to model a larger population of neurons in the corresponding brain region, we analyzed neurons from the entire group instead of only those explicitly matched to the experimentally observed neurons. Consistent with the experimentally observed neurons and other neurophysiological studies (Sharp and Green, 1994), SUB neurons showed higher firing rates, lower spatial information per spike, lower spatial selectivity, and lower spatial coherence than CA1 neurons (Wilcoxon's rank sum test, $p < 0.01$).

Table 5.1: Spatial analyses on the model units (mean $\pm$ standard deviation) in the five evolutionary runs (sim) and the experimentally recorded neurons (recorded). Values in bold fonts showed greater values in comparison between CA1 and SUB neurons (Wilcoxon's rank sum test, $p < 0.01$).

SpatialMetrics	**CA1 (sim)**	**SUB (sim)**	**CA1 (recorded)**	**SUB (recorded)**
meanFR (Hz)	0.85 ± 0.87	$\mathbf{2.16 \pm 2.00}$	0.88 ± 1.42	$\mathbf{3.62 \pm 4.23}$
maxFR (Hz)	27.66 ± 18.78	$\mathbf{41.75 \pm 33.35}$	31.57 ± 14.91	$\mathbf{38.86 \pm 21.35}$
spatialIfo (bits)	$\mathbf{2.87 \pm 1.04}$	1.92 ± 0.69	$\mathbf{2.97 \pm 1.19}$	1.56 ± 1.21
sparsity	0.12 ± 0.08	$\mathbf{0.20 \pm 0.10}$	0.12 ± 0.13	$\mathbf{0.35 \pm 0.25}$
selectivity	$\mathbf{64.53 \pm 74.53}$	27.29 ± 21.47	$\mathbf{63.76 \pm 50.79}$	31.18 ± 38.84
spatialCoherence	$\mathbf{0.83 \pm 0.05}$	0.81 ± 0.05	0.48 ± 0.12	0.49 ± 0.14

5.3.2 Population vector analysis

We conducted population vector analyses with positional reconstruction matrices to test whether the spatial representations of these simulated brain regions are similar to the rodent. The population activity in both models closely resembled that of the experimentally observed neurons. Computed with Equation 5.7, CA1 model units obtained a similarity score of 0.76 ± 0.01 and SUB model units obtained a score of 0.69 ± 0.01 (i.e., mean $\pm$ standard deviation).

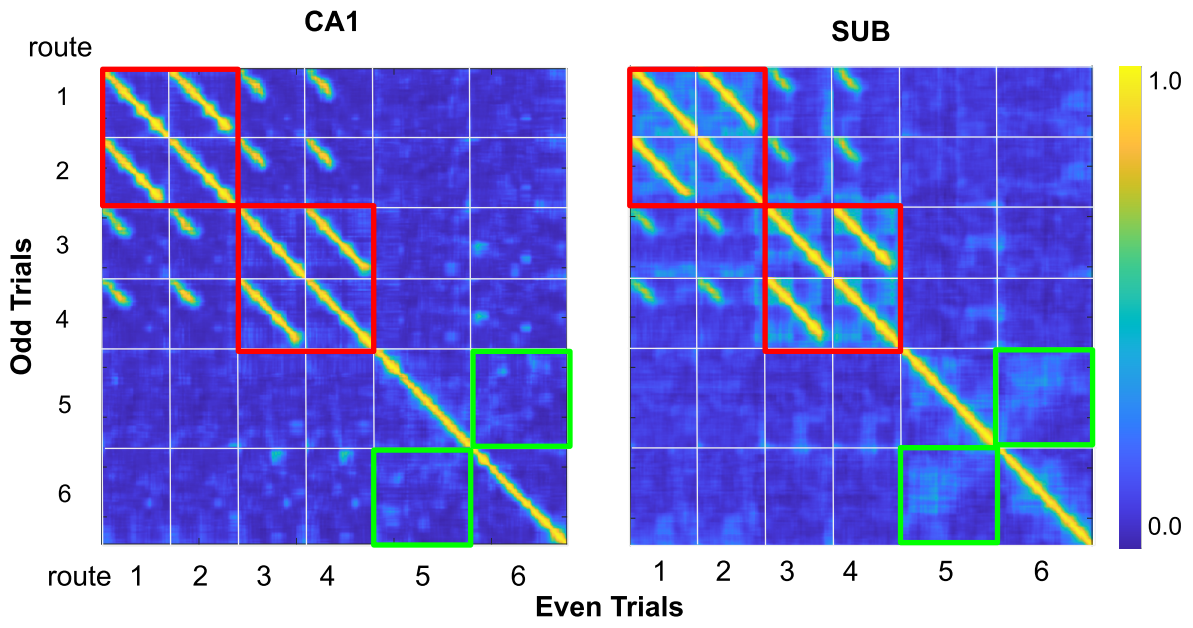


Figure 5.7: Positional reconstruction matrices of the simulated CA1 and SUB populations. Mean activity of the even trials are correlated against that of the odd trials. Position bins for each route are shown on the axes. At each position bin, the color represents the correlation value. High values on the diagonal lines indicate that position along the route is inferred from population activity of the neurons. Grids highlighted in red and green show stronger head-direction and analogous tuning of SUB ensemble compared to CA1.

Fig. 5.7 visualizes the positional reconstruction matrix for each model from a representative evolutionary run. Each value of the positional reconstruction matrix depicted the similarity of population activity in one location versus the activity in another location. Values on the diagonal line described the correlation of activity in the same location between odd and even trials. Both CA1 and SUB simulated neurons showed high correlation values on the diagonal line (median correlation value for CA1 averaged over 5 runs: 0.94 ± 0.01, for SUB: 0.96 ± 0.01), indicating that simulated neurons in both modeled regions reliably encoded locations. The two matrices also showed distinctive differences in off-diagonal values. Comparing the grids highlighted in red on both matrices, the SUB matrix had high correlation values around the diagonal in the grids while the CA1 matrix didn't show this pattern, indicating that the SUB model units had a stronger head-direction tuning. Additionally, for the grids highlighted in green, the SUB matrix showed a square region of higher correlation values near the top left of the grids, which correspond to a higher correlation of popu-

lation activity on the longer segments between Routes 5 and 6. These location pairs were spatially separated but shared the same head direction and analogous maze structure in the environment. These results indicate that the simulated SUB neurons had stronger head direction tuning and encoded analogous spaces more profoundly than the simulated CA1 neurons, which is consistent with the differences observed in the rodent CA1 and SUB (Olson et al., 2021).

5.3.3 Control experiments

To verify the necessity of evolutionary algorithms and STDP learning in the optimization process, we ran the models in two additional conditions; one in which there was STDP but no evolutionary algorithm, and another in which there was no STDP and no evolutionary algorithm. In both conditions, we had 5 runs for each model, with each run including 15 individual networks initialized with random hyper-parameters. Similar to the fully evolved and trained models, models in the control experiments were evaluated with a fitness function (Equation 5.4), and the best performing network individual in each run was selected for population vector analyses.

In both control conditions, the performance of the SNNs was worse than evolving STDP-H parameters for 50 generations. In the "no STDP no EA" condition, where each of the 15 networks was tested without being trained with STDP, the CA1 model obtained an average similarity score of 0.26 ± 0.36, and the SUB model obtained an average similarity score of 0.12 ± 0.28. In the "STDP no EA" condition, where each of the 15 networks went through the same training and testing procedure as the fully evolved models did but did not go through the evolutionary process, the CA1 model obtained an average similarity score of 0.46 ± 0.29 and the SUB model obtained an average similarity score of 0.60 ± 0.03.

Taken together, these results show that STDP greatly improved the performance of the network, and that having multiple generations of evolutionary computation was necessary for finding the hyper-parameters that allow for higher resemblance of simulated neuronal activity to that of the

modeled brain region. These control experiments suggest that parameter tuning through the evolutionary process and synaptic plasticity through STDP-H were necessary to replicate these brain regions. Similar results were reported when modeling the retrosplenial cortex (RSC) using this methodology (Rounds et al., 2018).

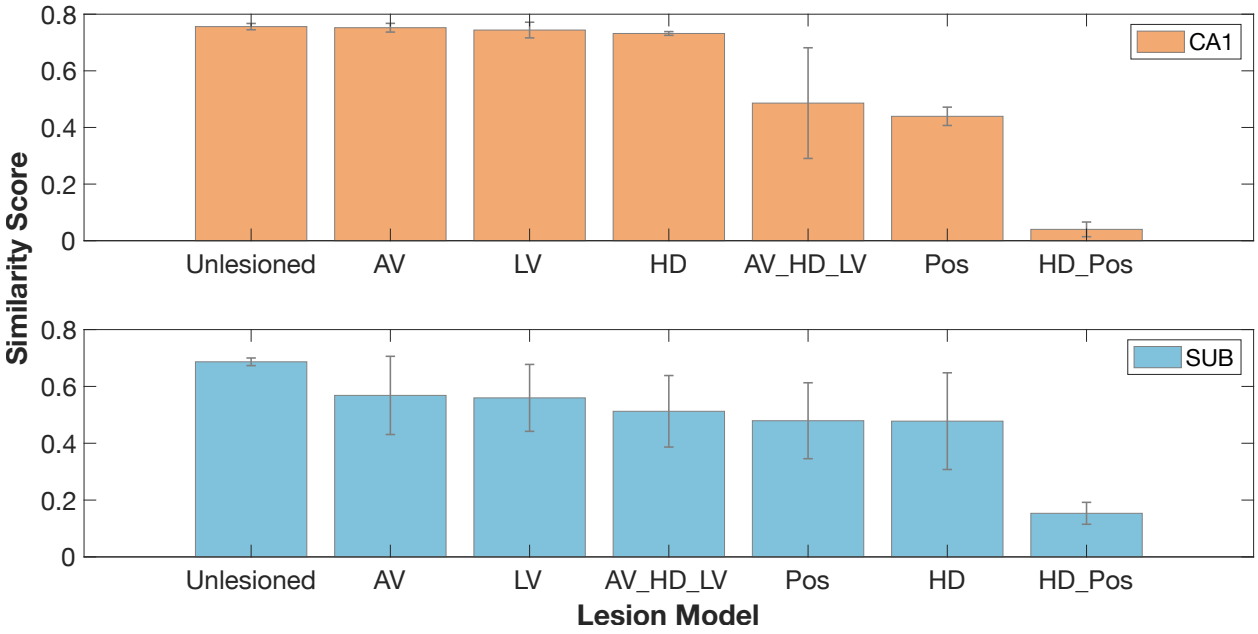


Figure 5.8: Ablation studies: similarity scores of the unlesioned and lesioned models. Bars show the mean values and the error bars show standard deviation of scores obtained by 5 instances of each model. Lesions of the positional input (Pos) in the CA1 model had the strongest impact on the network performance compared to lesions of other single input streams. In the SUB model, lesions of all four types of input had a similar level of impact on the network performance.

5.3.4 Ablation studies

To examine the effect of removing input streams on each modeled brain region, we conducted ablation studies using the fully evolved and trained networks. The ablation studies included lesions of connections from each of the input streams to both the excitatory and inhibitory neuron groups. Lesion models were created by loading the trained networks and removing the inter-group connections corresponding to the input stream(s). The lesion models were presented with the same input

variables as the non-lesioned models, and the network activity was recorded. Population vector analysis was then performed on the lesion models to assess the impact of lesions of input streams.

Lesions had differential effects on model performance that reflect the spatial encoding of CA1 and SUB (Fig. 5.8). In the CA1 model, lesions of the positional input (Pos) had a strong impact on the performance of the network, while lesions of one of the three idiothetic inputs (AV, HD, and LV) did not have a strong effect on performance. Lesions of all three idiothetic inputs together (AV_HD_LV) had a stronger impact than individual lesions, but were weaker than lesions of Pos alone. Additionally, lesions of the head direction and positional inputs together (HD_Pos) brought the similarity score down to near 0. In contrast, lesions of any input stream to the SUB had a moderate impact on network performance. Similar to the CA1 model, lesions of the head direction and positional inputs together (HD_Pos) in the SUB model greatly decreased the similarity score. Taken together, these ablation studies further support that the CA1 is more place specific and the SUB is more driven by inputs related to action or movement.

5.4 Discussion

The spiking neural network modeling framework presented here captured the differing spatial responses of hippocampal CA1 and the SUB through unsupervised learning, via STDP-H, and evolutionary algorithms. The resulting networks show highly place-specific responses in CA1 neurons and the emergence of pattern recurrence in the spatially specific firing of SUB neurons. These differing functional responses were reflected in the STDP-H parameters and the weight distributions of the simulated spiking neural networks (SNNs). Moreover, the present simulations make testable experimental predictions for the plasticity and connectivity in these brain areas.

The evolutionary algorithms automated the design of SNNs by indirect encoding of network learning parameters. This approach had been shown in a previous study to successfully replicate neural

dynamics observed in the retrosplenial cortex (RSC) as rodents traverse a W-shaped maze (Rounds et al., 2018). In the present work, we extended the approach to model two other regions that are important to spatial navigation: hippocampal CA1 and the SUB. Using data recorded in the two brain regions when rats performed the same navigational task in the same environment allowed us to compare our two models directly.

With the same input representations, the evolutionary algorithms selected different STDP-H learning parameters for the two SNN models, which led to different connection weight distributions. STDP-H parameters in the CA1 models had more LTD than in SUB, which led to a strong pruning effect in the CA1 excitatory group. In contrast, synapses connecting the AV, LV, and HD input groups with the SUB excitatory groups had larger connection weights than in the CA1 model, suggesting stronger vestibular inputs to the SUB model, which is consistent with neurophysiological observations (O'Mara, 2005; Allen and Hopkins, 1989).

Different distributions of connection weights in turn allowed for divergent spatial representations to emerge in the network models. These spatial representations, as analyzed through firing rate distributions, firing rate map visualizations, classic spatial metrics including spatial information, sparsity, selectivity, and spatial coherence, were consistent with those observed in the experimentally recorded neurons. The network models, though simplified in terms of the types of input information and network sizes, generated neural dynamics resembling those observed in the real neural circuits. It should be noted that although only a subset of simulated neurons were optimized to match with recorded neurons, all neurons in the simulated neuron groups were included in the analyses and showed consistent response properties within the group. This suggests that our modeling approach allowed for generalization of learned firing patterns to unobserved data.

Results of the control experiments underscore the importance of combining both STDP and evolutionary computation in the modeling framework. By evolving the learning parameters, the search space is dramatically reduced, as compared to directly evolving weights or using a method such as back-propagation. This study further suggests a functional role for STDP-H in stabilizing network

activity, developing spatial representationas, and reducing the dimensionality of the input space. The result is a viable SNN that can be used for a range of simulation studies. The success in modeling CA1, SUB and RSC (Rounds et al., 2018) suggests that the approach may be a general-purpose means to building SNNs.

In the ablation studies, we removed one or more input streams from the fully evolved and trained networks. Lesions of positional input to the CA1 model had a stronger impact on the network performance than the other three input streams, suggesting that spatial representations in CA1 are more reliant on the CA3 input to CA1 than self-motion signals. In the SUB model, lesions of any one of the four input streams had a similar level of impact on the network performance, suggesting that the SUB model utilized different input information more equally.

In addition to showing single place fields as CA1 neurons do, SUB neurons often encode multiple locations that share certain spatial features in a Triple-T maze environment(Olson et al., 2021). These representations may require an integration of idiothetic information as well as positional information. In both models, a significant drop in the similarity score was observed when two or more input streams were lesioned together, suggesting a conjunctive coding of multiple input variables in the CA1 and SUB models, which coincides with the evidence that CA1 and SUB neurons are encoding multiple types of signals (Muller et al., 1994; Kitanishi et al., 2020).

The presented modeling approach could be extended to include multiple brain regions to investigate how they interact. In future studies, we will integrate findings in this work and link together the CA1 and SUB models. As neurophysiological studies suggest, SUB receives a strong input from CA1, and SUB also sends backward projections to CA1 (O'Mara, 2005; Xu et al., 2016). Following these findings, we can investigate how the interaction between the two regions work together during navigation by connecting the two models and examining how information is integrated between these brain regions.

Acknowledgment

This material is based upon work supported by the United States Air Force Award #FA9550-19-1-0306. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Air Force.

Chapter 6

Summary and Future Directions

This dissertation aims to contribute to the understanding of how the brain processes information during navigation through the lens of computational modeling, specifically using spiking neural networks (SNNs) and evolutionary computation. The focus of this research centers on studying how information is encoded and processed in the cortical motion stream and the hippocampal area. To accomplish this, this study started by enhancing an open-source computational modeling tool, which was then applied to two modeling studies using a unique modeling approach that integrated evolving neural plasticity rules in SNNs. This approach produced models that account for response profiles of neurons related to visual motion perception and spatial cognition.

In Chapter 3, CARLsim 6 was introduced as a tool that allows for efficient construction and simulation of biologically realistic SNNs. The new release of this software extended beyond previous versions by introducing new neural plasticity features, while maintaining computational efficiency. In addition to their biological plausibility, SNNs also offer advantages in their event-driven nature, making them suitable for energy-saving neuromorphic implementations. Developing CARLsim to be integratable with neuromorphic hardware and enabling embedded systems could open up many opportunities for exploring embodiment of the brain and developing brain-based devices.

Chapter 4 presents the results from modeling MSTd neurons in an SNN model that was optimized through evolutionary algorithms and spike-timing-dependent plasticity (STDP). With the evolved STDP learning curves, the model learned to reconstruct the input optic flow patterns, which led to the emergence of MSTd-like receptive fields. Neurons in this model exhibited key response properties observed in MSTd neurons, such as spiral selectivity, 3D translational and rotational heading selectivity, and a population encoding of heading direction. This model demonstrated how synaptic plasticity can contribute to efficient encoding of the high-dimensional input space through sparse coding and dimensionality reduction. These findings suggest that STDP, a neurobiologically plausible learning rule, could potentially serve as a neural implementation of nonnegative sparse coding (NSC). A limitation of this study is the use of simplified and low-resolution stimuli that only contained coherent optic flow patterns that were projected to a frontoparallel plane. In natural scenes encountered during navigation, optic flows may contain segments that correspond to the sky and ground, which break the consistency in the optic flows patterns. It would be interesting to investigate whether the model can scale up to process naturalistic stimuli that are in higher resolution.

The presented model is biologically plausible and demonstrates a certain degree of resemblance to the macaque MSTd. It can serve as a valuable testbed for more complex visual motion processing scenarios, such as the processing of self-motion induced optic flows in the presence of moving objects and eye rotation. To further enhance its capabilities, depth information can be incorporated into the model, as this information was suggested to facilitate the estimate of self-motion and to separate object motion (Layton and Niehorster, 2019), and is also crucial for accurate heading perception in the presence of eye rotation (Manning and Britten, 2019).

This study focuses on the visual aspect of motion perception in MSTd. However, MST integrates multimodal information, including visual and vestibular signals (Gu et al., 2006; Takahashi et al., 2007; Britten, 2008). Incorporating vestibular signals into the model can help to understand how the brain separates self-motion and scene motion. As suggested in Rideaux et al. (2021), MSTd

single neurons are capable of performing causal inference when visual and vestibular signals provide conflicting information about self-motion. The same causal inference mechanism may also be employed to separate object motion from self-motion.

Chapter 5 presents SNN models of two neural structures, the hippocampal sub-region CA1 and the subiculum (SUB), which play a vital role in spatial memory. In this study, models of these two neural structures shared a common network architecture, and were optimized using the same approach. In the process of learning to reproduce the neural responses of CA1 and SUB in a working memory navigational task, differential spatial representations emerged in the two network models. Although only a subset of neurons in the models were optimized to match recorded neurons' activity profiles, the entire modeled neuron groups exhibited population statistics similar to those observed in CA1 and SUB recordings during behavior. This study suggests that that the place-specific responses of CA1 neurons and the response to "kinds of places" in SUB neurons may be attributed to different integration of a similar set of navigational variables.

The optimized models presented in Chapter 5 can be extended to help better understand the functional properties of neurons in the CA1 and SUB regions during navigational tasks. By incorporating a readout layer or a set of action units, these models can be trained to perform path integration tasks or goal-directed navigation tasks. The resulting models can then be compared with task-optimized models that are trained end-to-end on specific navigational tasks. Previous studies have demonstrated that networks trained to perform path integration exhibit grid-like, head-direction selective, and place-specific responses (Banino et al., 2018; Cueva and Wei, 2018; Cueva et al., 2020). By comparing the response properties, task performance, and generalizability of task-optimized models and models optimized to capture neural responses, we can gain further insights into the functional distinctions of these two brain areas and how the observed spatial representations may be used in navigation.

The hippocampal models developed in this study can be utilized for virtual experiments that may be difficult or expensive to conduct in real-world environments. One example of such experiments

is altering the shape of the maze to investigate how neuronal responses adapt as a result. Neurophysiological studies have demonstrated that environmental features can influence the response properties of neurons. For instance, Olson et al. (2017) observed axis-tuned cells, which are selective to two head directions 180° apart, only in the Triple-T maze task and not in open space foraging. Additionally, Zhang et al. (1993) suggested that environment symmetry shapes directional responses in neurons located in the retrosplenial cortex (RSC), indicating an experience-dependent effect of global environment symmetry on local firing symmetry. By testing the models in different environment, we can observe how the brain encodes environmental structure in a way that is relevant to the current perceptual situation, as well as a more global view of the environment beyond immediate perceptual reach. In addition, lesion studies, such as selective removal of certain connections or inactivation of specific regions, can be conducted to investigate how different components of the network contribute to network responses.

In this dissertation, two modeling studies were presented, both using the same modeling approach of evolving STDP parameters in SNNs, but with different optimization goals. The MSTd model was optimized to reconstruct stimuli by assuming that MSTd neurons act as basis functions, a linear combination of which encodes the stimulus space. In contrast, the CA1 and SUB models were optimized by matching a subset of neurons to the exact activity profile of neurons recorded in the target region. During model training, MSTd-like receptive fields emerged in the first model, and neurons in the CA1/ SUB model exhibited response properties of CA1 or SUB neurons, even for neurons that were not matched directly to recorded neurons. These results demonstrate the power and versatility of the modeling approach described in this dissertation. To extend this modeling approach, we may impose fewer assumptions about weight update mechanisms by using a more general design of the plasticity rules in the network, such as the approach presented in Jordan et al. (2021), which involves defining learning rules as modular mathematical expressions, with specific update mechanisms discovered through evolving the network to complete certain tasks. In addition to evolving the learning rules, we can also combine STDP with evolving the network architecture of the model, such as the number of neurons and their connections (Elbrecht et al.,

2020). Being agnostic about the formation of the network provides an opportunity to observe how network topology and its relationship with learning rules can optimize performance.

Another potential research direction is to integrate optic flow processing and spatial cognition into a single model to better understand the interaction between motion perception and the development of cognitive maps during navigation. Evidence suggests that optic flow can help individuals build mental representations of the environment, allowing for navigation along complex, multi-legged paths (Kirschen et al., 2000). Path integration, which involves calculating the momentary estimation of current position based on self-movement cues, has been extensively studied in rodents and primates. Human path integration can be performed by solely integrating optic flow information (Kearns et al., 2002). The relationship between optic flow processing and spatial representations has also been investigated in rodent studies, which demonstrated that optic flow enables systematic control of place fields through the recalibration of path integration, thus preventing or correcting accumulating errors (Madhav et al., 2022).

In addition to the perception of heading due to optic flow, neurons in MSTd were also shown to integrate heading and location information to encode the path and place of self-movement, potentially serving spatial cognition (Froehler and Duffy, 2002). Furthermore, Alefantis et al. (2022) demonstrated that humans and monkeys can use optic flow for sensory evidence accumulation in a naturalistic navigation task. By combining optic flow processing and spatial cognition in a single model, we can investigate how the brain encodes spatial features that allow for effective navigation using optic flow information for short-term reactive control, as well as using cognitive maps for long-term planning.

Overall, this dissertation demonstrates the potential of SNNs and evolutionary computation in modeling the complex neural computations involved in processing information related to navigation, particularly the processing of visual information and representation of spatial features. The findings of this research provide insights into the neural mechanisms underlying navigation, generate testable predictions regarding the integration and processing of information in the brain, and

may inform the development of novel approaches for modeling the brain. In addition to increasing our understanding of the brain, the models developed in this research also have the potential for practical applications in areas such as robotics and artificial intelligence.

Bibliography

L. F. Abbott and S. B. Nelson. Synaptic plasticity: Taming the beast. *Nature Neuroscience*, 3(11): 1178–1183, Nov. 2000. ISSN 1546-1726. doi: 10.1038/81453.

E. H. Adelson and J. R. Bergen. Spatiotemporal energy models for the perception of motion. *Journal of the Optical Society of America A: Optics, Image Science, and Vision*, 2, 1985.

E. H. Adelson and J. A. Movshon. Phenomenal coherence of moving visual patterns. *Nature*, 300 (5892):523–525, 1982. ISSN 00280836. doi: 10.1038/300523a0.

T. D. Albright. Direction and orientation selectivity of neurons in visual area MT of the macaque. *Journal of Neurophysiology*, 52(6):1106–1130, Dec. 1984. ISSN 0022-3077. doi: 10.1152/jn.1984.52.6.1106.

P. Alefantis, K. Lakshminarasimhan, E. Avila, J.-P. Noel, X. Pitkow, and D. E. Angelaki. Sensory Evidence Accumulation Using Optic Flow in a Naturalistic Navigation Task. *The Journal of Neuroscience*, 42(27):5451–5462, July 2022. ISSN 0270-6474, 1529-2401. doi: 10.1523/JNEUROSCI.2203-21.2022.

G. V. Allen and D. A. Hopkins. Mamillary body in the rat: Topography and synaptology of projections from the subicular complex, prefrontal cortex, and midbrain tegmentum. *J. Comp. Neurol.*, 286(3):311–336, 1989. ISSN 10969861. doi: 10.1002/cne.902860303.

I. Altintas, K. Marcus, I. Nealey, S. L. Sellars, J. Graham, D. Mishin, J. Polizzi, D. Crawl, T. DeFanti, and L. Smarr. Workflow-Driven Distributed Machine Learning in CHASE-CI: A Cognitive Hardware and Software Ecosystem Community Infrastructure. *arXiv:1903.06802 [cs]*, Feb. 2019.

R. A. Andersen, L. H. Snyder, D. C. Bradley, and J. Xing. Multimodal representation of space in the posterior parietal cortex and its use in planning movements. *Annual Review of Neuroscience*, 20(1):303–330, Mar. 1997. ISSN 0147-006X, 1545-4126. doi: 10.1146/annurev.neuro.20.1.303.

M. A. Arbib, editor. *The Handbook of Brain Theory and Neural Networks*. A Bradford Book, Cambridge, MA, USA, June 1995. ISBN 978-0-262-01148-8.

A. Arleo and W. Gerstner. Spatial cognition and neuro-mimetic navigation: A model of hippocampal place cell activity. *Biological Cybernetics*, 83(3):287–299, Aug. 2000. ISSN 0340-1200, 1432-0770. doi: 10.1007/s004220000171.

M. Avery, N. Dutt, and J. Krichmar. A large-scale neural network model of the influence of neuromodulatory levels on working memory and behavior. *Frontiers in Computational Neuroscience*, 7, 2013. ISSN 1662-5188. doi: 10.3389/fncom.2013.00133.

M. C. Avery and J. L. Krichmar. Improper activation of D1 and D2 receptors leads to excess noise in prefrontal cortex. *Frontiers in Computational Neuroscience*, 9, 2015. ISSN 1662-5188. doi: 10.3389/fncom.2015.00031.

M. C. Avery and J. L. Krichmar. Neuromodulatory systems and their interactions: A review of models, theories, and experiments. *Frontiers in Neural Circuits*, 11, 2017. ISSN 1662-5110. doi: 10.3389/fncir.2017.00108.

A. Balaji, P. Adiraju, H. J. Kashyap, A. Das, J. L. Krichmar, N. D. Dutt, and F. Catthoor. PyCARL: A PyNN Interface for Hardware-Software Co-Simulation of Spiking Neural Network. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–10, 2020. doi: 10.1109/IJCNN48605.2020.9207142.

A. Banino, C. Barry, B. Uria, C. Blundell, T. Lillicrap, P. Mirowski, A. Pritzel, M. J. Chadwick, T. Degris, J. Modayil, G. Wayne, H. Soyer, F. Viola, B. Zhang, R. Goroshin, N. Rabinowitz, R. Pascanu, C. Beattie, S. Petersen, A. Sadik, S. Gaffney, H. King, K. Kavukcuoglu, D. Hassabis, R. Hadsell, and D. Kumaran. Vector-based navigation using grid-like representations in artificial agents. *Nature*, 557(7705):429–433, May 2018. ISSN 14764687. doi: 10.1038/s41586-018-0102-6.

C. A. Barnes, B. L. McNaughton, S. J. Mizumori, B. W. Leonard, and L. H. Lin. Comparison of spatial and temporal characteristics of neuronal activity in sequential stages of hippocampal processing. *Progress in Brain Research*, 83:287–300, 1990. ISSN 0079-6123. doi: 10.1016/s0079-6123(08)61257-1.

S. A. Beardsley and L. M. Vaina. Computational modelling of optic flow selectivity in MSTd neurons. *Network: Computation in Neural Systems*, 9(4):467–493, 1998. ISSN 0954898X. doi: 10.1088/0954-898X_9_4_005.

T. E. J. Behrens, T. H. Muller, J. C. R. Whittington, S. Mark, A. B. Baram, K. L. Stachenfeld, and Z. Kurth-Nelson. What Is a Cognitive Map? Organizing Knowledge for Flexible Behavior. *Neuron*, 2018. doi: 10.1016/j.neuron.2018.10.002.

T. Bekolay, J. Bergstra, E. Hunsberger, T. DeWolf, T. Stewart, D. Rasmussen, X. Choo, A. Voelker, and C. Eliasmith. Nengo: A Python tool for building large-scale functional brain models. *Frontiers in Neuroinformatics*, 7, 2014. ISSN 1662-5196.

M. Beyeler, K. D. Carlson, T. S. Chou, N. Dutt, and J. L. Krichmar. CARLsim 3: A user-friendly and highly optimized library for the creation of neurobiologically detailed spiking neural networks. *Proceedings of the International Joint Conference on Neural Networks*, 2015-Septe, 2015. doi: 10.1109/IJCNN.2015.7280424.

M. Beyeler, N. Dutt, and J. L. Krichmar. 3D Visual Response Properties of MSTd Emerge from an Efficient, Sparse Population Code. *Journal of Neuroscience*, 2016. ISSN 0270-6474. doi: 10.1523/JNEUROSCI.0396-16.2016.

M. Beyeler, E. L. Rounds, K. D. Carlson, N. Dutt, and J. L. Krichmar. Neural correlates of sparse coding and dimensionality reduction. *PLoS computational biology*, 15(6):e1006908–e1006908, June 2019. doi: 10.1371/journal.pcbi.1006908.

T. Blickle and L. Thiele. A Comparison of Selection Schemes Used in Evolutionary Algorithms. *Evolutionary Computation*, 4(4):361–394, Dec. 1996. ISSN 1063-6560. doi: 10.1162/evco.1996.4.4.361.

A. Borst. Models of motion detection. *Nature Neuroscience*, 3(11):1168–1168, Nov. 2000. ISSN 1546-1726. doi: 10.1038/81435.

D. Boussaoud, L. G. Ungerleider, and R. Desimone. Pathways for motion analysis: Cortical connections of the medial superior temporal and fundus of the superior temporal visual areas in the macaque. *The Journal of Comparative Neurology*, 296(3):462–495, June 1990. ISSN 0021-9967. doi: 10.1002/cne.902960311.

K. H. Britten. Mechanisms of Self-Motion Perception. *Annual Review of Neuroscience*, 31(1):389–410, July 2008. doi: 10.1146/annurev.neuro.29.051605.112953.

K. H. Britten and R. J. Van Wezel. Electrical microstimulation of cortical area MST biases heading perception in monkeys. *Nature Neuroscience*, 1(1):59–63, May 1998. ISSN 10976256. doi: 10.1038/259.

K. H. Britten and R. J. A. Van Wezel. Area MST and heading perception in macaque monkeys. *Cerebral Cortex (New York, N.Y.: 1991)*, 12(7):692–701, July 2002. ISSN 1047-3211. doi: 10.1093/cercor/12.7.692.

K. H. Britten, W. T. Newsome, M. N. Shadlen, S. Celebrini, and J. A. Movshon. A relationship between behavioral choice and the visual responses of neurons in macaque MT. *Visual Neuroscience*, 13(1):87–100, Jan. 1996. ISSN 0952-5238. doi: 10.1017/S095252380000715X.

D. Bucher and E. Marder. SnapShot: Neuromodulation. *Cell*, 155(2):482–482.e1, Oct. 2013.

N. Burgess and J. O'Keefe. Models of place and grid cell firing and theta rhythmicity. *Current Opinion in Neurobiology*, 21(5):734–744, Oct. 2011. ISSN 1873-6882. doi: 10.1016/j.conb.2011.07.002.

N. Burgess, M. Recce, and J. O'Keefe. A model of hippocampal function. *Neural Networks*, 7(6):1065–1081, Jan. 1994. ISSN 0893-6080. doi: 10.1016/S0893-6080(05)80159-5.

N. Caporale and Y. Dan. Spike Timing–Dependent Plasticity: A Hebbian Learning Rule. *Annual Review of Neuroscience*, 31(1):25–46, July 2008. ISSN 0147-006X. doi: 10.1146/annurev.neuro.31.060407.125639.

M. Carandini. Area V1. *Scholarpedia*, 7(7):12105, July 2012. ISSN 1941-6016. doi: 10.4249/scholarpedia.12105.

M. Carandini, D. J. Heeger, and J. Anthony Movshon. Linearity and Gain Control in V1 Simple Cells. In P. S. Ulinski, E. G. Jones, and A. Peters, editors, *Models of Cortical Circuits*, Cerebral Cortex, pages 401–443. Springer US, Boston, MA, 1999. ISBN 978-1-4615-4903-1. doi: 10.1007/978-1-4615-4903-1_7.

K. D. Carlson, M. Richert, N. Dutt, and J. L. Krichmar. Biologically plausible models of homeostasis and STDP: Stability and learning in spiking neural networks. *Proceedings of the International Joint Conference on Neural Networks*, pages 1824–1831, 2013. ISSN 9781467361293. doi: 10.1109/IJCNN.2013.6706961.

K. D. Carlson, J. M. Nageswaran, N. Dutt, and J. L. Krichmar. An efficient automated parameter tuning framework for spiking neural networks. *Frontiers in Neuroscience*, 8(8 FEB):1–15, 2014. ISSN 1662453X. doi: 10.3389/fnins.2014.00010.

N. T. Carnevale and M. L. Hines. *The NEURON Book*. Cambridge University Press, Cambridge, 2006. ISBN 978-0-521-84321-8. doi: 10.1017/CBO9780511541612.

J. R. Cavanaugh, W. Bair, and J. Anthony Movshon. Nature and interaction of signals from the receptive field center and surround in macaque V1 neurons. *Journal of Neurophysiology*, 88(5): 2530–2546, Nov. 2002. doi: 10.1152/jn.00692.2001.

S. Celebrini and W. T. Newsome. Microstimulation of extrastriate area MST influences performance on a direction discrimination task. *Journal of Neurophysiology*, 73(2):437–448, 1995. doi: 10.1152/jn.1995.73.2.437.

K. Chen, A. Johnson, E. O. Scott, X. Zou, K. A. De Jong, D. A. Nitz, and J. L. Krichmar. Differential Spatial Representations in Hippocampal CA1 and Subiculum Emerge in Evolved Spiking Neural Networks. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, July 2021. doi: 10.1109/IJCNN52387.2021.9534139.

K. Chen, M. Beyeler, and J. L. Krichmar. Cortical Motion Perception Emerges from Dimensionality Reduction with Evolved Spike-Timing-Dependent Plasticity Rules. *Journal of Neuroscience*, 42(30):5882–5898, July 2022. ISSN 0270-6474, 1529-2401. doi: 10.1523/JNEUROSCI.0384-22.2022.

M. Chistiakova, N. M. Bannon, J.-Y. Chen, M. Bazhenov, and M. Volgushev. Homeostatic role of heterosynaptic plasticity: Models and experiments. *Frontiers in Computational Neuroscience*, 9, July 2015. doi: 10.3389/fncom.2015.00089.

T.-S. Chou, L. Bucci, and J. Krichmar. Learning touch preferences with a tactile robot using dopamine modulated STDP in a model of insular cortex. *Frontiers in Neurorobotics*, 9, 2015. ISSN 1662-5218. doi: 10.3389/fnbot.2015.00006.

T. S. Chou, H. J. Kashyap, J. Xing, S. Listopad, E. L. Rounds, M. Beyeler, N. Dutt, and J. L. Krichmar. CARLsim 4: An Open Source Library for Large Scale, Biologically Detailed Spiking Neural Network Simulation using Heterogeneous Clusters. *Proc. Int. Jt. Conf. Neural Networks*, 2018-July:1158–1165, 2018. ISSN 1932-7447. doi: 10.1109/IJCNN.2018.8489326.

M. A. Coletti, E. O. Scott, and J. K. Bassett. Library for Evolutionary Algorithms in Python (LEAP). In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion*, GECCO '20, pages 1571–1579, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 978-1-4503-7127-8. doi: 10.1145/3377929.3398147.

S. L. Cowen and D. A. Nitz. Repeating firing fields of CA1 neurons shift forward in response to increasing angular velocity. *J. Neurosci.*, 34(1):232–241, 2014. ISSN 02706474. doi: 10.1523/JNEUROSCI.1199-13.2014.

C. J. Cueva and X.-X. Wei. Emergence of grid-like representations by training recurrent neural networks to perform spatial localization. In *International Conference on Learning Representations (ICLR) 2018*, 2018.

C. J. Cueva, P. Y. Wang, M. Chin, and X.-X. Wei. Emergence of functional and structural properties of the head direction system by optimization of recurrent neural networks. In *International Conference on Learning Representations (ICLR) 2020*, 2020.

K. E. Cullen. Vestibular processing during natural self-motion: Implications for perception and action. *Nature Reviews Neuroscience*, 20(6):346–363, June 2019. ISSN 1471-003X, 1471-0048. doi: 10.1038/s41583-019-0153-1.

P. Dayan. Improving Generalization for Temporal Difference Learning: The Successor Representation. *Neural Computation*, 5(4):613–624, July 1993. ISSN 0899-7667. doi: 10.1162/neco.1993.5.4.613.

K. De Jong. *Evolutionary Computation: A Unified Approach*. MIT Press, 2006.

R. de Schepper, J. M. Eppler, A. Kurth, P. Nagendra Babu, R. Deepu, S. Spreizer, G. Trensch, J. Pronold, S. B. Vennemo, S. Graber, A. Morales-Gregorio, C. Linssen, M. A. Benelhedi, H. Mørk, A. Morrison, D. Terhorst, J. Mitchell, S. Diaz, I. Kitayama, M. Enan, N. L. Kamiji, and H. E. Plesser. NEST 3.2. Zenodo, Jan. 2022.

P. Diehl and M. Cook. Unsupervised learning of digit recognition using spike-timing-dependent plasticity. *Frontiers in Computational Neuroscience*, 9:99, 2015. ISSN 1662-5188. doi: 10.3389/fncom.2015.00099.

C. J. Duffy and R. H. Wurtz. Sensitivity of MST Neurons to Optic Flow Stimuli. I. A Continuum of Response Selectivity to Large-Field Stimuli. *The journal of Neurophysiology*, 65, 1991.

S. Dutta, V. Kumar, A. Shukla, N. R. Mohapatra, and U. Ganguly. Leaky Integrate and Fire Neuron by Charge-Discharge Dynamics in Floating-Body MOSFET. *Scientific Reports*, 7(1):8257, Aug. 2017. ISSN 2045-2322. doi: 10.1038/s41598-017-07418-y.

A. Eiben and J. Smith. *Introduction to Evolutionary Computing*. Natural Computing Series. Springer, Berlin, Heidelberg, 2015. ISBN 978-3-662-44873-1 978-3-662-44874-8. doi: 10.1007/978-3-662-44874-8.

D. Elbrecht, M. Parsa, S. R. Kulkarni, J. P. Mitchell, and C. D. Schuman. Training Spiking Neural Networks Using Combined Learning Approaches. In *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1995–2001, 2020. doi: 10.1109/SSCI47803.2020.9308443.

R. C. Emerson, J. R. Bergen, and E. H. Adelson. Directionally selective complex cells and the computation of motion energy in cat visual cortex. *Vision research*, 32(2):203–18, Feb. 1992. ISSN 0042-6989. doi: 10.1016/0042-6989(92)90130-b.

Y. Fang and J. Li. A review of tournament selection in genetic programming. In Z. Cai, C. Hu, Z. Kang, and Y. Liu, editors, *Advances in Computation and Intelligence*, pages 181–192, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg. ISBN 978-3-642-16493-4.

D. E. Feldman. Timing-based LTP and LTD at vertical inputs to layer II/III pyramidal cells in rat barrel cortex. *Neuron*, 27(1):45–56, July 2000. doi: 10.1016/s0896-6273(00)00008-8.

D. E. Feldman. The Spike-Timing Dependence of Plasticity. *Neuron*, 75(4):556–571, Aug. 2012. doi: 10.1016/j.neuron.2012.08.001.

D. J. Felleman and D. C. Van Essen. Distributed hierarchical processing in the primate cerebral cortex. *Cerebral Cortex (New York, N.Y.: 1991)*, 1(1):1–47, 1991 Jan-Feb. ISSN 1047-3211. doi: 10.1093/cercor/1.1.1-a.

J. Ferbinteanu and M. L. Shapiro. Prospective and retrospective memory coding in the hippocampus. *Neuron*, 40(6):1227–1239, Dec. 2003. doi: 10.1016/S0896-6273(03)00752-9.

L. A. Finelli, S. Haney, M. Bazhenov, M. Stopfer, and T. J. Sejnowski. Synaptic learning rules and sparse coding in a model sensory system. *PLoS Computational Biology*, 4(4):e1000062, Apr. 2008. doi: 10.1371/journal.pcbi.1000062.

P. Foldiak and D. Endres. Sparse coding. *Scholarpedia*, 3(1):2984, Jan. 2008. ISSN 1941-6016. doi: 10.4249/scholarpedia.2984.

D. J. Foster, R. G. M. Morris, and P. Dayan. A model of hippocampally dependent navigation, using the temporal difference learning rule. *Hippocampus*, 10(1):1–16, 2000. ISSN 10509631. doi: 10.1002/(SICI)1098-1063(2000)10:1<1::AID-HIPO1>3.0.CO;2-1.

N. Frémaux, H. Sprekeler, and W. Gerstner. Reinforcement Learning Using a Continuous Time Actor-Critic Framework with Spiking Neurons. *PLoS Computational Biology*, 9(4), 2013. doi: 10.1371/journal.pcbi.1003024.

M. T. Froehler and C. J. Duffy. Cortical neurons encoding path and place: Where you go is where you are. *Science*, 295(5564):2462–2465, Mar. 2002. ISSN 00368075. doi: 10.1126/science.1067426.

B. E. Frost, S. K. Martin, M. Cafalchio, M. N. Islam, J. P. Aggleton, and S. M. O'Mara. Anterior thalamic function is required for spatial coding in the subiculum and is necessary for spatial memory, 2020.

A. P. Georgopoulos, J. F. Kalaska, C. Roberto, and J. T. Massey. On the relations between the direction of two-dimensional arm movements and cell discharge in primate motor cortex. *The Journal of Neuroscience*, 2(11), 1982.

W. Gerstner, R. Kempter, J. L. van Hemmen, and H. Wagner. A developmental learning rule for coincidence tuning in the barn owl auditory system. In J. M. Bower, editor, *Computational Neuroscience: Trends in Research, 1997*, pages 665–669. Springer US, Boston, MA, 1997. ISBN 978-1-4757-9800-5. doi: 10.1007/978-1-4757-9800-5_103.

B. Golosio, G. Tiddia, C. De Luca, E. Pastorelli, F. Simula, and P. S. Paolucci. Fast simulations of highly-connected spiking cortical models using GPUs. *Frontiers in Computational Neuroscience*, 15, 2021. ISSN 1662-5188. doi: 10.3389/fncom.2021.627620.

D. Graham and D. Field. Sparse Coding in the Neocortex. In *Evolution of Nervous Systems*, pages 181–187. Elsevier, 2007. ISBN 978-0-12-370878-6. doi: 10.1016/B0-12-370878-8/00064-1.

M. S. A. Graziano, R. A. Andersen, and R. J. Snowden. Tuning of MST Neurons to Spiral Motions. *The Journal of neuroscience*, 1994.

S. Grossberg, E. Mingolla, and C. Pack. A Neural Model of Motion Processing and Visual Navigation by Cortical Area MST. *Cerebral Cortex*, 9(8):878–895, Dec. 1999. ISSN 1047-3211. doi: 10.1093/cercor/9.8.878.

Y. Gu, P. V. Watkins, D. E. Angelaki, and G. C. Deangelis. Visual and Nonvisual Contributions to Three-Dimensional Heading Selectivity in the Medial Superior Temporal Area. *The Journal of neuroscience*, 2006. doi: 10.1523/JNEUROSCI.2356-05.2006.

Y. Gu, D. E. Angelaki, and G. C. Deangelis. Neural correlates of multisensory cue integration in macaque MSTd. *NATURE NEUROSCIENCE VOLUME*, 11, 2008. doi: 10.1038/nn.2191.

Y. Gu, C. R. Fetsch, B. Adeyemo, G. C. Deangelis, and D. E. Angelaki. Decoding of MSTd Population Activity Accounts for Variations in the Precision of Heading Perception. *Neuron*, 66:596–609, 2010. doi: 10.1016/j.neuron.2010.04.026.

Y. Gu, G. C. Deangelis, and D. E. Angelaki. Causal links between dorsal medial superior temporal area neurons and multisensory heading perception. *The Journal of neuroscience : the official journal of the Society for Neuroscience*, 32(7):2299–313, Feb. 2012. ISSN 1529-2401. doi: 10.1523/JNEUROSCI.5154-11.2012.

T. Hafting, M. Fyhn, S. Molden, M. B. Moser, and E. I. Moser. Microstructure of a spatial map in the entorhinal cortex. *Nature*, 436(7052):801–806, Aug. 2005. doi: 10.1038/nature03721.

S. B. Hamed, W. Page, C. Duffy, and A. Pouget. MSTd Neuronal Basis Functions for the Population Encoding of Heading Direction. *J Neurophysiol*, 90:549–558, 2003. doi: 10.1152/jn.00639.2002.

T. Hartley, N. Burgess, C. Lever, F. Cacucci, and J. O'Keefe. Modeling place fields in terms of the cortical inputs to the hippocampus. *Hippocampus*, 10(4):369–379, 2000. ISSN 1098-1063. doi: 10.1002/1098-1063(2000)10:4<369::AID-HIPO3>3.0.CO;2-0.

D. O. Hebb. The organization of behavior; a neuropsychological theory. *Wiley*, 1949.

N. Heess and W. Bair. Direction Opponency, Not Quadrature, Is Key to the 1/4 Cycle Preference for Apparent Motion in the Motion Energy Model. *Journal of Neuroscience*, 30(34):11300–11304, Aug. 2010. ISSN 0270-6474, 1529-2401. doi: 10.1523/JNEUROSCI.1271-10.2010.

A. L. Hodgkin and A. F. Huxley. A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of Physiology*, 117(4):500–544, 1952. ISSN 1469-7793. doi: 10.1113/jphysiol.1952.sp004764.

B. K. P. Horn and B. G. Schunck. Determining optical flow. *Artificial Intelligence*, 17(1):185–203, Aug. 1981. ISSN 0004-3702. doi: 10.1016/0004-3702(81)90024-2.

Ø. A. Høydal, E. R. Skytøen, S. O. Andersson, M.-B. Moser, and E. I. Moser. Object-vector coding in the medial entorhinal cortex. *Nature*, 568(7752):400–404, Apr. 2019. ISSN 1476-4687. doi: 10.1038/s41586-019-1077-7.

P. O. Hoyer. Non-negative sparse coding. *Proceedings of the 12th IEEE Workshop on Neural Networks for Signal Processing*, 2002. doi: 10.1109/nnsp.2002.1030067.

P. O. Hoyer. Non-negative Matrix Factorization with Sparseness Constraints. *Journal of Machine Learning Research*, 5(Nov):1457–1469, 2004. ISSN ISSN 1533-7928.

D. H. Hubel and T. N. Wiesel. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of Physiology*, 160(1):106–154, Jan. 1962. ISSN 14697793. doi: 10.1113/jphysiol.1962.sp006837.

E. B. Issa, C. F. Cadieu, and J. J. DiCarlo. Neural dynamics at successive stages of the ventral visual stream are consistent with hierarchical error signals. *eLife*, 7:e42870, Nov. 2018. ISSN 2050-084X. doi: 10.7554/eLife.42870.

E. M. Izhikevich. Simple Model of Spiking Neuron. *IEEE Transactions on Neural Networks*, 14 (6):1569–1572, 2003.

E. M. Izhikevich. Which model to use for cortical spiking neurons? *IEEE Transactions on Neural Networks*, 15(5):1063–1070, Sept. 2004.

E. M. Izhikevich and G. M. Edelman. Large-scale model of mammalian thalamocortical systems. *Proceedings of the National Academy of Sciences*, 105(9):3593–3598, Mar. 2008. doi: 10.1073/pnas.0712231105.

T. Jansen, K. A. De Jong, and I. Wegener. On the choice of the offspring population size in evolutionary algorithms. *Evolutionary Computation*, 13(4):413–440, 2005. ISSN 1063-6560. doi: 10.1162/106365605774666921.

J. Jordan, M. Schmidt, W. Senn, and M. A. Petrovici. Evolving interpretable plasticity for spiking networks. *eLife*, 10:e66273, Oct. 2021. ISSN 2050-084X. doi: 10.7554/eLife.66273.

N. K. Kasabov. NeuCube: A spiking neural network architecture for mapping, learning and understanding of spatio-temporal brain data. *Neural Networks*, 52:62–76, 2014. ISSN 0893-6080. doi: 10.1016/j.neunet.2014.01.006.

M. J. Kearns, W. H. Warren, A. P. Duchon, and M. J. Tarr. Path integration from optic flow and body senses in a homing task. *Perception*, 31(3):349–374, 2002. ISSN 0301-0066. doi: 10.1068/p3311.

N. R. Kinsky, W. Mau, D. W. Sullivan, S. J. Levy, E. A. Ruesch, and M. E. Hasselmo. Trajectory-modulated hippocampal neurons persist throughout memory-guided navigation. *Nature Communications*, 11(1):2443, May 2020. ISSN 2041-1723. doi: 10.1038/s41467-020-16226-4.

M. P. Kirschen, M. J. Kahana, R. Sekuler, and B. Burack. Optic Flow Helps Humans Learn to Navigate through Synthetic Environments. *Perception*, 29(7):801–818, July 2000. ISSN 0301-0066, 1468-4233. doi: 10.1068/p3096.

T. Kitanishi, R. Umaba, and K. Mizuseki. Robust information routing by dorsal subiculum neurons. *bioRxiv : the preprint server for biology*, 2020. doi: 10.1101/2020.06.04.133256.

H. Komatsu and R. H. Wurtz. Relation of Cortical Areas MT and MST to Pursuit Eye Movements. I. Localization and Visual Properties of Neurons. *Journal of Neurophysiology*, 60, 1988.

A. O. Komendantov, D. W. Wheeler, D. J. Hamilton, and G. A. Ascoli. Simple models of quantitative firing phenotypes in hippocampal neurons: Comprehensive coverage of intrinsic diversity. *PLOS Computational Biology*, 15(10):e1007462, Oct. 2019.

A. J. Kooistra, S. Mordalski, G. Pándy-Szekeres, M. Esguerra, A. Mamyrbekov, C. Munk, G. M. Keserű, and D. E. Gloriam. GPCRdb in 2021: Integrating GPCR sequence, structure and function. *Nucleic Acids Research*, 49(D1):D335–D343, Dec. 2020. ISSN 0305-1048. doi: 10.1093/nar/gkaa1080.

J. D. Kopsick, C. Tecuatl, K. Moradi, S. M. Attili, H. J. Kashyap, J. Xing, K. Chen, J. L. Krichmar, and G. A. Ascoli. Robust resting-state dynamics in a large-scale spiking neural network model of area CA3 in the mouse hippocampus. *Cognitive Computation*, Jan. 2022.

J. Krichmar. Design principles for biologically inspired cognitive robotics. *Biologically Inspired Cognitive Architectures*, 1:73–81, July 2012. doi: 10.1016/j.bica.2012.04.003.

J. Krichmar. A Neurobiologically Inspired Plan Towards Cognitive Machines. In *AAAI Spring Symposium: Towards Conscious AI Systems*, 2019.

J. L. Krichmar. The neuromodulatory system: A framework for survival and adaptive behavior in a challenging world. *Adaptive Behavior*, 16:385–399, 2008. ISSN 1741-2633. doi: 10.1177/1059712308095775.

J. L. Krichmar, D. A. Nitz, J. A. Gally, and G. M. Edelman. Characterizing functional hippocampal pathways in a brain-based device as it solves a spatial memory task. *Proceedings of the National Academy of Sciences of the United States of America*, 102(6):2111–2116, Feb. 2005. doi: 10.1073/pnas.0409792102.

J. L. Krichmar, W. Severa, M. S. Khan, and J. L. Olds. Making BREAD: Biomimetic strategies for artificial intelligence now and in the future. *Frontiers in Neuroscience*, 13, 2019. doi: 10.3389/fnins.2019.00666.

E. Kropff, J. E. Carmichael, M.-B. Moser, and E. I. Moser. Speed cells in the medial entorhinal cortex. *Nature*, 523(7561):419–424, 2015.

J. L. Kubie, R. U. Muller, and E. Bostock. Spatial Firing Properties of Hippocampal Theta Cells. *Journal of Neuroscience*, 10:111–123, 1990.

Lakna. What is the difference between Neurotransmitter and Neuromodulator. *PEDIAA*, 2019.

O. W. Layton and N. A. Browning. A Unified Model of Heading and Path Perception in Primate MSTd. *PLoS Computational Biology*, 10(2), 2014. ISSN 15537358. doi: 10.1371/journal.pcbi.1003476.

O. W. Layton and D. C. Niehorster. A model of how depth facilitates scene-relative object motion perception. *PLOS Computational Biology*, 15(11):e1007397, Nov. 2019. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1007397.

D. Ledergerber, C. Battistin, J. S. Blackstad, R. J. Gardner, M. P. Witter, M.-B. Moser, Y. Roudi, and E. I. Moser. Task-dependent mixed selectivity in the subiculum. *Cell Reports*, 35(8):109175, May 2021. ISSN 22111247. doi: 10.1016/j.celrep.2021.109175.

J. Lehman and R. Miikkulainen. Neuroevolution. *Scholarpedia*, 8(6):30977, June 2013. ISSN 1941-6016. doi: 10.4249/scholarpedia.30977.

J. K. Leutgeb, S. Leutgeb, M.-B. Moser, and E. I. Moser. Pattern Separation in the Dentate Gyrus and CA3 of the Hippocampus. *Science (80-.).*, 315(5814):961 LP – 966, Feb. 2007. doi: 10.1126/science.1135801.

C. Lever, S. Burton, A. Jeewajee, J. O'Keefe, and N. Burgess. Boundary Vector Cells in the Subiculum of the Hippocampal Formation. *Journal of Neuroscience*, 29(31):9771–9777, Aug. 2009. ISSN 0270-6474, 1529-2401. doi: 10.1523/JNEUROSCI.1319-09.2009.

H. C. Longuet-Higgins and K. Prazdny. The interpretation of a moving retinal image. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 1980.

B. D. Lucas and T. Kanade. An Iterative Image Registration Technique with an Application to Stereo Vision. *Proceedings of the 7th international joint conference on Artificial intelligence*, 1981.

S. Luke. ECJ Then and Now. *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pages 1223–1230, 2017. doi: 10.1145/3067695.3082467.

M. S. Madhav, R. P. Jayakumar, B. Li, F. Savelli, J. J. Knierim, and N. J. Cowan. Closed-loop control and recalibration of place cells by optic flow. Preprint, Neuroscience, June 2022.

T. S. Manning and K. H. Britten. Retinal Stabilization Reveals Limited Influence of Extraretinal Signals on Heading Tuning in the Medial Superior Temporal Area. *The Journal of Neuroscience*, 39(41):8064–8078, Oct. 2019. ISSN 0270-6474, 1529-2401. doi: 10.1523/JNEUROSCI.0388-19.2019.

H. Markram, Y. Wang, and M. Tsodyks. Differential signaling via the same axon of neocortical pyramidal neurons. *Proceedings of the National Academy of Sciences*, 95(9):5323–5328, Apr. 1998. doi: 10.1073/pnas.95.9.5323.

N. Matsumoto, T. Kitanishi, and K. Mizuseki. The subiculum: Unique hippocampal hub and more. *Neuroscience Research*, 143:1–12, June 2019. ISSN 0168-0102. doi: 10.1016/j.neures.2018.08.002.

B. L. McNaughton, F. P. Battaglia, O. N. Jensen, Ole Jensen, O. Jensen, O. B. Jensen, E. I. Moser, and M.-B. Moser. Path integration and the neural basis of the 'cognitive map'. *Nature Reviews Neuroscience*, 7(8):663–678, Aug. 2006. doi: 10.1038/nrn1932.

R. Miikkulainen. Neuroevolution. In *Encyclopedia of Machine Learning*. Springer, New York, 2010.

P. Mineault, S. Bakhtiari, B. Richards, and C. Pack. Your head is there to move you around: Goal-driven models of the primate dorsal pathway. In *Advances in Neural Information Processing Systems*, volume 34, pages 28757–28771. Curran Associates, Inc., 2021.

P. J. Mineault, F. A. Khawaja, D. A. Butts, and C. C. Pack. Hierarchical processing of complex motion along the primate dorsal visual pathway. *Proceedings of the National Academy of Sciences*, 109(16):5930–5930, 2012. doi: 10.1073/pnas.1115685109.

K. Moradi and G. A. Ascoli. A comprehensive knowledge base of synaptic electrophysiology in the rodent hippocampal formation. *Hippocampus*, 30(4):314–331, Aug. 2019.

R. G. Morris, P. Garrud, J. N. Rawlins, and J. O'Keefe. Place navigation impaired in rats with hippocampal lesions. *Nature*, 297(5868):681–683, June 1982. ISSN 0028-0836. doi: 10.1038/297681a0.

R. G. Morris, F. Schenk, F. Tweedie, and L. E. Jarrard. Ibotenate Lesions of Hippocampus and/or Subiculum: Dissociating Components of Allocentric Spatial Learning. *Eur. J. Neurosci.*, 2(12):1016–1028, 1990. ISSN 14609568. doi: 10.1111/j.1460-9568.1990.tb00014.x.

R. U. Muller, E. Bostock, J. S. Taube, and J. L. Kubie. On the directional firing properties of hippocampal place cells. *J. Neurosci.*, 14(12):7235–7251, 1994. ISSN 02706474. doi: 10.1523/jneurosci.14-12-07235.1994.

F. Nadim and D. Bucher. Neuromodulation of neurons and synapses. *Current Opinion in Neurobiology*, 29:48–56, Dec. 2014. ISSN 0959-4388. doi: 10.1016/j.conb.2014.05.003.

W. T. Newsome, K. H. Britten, and J. A. Movshon. Neuronal correlates of a perceptual decision. *Nature*, 341(6237):52–54, Sept. 1989. doi: 10.1038/341052a0.

L. Niedermeier, K. Chen, J. Xing, A. Das, J. Kopsick, E. Scott, N. Sutton, K. Weber, N. Dutt, and J. L. Krichmar. CARLsim 6: An Open Source Library for Large-Scale, Biologically Detailed Spiking Neural Network Simulation. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pages 1–10, July 2022. doi: 10.1109/IJCNN55064.2022.9892644.

S. Nishida, T. Kawabe, M. Sawayama, and T. Fukiage. Motion Perception: From Detection to Interpretation. *Annual Review of Vision Science*, 4(1):501–523, Sept. 2018. doi: 10.1146/annurev-vision-091517-034328.

H. Nover, C. H. Anderson, and G. C. DeAngelis. A logarithmic, scale-invariant representation of speed in macaque middle temporal area accounts for speed discrimination performance. *Journal of Neuroscience*, 25(43):10049–10060, Oct. 2005. doi: 10.1523/JNEUROSCI.1661-05.2005.

E. Oja. Simplified neuron model as a principal component analyzer. *Journal of Mathematical Biology*, 15(3):267–273, Nov. 1982. doi: 10.1007/bf00275687.

E. Oja. Oja learning rule. *Scholarpedia*, 3(3):3612, 2008. doi: 10.4249/scholarpedia.3612.

J. O'Keefe and J. Dostrovsky. The hippocampus as a spatial map. Preliminary evidence from unit activity in the freely-moving rat. *Brain Research*, 34(1):171–175, Nov. 1971. ISSN 0006-8993. doi: 10.1016/0006-8993(71)90358-1.

J. O'Keefe and L. Nadel. The hippocampus as a cognitive map. *Behavioral and Brain Sciences*, 2 (4):487–494, 1979. doi: 10.1017/S0140525X00063949.

B. Olshausen and D. Field. Sparse coding of sensory inputs. *Current Opinion in Neurobiology*, 14 (4):481–487, Aug. 2004. doi: 10.1016/j.conb.2004.07.007.

J. M. Olson, K. Tongprasearth, and D. A. Nitz. Subiculum neurons map the current axis of travel. *Nat. Neurosci.*, 20(2):170–172, Feb. 2017. ISSN 15461726. doi: 10.1038/nn.4464.

J. M. Olson, A. B. Johnson, L. Chang, E. L. Tao, and D. A. Nitz. Complementary Maps for Location and Environmental Structure in CA1 and Subiculum. *bioRxiv*, page 2021.02.01.428537, Feb. 2021. doi: 10.1101/2021.02.01.428537.

S. O'Mara. The subiculum: What it does, what it might do, and what neuroanatomy has yet to tell us. *J. Anat.*, 207(3):271–282, Sept. 2005. ISSN 0021-8782. doi: 10.1111/j.1469-7580.2005.00446.x.

G. Pándy-Szekeres, M. Esguerra, A. S. Hauser, J. Caroli, C. Munk, S. Pilger, G. M. Keserű, A. J. Kooistra, and D. E. Gloriam. The G protein database, GproteinDb. *Nucleic Acids Research*, 50 (D1):D518–D525, Jan. 2022.

N. Pavlidis, O. Tasoulis, V. Plagianakos, G. Nikiforidis, and M. Vrahatis. Spiking neural network training using evolutionary algorithms. In *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, volume 4, pages 2190–2194 vol. 4, 2005. doi: 10.1109/IJCNN.2005.1556240.

J. A. Perrone. Anisotropic responses to motion toward and away from the eye. *Perception & Psychophysics*, 39(1):1–8, Jan. 1986. ISSN 0031-5117, 1532-5962. doi: 10.3758/BF03207577.

J. A. Perrone. A Model of Self-motion Estimation Within Primate Extrastriate Visual Cortex. *Vision Research*, 34:2917–2938, 1994.

J. A. Perrone and L. S. Stone. Emulating the Visual Receptive-Field Properties of MST Neurons with a Template Model of Heading Estimation. *Journal of Neuroscience*, 1998.

O. Potvin, F. Y. Doré, and S. Goulet. Contributions of the dorsal hippocampus and the dorsal subiculum to processing of idiothetic information and spatial memory. *Neurobiol. Learn. Mem.*, 87(4):669–678, May 2007. ISSN 10747427. doi: 10.1016/j.nlm.2007.01.002.

S. Raiguel, M. M. Van Hulle, D. K. Xiao, V. L. Marcar, L. Lagae, and G. A. Orban. Size and shape of receptive fields in the medial superior temporal area (MST) of the macaque. *NeuroReport*, 8 (12):2803–2808, 1997. doi: 10.1097/00001756-199708180-00030.

R. Rasmussen and K. Yonehara. Circuit Mechanisms Governing Local vs. Global Motion Processing in Mouse Visual Cortex. *Frontiers in Neural Circuits*, 11, 2017. ISSN 1662-5110.

F. Raudies. Optic flow. *Scholarpedia*, 8(7):30724, 2013. ISSN 1941-6016. doi: 10.4249/scholarpedia.30724.

A. D. Redish and D. S. Touretzky. Cognitive maps beyond the hippocampus. *Hippocampus*, 7(1): 15–35, 1997. ISSN 1098-1063. doi: 10.1002/(SICI)1098-1063(1997)7:1<15::AID-HIPO3>3.0.CO;2-6.

W. Reichardt. Autocorrelation, a principle for the evaluation of sensory information by the central nervous system. 1961.

B. A. Richards, T. P. Lillicrap, P. Beaudoin, Y. Bengio, R. Bogacz, A. Christensen, C. Clopath, R. P. Costa, A. de Berker, S. Ganguli, C. J. Gillon, D. Hafner, A. Kepecs, N. Kriegeskorte, P. Latham, G. W. Lindsay, K. D. Miller, R. Naud, C. C. Pack, P. Poirazi, P. Roelfsema, J. Sacramento, A. Saxe, B. Scellier, A. C. Schapiro, W. Senn, G. Wayne, D. Yamins, F. Zenke, J. Zylberberg, D. Therien, and K. P. Kording. A deep learning framework for neuroscience. *Nature Neuroscience*, 22(11):1761–1770, Nov. 2019. ISSN 1546-1726. doi: 10.1038/s41593-019-0520-2.

R. Rideaux, K. R. Storrs, G. Maiello, and A. E. Welchman. How multisensory neurons solve causal inference. *Proceedings of the National Academy of Sciences*, 118(32):e2106235118, Aug. 2021. doi: 10.1073/pnas.2106235118.

M. R. Rosenzweig, S. M. Breedlove, and N. V. Watson. *Biological Psychology*. Sinauer Associates, Sunderland, MA, fifth edition, June 2007.

E. L. Rounds, A. S. Alexander, D. A. Nitz, and J. L. Krichmar. Conjunctive Coding in an Evolved Spiking Model of Retrosplenial Cortex. *Behav. Neurosci.*, 2018. ISSN 19390084. doi: 10.1037/bne0000236.

C. S. Royden. Mathematical analysis of motion-opponent mechanisms used in the determination of heading and depth. *Journal of the Optical Society of America. A, Optics, Image Science, and Vision*, 14(9):2128–2143, Sept. 1997. ISSN 1084-7529. doi: 10.1364/josaa.14.002128.

C. S. Royden, M. S. Banks, and J. A. Crowell. The perception of heading during eye movements. *Nature*, 360(6404):583–585, Dec. 1992. ISSN 1476-4687. doi: 10.1038/360583a0.

N. C. Rust, O. Schwartz, J. A. Movshon, and E. P. Simoncelli. Spatiotemporal elements of macaque v1 receptive fields. *Neuron*, 46(6):945–956, June 2005. ISSN 0896-6273. doi: 10.1016/j.neuron.2005.05.021.

N. C. Rust, V. Mante, E. P. Simoncelli, and J. A. Movshon. How MT cells analyze the motion of visual patterns. *Nature Neuroscience*, 9(11):1421–1431, Nov. 2006. doi: 10.1038/nn1786.

C. Schmitt, B. R. Baltaretu, J. D. Crawford, and F. Bremmer. A Causal Role of Area hMST for Self-Motion Perception in Humans. *Cerebral Cortex Communications*, 1(1):tgaa042, Jan. 2020. ISSN 2632-7376. doi: 10.1093/texcom/tgaa042.

M. Sereno and M. Sereno. Learning to see rotation and dilation with a Hebb rule. *Advances in Neural Information Processing Systems 3.*, 1991.

P. E. Sharp and C. Green. Spatial correlates of firing patterns of single cells in the subiculum of the freely moving rat. *J. Neurosci.*, 14(4):2339–2356, 1994. ISSN 02706474. doi: 10.1523/jneurosci.14-04-02339.1994.

E. P. Simoncelli and D. J. Heeger. A model of neuronal responses in visual area MT. *Vision Research*, 38(5):743–761, Mar. 1998. ISSN 0042-6989. doi: 10.1016/s0042-6989(97)00183-1.

J. Sjöström and W. Gerstner. Spike-timing dependent plasticity. *Scholarpedia*, 5(2):1362, 2010. doi: 10.4249/scholarpedia.1362.

P. J. Sjöström, E. A. Rancz, A. Roth, and M. Häusser. Dendritic excitability and synaptic plasticity. *Physiological Reviews*, 88(2):769–840, Apr. 2008. doi: 10.1152/physrev.00016.2007.

W. E. Skaggs, B. L. McNaughton, K. M. Gothard, and E. J. Markus. An Information-Theoretic Approach to Deciphering the Hippocampal Code. *Proceedings of the 5th International Conference on Neural Information Processing Systems*, 1993. doi: 10.1109/EUMC.2007.4405137.

W. E. Skaggs, B. L. McNaughton, M. A. Wilson, and C. A. Barnes. Theta phase precession in hippocampal neuronal populations and the compression of temporal sequences. *Hippocampus*, 6(2):149–172, Jan. 1996. ISSN 1050-9631. doi: 10.1002/(SICI)1098-1063(1996)6:2<149::AID-HIPO6>3.0.CO;2-K.

R. J. Snowden, S. Treue, R. G. Erickson, and R. A. Andersen. The response of area MT and V1 neurons to transparent motion. *Journal of Neuroscience*, 11(9):2768–2785, Sept. 1991. ISSN 02706474. doi: 10.1523/jneurosci.11-09-02768.1991.

J. P. Snyder. Map projections: A working manual. *Professional Paper*, 1987. doi: 10.3133/pp1395.

T. Solstad, E. I. Moser, and G. T. Einevoll. From grid cells to place cells: A mathematical model. *Hippocampus*, 16(12):1026–1031, 2006. ISSN 1098-1063. doi: 10.1002/hipo.20244.

S. Song, K. Miller, and L. Abbott. Competitive Hebbian learning through spike timing-dependent plasticity. *Nature neuroscience*, 3:919–26, Oct. 2000. doi: 10.1038/78829.

K. L. Stachenfeld, M. M. Botvinick, and S. J. Gershman. The hippocampus as a predictive map. *Nature Neuroscience*, 20(11):1643–1653, 2017. doi: 10.1038/nn.4650.

M. Stimberg, R. Brette, and D. F. Goodman. Brian 2, an intuitive and efficient neural simulator. *eLife*, 8:e47314, Aug. 2019. ISSN 2050-084X. doi: 10.7554/eLife.47314.

Y. Sun, S. Jin, X. Lin, L. Chen, X. Qiao, L. Jiang, P. Zhou, K. G. Johnston, P. Golshani, Q. Nie, T. C. Holmes, D. A. Nitz, and X. Xu. CA1-projecting subiculum neurons facilitate object–place learning. *Nature Neuroscience*, 22(11):1857–1870, Nov. 2019. ISSN 15461726. doi: 10.1038/s41593-019-0496-y.

K. Takahashi, Y. Gu, P. J. May, S. D. Newlands, G. C. DeAngelis, and D. E. Angelaki. Multimodal Coding of Three-Dimensional Rotation and Translation in Area MSTd: Comparison of Visual and Vestibular Selectivity. *Journal of Neuroscience*, 2007. ISSN 0270-6474. doi: 10.1523/JNEUROSCI.0817-07.2007.

K. Tanaka and H. Saito. Analysis of motion of the visual field by direction, expansion/contraction, and rotation cells clustered in the dorsal part of the medial superior temporal area of the macaque monkey. *Journal of Neurophysiology*, 62(3):626–641, Sept. 1989. ISSN 0022-3077. doi: 10.1152/jn.1989.62.3.626.

K. Tanaka, Y. Fukada, and H. A. Saito. Underlying mechanisms of the response specificity of expansion/contraction and rotation cells in the dorsal part of the medial superior temporal area of the macaque monkey. *Journal of Neurophysiology*, 62(3):642–656, Sept. 1989. ISSN 0022-3077. doi: 10.1152/jn.1989.62.3.642.

J. S. Taube, R. U. Muller, and J. B. Ranck. Head-direction cells recorded from the postsubiculum in freely moving rats. I. Description and quantitative analysis. *Journal of Neuroscience*, 10(2):420–435, Feb. 1990. ISSN 0270-6474, 1529-2401. doi: 10.1523/JNEUROSCI.10-02-00420.1990.

C. Tecuatl, D. W. Wheeler, N. Sutton, and G. A. Ascoli. Comprehensive estimates of potential synaptic connections in local circuits of the rodent hippocampal formation by axonal-dendritic overlap. *Journal of Neuroscience*, 41(8):1665–1683, Feb. 2021.

E. C. Tolman. Cognitive maps in rats and men. *Psychological Review*, 55:189–208, 1948. ISSN 1939-1471. doi: 10.1037/h0061626.

M. Tsodyks and T. Sejnowski. Associative memory and hippocampal place cells. *International Journal of Neural Systems*, 6:81–86, 1995.

M. Tsodyks and S. Wu. Short-term synaptic plasticity. *Scholarpedia*, 8(10):3153, Oct. 2013. ISSN 1941-6016. doi: 10.4249/scholarpedia.3153.

G. G. Turrigiano and S. B. Nelson. Homeostatic plasticity in the developing nervous system. *Nature Reviews Neuroscience*, 5(2):97–107, Feb. 2004. ISSN 1471-0048. doi: 10.1038/nrn1327.

G. G. Turrigiano, K. R. Leslie, N. S. Desai, L. C. Rutherford, and S. B. Nelson. Activity-dependent scaling of quantal amplitude in neocortical neurons. *Nature*, 391(6670):892–896, Feb. 1998. ISSN 00280836. doi: 10.1038/36103.

L. G. Ungerleider and L. Pessoa. What and where pathways. *Scholarpedia*, 3(11):5342, Nov. 2008. ISSN 1941-6016. doi: 10.4249/scholarpedia.5342.

W. E. Vinje and J. L. Gallant. Sparse coding and decorrelation in primary visual cortex during natural vision. *Science (New York, N.Y.)*, 287(5456):1273–1276, Feb. 2000. doi: 10.1126/science.287.5456.1273.

M.-A. T. Vu, T. Adalı, D. Ba, G. Buzsáki, D. Carlson, K. Heller, C. Liston, C. Rudin, V. S. Sohal, A. S. Widge, H. S. Mayberg, G. Sapiro, and K. Dzirasa. A Shared Vision for Machine Learning in Neuroscience. *Journal of Neuroscience*, 38(7):1601–1607, Feb. 2018. ISSN 0270-6474, 1529-2401. doi: 10.1523/JNEUROSCI.0508-17.2018.

Y. Wang, H. Markram, P. H. Goodman, T. K. Berger, J. Ma, and P. S. Goldman-Rakic. Heterogeneity in the pyramidal network of the medial prefrontal cortex. *Nature Neuroscience*, 9(4):534–542, Apr. 2006. ISSN 1546-1726. doi: 10.1038/nn1670.

W. H. Warren and D. J. Hannon. Direction of self-motion is perceived from optical flow. *Nature*, 336(6195):162–163, Nov. 1988. ISSN 0028-0836, 1476-4687. doi: 10.1038/336162a0.

B. Wild and S. Treue. Primate extrastriate cortical area MST: A gateway between sensation and cognition. *Journal of neurophysiology*, 125(5):1851–1882, May 2021. doi: 10.1152/JN.00384.2020/ASSET/IMAGES/LARGE/JN.00384.2020_F0015.JPEG.

MA. Wilson and BL. McNaughton. Dynamics of the hippocampal ensemble code for space. *Science*, 261(5124):1055–1058, 1993. ISSN 0036-8075. doi: 10.1126/science.8351520.

Q. Xu, J. Peng, J. Shen, H. Tang, and G. Pan. Deep CovDenseSNN: A hierarchical event-driven dynamic framework with spiking neurons in noisy environment. *Neural Networks*, 121:512–519, 2020. ISSN 0893-6080. doi: 10.1016/j.neunet.2019.08.034.

X. Xu, Y. Sun, T. C. Holmes, and A. J. López. Noncanonical connections between the subiculum and hippocampal CA1. *Journal of Comparative Neurology*, 524(17):3666–3673, 2016. doi: 10.1002/cne.24024.

E. Yavuz, J. Turner, and T. Nowotny. GeNN: A code generation framework for accelerated brain simulations. *Scientific Reports*, 6(1):18854, Jan. 2016. ISSN 2045-2322. doi: 10.1038/srep18854.

C. P. Yu, W. K. Page, R. Gaborski, and C. J. Duffy. Receptive Field Dynamics Underlying MST Neuronal Optic Flow Selectivity. *J Neu-rophysiol*, 103:2794–2807, 2010. doi: 10.1152/jn.01085.2009.

R. S. Zemel and T. J. Sejnowski. A Model for Encoding Multiple Object Motions and Self-Motion in Area MST of Primate Visual Cortex. *The Journal of Neuroscience*, 1998.

K. Zhang, M. I. Sereno, and M. E. Sereno. Emergence of Position-Independent Detectors of Sense of Rotation and Dilation with Hebbian Learning: An Analysis. *Neural Computation*, 5(4):597–612, 1993. doi: 10.1162/neco.1993.5.4.597.