Title
Sparse Coding, Dimensionality Reduction, and Synaptic Plasticity: Evolving and Validating a Biologically Realistic Model of Retrosplenial Cortex

Permalink
https://escholarship.org/uc/item/5wh7x2sb

Author
Rounds, Emily Lauren

Publication Date
2017

License
CC BY 4.0

Peer reviewed|Thesis/dissertation
UNIVERSITY OF CALIFORNIA,
IRVINE

Sparse Coding, Dimensionality Reduction, and Synaptic Plasticity: Evolving and Validating a Biologically Realistic Model of Retrosplenial Cortex

DISSEPTION

submitted in partial satisfaction of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in Psychology

by

Emily Lauren Rounds

Dissertation Committee:
Professor Jeffrey L. Krichmar, Chair
Professor Nikil D. Dutt
Professor Ramesh Srinivasan

2017
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIST OF FIGURES</td>
<td>iv</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td>v</td>
</tr>
<tr>
<td>ACKNOWLEDGMENTS</td>
<td>vi</td>
</tr>
<tr>
<td>CURRICULUM VITAE</td>
<td>vii</td>
</tr>
<tr>
<td>ABSTRACT OF THE DISSERTATION</td>
<td>viii</td>
</tr>
<tr>
<td>INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>References</td>
<td>8</td>
</tr>
<tr>
<td>CHAPTER 1: Development of an Evolutionary Framework</td>
<td>10</td>
</tr>
<tr>
<td>Introduction</td>
<td>10</td>
</tr>
<tr>
<td>Retrosplenial Cortex Dataset and Experimental Task</td>
<td>11</td>
</tr>
<tr>
<td>Modeling of Behavioral Metrics</td>
<td>11</td>
</tr>
<tr>
<td>Parameters and Training</td>
<td>14</td>
</tr>
<tr>
<td>Evolutionary Algorithm</td>
<td>17</td>
</tr>
<tr>
<td>Fitness Values and Firing Rate Correlations</td>
<td>19</td>
</tr>
<tr>
<td>Replicating Empirical Data</td>
<td>21</td>
</tr>
<tr>
<td>Replicating Population Behavior</td>
<td>25</td>
</tr>
<tr>
<td>Discussion</td>
<td>27</td>
</tr>
<tr>
<td>References</td>
<td>30</td>
</tr>
<tr>
<td>CHAPTER 2: Development of a Virtual Testbed</td>
<td>31</td>
</tr>
<tr>
<td>Introduction</td>
<td>31</td>
</tr>
<tr>
<td>Experimental Task and Dataset</td>
<td>32</td>
</tr>
<tr>
<td>CARLsim</td>
<td>33</td>
</tr>
<tr>
<td>Examining the Impact of Lesions on Network Performance</td>
<td>33</td>
</tr>
<tr>
<td>Verification of Complete Model Functionality</td>
<td>34</td>
</tr>
<tr>
<td>Rearranging Turn Sequences</td>
<td>34</td>
</tr>
<tr>
<td>Rotating Track Positions</td>
<td>35</td>
</tr>
<tr>
<td>Linearly Translating Track Positions</td>
<td>36</td>
</tr>
<tr>
<td>Matched Neurons Conjunctively Encode</td>
<td>38</td>
</tr>
<tr>
<td>Multiple Spatial Reference Frames</td>
<td>38</td>
</tr>
<tr>
<td>Evolved SNNs Are Robust to Loss of Input Streams</td>
<td>40</td>
</tr>
<tr>
<td>Rearranging Turn Sequence Causes Functional Remapping</td>
<td>43</td>
</tr>
<tr>
<td>Model Generates Robust Representations for Novel Track Positions</td>
<td>46</td>
</tr>
<tr>
<td>Neurons Undergo Functional Remapping</td>
<td>53</td>
</tr>
<tr>
<td>Discussion</td>
<td>55</td>
</tr>
</tbody>
</table>
LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Network Architecture</td>
<td>12</td>
</tr>
<tr>
<td>1.2</td>
<td>CARLsim-ECJ Pipeline for Evolving SNNs</td>
<td>18</td>
</tr>
<tr>
<td>1.3</td>
<td>Fitness Curve</td>
<td>20</td>
</tr>
<tr>
<td>1.4</td>
<td>Functional Neuron Type Examples</td>
<td>22</td>
</tr>
<tr>
<td>1.5</td>
<td>Functional Neuron Type Distributions</td>
<td>24</td>
</tr>
<tr>
<td>1.6</td>
<td>Reconstruction of Position from Ensemble Correlations</td>
<td>26</td>
</tr>
<tr>
<td>2.1</td>
<td>Experimental-Computational Pipeline</td>
<td>31</td>
</tr>
<tr>
<td>2.2</td>
<td>Schematic of Rearranged Turn Sequences</td>
<td>35</td>
</tr>
<tr>
<td>2.3</td>
<td>Schematics of Track Orientation and Linear Translation</td>
<td>37</td>
</tr>
<tr>
<td>2.4</td>
<td>Synthetic Neuron Activity Conjunctively Encodes Multiple Reference Frames</td>
<td>39</td>
</tr>
<tr>
<td>2.5</td>
<td>Evolved Models of RSC Match Electrophysiological Data When Explicitly Matched Neurons are Excluded.</td>
<td>40</td>
</tr>
<tr>
<td>2.6</td>
<td>Fitness and Reconstruction Errors for Simulated Lesion Experiments</td>
<td>41</td>
</tr>
<tr>
<td>2.7</td>
<td>Model Behavior Generalizes to Novel Route Configurations</td>
<td>44</td>
</tr>
<tr>
<td>2.8</td>
<td>Network Responses to Orientation and Translation Manipulations</td>
<td>47</td>
</tr>
<tr>
<td>2.9</td>
<td>Network Representations Change Based on Amount of Conflict Between Relevant Behavioral Variables</td>
<td>49</td>
</tr>
<tr>
<td>2.10</td>
<td>Variable Conflict Results in Failure to Disambiguate Similar Routes</td>
<td>51</td>
</tr>
<tr>
<td>2.11</td>
<td>Example Neurons that Underwent Functional Remapping During Track Manipulation</td>
<td>54</td>
</tr>
</tbody>
</table>
Figure 3.1  NMF Can Recover Sparse, Parts-Based Representations of High-Dimensional Input Data

Figure 3.2  NSC Can Predict Response Properties of Biological Neurons

Figure 3.3  STDPH and NMF Can Yield Results Qualitatively Similar to Experimental Data

Figure 3.4  Receptive Fields Generated by NSC and STDPH are Qualitatively Similar
## LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table 1.1</td>
<td>Parameters Initialized Via the ECJ Framework</td>
<td>14</td>
</tr>
<tr>
<td>Table 1.2</td>
<td>Evolved Solutions and Uniqueness Over Ten Runs</td>
<td>15</td>
</tr>
<tr>
<td>Table 1.3</td>
<td>Average Runtimes in Minutes (Mean/Std. Dev)</td>
<td>19</td>
</tr>
<tr>
<td>Table 2.1</td>
<td>Firing Rates of Lesioned Networks</td>
<td>42</td>
</tr>
<tr>
<td>Table 2.2</td>
<td>Network Fitness Scores for Resequenced Routes Under Different Matching Conditions</td>
<td>45</td>
</tr>
<tr>
<td>Table 2.3</td>
<td>Percentages of Neurons that Remap During Track Manipulation Experiments</td>
<td>53</td>
</tr>
<tr>
<td>Table 3.1</td>
<td>Nonnegative Sparse Coding in the Brain</td>
<td>69</td>
</tr>
</tbody>
</table>
ACKNOWLEDGMENTS

I would like to thank the people who have supported me through the years as I developed and wrote this dissertation. It has been one of the most significant and challenging undertakings of my life, and has been strenuous and exciting in equal measure. Without the support of the following people, I might never have crossed the finish line.

- I would first and foremost like to thank my advisor and committee chair, Professor Jeffrey L. Krichmar, for his patience and guidance through the years. He has been a consistent source of inspiration, support, and careful feedback. Thanks are also due to Professor Douglas A. Nitz, who has functioned as a co-advisor on the projects I undertook. He has not only provided valuable feedback and guidance over the years, but also the data upon which this dissertation was built. I could not have completed it without their assistance.

- I would also like to thank my committee, Professor Nikil Dutt and Professor Ramesh Srinivasan. They have provided helpful suggestions, feedback, and motivation over the years.

- Thanks also to my colleagues, who have made the lab a truly great and pleasant place to work: Kristofor Carlson, Michael Beyeler, Alexis Craig, Ting-Shuo Chu, Tiffany Hwu, Hirak Kashyap, Xinyun Zou, and Colleen Chen. Thanks especially to Kristofor Carlson for having the tremendous patience and fortitude to teach me everything I now know about evolving spiking networks.

- Thanks to the National Science Foundation (NSF) and Northrop Grumman Aerospace Systems for their financial support.

- Thanks to my friends and family for supporting me through these years: Joseph Bomber, who believed in me and provided the mental fortitude I needed in the darkest times; Rukh, Jaime, and Lutece, who reminded me to prioritize self-care and to always center my values in all that I do; and my parents, Garth and Sue Rounds, for insisting that I should never give up once I set my mind to something. I could not have done this without them.
CURRICULUM VITAE

Emily Rounds

Department of Cognitive Sciences  
School of Social Sciences  
University of California, Irvine,  
92617

Education

University of California, Irvine  
Ph.D.  Psychology - Cognitive Neuroscience  
2017
University of California, Irvine  
M.Sc.  Psychology - Cognitive Neuroscience  
2014
Pacific University, Oregon  
B.Sc.  Psychology  
2011

Research Experience

Graduate Research Assistant  
2011 - 2017  
Dept. of Cognitive Sciences, University of California at Irvine  
I have worked in the Cognitive Anteater Robotics Laboratory (CARL), pursuing my Ph.D. I have worked to model neural networks of memory, navigation, and learning, and have aided in the development of automatic tuning methods for such models using evolutionary strategies.

Teaching Experience

Teaching Assistant  
2011 - 2017  
Dept. of Cognitive Sciences, University of California at Irvine  
I have aided professors in teaching a variety of courses from introductory psychology to cognitive robotics by running discussion and laboratory sections. Leading discussion sections involved formulating weekly topics and questions for students to answer and facilitating group discussions. Teaching laboratory sections involved setting up lab materials, introducing lab topics, and helping students to complete lab work.

Publications


Posters


Scientific Talks

• Rounds, E. A Framework for Evolving Models of the Brain: Complete and Continuous Representation of Space Emerges in SNNs Tuned to Match Neurophysiological Data. Talk given at the Technical University of Munich (2016) in Munich, Germany.

Awards

• Graduate Dean’s Dissertation Completion Fellowship, 2017
• Psi Chi Honors Society, 2010
• President’s Scholarship, 2007
ABSTRACT OF THE DISSERTATION

Sparse Coding, Dimensionality Reduction, and Synaptic Plasticity: Evolving and Validating a Biologically Realistic Model of Retrosplenial Cortex

By

Emily Rounds

Doctor of Philosophy in Psychology

University of California, Irvine, 2017

Professor Jeffrey L. Krichmar, Chair

In this dissertation, I set out to develop novel methodologies for the evolution and evaluation of spiking neural networks. This series of studies involved the use of GPU-accelerated, parallelized evolutionary algorithms. The project was intended to aid collaboration efforts between theoretical and experimental neuroscientists, who often spend tremendous time and money developing experiments that may not provide useful results. It was also intended to develop a veridical way of modeling neural systems by matching experimentally observed neurophysiological data. The networks evolve such that higher-order features of the region, such as functional behavior and population coding, emerge by virtue of replicated firing patterns. In the first study, I developed the automated tuning framework and applied it to a case study using a dataset recorded from rat retrosplenial cortex. The framework successfully takes as input the recorded behavioral metrics associated with neuronal firing patterns which are encoded by idealized input neurons and evolves a generic spiking neural network to match the data by correlating the synthetic neuronal response patterns with the experimentally observed ones. In the second
study, I developed a virtual testbed to evaluate evolved models of cognitive function. Within the framework, novel experimental designs can be simulated and model response patterns can be recorded. By simulating experiments such as lesioning of the network and manipulation of behavioral inputs, new predictions can be made about the function of the brain region, and new experiments to probe that function can be designed without expending unnecessary time and effort on the part of experimentalists. In the final study, I link spike-timing plasticity to dimensionality reduction in the brain by applying a statistical algorithm known as nonnegative matrix factorization (NMF) to the same dataset. I demonstrate that similar results, and a similar model of RSC functionality, can be achieved simply through nonnegative and parts-based dimensionality reduction, and propose that nonnegative sparse coding may be a canonical computation performed by plasticity rules in the brain to handle high-dimensional input spaces.
INTRODUCTION

As the power and availability of high-performance computing resources grows, large and biologically realistic networks of spiking neurons are becoming increasingly relevant as a computational modeling tool. Networks consisting of on the order of hundreds or thousands of neurons allow researchers to formulate models that can represent how neural circuits give rise to cognition and behavior (Krichmar, Coussy and Dutt, 2015), and they allow engineers to prototype novel mechanisms that may prove useful in applications of neuromorphic hardware (Hu, Li, Chen, Wu, Rose and Linderman, 2014).

An important step in the design of these networks is the selection of parameter values that enable the model to perform a desired target function. Simulations of spiking neural networks (SNNs) tend to be very computationally expensive, and involve a large number of free parameters. For instance, even after a model of a neurological system has been constrained with the best available physiological data, it is not uncommon for an SNN to exhibit tens or hundreds of thousands of unknown synaptic weight parameters that must be specified by the model designer. Furthermore, SNN applications are often based on recurrent network topologies, where gradient-based optimization methods (such as backpropagation) are inapplicable. For these reasons, the task of parameterizing an SNN to solve a particular task, or to accurately model particular biological data, is an especially difficult kind of neural network optimization problem.

In Chapter 1, we propose a two-pronged framework for tuning the parameters of spiking neural networks. First, we achieve a drastic reduction in the dimensionality of the parameter space by using a learning mechanism as an \textit{indirect encoding} method for
automatically adapting the weights of neural connections. This allows us to use an evolutionary algorithm (EA) to tune only the coarse-grained structure of the network and the global parameters of the learning method itself. Second, we use a GPU-based SNN simulator to accelerate fitness evaluation. This allows us to compensate for the increased computational effort that is required to train the networks through learning. To learn the synaptic weights, we apply a standard nearest neighbor implementation of spike-timing-dependent plasticity (STDP) (Izhikevich and Desai, 2003) a widely-used and biologically realistic model of synaptic plasticity which has been studied experimentally (Bi and Poo, 1998) as well as computationally.

We demonstrate the functionality of this framework by applying it to a case study in which an SNN is tuned to match neural recordings from the rat retrosplenial cortex (RSC) (Alexander and Nitz, 2015). To our knowledge, this is the first attempt to apply search algorithms to train SNNs to replicate neurophysiological data from awake, behaving animals. Existing work in the area of SNN synthesis has either trained recurrent networks to match high-level animal behavior in cognitive tasks (Carnevale, deLafuente, Romo, Barak and Parga, 2015); (Mante, Sussillo, Shenoy and Newsome, 2013); (Song, Yang and Wang, 2016), or it has focused on tuning the parameters of individual neuron models to match electrophysiological data (Fountas and Shanahan, 2015); (Prinz, Billimoria and Marder, 2003); (Prinz, Bucher and Marder, 2004); (Rossant, Goodman, Fontaine, Platkiewicz, Magnusson and Brette, 2011). However, to better understand the mechanisms underlying neurological circuits and to verify theoretical models of cognition, it is important that they can match neurological data in terms of neuronal firing rates as well as population functionality and behavior. Sometimes the choice of these parameters can be constrained
by high-quality physiological data (Tripathy, Savitskaya, Burton, Urban and Gerkin, 2014),
but even with the best-understood brain regions we almost never know the precise value
that these parameters should assume to best mimic nature. We show that this can be done
effectively using an evolutionary parameter tuning framework.

In general, neural networks have been successfully evolved using both direct and
indirect encoding schemes. The NEAT and HyperNEAT algorithms (Stanley and
Miikkulainen, 2002); (Stanley, DÁmbrosio and Gauci, 2009) utilize an indirect encoding
scheme in order to evolve increasingly complex network topologies, while Carlson et al.
(Carlson, Nageswaran, Dutt and Krichmar, 2014) used a similar approach to ours to evolve
SNNs whose neuronal responses gave rise to receptive fields similar to those found in
neurons from the primary visual cortex. However, this study used artificial data and did not
perform a behavioral task. Asher et al. (Asher, Krichmar and Oros, 2015) used a direct
encoding scheme to train an artificial neural network (ANN) to perform visually- and
memory-guided reaching tasks. However, this approach took thousands of generations to
evolve, and yielded a network that had less biologically realistic neuronal units. To our
knowledge, evolutionary algorithms utilizing indirect encoding have not been used to tune
the parameters of networks containing realistic spiking neurons to perform a cognitive
task. While other groups have used GPU acceleration to evolve neural networks, none have
used the method to generate mappings between experimentally recorded behavior and
neural activity.

In summary, Chapter 1 introduces a novel and generalizable framework for the
computational modeling of brain function by employing a GPU-accelerated and parallelized
framework using evolutionary algorithms to tune biologically plausible spiking SNNs with
realistic neural dynamics. Using this framework, SNNs can be trained to not only reproduce the behavior of neural circuits, but also to match empirical data at the neuron level while simultaneously capturing the holistic behavior of the circuit.

Although the framework can be applied to an electrophysiological dataset, methods to evaluate the capabilities of the evolved models are needed. To be useful, the framework must not only generate models of cognition that can describe features of a brain region, but must also be flexible enough to design novel experiments that can generate predictions about the function and behavior of the region, especially with respect to novel tasks and experimental designs; thus, Chapter 2 is devoted to examining the kinds of representations and coding schemes that emerged in the evolved model of RSC.

A body of recent literature shows that the retrosplenial cortex (RSC) acts as an association cortex for the integration of visual, spatial, and idiothetic information (Hindley, Nelson, Aggleton and Vann, 2014); (Alexander and Nitz, 2015); (Cooper and Mizumori, 1999); (Vedder, Miller, Harrison and Smith, 2016); (Chrastil, Sherrill, Hasselmo and Stern, 2016); (Epstein, 2008); (Epstein, Parker and Feiler, 2007); (Chrastil, Sherrill, Hasselmo and Stern, 2015). This integration is expressed in the RSC as a conjunctive code that binds incoming input to multiple internal and external frames of reference simultaneously (Alexander and Nitz, 2015). In principle, such encoding can support scene construction and/or scene recognition, which the RSC may use to access stored representations of familiar environments (Epstein, Parker and Feiler, 2007) and compute path integration (Chrastil, Sherrill, Hasselmo and Stern, 2015). In rodents, RSC ensembles are both flexible and robust, representing specific locomotor actions as well as the animal’s position within route space and within the broader environmental context (Alexander and Nitz, 2015).
While a great deal is known about the structure and connectivity of RSC (van Groen and Wyss, 1990); (Wyass and van Groen, 1992); (van Groen and Wyss, 2003) the contribution of the multisensory integration performed by the RSC to cognition has yet to be fully understood (Pothuizen, Aggleton and Vann, 2008). Although there is evidence to suggest that retrosplenial cortex may transform inputs from one frame of reference to another (Byrne, Becker and Burgess, 2007); (Alexander and Nitz, 2015) and that it plays a critical role in spatial representation and navigation (Takahashi, Kawamura, Shiota, Kasahata and Hirayama, 1997); (Miller, Vedder, Law and Smith, 2014); (Nelson, Hindley, Pearce, Vann and Aggleton, 2015); (Maguire, 2001), the full nature of its representations is unknown. In particular, how the disparate input streams targeting the RSC are pulled together to combine and encode different reference frames that shape animals’ navigation strategies is an outstanding and important question.

Computational neuroscience and theoretical modeling is in a unique position to help fill in the gaps of our knowledge concerning RSC. By modeling a brain region’s dynamics exhaustively, artificial neural networks can be used to design and simulate novel experiments to test hypotheses and guide experimental research. Explanatory and predictive power in such models requires synthetic neural activity capable of capturing and reproducing the observed dynamics of biological neurons in that region. In the case of RSC, such a model would need to be able to faithfully generate the complex firing patterns observed across ensembles of RSC cells, and their activity would need to be generated according to the same behavioral correlates to which activity is anchored. Finally, by developing such finely tuned models as embedded components of a larger network, it may be possible to generate highly functional networks of much larger scale.
In Chapter 2, we extended the evolutionary framework to include a virtual testbed for the simulation of novel virtual experiments. The evolved model of RSC underwent a series of virtual experiments that would be difficult and time-consuming to conduct in awake, behaving animals. Our results show that the model spontaneously generates representations for places within the environment to which it was never exposed during training and retains knowledge of position, orientation, and action within multiple routes. Model viability survives the loss of seemingly critical input streams, and disambiguates position when variables associated with route position, route orientation and position within the larger environment are in conflict. The results suggest that the RSC integrates multiple sources of information in a category-free fashion. Further, the results how this computational method can be used to analyze associative cortical regions.

Lastly, in Chapter 3, we explore the possibility that STDP may be employed by the brain to perform dimensionality reduction (in which the dimensionality of a space is reduced to the lowest possible space that encapsulates the variance of the original data via feature extraction). In the case of neuronal firing rate patterns, this means representing all possible firing rate patterns in the brain region using the smallest possible subset of the neurons. Brains face the fundamental challenge of processing, storing, and representing high-dimensional stimuli using patterns of neural activity. This challenge is complicated by strict constraints on metabolic cost (Lennie, 2003) and the widespread existence of anatomical bottlenecks (Kempermann, 2002); (Bar-Gad, Morris and Bergman, 2003); (Babinsky, Calabrese, Durwen, Markowitsch, Brechtelsbauer, Heuser and Gehlen, 1993); (Ganguli and Sompolinsky, 2012), which force the information stored in a large number of
neurons to be compressed into a typically 10 - 10,000 fold smaller population of downstream neurons.

One potential approach to addressing this challenge is to somehow reduce the number of variables required to represent a particular stimulus space (dimensionality reduction). This idea features prominently in efficient coding theories of brain function (Barlow, 1961); (Barlow, 2001); (Atick, 1992); (Linsker, 1990), which posit that the brain performs dimensionality reduction to reduce information bottlenecks by maximizing mutual information between the high-dimensional input and the low-dimensional output of neuronal populations. Several modern variants of the efficient coding hypothesis, such as independent component analysis (ICA) (Bell and Sejnowski, 1997) and nonnegative sparse coding (NSC) (Hoyer, 2002); (Hoyer, 2004) suggest that the role of cortical representations is to further reduce the redundancy of the sensory signal by separating it into its underlying causes, a process known as factor analysis.

In Chapter 3, we propose that a variety of cortical representations can be understood within the NSC framework. Using a combination of data analysis and computer simulations, we found that NSC, a combination of a statistical technique called nonnegative matrix factorization (NMF) (Paatero and Tapper, 1994); (Lee and Seung, 1999) and sparse population coding (Field, 1994), generates lower-dimensional embeddings of high-dimensional stimulus spaces that resemble neuronal population responses in a variety of brain regions. Furthermore, our observations suggest that commonly reported neuronal response properties might simply be a by-product of neurons performing a biological equivalent of dimensionality reduction on their inputs via Hebbian-like learning mechanisms.
References


FIELD, D. J. 1994. What is the goal of sensory coding? Neural computation, 6, 559-601.


CHAPTER 1: Development of an Evolutionary Framework

Introduction

In this chapter, I introduce a framework for evolving a population of spiking neural networks (SNNs) to match a electrophysiologically observed response patterns. The framework is designed to be generalizable and flexible. It takes as input an arbitrary number of behavioral metrics that are encoded by a set of idealized input neurons and converted into spike trains. The output layer consists of a population of 80% excitatory and 20% inhibitory Izhikevich neurons (Izhikevich, 2003). Synaptic plasticity rules are applied to the synaptic weights between the input and the output layers to allow the output layer to learn. Evolutionary algorithms are used to explore the parameter space governing synaptic plasticity, which depends on the precise timing between pre- and post-synaptic neuron spikes, to reduce the dimensionality of that space. A user-defined fitness function determines a matching criteria that dictates how well the synthetic neural activity in the network has captured the response patterns of neurons in the electrophysiologically recorded dataset.

We tested our STDP-based encoding method by fitting the activity of a network of 1,017 neurons to neurophysiological and behavioral data that have been previously collected by Alexander and Nitz (Alexander and Nitz, 2015). In neuroscience models, this topology is often loosely, manually specified based on the known, somewhat incomplete properties of a real structure in the brain. In the present case, we begin with a pre-specified network topology that defines the coarse-grained connectivity structure among several groups of neurons (Figure 1.1). The goal of parameter tuning is to adjust the details of the network (such as synaptic weights, the number of connections between groups,
and/or the behavioral parameters of the neurons in each group) such that the network successfully produces the desired target behavior.

**Retrosplenic Cortex Dataset and Experimental Task**

We used an electrophysiological dataset collected from 243 retrosplenic cortex (RSC) neurons (228 neurons after excluding head direction cells) in six male Long-Evans rats (Alexander and Nitz, 2015). The rats performed a spatial navigation task that involved completing inbound and outbound routes along a W-shaped track that involved either a left-right-left (LRL) or right-left-right (RLR) turn sequence. Four behavioral variables were recorded concurrently with neural activity as the rats performed the task: 1) head direction (HD), 2) allocentric position in x-y coordinates (Pos), 3) angular velocity (AV), and 4) linear velocity (LV). In each of the 71 recording sessions, rats were required to complete a series of inbound and outbound runs on the track that occupied two separate allocentric locations in the room (α and β). In each recording session, the track was placed in a different location. The rats collectively completed an average of 16 trials on each possible route (αLRL, αRLR, βLRL, βRLR) across all sessions. To analyze neural activity, the W-shaped track was divided into 200 equally sized bins, each approximately 1.2 cm wide. These bins were used to determine the precise points at which a rat reached a turn apex on each run through the track.

**Modeling of Behavioral Metrics**

Each SNN contained three groups of neurons (Figure 1.1): 417 excitatory input neurons, which handled the encoding of the behavioral inputs; 480 regular-spiking excitatory Izhikevich neurons and 120 fast-spiking inhibitory Izhikevich neurons (Izhikevich, 2003). The network had four types of connections: inputs to excitatory
(Inp→Exc), inputs to inhibitory (Inp→Inh), recurrent excitatory (Exc↔Exc), and inhibitory to excitatory (Inh→Exc). All synaptic projections were random with a 10% chance of connectivity. No topology was enforced. The behavioral inputs were parameterized by creating idealized input neurons using broadly tuned Gaussian and cosine tuning curves. By examining the range of values present for each variable in the dataset, we created an appropriate number of tuning curves to encompass the input space.

To create inputs for linear and angular velocity, the following Gaussian tuning curve was used (Eq. 1):

\[
f(n,b) = r_{max} \cdot e^{-\left(\frac{(input_b - pref_n)^2}{2\sigma^2}\right)}, \quad b = 1:200 \text{ and } n = 1:N
\]  

(1)
For linear velocity, the parameters were as follows: \( N = 7 \), \( r_{\text{min}} = 0 \), \( r_{\text{max}} = 60 \), and \( \sigma = 7 \). For angular velocity, \( N = 12 \), \( r_{\text{max}} = 40 \), and \( \sigma = 12 \).

To create inputs for head direction, the following cosine tuning curve was used (Eq. 2):

\[
f(n, b) = \begin{cases} 
  r_{\text{max}} \cos \left( \frac{\pi x}{360} \left( \text{input}_b - \pi \right) \right) - \left( \frac{\pi x}{360} \left( \text{pref}_n - \pi \right) \right) & \text{if } \text{input}_b \geq (\text{pref}_n - 45) \\
  0, & \text{else}
\end{cases} \quad (2)
\]

\[
b = 1:200 \text{ and } \text{num} = 1:N
\]

where \( N = 8 \), \( x = 3.99 \), \( r_{\text{max}} = 40 \), and \( \sigma = 8 \).

For allocentric position, which was tracked in Cartesian coordinates, a 2D Gaussian curve was used to create idealized place cells by adapting the equation from Foster et al. (Foster, Morris and Dayan, 2000) (Foster, Morris and Dayan, 2000) (Foster, Morris and Dayan, 2000) (Foster, Morris and Dayan, 2000) (Foster, Morris and Dayan, 2000) (Foster, Morris and Dayan, 2000) (Foster, Morris and Dayan, 2000) (Foster, Morris and Dayan, 2000) (Foster, Morris and Dayan, 2000) (Foster, Morris and Dayan, 2000) (Foster, Morris and Dayan, 2000) (Foster, Morris and Dayan, 2000) (Foster, Morris and Dayan, 2000) (Foster, Morris and Dayan, 2000) (Foster, Morris and Dayan, 2000) (Foster, Morris and Dayan, 2000) (Eq. 3):

\[
f(n, b) = r_{\text{max}} \times e^{-\frac{(\text{location}_b - \text{pref}_n)^2}{2\sigma^2}}, \quad b = 1:200 \text{ and } n = 1:N \quad (3)
\]

where \( N = 390 \), \( r_{\text{max}} = 60 \), and \( \sigma = 40 \). The variables location and pref are x-y coordinate pairs. Location represents the actual position of the agent and pref represents the center of the place cell.
To train the network, a learning rule known as STDP was used to update the weights of the network (Bi and Poo, 1998); specifically, a standard nearest-neighbor implementation (Izhikevich and Desai, 2003). Homeostatic synaptic scaling was incorporated into the STDP rule to keep the neuronal firing rates within a reasonable regime by scaling to a target firing rate (for more details see Carlson et al. (2013) (Carlson, Richert, Dutt and Krichmar, 2013)).

**Parameters and Training**

The automated tuning framework was used to evolve a total of 18 parameters, which were related to plasticity, overall firing rates and weight ranges (Table 1.1). These parameters were used as inputs to the CARLsim GPU-based simulation framework, which we used to run the SNN models 3,5. Twelve parameters related to STDP were evolved, which correspond to three types of STDP curves ($A^+$ and $A^-$, which dictate the change in a synaptic weight value; and $\tau^+$ and $\tau^-$, which correspond to the critical windows over which spike times are integrated in milliseconds). The remaining parameters control the homeostatic target base firing rates for the excitatory and inhibitory populations, and the initial and maximum weight values for each set of inter-group connections.

**Table 1.1 Parameters Initialized Via the ECJ Framework.**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>$A^+$</th>
<th>$A^-$</th>
<th>$\tau^+$</th>
<th>$\tau^-$</th>
<th>Base FR (exc)</th>
<th>Base FR (inh)</th>
<th>Inp-Exc Init.</th>
<th>Inp-Inh Init.</th>
<th>EE Init.</th>
<th>IE Init.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimum</td>
<td>-0.0002</td>
<td>-0.0002</td>
<td>5.0</td>
<td>5.0</td>
<td>5.0</td>
<td>5.0</td>
<td>0.01</td>
<td>0.01</td>
<td>0.001</td>
<td>0.001</td>
</tr>
<tr>
<td>Maximum</td>
<td>0.004</td>
<td>0.004</td>
<td>100.0</td>
<td>100.0</td>
<td>20.0</td>
<td>20.0</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
</tr>
<tr>
<td>Std.Dev</td>
<td>-0.00042</td>
<td>-0.00042</td>
<td>9.5</td>
<td>9.5</td>
<td>1.5</td>
<td>1.5</td>
<td>0.049</td>
<td>0.049</td>
<td>0.0499</td>
<td>0.0499</td>
</tr>
</tbody>
</table>

Following evolution, we examined the parameter solutions found by the evolutionary algorithm for each run to investigate whether the values between runs tended to fall in similar parts of the parameter space, or if there were multiple possible solutions to
the problem. To detect outliers, we used the Generalized Extreme Studentized Deviate (G-ESD) test to find parameter values that fell outside of two standard deviations of the mean, using \( r = 4 \) (maximum expected number of outliers) for ten total samples. Over 10 networks, with 18 evolved parameters per network, we only found two outliers, one which corresponded to the \( A^+ \) parameter for E-STDP on the inhibitory neurons, and another which corresponded to the initial weight value of the exc↔exc connections (Supplementary Table 1.2). Otherwise, the evolved parameters seemed to cluster closely around the mean, suggesting that the solutions found for each of the ten networks were similar.

### Table 1.2 Evolved Solutions and Uniqueness Over Ten Runs

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Mean</th>
<th>Std. Dev</th>
<th>Range</th>
<th>Outliers (( p &lt; 0.05 ))</th>
</tr>
</thead>
<tbody>
<tr>
<td>E-STDP ( A^+ ) (exc)</td>
<td>0.0018</td>
<td>0.001</td>
<td>0.0033</td>
<td>-</td>
</tr>
<tr>
<td>E-STDP ( A^- ) (exc)</td>
<td>0.0014</td>
<td>0.0011</td>
<td>0.0037</td>
<td>-</td>
</tr>
<tr>
<td>E-STDP ( \tau^+ ) (exc)</td>
<td>40.1161</td>
<td>24.7447</td>
<td>71.1596</td>
<td>-</td>
</tr>
<tr>
<td>E-STDP ( \tau^- ) (exc)</td>
<td>63.3515</td>
<td>18.9186</td>
<td>67.8766</td>
<td>-</td>
</tr>
<tr>
<td>I-STDP ( A^+ ) (exc)</td>
<td>0.0025</td>
<td>0.00075</td>
<td>0.002</td>
<td>-</td>
</tr>
<tr>
<td>I-STDP ( A^- ) (exc)</td>
<td>0.0024</td>
<td>0.0008</td>
<td>0.0026</td>
<td>-</td>
</tr>
<tr>
<td>I-STDP ( \tau^+ ) (exc)</td>
<td>47.3281</td>
<td>17.2909</td>
<td>55.7028</td>
<td>-</td>
</tr>
<tr>
<td>I-STDP ( \tau^- ) (exc)</td>
<td>45.9963</td>
<td>19.1325</td>
<td>60.3811</td>
<td>-</td>
</tr>
<tr>
<td>E-STDP ( A^+ ) (inh)</td>
<td>0.0023</td>
<td>0.0011</td>
<td>0.004</td>
<td>1 (-0.0002)</td>
</tr>
<tr>
<td>E-STDP ( A^- ) (inh)</td>
<td>0.0016</td>
<td>0.0011</td>
<td>0.0031</td>
<td>-</td>
</tr>
<tr>
<td>E-STDP ( \tau^+ ) (inh)</td>
<td>46.4212</td>
<td>22.5365</td>
<td>66.0629</td>
<td>-</td>
</tr>
<tr>
<td>E-STDP ( \tau^- ) (inh)</td>
<td>52.1326</td>
<td>27.1257</td>
<td>83.89</td>
<td>-</td>
</tr>
<tr>
<td>Base FR (exc)</td>
<td>14.6148</td>
<td>3.6604</td>
<td>10.0567</td>
<td>-</td>
</tr>
<tr>
<td>Base FR (inh)</td>
<td>13.3565</td>
<td>4.4315</td>
<td>12.5183</td>
<td>-</td>
</tr>
<tr>
<td>Inp-Exc</td>
<td>0.3697</td>
<td>0.0903</td>
<td>0.2441</td>
<td>-</td>
</tr>
<tr>
<td>Inp-Inh</td>
<td>0.1251</td>
<td>0.1417</td>
<td>0.3539</td>
<td>-</td>
</tr>
<tr>
<td>Exc-Exc</td>
<td>0.0635</td>
<td>0.1269</td>
<td>0.3830</td>
<td>1 (0.3846)</td>
</tr>
<tr>
<td>Inh-Exc</td>
<td>0.1959</td>
<td>0.1805</td>
<td>0.4394</td>
<td>-</td>
</tr>
</tbody>
</table>

A total of 4,593 trials, each of which consisted of a route (\( \alpha \)LRL, \( \alpha \)RLR, \( \beta \)LRL, and \( \beta \)RLR), were conducted and recorded over the course of all 71 recording sessions in the
original study conducted by Alexander and Nitz (Alexander and Nitz, 2015). This gave us a sizeable pool of data on which to train and test the SNNs. To establish the pools of training and testing trials, we took the number of trials for each route combination in each recording session and divided them in half to create the training and testing sets. The training and testing paradigm consisted of randomly choosing 150 trials for each route combination (600 trials in total; 150 * 4) in each phase that were presented to the network (Rounds, Scott, Alexander, DeJong, Nitz and Krichmar, 2016). To ensure that the experimental design was as similar as possible to the original experiment, trials were presented in terms of alternating routes. For example, if the first trial presented was selected from αLRL, then the following trial would be selected from αRLR, which would be repeated 10 times, and then the experiment would switch to track β. Following training, synaptic plasticity was disabled while the network was exposed to the testing set. Following testing, fitness was evaluated and the generation ended, at which point a new generation of individuals was initialized via the PTI. This process continued for 50 generations per each evolutionary run. Ten evolutionary runs were conducted to obtain a set of high fitness individual networks for subsequent analysis.

Following testing, the population was evaluated by summing the best correlations between the experimentally observed and simulated neurons for each SNN. The best correlations were found by first correlating every simulated neuron \((n = 600)\) against every experimentally observed neuron \((n = 228)\). Next, a match was chosen based on highest correlation value between each experimentally observed neuron and the corresponding simulated neuron (a neuron could only be chosen once). After all experimentally observed neurons had a match, the fitness score for that individual SNN was computed by summing
the correlations ρ between each pair (Equation 1.1). A maximum mean firing rate threshold was also incorporated into the fitness function to ensure that simulated firing rates were reasonable and realistic. The firing rate of each neuron in the network was averaged across all trials, and the highest observed value was considered the maximum mean. If the observed maximum mean firing rate maxFR exceeded the threshold, then the fitness score was penalized by subtracting the difference between the threshold and the observed firing rate (Equation 1.2):

\[
f(x) = \begin{cases} 
\sum_{i=1}^{n} \rho(\text{realFR}_i, \text{synFR}_{\text{match}}) & \text{if maxFR} < \text{FR}_{\text{target}}, \\
\sum_{i=1}^{n} \rho(\text{realFR}_i, \text{synFR}_{\text{match}}) - \text{FR}_{\text{error}} & \text{otherwise}
\end{cases}
\]

\[
\text{FR}_{\text{error}} = \text{FR}_{\text{max}} - \text{FR}_{\text{target}},
\]

and \( \text{FR}_{\text{target}} = 250 \text{ Hz} \) was the maximum mean firing rate allowed for any neuron.

After a generation, the fitness scores were sent to ECJ via the PTI for evaluation and constructing a new population (Figure 1.2). The simulations proceeded for 50 generations. The complete process was repeated 10 times to ensure repeatability. It is important to reiterate that the use of GPU processing speeds up the fitness function significantly. In this case, the fitness function runs 136,800 Pearson’s \( r \) correlations (600 synthetic neurons multiplied by 228 neurophysiological neurons) per each individual, which is computationally very expensive. This complexity could increase considerably with the size of the dataset being replicated, the size of the network being run, and/or the number of individuals in the population, making it very important that the fitness function can be calculated in parallel on GPU.
Evolutionary Algorithm

We represented the parameter space of the RSC model as vectors in $\mathbb{R}^{18}$, and then applied a $(\mu + \lambda)$-style, overlapping-generations EA with truncation selection to maximize $f(x)$. We used a mutation operator that takes each parameter and adds 1-dimensional Gaussian noise with probability 0.5. The width of the Gaussian mutation operator was fixed at 10% of the range that each parameter was allowed to vary within. The values of $\mu$

![Figure 1.2 CARLsim-ECJ Pipeline for Evolving SNNs. General approach to evolving SNNs using ECJ. ECJ is used to initialize the parameters being tuned, which are passed to CARLsim and used to set the parameters of each network in the population of SNNs (red arrow). The SNNs are initialized, run, and evaluated using CARLsim. Following fitness evaluation using some user-defined criteria, the resulting fitness values for each individual in the population are sent to ECJ (black arrow). ECJ chooses parents from the population and performs selection, replication, and mutation on the parameters of the parent individuals in order to initialize the next generation of SNNs. This continues until an ideal solution is found or until a user-defined number of evolutionary runs are completed. Adapted from Beyeler et al. (2015).]
and $\lambda$ were fixed at 3 and 15, respectively. It was straightforward to combine the SNN simulator with the ECJ evolutionary computation system (White, 2012) to create a unified parameter-tuning framework.

These decisions result in an algorithm with a small population and a strong selection pressure. This simple EA proved sufficient for our purpose, which is provide a proof of the feasibility of evolving SNNs with an STDP-based indirect encoding. We leave the problem of customizing EA design decisions to maximize performance for future work.

**Results**

**Fitness Values and Firing Rate Correlations**

Each of the 10 independent runs of the EA were executed for a small number of generations. Thanks to the indirect encoding, the best fitness found tended to be very high after just 50 generations (Figure 1.3) with a mean of 105.93 ± 0.91. The highest observed fitness was 107.79. A total of 228 experimentally correlated neurons were matched, thus the average firing rate correlation was about 0.47 per neuron. The lowest observed fitness was 104.7, resulting in a correlation of about 0.46 per neuron (strong correlations by experimental standards). At the start of each evolutionary run, the average maximum fitness score was 84.57 ± 19.78.

**Table 1.3 Average Runtimes in Minutes (Mean/Std. Dev).**

<table>
<thead>
<tr>
<th>Generation</th>
<th>1</th>
<th>10</th>
<th>20</th>
<th>30</th>
<th>40</th>
<th>50</th>
</tr>
</thead>
<tbody>
<tr>
<td>Testing</td>
<td>126.28/39.51</td>
<td>42.43/40.55</td>
<td>42.91/30.89</td>
<td>39.39/28.34</td>
<td>32.1/17.32</td>
<td>39.65/37.13</td>
</tr>
<tr>
<td>Total</td>
<td>240.48/59.24</td>
<td>64.42/59.57</td>
<td>65.3/50.4</td>
<td>59.94/45.25</td>
<td>50.66/28.38</td>
<td>60.91/49.86</td>
</tr>
</tbody>
</table>

Each of the ten evolutionary runs took 3.13 ± 1.26 days to complete. A breakdown of how long a generation took can be seen in Table 1.3. In the beginning, the population ran very slowly, taking approximately four hours to complete (slightly under two hours for
training, and slightly more for testing). By the tenth generation, the population took roughly an hour to complete, which stayed relatively constant across the remaining generations (breaking down to about 20 minutes for training and 30 for testing). However, there was considerable variance in how long a generation could take at each point (each generation had a standard deviation of about one hour) because of the different firing rates of individual SNNs. Although the fitness increased during the evolutionary run, the selection strategy tended to include high and low firing SNNs in the population, which affects runtime performance in CARLsim.

The tuning framework could closely match experimental neural activity to simulated neural activity. Figure 1.4 shows twelve representative examples of matched neurons with high correlations. Note that the correlation values in the figure are not much

**Figure 1.3. Fitness Curve.** Network fitness rapidly and consistently converged over 50 generations.
higher than the average correlation value, suggesting that they are typical examples of matched neurons indicative of the network’s overall fitness. Thus, the EA was able to generate networks whose neuronal firing rates were able to match those of the experimental dataset.

**Replicating Empirical Data**

Similar to RSC neurons recorded in behaving animals, the evolved networks effectively encoded all three spatial frames of reference (allocentric, egocentric, and route-centric). Figure 1.4 shows one example for each functional neuron type where the black trace is from the biological neuron and the red trace is from the simulated neuron. A ‘turn insensitive’ neuron (Figure 1.4a-d) is not tied in its firing to any particular action correlate, but nevertheless reliably generates a complex spatially-specific firing pattern. A subset of turn insensitive cells encode position with respect to route, evidenced by high correlations between firing rate profiles associated with the LRL/RLR route on both track positions $\alpha$ and $\beta$ (Figure 1.4c, d). In contrast, purely ‘turn sensitive’ neurons (Figure 1.4e-h) respond primarily to a preferred turn type (in this case, left turns), regardless of where such actions occur along a route. Between these extremes, some ‘turn sensitive’ neurons that have spike rates specific to route position (Figure 1.4i-l) also respond to specific turning actions. In such cases, higher firing rates are found for a specific instance along a route (e.g., Figure 1.4l; higher firing rates for the first right turn on the RLR route and the first left of the LRL route).

The evolved networks were able to replicate all observed features of the RSC data. For example, each network had a similar distribution of functional neuron types to those reported in Alexander and Nitz (Alexander and Nitz, 2015) where RSC firing according to
Figure 1.4. Functional Neuron Type Examples. Example matched neurons taken from a representative network. The original experimentally observed firing rate profile is shown in blue, while the matched synthetic neural activity is shown in red. (a-b) Turn insensitive neurons with no modulation by route. (c-d) Turn insensitive neurons whose activity is modulated by route (α and β LRL (RLR) firing rate profiles are highly correlated). (e-f) Turn sensitive neurons that prefer left turn instances with no route modulation, meaning that they respond roughly equally to all instances of left turns. (g-h) Turn sensitive neurons that prefer right turns, without route modulation. (i-j) Left turn sensitive neurons with route modulation, meaning that they consistently respond more to the first or second left turn on an LRL turn sequence. (k-l) Right turn sensitive neurons with route modulation.
route position, turning action, environmental position, and conjunctions of these variables were observed (Figure 1.5a). Neurons from each network were categorized into functional types, and then the numbers of neurons of each type were averaged for ten different network instantiations. About 64% of the neurons in each network ($\bar{n} = 360.5 \pm 26.12$ out of $575.3 \pm 15.57$) were considered turn sensitive, as determined by comparisons of mean firing rates for all left vs. right turn instances (Wilcoxon rank-sum test, $P < 0.05$). The remaining 37% of the neurons in each network ($\bar{n} = 214.8 \pm 33.05$ out of $575.3$) were considered turn insensitive (Wilcoxon rank-sum test, $P > 0.05$). Additionally, a subset of turn insensitive neurons ($\bar{n} = 42.8 \pm 8.97$ out of $214.8$) exhibited 'route-modulated' activity in that their mean firing rate profiles for same-route comparisons were highly correlated even when the route was moved to different locations in the larger environment (e.g., firing pattern for $\alpha$LRL correlated with firing pattern for $\beta$LRL, Pearson’s $r > 0.4$, $P < 0.05$). Turn sensitive neurons were also divided into two groups, in which approximately half were purely turn sensitive, in that they responded to all instances of the preferred turn equally ($\bar{n} = 179.8 \pm 48.36$ out of $360.5$), or were route-modulated in that they preferred either the first or second instance of the preferred turn along the route regardless of the allocentric position of the track ($\bar{n} = 180.7 \pm 35.07$ out of $360.5$). This was determined by comparing all mean firing rates associated with the first left or right turn on the route with those associated with the second left or right turn on the route for each neuron. Those with significant differences associated with the first vs. the second turn were considered to exhibit route modulated activity (Wilcoxon rank-sum, $P > 0.05$), while those without were considered purely turn sensitive (Wilcoxon rank-sum, $P < 0.05$). The former encode conjunctions of route position and associated action, effectively mapping the spatial
context within which particular navigational behaviors take place. The majority of the cells in the evolved networks were turn insensitive (i.e., turn insensitive, no mod. and turn insensitive, route mod. in Figure 1.5a), which was consistent with the dataset (see Figure 3a of Alexander and Nitz). Finally, we also found similar distributions of head direction cells in the evolved networks. On average, 4.12%, ± 2.6% ($\bar{n} = 24.7$, ± 15.57) of the neurons in each evolved network were head direction cells despite the fact that head direction tuned neurons were not included during model evolution. The percentage is slightly lower than that reported in Alexander and Nitz (Alexander and Nitz, 2015) (6%) and Cho and Sharp (Cho and Sharp, 2001) (10%), highlighting the ability of the approach to generate emergent forms of spatial information encoding in spiking activity.

To ensure that turn preference in the turn sensitive neuron population was not the product of chance, we computed a true and randomized turn bias ratio for each turn

![Figure 1.5. Functional Neuron Type Distributions. Results for averaged evolved SNNs corresponding to the original experiment. (a) Pie chart showing the average neuronal type distributions for turn sensitive and turn insensitive neurons. We add a segment for turn insensitive, route modulated neurons. (b) Random (red line) and non-random (blue line) cumulative distribution functions (cdfs) are shown for the turn bias ratios associated with all turn sensitive neurons in the population (averaged over all SNNs). The cdfs demonstrate turn-related biases in activity that are significantly higher than expected by chance in the non-randomized cdf.](image)
sensitive neuron. This was accomplished by creating a turn identity matrix in which the average firing rate was calculated for all instances of all turn types (twelve turns in total, six left and six right), and then the ratio of averaged activity for left versus right turns was taken. The turn identity matrices were subsequently randomized over 25 iterations and recomputed to create a randomized cumulative distribution function (red line in Figure 1.4b) that was plotted against the actual cumulative distribution function of turn bias ratios (blue line in Figure 1.5b). The true distribution of turn bias ratios for all turn sensitive synthetic cells was significantly greater than the distribution of ratios calculated by randomizing turn identity (25 of 25 iterations, one-tailed Kolmogorov-Smirnov, \( \overline{n} = 360.5, D > 0.4, P < 0.0001 \)).

**Replicating Population Behavior**

Similar to Alexander and Nitz (2015), we verified that the evolved networks could reconstruct the agent’s position along a route from positional ensemble correlations across the whole neuron population. We created positional ensemble reconstruction matrices following methods described in the original paper (Alexander and Nitz, 2015). This involved creating an average firing rate profile for every neuron based on even trials, odd trials, and all trials. For within-position comparisons (for example, position \( \alpha \), route LRL even vs. position \( \alpha \), route LRL odd) even and odd average firing rate profiles for all neurons were cross-correlated to generate the reconstruction matrix (Figure 1.6a, b). Neuronal activations that occur at the same points in route space should be highly correlated, while those that occur at different points should not be. This creates an ideal prediction line that crosses the matrix diagonally from the left top corner to right bottom (dotted white line). A black line was overlaid on the reconstruction matrix that falls along the points of actual
Figure 1.6. Reconstruction of Position from Ensemble Correlations. (a) A positional ensemble reconstruction matrix that shows the correlations for ensemble neural activity between even and odd trials for the αLRL route. The matrices for all ten evolved networks were averaged to create a representative visualization, which resulted in low reconstruction error. Adapted from Rounds et al. (2016). (b) A positional ensemble reconstruction matrix showing the correlations for ensemble neural activity between all trials associated with the αLRL and βLRL routes. The matrices for all ten evolved networks were again averaged to create a representative visualization. Adapted from Rounds et al. (2016). (c) Bar plot showing average reconstruction error for all positions and routes (arbitrary units). Reconstruction error for each of the networks was calculated by subtracting the theoretical perfect prediction line from the actual points of maximum correlation for each row and column on the matrix. The resulting vectors produced for every network were then averaged to obtain the average per-bin reconstruction error. The resulting vector was then averaged to get the overall average reconstruction error. Average error was extremely low (near zero) for reconstructions between even and odd trials for the same position and route (first four bars on plot), while average error was comparatively quite high for reconstructions between tracks α and β for both the LRL and RLR routes (fifth and sixth bars on plot). Significance bars show that reconstruction error between α and β was significantly higher for both routes than reconstruction error between same positions and routes at $P < 0.0001$ (as revealed by a Kruskal-Wallis test and Tukey’s HSD post-hoc comparison) Asterisks denote level of significance ($* = P < 0.05; ** = P < 0.01; *** = P < 0.001$).
highest correlation between bins. When even vs. odd trials were compared, the black line fell along the white prediction line (Figure 1.6a). However, when reconstruction matrices were generated by comparing trials recorded for track position α along the LRL route to trials recorded for track position β along the LRL route, the black reconstruction line did not follow closely along the white prediction line (Figure 1.6b). The reconstruction error was calculated by subtracting the black line from the white line and averaging the difference for each of the 200 bins along the track. Reconstruction errors were significantly higher for α/β reconstructions, compared to αEven/αOdd and βEven/βOdd reconstructions (Figure 1.6c), demonstrating that the network can distinguish similar routes that occur in different allocentric positions (Kruskal-Wallis, \( \chi^2(5) = 750.96, P < 0.0001 \)). This demonstrates that the network was also sensitive to position within the larger environment (a.k.a., the allocentric frame of reference) in addition to a route-centric frame. This finding is consistent with the original result; for further details, see Alexander and Nitz (Alexander and Nitz, 2015).

**Discussion**

In this chapter, I introduced an automated tuning framework that leverages the search power of evolutionary algorithms combined with the parallelization of GPUs, which can result in a speedup of up to 60 times faster than a CPU in CARLsim (Beyeler, Carlson, Chou, Dutt and Krichmar, 2015). This results in an efficient method for searching the SNN parameter space by drastically reducing its dimensionality via an indirect encoding scheme in which a learning rule, STDP, was used to specify the synaptic weights of each network. Performing fitness evaluation on each network in parallel further reduced the time necessary to tune the SNNs, even though every individual in the population was subjected
to both a training and a testing phase. We successfully applied this framework to a case study in which it was used to evolve a model of the brain region RSC using electrophysiologically recorded neurons. Rather than altering the synaptic weights of each SNN directly, an evolutionary algorithm was used to alter the learning parameters of each SNN until a close match between synthetic and recorded neuronal firing rates was found, which resulted in a reduction of the number of parameters to be tuned from thousands to only 18. Furthermore, the evolutionary algorithm took only 50 generations to converge, demonstrating the framework was able to efficiently evolve a solution. This is in stark contrast to direct encoding methods of evolving neural networks, which can take thousands of generations to converge (Asher, Krichmar and Oros, 2015).

The phenomenological results of this case study suggest that the approach of using STDP as an indirect encoding scheme will generalize to other types of SNN tuning problems, and can be used to match other neurophysiological datasets, since many electrophysiological recordings are collected under conditions similar to the present dataset. First, the SNNs successfully captured the underlying network activity, which was reflected in the fitness score of each evolved network. Secondly, the SNNs captured neuronal function observed in the data, which was reflected in empirically observed distributions of non-route modulated turn-sensitive neurons and route modulated turn-sensitive neurons, respectively. Thirdly, the ensemble activity of the synthetic neurons captured behavioral functionality, such as position and route reconstruction.

The capacity to efficiently synthesize networks that reproduce neuron and network functionality across these three levels is of considerable importance as we attempt to move toward a greater understanding of brain function. We have demonstrated that we have
created a powerful tool with this capacity by applying our framework to this case study of the RSC, which may be applied to a variety of modeling efforts and tuning problems involving SNNs. Further experiments are underway to investigate how the network responds to manipulations of its inputs, and to predict how neural activity in the retrosplenial cortex might change depending on environmental context. These predictions can then be tested by conducting new electrophysiological experiments, the results of which could lead to a better understanding of how neural responses give rise to behavior.
References


CHAPTER 2: Development of a Virtual Testbed

Introduction

Although a framework for the replication of neurophysiologically observed neuronal firing patterns adds another layer of verification to computational models of cognition, the models must still be validated against experimental work. Thus, I developed a virtual testbed to design and conduct novel simulated experiments on the evolved models. The virtual testbed is an important step toward the paramount goal of aiding collaboration between theoretical and experimental neuroscientists, since the virtual testbed can be used to quickly generate and test hypotheses about the function of neural circuitry. Experimentalists expend tremendous time and resources designing experiments that ultimately may not reveal anything new about the brain region being studied, but a

Figure 2.1. Experimental-Computational Pipeline. Experimental neuroscientists produce electrophysiologically recorded datasets. Collaborators in computational neuroscience apply the evolutionary framework in Chapter 1 to the dataset to evolve a model of the brain region. The virtual testbed is used to generate hypotheses about the nature of representation and function in the brain region, which can be verified by experimentalists in animal experiments. Eventually, after a sufficiently valid model has been produced, the SNN could be implemented in hardware to create neuroprosthetics.
virtual testbed for analyzing networks tuned to match datasets may be able to indicate which experiments are most likely to yield new and important insights. On the theoretical side, experimentalists can test and validate predictions generated through the use of simulated experiments, which helps theoretical neuroscientists to further refine their models. We hope that this framework may enable a feedback loop between experimental and theoretical neuroscientists (Figure 2.1). It is also possible that this framework can aid in the design and development of models that can be implemented in hardware, which may be used for neuroprosthetics that can replace lost or damaged brain tissue. In this chapter, we also demonstrate that this extension to the framework can be invaluable for the understanding of underlying neuronal representations. We applied the virtual testbed to the evolved models of retrosplenial cortex, which underwent a number of experimental manipulations, including lesion and track manipulation studies that provide valuable new insights into the retrosplenial cortical representation of spatial, navigation-related cues.

**Experimental Task and Dataset**

We used an electrophysiological dataset collected from 243 retrosplenial cortex (RSC) neurons (228 neurons after excluding head direction cells) in six male Long-Evans rats (Alexander and Nitz, 2015). The rats performed a spatial navigation task that involved completing inbound and outbound routes along a W-shaped track that involved either a left-right-left (LRL) or right-left-right (RLR) turn sequence. Four behavioral variables were recorded concurrently with neural activity as the rats performed the task: 1) head direction (HD), 2) allocentric position in x-y coordinates (Pos), 3) angular velocity (AV), and 4) linear velocity (LV). In each of the 71 recording sessions, rats were required to complete a series of inbound and outbound runs on the track that occupied two separate
allocentric locations in the room ($\alpha$ and $\beta$). In each recording session, the track was placed in a different location. The rats collectively completed an average of 16 trials on each possible route ($\alpha$LRL, $\alpha$RLR, $\beta$LRL, $\beta$RLR) across all sessions. To analyze neural activity, the W-shaped track was divided into 200 equally sized bins, each approximately 1.2 cm wide. These bins were used to determine the precise points at which a rat reached a turn apex on each run through the track.

**CARLsim**

Computer simulations were performed by using the CARLsim spiking neural network (SNN) simulator (Beyeler, Carlson, Chou, Dutt and Krichmar, 2015). CARLsim is a C/C++ based simulator that allows the user to construct biologically detailed spiking networks that may be specified at the synapse, neuron, and network level. In the present study, we used CARLsim version 3.0, which included excitatory spike-timing dependent plasticity (STDP), inhibitory STDP, and a homeostatic mechanism for synaptic scaling to prevent runaway activity (Carlson, Richert, Dutt and Krichmar, 2013). Both STDP and homeostasis act on the post-synaptic neuron. The CARLsim simulator is GPU-accelerated using off-the-shelf NVIDIA GPUs, which allows detailed large-scale networks to execute rapidly and efficiently and allows multiple SNNs to be run in parallel.

**Examining the Impact of Lesions on Network Performance**

To test the conjunctive nature of RSC activity, we simulated lesions of projections from one neural group to another by setting the synaptic weights of those synaptic connections to zero. These simulated lesions were conducted on the SNN with the highest fitness from an evolutionary run. The lesioned SNN underwent a new testing and fitness evaluation phase. The recorded behavioral inputs were presented to each lesioned network
again in the same order so as to ensure consistency between the baseline and lesioned conditions. The individual’s performance under each lesion condition was then compared to its intact performance in terms of its fitness, its ability to reconstruct a route, and the distributions of functional neuron types.

**Verification of Complete Model Functionality**

We ensured that the model did not function solely due to the presence of optimized neuronal responses in the network. Because the network contains 600 output neurons, but only 228 neurons were matched to electrophysiological recordings, it is conceivable that the network would cease to function correctly if the subset of matched neurons were removed from the network. Thus, we excluded that subset of neurons and re-evaluated the remaining population in terms of functional neuron type distributions, statistically significant turn bias-related activity, and population functionality.

**Rearranging Turn Sequences**

To ensure that the network had learned to respond generally to features of a route and not to the specific shape of the route it was trained on, we tested the network with different turn sequences. This was achieved by selecting a recording session and altering the behavioral variables such that the movement would correspond to this new track shape. This was done by rearranging the structure of the track (essentially by switching out bin segments in order to reflect the desired shape). We altered the trials corresponding to three separate recording sessions to obtain six possible alternate turn sequences: LLL (outbound)/RRR (inbound); LLR (outbound)/LRR (inbound); and RLL (outbound)/RRL (inbound) (Figure 2.2). For LLL/RRR, 10 new trials were constructed and presented to each evolved network; for LLR/LRR, 13 new trials were constructed, and for RLL/RRL, 9
new trials were constructed. Network performance was evaluated on fitness, ability to reconstruct the route, and functional neuron type distribution, as with the lesioned networks.

**Rotating Track Positions**

To systematically examine network responses to changing environments, we rotated the W track. Trials corresponded to five recording sessions where we imposed a gradual rotational shift of the track. Each track was shifted around its center point, or the middle turn of the track, in 20° increments. This resulted in 18 positions for each track, including the original position (Figure 2.3a-e) α and β. Positions α and β occupy disparate allocentric positions depending on the recording session. In recording session 1, the tracks

---

**Figure 2.2. Schematic of Rearranged Turn Sequences.**

- **a)** Schematic of the LLL/RRR turn sequence rearrangement. Tracks α (black line) and β (green line) were rearranged such that the outbound route the agent took required the agent to perform three left turns (LLL). The reversed inbound route required three sequential right turns (RRR). Turn locations are marked with diamonds.

- **b)** Schematic of the LLR/LRR turn sequence rearrangement. Tracks α (black line) and β (green line) were rearranged such that the outbound route the agent took required the agent to perform three left turns (LLR). The reversed inbound route required three sequential right turns (LRR). Turn locations are marked with diamonds.

- **c)** Schematic of the RRL/RLL turn sequence rearrangement. Tracks α (black line) and β (green line) were rearranged such that the outbound route the agent took required the agent to perform three left turns (RRL). The reversed inbound route required three sequential right turns (RLL). Turn locations are marked with diamonds.
overlap at their center point, meaning that at some point during rotation, it is possible for the tracks to occupy the same approximate allocentric position and orientation (Figure 2.3a). They are also situated close to one another at their center points in recording session 3 (Figure 2.3b). In recording sessions 2 and 4, their center points are relatively far apart, but are close enough to one another that at some point, track segments of \( \alpha \) and \( \beta \) will be overlapping, but not such that similar turns occupy similar allocentric positions (Figure 2.3c, d). In the case of recording session 5, the tracks are entirely non-overlapping (Figure 2.3e). These distances allowed us to investigate network responses under a range of conditions (overlapping, semi-overlapping, parallel but non-overlapping, etc.). In order to create these new routes, two behavioral variables were manipulated (Pos and HD). To dissociate differences in network performance on the fitness metric due to changing one behavioral variable (translation case) or two behavioral variables (orientation case), additional track orientation experiments were carried out in which the head direction was not shifted to be consistent with the new allocentric position. This led to another set of orientation experiments (orientation, fixed HD) in which head direction was always consistent with the original position of the track (position 1, or 0° shift, out of the 18 rotated positions).

**Linearly Translating Track Positions**

To systematically examine network responses to changing allocentric position, we translated the position of the W track. Five separate recording sessions were altered to reflect gradual movement of the track position. The recording sessions were chosen such that the original track positions were parallel to one another but occupied different allocentric positions (with varying amounts of distance between them). For each recording
session, this resulted in trials that corresponded to six track positions in which the track shifted in consistently sized steps to a final position (Figure 2.3f-j). The step size was consistent for all recording sessions - all tracks were moved in increments of 32 centimeters. In order to translate track locations, the x-y coordinates corresponding to allocentric position (Pos) were modified. Thus, HD, AV, and LV remain consistent with the original position of the track (track position 1 out of the 6 possible positions).

**Figure 2.3. Schematics of Track Orientation and Linear Translation.** (a-e) We manipulated all trials in five separate recording sessions to simulate track rotation in $20^\circ$ increments around the center point of the track. (f-j) We manipulated the behavioral position data associated with a separate five recording sessions to simulate movement of each track position linearly across the allocentric space in 32 cm increments. This manipulation did not require altering head direction. These experiments were repeated with one track ($\alpha$) rotated $180^\circ$ such that tracks $\alpha$ and $\beta$ occupied juxtaposed orientations.

To further investigate network responses under conditions in which the two tracks occupy opposite orientations, trials were constructed in which the orientation of track position $\alpha$ was reversed. This was done for all five recording sessions used for the translation experiments. In this manipulation, the allocentric position (Pos) and the head direction (HD) variables were altered. This also essentially flipped the direction of the route so that if the original start position were on the right-hand side of the track, then to
maintain the same turn sequence, the start position would shift to the left-hand side of the track under the reoriented track condition. The inverse translation experimental design constitutes investigation of network response when the track is reoriented as well as linearly translated.

**Results**

**Matched Neurons Conjunctively Encode Multiple Spatial Reference Frames**

Similar to RSC neurons recorded in behaving animals, the evolved networks effectively encoded all three spatial frames of reference (allocentric, egocentric, and route-centric). Figure 2.4 shows one example for each functional neuron type where the black trace is from the biological neuron and the red trace is from the simulated neuron. A ‘turn insensitive’ neuron (Figure 2.4a) is not tied in its firing to any particular action correlate, but nevertheless reliably generates a complex spatially-specific firing pattern. In contrast, purely ‘turn sensitive’ neurons (Figure 2.4b) respond primarily to a preferred turn type (in this case, left turns), regardless of where such actions occur along a route. Between these extremes, some ‘turn sensitive’ neurons that have spike rates specific to route position (Figure 2.4c) also respond to specific turning actions. In such cases, higher firing rates are found for a specific instance along a route (e.g., in Figure 2.4c; higher firing rates for the first right turn on the RLR route and the first left of the LRL route).

We also verified the functionality of the model by removing the explicitly matched synthetic neurons ($n = 228$) from the population. We found that the remainder of the neurons could still replicate the functional neuron type distributions, turn bias-related activation patterns, and population behavior of the original dataset (Figure 2.5). For the cases in which ensemble activity patterns associated with trials taken from two different track
positions were used to reconstruct position, discrimination between the two positions increased (Figure 2.5d compared to 2.5h). This appears to be unrelated to the higher percentage of turn-sensitive cells in the population as compared to the experimental data since the type distributions remain largely the same after matched neurons are removed (Figure 2.5c compared to 2.5g). The ensemble correlation matrices for within-position (i.e., even vs. odd trial comparisons) were slightly noisier following the removal of matched neurons, but it did not impact the significance of differences between within-position and between-position reconstructions. Positional ensemble
reconstruction matrices and error bar plots were created by the means described in Figure 1.6 except that they were not averaged over all networks.

![Figure 2.5](image)

**Figure 2.5.** Evolved models of RSC match electrophysiological data when explicitly matched neurons are excluded. a) Positional ensemble reconstruction matrices when explicitly matched neurons are included. Even vs. odd trial reconstructions are accurate, while between-track reconstructions are poor. b) Cumulative distribution functions of turn bias-related activity are significantly higher than would be expected by chance (non-randomized cdf, blue; randomized cdf, red) when explicitly matched neurons are included. c) Functional neuron type distributions approximate those seen in the original dataset when explicitly matched neurons are included. d) Bar plot of averaged reconstruction error when matched neurons are included. Error is very low for within-position reconstructions, while error is significantly higher for between-position reconstructions. e) Positional ensemble reconstruction matrices when explicitly matched neurons are excluded. Even vs. odd trial reconstructions are accurate, while between-track reconstructions are poor, consistent with the results when explicitly matched neurons are included. f) Cumulative distribution functions of turn bias-related activity are still significantly higher than would be expected by chance (non-randomized cdf, blue; randomized cdf, red) when explicitly matched neurons are excluded. g) Functional neuron type distributions approximate those seen in the original dataset when explicitly matched neurons are excluded, and do not differ at all from when they are included. h) Bar plot of averaged reconstruction error when matched neurons are excluded. Error is very low for within-position reconstructions, while error is significantly higher for between-position reconstructions, consistent with the results when matched neurons are included.

**Evolved SNNs Are Robust to Loss of Input Streams**

To investigate how different input streams might affect RSC responses, we eliminated specific inputs into the evolved networks and tested model fitness. With the
exception of allocentric position, the evolved networks were highly robust to lesions of single input streams, suggesting that RSC function in intact animals cannot overcome lesions to the hippocampus whose neuron ensembles accurately encode environmental location. Significant decreases in fitness were observed when two or more input streams were lesioned (Kruskal-Wallis, $\chi^2(7) = 62.01$, $P < 0.0001$; Figure 2.6a). Combined angular velocity and head direction lesions had about the same impact as lesions eliminating environmental position, and, in general, fitness decreased with the number of input lesions.

We also investigated how different intrinsic connections might affect RSC responses.

![Figure 2.6. Fitness and Reconstruction Errors for Simulated Lesion Experiments. a) Fitness, as measured by the values of correlations between matched synthetic and experimental neural activity, declines with the number of lesions. Fitness is near zero when all critical inputs are absent from the network (AV-HD-Pos). b) Bar plots of reconstruction error associated with each lesion case. c) When all critical inputs are lesioned (AV-HD-Pos), reconstruction error is universally high with no significant differences between reconstructions. Asterisks denote level of significance (* = $P < 0.05$; ** = $P < 0.01$; *** = $P < 0.001$).](image)

The evolved networks were relatively insensitive to the loss of the linear velocity input and to the loss of recurrent excitatory connections. However, lesions to inhibitory connections had a strong effect on fitness ($F_{\text{inh}} = 40.37 \pm 21.46$), primarily because inhibitory connections stabilized the neuronal firing rates and this loss of stability impacted the correlations between synthetic and experimental neural activity. The average firing rate
per neuron in unlesioned networks was only 4.41 Hz, with a maximum of 72.07 Hz, while, when inhibitory connections were lesioned, the average firing rate was 181.36 Hz, with a maximum of 481.56 Hz. The result is consistent with the known powerful impact of inhibitory networks on both network stabilization (bounded responsiveness of excitatory elements) and fidelity of representation (Froemke, 2015). It should be noted, however, that most cases in which at least one input was lesioned resulted in much higher firing rates than normal, and the impact on firing rate in the HD lesion case was about on par with that of the inhibitory neuron lesion case. Thus, it must be noted that a major impact on network fitness results from disruptions in constraints on firing rates (Table 2.1).

Despite the remapping of firing rates, the simulated RSC population could still accurately reconstruct the agent’s position when allocentric position of a given route was consistent. Of the lesion cases examined, only when HD and Pos were lesioned together did the

<table>
<thead>
<tr>
<th>Table 2.1 Firing Rates of Lesioned Networks.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lesion Type</td>
</tr>
<tr>
<td>-------------------</td>
</tr>
<tr>
<td>Unlesioned</td>
</tr>
<tr>
<td>AV</td>
</tr>
<tr>
<td>HD</td>
</tr>
<tr>
<td>Pos</td>
</tr>
<tr>
<td>AV-HD</td>
</tr>
<tr>
<td>AV-Pos</td>
</tr>
<tr>
<td>HD-Pos</td>
</tr>
<tr>
<td>AV-HD-Pos</td>
</tr>
<tr>
<td>Exc</td>
</tr>
<tr>
<td>Inh</td>
</tr>
</tbody>
</table>
reconstruction errors for $\alpha/\beta$ fall significantly such that they could not distinguish the difference in the two track positions. $\alpha/\alpha$ and $\beta/\beta$ (Kruskal-Wallis, $\chi^2(5) = 67.24$, $P < 0.0001$; Figure 2.6b). By comparison, reconstruction error subsequent to HD and Pos lesions rose more moderately, suggesting that the system cannot effectively disambiguate the environmental locations of routes in the absence of information input streams providing either head direction and environmental location. This may be related to the finding that some head direction cells in the RSC also conjunctively encode place.

On the other hand, when AV, HD, and Pos inputs were lesioned jointly, the network could no longer accurately reconstruct position, yielding universally high reconstruction errors (Kruskal-Wallis, $\chi^2(5) = 20.14$, $P < 0.01$; but Tukey’s HSD post-hoc comparison, $P > 0.05$) (Figure 2.6c). Taken together, these simulated lesion experiments predict that conjunctive coding makes the RSC robust to elimination of individual inputs and flexible enough to generate coherent spatial representations when pairs of some input streams are impacted.

**Rearranging Turn Sequence Causes Functional Remapping**

To predict how the RSC coding might respond to the imposition of novel routes and paths, we created three new simulated routes that gave rise to six new action sequences: LLL (outbound) and RRR (inbound) (Figure 2.3a), LLR (outbound) and LRR (inbound) (Figure 2.3b), and RLL (outbound) and RRL (inbound) (Figure 2.3c). Remarkably, we found that the networks performed similarly compared to the original LRL/RLR case, in that the progression along the route could be accurately predicted by RSC population activity between even and odd trials associated with the same track position and trajectory (Figure 2.7a, b). The result suggests that the RSC in intact animals is configured to provide first-
trial representation of route space, environmental space, and actions when novel navigational contexts are encountered. Significant differences emerged between comparisons of even vs. odd trials and comparisons of trials from tracks $\alpha$ and $\beta$ (Figure 2.7c). Again, the Kruskal-Wallis test indicated the presence of significant differences and

![Diagram](image)

**Figure 2.7. Model Behavior Generalizes to Novel Route Configurations.**

a) Average ensemble reconstruction matrices (created as described in Figure 1.6a and b) for correlations between activity associated with even and odd trials for all position and route combinations (e.g., $\alpha$LLL, $\alpha$RRR, $\beta$LLL, and $\beta$RRR). b) Ensemble reconstruction matrices for correlations between activity associated with all trials for both track positions and both routes. c) Bar plots of average reconstruction error for all position and route combinations (average per-bin error was calculated as in Figure 1.6c). These plots reveal that reconstructions between tracks $\alpha$ and $\beta$ result in significantly higher reconstruction error than reconstructions between even and odd trial activity for same position and route comparisons. d) Neuron type distributions for the altered tracks.

44
Tukey’s HSD was used to determine which groups were different. (LLL/RRR: Kruskal-Wallis, \( \chi^2(5) = 428.91, P < 0.0001 \); Tukey’s HSD post-hoc comparison, \( P < 0.05 \); LLR/LRR: Kruskal-Wallis, \( \chi^2(5) = 374.01, P < 0.0001 \); Tukey’s HSD post-hoc comparison, \( P < 0.05 \); RLL/RRL: Kruskal-Wallis, \( \chi^2(5) = 650.72, P < 0.0001 \); Tukey’s HSD post-hoc comparison, \( P < 0.05 \)). Furthermore, the neuron type distributions were similar to those reported in Alexander and Nitz (2015) (Figure 2.7d).

We again measured network fitness by correlating the synthetic firing rate profiles associated with the new route shapes with the experimentally recorded firing rate profiles associated with the original route shapes in order to see whether each neuron’s representation of the new route would still be correlated with its representation for old routes as a way to judge remapping. Despite the rearrangement of turn sequences, the fitness levels of the networks were approximately the same as under the original track conditions when new matches between synthetic and experimental neurons were allowed (Table 2.2, Unforced Match). Because fitness is based on how well synthetic neural activity correlates with experimental neural activity, and since these turn sequences were not investigated experimentally, the biological neurons could not have encoded those specific routes and turn sequences. However, when the synthetic neurons were forced to match the

<table>
<thead>
<tr>
<th>Turn Sequence</th>
<th>Unforced Match</th>
<th>Forced Match</th>
</tr>
</thead>
<tbody>
<tr>
<td>LLL / RRR</td>
<td>95.03 (3.97)</td>
<td>5.82 (9.36 \times 10^{-16})</td>
</tr>
<tr>
<td>LLR / LRR</td>
<td>95.38 (4.42)</td>
<td>10.24 (1.87 \times 10^{-15})</td>
</tr>
<tr>
<td>RLL / RRL</td>
<td>94.46 (5.66)</td>
<td>5.79 (9.36 \times 10^{-16})</td>
</tr>
</tbody>
</table>
experimentally observed neurons recorded under original route conditions, fitness was very low (Table 2.2, Forced Match). This suggests that although individual neuronal firing rate patterns remap when features of the environment (position, turn sequence, etc.) are altered, the simulated RSC population can flexibly and rapidly remap to novel environments. This suggests that the biological RSC supports flexible encoding of specific features while retaining information about the broader environmental context of the room. Note that measures of fitness are unrelated to either the quality of reconstructions of position within a route since reconstruction depends on ensemble activity across the population.

**Model Generates Robust Representations for Novel Track Positions**

We next systematically moved the track, either by linearly translating it across environmental space or by rotating the track around its center. These manipulations were intended to test the effects of perturbing the allocentric and egocentric frames of reference on RSC responses. We ran five sessions of each condition in which the tracks occupied different starting positions (Orientation: Figure 2.3a-e; Translation: Figure 2.3f-j). We also ran additional trials in which the track at position α was rotated 180 degrees so that it was in the opposite orientation and then linearly translated across the space.

We found that changing the position or orientation of the track resulted in neuron remapping, as can be observed by the drop in fitness scores at each new position (Figure 2.8a and 2.8b). In the orientation case, the fitness reached an asymptote at approximately 10 arbitrary units (a.u.), which corresponds roughly a Pearson’s R of 0.20, after the track had been rotated 80 degrees (Figure 2.8a). This drop in fitness, when the track was rotated around its center point, was more abrupt than when the track was simply translated across.
allocentric space (Figure 2.8b), evidencing the powerful effect of head orientation as a variable to disambiguate navigational contexts. During translation, network fitness fell somewhat more gradually to reach an asymptote at approximately 20 a.u. (approximately R = 0.10 at about 160 cm from the original position). We plotted lines of best fit associated with each change in fitness across nine positions for translation and rotation (Figure 2.8a, b) to demonstrate the influence exerted over neural activation when two input variables are changed (e.g., sharper decline in fitness when the track is rotated) as opposed to when only one input variable is changed (e.g., gradual decline in fitness when the track is translated). This may suggest a conjunctive coding scheme in the retrosplenial cortex by which novel route representations are generated.
To examine how the networks represent space and how these representations are grounded in both external and internal frames of reference, we calculated the average reconstruction errors, based on the positional ensemble reconstructions, between all possible locations of the track to see how well the network could predict position within route when routes were in orthogonal, parallel, or juxtaposed orientations, or when the routes overlapped in allocentric space for each selected recording session individually (for clarity, relevant examples are shown in Figure 2.9, but similar results were obtained for all sessions). We calculated the reconstruction errors via positional ensemble correlations for all eighteen positions covering the complete 360-degree orientation of the track, which were then averaged and visualized using a heat map. The values were superimposed over a polar plot (Figure 2.9). Each value in the plot is the averaged reconstruction error for track $\alpha$ vs. $\beta$ for four example sessions of trials in which the track was rotated. Each slice of the polar plot corresponds to the rotation of $\alpha$’s track position (0 degrees is the original position of $\alpha$). Inside each slice are eighteen radial bars that represent the average reconstruction error between $\alpha$ and $\beta$ for each orientation of $\beta$ (the bar at the center is the original position of track $\beta$, and the outermost bar is the position of $\beta$ at 340 degrees). The average reconstruction error for each slice were computed as described in Figure 1.6; that is, for all ten networks, the actual prediction line was subtracted from the theoretical prediction line to achieve a vector with a length of 200 bins that contained the bin-by-bin error. These vectors were averaged across the ten networks to derive the average bin-by-bin error. The average of that vector is represented in each slice of the polar plot.

The networks were most sensitive to rotated routes when allocentric positions closely overlap. For this particular recording session, the tracks’ center points occupy the
Figure 2.9. Network Representations Change Based on Amount of Conflict Between Relevant Behavioral Variables. 

a) When tracks $\alpha$ and $\beta$ closely overlap during rotation, the average reconstruction error for correlations between neural activity associated with each track is low, indicated by a line of lowest error (blue values in polar plot). 

b) When tracks are in similar positions, average reconstruction error is low (left, track locations with minimum error; red circle in polar plot). When the tracks are in opposite orientations, average reconstruction error is high (right, see track locations with maximum error; black circle in polar plot). 

c) When the center points of the tracks do not overlap in space, the tracks cannot overlap completely but may be parallel and partially overlap at various points. Due to heightened conflict between variables, a blue spiral line is no longer discernible. 

d) When the tracks are parallel and non-overlapping, error is lowest (left). When the tracks are non-overlapping but occupy opposite orientations (middle), error is highest. When the tracks are parallel but do not fully overlap (last third of $\alpha$, containing the second left turn, overlaps with the first third of $\beta$, containing the first left turn), there is high behavioral conflict, resulting in high average reconstruction error (blue circle in polar plot). 

e) When the tracks are sufficiently far apart that their positions never overlap as they undergo rotation, the blue spiral line of lowest error is discernible. 

f) When the tracks are parallel but occupy different locations, average error is low (left). When they occupy opposite orientations, average reconstruction error is high (right).
same approximate allocentric position. This means that as each track is rotated, there are points at which they occupy both the same orientation and position in space, and are thus the same route. One would expect that reconstruction error should be low at these points, since this is analogous to comparing even and odd trials from the same track position and route. This would give rise to a blue spiral of low reconstruction errors on the polar plot in which the positions and orientations systematically match. This expectation was confirmed (Figure 2.9a). To highlight this, we circle the location of the minimum value of the reconstruction error on the polar plot in black and plot the relative track positions of $\alpha$ and $\beta$ at that point (Figure 2.9b, left). We similarly highlight the maximum reconstruction error value by circling that point in red and plotting the relative track positions and positional ensemble reconstruction matrix (2.9b, right). This demonstrates that the network is sensitive to the route-centric frame of reference and the allocentric frame of reference because the network recognizes track positions that occur at similar positions and orientations in space, but is highly discriminative of the routes when they are in opposite orientations but share a similar allocentric position.

When the tracks are further apart in allocentric space such that the arms of the track can overlap at some positions, but are not fully overlapping, these spiral lines disappear, which shows that the network cannot consistently discriminate routes that occur in similar orientations. This suggests that confusion can arise in the network reconstructions when routes partially overlap in allocentric space (Figure 2.9c). Interestingly, when the track positions occupy analogous orientations but do not overlap in allocentric space, reconstruction error is relatively low. This is shown by the point of lowest error circled in red on the polar plot and the corresponding track positions (Figure 2.9c, d, left). On the
Figure 2.10. Variable Conflict Results in Failure to Disambiguate Similar Routes. a) When reconstructing the agent’s position on track $\alpha$ using even and odd trials from the same session, reconstruction error is low. When comparing activity from different positions of track $\alpha$, error is higher, scaling with the distance between them (top). This pattern is mirrored in reconstructions on track $\beta$ from $\alpha$ (middle). The original track locations are shown in black and dark green, respectively. For reference, all positions of both tracks are shown as they were translated across allocentric space (gray and light green, respectively) (bottom). b) When reconstructing position on track $\alpha$ for opposite turn sequences (routes LRL and RLR), reconstruction error is lower when the tracks occupy the same allocentric position versus when they do not (top, blue values). Only head direction differed dramatically between the routes, suggesting that the conflict in heading is responsible for the higher error. When reconstruction error was measured for trials where tracks occupied juxtaposed orientations, it scaled with the amount of conflict between the variables associated with each track (middle). For reference, the positions of the two tracks (with inverted $\alpha$) are shown as they were translated across the environment (bottom). c) We show three illustrative examples of variable conflict as it relates to reconstruction error. Error is high when the tracks occupy opposite orientations and are near one another, but not overlapping (left). It increases when the tracks overlap in space while occupying opposite orientations (middle). Variable conflict declines as the tracks move away from one another in opposite directions, resulting in reduced error (right).
other hand, when the track positions occupy opposite orientations with minimal overlap, reconstruction error is relatively high (Figure 2.9c, d, middle). Finally, when the tracks are in the same orientation and their allocentric positions overlap closely but the routes are not aligned, reconstruction error is high, suggesting that the behavioral variables at these two positions are in conflict such that the network represents them as two separate routes, allowing it to discriminate between them (Figure 2.9c, d, right). This conclusion is consistent with the track translation experiments where head direction and allocentric position were in conflict between routes (Figure 2.10). In this figure, a grid is shown for each track position combination. Each colored square represents the averaged bin-by-bin reconstruction error for all ten networks, consistent with the slices in the polar plots for Figure 2.9. In the translation case, the network’s holistic representations for each route were highly dissimilar, resulting in high reconstruction error (Figure 2.10).

When positions do not overlap in allocentric space, network sensitivity to similar route orientations is re-established. That is, when the allocentric positions of the tracks are entirely non-overlapping, these spirals of low reconstruction errors are again discernible (Figure 2.9e), suggesting that the networks remain sensitive to routes that occupy the same or opposite orientation, but entirely different allocentric position, in space. Although reconstruction errors are overall higher due to the distance between the track positions, this demonstrates a coherent and sophisticated representation of the route-centric frame of reference that is contextualized by the allocentric frame of reference (Figure 2.9e). As before, when the tracks occupy analogous orientations, reconstruction error for the session is at its lowest at a value of 32.25 arbitrary units (Figure 2.9f, left), while the error is at its
highest (a value of 99.17 arbitrary units) when the tracks occupy opposite orientations (Figure 2.9f, right).

**Neurons Undergo Functional Remapping**

Finally, we found that neuronal responses changed over the course of trials in which tracks were translated linearly or rotated in allocentric space, and could be re-assigned to different functional categories than the one to which it previously belonged at the track’s initial position. This was calculated by statistically evaluating the neuron’s firing rate profile at each new track position and averaging the number of times it remapped over the course of all manipulated trials. Consistent patterns of remapping emerged across the two manipulations: turn insensitive neurons not modulated by route were most likely to remap to turn sensitive neurons, but were unlikely to be route-modulated following remapping. Turn insensitive neurons whose activity profiles were modulated by route, on the other hand, typically retained their insensitivity to turns, but lost their route modulation. Turn sensitive, non-route modulated neurons tended to remap in both regards and became turn insensitive and route modulated. Turn sensitive neurons that were route modulated, in

<table>
<thead>
<tr>
<th>Orientation</th>
<th>Turn insensitive, no mod.</th>
<th>Turn insensitive, route mod.</th>
<th>Turn sensitive, no mod.</th>
<th>Turn sensitive, route mod.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Turn insensitive, no mod.</td>
<td>-</td>
<td>5.66% (0.84)</td>
<td>30.3% (3.2)</td>
<td>7.92% (1.02)</td>
</tr>
<tr>
<td>Turn insensitive, route mod.</td>
<td>45.4% (2.25)</td>
<td>-</td>
<td>24.01% (3.17)</td>
<td>7.19% (1.2)</td>
</tr>
<tr>
<td>Turn sensitive, no mod.</td>
<td>3.57% (0.41)</td>
<td>39.53% (3.2)</td>
<td>-</td>
<td>11.65% (0.76)</td>
</tr>
<tr>
<td>Turn sensitive, route mod.</td>
<td>3.15% (0.49)</td>
<td>36.37% (4.01)</td>
<td>42.81% (1.84)</td>
<td>-</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Translation</th>
<th>Turn insensitive, no mod.</th>
<th>Turn insensitive, route mod.</th>
<th>Turn sensitive, no mod.</th>
<th>Turn sensitive, route mod.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Turn insensitive, no mod.</td>
<td>-</td>
<td>7.44% (2.78)</td>
<td>20.13% (2.17)</td>
<td>5.57% (1.71)</td>
</tr>
<tr>
<td>Turn insensitive, route mod.</td>
<td>48.66% (4.34)</td>
<td>-</td>
<td>12.96% (1.9)</td>
<td>5.6% (0.95)</td>
</tr>
<tr>
<td>Turn sensitive, no mod.</td>
<td>3.04% (1.75)</td>
<td>35.95% (3.38)</td>
<td>-</td>
<td>10.26% (2.11)</td>
</tr>
<tr>
<td>Turn sensitive, route mod.</td>
<td>3.61% (1.61)</td>
<td>32.98% (3.84)</td>
<td>38.11% (1.76)</td>
<td>-</td>
</tr>
</tbody>
</table>
contrast, often retained their turn sensitivity but lost their modulation by route. These patterns were consistent across both linear translation and rotation (see Table 2.3). It was unusual for turn sensitive neurons to change their preferred type of turn instance (i.e., neurons that preferred left (right) turns continued to respond to left (right) turns if they retained their turn sensitivity). Interestingly, neurons remapped frequently on an individual basis; however, the overall distributions of each functional category across the population remained relatively stable (the maximum shift for each category never surpasses 6.8% for the orientation case and 11.5% for the translation case). Examples of neurons whose activity profiles demonstrate functional remapping are shown in Figure

Figure 2.11. Example Neurons that Underwent Functional Remapping During Track Manipulation. a) An example neuron whose firing rate pattern was consistent with turn insensitivity at the original track position (black trace), but whose firing pattern changed over the course of track rotation to exhibit turn sensitivity. b) Bar plots of mean firing rate for all six left (gray bars) and right (white bars) turn instances. The ratio of the neuron’s average activity for left to right turns was 5.61 before remapping, and was 26.68 following remapping. c) An example neuron whose firing rate pattern is consistent with turn insensitivity both before and after remapping. However, before remapping the neuron’s activity does not exhibit modulation by route (black trace). Following remapping, the neuron’s activity is route-modulated (red trace). d) Bar plots of the Pearson’s $r$ correlation coefficient before and after remapping between tracks $\alpha$ and $\beta$ for both turn sequences (LRL and RLR). Before remapping, the neuron’s activity for the same turn sequences and different track locations was negatively correlated ($\alpha$LRL vs. $\beta$LRL, $r = -0.07$; $\alpha$RLR vs. $\beta$RLR, $r = -0.28$). Following remapping, the activity neuron’s activity for those track positions and turn sequences were strongly positivity correlated ($\alpha$LRL vs. $\beta$LRL, $r = 0.78$; $\alpha$RLR vs. $\beta$RLR, $r = 0.57$).
In one instance, a previously turn insensitive neuron becomes sensitive to left turns (Figure 2.11a, b), and in another instance, a previously turn insensitive neuron without route modulation remaps to become route modulated but retains its turn insensitivity (Figure 2.11c, d). These results suggest, together with analysis of the networks’ ability to reconstruct routes under novel conditions in which the track positions are altered, that neurons in the retrosplenial cortex can reliably and flexibly encode multiple task-relevant variables such that their activities may shift to meet changing task demands. This is consistent with the observation that neurons in higher cortical regions may demonstrate category-free coding and/or mixed selectivity to handle more complex cognitive tasks (Raposo, Kaufman and Churchland, 2014); (Rigotti, Barak, Warden, Wang, Daw, Miller and Fusi, 2013); (Eichenbaum, 2017).

**Discussion**

To gain a better understanding of how the RSC flexibly encodes space, we used a novel method to match neurophysiological data from the rodent RSC to a simulated spiking neural network using behavioral measurements such as allocentric position, head direction, linear and angular velocity as inputs. In a series of virtual experiments where we simulated lesions and changes to the rodent’s environment, we were able to show that the evolved RSC models support: 1) Conjunctive encoding of multiple, disparate input streams, consistent with results reported previously in this brain region; 2) Resiliency to loss of sensory input streams; and 3) Flexible and generalizable encoding at the population level. Lastly, we showed that neuronal firing patterns in the RSC transcend assigned functional categories.

The retrosplenial cortex is well-suited to playing a role in navigation and the
generation of routes. It contains an integrated head direction system (Cho and Sharp, 2001); (Taube, 1995); (Clark, Bassett, Wang and Taube, 2010) based on input from the anterior thalamic nuclei and targets cells in the entorhinal cortex, which may provide temporal structure to the firing of hippocampal cells (Schlesiger, Cannova, Boublil, Hales, Mankin, Brandon, Leutgeb, Leibold and Leutgeb, 2015). A computational model of mental imagery suggests a role for the RSC in scene construction underlying imagination and episodic memory which hinges on the transformation of spatial information between different frames of reference (Byrne, Becker and Burgess, 2007); (Burgess, Becker, King and O'Keefe, 2001). Because the RSC is situated between the hippocampus and parietal cortex (Vann, Aggleton and Maguire, 2009); (Todd, Huszár, DeAngeli and Bucci, 2016); (Robinson, Keene, Iaccarino, Duan and Bucci, 2011), it may also support the development and deployment of different kinds of navigation strategies that are tied to different spatial frames of reference (e.g., allocentric, egocentric, and route-centric). A high-level computational model showed that the RSC is capable of choosing and deploying such strategies (Oess, Krichmar and Röhrbein, 2017).

Several important and specific predictions result from the present experiments. Our simulations of the RSC suggest that new routes through a familiar environment do not need to be learned – if an organism has experience within a broader environmental context, then learning may not be necessary for it to flexibly adapt to new paths through that environment. This is reminiscent of the schema concept proposed by Tse and colleagues (Tse, Langston, Kakeyama, Bethus, Spooner, Wood, Witter and Morris, 2007); (Tse, Takeuchi, Kakeyama, Kajii, Okuno, Tohyama, Bito and Morris, 2011); (Preston and Eichenbaum, 2013). Specifically, our evolved SNNs adapted flexibly to new routes within
allocentric space that it had never seen before during the evolutionary training process, as shown by experiments where we altered the turn sequences associated with specific routes within the simulated room. The networks could reconstruct the agent’s position on the route from neural ensemble activity, but could also disambiguate similar routes that occupied different positions in allocentric space. This may help to explain a phenomenon associated with damage to the RSC (particularly the right RSC) called ‘topographic disorientation,’ a condition in which patients with this kind of damage can recognize familiar landmarks, but can no longer construct or navigate routes between them (Takahashi, Kawamura, Shiota, Kasahata and Hirayama, 1997); (Kim, Aminoff, Kastner and Behrmann, 2015). If the RSC were responsible for planning and/or deploying routes through familiar environments, then damage to the region would predict topographic disorientation. This is also consistent with theoretical results reported in Oess et al. (2017) (Oess, Krichmar and Röhrbein, 2017), in which the RSC was used to build a confidence level for the deployment of different kinds of navigation strategies (allocentric, egocentric, or route-centric) based on the kinds of incoming sensory input. Our results may provide a biologically plausible substrate for these navigation strategies. In contrast, our model is different from the model created by Byrne et al. (2007) (Byrne, Becker and Burgess, 2007) in that representations of new places or routes in a familiar environment do not need to be learned. Instead, neuronal activity in the RSC remaps to accommodate new representations of different routes that occupy the same general space. Our model is broadly consistent with the idea that what reference frame is employed by the RSC depends on the sensory inputs to the region, but its activity changes flexibly and immediately without relearning.
We also predict that spatial representations in the RSC are closely tied to specific associations between inputs to the region. In a set of simulated lesion experiments, we found that our model’s learned associations between the different input streams were sufficient for the network to perform pattern completion in the absence of any input. However, if these associations were disrupted in such a way that there was conflict between the behavioral variables associated with a specific representation, then the network was sensitive to that conflict and the representations were disrupted, as revealed by experiments in which the track was rotated but the head direction input remained fixed. This is consistent with findings reported by Nelson et al. (2015) (Nelson, Powell, Holmes, Vann and Aggleton, 2015) in which rats with lesions to the RSC were able to solve a standard T-maze alteration task if intra-maze cues were consistent, but showed impairments when cue placements were in conflict. The findings suggested that the RSC is sensitive to multiple cue types and that damage to the RSC might impair the rats’ ability to flexibly switch their attention from between either different kinds of spatial information or between different spatial navigation strategies, but the two possibilities could not be disambiguated. Based on activity seen in our evolved SNNs, we suggest the impairments are likely due to both, since specific kinds of inputs seem to be closely related to specific navigation strategies. These findings are also consistent with evidence from the RSC that it is an association cortex where multimodal sensory information is integrated. For example, evidence suggests that the RSC integrates visual and idiothetic information for error correction during navigation, and that lesions disrupt cross-modal object recognition (Hindley, Nelson, Aggleton and Vann, 2014); (Cooper and Mizumori, 1999). Moreover, neurons in the region exhibit conjunctive reward-location representation similar to the
hippocampus (Vedder, Miller, Harrison and Smith, 2016). Furthermore, our experiments revealed that the neural activity underpinning these representations are not fixed. Instead, activity is flexible and adaptable, such that neurons can represent changing contexts and task demands in a flexible and parameter-free way, which is similar to what has been observed in the parietal cortex (Raposo, Kaufman and Churchland, 2014); (Vedder, Miller, Harrison and Smith, 2016).

On a separate note, this model represents a breakthrough for theoretical neuroscience. By training generic but biologically plausible SNNs to replicate electrophysiologically recorded neural activity, we have added a level of veracity to the results provided by the simulated experiments. The present experiments further show that such models can be used to make powerful theoretical predictions for biological brain regions, which can be tested and verified by experimentalists to the mutual benefit of computational and experimental researchers. Although similar methods have been used for computational modeling, they have not replicated data from entire brain regions to test cognition and are not as flexible or generalizable. Such methods have instead focused on the output of specific kinds of behavior (Dura-Bernal, Neymotin, Kerr, Sivagnanam, Majumdar, Francis and Lytton, 2016), the replication of specific cell types (Carlson, Nageswaran, Dutt and Krichmar, 2014), or on the evolution of specific neural models (Fountas and Shanahan, 2015). More abstract models, such as NEAT, have focused on evolving efficient neural network topologies rather than replicating neuronal behavior (Stanley and Miikkulainen, 2002) and other methods of evolving SNNs using GPU-accelerated frameworks have focused on using them for data analysis (Kasabov, Dhoble, Nuntalid and Indiveri, 2013). Our framework may represent a promising avenue for the
generation of new predictions and theories about cognitive function, since simulated experiments can be used to inform researchers of the best avenue of research, which could potentially help to save tremendous time and resources.

We would also like to emphasize that the framework is generalizable and adaptable, in contrast with other computational models that are usually designed with a specific task, behavior, and/or brain region in mind (Byrne, Becker and Burgess, 2007); (Oess, Krichmar and Röhrbein, 2017). Our method was not designed for one specific brain region or task; therefore, any model can be evolved in response to any number of user-defined inputs, meaning that it can be applied to electrophysiological datasets from virtually any brain region. The evolved models resulting from the method can also be adapted for use in different tasks than the ones they were designed to solve. This represents a major stepping stone for computational neuroscientists because it is a new, flexible, and dynamic way of creating and testing theoretical models of cognition and behavior. We suggest that these kinds of tools and techniques will be very powerful as we further our efforts to understand how neural activity gives rise to behavior.

In summary, these results support the idea that the RSC is an association cortex that can rapidly and flexibly integrate multisensory information. In a navigation task, such as that explored here, the RSC contains multiple frames of reference (i.e., allocentric, egocentric, route-centric) necessary to support spatial memory (Alexander and Nitz, 2015); (Oess, Krichmar and Röhrbein, 2017). Moreover, the present results support the idea that the RSC carries a conjunctive code that makes it resilient to perturbations. Finally, the present results highlight the ability of the RSC to remap due to contextual changes.
References


EICHENBAUM, H. 2017. Barlow versus Hebb: When is it time to abandon the notion of feature detectors and adopt the cell assembly as the unit of cognition? Neuroscience Letters.


CHAPTER 3: Nonnegative Sparse Coding as a Canonical Computation

Introduction

In this chapter, a novel framework for understanding neuronal response properties and receptive fields is presented. Supported by recent computational studies, nonnegative sparse coding (NSC) is emerging as a ubiquitous coding strategy across brain regions and modalities. A combination of nonnegative matrix factorization (NMF) and sparse coding, NSC allows neurons to efficiently encode high-dimensional stimulus spaces using a sparse and parts-based population code. Reducing the dimensionality of complex, multimodal sensory streams is particularly important for brain areas to represent the world. In this article, we provide an overview of NSC, summarize evidence for its role in neural computation in several disparate regions of the brain, and speculate that specific forms of synaptic plasticity and homeostatic modulation may underlie its implementation. We suggest that NSC may be an organizing principle in the nervous system.

Nonnegative Sparse Coding

Nonnegative sparse coding (NSC) is a framework that combines nonnegative matrix factorization (NMF), a linear dimensionality reduction technique from statistical learning, with sparse population coding from neural network theory (Hoyer, 2002; (Eggert and Körner, 2004). NMF belongs to a class of methods that can be used to decompose a multivariate data matrix \( V \) into an inner product of two reduced-rank matrices \( W \) and \( H \). NMF assumes that the observed data in \( V \) are caused by a collection of latent factors weighted by nonnegative numbers, representing both the presence and the intensity of the cause.

In the context of NSC, \( V \) and \( H \) correspond to activation values of two distinct
Nonnegative matrix factorization (NMF)

Original NMF

Reconstruction

$\vec{v}_s', \vec{h}_s' \approx \vec{v}_s, \vec{h}_s \approx F = B_1 B_2 S F W B_3 S^* + F W B_3 S^* \times B_2 S$
neuronal populations, which are connected to each other via synaptic weight values in $W$ (Figure 3.1). Consider a number of data samples $s \in [1, S]$, for example, in the form of observed firing rates of a population of $F$ neurons. If we arrange the observed values of the $s$th observation into a vector $\overline{v}_s$, and if we arrange all vectors into the columns of a data matrix $V$, then linear decompositions describe these data as

$$V \approx WH.$$ 

Here, $W$ is a matrix that contains as its columns a total of $B$ basis vectors (a lower-dimensional set of linearly independent elements that can represent a high-dimensional input space given a weighted sum of these elements) of the decomposition. $H$ contains as its rows the hidden coefficients (a set of weights associated with each element of the input vector) that give the contribution of each basis vector in the input vectors.

The difference between $V$ and $WH$ is termed the reconstruction error.

The goal of NSC is then to find a linear decomposition of $V$ that minimizes the
reconstruction error, while guaranteeing that both $\mathbf{W}$ and $\mathbf{H}$ are sparse. This can be achieved by minimizing the following cost function (Equation 3.1) (Hoyer, 2002):

$$\min_{\mathbf{W}, \mathbf{H}} \frac{1}{2} \| \mathbf{X} - \mathbf{WH} \|^2 + \lambda \sum_{ij} f(H_{ij}),$$

subject to the constraints $\forall i\, : \, W_{ij} \geq 0, H_{ij} \geq 0$, and $\| w^i \| = 1$, where $w^i$ denotes the $i$th column of $\mathbf{W}$. Here, the left-hand term describes the reconstruction error, which can be minimized with NMF, and the right-hand term describes the sparseness of the decomposition. The trade-off between sparseness and accurate reconstruction is controlled by the parameter $\lambda$ ($\lambda \geq 0$), whereas the form of $f$ defines how sparseness is measured (a typical choice is the L1 norm on $\mathbf{H}$).

**Nonnegative Sparse Coding as a Modern Variant of the Efficient Coding Hypothesis**

Early theories of efficient coding (Barlow, 1961); (Attneave, 1954) were developed based on the visual system. Realizing that the set of natural scenes was much smaller than the set of all possible images, it was argued that the visual system should not waste representational resources on processing arbitrary images. Instead, the nervous system should use statistical knowledge about its environment to represent the relevant input space as economically as possible.

Modern renditions of this theory (such as sparse coding (Olshausen and Field, 1996) and ICA (Bell and Sejnowski, 1997) have tried to refine the original hypotheses by tying neuronal response properties to the statistics of the natural environment (for a review, see Simoncelli and Olshausen (2001) (Simoncelli and Olshausen, 2001)). These theories were shaped by two fundamental empirical observations about early visual cortex: 1) a neuron’s receptive field (RF) resembled a decomposition of the visual stimulus into a series of local,
largely independent feature components (e.g., a 2-D Gabor function is basically a local approximation of the directional spatial derivative of an image), and 2) any individual neuron responded only sparsely to a small subset of stimulus features (e.g., orientation or color at a particular spatial location). Thus a neuron’s RF could be understood as a sparse, low-dimensional embedding of high-dimensional input stimuli. Such a representation is desirable from an energy-expenditure point of view, since it allows the visual system to represent any visual stimulus by activating only a small set of neurons, while most neurons in the population remain silent.

Olshausen and Field (Olshausen and Field, 1996) found that linear sparse coding of natural images yielded features qualitatively similar to the receptive fields of simple cells in primary visual cortex (V1), thus giving empirically observed RFs an information-theoretic explanation. However, as pointed out by Hoyer (Hoyer, 2003), sparse coding falls short of providing a literal interpretation for V1 simple-cell behavior for two reasons: 1) every neuron could be either positively or negatively active, and 2) the input to the neural network was typically double-signed, whereas V1 neurons receive visual input from the lateral geniculate nucleus (LGN) in the form of separated, nonnegative ON and OFF channels.

Hoyer (Hoyer, 2002); (Hoyer, 2003) proposed the NSC framework as a way to transform Olshausen and Field’s sparse coding from a relatively abstract model of image representation into a biologically plausible model of early visual cortex processing by enforcing both input signal and neuronal activation to be nonnegative. This seemingly simple fix had remarkable consequences on the quality of the sensory representation: Whereas elementary image features in the standard sparse coding model could “cancel
each other out” through subtractive interactions, enforcing nonnegativity ensured that
features combined additively, much like the intuitive notion of combining parts to form a
whole.

**Reconstructing Stimulus Spaces Using Sparse, Parts-Based Representations**

The idea that NSC could be used to explain neuronal response properties
goes back to an influential paper by Lee and Seung (1999) (Lee and Seung, 1999): They
found that applying a linear dimensionality reduction technique known as NMF to a
database of face images yielded sparse, localized features that resembled neuronal
responses in macaque inferotemporal cortex (Wachsmuth, Oram and Perrett, 1994), which
is the highest-order area in the ventral visual stream (Ungerleider and Haxby, 1994). NMF
is a statistical matrix decomposition technique that takes an input matrix V whose rows
correspond to distinct features of the input and whose columns correspond to a particular
observation of those features. Matrix V must be nonnegative, and NMF decomposes the
matrix into two reduced-rank matrices whose linear combination can be weighted such
that the product of W and H provides an accurate reconstruction of V (Figure 3.1A).

On its surface, NMF would appear to be unrelated to the mechanisms underlying
artificial or biological neural networks; however, it was Lee and Seung’s intuitive mapping
of these variables onto a neural network that forms the cornerstone of our conception of
the NSC framework: A particular image, encoded by a number of pixels \(v_1, \ldots, v_F\) (i.e., a
column of V), could be explained by a linear mixing of encoding variables \(h_1, \ldots, h_B\) (i.e., a
column of H). A single encoding variable influences multiple image pixels, owing to the fan-
out of the connections from the encoding variable. As a result, a particular image of a face
(Figure 3.1B1) could be represented by a linear combination of “basis images” (i.e., parts of
<table>
<thead>
<tr>
<th>Area</th>
<th>Sparse</th>
<th>Parts-based</th>
<th>Experimental Evidence</th>
<th>Modeled by NMF or NSC</th>
<th>Computational Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>Retina</td>
<td>X</td>
<td>X</td>
<td>(Onken, Liu, Karunasekara, Delis, Gollisch and Panzeri, 2016)</td>
<td>X</td>
<td>(Onken, Liu, Karunasekara, Delis, Gollisch and Panzeri, 2016)</td>
</tr>
<tr>
<td>Early visual cortex</td>
<td>X</td>
<td>X</td>
<td>(Olshausen and Field, 1996); (Hoyer, 2003); (Hoyer and Hyvärinen, 2002); (van Hateren and Ruderman, 1998)</td>
<td>X</td>
<td>(Olshausen and Field, 1996); (Hoyer, 2003); (Carlson, Nageswaran, Dutt and Krichmar, 2014); (Hyvärinen and Hoyer, 2001)</td>
</tr>
<tr>
<td>Ventral visual stream</td>
<td>X</td>
<td>X</td>
<td>(Wachsmuth, Oram and Perrett, 1994)</td>
<td>X</td>
<td>(Lee and Seung, 1999); (Hosoda, Watanabe, Wersing, Körner, Tsujino and Fujita, 2009)</td>
</tr>
<tr>
<td>Dorsal visual stream</td>
<td>X</td>
<td>X</td>
<td>(Hamed, Page, Duffy and Pouget, 2003); (Pouget and Sejnowski, 1997); (Pouget and Snyder, 2000)</td>
<td>X</td>
<td>(Beyeler, Dutt and Krichmar, 2016)</td>
</tr>
<tr>
<td>Auditory cortex</td>
<td>X</td>
<td>?</td>
<td>(Hromádka, DeWeese and Zador, 2008)</td>
<td>?</td>
<td>?</td>
</tr>
<tr>
<td>Olfactory cortex</td>
<td>X</td>
<td>?</td>
<td>(Koulakov and Rinberg, 2011)</td>
<td>?</td>
<td>(Moreno-Bote and Drugowitsch, 2015)</td>
</tr>
<tr>
<td>Retrosplenial cortex</td>
<td>X</td>
<td>?</td>
<td>(Alexander and Nitz, 2015)</td>
<td>X</td>
<td>Present paper</td>
</tr>
<tr>
<td>Motor cortex</td>
<td>X</td>
<td>X</td>
<td>(Graziano and Aflalo, 2007); (Turner and DeLong, 2000)</td>
<td>?</td>
<td>(Vargas-Irwin, Shakhnarovich, Yadollahpour, Mislow, Black and Donoghue, 2010)</td>
</tr>
<tr>
<td>Basal ganglia</td>
<td>X</td>
<td>?</td>
<td>(Bar-Gad, Morris and Bergman, 2003); (Bar-Gad, Havazelet-Heimer, Goldberg, Ruppin and Bergman, 2000)</td>
<td>X</td>
<td>(Bar-Gad, Morris and Bergman, 2003), advanced RDDR</td>
</tr>
<tr>
<td>Barrel cortex</td>
<td>X</td>
<td>?</td>
<td>(Kerr, De Kock, Greenberg, Bruno, Sakmann and Helmchen, 2007)</td>
<td>?</td>
<td>?</td>
</tr>
</tbody>
</table>
faces). Similar to the preferred stimuli of neurons in the inferotemporal cortex (Wachsmuth, Oram and Perrett, 1994), these basis images resembled parts of faces, which could be additively combined to represent a whole face. In this experiment, the network was trained on a database of 2,429 facial images, which was encapsulated by 49 basis components in the matrix $W$ produced by the model.

Remarkably, such a parts-based representation is not specific to information processing in inferotemporal cortex, which is a high order cortical area in the “what” visual pathway (Beyeler, Dutt and Krichmar, 2016). The same principle can be extended to other areas of the visual system, such as the medial superior temporal dorsal (MSTd) area, which is part of the visual motion pathway. Neurons in MSTd respond to relatively large and complex patterns of retinal motion (‘optic flow’), owing to input from direction and speed selective neurons in the medial temporal (MT) lobe (for a recent review, see (Orban, 2007)). Although MSTd had long been suspected to be involved in the analysis of self-motion, the complexity of neuronal response properties has made it difficult to experimentally investigate how neurons in MSTd might perform this function. However, when Beyeler and colleagues (Beyeler, Dutt and Krichmar, 2016) applied NMF to MT-like patterns of activity, they found a sparse, parts-based representation of retinal flow (Figure 3.1B2) similar to the parts-based representation of faces encountered by Lee and Seung (Lee and Seung, 1999). The resulting “basis flow fields” showed a remarkable resemblance to receptive fields of MSTd neurons, as they preferred an intricate mixture of 3D translation and rotational flow components in a subset of the visual field. As a result, any flow field that could possibly be
encountered during self-movement through a 3D environment could be represented by a linear combination of only 64 basis flow fields.

Analogously, the NSC principle can explain response properties of neurons outside the visual system, such as in the RSC, an area important for navigation and spatial memory (Miller, Vedder, Law and Smith, 2014); (Nelson, Hindley, Pearce, Vann and Aggleton, 2015); (Vann, Aggleton and Maguire, 2009). Neurons in the RSC conjunctively encode multiple variables related to the environment and one’s position and movement within it, allowing the representation of spatial features of the environment with respect to multiple reference frames (Alexander and Nitz, 2015). However, establishing a mechanistic link between physiological response properties of RSC neurons and their underlying representations of space has proved difficult, due to the complexity of their response properties and because inputs to the region are not easily isolated. Yet by applying NSF to idealized input neurons that encoded experimentally recorded behavioral variables associated with rats running along a track, we were able to replicate functionality observed in the biological RSC (Figure 3.1B3). Once again, the dimensionality was massively reduced from a set of 417 input neurons that encoded approximately 445,000 stimulus instances to a set of 30 basis functions. The encoding of each stimulus column was also sparse, with an average of 3.27 (± 3.66) basis vectors critically participating in the reconstruction of each column \(v_i\) in \(V\). This was defined as the number of elements in \(h_i\) (of matrix \(H\)) that were greater than the average of all elements in \(h_i\) for every stimulus presentation \(i\) in the testing set. All basis vectors contributed equally for approximately 30% of the columns in \(V\).

Although there seems to be a consensus that information-theoretic explanations are relevant when investigating the early visual system, higher-order brain areas are often
considered to be specialized for performing tasks (e.g., recognizing objects, making
decisions, navigating an environment), rather than efficiently encoding of information. 
However, the finding that the NSC framework could be used to explain neuronal responses
across the visual and retrosplenial cortices introduces the possibility that it might apply
elsewhere in the brain. In fact, sparse (and potentially parts-based) representations have
been reported not only in visual cortex, but also in olfactory, auditory, somatosensory, and
premotor cortices (Table 3.1). This introduces the possibility that the NSC framework
might in fact be a general principle to which neuronal computations adhere.

**Understanding Neuronal Response Properties Within the Framework of NSC**

Neuronal response properties can also be computed for model neurons with the
NSC framework, which means that model neurons can be evaluated using methods similar
to the ones employed by experimental researchers to understand biological neurons, and
by theoretical neuroscientists to understand computational models. This is important
because it means that NSC can be used to model neural activity in the brain, and the
resulting activity patterns generated by NSC can be compared to and evaluated against
experimental findings. Thus, NSC can be interpreted as a paradigm implemented in
biological brains.

Neuronal response patterns can be computed in a way that is similar to
reconstructing the activity of a population of “visible” neurons (i.e., column s in \( \mathbf{V} \)) from a
population of “hidden” neurons (i.e., column s in \( \mathbf{H} \)) via their synaptic weights (i.e., the
matrix \( \mathbf{W} \); Figure 3.2A). It is possible to calculate \( \mathbf{H} \) from \( \mathbf{W}^T \) and \( \mathbf{V} \). In this context, a single
element of \( \mathbf{H} \) corresponds to the activity of a particular model neuron \( b \) to a particular
stimulus \( s \), which is given by the dot product of its presynaptic connections (i.e., column \( s \) in
and the corresponding synaptic weights (i.e., row \( b \) in \( W^T \)). Note that the response of the model neuron to different stimuli \( s \) involves different columns of \( H \) and \( V \), but always relies on the same weight matrix \( W \). Thus we can utilize \( W \) (which must remain fixed once learned with NMF) to simulate a model neuron's response to arbitrary input stimuli by replacing the column in \( V \) with new input. This allows us to investigate the response properties of individual model neurons much in the same way that experimental neuroscientists study biological neurons.

For example, subjecting model MSTd neurons to simulated optic flow fields mimicking natural viewing conditions during locomotion, Beyeler and colleagues (Beyeler, Dutt and Krichmar, 2016) found that individual model neurons (i.e., elements of the matrix \( H \)) preferentially responded to a particular 3D direction of self-translation or self-rotation,

\[
\begin{align*}
V &\text{ and the corresponding synaptic weights (i.e., row } b \text{ in } W^T. \\
\text{Note that the response of the model neuron to different stimuli } s \text{ involves different columns of } H \text{ and } V, \text{ but always relies on the same weight matrix } W. \text{ Thus we can utilize } W \text{ (which must remain fixed once learned with NMF) to simulate a model neuron's response to arbitrary input stimuli by replacing the column in } V \text{ with new input. This allows us to investigate the response properties of individual model neurons much in the same way that experimental neuroscientists study biological neurons.}
\end{align*}
\]
much like individual neurons in macaque MSTd (Figure 3.2B). In addition, known statistical properties of the MSTd population emerged naturally from the NMF based model, such as a relative over-representation of lateral headings.

Other groups have successfully applied NSC-like paradigms to model neuronal response properties. For example, in the visual system, an NMF-based model was able to reconstruct neuronal spike trains in the salamander retina (Onken, Liu, Karunasekara, Delis, Gollisch and Panzeri, 2016). Following testing on known ground truth, the researchers recorded spikes from in vitro retinal ganglion cells while the cells were exposed to natural images (either still photographs or videos). They then applied several factorization methods to the data. Space-by-time NMF could decompose the data into separate spatial and temporal modules that yielded sparser and more compact representations compared to other techniques, including orthogonal Tucker-2 and basic NMF. NSC could be used to reproduce response properties of V1 simple and complex cells (Hoyer, 2003); (Hoyer and Hyvärinen, 2002) as well as V2 hypercomplex cells (Hyvärinen, Gutmann and Hoyer, 2005). Outside the visual stream, a model known as Reinforcement-Driven Dimensionality Reduction (RDDR) successfully used Hebbian learning to reproduce input patterns associated with reward (Bar-Gad, Havazelet-Heimer, Goldberg, Ruppin and Bergman, 2000), a function associated with cortico-striato-pallidal circuitry. The authors later applied nonnegativity constraints to the Hebbian learning in the model so that it performed NMF on its inputs. The model advanced understanding of the cortico-striato-pallidal loop by capturing behavior of the circuit while explaining the existence of convergent and lateral connections in the region that other models have historically ignored (Bar-Gad, Morris and Bergman, 2003), which led to new predictions that can be
verified experimentally. The authors suggest that the basal ganglia uses unsupervised, reward-driven learning to perform dimensionality reduction on cortical inputs for the efficient compression of information in order to plan upcoming actions in the frontal cortex.

The finding that NSC can account for neuronal response properties across brain areas and modalities supports the view that these single-unit preferences are not due to any hard-wired neuronal specificity, but instead emerge from the pressure to find efficient representations of perceptually or behaviorally relevant stimulus spaces. At the population level, NSC promotes representations in which neurons act as generalists rather than specialists, allowing for the simultaneous encoding of multiple variables of interest (e.g., heading, eye rotation velocity in MSTd (Beyeler, Dutt and Krichmar, 2016)) with respect to multiple frames of reference (e.g., egocentric, route-based in RSC (Alexander and Nitz, 2015)). Among the advantages of such basis function representations (Pouget and Sejnowski, 1997); (Pouget and Snyder, 2000); (Poggio, 1990) (also called mixed-selectivity representations (Eichenbaum, 2017); (Fusi, Miller and Rigotti, 2016); (Barak, Rigotti and Fusi, 2013)) are robustness to noise as well as the ability to decode various variables of interest by taking a linear combination of neuronal responses.

**Approximating NMF with STDPH**

That NSC can explain and reproduce response properties observed in biological neurons may be an important clue as to how brains have evolved to parse and store information. In fact, we propose that this seeming coincidence derives from the fact that NSC and the brain take the same approach to handling information, and may be employing similar functional mechanisms to do so. One candidate NSC process in the brain is synaptic
plasticity, which may act on synapses for the purpose of reducing dimensionality on an input space in order to represent it efficiently and sparsely, the same as NSC. We propose that NSC and STDPH may indeed be functionally equivalent processes. STDP is a spike timing based correlative learning rule that has been shown to shape how input streams are represented in the dendrite, essentially partitioning the dendritic tree in terms of synaptic efficacy (Iannella, Launey and Tanaka, 2010). It acts on specific synapses over a short time scale (e.g., milliseconds to seconds). Homeostasis, which keeps neuronal activity in a good working range, acts over a longer timescale (e.g., minutes to days) and is applied to multiple synapses on a neuron or over multiple neurons (Turrigiano, Leslie, Desai, Rutherford and Nelson, 1998). Homeostasis also facilitates synaptic competition by normalizing the inputs to a neuron. This evens the playing field for synapses that weaken due to imbalances in activity, but might otherwise strengthen if left to their own devices (Chistiakova, Bannon, Chen, Bazhenov and Volgushev, 2015).

Experimental and theoretical evidence suggest that biological processes, such as synaptic plasticity and homeostatic mechanisms, may be reducing the dimensionality of inputs in a similar way to NMF. For example, Carlson and colleagues (Carlson, Richert, Dutt and Krichmar, 2013) delivered a mathematical proof that STDPH can approximate the NMF algorithm. Similar to Oja’s rule (Oja, 1982), which was developed to stabilize rate-based Hebbian learning (effectively resulting in PCA), synaptic scaling acts as a homeostatic mechanism to stabilize STDP (effectively resulting in NMF). This finding suggests that neurons are able to find accurate factorial representations of their input stimulus spaces through means of Hebbian-like synaptic learning. In addition, sparsity of the encoding
might be enforced by spike thresholding (Rozell, Johnson, Baraniuk and Olshausen, 2008) and lateral inhibitory connections (Coultrip, Granger and Lynch, 1992).

**Equivalence of NMF to STDPH and Neural Mechanisms of NSC**

Spike-timing dependent plasticity (STDP) is a form of Hebbian learning in which synaptic weight changes depend on the relative timing between pre- and post-synaptic spikes. If pre-synaptic spike counts integrated over a critical window precede those of the post-synaptic neuron, then long-term potentiation (LTP) is induced, strengthening the weight. Otherwise, long-term depression (LTD) is induced, suppressing the weight (Bi and Poo, 1998); (Song, Miller and Abbott, 2000). STDP is ubiquitous throughout the brain across the lifespan and can take on different forms (Caporale and Dan, 2008); (Holtmaat and Svoboda, 2009). Theoretical studies have shown that STDP can result in high firing rates that can kill or damage neurons. To obviate this problem, the brain uses homeostatic mechanisms that modulate synaptic input neuronal firing thresholds (Watt and Desai, 2010). We focus on synaptic scaling, a particular form of homeostasis which multiplicatively scales weights up or down depending on the average firing rate of the postsynaptic neuron, which has been demonstrated to stabilize STDP. We refer to this combination of STDP and homeostatic synaptic scaling as STDPH.

Carlson and colleagues (Carlson, Richert, Dutt and Krichmar, 2013) have shown that, given a network with fewer output than input neurons and full connectivity from the input layer to the output layer, STDPH iteratively acts to preserve the information in the first layer with the output layer neurons. Given an input layer of neurons (represented by matrix $V$) connected to a single output layer neuron (represented by row vector $h^T$) we have proven that the STDPH rule iteratively updates the synaptic weights in $w$ in such a
way as to minimize the reconstruction error of $|V - w^T h|$ that the NMF update rule is mathematically equivalent to the STDPH update rule (Carlson, Richert, Dutt and Krichmar, 2013).

To induce sparsity in networks of neurons, there must be a mechanism that causes each output layer to learn a different component of the total information represented by the input layer (Földiak, 1990), which may be achieved by competition among units. Lateral inhibition is often used to induce competition in SNNs, since it offers a biologically plausible mechanism to implement a WTA architecture (Coultrip, Granger and Lynch, 1992), although other groups have suggested that spike thresholding can also achieve sparse codes (Rozell, Johnson, Baraniuk and Olshausen, 2008). Therefore, NSC can be attained in a two-layer network with STDPH and a WTA architecture via lateral inhibitory connections within the output layer. However, popular implementations of NMF (e.g., MATLAB, Scikit-Learn) include a constraint on the L2 norm of H that automatically lead to sparsity. Alternatively, an explicit sparsity level can be incorporated into models of NMF, such as in Hoyer’s model (Hoyer, 2004).

**Computational Evidence Supports STDPH Approximation of NSC**

To investigate this equivalence, we conducted two experiments in which a model based on STDPH and a model based on NSC were applied to a dataset of recorded neuronal activity observed in the RSC of rats in a spatial navigation task (Alexander and Nitz, 2015). In this task, rats ran back and forth on a W-shaped track that could occupy different spatial locations within the room. During the experiment, 228 RSC neurons were recorded along with four behavioral metrics linear velocity (LV), angular velocity (AV), head direction
(HD), and position (Pos). Using Gaussian and cosine tuning curves, we created idealized input neurons that encoded these four variables.

For the STDPH experiment, the idealized input neurons generated spike trains as input to a population of SNNs that were trained using an evolutionary strategy to match the recorded electrophysiological data. The evolutionary algorithm was used to optimize the metaparameters associated with synaptic plasticity rules. During each generation of the evolutionary run, models were trained on trials randomly selected from half of the available dataset, then tested on trials randomly selected from the latter half reserved for testing. At the end of each generation, synthetic neural activity was correlated and matched to electrophysiologically recorded neural activity, which served as the fitness metric for the algorithm. After the model finished an evolutionary run, SNN activity was recorded and evaluated using the same analyses as reported in Alexander and Nitz (Alexander and Nitz, 2015); most importantly, neurons were classified into functional categories, and population behavior was assessed. For further details about the evolutionary algorithm and method, see Chapter 1.

For the NSC experiment, we constructed an input matrix containing half the trials associated with the task to be used as a training set, in which each row was one of these four features, and each column was associated with an element in one of the recorded trials. We applied NMF to the input matrix and then used the remaining half of the trials to test the model and gather responses from the model neurons, which were subsequently evaluated using the same methods as for activity recorded from the SNNs in the STDPH experiment.

We found that the activity patterns of both NSC and STDPH model neurons could
replicate the response properties seen in the electrophysiologically recorded neurons in the dataset (Figure 3.3B, C). That is, the model neuron activity could be classified into three broad categories, with remarkably similar population statistics to rat RSC (Rounds, Scott,
Alexander, DeJong, Nitz and Krichmar, 2016); (Alexander and Nitz, 2015): 1) neurons responding exclusively whenever the animal made a left or right turn on the track (light gray); 2) neurons responding whenever the animal made a turn, with increased firing whenever the animal occupied a specific location on the route, independent of allocentric location (dark gray); and 3) neurons that could not be classified according to the above, but nonetheless exhibited complex and robust firing patterns (white).

In addition, both NSC and STDPH model RSC produced simulated neurons whose ensemble activity vectors could be used to predict the agent’s location on a route with respect to the allocentric frame of reference. Ensemble activity patterns could also disambiguate the agent’s location within routes that occupied different locations in the room, consistent with findings of population behavior in the biological RSC (Alexander and Nitz, 2015). This is shown by bar plots of average reconstruction error, in which correlations between ensemble activity vectors produced a prediction line that lay along the points of highest correlation for each point along the track. When same-track position trials were correlated, prediction error was very low, but when conflicting-track position trials were correlated, prediction error was significantly higher in all three cases (Figure 3.3A-C, right); for further details on this analysis, see Chapters 1 and 2 and Alexander and Nitz (2015).

These findings lend credence to our proposal that NMF and STDPH are functionally equivalent. Indeed, in the few examples where synaptic weight matrices of biological neurons are available, such as from reverse correlation experiments in V1 (Smyth, Willmore, Baker, Thompson and Tolhurst, 2003), both NMF-based methods (Hoyer, 2003) and STDPH-based methods (Carlson, Richert, Dutt and Krichmar, 2013) yielded similar
results (Figure 3.4). Similarly, the matrix $W$ produced by NMF when applied to the RSC dataset was qualitatively similar to the synaptic weight matrices produced by experiments in which STDPH was evolved to match the same dataset. This preliminary evidence suggests that there is merit to comparing synaptic weights generated with both NMF and

**Figure 3.4. Receptive field generated by NSC and STDPH are qualitatively similar. A)** Receptive fields generated through reverse correlation experiments in ferrets, an NMF-based model and through STDPH yield qualitative similarities. Because the STDPH experiment used images of gratings instead of natural images, as in the ferret experiments and NMF model, the receptive fields are higher contrast, more organized, and have multiple gratings. **B)** Synaptic weight matrices generated by STDPH (top) and NMF (bottom). Although the STDPH weight matrices are much noisier, similar structure emerges.

STDPH to synaptic connections in the brain. In cases where synaptic weights from biological neurons are unavailable, simulated weight matrices could be used to advance theories of brain areas with complicated neuronal responses (such as MSTd or RSC); for example, by predicting neuronal responses to novel stimuli, which can be tested empirically.

**Discussion**
In this article, we have argued that a variety of neuronal response properties can be understood not as a consequence of hard-wired neuronal specificity, but as an emergent property of efficient population coding based on dimensionality reduction. We offer three testable predictions of this theory: First, we predict that parts-based representations can explain the receptive fields of neurons in a variety of brain regions, including but not limited to those brain areas discussed here (i.e., V1, MSTd and RSC). In agreement with the literature on basis function representations (Pouget and Sejnowski, 1997); (Pouget and Snyder, 2000); (Poggio, 1990), we expect parts-based representations to be prevalent in regions where neurons exhibit a range of tuning behaviors (Beyeler, Dutt and Krichmar, 2016), display mixed selectivity (Raposo, Kaufman and Churchland, 2014); (Rigotti, Barak, Warden, Wang, Daw, Miller and Fusi, 2013); (Fusi, Miller and Rigotti, 2016), or encode information in multiple reference frames (Alexander and Nitz, 2015).

Second, where such representations occur, we expect the resulting neuronal population code to be sparse, in order to encode information both accurately and efficiently. Sparse codes are ideally suited for such an encoding, as they offer a trade-off between dense codes (where every neuron is involved in every context, leading to great memory capacity but suffering from cross talk among neurons) and local codes (where there is no interference, but also no capacity for generalization) (Spanne and Jörntell, 2015).

We also speculate that there may be an important relationship between sparse and distributed coding. Barnes et al. (Barnes, McNaughton, Mizumori, Leonard and Lin, 1990) observed that output from the hippocampus was expressed as a much more distributed code, while the coding scheme within the hippocampus proper was sparse. They suggested
that sparse coding might be employed to increase the storage capacity of the region, consistent with our proposed framework. Sparse and distributed codes may represent two sides of the same coin, evidenced by anatomical bottlenecks, in which the brain utilizes synaptic expansion when transmitting information between brain regions and synaptic reduction and sparsification for information storage within regions.

Third, we propose that STDPH is carrying out a similar function to NMF, and may be attempting to approximate the linear receptive field properties of neurons participating in sparse, parts-based representations throughout the brain. STDPH and NMF can effectively produce NSC. With the emergence of computational tools developed to understand the neural code in high stimulus dimensions (Pillow and Simoncelli, 2006), we expect to see qualitative similarities between empirically observed receptive fields and those recovered by NMF and STDPH. Such findings would be consistent with the idea that neurons can perform statistical inference on their inputs via Hebbian-like learning mechanisms (Carlson, Richert, Dutt and Krichmar, 2013); (Moreno-Bote and Drugowitsch, 2015); (Oja, 1982); (Nessler, Pfeiffer and Maass, 2009).

In summary, we suggest that the evidence reviewed in this paper may indicate that dimensionality reduction through nonnegative sparse coding, implemented by synaptic plasticity rules, may be a canonical computation throughout the brain. Initial experiments using STDPH and NSC indicate that they may be functionally equivalent processes, which is backed by computational evidence in which NSC and STDPH were applied to an experimentally recorded dataset. In summary, NSC offers a promising theoretical framework to further our understanding of how high-dimensional data is encapsulated by the often complex and nonlocal nature of cortical computation.
References


EICHENBAUM, H. 2017. Barlow versus Hebb: When is it time to abandon the notion of feature detectors and adopt the cell assembly as the unit of cognition? *Neuroscience Letters*.


CONCLUSIONS

In this dissertation, I set out to develop a framework for evolving spiking neural networks (SNNs) that would enable synthetic neural activity to match electrophysiological neural activity to validate computational models of cognitive function. The reasons for this are twofold: First, as neuroscientists endeavor to understand the nature of representation in the brain, computational models are of critical importance. In a SNN, every neuron can be recorded, which is not currently possible in animal experiments, making them a method more conducive to examining the structure and function of a neural circuit. However, although SNNs incorporate neuron models with biologically realistic dynamics and learning rules, they are not typically validated against experimental data. If a model cannot match the firing patterns observed in the brain region, then the model cannot be said to fully encapsulate the behavior of the brain region under study. Furthermore, computational models that can be validated at the level of neuron function can provide insight into the nature of neuronal representations. Second, SNNs can be implemented in hardware and could potentially be implemented in neuromorphic chips to be used as neuroprosthetics to replace lost or damaged brain tissue. However, this necessitates that they can accurately decode incoming inputs and must therefore be capable of matching neuronal firing patterns.

I also set out to develop a virtual testbed for evolved models that can be used to conduct virtual experiments, which may provide a fruitful avenue for collaboration between experimental and theoretical neuroscientists. The evolutionary framework and virtual testbed can be used to evolve models that match datasets provided by experimentalists, which can then be tested under simulated experimental conditions to
generate new hypotheses and predictions that can then be tested in animal experiments. This way, experimentalists may save time and resources since the models can provide guidance as to what should be examined in the brain, and theoreticians can further validate and refine their models by testing them against new data. The evolutionary framework and virtual testbed were both applied to a dataset from rat retrosplenial cortex (RSC), a brain region known for its contribution to navigation and memory (Miller, Vedder, Law and Smith, 2014); (Takahashi, Kawamura, Shiota, Kasahata and Hirayama, 1997); (Nelson, Hindley, Pearce, Vann and Aggleton, 2015); (Vann, Aggleton and Maguire, 2009). Using the framework, I evolved SNNs that replicated features of the RSC, and provided a number of new hypotheses and predictions about the nature of spatial representation in this region.

Lastly, in this dissertation I set out to understand the nature of representation in the brain. To this end, I helped to develop a framework for dimensionality reduction in the brain, motivated by the observation that the brain must, on a daily basis, handle tremendously high-dimensional input under tight metabolic constraints. My colleagues and I found that a statistical linear dimensionality reduction technique called nonnegative matrix factorization (NMF) with sparsity constraints can account for response properties in multiple disparate brain regions, including RSC. We call this framework nonnegative sparse coding (NSC). We further found that a biologically plausible learning rule, spike-timing dependent plasticity with homeostasis (STDPH) can approximate this technique, suggesting that neuron responses might be a by-product of plasticity rules in the brain, and that dimensionality reduction may be a canonical computation employed generally to handle high-dimensional input spaces. In the final chapter of this dissertation, I will consider the contributions of my work and discuss directions for future research.
Contributions

In Chapter 1, I introduced a novel and generalizable framework for the replication of experimentally observed neuronal response patterns. This framework was used to successfully evolve SNNs that reproduced neuronal, functional, and population features of a dataset recorded from rat RSC (Alexander and Nitz, 2015) by using an indirect encoding scheme to evolve the learning parameters on the network. The framework can generalize to any dataset recorded under similar experimental conditions. The network takes as input a set of parameterized behavioral metrics, and synthetic neural activity is trained on those inputs over successive trials. Following training, the neuronal responses are evaluated with a user-defined function that matches synthetic activity patterns to electrophysiologically observed ones. The algorithm converged on a solution in approximately 20 generations and was applied to a population of 15 SNNs. This framework may represent a new approach to computational modeling that can rapidly tune large and complex SNNs.

In Chapter 2, I developed a virtual testbed for the models produced by the evolutionary framework. The virtual testbed was applied to the SNNs evolved to match the RSC data. In a novel virtual testbed, I simulated a variety of experimental manipulations to probe the networks’ representation of space. The experiments, which involved a series of lesion and track manipulation experiments, revealed important insights into the nature of the RSC code. First, they provide confirmation that the RSC conjunctively encodes multiple reference frames. Second, the SNNs were resilient to damage of their inputs, and were able to produce coherent representations of space except in the face of catastrophic information loss (when nearly all inputs were lesioned). Third, we found that the evolved SNNs learned associations between its inputs and could spontaneously generate stable, coherent
representations for novel routes configurations. If inputs were altered inconsistently, the networks could not produce coherent representations, which may have significant implications for the nature of representation in the region, and supports its proposed role as an association cortex. Finally, we found that synthetic neuronal responses in the region were flexible and adaptable. They spontaneously produced coherent representations for routes and novel route configurations never before experienced during the training process, suggesting that the RSC may be capable of flexibly responding to changing task demands by conjunctively representing multiple task relevant variables. This functionality has been observed in higher cortical regions such as the parietal cortex (Raposo, Kaufman and Churchland, 2014) and supports the notion that neurons do not have specialized functionality, but instead demonstrate mixed selectivity to be flexible in the face of changing, high-dimensional input spaces (Fusi, Miller and Rigotti, 2016); (Eichenbaum, 2017); (Barak, Rigotti and Fusi, 2013).

The framework itself is significant because it will enable collaboration between experimentalists and theoretical neuroscientists. I have, for example, used the virtual testbed to generate new hypotheses about functionality in the RSC, which can be tested in animal experiments, the outcome of which can be used to further refine and test the evolved models. Because the framework is flexible and generalizable, this framework can, in theory, be applied to virtually any brain region. The evolutionary framework and testbed together may represent an entirely new way of modeling cognitive function. Our model was also robust to perturbation and loss of input. This framework may help to overcome these challenges associated with computational modeling.
In Chapter 3, I advanced the NSC framework for understanding neuronal response properties and receptive fields in the brain. In this chapter, my colleagues and I propose that the goal of representation in the brain may be to perform dimensionality reduction to handle complex, high-dimensional input spaces. Inspired by work by Lee and Seung (Lee and Seung, 1999) in which NMF was used to learn parts-based representations of facial images, we revisit a proof showing that a particular implementation of spike-timing dependent plasticity with homeostasis (STDPH) can approximate NMF (Carlson, Richert, Dutt and Krichmar, 2013). We expand upon prior simulation results showing that functionality in MSTd could be reproduced through the application of NMF to MST-like inputs (Beyeler, Dutt and Krichmar, 2016). By creating an NMF-based model of the RSC dataset, we were able to reproduce features and functionality similar to that observed in the dataset and in the evolved SNNs. Furthermore, we review evidence that NMF- and NSC-based techniques can account for response properties in multiple disparate brain regions, including the basal ganglia (Bar-Gad, Morris and Bergman, 2003). Our results suggest that dimensionality reduction, implemented by STDPH, may produce sparse and parts-based representations all over the brain, and we propose that our framework, nonnegative sparse coding, maybe a canonical computation in the brain for handling complex inputs.

**Future Work**

There are currently several limitations with respect to the evolutionary framework. The most obvious of these is that it has yet to be validated against multiple datasets – although it can theoretically generalize to any dataset recorded under similar experimental conditions to Alexander and Nitz (Alexander and Nitz, 2015), it has yet to be fully tested on other datasets. In a similar vein, the limitations of the framework are currently unknown.
The task performed by the rats associated with the RSC dataset was a simple navigation task that featured no decision-making or reward-driven learning. One avenue for future research is to evolve multiple brain regions to perform one task (for example, using the framework to evolve a model to match hippocampus, parietal cortex, and retrosplenial cortex data simultaneously while performing a task such as the Morris water maze (Morris, 1984).

Another avenue for future research involves testing the network to see how many neurons can be added to a SNN before it cannot holistically reproduce the functionality observed in the dataset. In the RSC experiment, each SNN had 600 neurons in total (480 excitatory, 120 inhibitory), but only 228 of them could be matched to an electrophysiological counterpart. Yet over the course of evolution, the network as a whole could reproduce the population behavior and functional responses of the dataset, regardless of whether or not the neuron was an explicitly matched neuron. This level of flexibility is potentially very powerful, but it is as of yet unknown whether or not the same results could be obtained by evolving, for example, 1,000-2,000 neurons to match a dataset of only 200 electrophysiological neurons. If large-scale networks can be evolved in the same manner as in the RSC experiments, then this form of modeling is potentially even more valuable.

A related goal for the framework is to develop better and more accurate fitness functions. Over the course of the development of the evolutionary framework, several ways of measuring fitness were examined, including Euclidean distance and Mean Squared Error (MSE) metrics. Ultimately, relying on correlation and maximum firing rate thresholds yielded the best qualitative neuronal matches and facilitated rapid evolution, whereas
other objective functions we tried did not improve at all within the first ten generations of evolution. However, correlations between neuronal matches have limited utility as a fitness metric. Although correlations ensure that synthetic neuronal activity patterns functionally match those of recorded biological neurons, they do not guarantee that the firing rates of both neurons are comparable (i.e., a synthetic neuron might fire at a higher or lower rate than its matched counterpart). This can be mitigated to some extent by incorporating firing rate thresholds into the objective function, but in general, more accurate fitness functions must be developed to ensure accurate and precise matches between synthetic and biological neuronal activation patterns. One possible solution is to incorporate upper and lower bounds on activity for each synthetic neuronal match individually. Another possible solution is to focus on replicating spike trains instead of mean firing rates integrated over a specified window. Because this is a more precise metric for quantifying the behavior of an experimentally observed neuron, this avenue might do much to improve the accuracy and precision of replicated activation patterns.

One criticism levied at the evolutionary framework is that because the evolved network architecture is generic (10% connectivity, 80% excitatory/20% inhibitory Izhikevich network), it cannot really validate neuronal behavior from a multitude of brain regions, which all exhibit different kinds of neural dynamics, synaptic plasticity rules, and even cell types. Thus, one avenue for future work involves incorporating connectivity constraints and neuron parameters that are more biologically realistic with respect to the area being modeled. This, of course, would necessitate the collection and analysis of connectivity and neuron types in the brain region being modeled. Relatedly, while it is known that areas projecting to the retrosplenial cortex represent behavioral variables
relating to position and action (e.g., the parietal cortex (Nitz, 2006); (Nitz, 2009); (Nitz, 2012); (Wilber, Clark, Forster, Tatsuno and McNaughton, 2014); (Cohen and Andersen, 2002); (Burgess, 2006)) we modeled the recorded variables as inputs using relatively simple Gaussian tuning curves. To make faithful idealized input neurons that project from the parietal cortex, hippocampus, and other areas, the recorded variables must be modeled in a way that is consistent with how they would be represented within those regions for our models to be truly veridical. The inputs as currently modeled are a sufficient approximation, but may not be entirely faithful to the biological brain.

Lastly, as has been mentioned briefly, an eventual goal of this framework is to produce SNNs that could eventually be implemented in hardware for the purpose of neuroprosthetics. Although this is a long way off, it is an avenue for future research, and a similar evolutionary methodology was used by Dura-Bernal et al. (2017) (Dura-Bernal, Neymotin, Kerr, Sivagnanam, Majumdar, Francis and Lytton, 2016) to develop a biomimetic neuroprosthetic model. In this study, SNNs were evolved (each containing 8,000 neurons) to perform a reaching task in response to recorded data from macaque primary motor cortex. The models could accurately replicate experimentally observed reaching movements and the evolved SNNs were used to successfully control a robotic arm.

There are also several avenues for future work regarding the virtual testbed component of the framework. In experiments designed to probe the function of SNNS evolved to match RSC data, the simulated navigation tasks were still relatively simple in that they were not goal-driven. However, this provides an interesting opportunity to assess the generalizability of the evolved SNNs. One possible experiment would involve designing a new virtual task (or even entirely different kinds of inputs than the ones it was trained
on), such a T-maze task, in which the neural agent would have to learn to turn left or right depending on some cue (e.g., a flash of light, as in Vedder et al. (2016) (Vedder, Miller, Harrison and Smith, 2016) to get a reward. The new network could be given a series of new trials associated with this task with synaptic plasticity re-enabled. If the network is capable of learning the new task while retaining its prior spatial representations associated with the previous task, then the model could be said to be generalizable. This avenue of research could help to overcome catastrophic forgetting, which remains a challenge for many computational models (French, 1999); (Kirkpatrick, Pascanu, Rabinowitz, Veness, Desjardins, Rusu, Milan, Quan, Ramalho and Grabska-Barwinska, 2017).

Relatedly, evolved models could be used to test theories about learning and memory. For example, the evolved models could be used to test Complementary Learning Systems theory (McClelland, 2013) and schema learning (Tse, Langston, Kakeyama, Bethus, Spooner, Wood, Witter and Morris, 2007); (Tse, Takeuchi, Kakeyama, Kajii, Okuno, Tohyama, Bito and Morris, 2011) by interleaving trials associated with new simulated tasks and trials associated with previous tasks to see if and how many tasks the network could learn before it experiences catastrophic interference. These models could also be used to investigate memory consolidation, especially if coupled with other evolved models such as the hippocampus or regions of the prefrontal cortex.

Lastly, there are many open questions associated with the NSC framework that are possible avenues for future research. NSC, or even NMF-based methods, have not been used to model response properties in many brain regions, despite the fact that the method is well suited to the task (Onken, Liu, Karunasekara, Delis, Gollisch and Panzeri, 2016). In fact, it has mostly been applied to data associated with the visual system; thus, to validate
its role in neural computation generally, it must be tested on very different regions. First and foremost, NSC should be tested by applying it to datasets from other brain regions, such as the olfactory and auditory cortex, or even regions such as the amygdala.

One important avenue for future research is mathematical in nature. Carlson and colleagues (Carlson, Richert, Dutt and Krichmar, 2013) showed in a mathematical proof that a particular implementation of STDPH was mathematically equivalent to NMF. However, this proof concerned only a single synapse, and may not necessarily extend to the network level. Therefore, more evidence is needed to strengthen the mathematical equivalence between STDPH and NMF, either through the generalization of the proof or through simulation studies such as the one conducted in RSC.

Another fundamental question in NSC concerns the presence of parts-based representations. NSC produces parts-based representations due to nonnegativity constraints, meaning that the resulting basis functions must be combined additively as a weighted sum. However, there is very little data available on the nature of receptive fields for the majority of regions in the brain. One area in which data is available is in primary visual cortex (V1); NSC models of V1 simple cell receptive fields are consistent with what is known about biological V1 simple receptive fields, suggesting that they are parts-based. This is encouraging evidence that there may be parts-based representations elsewhere in the brain, especially where sparse coding schemes are employed, and that comparisons between biological data, NSC models, and STDPH models may be fruitful for understanding neuronal representations. Relatedly, Bar-Gad et al. (Bar-Gad, Morris and Bergman, 2003) observed that nonnegativity constraints in the brain may not be hard constraints since inhibitory connections can effectively cancel out excitatory ones, which would equate to a
subtractive computation. However, Hoyer (Hoyer, 2003) interpreted the basis functions produced by a nonnegative and sparse model as containing both excitation and inhibition, but still found parts-based representations. One important avenue for research related to the NSC framework involves studying synaptic plasticity rules and inhibition in multiple brain regions to understand differences in the way dimensionality reduction might be implemented across the brain, and whether the mechanism is broadly consistent everywhere. Relatedly, it is an open question whether dimensionality reduction happens everywhere in the brain, or only in selected regions.

Another avenue for research involves examining neuronal responses to multiple kinds of inputs. In the NSC framework, my colleagues and I investigated a linear dimensionality reduction technique. However, neuronal behavior is high-dimensional and nonlinear, so it is possible that linear dimensionality reduction cannot produce sufficiently rich and sophisticated compressed representations of high-dimensional input that they are functionally and behaviorally relevant. Thus, we propose that nonlinear dimensionality reduction techniques such as nonlinear NMF should be tested in the context of the NSC framework for the modeling of neuronal responses and receptive fields.

In conclusion, I have presented several frameworks for developing computational models of neuron functionality and cognition and for understanding neuronal representations, which I have applied to data from the biological RSC to advance our understanding of the region. The evolutionary framework and virtual testbed are important advances for the computational neuroscience community because they introduce a new and flexible way of modeling the brain that may be beneficial to both the experimental and theoretical communities. The NSC framework is an important advance
because it provides a new way of looking at neuronal response properties, and is a step toward a broader view of computation that advocates for neurons not as highly selective and specialized units for certain kinds of input, but instead as an inventive evolutionary innovation that allows the brain to handle incoming inputs by extracting as much information as possible from the input space using dimensionality reduction.
References


EICHENBAUM, H. 2017. Barlow versus Hebb: When is it time to abandon the notion of feature detectors and adopt the cell assembly as the unit of cognition? Neuroscience Letters.


