Bridging biophysics and emergent dynamics through deep learning assisted modeling of the neocortex

By

Nicholas M. Tolley

B.S., University of Washington, 2018

Sc.M., Brown University, 2024

Thesis

Submitted in partial fulfillment of the requirements for the Degree of Doctor of Philosophy in the Department of Neuroscience at Brown University

PROVIDENCE, RHODE ISLAND

October 2025

This dissertation by Nicholas Tolley is accepted in its present form by the Department of Neuroscience as satisfying the dissertation requirement for the degree of Doctor of Philosophy.

Date _____ _____

Stephanie R. Jones, Advisor

Recommended to the Graduate Council

Date _____ _____

Carlos Vargas-Irwin, Reader

Date _____ _____

John P. Donoghue, Reader

Date _____ _____

Christopher I. Moore, Reader

Date _____ _____

Gaute T. Einevoll, Reader

Approved by the Graduate Council

Date _____ _____

Thomas A. Lewis, Dean of the Graduate School

Curriculum Vitae

Education

2019–2025
Ph.D., Neuroscience
Brown University (Providence, RI, USA)

2021-2024
Sc.M., Data Science
Brown University (Providence, RI, USA)

2014-2018
B.S., Neurobiology and Biochemistry
University of Washington (Seattle, WA, USA)

Publications

Papers

Tolley N., Rodrigues P., Gramfort A., Jones S. R. (2024). Methods and considerations for estimating parameters in biophysically detailed neural models with simulation based inference. *PLOS Computational Biology.*

Jas, M., Thorpe, R., **Tolley, N.**, Bailey, C., Brandt, S., Caldwell, B., Cheng, H., Daniels, D., Pujol, C. F., Khalil, M., Kanekar, S., Kohl, C., Kolozsvári, O., Lankinen, K., Loi, K., Neymotin, S., Partani, R., Pelah, M., Rockhill, A., Sherif, M., Hamalainen, M., and Jones, S. R. (2023). HNN-core: A Python software for cellular and circuit-level interpretation of human MEG/EEG. *Journal of Open Source Software.*

Mondello, S. E., Young, L., Dang, V., Fischedick, A. E., **Tolley, N.**, Wang, T., Bravo, M. A., Lee, D., Tucker, B., Knoernschild, M., Pedigo, B. D., Horner, P. J., and Moritz, C. (2023). Optogenetic spinal stimulation promotes new axonal growth and skilled forelimb recovery in rats with sub-chronic cervical spinal cord injury. *Journal of Neural Engineering.*

Samejima, S., Ievins, A. M., Boissenin, A., **Tolley, N.**, Khorasani, A., Mondello, S. E., and Moritz, C. T. (2022). Automated lever task with minimum antigravity movement for rats with cervical spinal cord injury. *Journal of Neuroscience Methods.*

Samejima, S., Khorasani, A., Ranganathan, V., Nakahara, J., **Tolley, N.**, Boissenin, A., Shalchyan, V., Daliri, M. R., Smith, J. R., and Moritz, C. T. (2021). Brain-Computer-

Spinal Interface Restores Upper Limb Function after Spinal Cord Injury. *IEEE Transactions on Neural Systems and Rehabilitation Engineering.*

Meeker, S. M., Mears, K. S., Sangwan, N., Brittnacher, M. J., Weiss, E. J., Treuting, P. M., **Tolley, N.**, Pope, C. E., Hager, K. R., Vo, A. T., Paik, J., Frevert, C. W., Hayden, H. S., Hoffman, L. R., Miller, S. I., and Hajjar, A. M. (2020). CFTR dysregulation drives active selection of the gut microbiome. *PLOS Pathogens.*

Ranganathan, V., Nakahara, J., Samejima, S., **Tolley, N.**, Khorasani, A., Moritz, C. T., and Smith, J. R. (2019). NeuralCLIP: A Modular FPGA-Based Neural Interface for Closed-Loop Operation. *9th International IEEE/EMBS Conference on Neural Engineering (NER).*

Patents

Tolley, N., Jones, S. R. (2025). Estimation of parameters in biophysically detailed neural models with simulation based inference. *Provisional App. 63/567,108.*

Mortiz, C. T., Khorasani, A., Samejima, S., **Tolley, N.** (2023). High frequency epidural stimulation to control sensation. *US Patent App. 18/007,948.*

Software Contributions

HNN-core `https://github.com/jonescompneurolab/hnn-core`

Jaxley `https://github.com/jaxleyverse/jaxley`

Teaching

2024
Teaching Assistant, Dept. of Neuroscience, Brown University (Providence, RI, USA)
Course: *Mechanisms & Meaning of Neural Dynamics* (NEUR 1440)

2022, 2023, 2024
Mentor, Google Summer of Code
Projects: *Generalize parameter optimization routines*, and *Develop IO Routines for HNN-core outputs*, and *A GUI for HNN-core with ipywidgets and the calculation and visualization of CSD signals*, and *Implement batch simulation and optimization routine*

2022
Teaching Assistant, Dept. of Neuroscience, Brown University (Providence, RI, USA)

Course: *Advanced Systems Neuroscience* (NEUR 2050)

2021
Teaching Assistant, Dept. of Neuroscience, Brown University (Providence, RI, USA)
Course: *Quantitative Reasoning for Neuroscientists* (NEUR 2060)

2020, 2021
Mentor, *Summer Research Early Identification Program*, The Leadership Alliance, Brown University (Providence, RI, USA)
Projects: *Biophysical modeling of Alzheimer's disease mediated changes in magnetoencephalography (MEG) alpha oscillations*, and *Expansion of the Human Neocortical Neurosolver neural modeling tool to validate and inform modeling predictions with local field potential and current source density recordings*

Professional Development

Invited Lectures and Workshops Given

2024
The digital brain: bridging deep learning and biophysical models (lecture)
MEG Seminar Series, Athinoula A. Martinos Center for Biomedical Imaging Athinoula A. Martinos Center for Biomedical Imaging, Massachusetts General Hospital (Charlestown, MA, USA)

2024
Human Neocortical Neurosolver (HNN): An open-source software for cellular and circuit-level interpretation of human MEG/EEG (lecture)
Nanosymposium, 2024 Society for Neuroscience Annual Meeting (Chicago, IL, USA)

2024
Human Neocortical Neurosolver: A Software Tool for Cell and Circuit Level Interpretation of MEG/EEG signals (lecture)
Neuroscience Gateway - Modeling, Data Analysis and Software Dissemination on Supercomputers, Satellite Workshop - 2024 Society for Neuroscience Annual Meeting (Chicago, IL, USA)

2024
HNN-core: A Python software for cellular and circuit-level interpretation of human MEG/EEG (lecture)
15th International Conference on Bioelectromagnetism (Espoo, Finland)

2023
Fishing for beta: uncovering mechanisms underlying cortical oscillations in large-scale biophysical models (lecture)
ICERM Semester Program, Math + Neuroscience: Strengthening the Interplay Between Theory and Mathematics, Post Doc/Graduate Student Seminar (Providence RI, USA)

2023
Human Neocortical Neurosolver: A Software Tool for Cell and Circuit Level Interpretation of MEG/EEG signals (lecture)
Society for Neuroscience Annual meeting, Neuroscience Gateway - Software Dissemination, Large Scale Modeling, Data Analysis on Supercomputers Satellite Workshop (Washington D.C., USA)

2023
Applying simulation based inference in the Human Neocortical Neurosolver neural modeling software to study the mechanisms of electrophysiological biomarkers (lecture)
32nd Annual Computational Neuroscience Meeting, Neuro-Inspired Topology and Machine Learning Workshop (Leipzig, Germany)

2022
Human Neocortical Neurosolver: A Software Tool for Cell and Circuit Level Interpretation of MEG/EEG signals (lecture)
Society for Neuroscience Annual meeting, Neuroscience Gateway - Enabling Large Scale Modeling, Data Processing and Dissemination of Software Virtual Workshop (San Diego, CA, USA)

2022
Hands-on workshop on Human Neocortical Neurosolver: a new tool for cell and circuit level interpretation of MEG/EEG signals, with S. R. Jones, M. Jas, and R. V. Thorpe (workshop)
BIOMAG International Conference, University of Birmingham (Birmingham, UK)

2021
Human Neocortical Neurosolver (HNN) workshop, with S. R. Jones, M. Jas, and R. V. Thorpe (workshop)
CuttingEEG Annual Symposium, Le Cube Université Aix-Marseille (France)

2020
Neuroscience for Everyone: Neurotechnology (lecture and panel)
University of Washington Neurobiology Club (virtual)

Conferences

2024
NINDS DTR Therapeutic Development Projects: Innovation from Academic and Small Business Investigators Satellite Event (Chicago, IL, USA)
Poster: *Unlocking EEG-based insights for drug development with biophysical network simulations*

2024
BRAIN Initiative Investigators Meeting (Rockville, MD, USA)
Poster: *Human Neocortical Neurosolver (HNN): An open-source software for cellular and circuit-level interpretation of human MEG/EEG*

2024
Computational and Systems Neuroscience - COSYNE (Lisbon, Portugal)
Poster: *Ultrafast connectivity optimization of large-scale biophysical network models with deep learning*

2023
Society for Neuroscience Annual Meeting (Washington D.C., USA)
Poster: *HNN-core: A Python software for cellular and circuit-level interpretation of human MEG/EEG*

2023
Neural Control of Movement Annual Meeting (Victoria, BC, Canada)
Poster: *Compensating for missing neural information with surrogate sensory input*

2022
Society for Neuroscience Annual Meeting (Washington D.C., USA)
Poster: *Simulation based inference in the Human Neocortical Neurosolver software reveals a calcium-mediated decoupling between spiking and field activity underlying macroscale M/EEG event related potentials*

2021
Society for Neuroscience Annual Meeting (virtual)
Poster: *Simulation based inference in the Human Neocortical Neurosolver neural modeling software uncovers circuit mechanisms of primate motor cortical 15-29 Hz beta activity during motor planning*

Certificates

2024
Summer Workshop on the Dynamic Brain

Allen Institute (Friday Harbor, WA, USA)

2023
Math + Neuroscience: Strengthening the Interplay Between Theory and Mathematics
Institute for Computational and Experimental Research in Mathematics (Providence, RI, USA)

2023
Neurotechnology: from Research to Application
Ben-Gurion University of the Negev (Be'er Sheva, Israel)

2019
Computational Neuroscience Summer Course: Beyond the Local Circuit
Chinese University of Hong Kong (Hong Kong)

Awards and Fellowships

2021: Open Graduate Education Fellowship
Brown University

2021: Chateaubriand Fellowship
€2682, Embassy of France in the United States

Acknowledgments

Scientific works are often an idealized narrative pulled from a messy, chaotic, yet ultimately rewarding set of incursions into new intellectual domains. I have been exceedingly fortunate to have a strong community of friends, family, and colleagues to help me face these challenges head on, and stay motivated throughout the entirety of my doctoral studies. I would first like to thank my parents, whose love and support have carried me through every stage of my life, and will always be my role models for a healthy professional and family life. To my sister Meg, I am exceptionally lucky that you have always been looking out for me, and never fail to make me laugh when I need something to lift my spirits. I would also like to thank my good friend Enoch, who made the journey with me from Seattle to New England, and has been a huge source of support over the years. To my girlfriend Cat, you are one of the most caring, hard-working, and talented persons I have ever met. You are an endless source of motivation, and give me so much optimism for the future knowing I will be spending it with you.

Academically, I have had the pleasure to be mentored by some of the kindest and most intelligent people I have ever met. To my advisor Stephanie Jones, I am extremely grateful for all the guidance and opportunities you have given me. Your mentorship has truly been a highlight of my graduate school journey, and your work ethic and dedication to science is nothing short of inspiring. I sincerely look forward to our continued collaborations in the years to come. To Carlos and John, your scientific rigor and overall optimism have always stood out to me, and your mentorship over the years has been instrumental in

helping me become a better scientist. To Chris, our conversations have genuinely changed the way I think about the brain, and you will always be a role model for how to wield creativity to push science forward.

I am immensely appreciative of all of the lab mates and collaborators I have had over the years, notably Ryan Thorpe, Mainak Jas, Chloe Zimmerman Gunsilius, Katharina Duecker, David Zhou, Dylan Daniels, Danielle Sliva, and Darcy Diesburg, as well as everyone in my PhD cohort, whose conversations and friendship will be an unforgettable part of my time at Brown.

I would lastly like to thank the many colleagues I have had over the years who have generously donated their time to help me develop my scientific ideas. Notably Alexandre Gramfort, Pedro L. C. Rodrigues, Jason Ritt, Alexander Fengler, Ivan Rodriguez, and Lakshmi Govindarajan.

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

A central and unsolved problem in neuroscience is linking cell-level biophysical properties of neurons to network-level emergent properties of neural systems. Overcoming this challenge offers the ability to identify computational principles of neural function that are biologically interpretable, and provides a direct path towards translational applications of neuroscience research. As an example, the most common clinical tool to measure neural activity is electroencephalography (EEG). EEG is a non-invasive technique which provides a direct readout of network-level emergent phenomena (i.e. neural oscillations), and can be used to derive biomarkers of neuropsychiatric diseases like schizophrenia and depression (de Aguiar Neto and Rosa (2019); Salisbury et al. (2010)). Understanding the cell-level mechanisms underlying network-level emergent phenomena would advance our understanding of basic neuroscience, assist in developing targeted treatments for neuropsychiatric disease, and identify computational principles that can be translated to artificial intelligence (AI) systems.

To better understand the mechanistic links between cells and emergent neural dynamics, computational neuroscientists create models with varying levels of realism which can be simulated to produce experimentally testable predictions (Panahi et al. (2021); Einevoll et al. (2019); Abbott and Dayan (2001)). Abstract models like recurrent neural networks

can be used to model network-level neural dynamics, but they are insufficient for capturing cell-level mechanisms (Panahi et al. (2021)). Detailed biophysical models can be used to capture cell-level properties of neurons, however, building biophysical models that explain the impact of cellular physiology on network-level dynamics poses substantial challenges. The major challenges include 1) fitting model parameters to data (Gonçalves et al. (2020)), and 2) the high computational cost (Cremonesi and Schürmann (2020); Zhang et al. (2023)), which have both hindered the adoption of biophysical models by a broader scientific community. Fortunately, the massive expansion of deep learning in the last decade has led to novel solutions addressing these challenges, creating an opportunity for biophysical neural models to be used more widely.

In this thesis, I focus on computational models of the neocortex, and combine them with deep learning to establish mechanistic links between cell-level biophysical properties and network-level emergent dynamics. I specifically apply these techniques to understand the cellular properties that impact 1) neural oscillations measured by magneto- and electroencephalography (MEG/EEG), and 2) network dynamics that support working memory.

For the study of neural oscillations, I use the validated modeling framework the Human Neocortical Neurosolver, a biophysically detailed model of a cortical column that was specifically designed to link cell and circuit-level properties to MEG/EEG biomarkers (Neymotin et al. (2020); Jas et al. (2023)). Neural oscillations are one of the earliest studied emergent dynamics observed in neural activity, dating back to the seminal studies by Hans Berger, the inventor of EEG (Brigo (2025)). Since the introduction of EEG, neural oscillations have been a central topic of study, with numerous theories on their significance for brain-wide function (van Ede et al. (2018); Buzsáki and Draguhn (2004); Spitzer and Haegens (2017); Arnal and Giraud (2012); Pritchett et al. (2015); Klimesch et al. (2007); Voloh and Womelsdorf (2018)). Further, clinical studies have revealed that EEG measured biomarkers can serve as robust predictors of disease, as they directly

measure activity produced by neural circuits and can signal their potential dysfunction (de Aguiar Neto and Rosa (2019); Sakkalis (2011); Loo et al. (2016); Horvath et al. (2018)).

For the study of working memory, I developed a novel reservoir computing framework which employs cell and circuit-level properties inspired by the neocortex, and can be used to guide the development of AI systems informed by neuroscience principles. I chose working memory, as it is an essential building block for the vast array of cognitive functions exhibited by the human brain. Planning, abstract reasoning, attention, and many other functions all share the common problem of maintaining a persistent pattern of neural activity which encodes information to be remembered and transformed (Foster et al. (2024)). Unlike long-term memory, which is believed to be encoded in synaptic weights, it is generally accepted that working memory emerges from the ongoing activity of a network of neurons (Wang (2021)). Understanding the mechanistic origins of working memory is an essential first step towards linking cell-level properties to more complex cognitive functions. Similar to the study of neural oscillations, mechanistic models of working memory have the potential to inform targeted treatments for neuropsychiatric disease that exhibit working memory deficits (Murray et al. (2014); Geerts et al. (2021, 2013)). A future goal will be to connect these lines of research, as neural oscillations are readily observed to correlate with working memory (Hsieh et al. (2011); Riddle et al. (2020); Roux and Uhlhaas (2014); Pina et al. (2018)), providing ample opportunity to explore the cellular origins of these two fundamental emergent dynamics.

While neural oscillations and working memory are two of many hallmark emergent dynamics of the neocortex, the techniques developed in this thesis set the stage for future work on the mechanistic underpinnings of numerous other cortical functions.

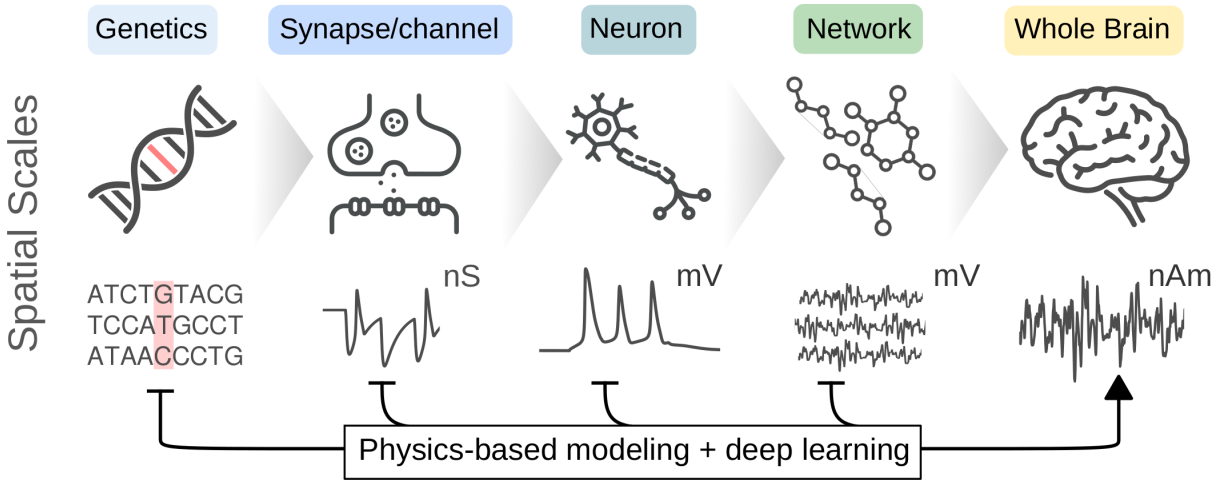

Figure 1.1: **Bridging scales with physics-based modeling and deep learning**. Brain activity is the product of an extensive hierarchy of spatial scales. At the level of DNA, differential gene expression in a neuron leads to the addition/removal of proteins which impact the electrical properties of the neuronal membrane (i.e. biophysics). The biophysical properties of proteins like ion channels combine at the neuron-level to produce behavior such as spiking. At the network-level, interconnected neurons produce complex activity patterns measured in the form of local field potentials. Finally, at the whole-brain level, an ensemble of networks act in concert to fulfill the extensive array of functions carried out by the nervous system. A central challenge in computational neuroscience is creating models that bridge these spatial scales. Currently, a highly promising approach to overcoming this challenge is equipping physics-based neural models with deep learning algorithms.

1.1 Emergent properties as an organizing principle of computational neuroscience

An **emergent property** describes the phenomenon where a system of many interacting elements gives rise to behavior that is not apparent from the properties of any individual element (O'Connor (2020); Morrison (2012); Salt (1979)). "More is different" (Anderson (1972)) rather cleanly captures this concept. The most compelling questions in neuroscience can often be formulated as understanding the mechanisms underlying emergent properties of highly non-linear neural systems. How does a network of cortical neurons maintain a memory (Postle (2006))? How does the interaction of multiple brain regions support complex functions like abstract reasoning and conscious experience (Miller et al. (2024); Thiebaut de Schotten and Forkel (2022))? What makes the study of emergent properties

in the brain so challenging is the vast span of spatial scales which reflect neural function (Figure 1.1).

My study of neural oscillations and working memory is motivated by the same overarching question: what are the precise biological underpinnings of emergent properties in neural systems? This question is of central importance, as understanding these mechanistic links is critical for the translation of neuroscience findings to clinical and technological applications. One approach to testing hypotheses on the biological underpinnings of emergent properties is through the simulation of neural models. Computational neural modeling is well suited for the task, as the field covers models at dramatically different scales and levels of abstraction (Panahi et al. (2021); Abbott and Dayan (2001)). Ideally, these models produce predictions that can be directly tested through in vivo experiments (Einevoll et al. (2019)). Models which can be directly related to experimental observations such as electrophysiology and calcium imaging (i.e. biophysical neural models) are typically the most useful for producing testable experimental predictions. However, connecting these models to emergent properties of large scale neural systems (e.g. neural oscillations measured by EEG) remains an exceedingly challenging domain in computational neuroscience.

The following sections summarize well-established techniques computational neuroscientists have historically employed to study emergent dynamics in biophysical neural models. Specifically, I will 1) introduce an example of how mathematical modeling has aided in bridging microscale properties to macroscale behavior in the context of statistical mechanics, 2) describe the challenges of applying similar mathematical approaches to neuroscience questions, and 3) introduce modern deep learning techniques as a solution to the challenges of characterizing emergent properties in biophysically detailed models of cells and circuits.

1.2 Taming the complexity of neural systems with statistical mechanics

Statistical mechanics in physics is one of few fields which has developed mathematical models on the mechanistic origins of emergent properties (Jensen (2022)). Given this success, it is natural to try and adopt the validated methods from physics, and apply them to new fields like neuroscience. Below I briefly describe the historical development of statistical mechanics, its fruitful application to neural systems, and outline the major limitations of statistical mechanics for neuroscience.

Leading up to the 19th century, the dominant mathematical framework referred to as "classical mechanics", largely relied on deterministic models that described the behavior of macroscale objects. These deterministic models were often represented as a dynamical system. The fundamental challenge of applying classical mechanics to the study of emergent behavior is the requirement that every individual component's trajectory must be directly represented and tracked. Even for conservatively sized systems (e.g. a sealed gas container) a classical mechanistic model is often computationally infeasible. For example, a node on a modern computing cluster has a RAM size of 1028 gigabytes (that is 8^{12} bits). A 1 cm^3 container of gas contains $\sim 10^{19}$ molecules (Panahi et al. (2021)), meaning that modern supercomputers are orders of magnitude off from keeping track of binary information about each molecule, much less simulating the movement and interaction of these molecules.

A major conceptual leap that enabled physicists to connect microscale properties to macroscale emergent behavior was the development of statistical mechanics. Statistical mechanics refers to a mathematical framework which leverages probability theory to describe the behavior of a large assembly of microscopic components (Schwabl (2006)). Instead of directly considering every individual gas molecule, James Clerk Maxwell and Ludwig Boltzmann proposed to treat their behavior probabilistically (M.A (1860); Clausius

(1857); Sharp and Matschinsky (2015)). In place of tracking the individual movement and collisions of molecules in a container, they posited that one could treat the velocity of an individual particle as coming from a distribution. This statistical treatment has been developed into the field of thermodynamics, and explains how properties such as heat and pressure emerge from the interaction of a large assembly of gas molecules.

In the context of neuroscience modeling, there is an extensive literature on applying techniques from statistical mechanics to model the behavior of neural systems. One of the most widely accepted frameworks, dynamic causal modeling (DCM), is rooted in the mathematical formalism of mean-field theory, a sophisticated expansion of the rationale outlined by Maxwell and Boltzmann (Friston et al. (2003)). The fundamental unit of DCM is a neural mass model, a mathematically simplified representation of a biological neural network. Similar to how individual gas molecules can be described by their position, mass, and velocity, neural mass models can describe a "node" (small region of neural tissue) through the average firing rates of neurons, and the synaptic coupling strengths of excitatory and inhibitory neurons.

It is important to highlight that the explanatory strength of statistical mechanics in physics relies on the existence of first principles which govern the systems being studied. While many problems in physics can be described from a first principles approach, the same cannot be said for neuroscience (Markram (2013); Rabinovich et al. (2006)). The computational principles that allow the brain to operate remain unknown, and this lack of principles requires strong and often subjective assumptions when conducting a statistical mechanics treatment of the brain. In the case of neural mass models, while the firing rates and synaptic coupling are undoubtedly important features of a neuron's behavior, it is unknown if this information is sufficient to accurately predict the macroscale behavior of a large population of interacting neurons, or be causally related to brain function.

An alternative approach is to avoid the strong assumptions imposed by neural mass models and attempt to faithfully preserve the microscale properties. This approach, known

as biophysical neural modeling, attempts to study emergent neural dynamics from the bottom up and directly simulate their underlying neural mechanisms. It is important to note however, that even here, there are varying levels of biophysical detail included in such models depending on the scientific question being asked.

1.3 Embracing the complexity of neural systems with biophysical neural modeling and deep learning

The fundamental value of using detailed biophysical models is the ability to make inferences on the mechanistic origin of experimental observations at a given scale, assuming the detailed biophysics that reproduce the observation are included in the model. For example, to directly connect invasive experimental recordings to theories of neural function, biophysical neural models can be used for the simulation of realistic neurons with complex electrical properties and dendritic structures (Dai et al. (2020); Borges et al. (2022)). The backbone of biophysical modeling for linking electric and ionic properties of neurons to emergent properties in networks is a combination of 1) a mathematical representation of a neuronal membrane and its associated ion channels, 2) the cable equation from physics which allows the simulation of cylindrical compartments and are connected to form neurons of different shapes, and 3) a model of synaptic activation that captures the time course of excitatory and inhibitory postsynaptic potentials (EPSP and IPSP). Together, these building blocks allow the flexible integration of experimental data measuring cell biophysics, morphology, and synaptic connectivity in the form of model parameters (Hines and Carnevale (1997); Abbott and Dayan (2001); Rall (1959)). In connected networks, the model parameters can be further constrained by network-level measurements of neural activity like spiking, local field potentials, and MEG/EEG (Lindén et al. (2014); Neymotin et al. (2020)).

While the inclusion of biophysical detail offers mechanistic grounding and a path

towards validation of model predictions (Einevoll et al. (2019)), this added complexity poses two major challenges that have blocked the use of biophysical models for a wide range of neuroscience questions: 1) the high computational cost of simulating large-scale biophysically detailed network models (Cremonesi and Schürmann (2020); Zhang et al. (2023)), and 2) the complex and often non-unique mapping between model parameters and simulated outputs (Cranmer et al. (2020); Gonçalves et al. (2020)). Ultimately model predictions are the specific parameter values that produce desired simulation outputs (e.g. recorded neural activity). There are a diverse array of optimization techniques that can be used to fit these parameters to data, however, the unique challenges of biophysical models often leads to the failure of standard optimization algorithms. In contrast, a major strength of neural mass models is that for certain optimization techniques, their mathematical structure permits analytic solutions that enable extremely efficient parameter estimation. For detailed biophysical models, there is typically no analytic solution that permits efficient optimization. Given these difficulties, detailed biophysical models have historically been restricted in the scale and complexity of neuroscience phenomena investigated.

For individual researchers, the primary use of detailed biophysical models has been characterizing the behavior of individual neurons by including rich morphologic structure and complex ion channel dynamics. Studies on interconnected networks of detailed biophysical models are historically rare due the computational and experimental costs, however, the field has significantly grown and will continue to evolve in the near future. As an example, large institutions like the Blue Brain Project and the Allen Institute have attempted overcoming these challenges by running large-scale simulations on high performance computing clusters, in addition to conducting monumental experimental studies to collect enough data to constrain their models (Markram (2006); Hawrylycz et al. (2014); de Vries et al. (2023); Billeh et al. (2020)). While these efforts provide a solid foundation for future studies utilizing biophysical network models, significant work is still needed to make these tools accessible given the computational overhead. How do we

find a middle ground that will propel us forward with the knowledge, tools, and machines we have today?

Fortunately, in the last decade, the proliferation of deep learning techniques has produced promising solutions to these challenges, opening the door for biophysical models to be used for a new wealth of neuroscience questions. For the challenge of high computational cost, surrogate modeling, a technique in which a deep neural network is trained to directly approximate simulations of biophysical neural models, has been shown to reduce simulation times by multiple orders of magnitude (Gherman et al. (2023); Oláh et al. (2022); Verhellen et al. (2024)). A great deal of research is still necessary to validate the accuracy and scalability of surrogate models, nevertheless, they represent an exciting and novel path forward. For the challenge of complex parameter-to-output mappings, recently developed deep learning techniques such as simulation-based inference (SBI) (Cranmer et al. (2020); Gonçalves et al. (2019)) and emergent property inference (EPI) (Bittner et al. (2021)) enable the identification of parameters and parameter distributions that reproduce desired model behaviors. These approaches represent a fundamental shift in how deep learning is used in neuroscience. In contrast to the traditional use-case of extracting trends from large datasets, this new generation of model-augmented deep learning produces neuroscience predictions directly at the level of biological mechanisms.

1.4 Outline of Chapters

Chapter 2 Developing widely accessible and sustainable neuro-modeling tools describes my contributions to the open-source software neural modeling platform HNN-core, and reviews the current revolution in computational neuroscience towards open-source scientific software.

Chapter 3 Advancing biophysical modeling with deep learning tools consists of two

methodological studies that demonstrate how deep learning addresses the critical limitations of biophysical modeling (computational cost and parameter inference). The Human Neocortical Neurosolver (HNN) is used as an example biophysical model to demonstrate how these deep learning techniques assist with linking cell-level mechanistic properties to MEG/EEG measured neural oscillations.

Chapter 4 A novel framework for expanding RNNs with biophysical detail to solve cognitive tasks details a scientific study which introduces biophysical reservoir computing, a framework that guides the expansion of traditional task-trained RNNs, to include biology-inspired details like dendrites and biophysics. The framework is applied to study how choices in extrinsic cue input properties like synapse location (i.e. dendrite vs soma) and receptor time constants (i.e. NMDA vs AMPA) impact a network's ability to solve a simplified working memory task. The study reveals that NMDA receptors are uniquely well-suited for transmitting cue information through dendrites, in addition to an increase in training efficiency across synapse locations.

Chapter 5 Conclusions provides commentary on the future of biophysical modeling with respect to its scientific benefits, and technological hurdles the field must overcome to see these benefits. In the final section Digital twins as the future of neural modeling, I describe the promising translational applications of biophysical modeling and deep learning, with particular attention to clinical decision making for neuropsychiatric disease.

Box 1: Important definitions

Emergent property: a property of a physical system that arises from the interaction of many simpler components, and cannot be inferred from studying any individual component

Biophysics: the study of biological phenomena using mathematically rigorous techniques traditionally applied in physics

Dynamical systems: a discipline of mathematics involving differential equations that describes how systems evolve over time

Mechanistic inference: drawing conclusions about the underlying properties of a system that can explain observed data

Deep learning: a subset of machine learning that involves training multi-layer artificial neural networks to solve computational tasks

References

Abbott, L. F. and Dayan, P. (2001). *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems*. Computational Neuroscience Series. MIT Press, Cambridge, MA, USA.

Anderson, P. W. (1972). More Is Different. *Science*, 177(4047):393–396. Publisher: American Association for the Advancement of Science.

Arnal, L. H. and Giraud, A.-L. (2012). Cortical oscillations and sensory predictions. *Trends in Cognitive Sciences*, 16(7):390–398.

Billeh, Y. N., Cai, B., Gratiy, S. L., Dai, K., Iyer, R., Gouwens, N. W., Abbasi-Asl, R., Jia, X., Siegle, J. H., Olsen, S. R., Koch, C., Mihalas, S., and Arkhipov, A. (2020). Systematic Integration of Structural and Functional Data into Multi-scale Models of Mouse Primary Visual Cortex. *Neuron*.

Bittner, S. R., Palmigiano, A., Piet, A. T., Duan, C. A., Brody, C. D., Miller, K. D., and Cunningham, J. (2021). Interrogating theoretical models of neural computation with emergent property inference. *eLife*, 10:e56265. Publisher: eLife Sciences Publications, Ltd.

Borges, F. S., Moreira, J. V. S., Takarabe, L. M., Lytton, W. W., and Dura-Bernal, S. (2022). Large-scale biophysically detailed model of somatosensory thalamocortical circuits in NetPyNE. *Frontiers in Neuroinformatics*, 16.

Brigo, F. (2025). Hans Berger (1873–1941) and The Discovery of Clinical Electroencephalography. In *EEG: The First 100 Years*, pages 25–35. Springer.

Buzsáki, G. and Draguhn, A. (2004). Neuronal Oscillations in Cortical Networks. *Science*, 304(5679):1926–1929.

Clausius, R. (1857). XI. On the nature of the motion which we call heat. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*. Publisher:

Taylor & Francis Group.

Cranmer, K., Brehmer, J., and Louppe, G. (2020). The frontier of simulation-based inference. *Proceedings of the National Academy of Sciences*, 117(48):30055–30062. Publisher: National Academy of Sciences Section: Colloquium Paper.

Cremonesi, F. and Schürmann, F. (2020). Understanding Computational Costs of Cellular-Level Brain Tissue Simulations Through Analytical Performance Models. *Neuroinformatics*, 18(3):407–428.

Dai, K., Gratiy, S. L., Billeh, Y. N., Xu, R., Cai, B., Cain, N., Rimehaug, A. E., Stasik, A. J., Einevoll, G. T., Mihalas, S., Koch, C., and Arkhipov, A. (2020). Brain Modeling ToolKit: An open source software suite for multiscale modeling of brain circuits. *PLOS Computational Biology*, 16(11):e1008386. Publisher: Public Library of Science.

de Aguiar Neto, F. S. and Rosa, J. L. G. (2019). Depression biomarkers using non-invasive EEG: A review. *Neuroscience & Biobehavioral Reviews*, 105:83–93.

de Vries, S. E., Siegle, J. H., and Koch, C. (2023). Sharing neurophysiology data from the Allen Brain Observatory. *eLife*, 12:e85550. Publisher: eLife Sciences Publications, Ltd.

Einevoll, G. T., Destexhe, A., Diesmann, M., Grün, S., Jirsa, V., Kamps, M. d., Migliore, M., Ness, T. V., Plesser, H. E., and Schürmann, F. (2019). The Scientific Case for Brain Simulations. *Neuron*, 102(4):735–744. Publisher: Elsevier.

Foster, J. J., Vogel, E. K., and Awh, E. (2024). 13 Working memory as persistent neural activity. In *The Oxford Handbook of Human Memory, Two Volume Pack: Foundations and Applications, online edn*, pages 353–370. Academic, Oxford.

Friston, K. J., Harrison, L., and Penny, W. (2003). Dynamic causal modelling. *NeuroImage*, 19(4):1273–1302.

Geerts, H., Roberts, P., and Spiros, A. (2013). A Quantitative System Pharmacology Computer Model for Cognitive Deficits in Schizophre-

nia. *CPT: Pharmacometrics & Systems Pharmacology*, 2(4):36. _eprint: https://ascpt.onlinelibrary.wiley.com/doi/pdf/10.1038/psp.2013.12.

Geerts, H., Roberts, P., and Spiros, A. (2021). Exploring the relation between BOLD fMRI and cognitive performance using a computer-based quantitative systems pharmacology model: Applications to the COMTVAL158MET genotype and ketamine. *European Neuropsychopharmacology*, 50:12–22.

Gherman, I. M., Abdallah, Z. S., Pang, W., Gorochowski, T. E., Grierson, C. S., and Marucci, L. (2023). Bridging the gap between mechanistic biological models and machine learning surrogates. *PLOS Computational Biology*, 19(4):e1010988. Publisher: Public Library of Science.

Gonçalves, P. J., Lueckmann, J.-M., Deistler, M., Nonnenmacher, M., Öcal, K., Bassetto, G., Chintaluri, C., Podlaski, W. F., Haddad, S. A., Vogels, T. P., Greenberg, D. S., and Macke, J. H. (2019). Training deep neural density estimators to identify mechanistic models of neural dynamics. *bioRxiv*, page 838383.

Gonçalves, P. J., Lueckmann, J.-M., Deistler, M., Nonnenmacher, M., Öcal, K., Bassetto, G., Chintaluri, C., Podlaski, W. F., Haddad, S. A., Vogels, T. P., Greenberg, D. S., and Macke, J. H. (2020). Training deep neural density estimators to identify mechanistic models of neural dynamics. *eLife*, 9:e56261. Publisher: eLife Sciences Publications, Ltd.

Hawrylycz, M., Ng, L., Feng, D., Sunkin, S., Szafer, A., and Dang, C. (2014). The Allen Brain Atlas. In Kasabov, N., editor, *Springer Handbook of Bio-/Neuroinformatics*, pages 1111–1126. Springer, Berlin, Heidelberg.

Hines, M. L. and Carnevale, N. T. (1997). The NEURON Simulation Environment. *Neural Computation*, 9(6):1179–1209. Conference Name: Neural Computation.

Horvath, A., Szucs, A., Csukly, G., Sakovics, A., Stefanics, G., and Kamondi, A. (2018). EEG and ERP biomarkers of Alzheimer's disease: a critical review. *Frontiers in bioscience (Landmark edition)*, 23:183–220.

Hsieh, L.-T., Ekstrom, A. D., and Ranganath, C. (2011). Neural Oscillations Associated with Item and Temporal Order Maintenance in Working Memory. *Journal of Neuroscience*, 31(30):10803–10810. Publisher: Society for Neuroscience Section: Articles.

Jas, M., Thorpe, R., Tolley, N., Bailey, C., Brandt, S., Caldwell, B., Cheng, H., Daniels, D., Pujol, C. F., Khalil, M., Kanekar, S., Kohl, C., Kolozsvári, O., Lankinen, K., Loi, K., Neymotin, S., Partani, R., Pelah, M., Rockhill, A., Sherif, M., Hamalainen, M., and Jones, S. (2023). HNN-core: A Python software for cellular and circuit-level interpretation of human MEG/EEG. *Journal of Open Source Software*, 8(92):5848.

Jensen, H. J. (2022). Complex systems and emergent phenomena. In *Statistical and Nonlinear Physics*, pages 3–22. Springer.

Klimesch, W., Sauseng, P., and Hanslmayr, S. (2007). EEG alpha oscillations: The inhibition–timing hypothesis. *Brain Research Reviews*, 53(1):63–88.

Lindén, H., Hagen, E., Leski, S., Norheim, E. S., Pettersen, K. H., and Einevoll, G. T. (2014). LFPy: a tool for biophysical simulation of extracellular potentials generated by detailed model neurons. *Frontiers in Neuroinformatics*, 7.

Loo, S. K., Lenartowicz, A., and Makeig, S. (2016). Research Review: Use of EEG biomarkers in child psychiatry research – current state and future directions. *Journal of Child Psychology and Psychiatry*, 57(1):4–17. _eprint: https://acamh.onlinelibrary.wiley.com/doi/pdf/10.1111/jcpp.12435.

M.A, J. C. M. (1860). II. Illustrations of the dynamical theory of gases. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*. Publisher: Taylor & Francis Group.

Markram, H. (2006). The Blue Brain Project. *Nature Reviews Neuroscience*, 7(2):153–160. Publisher: Nature Publishing Group.

Markram, H. (2013). Seven challenges for neuroscience. *Functional neurology*, 28(3):145.

Miller, E. K., Brincat, S. L., and Roy, J. E. (2024). Cognition is an emergent property. *Current Opinion in Behavioral Sciences*, 57:101388.

Morrison, M. (2012). Emergent Physics and Micro-Ontology. *Philosophy of Science*, 79(1):141–166.

Murray, J. D., Anticevic, A., Gancsos, M., Ichinose, M., Corlett, P. R., Krystal, J. H., and Wang, X.-J. (2014). Linking Microcircuit Dysfunction to Cognitive Impairment: Effects of Disinhibition Associated with Schizophrenia in a Cortical Working Memory Model. *Cerebral Cortex*, 24(4):859–872.

Neymotin, S. A., Daniels, D. S., Caldwell, B., McDougal, R. A., Carnevale, N. T., Jas, M., Moore, C. I., Hines, M. L., Hämäläinen, M., and Jones, S. R. (2020). Human Neocortical Neurosolver (HNN), a new software tool for interpreting the cellular and network origin of human MEG/EEG data. *eLife*, 9:e51214. Publisher: eLife Sciences Publications, Ltd.

Oláh, V. J., Pedersen, N. P., and Rowan, M. J. (2022). Ultrafast simulation of large-scale neocortical microcircuitry with biophysically realistic neurons. *eLife*, 11:e79535. Publisher: eLife Sciences Publications, Ltd.

O'Connor, T. (2020). Emergent Properties. Last Modified: 2020-08-10.

Panahi, M. R., Abrevaya, G., Gagnon-Audet, J.-C., Voleti, V., Rish, I., and Dumas, G. (2021). Generative Models of Brain Dynamics – A review. Technical Report arXiv:2112.12147, arXiv. arXiv:2112.12147 [q-bio] type: article.

Pina, J. E., Bodner, M., and Ermentrout, B. (2018). Oscillations in working memory and neural binding: A mechanism for multiple memories and their interactions. *PLOS Computational Biology*, 14(11):e1006517. Publisher: Public Library of Science.

Postle, B. R. (2006). Working memory as an emergent property of the mind and brain.

Neuroscience, 139(1):23–38.

Pritchett, D. L., Siegle, J. H., Deister, C. A., and Moore, C. I. (2015). For things needing your attention: the role of neocortical gamma in sensory perception. *Current Opinion in Neurobiology*, 31:254–263.

Rabinovich, M. I., Varona, P., Selverston, A. I., and Abarbanel, H. D. I. (2006). Dynamical principles in neuroscience. *Reviews of Modern Physics*, 78(4):1213–1265. Publisher: American Physical Society.

Rall, W. (1959). Branching dendritic trees and motoneuron membrane resistivity. *Experimental neurology*, 1(5):491–527. Publisher: Elsevier.

Riddle, J., Scimeca, J. M., Cellier, D., Dhanani, S., and D'Esposito, M. (2020). Causal Evidence for a Role of Theta and Alpha Oscillations in the Control of Working Memory. *Current Biology*, 30(9):1748–1754.e4. Publisher: Elsevier.

Roux, F. and Uhlhaas, P. J. (2014). Working memory and neural oscillations: alpha–gamma versus theta–gamma codes for distinct WM information? *Trends in Cognitive Sciences*, 18(1):16–25. Publisher: Elsevier.

Sakkalis, V. (2011). Applied strategies towards EEG/MEG biomarker identification in clinical and cognitive research. *Biomarkers in Medicine*, 5(1):93–105. Publisher: Future Medicine.

Salisbury, D. F., Collins, K., and McCarley, R. W. (2010). Reductions in the N1 and P2 Auditory Event-Related Potentials in First-Hospitalized and Chronic Schizophrenia. *Schizophrenia Bulletin*, 36(5):991–1000.

Salt, G. W. (1979). A Comment on the Use of the Term Emergent Properties. *The American Naturalist*, 113(1):145–148. Publisher: The University of Chicago Press.

Schwabl, F. (2006). *Statistical Mechanics*. Springer Science & Business Media. Google-Books-ID: 7VnKAW284PgC.

Sharp, K. and Matschinsky, F. (2015). Translation of Ludwig Boltzmann's Paper "On the Relationship between the Second Fundamental Theorem of the Mechanical Theory of Heat and Probability Calculations Regarding the Conditions for Thermal Equilibrium" Sitzungberichte der Kaiserlichen Akademie der Wissenschaften. Mathematisch-Naturwissen Classe. Abt. II, LXXVI 1877, pp 373-435 (Wien. Ber. 1877, 76:373-435). Reprinted in Wiss. Abhandlungen, Vol. II, reprint 42, p. 164-223, Barth, Leipzig, 1909. *Entropy*, 17(4):1971–2009. Number: 4 Publisher: Multidisciplinary Digital Publishing Institute.

Spitzer, B. and Haegens, S. (2017). Beyond the Status Quo: A Role for Beta Oscillations in Endogenous Content (Re)Activation. *eNeuro*, 4(4).

Thiebaut de Schotten, M. and Forkel, S. J. (2022). The emergent properties of the connected brain. *Science*, 378(6619):505–510. Publisher: American Association for the Advancement of Science.

van Ede, F., Quinn, A. J., Woolrich, M. W., and Nobre, A. C. (2018). Neural Oscillations: Sustained Rhythms or Transient Burst-Events? *Trends in Neurosciences*, 41(7):415–417.

Verhellen, J., Beshkov, K., Amundsen, S., Ness, T. V., and Einevoll, G. T. (2024). Multitask learning of a biophysically-detailed neuron model. *PLOS Computational Biology*, 20(7):e1011728. Publisher: Public Library of Science.

Voloh, B. and Womelsdorf, T. (2018). Cell-Type Specific Burst Firing Interacts with Theta and Beta Activity in Prefrontal Cortex During Attention States. *Cerebral Cortex*, 28(12):4348–4364.

Wang, X.-J. (2021). 50 years of mnemonic persistent activity: quo vadis? *Trends in Neurosciences*, 44(11):888–902. Publisher: Elsevier.

Zhang, Y., He, G., Ma, L., Liu, X., Hjorth, J. J. J., Kozlov, A., He, Y., Zhang, S., Kotaleski, J. H., Tian, Y., Grillner, S., Du, K., and Huang, T. (2023). A GPU-based computational framework that bridges neuron simulation and artificial intelligence.

Nature Communications, 14(1):5798. Publisher: Nature Publishing Group.

CHAPTER 2

Developing widely accessible and sustainable neuro-modeling tools

An essential component to successfully modeling the cell-level biophysical mechanisms underlying network-level emergent dynamics is the modeling software. The importance of robust and well-documented scientific software cannot be understated, as its design directly influences critical choices made during scientific investigations. In the case of biophysical modeling, a well-designed software allows for scientists to test hypotheses across scales of neuronal structure and recording modalities.

This chapter begins with a description of the growing importance and impact of open-source scientific software on the neuroscience community. The following sections summarize my contributions to the open-source neural modeling platform the Human Neocortical Neurosolver (HNN) (Neymotin et al. (2020)), and concludes with a co-first-authored publication detailing the Python package HNN-core (Jas et al. (2023)).

2.1 The foundational role of open-source software in neuroscience

Computational neuroscience has flourished in recent decades in large part because of the ever-expanding open-source scientific community (Freeman (2015); White et al. (2019); Muller et al. (2015)). The popularity of open-source software is reflective of a broader shift towards scientists prioritizing transparency, reproducibility, and rigor. As it stands, there exists popular and well maintained open-source tools for a majority of the experimental and computational needs specific to neuroscience, including behavior analysis (Nath et al. (2019)), electrophysiology (Siegle et al. (2017)), calcium imaging (Giovannucci et al. (2019)), neuro-imaging (Gorgolewski et al. (2011)), microscopy (Longair et al. (2011)), as well as standardized tools for managing large scale neurophysiology datasets (Teeters et al. (2015)). Further, neuroscientists are able to take advantage of the broader open-source scientific ecosystem for custom data analysis workflows (Oliphant (2007)).

The field of computational modeling in neuroscience has greatly benefited from the wide-spread adoption of open-source tools. Historically, mathematical modeling in neuroscience was restricted to individuals primarily trained in physics given the strong reliance on dynamical systems theory and statistical mechanics, along with the practical implementation of these algorithms in low-level programming languages. While this knowledge remains valuable for scientifically rigorous investigations, it is no longer a strict barrier for entry. Now a moderate familiarity with programming languages such as Python is sufficient to work with numerous modeling frameworks (Hines et al. (2009); Eppler et al. (2009); Goodman and Brette (2008); Lindén et al. (2014); Dura-Bernal et al. (2019); Jas et al. (2023); Dai et al. (2020)), where the level of abstraction code-wise deals with biological concepts rather than the inner-workings of differential equation solvers.

A parallel development has taken place in the field of deep learning, with several Python package options offering a high-level interface for the construction and optimization of

artificial neural networks (Paszke et al. (2019); Developers (2021); Ketkar (2017); Bradbury et al. (2021); Kidger and Garcia (2021)). Importantly, the technological advancements in deep learning have directly led to improvements in the simulation of neural models with respect to hardware (i.e. GPU computing Stimberg et al. (2020); Kumbhar et al. (2019)) and software (Deistler et al. (2024); Mozafari et al. (2019)).

In Chapter 3 and Chapter 4, I further detail the integration of neural simulations and deep learning for the purpose of asking scientific questions. Below I describe the open-source neural modeling software HNN, my contributions to a Python interface to the modeling software known as HNN-core, as well my contributions to pedagogical material used to educate researchers on the origins of MEG/EEG signals with simulations and interactive code examples.

2.2 Enabling novel scientific investigations in HNN through open-source software

The Human Neocortical Neurosolver (HNN) is a biophysical modeling platform that allows scientists to simulate the cell-level origin of MEG/EEG signals (Neymotin et al. (2020)). HNN-core (Jas et al. (2023)) was started as a project with the goal of transforming the original HNN software into a Python package that meets the current best standards of practice in the open-source community. While the best standards of practice continue to evolve, the key components adopted by HNN-core have been continuous integration, unit testing, and democratic development. In this way, HNN-core seamlessly fits within the Python scientific ecosystem, and is able to be easily built upon.

A major component of my PhD thesis has been developing models and techniques specific to my research questions, and then refining the code associated with these projects into software that allows others to easily reproduce and expand upon the work. My contributions to HNN-core that I see as being the most beneficial for scientific investigations

have been: 1) enabling user-defined network connectivity (Appendix A: Connectivity tutorial), 2) the ability to record membrane potentials and synaptic currents from any neuron compartment (Appendix B: Firing pattern tutorial), 3) creating an animation class that permits 3D visualizations of simulated neural activity in the form of movies (Appendix C: HNN animation tutorial), and 4) adapting the models and simulations from Law et al. (2022) to a pedagogic Jupyter notebook tutorial to allow researchers to directly build upon the publication's results (Appendix D: Simulate beta tutorial).

The following section is a publication adapted from Jas et al. (2023) that was originally published in the Journal of Open Source Software, and details the motivation and functionality of HNN-core. The work was co-first authored by myself and the lead developers of HNN-core.

2.3 HNN-core: A Python software for cellular and circuit-level interpretation of human MEG/EEG

Contributing Authors

Mainak Jas[1,*], Ryan Thorpe[2,3,*], **Nicholas Tolley**[2,3,*], Christopher Bailey[4], Steven Brandt[2,3], Blake Caldwell[5], Huzi Cheng[6], Dylan Daniels[3], Carolina Fernandez Pujol[7], Mostafa Khalil[8], Samika Kanekar[9], Carmen Kohl[3], Orsolya Kolozsvári[10,11], Kaisu Lankinen[1,12], Kenneth Loi[3,13], Sam Neymotin[14,15], Rajat Partani[16], Mattan Pelah[3,17], Alex Rockhill[18], Mohamed Sherif[19,20], Matti Hämäläinen[21], Stephanie Jones[2,3].

[1]Athinoula A. Martinos Center for Biomedical Imaging, Massachusetts General Hospital, Boston, MA, USA
[2]Department of Neuroscience, Brown University, Providence, RI, USA
[3]Robert J. and Nancy D. Carney Institute for Brain Science, Brown University, Providence, RI, USA
[4]Department of Clinical Medicine, Aarhus University, Aarhus, Denmark
[5]Bassett Medical Center, Cooperstown, NY, USA
[6]Department of Psychological and Brain Sciences, Indiana University Bloomington, Bloomington, IN, USA
[7]Department of Biomedical Engineering, University of Miami, Coral Gables, FL, USA
[8]Department of Psychiatry and Behavioral Health, Penn State Milton S. Hershey Medical Center, Penn State College of Medicine, Hershey, PA, USA
[9]Warren Alpert Medical, Brown University, Providence, RI
[10]Department of Psychology, University of Jyväskylä, Jyväskylä, Finland
[11]Jyväskylä Centre for Interdisciplinary Brain Research, University of Jyväskylä, Jyväskylä, Finland
[12]Department of Radiology, Harvard Medical School, Boston, MA, USA
[13]Department of Molecular and Cell Biology; Innovative Genomics Institute, University of California, Berkeley, Berkeley, CA, USA
[14]Center for Biomedical Imaging and Neuromodulation, Nathan S. Kline Institute for Psychiatric Research, Orangeburg, NY, USA
[15]Department of Psychiatry, New York University Grossman School of Medicine, New York, NY, USA
[16]Department of Computer Science and Engineering, National Institute of Technology Karnataka, Karnataka, India
[17]Florida State University, Tallahassee, FL, USA
[18]Department of Human Physiology, University of Oregon, Eugene, OR, USA
[19]Department of Psychiatry and Human Behavior, Brown University, Providence, RI, USA
[20]Rhode Island Hospital, Providence, RI, USA
[21]Department of Neuroscience and Biomedical Engineering, Aalto University, Espoo, Finland

*These authors contributed equally.

2.3.1 Summary

HNN-core is a library for circuit and cellular level interpretation of non-invasive human magneto-/electro-encephalography (MEG/EEG) data. It is based on the Human Neocortical Neurosolver (HNN) software (Neymotin et al. (2020)), a modeling tool designed to simulate multiscale neural mechanisms generating current dipoles in a localized patch of neocortex. HNN's foundation is a biophysically detailed neural network representing a canonical neocortical column containing populations of pyramidal and inhibitory neurons together with layer-specific exogenous synaptic drive (Figure 2.1 left). In addition to simulating network-level interactions, HNN produces the intracellular currents in the long apical dendrites of pyramidal cells across the cortical layers known to be responsible for macroscopic current dipole generation.

HNN-core reproduces the workflows and tutorials provided in the original HNN software to generate commonly observed MEG/EEG signals including evoked response potentials (ERPs) and alpha (8-10 Hz), beta (15-30 Hz), and gamma (30-80 Hz) rhythms. HNN-core enables simultaneous calculation and visualization of macro- to micro-scale dynamics including MEG/EEG current dipoles, local field potential, laminar current-source density, and cell spiking and intrinsic dynamics. Importantly, HNN-core adopts modern open source development standards including a simplified installation procedure, unit tests, automatic documentation builds, code coverage, continuous integration, and contributing guidelines, supporting community development and long-term sustainability.

2.3.2 Statement of need

HNN-core addresses a key need in the fields of computational and experimental neuroscience by providing an extensively documented application programming interface (API) that allows both novel and advanced users to run biophysically-principled neural network simulations out-of-the-box with a few lines of code. HNN-core modularizes the model components originally introduced by HNN and its associated graphical user

interface (GUI) and provides an interface to modify it directly from Python. This has allowed for significant expansion of the HNN functionality through scripting, including the ability to modify additional features of local network connectivity and cell properties, record voltages in extracellular arrays, and more advanced parameter optimization and batch processing. A new web-based GUI has been developed as a thin layer over the Python interface making the overall software more maintainable.

2.3.3 HNN-core implements a biophysically detailed model to interpret MEG/EEG primary current sources

MEG/EEG are the two electrophysiological methods to non-invasively study the human brain. They have been used in developing biomarkers for healthy and pathological brain processes. Yet, the underlying cellular and circuit level generators of MEG/EEG signals have been difficult to infer. This detailed understanding is critical to develop theories of information processing based on these signals, or to use these techniques to develop new therapeutics. Computational neural modeling is a powerful technique to hypothesize the neural origin of these signals and several modeling frameworks have been developed. Since MEG/EEG recordings are dominated by neocortical sources, all models developed so far simulate neocortical activity, but they are developed with different levels of biophysical detail and correspondingly different use cases. The level of detail in HNN's pre-tuned network models and the workflows developed to study ERPs and low frequency oscillations are unique within the field of MEG/EEG modeling. One class of models known as neural mass models (NMMs) uses simplified representations to simulate net population dynamics, where hypothesized connectivity among neural "nodes" can be inferred from recordings. The Virtual Brain Project (Sanz Leon et al. (2013)) and Dynamic Causal Modeling from the SPM software (Friston et al. (2003); Litvak et al. (2011)) are prominent examples of software that implement NMMs. While NMMs are computationally tractable and advantageous for studying brain-wide interactions, they do not provide detailed

interpretation of cell and circuit level phenomena underlying MEG/EEG. The primary electrical currents that create MEG/EEG sensor signals are known to be oriented along the long and spatially aligned cortical pyramidal neuron dendrites, and their direction corresponds to that of the intracellular current flow (Hämäläinen et al. (1993)). For a detailed discussion see Neymotin et al. (2020). Further, source localization methods such as minimum-norm estimate (MNE) calculate the primary currents (assuming constraints defined by the technique (Gramfort et al. (2013))). As such, models created to study the cell and circuit origin of these signals are designed with detailed pyramidal neuron morphology and physiology, and are often embedded in a full neocortical column model. HNN is one such detailed neocortical column model (Neymotin et al. (2020)), and other examples have been employed using the software LFPy (Lindén et al. (2014)). LFPy simulates fields and currents from detailed models of pyramidal neurons and networks (Lindén et al. (2014)). LFPy does not support a specific neocortical model or workflows/parameter inference to study particular signals of interest. Rather, it provides multi-use Python scripts that can be integrated into any NEURON based neural model containing multi-compartment pyramidal neurons. A unique feature of HNN is its workflows for interacting with the template neocortical model through layer-specific activations to study ERPs and low frequency brain rhythms. HNN also enables direct comparison between simulation output and the waveforms of estimated sources in the same units of measure and supports parameter inference. HNN-core was created to maintain all of the functionality of the original HNN software with additional utility (described below) and a well-defined, well-tested and documented API. Its adoption of open source development standards, including a simplified installation procedure, unit tests, automatic documentation builds, code coverage, and continuous integration, enables community development and long-term sustainability.

2.3.4 HNN-core facilitates reproducibility and computationally expensive workflows

Scripting in HNN-core greatly expands the software utility particularly for large research projects where reproducibility and batch processing is of key importance. The scripted interface allows multi-trial simulations enabling the use of computationally expensive parameter optimization algorithms and parameter sweeps using parallel processing on computer clusters. HNN scripting also facilitates the creation of publication-quality figures and advanced statistical analysis. Further, the software can be integrated with existing scripted workflows, such as those developed in MNE-Python (Gramfort et al. (2013)), a well-established MEG/EEG signal processing and source localization software, enabling source localization and circuit interpretation in just a few lines of code (see a tutorial of this in the HNN-core documentation).

2.3.5 Notable features of HNN-core

HNN-core functionality supports advanced simulations through scripting that are not currently possible in the GUI including:

- the ability to record extracellular local field potentials from user defined positions, as well as voltages and synaptic currents from any compartment in the model;
- the ability to modify all features of the morphology and biophysical properties of any cell in the network;
- an API that enables complete control of cell-cell and drive-cell connectivity in the network;
- an API that allows for flexibility in defining the exogenous layer-specific drive to the neocortical network;
- the ability to choose from multiple template models based on previous publications

(e.g., `jones_2009_model()` (Jones et al. (2009)), `law_2021_model()` (Law et al. (2022)), and `calcium_model()` adapted from (Kohl et al. (2022)));

- built-in ERP optimization functionality designed for faster convergence;
- the choice of two parallel backends for either parallelizing across cells to speed up individual simulations (MPI), or across trials to speed up batches of simulations (Joblib).

HNN-core code has also enabled the creation of a new and improved web-based GUI based on ipywidgets (ipywidgets developers (2015)) and voila (voila developers (2019)) that can be run remotely with port forwarding.

All of the code associated with HNN-core has been extensively documented at multiple levels, including an API describing basic functions/parameters and examples of use for hypothesis generation and/or testing. Specifically, we distribute tutorials (hnn developers (2023h)) that mimic the original GUI tutorial workflows using HNN-core functions, with commentary on the known biophysical mechanisms of these signals.

2.3.6 Use cases and quick example code of running a simulation

As summarized above, HNN-core reproduces the workflows and tutorials provided in the original GUI driven HNN software designed to investigate the origin of commonly observed MEG/EEG signals. The HNN-core tutorials (hnn developers (2023h)) include examples of how to simulate ERPs (hnn developers (2023b)), as well as low frequency rhythms such as alpha (hnn developers (2023d)) (8-10 Hz), beta (hnn developers (2023d)) (15-30 Hz), and gamma (hnn developers (2023e)) (30-80 Hz). The tutorials also include an example of directly comparing simulations to real data (i.e., the median nerve evoked response (hnn developers (2023f))). We also provide short and targeted "How to" examples that describe how to use specific functionality, such as plotting firing patterns (hnn developers (2023a)), or recording extracellular LFPs (hnn developers (2023c)).

In practice, users learn how to study the multi-scale origin of ERPs and low frequency oscillations by first following the tutorials in the HNN-GUI (hnn developers (2023g)), and then recapitulating these tutorials in HNN-core (hnn developers (2023h)). The tutorials provide an interactive investigation that gives intuition on how exogenous drives and other parameters in the model impact the outputs of the simulations. From there, users can test hypotheses about what parameters or sets of parameters need to be adjusted to account for their recorded data by directly comparing simulation output to data. Automated parameter inference can be performed to optimize parameters to produce a close fit (i.e., small root mean squared error) to current source ERPs, and more advanced parameter inference methods are in development.

HNN-core has minimal dependencies which allows for effortless installation using the pip Python installer. In addition to NumPy (Harris et al. (2020)), SciPy (Virtanen et al. (2020)), and Matplotlib (Hunter (2007)) common in most libraries in the scientific Python stack, HNN-core uses NEURON (Hines and Carnevale (1997)) for the cell and circuit modeling. Here, we demonstrate how the HNN-core interface can be used to quickly simulate and plot the net cortical dipole response to a brief exogenously evoked drive representing "feedforward" thalamocortical input (Figure 2.1 right). This input (referred to as 'evprox1') effectively targets the proximal dendrites of the pyramidal neurons in L2/3 and L5, using the template neocortical model as in Jones et al. (2009) (Figure 2.1 left). Note that this simulation is not addressing a specific scientific question, and is simply an educational example.

```
from hnn_core import jones_2009_model, simulate_dipole
# 1) Create the network model
net = jones_2009_model()
# 2) Define weights and delay times of inputs to network
weights_ampa = {'L2_basket': 0.09, 'L2_pyramidal': 0.02,
                'L5_basket': 0.2, 'L5_pyramidal': 8e-3}
synaptic_delays = {'L2_basket': 0.1, 'L2_pyramidal': 0.1,
                   'L5_basket': 1.0, 'L5_pyramidal': 1.0}
# 3) Attach inputs to to network
```

```
net.add_evoked_drive(name='evprox1', mu=20.0, sigma=2.0, numspikes=1,
                     weights_ampa=weights_ampa, location='proximal',
                     synaptic_delays=synaptic_delays)
# 4) Run simulation and plot results
dpl = simulate_dipole(net, tstop=100.0)
```

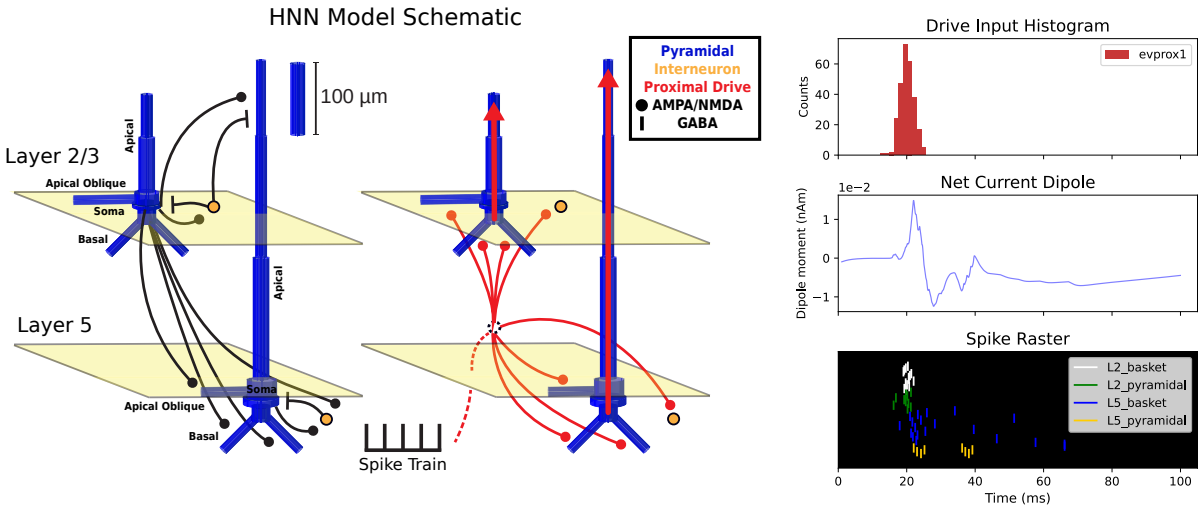

Figure 2.1: **Left**: Reduced schematic of HNN model detailing the cell types, layer-specific synaptic connectivity structure, and locations of proximal drive synapses. The default size of the full network is a grid of 100 pyramidal neurons, and 35 inhibitory neurons, synaptically connected in each layer. Figure adapted from Neymotin et al. (2020). **Right**: Plots of the network and simulated results can be generated using the HNN-core visualization API. The drive input histogram with `net.cell_response.plot_spikes_hist()`, the net current dipole with `plot_dipole(dpl)`, and the spike raster with `net.cell_response.plot_spikes_raster()`.

2.3.7 Ongoing research using HNN-core

The scripted interface of HNN-core has enabled the development of advanced parameter inference techniques (Tolley et al. (2024)) using Simulation-Based Inference (Tejero-Cantero et al. (2020)). It has been used in Thorpe et al. (2021) to propose new mechanisms of innocuous versus noxious sensory processing in the primary somatosensory neocortex. Lankinen et al. (2024) have used HNN-core to study crossmodal interactions between auditory and visual cortices. They performed group analysis on multiple subjects along with optimization and nonparametric statistical testing. Additionally, Szul et al. (2023) used it for understanding features of beta bursts in motor cortex and Fernandez Pujol

et al. (2023) to study auditory perception.

Overall, HNN-core provides an expandable and sustainable Python-based software package that can help advance understanding of the cellular and circuit mechanisms of MEG/EEG signal generation and ultimately lead to new neuroscience discoveries.

Acknowledgements

HNN-core was supported by NIH grants R01EB022889, R01NS104585, R01MH130415, R01AG076227, and Google Summer of Code.

References

Bradbury, J., Frostig, R., Hawkins, P., Johnson, M. J., Leary, C., Maclaurin, D., Necula, G., Paszke, A., VanderPlas, J., Wanderman-Milne, S., and Zhang, Q. (2021). JAX: Autograd and XLA. *Astrophysics Source Code Library*, page ascl:2111.002. ADS Bibcode: 2021ascl.soft11002B.

Dai, K., Gratiy, S. L., Billeh, Y. N., Xu, R., Cai, B., Cain, N., Rimehaug, A. E., Stasik, A. J., Einevoll, G. T., Mihalas, S., Koch, C., and Arkhipov, A. (2020). Brain Modeling ToolKit: An open source software suite for multiscale modeling of brain circuits. *PLOS Computational Biology*, 16(11):e1008386. Publisher: Public Library of Science.

Deistler, M., Kadhim, K. L., Pals, M., Beck, J., Huang, Z., Gloeckler, M., Lappalainen, J. K., Schröder, C., Berens, P., Gonçalves, P. J., and Macke, J. H. (2024). Differentiable simulation enables large-scale training of detailed biophysical models of neural dynamics. Pages: 2024.08.21.608979 Section: New Results.

Developers, T. (2021). TensorFlow. *Zenodo*. Publisher: Zenodo ADS Bibcode: 2021zndo...4758419D.

Dura-Bernal, S., Suter, B. A., Gleeson, P., Cantarelli, M., Quintana, A., Rodriguez, F., Kedziora, D. J., Chadderdon, G. L., Kerr, C. C., Neymotin, S. A., McDougal, R. A., Hines, M., Shepherd, G. M., and Lytton, W. W. (2019). NetPyNE, a tool for data-driven multiscale modeling of brain circuits. *eLife*, 8:e44494.

Eppler, J. M., Helias, M., Muller, E., Diesmann, M., and Gewaltig, M.-O. (2009). PyNEST: a convenient interface to the NEST simulator. *Frontiers in Neuroinformatics*, 2. Publisher: Frontiers.

Fernandez Pujol, C., Blundon, E. G., and Dykstra, A. R. (2023). Laminar specificity of the auditory perceptual awareness negativity: A biophysical modeling study. *PLOS Computational Biology*, 19(6):1–23. Publisher: Public Library of Science.

Freeman, J. (2015). Open source tools for large-scale neuroscience. *Current Opinion in Neurobiology*, 32:156–163.

Friston, K. J., Harrison, L., and Penny, W. (2003). Dynamic causal modelling. *NeuroImage*, 19(4):1273–1302.

Giovannucci, A., Friedrich, J., Gunn, P., Kalfon, J., Brown, B. L., Koay, S. A., Taxidis, J., Najafi, F., Gauthier, J. L., Zhou, P., Khakh, B. S., Tank, D. W., Chklovskii, D. B., and Pnevmatikakis, E. A. (2019). CaImAn an open source tool for scalable calcium imaging data analysis. *eLife*, 8:e38173. Publisher: eLife Sciences Publications, Ltd.

Goodman, D. F. M. and Brette, R. (2008). Brian: a simulator for spiking neural networks in Python. *Frontiers in Neuroinformatics*, 2. Publisher: Frontiers.

Gorgolewski, K., Burns, C. D., Madison, C., Clark, D., Halchenko, Y. O., Waskom, M. L., and Ghosh, S. S. (2011). Nipype: A Flexible, Lightweight and Extensible Neuroimaging Data Processing Framework in Python. *Frontiers in Neuroinformatics*, 5. Publisher: Frontiers.

Gramfort, A., Luessi, M., Larson, E., Engemann, D. A., Strohmeier, D., Brodbeck, C., Goj, R., Jas, M., Brooks, T., and Parkkonen, L. (2013). MEG and EEG data analysis with MNE-Python. *Frontiers in Neuroinformatics*, 7:267. Publisher: Frontiers Media SA.

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del Río, J. F., Wiebe, M., Peterson, P., Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C., and Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, 585(7825):357–362. Number: 7825 Publisher: Nature Publishing Group.

Hines, M., Davison, A. P., and Muller, E. (2009). NEURON and Python. *Frontiers in Neuroinformatics*, 3. Publisher: Frontiers.

Hines, M. L. and Carnevale, N. T. (1997). The NEURON Simulation Environment. *Neural Computation*, 9(6):1179–1209. Conference Name: Neural Computation.

hnn developers (2023a). 01. Plot firing pattern — hnn-core 0.4.1 documentation.

hnn developers (2023b). 01. Simulate Event Related Potentials (ERPs) — hnn-core 0.4.1 documentation.

hnn developers (2023c). 02. Record extracellular potentials — hnn-core 0.4.1 documentation.

hnn developers (2023d). 02. Simulate Alpha and Beta Rhythms — hnn-core 0.4.1 documentation.

hnn developers (2023e). 03. Simulate Gamma Rhythms — hnn-core 0.4.1 documentation.

hnn developers (2023f). 04. From MEG sensor-space data to HNN simulation — hnn-core 0.4.1 documentation.

hnn developers (2023g). HNN-GUI Tutorials | HNN.

hnn developers (2023h). Tutorials — hnn-core 0.4.1 documentation.

Hunter, J. D. (2007). Matplotlib: A 2D Graphics Environment. *Computing in Science & Engineering*, 9(3):90–95. Conference Name: Computing in Science & Engineering.

Hämäläinen, M., Hari, R., Ilmoniemi, R. J., Knuutila, J., and Lounasmaa, O. V. (1993). Magnetoencephalography—theory, instrumentation, and applications to noninvasive studies of the working human brain. *Reviews of Modern Physics*, 65(2):413–497. Publisher: American Physical Society.

ipywidgets developers (2015). ipywidgets: Interactive HTML Widgets. original-date: 2015-04-17T20:29:25Z.

Jas, M., Thorpe, R., Tolley, N., Bailey, C., Brandt, S., Caldwell, B., Cheng, H., Daniels, D., Pujol, C. F., Khalil, M., Kanekar, S., Kohl, C., Kolozsvári, O., Lankinen, K.,

Loi, K., Neymotin, S., Partani, R., Pelah, M., Rockhill, A., Sherif, M., Hamalainen, M., and Jones, S. (2023). HNN-core: A Python software for cellular and circuit-level interpretation of human MEG/EEG. *Journal of Open Source Software*, 8(92):5848.

Jones, S. R., Pritchett, D. L., Sikora, M. A., Stufflebeam, S. M., Hämäläinen, M., and Moore, C. I. (2009). Quantitative Analysis and Biophysically Realistic Neural Modeling of the MEG Mu Rhythm: Rhythmogenesis and Modulation of Sensory-Evoked Responses. *Journal of Neurophysiology*, 102(6):3554–3572. Publisher: American Physiological Society.

Ketkar, N. (2017). Introduction to Keras. In Ketkar, N., editor, *Deep Learning with Python: A Hands-on Introduction*, pages 97–111. Apress, Berkeley, CA.

Kidger, P. and Garcia, C. (2021). Equinox: neural networks in JAX via callable PyTrees and filtered transformations.

Kohl, C., Parviainen, T., and Jones, S. R. (2022). Neural Mechanisms Underlying Human Auditory Evoked Responses Revealed By Human Neocortical Neurosolver. *Brain Topography*, 35(1):19–35.

Kumbhar, P., Hines, M., Fouriaux, J., Ovcharenko, A., King, J., Delalondre, F., and Schürmann, F. (2019). CoreNEURON : An Optimized Compute Engine for the NEURON Simulator. *Frontiers in Neuroinformatics*, 13. Publisher: Frontiers.

Lankinen, K., Ahveninen, J., Jas, M., Raij, T., and Ahlfors, S. P. (2024). Neuronal Modeling of Cross-Sensory Visual Evoked Magnetoencephalography Responses in the Auditory Cortex. *Journal of Neuroscience*, 44(17). Publisher: Society for Neuroscience Section: Research Articles.

Law, R. G., Pugliese, S., Shin, H., Sliva, D. D., Lee, S., Neymotin, S., Moore, C., and Jones, S. R. (2022). Thalamocortical Mechanisms Regulating the Relationship between Transient Beta Events and Human Tactile Perception. *Cerebral Cortex*, 32(4):668–688.

Lindén, H., Hagen, E., Leski, S., Norheim, E. S., Pettersen, K. H., and Einevoll, G. T. (2014). LFPy: a tool for biophysical simulation of extracellular potentials generated by detailed model neurons. *Frontiers in Neuroinformatics*, 7.

Litvak, V., Mattout, J., Kiebel, S., Phillips, C., Henson, R., Kilner, J., Barnes, G., Oostenveld, R., Daunizeau, J., and Flandin, G. (2011). EEG and MEG data analysis in SPM8. *Computational intelligence and neuroscience*, 2011(1):852961. Publisher: Wiley Online Library.

Longair, M. H., Baker, D. A., and Armstrong, J. D. (2011). Simple Neurite Tracer: open source software for reconstruction, visualization and analysis of neuronal processes. *Bioinformatics*, 27(17):2453–2454.

Mozafari, M., Ganjtabesh, M., Nowzari-Dalini, A., and Masquelier, T. (2019). SpykeTorch: Efficient Simulation of Convolutional Spiking Neural Networks With at Most One Spike per Neuron. *Frontiers in Neuroscience*, 13. Publisher: Frontiers.

Muller, E., Bednar, J. A., Diesmann, M., Gewaltig, M.-O., Hines, M., and Davison, A. P. (2015). Python in neuroscience. *Frontiers in Neuroinformatics*, 9. Publisher: Frontiers.

Nath, T., Mathis, A., Chen, A. C., Patel, A., Bethge, M., and Mathis, M. W. (2019). Using DeepLabCut for 3D markerless pose estimation across species and behaviors. *Nature Protocols*, 14(7):2152–2176. Publisher: Nature Publishing Group.

Neymotin, S. A., Daniels, D. S., Caldwell, B., McDougal, R. A., Carnevale, N. T., Jas, M., Moore, C. I., Hines, M. L., Hämäläinen, M., and Jones, S. R. (2020). Human Neocortical Neurosolver (HNN), a new software tool for interpreting the cellular and network origin of human MEG/EEG data. *eLife*, 9:e51214. Publisher: eLife Sciences Publications, Ltd.

Oliphant, T. E. (2007). Python for Scientific Computing. *Computing in Science & Engineering*, 9(3):10–20.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. (2019). PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Sanz Leon, P., Knock, S. A., Woodman, M. M., Domide, L., Mersmann, J., McIntosh, A. R., and Jirsa, V. (2013). The Virtual Brain: a simulator of primate brain network dynamics. *Frontiers in Neuroinformatics*, 7. Publisher: Frontiers.

Siegle, J. H., López, A. C., Patel, Y. A., Abramov, K., Ohayon, S., and Voigts, J. (2017). Open Ephys: an open-source, plugin-based platform for multichannel electrophysiology. *Journal of Neural Engineering*, 14(4):045003. Publisher: IOP Publishing.

Stimberg, M., Goodman, D. F. M., and Nowotny, T. (2020). Brian2GeNN: accelerating spiking neural network simulations with graphics hardware. *Scientific Reports*, 10(1):410. Publisher: Nature Publishing Group.

Szul, M. J., Papadopoulos, S., Alavizadeh, S., Daligaut, S., Schwartz, D., Mattout, J., and Bonaiuto, J. J. (2023). Diverse beta burst waveform motifs characterize movement-related cortical dynamics. *Progress in Neurobiology*, 228:102490.

Teeters, J. L., Godfrey, K., Young, R., Dang, C., Friedsam, C., Wark, B., Asari, H., Peron, S., Li, N., Peyrache, A., Denisov, G., Siegle, J. H., Olsen, S. R., Martin, C., Chun, M., Tripathy, S., Blanche, T. J., Harris, K., Buzsáki, G., Koch, C., Meister, M., Svoboda, K., and Sommer, F. T. (2015). Neurodata Without Borders: Creating a Common Data Format for Neurophysiology. *Neuron*, 88(4):629–634. Publisher: Elsevier.

Tejero-Cantero, A., Boelts, J., Deistler, M., Lueckmann, J.-M., Durkan, C., Gonçalves, P. J., Greenberg, D. S., and Macke, J. H. (2020). sbi: A toolkit for simulation-based inference. *Journal of Open Source Software*, 5(52):2505.

Thorpe, R. V., Black, C. J., Borton, D. A., Hu, L., Saab, C. Y., and Jones, S. R. (2021).

Distinct neocortical mechanisms underlie human SI responses to median nerve and laser evoked peripheral activation. Pages: 2021.10.11.463545 Section: New Results.

Tolley, N., Rodrigues, P. L. C., Gramfort, A., and Jones, S. R. (2024). Methods and considerations for estimating parameters in biophysically detailed neural models with simulation based inference. *PLOS Computational Biology*, 20(2):e1011108. Publisher: Public Library of Science.

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., Carey, C. J., Polat, , Feng, Y., Moore, E. W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E. A., Harris, C. R., Archibald, A. M., Ribeiro, A. H., Pedregosa, F., and van Mulbregt, P. (2020). SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nature Methods*, 17(3):261–272. Number: 3 Publisher: Nature Publishing Group.

voila developers (2019). Voilà. original-date: 2018-08-21T14:21:42Z.

White, S. R., Amarante, L. M., Kravitz, A. V., and Laubach, M. (2019). The Future Is Open: Open-Source Tools for Behavioral Neuroscience Research. *eNeuro*, 6(4). Publisher: Society for Neuroscience Section: Commentary.

CHAPTER 3

Advancing biophysical modeling with deep learning tools

This chapter presents two major methodological advances which employ deep learning to address well-known shortcomings of biophysical modeling. The first section is adapted from my original research article Tolley et al. (2024) which details the use of simulation-based inference (SBI), a deep learning technique used to identify model parameter distributions that fit experimental data. In the paper I apply SBI to the Human Neocortical Neurosolver (HNN), a biophysically detailed model of a neocortical column designed to connect cell- and circuit-level properties to the generation of neural oscillations measured by magneto- and electroencephalography (MEG/EEG). The second section is adapted from an accepted abstract presented at the annual Computational and Systems Neuroscience conference (COSYNE, March 2024), and describes the use of surrogate modeling, a deep learning technique that approximates the simulation of biophysical neural models with dramatically lower computational costs.

3.1 Methods and considerations for estimating parameters in biophysically detailed neural models with simulation based inference

Contributing Authors

Nicholas Tolley[1], Pedro L. C. Rodrigues[2], Alexandre Gramfort[3,*], Stephanie R. Jones[1,*].

3.1.1 Abstract

Biophysically detailed neural models are a powerful technique to study neural dynamics in health and disease with a growing number of established and openly available models. A major challenge in the use of such models is that parameter inference is an inherently difficult and unsolved problem. Identifying unique parameter distributions that can account for observed neural dynamics, and differences across experimental conditions, is essential to their meaningful use. Recently, simulation based inference (SBI) has been proposed as an approach to perform Bayesian inference to estimate parameters in detailed neural models. SBI overcomes the challenge of not having access to a likelihood function, which has severely limited inference methods in such models, by leveraging advances in deep learning to perform density estimation. While the substantial methodological advancements offered by SBI are promising, their use in large scale biophysically detailed models is challenging and methods for doing so have not been established, particularly when inferring parameters that can account for time series waveforms. We provide

[1]Department of Neuroscience, Brown University, Providence, Rhode Island, United States of America
[2]Université Grenoble Alpes, Inria, CNRS, Grenoble INP, LJK, Grenoble, France
[3]Université Paris-Saclay, Inria, CEA, Palaiseau, France

*These authors contributed equally.

guidelines and considerations on how SBI can be applied to estimate time series waveforms in biophysically detailed neural models starting with a simplified example and extending to specific applications to common MEG/EEG waveforms using the the large scale neural modeling framework of the Human Neocortical Neurosolver. Specifically, we describe how to estimate and compare results from example oscillatory and event related potential simulations. We also describe how diagnostics can be used to assess the quality and uniqueness of the posterior estimates. The methods described provide a principled foundation to guide future applications of SBI in a wide variety of applications that use detailed models to study neural dynamics.

Author summary

A central problem in computational neural modeling is estimating model parameters that can account for observed activity patterns. While several techniques exist to perform parameter inference in special classes of abstract neural models, there are comparatively few approaches for large scale biophysically detailed neural models. In this work, we describe challenges and solutions in applying a deep learning based statistical framework to estimate parameters in a biophysically detailed large scale neural model, and emphasize the particular difficulties in estimating parameters for time series data. Our example uses a multi-scale model designed to connect human MEG/EEG recordings to the underlying cell and circuit level generators. Our approach allows for crucial insight into how cell-level properties interact to produce measured neural activity, and provides guidelines for diagnosing the quality of the estimate and uniqueness of predictions for different MEG/EEG biomarkers.

3.1.2 Introduction

Biophysically detailed neural modeling is a fundamental and established framework to study fast time scale neural dynamics (Abbott and Dayan (2001); Einevoll et al. (2019)).

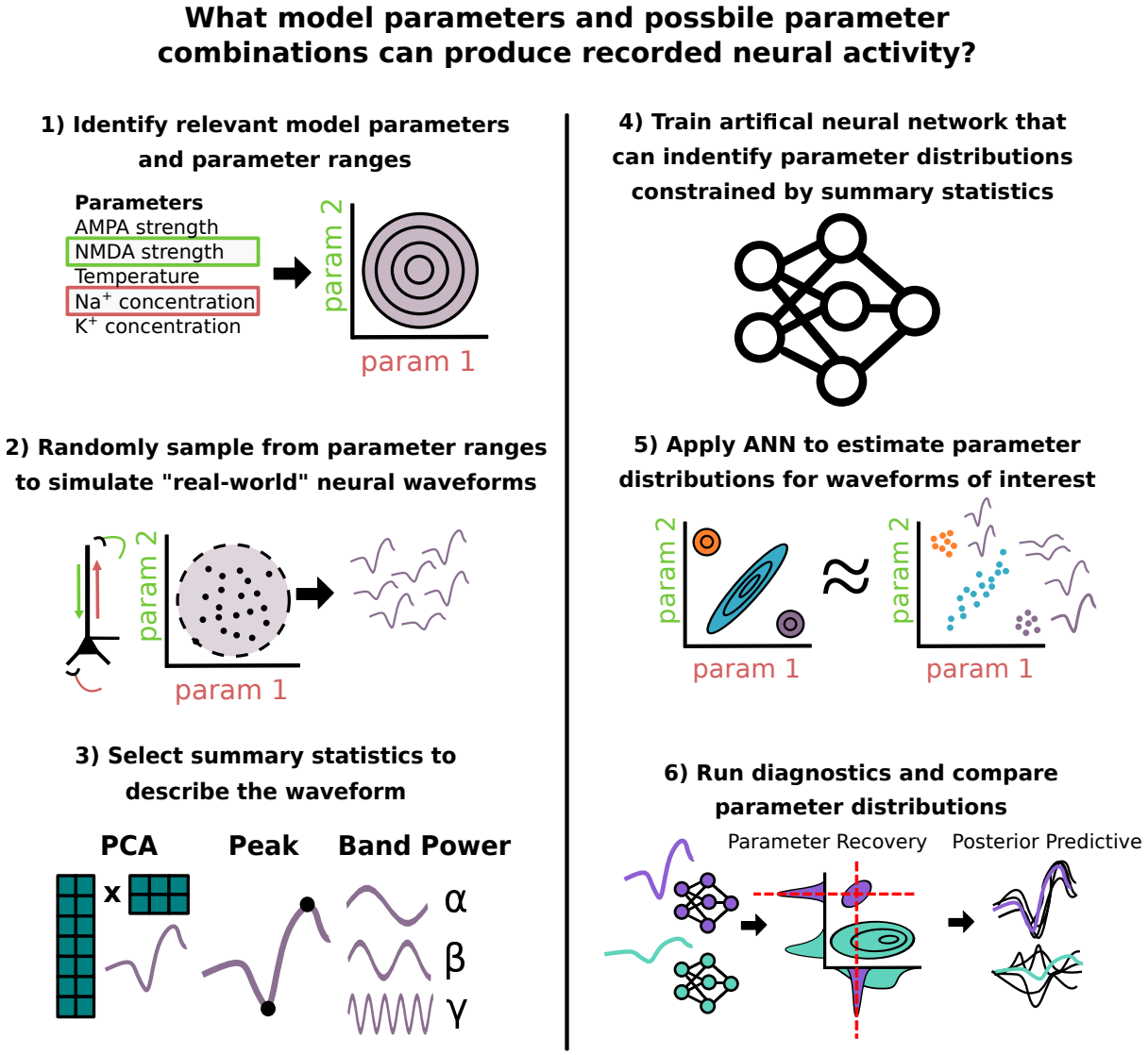

Figure 3.1: **Graphical Abstract.** Summary of SBI workflow used to infer model parameters that can account for recorded neural dynamics. 1) A prior distribution of assumed relevant model parameters and ranges is constructed. 2) A dataset of simulated neural activity patterns is generated with parameters sampled from the prior distribution. 3) User defined summary statistics are chosen to describe waveform features of interest. 4) A specialized deep learning architecture is trained to learn the mapping from neural activity constrained by summary statistics to underlying model parameters. 5) Specific neural activity patterns of interest are fed into the trained neural network, which subsequently outputs a distribution over the potential underlying model parameters. 6) Parameter estimates for different waveforms can be compared through diagnostics like parameter recovery (if the ground truth is known), or posterior predictive checks.

While challenging to construct, advances in computational resources have enabled the proliferation of detailed models from principled models of single neurons (Hay et al. (2011))

to large scale biophysically detailed neural networks (Markram et al. (2015)) that enable multi-scale interpretation from cell spiking to local field potentials to macroscale magneto- and electroencephalographic (MEG/EEG) signals (Neymotin et al. (2020); Hagen et al. (2018); Markram et al. (2015); Billeh et al. (2020)). Numerous detailed models are now openly distributed to encourage their use and expansion (McDougal et al. (2017); Dai et al. (2020); Gleeson et al. (2019); Neymotin et al. (2020); Billeh et al. (2020); Markram et al. (2015); Borges et al. (2022); Dura-Bernal et al. (2022b,a)). A common goal of detailed neural modeling is to infer biophysical parameters in individual cells and/or network connections that can account for observed changes in neural activity over time. Given the large-scale nature of any detailed model, parameter inference is an inherently challenging and unsolved problem. The difficulty of parameter inference is closely tied to the level of biophysical detail in the model, as the number of parameters increases with more realistic models. In practice, parameters can not all be estimated simultaneously, but rather model elements are estimated in a serial fashion (e.g. cell dynamics followed network connectivity) and fixed. Then, a limited set of parameters are chosen as the target for estimation. This limited set is chosen based on a prior hypothesis that a certain set of unknown parameters can be estimated based on data features. Even with this limited set, the parameter estimation process is complex. The problem is confounded by the fact that there may be many parameter configurations that produce an equally good representation of the data (Marder and Goaillard (2006)). In this study, we focus on the latter problem of identifying parameter indeterminacies in a limited set of parameters, as to date there is no means to estimate large numbers of parameters at once. The goal is not to present a method to mitigate indeterminacies, but develop a framework that can fully represent them. Identifying unique biophysically constrained parameter sets that can account for observed neural dynamics, and differences across experimental conditions, is essential to the meaningful use of large scale biophysically detailed models for neuroscience research. For example, if you want to use a biophysically detailed model to infer circuit level mechanisms generating an EEG waveform that is a biomarker of a

healthy compared to neuropathological condition, you need a way not only to estimate the parameter distributions that can generate the waveforms but also to assess if the distributions are distinguishable.

A powerful approach to estimate parameters in neural models is Bayesian inference. There is an extensive history of research applying Bayesian inference, and specifically the algorithm of variational inference, to estimate parameters in reduced models of neural activity, for example in the Dynamic Causal Modeling (DCM) (Kiebel et al. (2006)) framework that relies on reduced "neural mass models". However, while compatible with reduced models that are mathematically tractable, the algorithm of variational inference is not compatible with detailed biophysical models due to their computational complexity, and specifically lack of access to a known likelihood function (see also Discussion). In recent years, simulation based inference (SBI) has been proposed as an alternative Bayesian inference framework to estimate parameters in detailed neural models. SBI overcomes the challenge of not having access to a likelihood function by leveraging advances in deep learning to perform density estimation (Cranmer et al. (2020); Greenberg et al. (2019); Gonçalves et al. (2020)). An advantage of SBI is that it only relies on a dataset of simulations from the model being investigated, rather than requiring knowledge of a parameter likelihood function, which is typically not accessible in large-scale biophysically detailed models. From this dataset, a neural density estimator (i.e. artificial neural network specifically made to approximate probability distribution functions) is then trained to learn a mapping of observed neural dynamics (e.g. time series waveforms) to corresponding model parameter distributions. Another advantage is that SBI estimates a full distribution over model parameters that may account for the data and provides information about parameter interactions (Marder and Goaillard (2006); Gonçalves et al. (2020)). This information is not possible with optimization techniques that have historically been used in large-scale biophysical models such as COBYLA (Powell (1964); Neymotin et al. (2020)) and genetic algorithms (Druckmann et al. (2007); Hay et al. (2011)), which estimate only

a single parameter configuration that best fits the data. Figure 3.1 outlines the overall SBI workflow to estimate parameters and possible parameter combinations that can reproduce recorded neural activity.

While the substantial methodological advancements offered by SBI are promising, and have been applied to estimate parameters in small biophysically detailed neuron models (Gonçalves et al. (2020); Lueckmann et al. (2017)) and in models with reduced representations of neural dynamics (Coelho Rodrigues and Gramfort (2020)), there is currently little guidance on how these methods should be used with large-scale biophysical models, with the notable exception of (Hashemi et al. (2023)) offering a thorough discussion of using SBI on simplified models coupled in a large-scale brain network. Guidance is particularly lacking in the context of performing inference on neural time series data, and in comparing estimates for data from different experimental conditions. In this paper, we provide guidelines on how SBI can be applied to estimate parameters underlying time series waveforms generated by biophysically detailed neural models. We emphasize the importance of the first steps of (i) identifying the parameters and ranges over which the inference process will be performed (i.e. prior distribution), which necessarily depends on user-defined hypotheses, and (ii) of selecting informed summary statistics of the waveform activity. We also describe how diagnostics can be used to assess the uniqueness and quality of the posterior estimates and to assess the overlap of distributions from estimation applied to two different waveforms. These evaluation steps are particularly important to resolve the uniqueness of distributions for two or more waveforms.

We begin with a simplified example of a non-linear resistor-capacitor circuit, and then extend the results to an example large-scale biophysically detailed modeling framework that was developed by our group to study the multi-scale neural origin of human MEG/EEG signals, namely the Human Neocortical Neurosolver (HNN) (Neymotin et al. (2020)). The foundation of HNN is a biophysically detailed model of a neocortical column, with layer specific synaptic activation representing thalamocortical and cortico-cortical

drive (Figure 3.2). HNN has been applied to study the cell and circuit origin of commonly measured MEG/EEG signals, including low frequency oscillations (e.g. Beta Events (Sherman et al. (2016)) and event related potentials (ERPs) (Jones et al. (2007, 2009))), along with differences across experimental conditions (Kohl et al. (2022); Jones et al. (2009, 2007); Law et al. (2022); Thorpe et al. (2021)). We demonstrate applications of SBI to estimate parameter distributions that can account for variation in example Beta Events based on empirical data and in simulated ERP waveforms with selected parameter priors (see Step 1 in Figure 3.1) based on our previous studies (Jones et al. (2009, 2007); Sherman et al. (2016)). We show that due to the model complexity some parameters can be inferred uniquely while others are indeterminate. The methods described and proof of concept examples provide a principled foundation to guide future applications of SBI in a wide variety of applications that use detailed models to study neural dynamics.

3.1.3 Materials and methods

Below we provide a summary of the primary techniques used in this work. Specific aspects are emphasized to provide better context on the significance/motivation of analyses performed. We invite readers to refer to the citations for a more thorough treatment of each subject. In particular, the software publication detailing HNN (Neymotin et al. (2020)), and a study demonstrating the use of SBI on classical neural models (Gonçalves et al. (2020)). We also detail an RC circuit example, which is a building block of HNN-type models, and which offers a SBI setup with time series inputs and the presence of indeterminacies.

Resistor-capacitor circuit simulations

Resistor-capacitor (RC) circuit simulations were performed using the `odeint` ordinary differential equation (ODE) solver of the `SciPy` Python package. A more thorough description of how RC circuit simulations were performed can be found in the results

section RC circuit: a simple example to describe indeterminacies that can occur with time series inference.

Human Neocortical Neurosolver

Biophysical modeling of cortical activity underlying MEG/EEG signals was performed using the Human Neocortical Neurosolver (HNN) (Neymotin et al. (2020)). The standard HNN model used in this study and described in (Jones et al. (2009)) is composed of 100 pyramidal neurons and 33 inhibitory neurons in both layers 2/3 (L2) and 5 (L5) for a total of 266 neurons and represents a localized small patch of neocortex (Figure 3.2). To accurately reproduce macroscale electrical signals, HNN utilizes multi-compartment pyramidal neuron models (Ermentrout and Terman (2010)), and synchronous intracellular current flow of aligned L2 and L5 pyramidal neuron dendrites is assumed to be the generator of the primary electrical current sources underlying recorded extracranial MEG/EEG signals, due to their length and parallel alignment (Hämäläinen et al. (1993)). Inhibitory basket cells are modeled as single compartment point neurons given their negligible impact in producing the recorded electrical currents, but are none-the-less crucial to the local network dynamics (see Neymotin et al. (2020) for further background). Pyramidal cells in the model connect to other cells with both AMPA and NMDA synapses, while basket cells produce $GABA_A$ and $GABA_B$ synaptic connections.

In addition to the local circuitry, HNN models extrinsic inputs to the column via layer specific synaptic excitatory drives generated by predefined patterns of action potentials presumed to come from exogenous brain areas (i.e. thalamus and other cortical regions). In general these are referred to as proximal and distal drives, reflecting the effective location of the synapses on the pyramidal neuron proximal and distal dendrites. The proximal drive reflects so called feedforward inputs from lemniscal thalamus, and the distal drive reflecting inputs from either non-lemniscal thalamus (Barbas et al. (2013); Jones (2001)), or "feedback" connections from other cortical regions. These proximal and

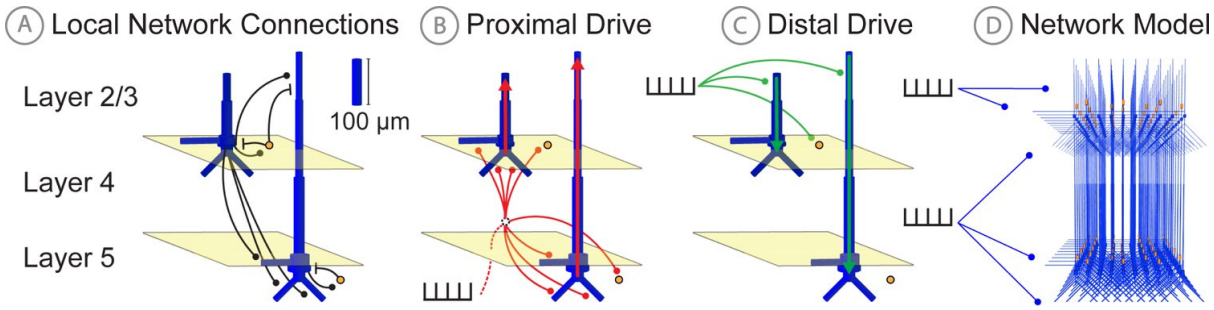

Figure 3.2: **HNN model schematic.** **A**: Local network connections of the HNN model include: 1) excitatory AMPA and NMDA synaptic connections (black circles) originating from pyramidal neurons (blue), and 2) inhibitory $GABA_A$ and $GABA_B$ synaptic connections (black lines) originating from inhibitory interneurons (yellow). **B**: Proximal exogenous input connection pattern. **C**: Distal exogenous input connection pattern, see text for further description. **D**: 3D rendering of full neocortical column model. Figure adapted from Neymotin et al. (2020).

distal drives induce excitatory post-synaptic currents that drive current flow up and down the aligned pyramidal cell dendrites (see red and green arrows in Figure 3.2) that can generate positive and negative deflections in the primary electric current dipole of source localized MEG/EEG signals. Intracellular current flow due to these extrinsic inputs, as well as induced local spiking dynamics, combine to produce the recorded MEG/EEG signal.

The results detail 3 examples of problems suited for parameter inference in HNN. In the first two examples, the parameters to be estimated control either the timing or strength of the input. In the first example (Figure 3.5), the target of estimation is the parameter vector $\theta_0 \in \mathbb{R}^3$, where the first two parameters control the strength of a single spike (action potential) of excitatory proximal and distal input (in units of nS for the synaptic conductance $\bar{g}$), and the third parameter Δt controls the timing between the two inputs (in units of milliseconds (ms)). In the second example Figure 3.7, the target of estimation is $\theta_0 \in \mathbb{R}^2$ where the parameters control the variance (ms^2) of a Gaussian distribution from which 10 spikes are drawn for the proximal and distal inputs. In the final example Figure 3.8, the target of estimation is $\theta_0 \in \mathbb{R}^4$, where the parameters control the synaptic strength (nS) of excitatory/inhibitory local connections onto the proximal/distal

dendrites of L5 pyramidal neurons.

Posterior Diagnostics

When working with a posterior approximation Φ, it is useful to characterize its behavior in different regions of the parameter space. For a given parameter configuration θ_0 and simulated output $x_0 \sim p(x \mid \theta_0)$, we quantify how concentrated $\Phi(\theta \mid x_0)$ is around θ_0. We refer to this quantity as the parameter recovery error (PRE) and define it as the Wasserstein distance between the k-th marginal of $\Phi(\theta \mid x_0)$, and a Dirac delta centered at $\theta_0[k]$. This can be empirically estimated as per

$$\mathrm{PRE}_k(\Phi, \theta_0) = \frac{1}{N} \sum_{i=1}^{N} \Big(\theta_i[k] - \theta_0[k]\Big)^2 \tag{3.1}$$

where we generate N samples $\{\theta_1, \ldots, \theta_N\} \sim \Phi(\theta \mid x_0)$ from our posterior approximation conditioned at $x_0 \sim p(x \mid \theta_0)$, which is an observation from the simulator at ground truth θ_0. Note that θ_0 is composed of k distinct values for each individual parameter, therefore there are k distinct PRE values. Additionally, each parameter $\theta_0[k]$ is mapped from its range defined in the prior distribution, to the range (0,1). Therefore, the maximum PRE is 1.0, indicating the worst possible recovery, whereas a PRE of 0.0 indicates perfect recovery.

The previous diagnostic quantified how well Φ represents the relationship between x_0 and θ_0 in terms of the parameter space. Alternatively, we can assess the relationship in the observation space. Specifically, given samples from our approximate posterior $\theta_i \sim \Phi(\theta \mid x_0)$, we generate simulations $x_i \sim p(x \mid \theta_i)$, and assess how close x_i is to the conditioning observation x_0. This is known as a posterior predictive check (PPC) (Gelman et al. (1996); Box (1980)) and is quantified in this manuscript as the root mean squared

error between x_0 and the x_i as per

$$\mathrm{PPC}(x_i, x_0) = \sqrt{\frac{1}{N}\sum_{i=1}^{N} \|x_i - x_0\|^2} \,. \tag{3.2}$$

.

We note that this is a simplified treatment of how to perform a PPC, as observation and neural noise may significantly change the interpretation of model fit. A more robust assessment of PPC in the context of systems neuroscience modeling can be found in (Hashemi et al. (2021)) which describes measures such as the Watanabe–Akaike information criterion (WAIC), as well as alternative measures to PRE for assessing posterior fit such as posterior shrinkages.

Lastly, in many applications it is useful to characterize how well two distributions can be distinguished from one another. To this end we introduce the distribution overlap coefficient (OVL) (Pastore and Calcagnì (2019)) which varies on a scale of [0,1] such that 1 indicates complete overlap, and 0 indicates no overlap. Given our approximate posterior Φ, we define OVL as:

$$\mathrm{OVL}_k(\Phi, x_0, x_1) = \int \min\Big(\Phi(\theta[k] \mid x_0), \Phi(\theta[k] \mid x_1)\Big) d\theta \tag{3.3}$$

where x_0 and x_1 are two different observations whose posterior distributions we seek to compare on the marginal distribution for the k-th parameter. To calculate the OVL_k numerically we used an evenly spaced grid of 50^d for a prior distribution over $d \in \mathbb{N}_+$ parameters. OVL_k operates similarly to Kullback–Leibler divergence and related measures, with the primary advantage being that it is bounded on the interval [0,1] readily permitting identification of distributions with large and small amounts of distribution overlap.

MEG Beta Events

Beta Event examples represent empirical human MEG data source localized to SI taken from the previous studies (Jones et al. (2009); Shin et al. (2017); Sherman et al. (2016)) and preprocessed as described in Sherman et al. (2016). Subjects 7 and 5 from Sherman et al. (2016) were chosen as exemplars for large (blue) and small (orange) Beta Events, and here are referred to as subjects 1 and 2, respectively. The Beta Events used for inference were averaged over all recorded events for each subject that were included in the dataset: 378 events for subject 1, and 376 events for subject 2. The full dataset of source localized MEG Beta Events used in this study can be found in the Open Science Framework repository associated with this manuscript: https://doi.org/10.17605/OSF.IO/VZ97X.

3.1.4 Results

Approach to applying SBI in biophysically detailed models that simulate time series data

Recently SBI has been established as an approach to estimate parameters in detailed biophysical models (Gonçalves et al. (2020)) that simulate time series data. This approach overcomes the challenges of applying Bayesian inference in highly detailed non-linear models, namely estimation of complex posterior distributions that exhibit parameter interactions, by leveraging recent advances in likelihood-free inference and deep learning (Greenberg et al. (2019)). We begin by reviewing the SBI process and providing the mathematical description of each of the steps outlined in Figure 3.1.

The primary goal of SBI in the context of our manuscript is to estimate parameters and possible parameter distributions that can account for an observed neural dynamic (e.g. time series waveform). In mathematical terms, this goal is stated as follows. Given an observation x and model with parameters θ, SBI seeks to create an approximation

$\Phi(\theta \mid x)$ of a posterior parameter distribution $p(\theta \mid x)$ such that

$$\Phi(\theta \mid x) \approx p(\theta \mid x) \propto p(x \mid \theta)\ p(\theta)\ . \tag{3.4}$$

Bayes' rule specifies a closed form for the desired posterior distribution $p(\theta \mid x)$ as being proportional to the likelihood $p(x \mid \theta)$ multiplied with the prior $p(\theta)$ (Bishop and Nasrabadi (2006)). Unfortunately, in detailed biophysical models the likelihood function $p(x \mid \theta)$, which encodes the relationship between model parameters θ and outputs x, is often analytically intractable but can be approximated from a large number of simulations. The novelty of SBI is that it circumvents likelihood evaluations altogether, and instead approximates the posterior distribution directly from a simulated dataset of model outputs $x_i \sim p(x \mid \theta_i)$ with parameter values sampled from a user defined prior distribution $\theta_i \sim p(\theta)$. There are numerous approaches in the literature to achieve this goal, each with their own unique considerations, benefits, and challenges (Franks (2020)). In this study, we use a deep learning architecture known as a conditional neural density estimator (Φ), which is a function that takes an observation x as input, and returns a probability density function defined over the parameter space. More specifically, the conditional neural density estimator utilizes normalizing flows following standard practices described in (Greenberg et al. (2019); Gonçalves et al. (2020)). The detailed steps in applying SBI (Figure 3.1) are as follows.

Step 1: Define prior distribution. SBI begins with the user choosing the parameters of interest to be estimated, and the range and statistical distribution of values (e.g. uniform distribution) those parameters can take. This constitutes the prior distribution $p(\theta)$ over model parameters θ to be inferred. The importance of a well-constructed prior cannot be understated, as it encodes the assumptions and hypotheses of the inference problem considered, and strongly impacts any resulting predictions. Creating a good prior distribution requires domain expertise to choose meaningful parameters, and a biologically

realistic range of values. That is not to say that the predictions are predetermined by the prior, as uncertainty can be encoded using flat/uninformative priors where the probability mass is evenly spread over the desired parameter range. Nevertheless this aspect is highly important for detailed neural models where the inferred parameters represent a small subset of the total set of parameters.

Step 2: Generate training data. With the prior constructed, a simulated dataset of observations $x_{1:N}$ is generated by simulating a large number of N time series using model parameters $\theta_{1:N}$ drawn from the prior parameter distribution.

Steps 3-4: Training neural network based on chosen summary statistics to describe the time series waveform. With the simulated dataset, a specialized deep learning architecture known as a conditional neural density estimator Φ is trained to approximate the posterior distribution $p(\theta|x)$ that can account for chosen summary statistics for any observation x. The output is a distribution of parameters which can generate simulations close to the summary statistics of the conditioning observation x_0. The neural density estimator Φ can be trained directly from the entire time series x, or from summary statistics $s = S(x)$ which constitute a lower dimensional vector of values. The choice of s should aim to obtain $\Phi(\theta \mid s) \approx p(\theta \mid s) \approx p(\theta \mid x)$, meaning that the posterior estimates are well enough approximated from s alone. A more in depth description of the choice and role of summary statistics is given after describing Steps 5-6.

Steps 5-6: Estimate and compare parameter distributions for distinct waveforms. With a trained neural density estimator, users can finally feed in new waveforms and assess the predicted parameter distributions underlying their generation. Diagnostics that assess the quality of the distribution can then be performed (see Materials and methods for details on the calculation of each diagnostic). If the ground truth parameters underlying the waveform of interest are known, users can calculate the dispersion of the posterior around this ground truth using parameter recovery error (PRE). This is known as face validity when using simulated data to check if the ground

truth parameters can be recovered (Kiebel et al. (2007); Stephan et al. (2010)). If the ground truth is unknown, users can use posterior predictive checks (PPC) to assess if the parameter distribution consistently produces waveforms close to the conditioning observation. Finally, uniqueness of posterior estimates for two different waveforms can be assessed by directly comparing the overlap coefficient of the distributions (OVL).

The important role of summary statistics in parameter inference: Using the inference framework described above, in this study we emphasize the role of summary statistics and how their selection directly impacts predicted parameter distributions. Inference on models with full time series outputs is challenging because the observations are high dimensional. This challenge comes in the form of interpretability and model misspecification due the simulator not capturing finer characteristics of the real data generating process (Ward et al. (2022)).

Summary statistics can either be hand-crafted, leveraging domain expertise and hypotheses regarding the data, or they can be automatically extracted. A summary statistic $s = S(x)$ is sufficient if all the relevant information for mapping the full observation to its underlying parameters is retained. More specifically, sufficiency is satisfied if $p(\theta \mid s) = p(\theta \mid x)$ (Franks (2020); Coelho Rodrigues and Gramfort (2020)). In practice, truly sufficient summary statistics are rare, but they can still provide a close approximation to inferences achieved when using the full observations x. A major advantage of hand-crafted features is that they can be readily interpretable, and come associated with hypotheses on their physiologic significance depending on previous research. Alternatively, full time series informed approaches like principal component analysis (PCA) may do a better job at retaining more complex relationships between observations and parameters. We note that as an alternative to PCA, automatic extraction of summary statistics through embedding networks are becoming increasingly common (Dax et al. (2021a,b); Greenberg et al. (2019); Coelho Rodrigues and Gramfort (2020)). However, a systematic analysis of the numerous architectures employed in this domain is outside the scope of this study.

Given the current ambiguity around summary statistic selection, principled approaches to compare alternate approximations of the posterior distribution are essential. To this end, we have constructed educational examples which allow for an intuitive understanding of the role of different summary statistics. Additionally, we introduce diagnostic analyses that can be used to quantitatively compare desirable properties of posterior approximations produced using different summary statistics, such as PPC, PRE, and OVL. Note that we use the term "desirable properties" as the goals of inference may differ depending on the use-case. For example, predicting precise values for biophysical values (parameter recovery) may be one goal. Alternatively, characterizing the range of parameters consistent with a known biomarker may be another goal. These analyses are detailed in the examples below.

RC circuit: a simple example to describe indeterminacies that can occur with time series inference

One of the most challenging aspects of likelihood free inference is that the models studied typically do not permit access to a ground truth posterior distribution, over which we can validate our inferences. To highlight this challenge and better understand how decisions in the SBI pipeline impact the resulting approximate posterior, we will first apply the SBI pipeline on a model where the ground truth posterior is known; namely an RC circuit model.

The equation for the RC circuit simulations is as follows:

$$C\frac{\mathrm{d}V}{\mathrm{dt}}(t) = \frac{E - V(t)}{R} + I_{\mathrm{e}}(t) \tag{3.5}$$

where $V(t)$ is the voltage response of an RC circuit to a current injection $I_{\mathrm{e}}(t)$. In our example we use a capacitance $C = 6\,\mathrm{F}$, resistance $R = 1\,\Omega$, and constant voltage source $E = 0\,\mathrm{V}$ (Abbott and Dayan (2001)).

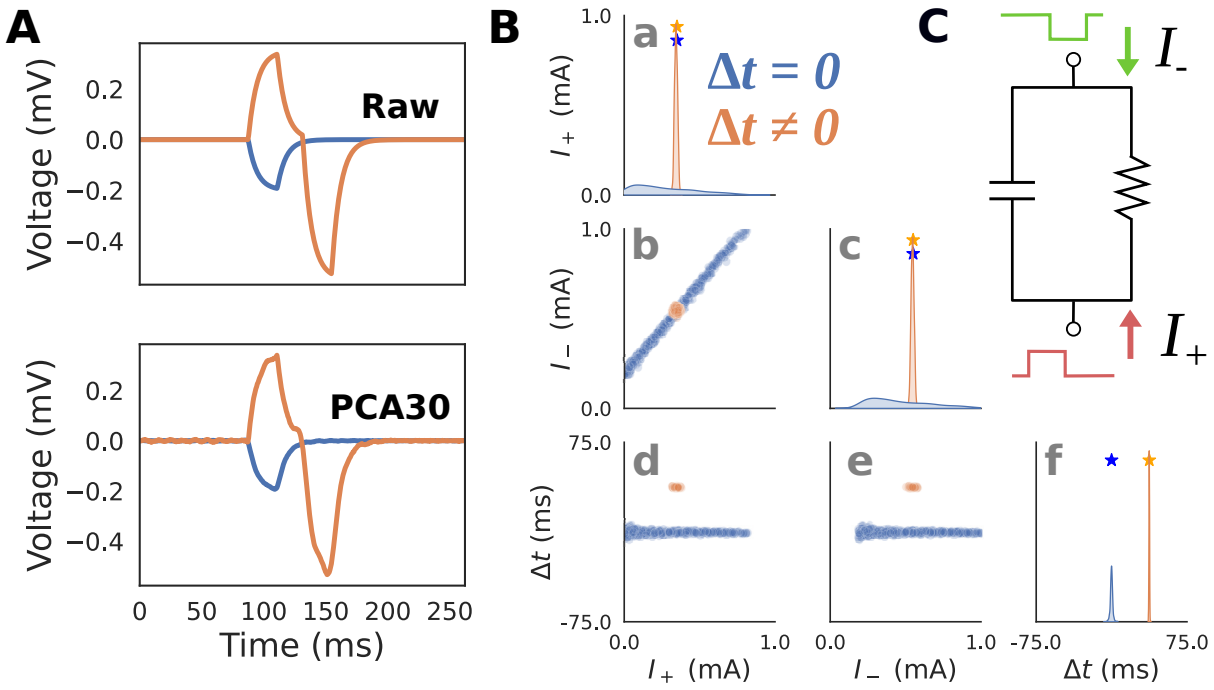

Figure 3.3: **RC Circuit Simulations.** **A**: Simulated voltage and current dipole waveforms are shown for two exemplar parameter configurations with latency between positive I_+ and negative I_- square current injections, $\Delta t = 0$ (blue), and $\Delta t \neq 0$ (orange). An example of the original "Raw" waveform (top), as well as the PCA transformed waveform (bottom) with the first 30 components (`PCA30`) are shown to demonstrate that this summary statistic retains almost identical information. **B**: Posterior distributions showing the inferred values that can generate the blue and orange waveforms from panel A (`PCA30` used to generate distributions) demonstrate that when the latency Δt between the inputs is zero, their amplitudes are indeterminate as visible as a highly dispersed distribution on panels B(a-c, blue), and with a positive correlation between the parameters I_+ and I_- on panel B(b, blue). In contrast, when $\Delta t \neq 0$ (orange), the distributions are tightly concentrated around the ground truth parameters (stars on panels B(a,c)) used to generate each simulation. The posterior distributions for the parameter Δt are concentrated around the ground truth parameters for both conditions (panel B(f)). **C**: Schematic of how RC circuit is driven by positive I_+ and negative I_- square current injections, where the amplitude and latency Δt between pulses serve as parameters. This simulation parallels HNN simulations below (Figure 3.5) in which a single excitatory proximal/distal input with variable synaptic conductances and latencies produce positive/negative deflections in the current dipole.

The example RC circuit simulations described here were parameterized in a way that will enable comparison to similar simulations in the biophysically detailed simulations described below for HNN (Figure 3.3C). More specifically, we drive the RC circuit with two square wave pulse injections, with positive and negative amplitudes lasting 20 ms each. Two parameters (I_+ and I_-) controlled the magnitude of positive and negative square

pulse current injections, such that the sum $I_e = I_+ + I_-$ determined the final injected current for each time step. A third parameter, latency Δt, controlled the time delay between the two pulses. Specifically, Δt shifts the negative current pulse in time, while the positive current pulse remains fixed. These inputs play a similar role as excitatory proximal and distal inputs in HNN (Figure 3.2). Table 4.1 details the prior distribution over the parameters used to generate training examples for SBI.

The relationship between current injection amplitude and the RC circuit voltage response strongly depends on if Δt is zero or non-zero. When Δt is zero, the total injected current sums to a single square pulse since the negative and positive pulses perfectly overlap. Figure 3.3A(top, blue) shows an example of this where a -0.2 mA square pulse current injection is delivered at 80 ms producing a voltage response with an exponential rise and decay with one peak. Since any combination of I_+ and I_- which sum to -0.2 mA will produce an identical voltage, there will be an indeterminacy when attempting to infer these parameters from the voltage response. In contrast, Figure 3.3A(top, orange) shows an example with a non-zero latency $\Delta t \neq 0$. Specifically, the first current injection with $I_+ = 0.3$ mA is delivered at 80 ms, and the second current injection with $I_- = 0.5$ mA at 117.5 ms, therefore $\Delta t = 37.5$ ms. The voltage response exhibits two unique peaks due to the offset between the square pulses. Since there is only one combination of I_+ and I_- that can produce this waveform, their values can be inferred exactly from the waveform. In other words, the amplitude parameters underlying the voltage response can be inferred exactly only when $\Delta t \neq 0$. Note that since we are approximating the posterior distribution, even values close to $\Delta t \approx 0$ will still produce an indeterminacy.

To visualize and interpret posterior distributions produced by SBI, we must first draw a sufficient number of random samples from the distribution. Since posterior distributions are often multidimensional, it is useful to plot the samples using a "pair-plot" (Figure 3.3B). The diagonal of the pair-plot is used to visualize the univariate (also known as marginal) distribution of each parameter, here using a kernel density estimate of $n = 1000$ posterior

samples. The squares below the diagonal on the other hand visualize the bivariate distribution for pairs of parameters by plotting the samples explicitly on a scatter plot. This example highlights that even with simple simulators, indeterminacies can easily arise necessitating the use of flexible posterior approximators (like masked autoregressive flows).

Note that PCA with 30 components (`PCA30`, variance explained=0.883) was used as the summary statistic in this example (Figure 3.3A(bottom)) rather than the full time series to avoid the potential computational issues of conditioning posteriors on high dimensional data, while still retaining the majority of the waveform variance. Figure 3.3A (bottom) plots the inverse transformed `PCA30` waveform to highlight that the summary statistic retains almost identical information. In all subsequent examples, `PCA30` and `PCA4` will refer to the $d = 4$ and $d = 30$ dimensional summary statistic vectors containing the loadings on the first 4 and 30 principle components. The PCA transformation was fit separately for each example using the dataset of 100,000 simulated time series waveforms (see Supporting information for simulation details).

The results in Figure 3.3B show that the expected posterior distribution described above can be recovered with SBI. As shown in Figure 3.3B(a-e), when conditioned on the voltage response with $\Delta t = 0$, any distribution involving I_+ or I_- (blue) will exhibit an indeterminacy (i.e., multiple recovered values along one dimension). The high correlation between I_+ and I_- of 0.998 ($p \approx 0$) demonstrates that the indeterminacy is characterized by a strong linear interaction between these parameters. Specifically, the line in Figure 3.3B(b, blue) corresponds to all values in which the amplitudes sum to a constant value of $V = -0.2$, and the resulting voltage waveform is identical. In contrast, the voltage response with $\Delta t \neq 0$ (orange) produces a posterior distribution concentrated on a single point around the ground truth parameters (Figure 3.3B(b, orange), correlation between I_+ and I_-: 0.375; $p <$ 1e-33).

Diagnostics enable comparison of posterior estimates using different summary statistics

In the previous example, we utilized `PCA30` as a summary statistic to learn a low dimensional representation of the voltage time series. PCA is a common choice for dimensionality reduction, and has been used in historical MEG/EEG inference work with only the first 3-4 principal components (Kiebel et al. (2006, 2007)). However, it is not guaranteed that PCA, which aims to only capture variance, will retain the features that best allow SBI to map waveforms to simulation parameters. An alternative approach is to leverage domain-expertise to construct hand-crafted summary statistics specific to the model and inference problem. Unfortunately, it cannot be known a priori which summary statistic will allow SBI to perform best, necessitating quantitative diagnostics that allow a systematic comparison. Here, we introduce two simple hand-crafted summary statistics, as well as the posterior diagnostics PRE and PPC, with the intention to build an intuitive understanding of how emphasizing different summary statistics can impact the final estimates produced by SBI.

The first hand-crafted summary statistic we defined is a four dimensional vector including the magnitude and timing of the maximum and minimum peaks (`Peak`) in the time domain of the simulated voltage response (Figure 3.4A(iii)). It can be readily seen that these features reflect the underlying simulation parameters. Upon visual inspection of the voltage response with $\Delta t \neq 0$ (orange), the height of the maximum and minimum peak directly correspond to the parameters I_+ and I_-, and the distance between these peaks correspond to the latency parameter Δt. We will show below that `Peak` features permits inference that is close to that achieved with `PCA30` Figure 3.4A(i) and also `PCA4` (variance explained=0.610) Figure 3.4A(ii).

The second hand-crafted summary statistic we defined was a four dimensional vector including the band power (`BandPower`) of common frequency ranges used to study neural oscillations (Figure 3.4A(iv)). Specifically, we considered the beta (13-30 Hz), low-gamma

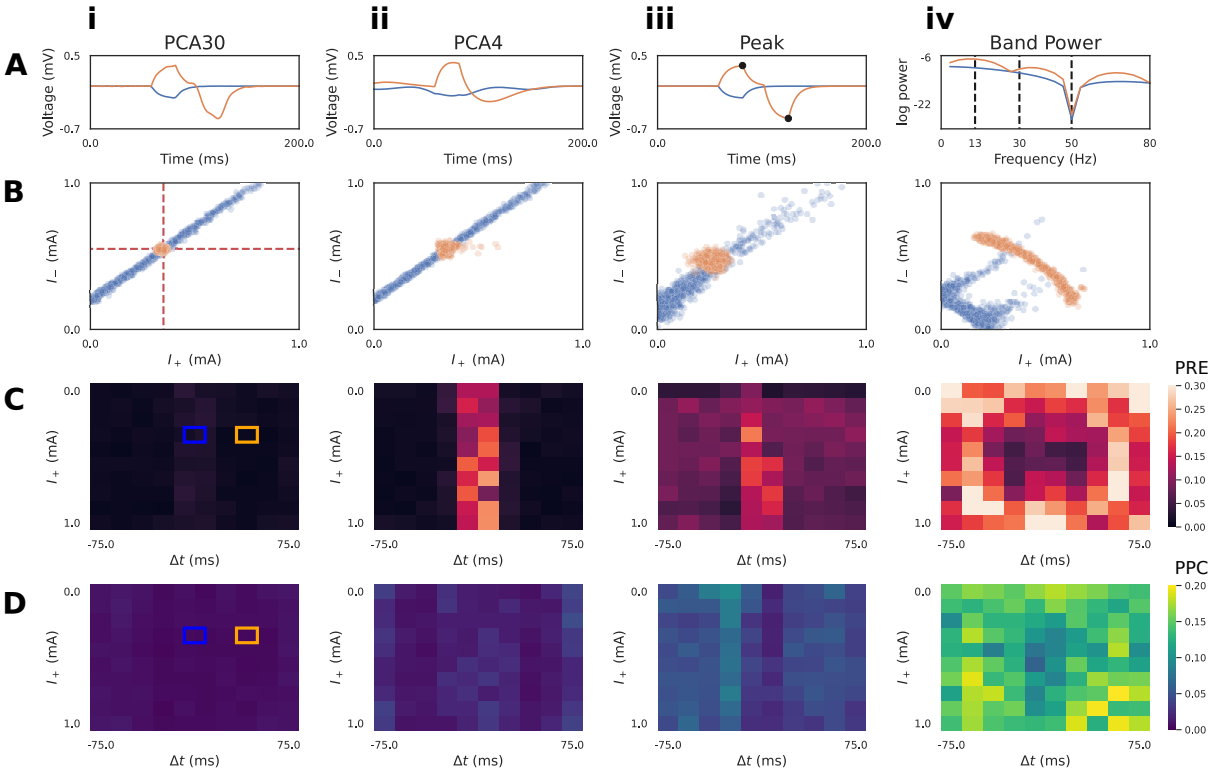

Figure 3.4: **Diagnostics to compare summary statistics on RC circuit model. A**: Summary statistics applied to the simulated time series included: `PCA30` (i), `PCA4` (ii), `Peak` (magnitude and timing of max/min, iii), and `BandPower` (four bands between dotted lines, iv). PCA plots (i-ii) show the associated inverse transformed signal. Two exemplar simulations with pulse latencies $\Delta t = 0$ (blue) and $\Delta t \neq 0$ (orange) are shown. **B**: Conditioning the approximate posterior distribution on the $\Delta t = 0$ time series produces indeterminacies for all summary statistics, such that the ground truth (red dotted lines) current injection amplitudes (I_+ and I_-) cannot be uniquely recovered. `PCA30`, `PCA4`, and `Peak`features exhibit a linear interaction between parameters for the $\Delta t = 0$ time series, whereas the ground truth is recovered for the $\Delta t \neq 0$ (orange) time series. `BandPower` produces non-linear interactions for both time series. **C**: Local parameter recovery error (PRE) heatmaps are shown. Brighter colors indicate higher dispersion of the posterior around the ground truth parameters defined by each square. Errors tend to be concentrated around $\Delta t = 0$ for `PCA30`, `PCA4`, and `Peak` features. **D**: Local posterior predictive check (PPC) heatmaps are shown. Brighter colors indicate regions where simulations generated from posterior samples are further from the conditioning time series. For both diagnostics, it is clear that `PCA30` produces the lowest PRE and PPC across the parameter grid. The ground truths of the exemplar simulations of panels A/B are indicated by blue/orange squares.

(30-50 Hz), and high-gamma (50-80 Hz) ranges, as well as the aggregate band power of the alpha and lower frequency ranges (0-13 Hz). As we will show below, this feature was intentionally selected as a cautionary example of a summary statistic that is ill-suited for

the inference problem, but has a basis in previous neuroscience and Bayesian inference literature (Buzsáki and Draguhn (2004)).

Next, we describe two diagnostics that allow comparison of desirable properties of the posterior distribution for different summary statistics, namely PRE and PPC (see Figure 3.1 Step 6 and Materials and methods). In the results below, we calculated PRE values over a grid, with 10 points for every parameter dimension, spanning the range of the prior distribution (Figure 3.4C).

Taking inspiration from Zhao et al. (2021), we visualize the PRE as heatmaps over the parameter grid (Figure 3.4C-D). One of the advantages of inspecting local posterior diagnostics ("local" as in specific to the pair (x_0, θ_0)) is the ability to identify patterns. Figure 3.4C plots the PRE for the I_+ parameter, with respect to different ground truth values of I_+ itself, and the Δt parameter. Blue and orange squares mark the ground truth values used to generate the waveforms in Figure 3.4A. We observe that the summary statistics `PCA30`, `PCA4`, and `Peak` all exhibit a pattern where Δt values near zero produce a larger PRE compared to the rest of the parameter grid (Figure 3.4C(i-iii)). This is due to the indeterminacy in I_+ as seen in the blue posterior samples of Figure 3.4B(i-iii). It is apparent, however, that `PCA30` values produce the lowest PRE values, even near a Δt of zero. In contrast, the `BandPower` summary statistic produces a posterior distribution with complex indeterminacies for both observations. This results in high PRE values across the entire parameter grid (Figure 3.4C(iv)), indicating that this summary statistic is not effective at recovering the ground truth parameters.

The PPC is a method to describe how well samples from the posterior match the conditioning observation (Gelman et al. (1996); Box (1980)). Given a well-estimated posterior distribution $p(\theta \mid x)$ and a conditioning observation $x_0 \sim p(x \mid \theta_0)$, one would expect simulations $x_i \sim p(x_0 \mid \theta_i)$ to be close to the original conditioning observation. Unlike the PRE heatmaps, the PPC plots shown in Figure 3.4D does not exhibit obvious patterns with respect to the underlying parameter grid.

These diagnostics demonstrate `PCA30` performs the best for this inference problem. Additionally, the local PRE analysis revealed differences between summary statistics that were not apparent with the global diagnostics. It is important to note that neither of these diagnostics quantify the closeness of the approximation Φ to the true posterior. For instance, if there is an interaction between parameters of the model causing a parameter indeterminacy, then the PRE will always be non-zero, since the posterior distribution will be spread in the parameter space. This is the case for the posterior of the RC circuit with $\Delta t = 0$ in Figure 3.3B. Similarly, if model simulations are stochastic, then a given set of parameters may map to multiple equally valid outputs, producing a non-zero PPC. Nevertheless, both diagnostics provide useful information to compare desirable properties of the posterior approximation.

Subthreshold HNN simulations mimicking the RC circuit shows that inference with summary statistics that account for the full time series waveform perform best

Building from the RC circuit, we now describe a nearly identical inference problem in our large-scale biophysically detailed model constructed to study the neural mechanisms of human MEG/EEG, HNN (Figure 3.5). As described further in Materials and methods, HNN is a neocortical column model with multiple layers and biophysically detailed local cell types. The local network receives exogenous excitatory synaptic input through layer specific pathways that effectively synapse on the proximal and distal dendrites of the pyramidal neurons, representing "feedforward" and "feedback" input from thalamus and higher order cortical areas. These inputs are simulated with external "spikes" that activate layer specific excitatory synapses (see Figure 3.2B-C, and reduced schematic in Figure 3.5C). This synaptic activity induces current flow within the pyramidal neuron dendrites, which is summed across the population to simulate a net current dipole that is directly comparable to that measured with MEG/EEG. Several previous studies have shown that patterns of activation of the local network through these pathways can generate

commonly measured MEG/EEG current dipole signals such as event related potentials and low frequency brain rhythms, e.g., see Neymotin et al. (2020).

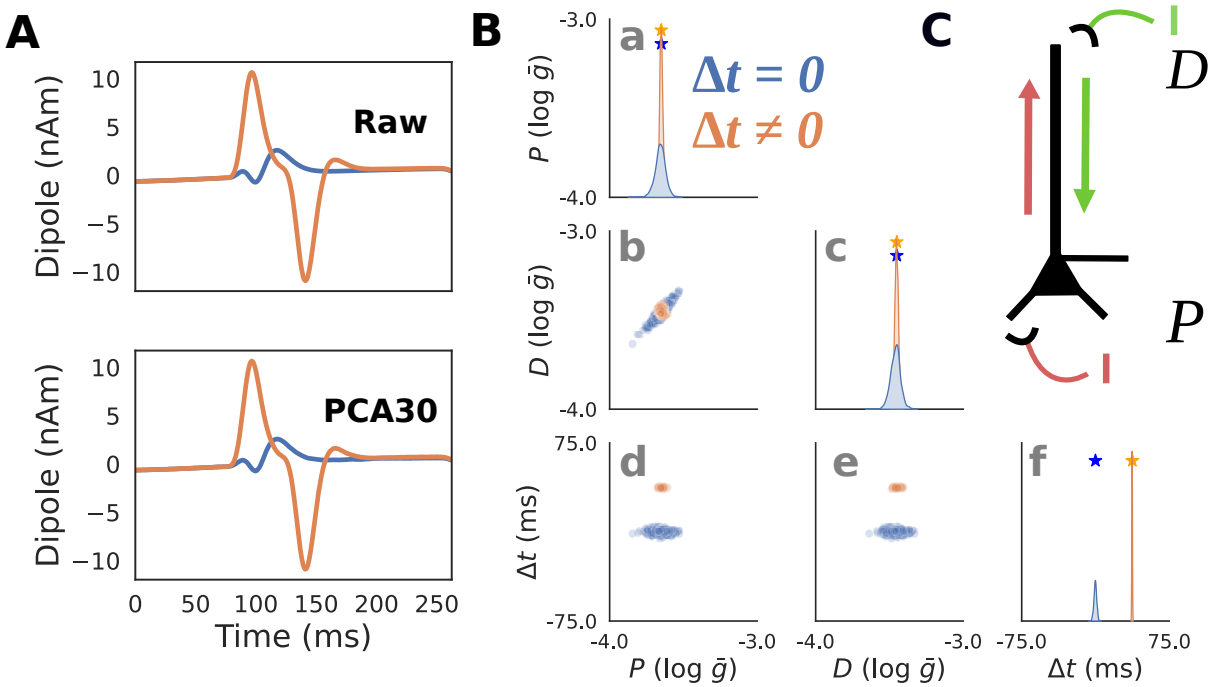

Figure 3.5: **HNN simulations that mimic RC circuit.** HNN simulations that reflect the nearly identical parameter configuration as the RC circuit in Figure 3.3. **A**: Simulated current dipole waveforms are shown for two exemplar parameter configurations with $\Delta t = 0$ (blue) and $\Delta t \neq 0$ (orange). The original "Raw" simulated waveform (top) is plotted in comparison with the PCA inverse transformed waveform with 30 components (`PCA30`, bottom). **B**: Posterior distributions showing the inferred values that can generate the waveforms from panel A demonstrate that when the latency between the inputs is zero (blue), their amplitudes are indeterminate as visible as a dispersed distribution on panels B(a-c, blue), with a positive correlation between the parameters P and D on panel B(b, blue). Unlike the previous example (Figure 3.3), the indeterminacy is notably smaller for $\Delta t = 0$, with the posterior distributions primarily being concentrated around the ground truth parameters for P and D (stars on panels B(a,c)). **C**: Schematic of HNN simulations in which a single excitatory proximal/distal input with variable synaptic conductances and latencies produce positive (red)/negative (green) deflections in the current dipole.

To set up an inference problem that is comparable to our RC circuit example, we begin by considering patterns of drive to the network that create subthreshold current flow in the pyramidal neurons, effectively "disconnecting" the network, because local synaptic interactions depend on local cell firing. Simulations with spiking dynamics will be demonstrated in the following section. For simplicity, we first describe the subthreshold

current flow in the L5 pyramidal cells only (Figure 3.5C). Specifically, we ran HNN simulations with single exogenous spikes that activate excitatory synapses on the proximal and distal dendrites of L5 pyramidal cells. Synaptic excitation of distal synapses generates current flow down the dendrites (e.g. see green arrow Figure 3.5C), and excitation of proximal dendrites generates current flow up the dendrites (e.g. see red arrow Figure 3.5C). A delay between the time of the two driving spikes can create a net current dipole signal that is analogous to that observed in the RC circuit for a non-zero time delay between the applied currents (see Figure 3.5A, orange). Further, when the delay between the spikes is zero (see Figure 3.5A, blue) an indeterminacy in parameter estimation can occur, as described below.

With this set up, we applied SBI to infer parameters that mimic those of the RC circuit example, using `PCA30` (variance explained=0.976) as the chosen summary statistic to constrain the inference problem. Namely, the strength of proximal and distal excitatory inputs, referred to as P and D, parameterized as the maximal conductance at their respective synapses and the latency Δt between the two inputs . We kept the proximal input time fixed, and let the distal input time vary with Δt. The prior distribution over P and D was set to ensure that all simulations remained subthreshold (see Table 4.1 for prior distribution ranges).

Similar to the RC circuit example, simulations with $\Delta t = 0$ produce a current dipole with a reduced magnitude (Figure 3.5A(blue)), whereas $\Delta t \neq 0$ produces a clear positive and negative peak (orange). As shown in Figure 3.5B(a-e, blue), when conditioned with $\Delta t = 0$, any posterior distribution involving P and D will exhibit an indeterminacy. Intuitively, this indicates that the proximal and distal inputs can compensate within a small range to produce similar current dipole waveforms. Unlike the RC circuit, this interaction does not span the full range of input strengths, and instead is more tightly concentrated around the ground truth (Figure 3.5B(a,c,f) stars).

Once again, we show that the choice of summary statistics impact the learned posterior

distribution approximation, and that diagnostics can be used to evaluate the quality of the parameter estimation (Figure 3.6A-B). When $\Delta t \neq 0$, both `PCA30` and `PCA4` (variance explained=0.665) produced a posterior that is localized around the ground truth (Figure 3.6B(i-ii), orange). When $\Delta t = 0$, the posterior was still concentrated but with a slight indeterminacy (Figure 3.6B(i-ii, blue)) that was less prominent than the analogous simulation in the RC circuit (Figure 3.4B(i-ii, blue)). The `Peak` summary statistic produced a posterior that is well clustered around the ground truth for $\Delta t \neq 0$ (Figure 3.6B(iii, orange)), but for $\Delta t = 0$ exhibited a much more striking indeterminacy (Figure 3.6B(iii, blue)). In contrast, `BandPower` was insufficient for ground truth recovery for both the $\Delta t \neq 0$ and $\Delta t = 0$ as was the case for the RC circuit simulations (Figure 3.6B(iv)).

We quantified desirable properties of the posterior estimates, with the local PRE and PPC analysis described above. At $\Delta t \approx 0$, PRE values were large when using `BandPower` features (Figure 3.6C(iv)), relatively smaller for `PCA4` and `Peak` (Figure 3.6C(ii-iii)), and almost completely disappears for `PCA30` (Figure 3.6C(i)). The PPC values for the `BandPower` show that inference using this summary statistic produced results that were highly dissimilar to the conditioning observations (Figure 3.6D(iv)), while `PCA30`, `PCA4`, and `Peak` exhibit a much lower PPC values (Figure 3.6D(i-iii)). Local PRE and PPC analysis largely agrees with the summary statistic performance captured by the RC circuit PPC heatmaps (Figure 3.4C-D).

In summary, intelligently chosen summary statistics like `Peak` features can perform well, but leveraging information from the entire time series using `PCA30` produced consistently lower PRE and PPC values. The highly effective parameter recovery across the entire parameter grid for `PCA30` suggests that SBI permits a near unique mapping from dipole waveform to parameters when the summary statistic accounts for the full time series waveform and the parameters are kept in a subthreshold regime. In the next example, we show that this is not true in general. Even when using information from the full dipole waveform with PCA, inference in HNN simulations that include stochasticity can produce

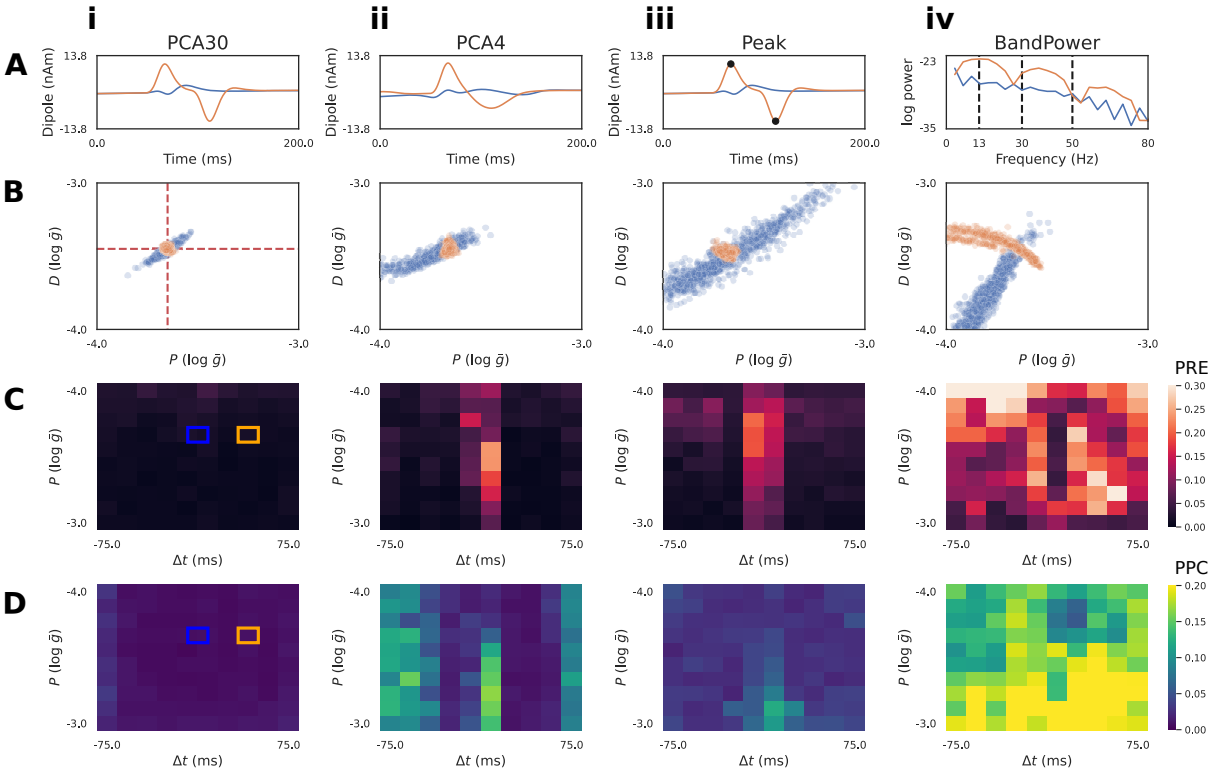

Figure 3.6: **SBI diagnostics of summary statistics in HNN.** The analysis shown in Figure 3.4 is repeated on a simplified HNN simulation for comparison. **A**: Summary statistics included `PCA30` (i), `PCA4` (ii), `Peak` (iii), and `BandPower` (iv). Two exemplar simulations with input latencies $\Delta t = 0$ (blue) and $\Delta t \neq 0$ (orange) are shown. **B**: We show the approximate posterior when conditioned on both the exemplar waveforms. The $\Delta t = 0$ time series produces a positive correlation between P and D for all summary statistics. **C**: Local parameter recovery error (PRE) is shown. Unlike the RC circuit, `PCA30` and `PCA4` permit better ground truth recovery even when Δt is near zero. In contrast, `Peak` features have poor parameter recovery similar to the RC example **D**: Local posterior predictive checks (PPC) are shown. `PCA30` and `PCA4` produce the values across the parameter grid.

substantial parameter indeterminacies.

SBI expands previously proposed mechanisms of subthreshold Beta Event simulations in HNN and shows stochastic simulation can lead to indeterminacies

Previous studies have applied the HNN modeling framework to propose novel mechanisms for the cellular and circuit level generation of transient low frequency oscillations in the 15-29 Hz beta-frequency band, referred to as Beta Events or beta bursts (Sherman et al. (2016); Law et al. (2022); Bonaiuto et al. (2021)). Many studies have shown that Beta

Events occur throughout the brain (e.g. see van Ede et al. (2018)) and their expression correlates with healthy and pathological sensory and motor processing (Shin et al. (2017); Bonaiuto et al. (2021); Wessel (2020)). Further, they often have a stereotypical waveform shape that resembles an inverted Ricker wavelet lasting ∼150 ms (Sherman et al. (2016); Bonaiuto et al. (2021); Brady and Bardouille (2022); Wessel (2020); Cole et al. (2017)) (see Figure 3.7). HNN provides potential mechanistic explanations for how the Beta Event waveform is generated and how changes in waveform shape may emerge. The HNN derived novel Beta Event mechanism put forth by Sherman et al. (2016) showed that Beta Events can arise from the dendritic integration of coincident bursts of subthreshold proximal and distal dendritic excitatory synaptic inputs to cortical pyramidal neurons (Sherman et al. (2016); Law et al. (2022)), such that the distal drive is effectively stronger and lasts one beta period (∼50 ms); a prediction that was supported by invasive laminar recordings in mice and monkeys (Sherman et al. (2016)) and high-resolution MEG in humans (Bonaiuto et al. (2021)). More specifically, HNN reproduced Beta Events with the observed stereotypical shape when stochastic trains of action potentials were simulated to drive the proximal and distal dendrites of the pyramidal neurons, nearly simultaneously. Bursts of input whose mean timing and standard deviation were chosen from Gaussian distributions activated excitatory synapses in a proximal and distal connection pattern, as shown in Figure 3.7A. The inverted Ricker waveform shape depended on the standard deviation of the proximal burst being broader than the distal burst, with the proximal burst occurring over ∼150 ms and the distal burst occurring over ∼50 ms, and the mean time of each burst being the same, i.e. reminiscent of $\Delta t = 0$ above. The proximal drive pushes current flow up the pyramidal neuron dendrites, while the distal drive pushes it back down (see Figure 3.7A).

Previous work also showed that, when the parameters of the proximal drive were held constant, the magnitude of the prominent middle trough changed with the variance of the distal drive, such that a parametric lowering of the variance pushed more current flow

down the dendrites generating an increased magnitude and sharper trough (Sherman et al. (2016)). This previous study led to the hypothesis that variance of the inputs impacted the waveform shape but was limited in that it did not perform an automated parameter inference on empirical data, nor did it investigate the additional role of proximal drive variance. Extending these prior results, here we hypothesized that both proximal and distal drive variances can impact the waveform shape and thus apply the SBI methods to HNN to infer distributions of proximal and distal drive variance in empirical MEG data from two example subjects whose average Beta Event trough magnitudes are distinct, collected in the prior study (see Methods). In this example, Subject 1 (Figure 3.7B(blue)) had larger magnitude peaks and troughs compared to subject 2 (Figure 3.7B(orange)). The same diagnostics used above are applied to assess the quality of the estimates.

We used the same parameters for Beta Event generation as in Sherman et al. (2016) and defined a prior distribution over proximal and distal input variance (P_{σ^2}, D_{σ^2}), see Table 4.1. All other parameters, including the number of spikes and mean input time, were held fixed as in Sherman et al. (2016). We used `PCA30` (variance explained=0.998) as the summary statistic motivated from the results above. We ran the HNN-SBI workflow to obtain a posterior distribution approximation that allows us to infer P_{σ^2} and D_{σ^2} for a given waveform (Figure 3.7B-C).

For both small and large magnitude Beta Events, there is a clear indeterminacy in D_{σ^2}. such that the distribution of D_{σ^2} is widely spread over the range of the prior distribution, indicating high uncertainty in the parameter estimates (Figure 3.7C). In contrast P_{σ^2} exhibits highly concentrated distributions, with the larger magnitude (blue) Beta Event corresponding to larger values ($P_{\sigma^2} \approx 30$ ms^2), and the smaller magnitude (orange) Beta Event corresponding to smaller values ($P_{\sigma^2} \approx 18$ ms^2). With a large proximal variance and smaller distal variance, the distal drive is effectively stronger and able to counteract the upward current flow to create a large downward deflection. With a small proximal variance, the stronger upward current flow is more equally matched with the downward current

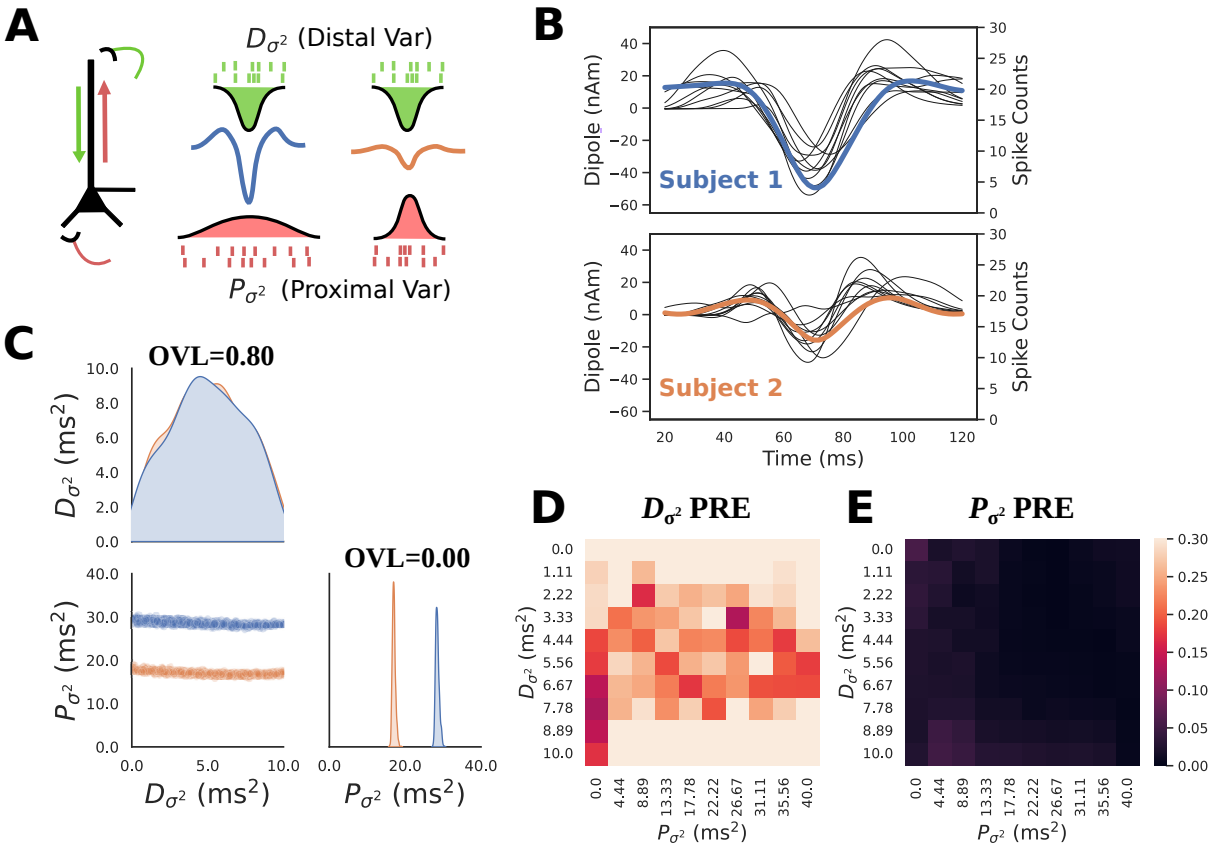

Figure 3.7: **HNN-SBI recovers circuit mechanism of Beta Event magnitude described in previous studies.** **A**: Schematic of Beta Event simulations in HNN. Beta Events are generated by a simultaneous burst of subthreshold proximal (red) and distal (green) excitatory inputs to L5 pyramidal neurons. **B**: Average source localized MEG Beta Event waveforms recorded from two subjects. Subject 1 (top, blue) exhibits a larger magnitude trough compared to subject 2 (bottom, orange). Simulations corresponding to a posterior predictive check (PPC) are shown in black, such that the parameters were sampled from the posterior (panel C) of each respective waveform. **C**: Posterior distributions conditioned on large (blue) and small (orange) magnitude Beta Events demonstrate that larger proximal variance produces a larger magnitude trough. Overlap coefficients (OVL) quantifying the separability of the marginal posterior distributions conditioned on each waveform are shown on the diagonal for the corresponding parameters. The marginal distributions for distal variance are highly overlapping, and non-overlapping for the proximal variance **D**: PRE heatmap of D_{σ^2} shows accurate parameter recovery (dark colors) when the ground truth parameters of D_{σ^2} are around 5 ms^2, and quickly worsen (light colors) as D_{σ^2} increases or decreases. **E**: PRE heatmap of P_{σ^2} shows accurate parameter recovery across the entire range of the prior distribution for both P_{σ^2} and D_{σ^2}.

flow to produce an overall smaller and noisier downward deflection. This proof-of-concept example based on two subjects is not meant to make strong scientific predictions on parameters controlling Beta Event shapes, but exemplifies how the HNN-SBI framework

can be applied to real data.

To quantify the separability of these distributions, we can employ the distribution overlap coefficient (OVL) which varies on a scale of [0,1] such that 1 indicates complete overlap, and 0 indicates no overlap. Unsurprisingly, the D_{σ^2} distributions for the large and small Beta Events produce an OVL of 0.80 due to the clearly visible high degree of overlap, whereas when comparing the D_{σ^2} distributions they exhibit almost no overlap with an OVL of approximately 0.00 (Figure 3.7C).

We note that unlike the simulations from the previous sections, where all parameters were deterministic, the exogenous drive times considered here are stochastic. As a result, there is not a 1:1 mapping from parameters to simulation output. To highlight this, Figure 3.7B shows simulations with parameters drawn from their respective posterior distributions (black traces). This visually represents the PPC diagnostic, where simulations from each posterior are close to the conditioning observation, but not a perfect match. In this setting, the stochasticity in the simulations make it so that measures such as PPC's are not guaranteed to be zero even with a perfect approximation of the ground truth posterior distribution. We additionally observe a pattern in the local PRE heatmap of D_{σ^2}, indicating that this parameter is accurately recovered from simulations generated with a D_{σ^2} of approximately 5 ms^2 (Figure 3.7D) and worsens for higher and lower values of D_{σ^2}. The PRE heatmap of P_{σ^2} shows all values close to zero demonstrating that the parameter is accurately recovered across the entire range of the prior distribution (Figure 3.7E).

In summary, the Beta Event example demonstrates how SBI can be used to estimate parameter distributions for given time series waveforms, and to compare potential mechanisms underlying different waveforms. For the example comparison shown, the variance of the proximal drive could be uniquely inferred while the variance of distal drive could not.

SBI reveals parameter interdependencies for suprathreshold Event Related Potential simulations in HNN

In the Beta Event example above, the effective strength of the proximal and distal input were maintained in a range where the activity of the cells remained subthreshold, which naturally limits the dynamic range of the simulation. Next, we consider a more complex example, in which the cells are driven to a suprathreshold spiking regime and show that this additional complexity can lead to parameter estimation indeterminacy that indicates a compensatory interaction between parameters.

The example considered describes simulations of a sensory evoked response or event related potential (ERP). HNN has been applied to study source localized ERPs in primary somatosensory cortex from tactile stimuli (e.g. Jones et al. (2007, 2009)) and in primary auditory cortex from auditory stimuli (e.g. Kohl et al. (2022)). In both cases, ERPs were simulated with a sequence of exogenous input that represented an evoked volley of drive to the local circuit that occurs after a sensory stimulus. The drive sequence consisted of a proximal drive representing the initial feedforward input from the thalamus, followed by a distal drive representing feedback input from higher order cortex, followed by a second proximal drive representing a loop of re-emergent thalamocortical drive (see schematic red and green arrows in Figure 3.8A). These drives were strong enough to generate spiking interactions in the local network and induced current flow up and down the pyramidal neuron dendrites to generate a current dipole ERP waveform analogous to those experimentally recorded (Figure 3.8A). Note, here we are not examining recorded data, but only example simulations. The specific timing of this exogenous drive sequence for example simulations is shown with arrows in Figure 3.8B. The parameters regulating the timing and the strength of these drives were fixed to the same values for the different conditions considered.

HNN has also been applied to infer neural mechanisms underlying differences in ERP waveform shapes recorded across different experimental conditions (e.g. for tactile evoked

responses in Jones et al. (2007)). Here, we are not trying to reproduce any empirical findings but rather apply SBI to an ERP simulation as in Jones et al. (2007) to examine the influence of changes in local network connectivity on the ERP waveform as a proof of concept example that examines a small subset of parameters distinct from our prior investigation.

We start by simulating example ERPs with different peak magnitudes as shown in Figure 3.8B as follows. ERPs were generated using a sequence of exogenous proximal-distal-proximal inputs (Figure 3.8A and as described above). The parameters representing the timing and strength of the sequence of exogenous proximal and distal input to the local circuit were chosen to be those distributed with the HNN software representing an example tactile evoked response simulation and fixed to those values (see Table 4.1). We then defined a prior distribution over parameters that define a small subset of the local network excitatory and inhibitory connectivity (Figure 3.2A). These parameters included the maximum conductance ($\bar{g}$) of layers 2 and 5 excitatory AMPA (E_{L2} / E_{L5}) and inhibitory $\mathrm{GABA_A}$ (I_{L2} / I_{L5}) connections to the L5 pyramidal cell. Specifically, E_{L2} and I_{L2} pertains to synapses on the distal dendrites, E_{L5} on proximal dendrites, and I_{L5} on the soma. Note that there exist more local network connections than those varied here as shown in Figure 3.2A and this chosen prior distribution was not based on any hypothesis or prior knowledge of the impact of local parameters on the ERP, but rather as a tractable example.

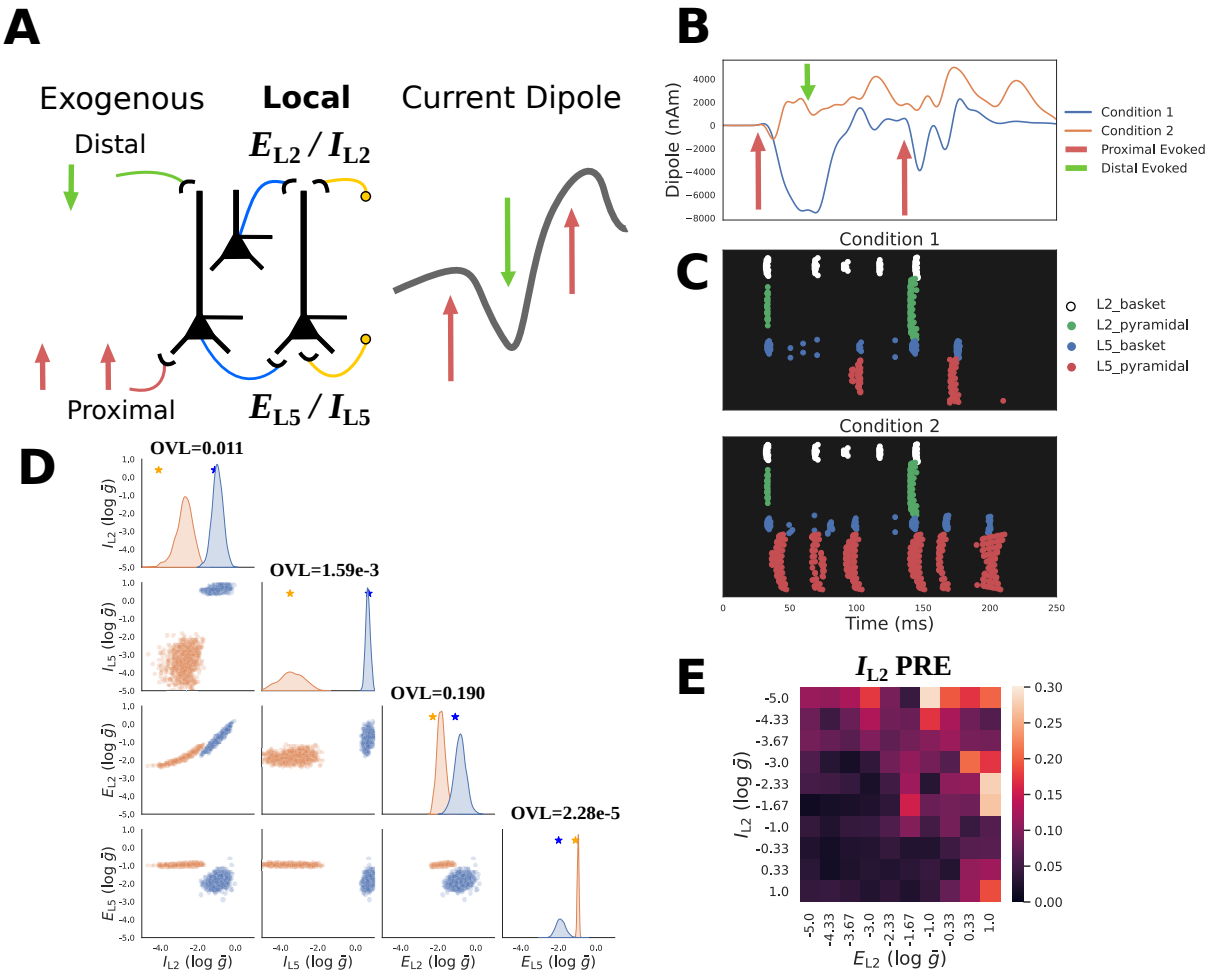

Figure 3.8: **Inferring local connectivity parameters from ERP waveforms. A**: Schematic of ERP simulations in HNN. Evoked activity is driven by a fixed sequence of proximal-distal-proximal exogenous inputs. SBI is used to infer the maximal conductance strength ($\bar{g}$) of local excitatory/inhibitory connections to the proximal/distal dendrites of L5 pyramidal neurons for example waveforms. **B**: Exemplar simulated ERPs (blue and orange solid lines) with differing local connectivity strengths chosen from a defined prior distribution (described in the text) are shown, along with the fixed timing of the sequence of exogenous inputs for each simulation (red and green arrows). **C**: Spike raster plots of cell specific firing for the two ERP simulation conditions from panel B. **D**: Posterior distributions over local connectivity parameters alongside ground truth parameters (stars on diagonal) for conditioning observations. A strong interaction between excitatory/inhibitory distal inputs (E_{L2} and I_{L2}) is visible in the lower square. Overlap coefficients (OVL) quantifying the separability of the marginal posterior distributions conditioned on each waveform are shown on the diagonal for the corresponding parameters. E_{L2} and I_{L2} exhibit a small amount of overlap with OVL values of 0.011 and 0.190 respectively. In contrast E_{L5} and I_{L5} were much more distinguishable, exhibiting OVL values of 2.28e-5 and 1.59e-13 respectively. **E**: Local parameter recovery error (PRE) for distal inhibition I_{L2} indicates errors are higher for observations generated with strong excitatory E_{L2} and weak inhibitory I_{L2} distal connections.

Two example ERPs produced by networks with different local connectivity are shown in Figure 3.8B. The ground truth parameters that created these waveforms are shown with stars in Figure 3.8D. Despite being activated by an identical exogenous input sequence, it is apparent that the local network connectivity differences lead to dramatically distinct current dipole ERP waveforms and corresponding spiking activities. In Figure 3.8C the spiking activity associated with each waveform is largely distinguished by the activity of L5 pyramidal neurons (red dots), with more firing in Condition 2 (orange waveform). For Condition 1 (blue), the first proximal input leads to the beginning of a sustained negative deflection in the current dipole, which persists during the subsequent distal input due to prolonged activation of the L5 basket cells which inhibit the L5 pyramidal neuron soma to pull current flow down the dendrites. Once this inhibition ends, the L5 pyramidal neuron is able to spike, pushing current flow back up the dendrites and the subsequent volley of proximal drive continues to push current flow up and down the pyramidal neuron dendrites due to a similar spiking dynamic. This is in contrast to Condition 2 (orange) in which L5 pyramidal neuron spiking starts almost immediately after the first proximal input and persists, pushing current flow up the dendrites to create a sustained positive deflection in the current dipole that persists through the entire simulation.

Figure 3.8D shows the results of applying the HNN-SBI framework with the `PCA30` (variance explained=0.999) summary statistic to estimate the ground truth parameters that generated the ERP waveforms described above. It is apparent that the posterior distributions conditioned on each waveform place high probability mass around the corresponding ground truth parameters (Figure 3.8D, stars on diagonal), but also exhibit strong indeterminacies. For example, for each condition, there is a clear interaction between E_{L2} and I_{L2}, such that as one parameter increases the other also increases, suggesting that these two parameters can compensate one another in a limited range to maintain a constant waveform. We can also observe that between the two conditions there are apparent differences in L5 pyramidal neuron spiking, as well as the sustained

negativity (blue) and positivity (orange) observed in the current dipole due to the complex dynamics that each parameter configuration creates.

To quantify the separability of these distributions, we calculated the OVL coefficient for the marginal distributions of all parameters (Figure 3.8D diagonal). Estimated parameter distributions of the synapses on the layer 5 distal dendrites, E_{L2} and I_{L2}, exhibit a small amount of overlap across conditions with OVL values of 0.190 and 0.011 respectively. The parameter distribution of the synapses on layer 5 somas however were much more distinguishable for the two conditions, exhibiting OVL values of 2.28e-5 for E_{L5}, and 1.59e-13 for I_{L5}. We additionally performed diagnostic analysis of the local PRE to see how recovery of I_{L2} changes as a function of I_{L2} itself, and E_{L2}. Figure 3.8E exhibits a clear pattern indicating that the recovery of I_{L2} is worse when the network exhibits low I_{L2} and large E_{L2}.

In summary, the ERP example provides another demonstration of how SBI can be used to estimate parameter distributions for given time series waveforms and to compare potential mechanisms underlying different waveforms. For the hypothetical example comparison shown, E_{L5} and I_{L5} were uniquely inferred with distributions for the waveforms compared exhibiting very little overlap. Distributions for E_{L2} and I_{L2} were also separable, albeit with slightly more overlap, and a marked interaction between these two parameters.

3.1.5 Discussion

Inference in detailed biophysical models like HNN is far from a solved problem. Nevertheless, the recent developments in likelihood-free inference techniques have enabled predictions of parameter distributions with a level of detail and complexity that was simply not feasible until now. In this study, we detailed step-by-step methods to employ SBI in detailed models of neural time series data and to assess the quality and uniqueness of the estimated parameter distributions. We began with a simplified RC-circuit example which exemplified the possibility of parameter interactions and highlighted limitations

with chosen summary statistics that do not consider the full time series waveform. We then demonstrated how distributions of biophysical parameters that can account for a given time series waveform can be inferred using an integrated HNN-SBI workflow applied to two common MEG/EEG signal motifs (Beta Events and ERPs) and how to assess overlap of the distributions from two different waveforms. This work does not aim to be an exhaustive guide to inference in HNN, nor to focus on specific neuroscientific questions, but instead to provide useful examples and methods to highlight critical decisions in the inference pipeline. There are several major takeaways from our study. First, highly non-linear biophysically detailed neural models like HNN are not suitable for Bayesian estimation methods that require access to a likelihood function (e.g. variational inference) or that approximate posterior distributions with independent Gaussians (a.k.a Laplace approximation). Rather they necessitate a method that can estimate complex posterior distributions and parameter interactions from a simulated dataset (e.g. SBI with masked autoregressive flows). Second, an important initial step in the SBI process is to identify a limited set of parameters and a range of values for those parameters that are assumed to be able to generate the waveform of interest and variation around it (i.e. the prior distribution). Due to the large-scale nature of biophysically detailed network models, it is not possible to perform SBI on all parameters at once. The choice of the prior distribution represents a hypothesis about parameters that are assumed to contribute to variation in the waveform. This hypothesis can be informed by domain knowledge of the question of interest. In the HNN examples shown, the hypothesized parameters of interest for estimation were the strength of the proximal and distal excitatory synaptic drive for the Beta Event simulation, and local excitatory and inhibitory connectivity for the ERP simulation; these parameters were chosen only for didactic purposes. All other parameters were fixed based on previous studies. Third, posterior diagnostics like PRE and PPC, are valuable tools to guide decisions in the inference pipeline, e.g. optimal summary statistic selection, and OVL can be used to assess the uniqueness of estimated distribution for two different waveforms. Fourth, when estimating parameters that account for time series

waveforms, summary statistics informed by the full time series such as PCA are the most effective at retaining essential information for mapping recorded signals to underlying parameters. While hand-crafted summary statistics, such as peak latency or magnitude, can permit an accurate mapping for certain waveform features, their selection may be insufficient to identify unique parameter distributions.

Comparison with inference in MEG/EEG neural modeling frameworks that rely on dynamic causal modeling

To our knowledge, while there are several modeling frameworks for simulating MEG/EEG signals, the other frameworks that use likelihood-based Bayesian inference to estimate parameters fall in the category of Dynamic Causal Modeling (DCM) (Kiebel et al. (2006); Daunizeau et al. (2011)). It is important to emphasize that while the HNN-SBI framework conceptually overlaps with DCM, they are two fundamentally distinct techniques which address different questions. At its base, DCM combines variational inference, a computationally efficient Bayesian inference algorithm, with neural mass models, as well as an observation model which translates simulated activity to experimental measures (e.g., MEG/EEG). Neural mass models refer to a specific class of neural models where a single variable represents the aggregate activity of large neural populations (e.g. population spike rates). The inferred parameters in the DCM framework most often represent the coupling strength between distinct neural masses (e.g. population nodes). By making simplifying anatomical and physiologic assumptions, neural mass models in the DCM framework can be employed in a large variety of inference problems due to their computational efficiency. However, their ability to make precise biophysical predictions on cellular and local circuit level processes is limited as the parameters are an abstraction representing population level activity (Daunizeau et al. (2011)), preventing a one-to-one comparison between model predictions and experimental measurements. Further, physiologically important effects like dendritic backpropagation are not represented in neural mass models unlike HNN. There are, however, advantages of DCM over the HNN-SBI framework that access

to a known likelihood function and other simplifying assumptions allow. For example, one critical question that the HNN-SBI framework is currently not suited to address is inference with multiple spatially separated cortical sources. While theoretically possible, the high computational demands of HNN-SBI severely limit the ability to explore multi-area interactions, and highlight the importance of using neural mass models and DCM in the analysis of whole-brain neuroimaging data. Recent work has shown that neural mass models can also be integrated with the SBI-framework for whole-brain studies (Hashemi et al. (2023)), highlighting the adaptability of the SBI methodology to a wide variety of neural models. Another major limitation of the current framework is that a small number of parameters must be chosen to create a sufficiently large dataset for amortized inference. Recent methodological developments in Bayesian model reduction with DCM have shown that large parameter spaces can be searched over in a computationally efficient manner (Friston et al. (2019)). Alternatively, previous works have attempted to merge the DCM and HNN modeling frameworks (Pinotsis et al. (2017); Pinotsis (2020)). This approach potentially enables one to benefit from the computationally efficient inference procedures offered by DCM, as well as the biophysical interpretability offered by HNN. Importantly, this approach necessitated 1) grouping of biophysical parameters in HNN to enable a one-to-one mapping to DCM parameters, and 2) an equivalent number of excitatory and inhibitory neurons in each cortical layer to permit a mean-field approximation and was applied to infer parameters in non-spiking HNN simulations of brain rhythms. For hypotheses where these assumptions apply and precise biophysical detail and network heterogeneity is not a focus the HNN-DCM approach offers a compelling alternative to HNN-SBI inference framework.

Comparison to other biophysically detailed neural modeling studies and estimation techniques

The simulation process outlined here extends prior work using SBI to estimate parameters in detailed neural models. Prior work applying SBI to neural models has

included a single compartment Hodgkin-Huxley model, and the stomatogastric ganglion model (Gonçalves et al. (2020)), both of which include an extensive parameter set, but contain significantly less detail and are smaller scale models than HNN. Additionally, non-amortized inference was performed in these models using sequential neural posterior estimation, allowing a much larger parameter set to be inferred, but only for a specific single observation. In contrast, we omit the use of sequential methods to perform inference on multiple observations using the same trained neural density estimator, but at the expense of the number of parameters that can be inferred simultaneously. Nevertheless, when only a few parameters are necessary for estimation, SBI offers a powerful tool for amortized inference in high-dimensional models, with examples ranging from whole-brain models in systems neuroscience (Lavanga et al. (2022); Hashemi et al. (2023)), to particle physics models investigating the formation of the Higgs boson (Brehmer et al. (2020)). The SBI workflow applied here used the neural density estimation technique known as masked autoregressive flows. There are currently a large number of neural density estimation techniques beyond this choice, each offering distinct advantages such as sample efficiency, expressivity, and likelihood evaluation (Cranmer et al. (2020)). While this field is rapidly evolving, recent concerns have been raised about the limits of such tools in the domain of Bayesian inference for scientific discovery (Hermans et al. (2021); Ward et al. (2022)). Unfortunately, there currently exist very few techniques for the validation of posterior distributions learned through neural density estimation beyond PPC and PRE diagnostics shown here. One promising work is simulation-based calibration (Talts et al. (2020)), which plays a similar role as PPC's by measuring properties the posterior approximation should satisfy if it is close to the ground truth. It is important to note, however, that this technique assesses the quality of the posterior approximation for the marginals of each parameter separately. More research in the domain of multidimensional calibration in the context of likelihood-free inference will be crucial to better represent complex parameter interactions like local E/I balance, as shown in Figure 3.8.

Nearly all parameter estimation techniques in high-dimensional biophysically detailed neural modeling will be computationally expensive. Indeed, SBI with HNN has a high computational load (for each example shown here, 100,000 simulations were run on a computing cluster in parallel over 512 CPU cores). While the upfront computational costs are high, there are advantages to SBI over other estimation techniques, for example COBYLA estimation, which has also been applied in HNN (Neymotin et al. (2020)). The main distinguishing factor is that SBI makes use of every simulation to build an accurate approximation of the posterior distribution for many waveforms. In contrast, COBYLA uses simulations to iteratively search for an optimized parameter set for a single waveform. Once trained, the neural density estimator in the SBI framework can be applied again on new time series waveforms (that fall within the prior distribution) without retraining. As shown in the results, the posterior distribution is an object with several utilities. We emphasize the mapping between observations and ground truth parameters in this paper, but there are alternative uses such as parameter optimization via non-amortized inference (Greenberg et al. (2019)), as well as building a more basic understanding of the model itself. Further, significant research efforts currently underway have the potential to decrease the computational cost of likelihood-free inference, making these techniques more accessible.

As an alternative to SBI, there has been substantial work in using probabilistic programming languages (PPL) to enable Bayesian inference on stochastic simulators. Models implemented in a PPL can be used in combination with modern inference algorithms (i.e. NUTS and ADVI) which have a dramatically lower computational cost compared to SBI (Hashemi et al. (2020, 2021); Vattikonda et al. (2021); Jha et al. (2022)). Unfortunately such approaches have the drawback that existing simulators must be completely rewritten in a PPL which may even compromise the efficiency of running forward simulations. Nevertheless, these techniques have proved highly useful for inference in large-scale neural simulators, and can potentially be combined with surrogate models that approximate the

forward simulations of existing models.

Other important future directions

In this study, we showed that PCA is the appropriate choice when compared to simple hand-crafted summary statistics when performing SBI on time series waveforms. However, PCA is constrained to preserve high-variance features, when in fact low-variance features may also be critical for identifying certain parameters. An important line of future work is the improvement of methods to learn summary statistics from neurophysiological signals that can help identify features of the signal that are essential for accurate parameter estimation. A promising development in this domain is the use of embedding networks that are trained to estimate summary statistics simultaneously with the neural density estimator used to approximate the posterior distribution that can account for those summary statistics (Dax et al. (2021a,b); Greenberg et al. (2019); Coelho Rodrigues and Gramfort (2020)). Currently, it is unclear if existing methods to train these embedding networks coupled to neural density estimators are sufficient and require further analysis.

Our HNN-SBI examples focused on making inferences by constraining only to one output of the model; namely simulated current dipole waveforms. However, due to the multi-scale nature of the HNN model there are many other model outputs that could help constrain the inference problem, such as cell spiking profiles, and/or local field potential signals. A major advantage of the Bayesian framework is the ability to flexibly integrate multiple features into the parameter estimation. If additional multiscale data is known, properties of this data can provide further summary statistics over which the inference problem can be constrained.

An important limitation of the current study is a characterization of how observation noise impacts posterior parameter inferences, and which summary statistics are more or less robust to such noise. In our study, `PCA30` was found to perform best for parameter recovery, however the information gained from such low variance features may be highly sensitive

to noise. Similarly, the hand-crafted summary statistics like `Peak` and `BandPower` may be particularly impacted by noise. The sensitivity of neural density estimators to observation noise, and more generally identifying model misspecification, is an open problem in the SBI field (Ward et al. (2022); Cranmer et al. (2020)) and an important direction for future research. Careful parameterization of observation noise, and establishment of predictive validity of the model using real data, will be critical next steps for robust inferences with the HNN-SBI framework.

Conclusion

Using detailed neural models in a Bayesian framework is the product of significant developments in machine learning, biophysical modeling, and high-performance computing that have evolved largely independently. Our results demonstrate that large-scale biophysically detailed models, like HNN, are now amenable to Bayesian methods via the SBI framework, an approach that has not been feasible in the past. However, this novel combination produces new conceptual and technical challenges that must be addressed to effectively use these techniques. While in this work we provide guidelines for addressing such challenges, more research in the domains of neural modeling and likelihood-free inference are needed. It is apparent that the combination of HNN with SBI is a step forward for making mechanistic inferences underlying MEG/EEG biomarkers, with the potential to provide novel circuit-level predictions on disease and neural function. Our results lay the foundation for similar integration of SBI into the growing number of biophysically detailed neural modeling frameworks to advance neuroscience discovery.

3.1.6 Supporting information

Prior distribution setup and sampling

Prior samples $\theta_i \in \mathbb{R}^d$ were generated by using the PyTorch Uniform distribution on the interval [0,1). The values in each dimension were then linearly mapped to the range of their

corresponding parameter values. For parameters specifying the maximum conductance $\bar{g}$ (nanosiemens, nS) of synaptic connections in HNN, the values were exponentiated in base 10 after being mapped to the appropriate range. This is to account for the saturating impact of conductance on model outputs, and is employed in previous SBI work with biophysical models (Gonçalves et al. (2020)).

Table 3.1: **Simulation parameters and SBI training**

RC Circuit	**Range**	**Transform**
Amplitude 1 (mA)	(0, 1)	linear
Amplitude 2 (mA)	(0, 1)	linear
Latency (ms)	(-75, 75)	linear
HNN "RC"	**Range**	**Transform**
Distal Exc (nS)	(1e-4, 1e-3)	exponential
Proximal Exc (nS)	(1e-4, 1e-3)	exponential
Latency (ms)	(-75, 75)	linear
Beta Events	**Range**	**Transform**
Distal Var (ms^2)	(0, 10)	linear
Proximal Var (ms^2)	(0, 40)	linear
ERPs	**Range**	**Transform**
Distal Exc (nS)	(1e-5, 1)	exponential
Proximal Exc (nS)	(1e-5, 1)	exponential
Distal Inh (nS)	(1e-5, 1)	exponential
Proximal Inh (nS)	(1e-5, 1)	exponential

The prior support and transform function for the parameters of examples shown in the text.

Simulation and SBI training

Prior samples and simulations were all generated and stored in the form of `NumPy` binary arrays before neural density estimator training. The `SBI` Python package was used for all neural density estimator training and posterior evaluation. A masked autoregressive flow architecture was utilized for approximation of the posterior distribution. Posteriors for all examples were trained using a dataset of 100,000 samples from the prior distribution. Gaussian white noise was added to training observations x_i (Rothfuss et al. (2020)). The variance of the Gaussian noise added to observations was 0.01 for RC circuit simulations, and 1e-5 for HNN simulations.

All analysis was performed on the Expanse supercomputing cluster managed by XSEDE and the Neuroscience Gateway. HNN simulations were generated using the Dask distributed scheduler configured for the SLURM workload manager. When distributed across 256 CPU cores, the simulated dataset for each example generally took <8 hours to generate. The neural density estimator took <20 minutes to converge when trained on the CPU of a single computing cluster node with 32 cores.

Diagnostic heatmaps were constructed by defining a grid over the support of the prior with a range of $[0.05, 0.95]^d$, with a resolution of 10 samples in each dimension d.

Acknowledgments

Simulations were implemented using HNN-core, a command line interface to the HNN model: https://github.com/jonescompneurolab/hnn-core

Experiments were made possible thanks to the Python scientific ecosystem: Python (Van Rossum and Drake (2009)), SBI (Tejero-Cantero et al. (2020)), PyTorch (Paszke et al. (2019)), NumPy (Harris et al. (2020)), SciPy (Virtanen et al. (2020)), Matplotlib (Hunter (2007)), Seaborn (Waskom (2021)), and Dask (Dask Development Team (2016)).

This work was supported by the Chateaubriand Fellowship, computing resources offered by the Neuroscience Gateway (NSG), l'Agence nationale de la recherche (ANR; grant number ANR-20-CHIA-0016 awarded to A.G.), the National Institute of Health (NIH; grant numbers R01AG076227 and RF1MH130415 awarded to S.J., and grant number U24EB029005 awarded to the NSG team), and the National Science Foundation (NSF; grant numbers 1935749 and 1935771 awarded to the NSG team). S.J. received salary support from NIH grant numbers R01AG076227, RF1MH130415, and U24EB029005. A.G. received salary support from ANR grant number ANR-20-CHIA-0016.

The funders had no role in study design, data collection and analysis, decision to

publish, or preparation of the manuscript.

We would like to thank Ryan Thorpe for feedback on the manuscript, and Blake Caldwell for technical guidance early on in this study.

3.1.7 Data and code availability

All code for analysis and figure generation is available at https://github.com/ntolley/hnn_sbi_examples. Data produced from simulation generating code, and experimentally recorded MEG Beta Event data from Sherman et al. (2016); Jones et al. (2009), is available on the Open Science Framework at https://doi.org/10.17605/OSF.IO/VZ97X.

Results of diagnostic analyses of the RC circuit and simplified HNN examples for all parameters can be found in the files `hnn_rc_diagnostics.csv` and `rc_circuit_diagnostics.csv` in the same repository.

3.2 Ultrafast connectivity optimization of large-scale biophysical network models with deep learning

Contributing Authors

Nicholas Tolley[1] and Stephanie R. Jones[1].

3.2.1 Summary

Understanding the relationship between network connectivity and emergent neural dynamics is a fundamental and unsolved problem in neuroscience. Detailed biophysical models can simulate highly realistic neural circuits with biologically interpretable parameters, providing a powerful way to study neural dynamics. However, their use is challenged

[1]Department of Neuroscience, Brown University, Providence, Rhode Island, United States of America

by an overwhelming number of model parameters, computationally expensive simulations, and complex mappings from model parameters to simulation outputs. Previous work has demonstrated that deep neural networks (surrogate models) can be trained to approximate compartmental neuron models, offering simulation speeds that are orders of magnitude faster. Here, we extend this work by implementing surrogate models of individual cells (Figure 3.9B-E) and using a spiking neural network (SNN) architecture to connect them into a biophysical network model (Figure 3.10). This construction permits efficient gradient-based optimization of cell-cell connectivity parameters (Figure 3.11), and the ability to optimize to complex neural activity patterns. We demonstrate the effectiveness of this approach by using surrogate models to approximate a detailed model of the neocortex, the Human Neocortical Neurosolver (HNN) (Figure 3.9A). As a proof-of-concept, we infer the strength of connectivity among neurons that gives rise to 15-60 Hz oscillations from noisy background drive (Figure 3.11A-C), with corresponding predictions on cell spiking activity (Figure 3.11C-F).

These methods open detailed biophysical modeling to questions that have been previously restricted to more abstract mathematically-tractable models with limited biological interpretability. Important future applications include the study of multi-area network models and cortical traveling waves.

3.2.2 Training deep neural networks to be surrogate models for biophysical neurons

We chose HNN (Neymotin et al. (2020)) as the biophysical network model to be approximated by surrogate models (Figure 3.9A). HNN is a large-scale detailed model of a cortical column designed to simulate electrical currents (i.e., current dipoles) underlying M/EEG signals with interpretability at the cell and circuit level. The 4 principle cell types in the model are L2/3 and L5 excitatory pyramidal (e) and inhibitory (i) neurons.

A TCN-LSTM (temporal convolutional network - long short-term memory network)

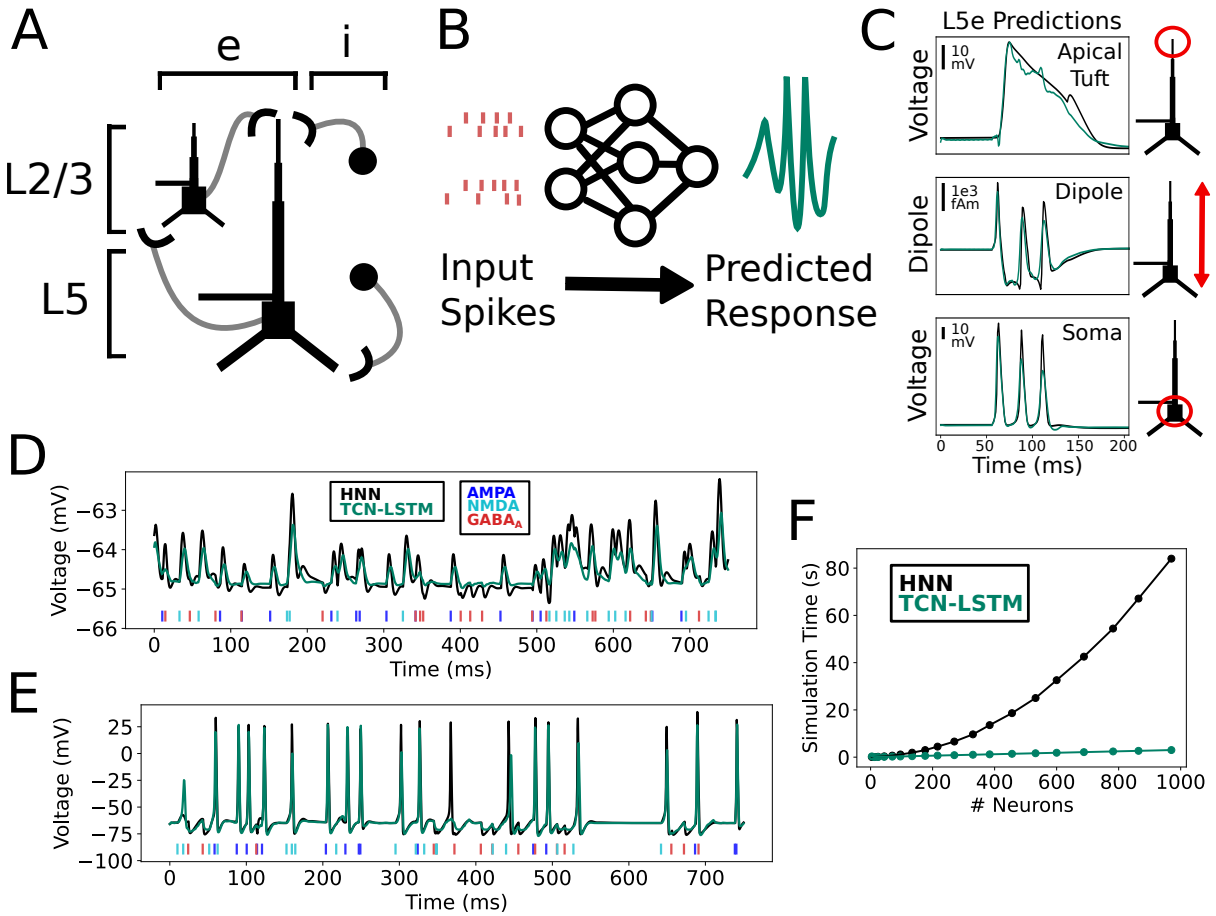

Figure 3.9: Surrogate model accurately predicts compartmental voltage responses and net current dipole of individual neurons in HNN. **A**: The HNN cortical column model is composed of 4 distinct cell types: L2/3 and L5 excitatory pyramidal (e) and inhibitory (i) neurons. **B**: The surrogate model is trained to take synapse activation times as input, and outputs compartmental voltages and current dipole signals. **C**: Predictions of the surrogate model (**teal**) compared to L5e neuron simulations (**black**). Predictions include membrane potentials of the apical tuft (top panel) and soma (bottom panel), as well as intracellular axial current flow which underlies EEG measured current dipoles (bottom). **D**: Detailed view of somatic voltage during subthreshold synaptic inputs for HNN and surrogate model (TCN-LSTM). Colored ticks at the bottom correspond to AMPA (dark blue), NMDA (light blue), and GABAa (red) inputs. **E**: Same as panel D for suprathreshold synaptic inputs. **F** Comparison of parallelized simulation time for multiple L5e neurons. Surrogate model (TCN-LSTM) simulations are orders of magnitude faster than HNN simulations for large numbers of neurons (>500).

implemented in PyTorch was trained to predict the response of individual neurons to a train of input spikes based on the methods of Oláh et al. (2022) and Verhellen et al. (2024) (Figure 3.9B). The training and validation sets were produced by activating all excitatory

AMPA synapses with random spike trains of 10 Hz Poisson noise (independently across compartments). The final trained surrogate model accurately predicted compartmental voltages and current dipoles (Figure 3.9C). For L5e, the correlation between the HNN simulation and surrogate model predictions on a held-out validation set was 0.843 and 0.861 for the membrane potentials at the soma and apical tuft, and 0.772 for the current dipole.

A higher resolution view of validation set simulations is shown in Figure 3.9D for subthreshold synaptic inputs, and Figure 3.9E for suprathreshold synaptic inputs. In these simulations, synaptic input is only provided to the soma to clearly visualize the response to activation of different synaptic receptors, namely AMPA (dark blue), NMDA (light blue), and GABAa (red). The surrogate model accurately approximates the subthreshold response to synaptic inputs. For example, the surrogate model closely matches the depolarization produced by temporal summation of the repeated excitatory inputs visible at 500 ms (Figure 3.9D). The surrogate model is additionally able to accurately predict the timing and shape of action potentials produced by suprathreshold synaptic inputs (Figure 3.9E), with the only discrepancy occurring at approximately 400 ms where a spike occurs in the HNN simulation (**black**) that is not present for the surrogate model (**teal**).

A major benefit of using surrogate models is the computational efficiency of simulating many neurons in parallel. Figure 3.9F plots the simulation time of HNN (**black**) and surrogate model (**teal**) L5e neurons with respect to the number of neurons simulated in parallel. At low numbers of neurons (<100), HNN and surrogate models simulation times are roughly comparable. However, as the number of neurons increase HNN simulations exhibit a supralinear increase in simulation time, with respect to a nearly linear increase for surrogate models. For 1000 neurons, the surrogate model simulation time is approximately 1 second, compared to 80 seconds for the HNN model. These results suggest that surrogate models may be a good choice for approximating the simulation of large networks of biophysical neurons. The results shown here were produced with a single GPU, suggesting

that surrogate models can enable investigations with large-scale biophysical network models even when access to computational resources are limited.

3.2.3 Connecting single-cell surrogate models with a deep learning based SNN architecture

The single-cell surrogate models were then connected in a cortical microcircuit based on the cell-cell connectivity structure defined by HNN, with 50 excitatory and 16 inhibitory cells in each layer. Figure 3.10 directly compares simulated current dipoles and spiking responses between HNN and the surrogate model with the default local synaptic connectivity of the HNN model (Figure 3.10A-B), and a modified connectivity where the L5i $\rightarrow$ L5e synaptic conductance was increased by a factor of 10 (Figure 3.10C-D). Each network was activated with a 10 Hz poisson drive delivered to AMPA receptors at the proximal and distal dendrites of the model (Neymotin et al. (2020)). Simulations for the default model connectivity were largely similar with large synchronous bursts of L5e neurons occurring approximately 200 ms apart (Figure 3.10A-B bottom panel), and a corresponding large negative deflection in the current dipole (Figure 3.10A-B top panel). The surrogate model does exhibit slight discrepancies, specifically with narrower peaks in the current dipole compared to HNN simulations, and substantially more synchronized firing of L5e neurons (Figure 3.10A-B).

In the simulations with increased L5i $\rightarrow$ L5e connection strength, L5e neuron spiking is completely eliminated in both HNN and the surrogate model (Figure 3.10C-D bottom panel). As a result, the current dipole response is observed to be an order of magnitude smaller than the default network (Figure 3.10C-D top panel, note scale bar of 1 nAm). In summary, connecting single-cell surrogate models into a cortical microcircuit that matches the HNN model provides a good approximation of network-level simulations. There are slight discrepancies between the HNN model simulations and the surrogate network model, which would likely be reduced with increased training of the single-cell surrogate models.

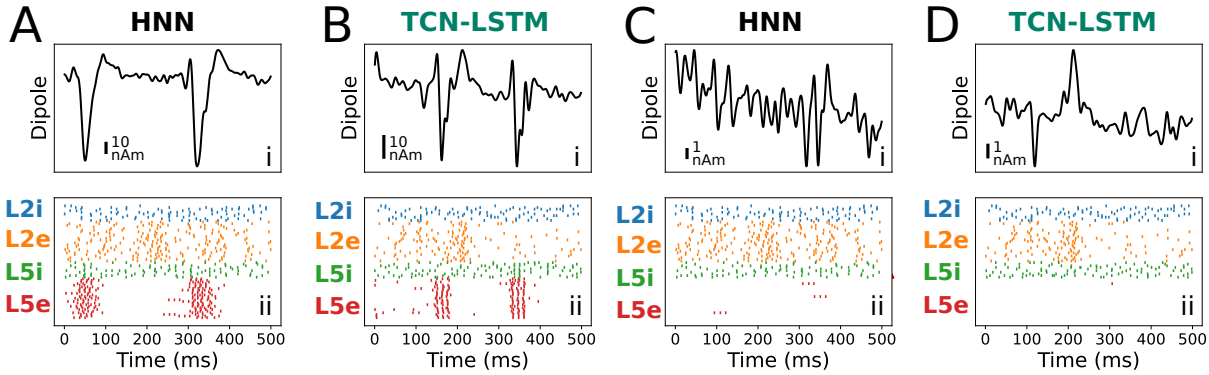

Figure 3.10: Network surrogate model approximates spiking and dipole response. **A-B**. The default synaptic connectivity of the HNN model was simulated with noisy background inputs (panel A) and compared to a trained surrogate network model with the same connectivity (panel B). The dipole response (top) and spike response (bottom) are largely similar, with the surrogate model exhibiting narrower peaks in the dipole (panel B top) and more synchronous spiking (panel B bottom). **C-D**: Same as panels A-B except the local synaptic connectivity was modified such that the L5i $\rightarrow$ L5e synapse conductance was increased by a factor of 10.

3.2.4 Inferring excitatory-inhibitory connectivity that produces neural oscillations

A potential benefit of surrogate modeling is the ability to use gradient-based optimizers to tune the connectivity structure (Adam implemented in PyTorch used below). A naive implementation of a surrogate network model cannot be optimized because thresholding functions used to determine spiking are not differentiable. Therefore, we adopted techniques from SNN methods and equipped our spike threshold function with an approximate gradient (the fast sigmoid function) which is critical for performing backpropagation through the surrogate network model. The major benefit of this approach is the ability to define complex neural activity patterns as the optimization objective, enabling rapid identification of connectivity parameters that produce desired outputs.

We tested the network's ability to optimize connectivity parameters by tasking it to maximize spontaneous high frequency oscillations (15-60 Hz) in a network driven by 10 Hz Poisson noise (Figure 3.11A). The loss function was defined as $L = -\log(\int_{15}^{60} S_X(\omega)\, d\omega)$ and effectively maximizes 15-60 Hz power in the dipole signal

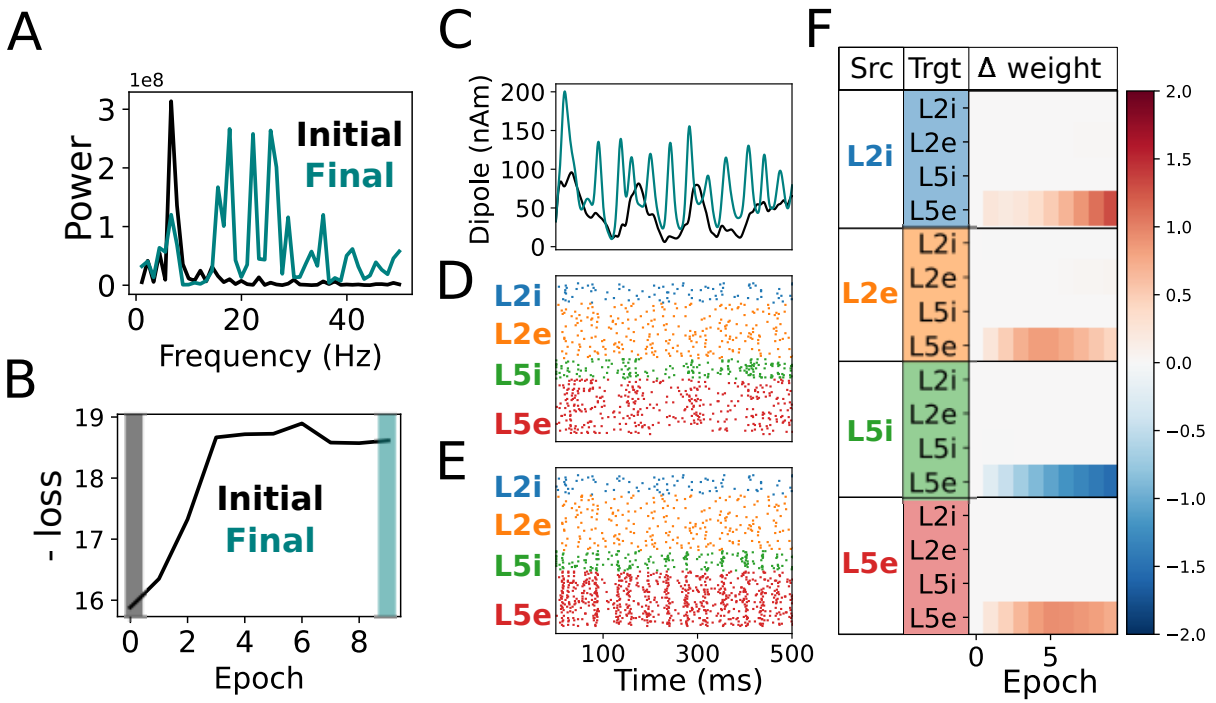

Figure 3.11: **A**: Power spectral density of dipole produced by initial (**black**) and final (**teal**) connectivity parameters. **B**: The optimizer efficiently maximized high frequency band power. **C**: Simulated dipole shows high frequency oscillations (**teal**) with final connectivity parameters. **D**: Spike raster of initial parameters. **E**: Spike raster of final parameters. **F**: Optimization of cell-cell connection weights on each epoch.

$X(t)$. The optimizer rapidly increased high frequency power from an initial connectivity configuration that produced low frequency activity in the alpha (8-12 Hz) range to a final configuration that produced substantially more activity in the beta (15-30 Hz) range after 10 epochs (Figure 3.11A-C). Spike rasters of the initial (Figure 3.11D) and final (Figure 3.11E) configurations show that both oscillations are due to synchronous firing of L5e cells. Inspection of connectivity modifications (Figure 3.11F) reveals changes were made exclusively to L5e targeting projections, reflecting L5 pyramidal cells as the main contributor to M/EEG dipoles. L2i→L5e connections were strengthened, whereas L5i→L5e connections were weakened, suggesting that L2 and L5 inhibitory neurons play opposing roles in determining oscillation frequency in this proof-of-concept example.

References

Abbott, L. F. and Dayan, P. (2001). *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems*. Computational Neuroscience Series. MIT Press, Cambridge, MA, USA.

Barbas, H., García-Cabezas, M. , and Zikopoulos, B. (2013). Frontal-thalamic circuits associated with language. *Brain and Language*, 126(1):49–61.

Billeh, Y. N., Cai, B., Gratiy, S. L., Dai, K., Iyer, R., Gouwens, N. W., Abbasi-Asl, R., Jia, X., Siegle, J. H., Olsen, S. R., Koch, C., Mihalas, S., and Arkhipov, A. (2020). Systematic Integration of Structural and Functional Data into Multi-scale Models of Mouse Primary Visual Cortex. *Neuron*.

Bishop, C. M. and Nasrabadi, N. M. (2006). *Pattern recognition and machine learning*, volume 4. Springer.

Bonaiuto, J. J., Little, S., Neymotin, S. A., Jones, S. R., Barnes, G. R., and Bestmann, S. (2021). Laminar dynamics of high amplitude beta bursts in human motor cortex. *NeuroImage*, 242:118479.

Borges, F. S., Moreira, J. V. S., Takarabe, L. M., Lytton, W. W., and Dura-Bernal, S. (2022). Large-scale biophysically detailed model of somatosensory thalamocortical circuits in NetPyNE. *Frontiers in Neuroinformatics*, 16.

Box, G. E. P. (1980). Sampling and Bayes' Inference in Scientific Modelling and Robustness. *Journal of the Royal Statistical Society. Series A (General)*, 143(4):383–430. Publisher: [Royal Statistical Society, Wiley].

Brady, B. and Bardouille, T. (2022). Periodic/Aperiodic parameterization of transient oscillations (PAPTO)–Implications for healthy ageing. *NeuroImage*, 251:118974.

Brehmer, J., Louppe, G., Pavez, J., and Cranmer, K. (2020). Mining gold from implicit models to improve likelihood-free inference. *Proceedings of the National Academy*

of Sciences, 117(10):5242–5249. Publisher: Proceedings of the National Academy of Sciences.

Buzsáki, G. and Draguhn, A. (2004). Neuronal Oscillations in Cortical Networks. *Science*, 304(5679):1926–1929.

Coelho Rodrigues, P. L. and Gramfort, A. (2020). Learning summary features of time series for likelihood free inference. In *Workshop "Machine Learning and the Physical Sciences" at the NeurIPS 2020 - 34th Conference on Neural Information Processing Systems*, Vancouver, Canada.

Cole, S. R., Meij, R. v. d., Peterson, E. J., Hemptinne, C. d., Starr, P. A., and Voytek, B. (2017). Nonsinusoidal Beta Oscillations Reflect Cortical Pathophysiology in Parkinson's Disease. *Journal of Neuroscience*, 37(18):4830–4840. Publisher: Society for Neuroscience Section: Research Articles.

Cranmer, K., Brehmer, J., and Louppe, G. (2020). The frontier of simulation-based inference. *Proceedings of the National Academy of Sciences*, 117(48):30055–30062. Publisher: National Academy of Sciences Section: Colloquium Paper.

Dai, K., Gratiy, S. L., Billeh, Y. N., Xu, R., Cai, B., Cain, N., Rimehaug, A. E., Stasik, A. J., Einevoll, G. T., Mihalas, S., Koch, C., and Arkhipov, A. (2020). Brain Modeling ToolKit: An open source software suite for multiscale modeling of brain circuits. *PLOS Computational Biology*, 16(11):e1008386. Publisher: Public Library of Science.

Dask Development Team (2016). Dask: Library for dynamic task scheduling.

Daunizeau, J., David, O., and Stephan, K. E. (2011). Dynamic causal modelling: A critical review of the biophysical and statistical foundations. *NeuroImage*, 58(2):312–322.

Dax, M., Green, S. R., Gair, J., Deistler, M., Schölkopf, B., and Macke, J. H. (2021a). Group equivariant neural posterior estimation. arXiv:2111.13139 [astro-ph, physics:gr-qc, stat].

Dax, M., Green, S. R., Gair, J., Macke, J. H., Buonanno, A., and Schölkopf, B. (2021b). Real-Time Gravitational Wave Science with Neural Posterior Estimation. *Physical Review Letters*, 127(24):241103. Publisher: American Physical Society.

Druckmann, S., Banitt, Y., Gidon, A., Schürmann, F., Markram, H., and Segev, I. (2007). A novel multiple objective optimization framework for constraining conductance-based neuron models by experimental data. *Frontiers in Neuroscience*, 1.

Dura-Bernal, S., Griffith, E. Y., Barczak, A., O'Connell, M. N., McGinnis, T., Schroeder, C. E., Lytton, W. W., Lakatos, P., and Neymotin, S. A. (2022a). Data-driven multiscale model of macaque auditory thalamocortical circuits reproduces in vivo dynamics. Pages: 2022.02.03.479036 Section: New Results.

Dura-Bernal, S., Neymotin, S. A., Suter, B. A., Dacre, J., Schiemann, J., Duguid, I., Shepherd, G. M., and Lytton, W. W. (2022b). Multiscale model of primary motor cortex circuits predicts in vivo cell type-specific, behavioral state-dependent dynamics. Pages: 2022.02.03.479040 Section: New Results.

Einevoll, G. T., Destexhe, A., Diesmann, M., Grün, S., Jirsa, V., Kamps, M. d., Migliore, M., Ness, T. V., Plesser, H. E., and Schürmann, F. (2019). The Scientific Case for Brain Simulations. *Neuron*, 102(4):735–744. Publisher: Elsevier.

Ermentrout, G. B. and Terman, D. H. (2010). Dendrites. In Ermentrout, G. B. and Terman, D. H., editors, *Mathematical Foundations of Neuroscience*, Interdisciplinary Applied Mathematics, pages 29–48. Springer, New York, NY.

Franks, J. J. (2020). Handbook of Approximate Bayesian Computation. *Journal of the American Statistical Association*, 115(532):2100–2101. Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/01621459.2020.1846973.

Friston, K., Parr, T., and Zeidman, P. (2019). Bayesian model reduction. arXiv:1805.07092 [stat].

Gelman, A., Meng, X.-L., and Stern, H. (1996). Posterior predictive assessment of model fitness via realized discrepancies. *Statistica Sinica*, 6(4):733–760. Publisher: Institute of Statistical Science, Academia Sinica.

Gleeson, P., Cantarelli, M., Marin, B., Quintana, A., Earnshaw, M., Sadeh, S., Piasini, E., Birgiolas, J., Cannon, R. C., Cayco-Gajic, N. A., Crook, S., Davison, A. P., Dura-Bernal, S., Ecker, A., Hines, M. L., Idili, G., Lanore, F., Larson, S. D., Lytton, W. W., Majumdar, A., McDougal, R. A., Sivagnanam, S., Solinas, S., Stanislovas, R., van Albada, S. J., van Geit, W., and Silver, R. A. (2019). Open Source Brain: A Collaborative Resource for Visualizing, Analyzing, Simulating, and Developing Standardized Models of Neurons and Circuits. *Neuron*, 103(3):395–411.e5.

Gonçalves, P. J., Lueckmann, J.-M., Deistler, M., Nonnenmacher, M., Öcal, K., Bassetto, G., Chintaluri, C., Podlaski, W. F., Haddad, S. A., Vogels, T. P., Greenberg, D. S., and Macke, J. H. (2020). Training deep neural density estimators to identify mechanistic models of neural dynamics. *eLife*, 9:e56261. Publisher: eLife Sciences Publications, Ltd.

Greenberg, D., Nonnenmacher, M., and Macke, J. (2019). Automatic Posterior Transformation for Likelihood-Free Inference. In *Proceedings of the 36th International Conference on Machine Learning*, pages 2404–2414. PMLR. ISSN: 2640-3498.

Hagen, E., Næss, S., Ness, T. V., and Einevoll, G. T. (2018). Multimodal modeling of neural network activity: computing LFP, ECoG, EEG and MEG signals with LFPy2.0. *bioRxiv*, page 281717.

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del Río, J. F., Wiebe, M., Peterson, P., Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C., and Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, 585(7825):357–362. Number: 7825 Publisher: Nature Publishing Group.

Hashemi, M., Vattikonda, A. N., Jha, J., Sip, V., Woodman, M. M., Bartolomei, F., and Jirsa, V. K. (2023). Amortized Bayesian inference on generative dynamical network models of epilepsy using deep neural density estimators. *Neural Networks*, 163:178–194.

Hashemi, M., Vattikonda, A. N., Sip, V., Diaz-Pier, S., Peyser, A., Wang, H., Guye, M., Bartolomei, F., Woodman, M. M., and Jirsa, V. K. (2021). On the influence of prior information evaluated by fully Bayesian criteria in a personalized whole-brain model of epilepsy spread. *PLOS Computational Biology*, 17(7):e1009129. Publisher: Public Library of Science.

Hashemi, M., Vattikonda, A. N., Sip, V., Guye, M., Bartolomei, F., Woodman, M. M., and Jirsa, V. K. (2020). The Bayesian Virtual Epileptic Patient: A probabilistic framework designed to infer the spatial map of epileptogenicity in a personalized large-scale brain model of epilepsy spread. *NeuroImage*, 217:116839.

Hay, E., Hill, S., Schürmann, F., Markram, H., and Segev, I. (2011). Models of Neocortical Layer 5b Pyramidal Cells Capturing a Wide Range of Dendritic and Perisomatic Active Properties. *PLOS Computational Biology*, 7(7):e1002107. Publisher: Public Library of Science.

Hermans, J., Delaunoy, A., Rozet, F., Wehenkel, A., and Louppe, G. (2021). Averting A Crisis In Simulation-Based Inference. Technical Report arXiv:2110.06581, arXiv. arXiv:2110.06581 [cs, stat] type: article.

Hunter, J. D. (2007). Matplotlib: A 2D Graphics Environment. *Computing in Science & Engineering*, 9(3):90–95. Conference Name: Computing in Science & Engineering.

Hämäläinen, M., Hari, R., Ilmoniemi, R. J., Knuutila, J., and Lounasmaa, O. V. (1993). Magnetoencephalography—theory, instrumentation, and applications to noninvasive studies of the working human brain. *Reviews of Modern Physics*, 65(2):413–497. Publisher: American Physical Society.

Jha, J., Hashemi, M., Vattikonda, A. N., Wang, H., and Jirsa, V. (2022). Fully Bayesian

estimation of virtual brain parameters with self-tuning Hamiltonian Monte Carlo. *Machine Learning: Science and Technology*, 3(3):035016. Publisher: IOP Publishing.

Jones, E. G. (2001). The thalamic matrix and thalamocortical synchrony. *Trends in Neurosciences*, 24(10):595–601.

Jones, S. R., Pritchett, D. L., Sikora, M. A., Stufflebeam, S. M., Hämäläinen, M., and Moore, C. I. (2009). Quantitative Analysis and Biophysically Realistic Neural Modeling of the MEG Mu Rhythm: Rhythmogenesis and Modulation of Sensory-Evoked Responses. *Journal of Neurophysiology*, 102(6):3554–3572. Publisher: American Physiological Society.

Jones, S. R., Pritchett, D. L., Stufflebeam, S. M., Hämäläinen, M., and Moore, C. I. (2007). Neural Correlates of Tactile Detection: A Combined Magnetoencephalography and Biophysically Based Computational Modeling Study. *Journal of Neuroscience*, 27(40):10751–10764.

Kiebel, S. J., David, O., and Friston, K. J. (2006). Dynamic causal modelling of evoked responses in EEG/MEG with lead field parameterization. *NeuroImage*, 30(4):1273–1284.

Kiebel, S. J., Garrido, M. I., and Friston, K. J. (2007). Dynamic causal modelling of evoked responses: The role of intrinsic connections. *NeuroImage*, 36(2):332–345.

Kohl, C., Parviainen, T., and Jones, S. R. (2022). Neural Mechanisms Underlying Human Auditory Evoked Responses Revealed By Human Neocortical Neurosolver. *Brain Topography*, 35(1):19–35.

Lavanga, M., Stumme, J., Yalcinkaya, B. H., Fousek, J., Jockwitz, C., Sheheitli, H., Bittner, N., Hashemi, M., Petkoski, S., Caspers, S., and Jirsa, V. (2022). The virtual aging brain: a model-driven explanation for cognitive decline in older subjects. Pages: 2022.02.17.480902 Section: New Results.

Law, R. G., Pugliese, S., Shin, H., Sliva, D. D., Lee, S., Neymotin, S., Moore, C., and

Jones, S. R. (2022). Thalamocortical Mechanisms Regulating the Relationship between Transient Beta Events and Human Tactile Perception. *Cerebral Cortex*, 32(4):668–688.

Lueckmann, J.-M., Gonçalves, P. J., Bassetto, G., Öcal, K., Nonnenmacher, M., and Macke, J. H. (2017). Flexible statistical inference for mechanistic models of neural dynamics. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pages 1289–1299, Red Hook, NY, USA. Curran Associates Inc.

Marder, E. and Goaillard, J.-M. (2006). Variability, compensation and homeostasis in neuron and network function. *Nature Reviews Neuroscience*, 7(7):563–574. Bandiera_abtest: a Cg_type: Nature Research Journals Number: 7 Primary_atype: Reviews Publisher: Nature Publishing Group.

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

McDougal, R. A., Morse, T. M., Carnevale, T., Marenco, L., Wang, R., Migliore, M.,

Miller, P. L., Shepherd, G. M., and Hines, M. L. (2017). Twenty years of ModelDB and beyond: building essential modeling tools for the future of neuroscience. *Journal of Computational Neuroscience*, 42(1):1–10.

Neymotin, S. A., Daniels, D. S., Caldwell, B., McDougal, R. A., Carnevale, N. T., Jas, M., Moore, C. I., Hines, M. L., Hämäläinen, M., and Jones, S. R. (2020). Human Neocortical Neurosolver (HNN), a new software tool for interpreting the cellular and network origin of human MEG/EEG data. *eLife*, 9:e51214. Publisher: eLife Sciences Publications, Ltd.

Oláh, V. J., Pedersen, N. P., and Rowan, M. J. (2022). Ultrafast simulation of large-scale neocortical microcircuitry with biophysically realistic neurons. *eLife*, 11:e79535. Publisher: eLife Sciences Publications, Ltd.

Pastore, M. and Calcagnì, A. (2019). Measuring Distribution Similarities Between Samples: A Distribution-Free Overlapping Index. *Frontiers in Psychology*, 10.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. (2019). PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Pinotsis, D. A. (2020). Statistical decision theory and multiscale analyses of human brain data. *Journal of Neuroscience Methods*, 346:108912.

Pinotsis, D. A., Geerts, J. P., Pinto, L., FitzGerald, T. H. B., Litvak, V., Auksztulewicz, R., and Friston, K. J. (2017). Linking canonical microcircuits and neuronal activity: Dynamic causal modelling of laminar recordings. *NeuroImage*, 146:355–366.

Powell, M. J. D. (1964). An efficient method for finding the minimum of a function of several variables without calculating derivatives. *The Computer Journal*, 7(2):155–162.

Rothfuss, J., Ferreira, F., Boehm, S., Walther, S., Ulrich, M., Asfour, T., and Krause, A. (2020). Noise Regularization for Conditional Density Estimation. Technical Report arXiv:1907.08982, arXiv. arXiv:1907.08982 [cs, stat] type: article.

Sherman, M. A., Lee, S., Law, R., Haegens, S., Thorn, C. A., Hämäläinen, M. S., Moore, C. I., and Jones, S. R. (2016). Neural mechanisms of transient neocortical beta rhythms: Converging evidence from humans, computational modeling, monkeys, and mice. *Proceedings of the National Academy of Sciences*, 113(33):E4885–E4894. Publisher: Proceedings of the National Academy of Sciences.

Shin, H., Law, R., Tsutsui, S., Moore, C. I., and Jones, S. R. (2017). The rate of transient beta frequency events predicts behavior across tasks and species. *eLife*, 6:e29086. Publisher: eLife Sciences Publications, Ltd.

Stephan, K. E., Penny, W. D., Moran, R. J., den Ouden, H. E. M., Daunizeau, J., and Friston, K. J. (2010). Ten simple rules for dynamic causal modeling. *NeuroImage*, 49(4):3099–3109.

Talts, S., Betancourt, M., Simpson, D., Vehtari, A., and Gelman, A. (2020). Validating Bayesian Inference Algorithms with Simulation-Based Calibration. Technical Report arXiv:1804.06788, arXiv. arXiv:1804.06788 [stat] type: article.

Tejero-Cantero, A., Boelts, J., Deistler, M., Lueckmann, J.-M., Durkan, C., Gonçalves, P. J., Greenberg, D. S., and Macke, J. H. (2020). sbi: A toolkit for simulation-based inference. *Journal of Open Source Software*, 5(52):2505.

Thorpe, R. V., Black, C. J., Borton, D. A., Hu, L., Saab, C. Y., and Jones, S. R. (2021). Distinct neocortical mechanisms underlie human SI responses to median nerve and laser evoked peripheral activation. Pages: 2021.10.11.463545 Section: New Results.

Tolley, N., Rodrigues, P. L. C., Gramfort, A., and Jones, S. R. (2024). Methods and considerations for estimating parameters in biophysically detailed neural models with simulation based inference. *PLOS Computational Biology*, 20(2):e1011108. Publisher:

Public Library of Science.

van Ede, F., Quinn, A. J., Woolrich, M. W., and Nobre, A. C. (2018). Neural Oscillations: Sustained Rhythms or Transient Burst-Events? *Trends in Neurosciences*, 41(7):415–417.

Van Rossum, G. and Drake, F. L. (2009). *Python 3 Reference Manual*. CreateSpace, Scotts Valley, CA.

Vattikonda, A. N., Hashemi, M., Sip, V., Woodman, M. M., Bartolomei, F., and Jirsa, V. K. (2021). Identifying spatio-temporal seizure propagation patterns in epilepsy using Bayesian inference. *Communications Biology*, 4(1):1–13. Number: 1 Publisher: Nature Publishing Group.

Verhellen, J., Beshkov, K., Amundsen, S., Ness, T. V., and Einevoll, G. T. (2024). Multitask learning of a biophysically-detailed neuron model. *PLOS Computational Biology*, 20(7):e1011728. Publisher: Public Library of Science.

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., Carey, C. J., Polat, , Feng, Y., Moore, E. W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E. A., Harris, C. R., Archibald, A. M., Ribeiro, A. H., Pedregosa, F., and van Mulbregt, P. (2020). SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nature Methods*, 17(3):261–272. Number: 3 Publisher: Nature Publishing Group.

Ward, D., Cannon, P., Beaumont, M., Fasiolo, M., and Schmon, S. M. (2022). Robust Neural Posterior Estimation and Statistical Model Criticism.

Waskom, M. L. (2021). seaborn: statistical data visualization. *Journal of Open Source Software*, 6(60):3021.

Wessel, J. R. (2020). -Bursts Reveal the Trial-to-Trial Dynamics of Movement Initiation

and Cancellation. *Journal of Neuroscience*, 40(2):411–423. Publisher: Society for Neuroscience Section: Research Articles.

Zhao, D., Dalmasso, N., Izbicki, R., and Lee, A. B. (2021). Diagnostics for conditional density models and Bayesian inference algorithms. In *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence*, pages 1830–1840. PMLR. ISSN: 2640-3498.

CHAPTER 4

A novel framework for expanding RNNs with biophysical detail to solve cognitive tasks

Contributing Authors

Nicholas Tolley[1] and Stephanie R. Jones[1].

[1]Department of Neuroscience, Brown University, Providence, Rhode Island, United States of America

4.1 Abstract

Recurrent neural networks (RNNs) have proven to be highly successful in emulating human-like cognitive functions such as working memory. In recent years, RNNs are evolving to incorporate more biophysical realism to produce more plausible predictions on how cognitive tasks are solved in real neural circuits. However, there are major challenges in constructing and training networks with the complex and nonlinear properties of real neurons. A major component of the success of RNNs is that they share the same mathematical base as deep neural networks, permitting highly efficient optimization of model parameters using standard deep learning techniques. To do so, they use abstract representations of neurons which fail to capture the impact of cell-level biophysical and morphologic properties that may benefit network-level function. Expanding task-trained RNNs with biophysical properties such as dendrites and active ionic currents poses substantial challenges, as it moves these models away from the validated training regimes known to be highly effective for RNNs.

To address this gap, we developed a biophysically detailed reservoir computing (BRC) framework with the goal of extracting mechanistic insights from biophysical neural models, and propose that these insights can be used to guide model choices that will work for specific categories of cognitive tasks. The BRC network was constructed with synaptically coupled excitatory and inhibitory cells, in which the excitatory cells include multicompartment biophysically active dendrites; motivated by empirical studies suggesting dendrites have desirable computational benefits (e.g. pattern classification and coincidence detection). We trained the BRC network to do a simplified working memory task where it had to maintain the representation of an extrinsic "cue" input. We studied the impact of extrinsic input time constants (fast AMPA vs slow NMDA) and location (dendrite vs soma) on the ability of a network to solve the task. Our results revealed that cue inputs through NMDA receptors are particularly efficient for solving the working memory task. Further, the properties of NMDA receptors are uniquely suited for cue inputs delivered at the dendrite,

as networks trained with dendritic AMPA cue inputs failed to solve the task. Detailed examination of the cell and network dynamics that solve the task reveals distinct local network configurations and computing principles for the different types of extrinsic input. Overall, much like the body of mechanistic insights that have underpinned the success of training RNNs, this study lays the groundwork for applying the BRC framework to train biophysically detailed neural models to solve complex human-like cognitive tasks.

4.2 Introduction

Recurrent neural networks (RNNs) are a highly popular deep learning architecture that have most prominently been used for forecasting and classification tasks involving time series (Yu et al. (2019)). In recent years, they have been adapted to solve more neuroscience-relevant cognitive tasks that emulate human behaviors, such as working memory (Soo et al. (2023); Pals et al. (2024); Vyas et al. (2020); Pascanu and Jaeger (2011)). Task-trained neural models such as RNNs are useful tools for emulating neural dynamics that can solve cognitive tasks, but they lack biological realism that may convey useful computational properties (Yang and Molano-Mazón (2021)). While adding biology-inspired properties is a valuable goal for advancing the field, it is important to recognize that the historical success of task-trained neural models is largely due to their shared building blocks with deep neural networks, and the decades of research allowing scientists to know "what works" for training these simplified non-biophysical networks. Deep neural networks don't work out of the box, their success builds upon the mechanistic insights developed over vast prior research which inform choices such as network architecture (Khan et al. (2020)), data augmentation (Taylor and Nitschke (2018); Shorten and Khoshgoftaar (2019)), weight initialization and parameterization (Narkhede et al. (2022); Hanin and Rolnick (2018); Salimans and Kingma (2016)), loss function construction (Janocha and Czarnecki (2017); Wang et al. (2022)), and many other features. When adding biological realism, we are moving entirely away from the validated regimes where we confidently

know which choices work, as evidenced by the highly distinct set of methods that have been developed for the training of spiking neural networks (SNN, Eshraghian et al. (2023); Kim et al. (2019); Nicola and Clopath (2017); Arthur et al. (2023); Wang et al. (2023); Neftci et al. (2019)). SNNs already represent a dramatic departure from the basic foundations of RNNs (communication by spikes), however they represent a relatively minor modification with respect to the complexity of biophysically-detailed neural models. These issues highlight a need to develop mechanistic insights that help anticipate which choices work for a given task, and for the level of biological detail added to the network. Establishing mechanistic insights in task-trained biology-inspired RNNs with simple tasks (e.g. maintaining persistent representations) is an essential first step to filling this gap, and sets the foundation for scaling these models to challenging tasks that require manipulation of information held in working memory. Examples of complex cognitive functions that interact with working memory include abstract reasoning (D'Esposito (2007)), decision making (Murray et al. (2017)), planning (Gilhooly (2004)), and the execution of behavioral sequences (Rhodes et al. (2004)).

Motivated by identifying such mechanistic principles, in this study we considered the problem of expanding an RNN to include multicompartment biophysical neurons that have dendrites and active ion channels, and training it to solve a simplistic working memory task that requires maintaining a persistent representation of an external cue (e.g. external input Figure 4.1A and B). The addition of dendrites with active ion channels is, in our view, the most useful first step to expand biological realism, as dendrites are considered a promising addition for constructing more brain-inspired RNN models, with numerous empirical and computational studies indicating their unique computational benefits for processing task-relevant information from extrinsic sources (Larkum (2022); Poirazi and Papoutsi (2020)).

A major contribution of this study is the creation of a novel biophysically-detailed reservoir computing (BRC) framework (Figure 4.1B), which was specifically developed for

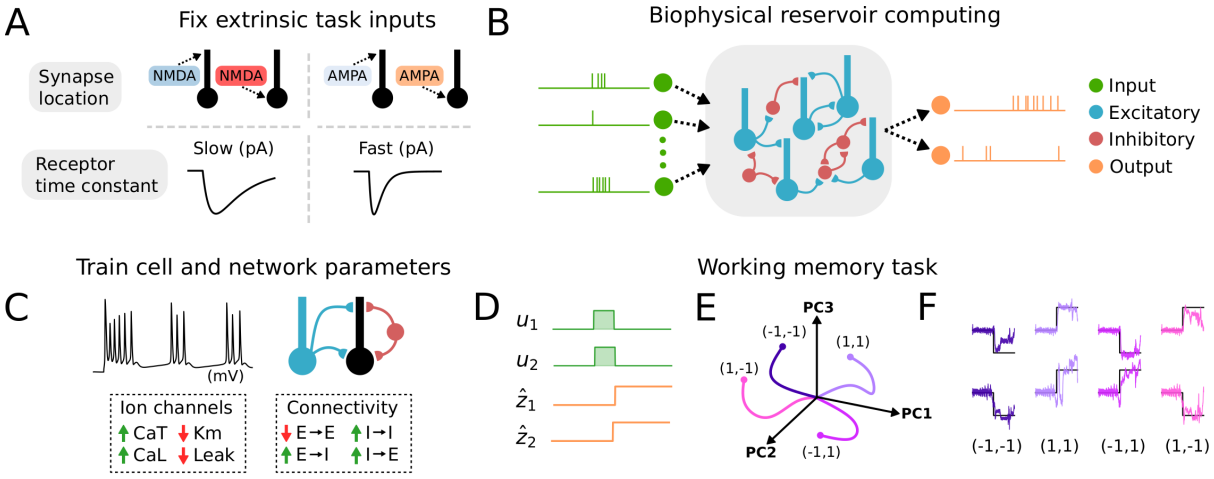

Figure 4.1: **Biophysical reservoir computing.** **A:** The goal of this study was to provide mechanistic insights to guide the construction and training of biophysical neural models to solve cognitive tasks. We focused on choices related to providing task-relevant extrinsic inputs to the network. 4 distinct networks were trained to perform a working memory task with fixed extrinsic input properties. These included slow time constant "NMDA-like" inputs at the dendrite or soma ($\texttt{NMDA}_{\texttt{dend}}$ and $\texttt{NMDA}_{\texttt{soma}}$), and fast time constant "AMPA-like" inputs at the dendrite or soma ($\texttt{AMPA}_{\texttt{dend}}$ and $\texttt{AMPA}_{\texttt{soma}}$). **B:** Schematic of the biophysical reservoir computer (BRC). The basic building blocks are multicompartment neurons representative of cortical excitatory and inhibitory neurons. Extrinsic inputs are delivered through a population of spiking neurons that are tuned to task-related information. The biophysical reservoir consists of recurrently connected excitatory and inhibitory neurons. Outputs are calculated as a weighted combination of the firing rates of a subset of neurons. **C**: With extrinsic input properties fixed (panel A), the local cell (i.e. ion channel conductance) and network (i.e. synaptic connectivity) parameters of the BRC were optimized for the working memory task. **D:** Cue information for the working memory task (u_1 and u_2) is delivered for a brief 25 ms period. The target output of the network ($\hat{z}_1$ and $\hat{z}_2$) is a 2 dimensional vector that matches the sign of the cue inputs. The goal of the task is to match the target output after the termination of the cue input. **E:** Schematic neural trajectories of a network successfully trained to perform the working memory task. Each trajectory represents a separate trial condition. **F:** Outputs of a network successfully trained to perform the working memory task.

deriving mechanistic insights from more biophysical RNN models trained to solve cognitive tasks (i.e. working memory). We directly build upon the biophysical modeling platform Jaxley (Deistler et al. (2024)), a powerful software package which applies modern deep learning techniques to the simulation and training of detailed biophysical neural models. Due to the complexity of the BRC network, current optimization methods available in Jaxley (e.g. gradient-based optimization) were insufficient to train the network on the

working memory task. Therefore, we developed a novel deep learning based evolutionary algorithm (Figure 4.2A, see "Neural flow evolution" section of Methods).

Reservoir computing was intentionally chosen for the purpose of finding network configurations that can perform the task, as opposed to targeted optimization of parameters on a cell-by-cell basis. Reservoir computing refers to a system where input signals are mapped into a high- dimensional space through the activity of a fixed dynamical system (i.e. the reservoir) (Tanaka et al. (2019); Damicelli et al. (2022); Zhang and Vargas (2023)). The training primarily occurs on the linear readout weights, and shared parameters of the reservoir that shape its response characteristics. The rationale is that we are looking to identify computationally useful properties (e.g. connectivity rules, spatial distribution of ion channel conductances, ideal morphologic complexity of dendrites) that can be generalized to any network obeying a limited set of biological constraints. This method is similar to pre-training of deep neural networks, where the goal is to create a generically useful base network that can be fine-tuned to a specific task (Hasson et al. (2020); Erhan et al. (2010); Larochelle et al. (2009)).

We hypothesize that expanding task-trained neural models to include dendrites necessitates a careful treatment of 1) the location (e.g. dendrite vs. soma) and synaptic time constants of the extrinsic "cue" input onto the dendrite-equipped neurons, and 2) the intrinsic biophysics (e.g. ionic currents) and local synaptic connectivity of the dendrite-equipped neurons (Figure 4.1A). These hypotheses are largely motivated by experimental neuroscience indicating the importance of the aforementioned properties for working memory.

With respect to synapse time constants, numerous experimental and computational studies have implicated NMDA receptors as a vital component of working memory in the neocortex (Wang (2021); Self et al. (2012); Vugt et al. (2020); Stefanos et al. (2019)). The most compelling forms of evidence demonstrate that selective blockade of NMDA receptors in dorsolateral prefrontal cortex impairs persistent firing of cortical neurons

(Vugt et al. (2020); Wang et al. (2013)). The distinctive property of NMDA receptors is their slow time constants, and magnesium ion block (Abbott and Dayan (2001)). These properties combine to function as a coincidence detector, where slight depolarization of the postsynaptic neuron is required to remove the magnesium ion block from NMDA receptors. If a presynaptic neuron delivers glutamate to a postsynaptic NMDA receptor, coincident with a slight depolarization of the postsynaptic neuron, then a sustained depolarization occurs due to the slow time constants of NMDA receptors.

With respect to synapse location, NMDA receptors in rodents are observed to be selectively enriched at distal regions of the apical dendrite, and decrease with closer proximity to the soma (Bittner et al. (2012)). These experimental observations suggest that NMDA serves a functional role that is potentially related to the physiology of pyramidal neuron dendrites. Further, numerous studies have emphasized that this spatial localization of NMDA receptors underlies several of the potential computational benefits provided by dendrites (Rhodes (2006); Palmer et al. (2014); Schulz et al. (2018); Oláh et al. (2025); Major et al. (2013)).

While these experimental and computational studies are promising indicators that both synapse location and time constants are critical components of working memory, it is unclear if this conclusion holds in task-trained biophysical models that are not necessarily constrained to match real brain activity. It is important to emphasize that the framework described here was not designed for determining how computations are carried out in real brains. Instead we ask the following question: if you are allowed to recombine cell-level building blocks of biology, which biophysical properties provide task-specific computational benefits that can be transferred to biology-inspired artificial intelligence (AI) systems? While the choice in biophysical properties can be motivated by neuroscience literature, the trained models are not required to mimic realistic brain activity. This allows a disentanglement of the functional role of biophysical properties, from biological constraints imposed by real brains that are unrelated to the cognitive task

being considered.

With these considerations in mind, we ask the following questions: Does the synaptic location and time constants of task-relevant extrinsic inputs representing the cue impact the ability of a network to maintain persistent representations? (Figure 4.1A-B and D-F)? How do these variations in extrinsic inputs impact the learned intrinsic biophysics and local connectivity necessary for persistent representations (Figure 4.1C)? Answering these questions would provide valuable mechanistic insights for intelligently constructing task-trained biophysical models, and improve training efficiency by providing validated choices for biological details (such as connectivity, receptor time constants, morphological complexity, ion channels, cell heterogeneity, and combinations of these details) thus avoiding choices that are destined to fail due to the unique mechanistic properties of biophysical neural models.

4.3 Methods

4.3.1 Biophysical reservoir computing

Neuron simulations were carried out with the simulation software Jaxley (Deistler et al. (2024)). The biophysical reservoir consisted of 50 excitatory neurons, and 25 inhibitory neurons (Figure 4.1A).

Excitatory neurons were modeled such that their main morphologic feature was a large apical dendrite that attenuated synaptic inputs at distal locations (assuming a passive dendrite). This choice was made to focus on networks with neurons that use active ion channels to propagate distal synaptic inputs to the soma. Figure 4.6A(iv) shows a schematic of the excitatory neuron's morphology. The morphology of the excitatory neurons was based on an example from Pagkalos et al. (2023) (Figure 3 in the corresponding publication) titled "A reduced compartmental model that replicates active dendritic properties". The excitatory neuron was constructed with 4 compartments total, where each compartment

had 4 subcompartments over which the voltage was calculated. The length and diameter of each compartment was as follows: soma (25×25 μm), 1st apical segment (100×2.5 μm), 2nd apical segment (100×1 μm), 3rd apical segment (100×0.5 μm). Inhibitory neurons consisted of a single compartment (with 4 subcompartments) representing the soma, with a length and diameter of (5×5 μm).

Both excitatory and inhibitory neurons were equipped with a diverse set of ion channels available in the base software of Jaxley. The ion channels were based on models described in Pospischil et al. (2008). These included sodium, potassium, leak, T-type calcium, L-Type calcium, and slow M potassium channels. The maximal conductance ($\bar{g}$) of sodium and potassium channels were fixed to their default values of 0.05 S/cm^2 and 0.005 S/cm^2. The maximal conductance of the remaining ion channels were treated as free parameters to be optimized. The range of values for each parameter are described in Table 4.1.

Our network model included AMPA, NMDA, and GABAa synapses. The synapse models were imported from the Jaxley-mech library (`https://github.com/jaxleyverse/jaxley-mech`), a companion repository to Jaxley (Deistler et al. (2024)). In general, we used the default parameter values for each synapse, which match the fitted values in Destexhe et al. (1998). The only exception is for experiments where the time constants of NMDA and AMPA synapses were modified for extrinsic inputs. See the "Extrinsic synaptic receptors" section of Methods for a detailed description. The maximal conductance for each ion channel was set to be equal across all neurons. For excitatory neurons, ion channel maximal conductance parameters were set separately for the soma (1 value for 1 compartment) and the dendrite (1 value for 3 compartments).

Background noise was delivered through AMPA synapses located on the soma of all neurons with spike times generated from a poisson-like process. Each neuron received independent noisy inputs with an average firing rate of 1 Hz and maximum conductance of 1×10^{-4} nS.

A Watts-Strogatz graph with k=5 neighbors was used to establish the baseline connectivity with a small-world topology (Watts and Strogatz (1998)). Cortical circuits have been shown to obey small-world connectivity patterns (Sporns and Zwi (2004)), and this unique topology is believed to promote dynamical complexity in neural circuits (Shanahan (2008); Pazzini et al. (2021)). Excitatory and inhibitory neurons were recurrently connected through AMPA and GABAa synapses. Synapse locations included the soma and the 3rd apical segment (furthest from soma) for excitatory neurons, and exclusively the soma for inhibitory neurons.

A population of 25 input neurons was used to deliver task-related cue information. Input neurons synapse at 2 possible locations on excitatory neurons (soma and 3rd apical segment). Inhibitory neurons received synaptic input at the soma. Synapse identity to excitatory neurons was either NMDA or AMPA. Input neurons always connected to inhibitory neurons through AMPA synapses. Input neurons were only connected to half of the neurons in the biophysical reservoir. The firing rates of the other half of neurons were used as readouts. This constraint required cue related information to be transmitted through at least one recurrent synapse for successful performance of the task.

The conductance and connection probability of all connections (input and recurrent) was treated as a free parameter to be optimized. The range of values for each parameter are described in Table 4.1.

All parameters described in Table 4.1, as well as fixed parameter values, were set to be equal for every neuron belonging to the same class (i.e. excitatory soma, excitatory dendrite, inhibitory, and input neurons). During training with "Neural flow evolution" (see corresponding section below), the parameter updates were applied uniformly across all neurons. The primary sources of stochasticity and heterogeneity in the network were 1) distinct populations of excitatory and inhibitory neurons, 2) noisy background inputs, and 3) probabilistic connectivity.

4.3.2 Extrinsic synaptic receptors

The kinetics of AMPA and NMDA receptors were modeled according to Destexhe et al. (1998) and Deistler et al. (2024) as

$$\frac{dr}{dt} = \alpha[T](1 - r) - \beta r \tag{4.1}$$

where r is the fraction of receptors in the open state, T is transmitter concentration, α is the forward rate constant, and β is the backward rate constant. The AMPA receptor postsynaptic current I_{AMPA} was calculated as

$$I_{AMPA} = \bar{g}_{AMPA} r (V - E_{AMPA}) \tag{4.2}$$

where $\bar{g}_{AMPA}$ is the maximal conductance, V is the postsynaptic membrane potential, and E_{AMPA} is the reversal potential.

The NMDA receptor postsynaptic current I_{NMDA} was calculated as

$$I_{NMDA} = \bar{g}_{NMDA} B(V) r (V - E_{NMDA}) \tag{4.3}$$

where $B(V)$ represented a voltage-dependent magnesium block modeled as

$$\frac{1}{1 + \exp(-0.062V)[\mathrm{Mg}^{2+}]_o / 3.57} \tag{4.4}$$

where $[\mathrm{Mg}^{2+}]_o$ is the extracellular magnesium concentration. We used the default $[\mathrm{Mg}^{2+}]_o = 1mM$.

As described in Destexhe et al. (1998), the forward and backward rate constants for AMPA receptors were set to $\alpha = 1.1 \times 10^6 M^{-1}s^{-1}$ and $\beta = 190s^{-1}$. The rate constants for NMDA receptors were set to $\alpha = 7.2 \times 10^4 M^{-1}s^{-1}$ and $\beta = 6.6s^{-1}$.

In the final two sections of the results, we investigated what aspect of NMDA receptor

properties were important for attractor dynamics. To disentangle the importance of the rate constants from the magnesium block, we ran simulations where α and β for AMPA and NMDA were swapped. This allowed simulations of a slow "AMPA-like" receptor, and a fast "NMDA-like" receptor.

4.3.3 Working memory task

A simple working memory task was used to assess the impact of cell-level biophysical properties on network-level dynamics. The task required responding to a cue signal, and maintaining a unique representation of the cue signal until the end of the simulation (Figure 4.1(D-F)). Cue information was delivered through input neurons activated at 250 ms for a duration of 25 ms. The cue signal consisted of a 2-dimensional vector that remained at zero in the pre-stimulus period, and immediately took on a value $\hat{u} \in \{-1, 1\}^2$ in the cue stimulus period for a total of 4 unique task conditions (Figure 4.1F).

The population of input neurons encoded cue information by mimicking sensory neurons with a gaussian receptive field. Each neuron was assigned a 2-dimensional gaussian distribution with a random mean $\mu \in [-2, 2]^2$ and standard deviation $\sigma = \left(\begin{smallmatrix} 0.3 & 0 \\ 0 & 0.3 \end{smallmatrix}\right)$. During the stimulus period where cue information was delivered, a 25 ms current injection was applied to the soma of input neurons with an amplitude scaling factor proportional to the gaussian receptive field evaluated at the cue location. The maximum amplitude of the current injection was set to 0.06 nA.

The output $z(t)$ of the network was defined as a linear readout such that

$$z(t) = Wh(t) + b \tag{4.5}$$

where $h(t)$ approximates the instantaneous firing rate of excitatory neurons, W is a weight matrix of shape $d \times n$ (d = dimensionality of output, n = number of excitatory neurons), and b is a bias term. The terms W and b are learned through linear regression. In this

study we used Ridge regression with regularization parameter $\alpha = 2.0$.

Task performance was evaluated with the loss function:

$$L = \frac{1}{T} \sum_{t} (z(t) - \hat{z}(t))^2 \tag{4.6}$$

where $z(t)$ is the network output, $\hat{z}(t)$ is the target output, T is the total number of time steps, and t is the current time step. For each set of parameters, the network was simulated with the 4 unique task conditions for 5 trials making a total of 20 simulations per parameter set. The loss used for optimization was calculated as the average loss from leave-one-out cross validation, where 4 of the trials were used to train a linear regression model to predict outputs on the remaining held out trial.

4.3.4 Neural flow evolution

To optimize parameters of the network model, we took an approach inspired by evolutionary algorithms (Yu and Gen (2010)), and combined it with neural density estimation (Papamakarios (2019)) (Figure 4.2A).

The key steps of an evolutionary algorithm include 1) generate the initial population $\theta \sim p(\theta)$ through uniform sampling of the prior distribution, 2) evaluate the fitness of the population by simulating parameters θ, 3) select the parent population by choosing individuals with high fitness (here we selected parents whose fitness exceeded the 90th percentile of the current generation), and 4) produce offspring by recombining parameters of the parents. For this study, the fitness of each parameter combination (step 2) was evaluated as the negative of the loss function such that the fitness is maximized.

A methodological innovation in this study is the use of neural density estimation (Papamakarios (2019)) to produce offspring from the parent population. Given the parameters θ of the parent population, the neural density estimator $q_\phi(\theta)$ is trained to maximize the average log likelihood:

$$L(\phi) = \frac{1}{N} \sum_{n} \log q_{\phi}(\theta_n). \tag{4.7}$$

In this study we used a masked autoregressive flow architecture for neural density estimation, and trained the estimator with Adam for 5000 iterations. For each generation, a newly initialized estimator was used. Elitism was implemented in the evolutionary algorithm such that all parameters and simulations were cached in a common dataset during optimization. This allowed parameters from previous generations that exceeded the fitness threshold to be included in subsequent parent populations. To prevent the algorithm from prematurely converging on parameters sets that produced minimal activity, we subtracted a large penalty from the fitness of parameters where the mean absolute value of the output z was less than 0.1.

The formal steps of neural flow evolution are as follows:

Algorithm 1: Neural flow evolution

input : prior distribution $p(\theta)$, estimator $q_\phi(\theta)$, complete dataset $\mathcal{D}$, training dataset $\mathcal{D}_{\text{train}}$, number of rounds R, simulations per round N

output : error filtered distribution $p_{\mathcal{D}}(L, \theta)$

set $p_{\mathcal{D}}(\theta) = p(\theta)$; $\mathcal{D} = \{\}$; $\mathcal{D}_{\text{train}} = \{\}$

for $r = 1 : R$ **do**
  initialize $q_\phi(\theta)$
  for $n = 1 : N$ **do**
    sample $\theta_n \sim p_{\mathcal{D}}(\theta)$
    simulate $h_n \sim p(h \mid \theta_n)$
    calculate loss $L_n \leftarrow \text{Ridge}(\text{h}_\text{n}, \hat{\text{z}})$
    add (L_n, θ_n) into $\mathcal{D}$
  end
  filter $\mathcal{D}_{\text{train}} = \mathcal{D}\mathbb{I}(L < \text{quantile}(\text{L}_{1:(\text{r}\times\text{N})}))$
  train $q_\phi(\theta)$ on $\mathcal{D}_{\text{train}}$
end

We used a simulation budget of $N = 100$ and trained the networks for $R = 10$ rounds. Each individual "simulation" actually corresponded to 4 separate simulations for each condition in the working memory task. As noted above, loss was estimated using leave-one-out cross validation with 5 repeats. Therefore, $100 \times 10 \times 4 \times 5 = 20,000$ 1 second simulations (differential equation solver time step $dt = 0.025$ ms) were run for the optimization of each extrinsic input variant. Simulations were run on GPU nodes available on Brown University's high-performance computing cluster (Oscar). Optimization of an individual extrinsic input variant on a single GPU took approximately 15 hours using the optimization parameters described above.

4.3.5 Overlap coefficient

To compare the optimized parameter distributions learned by the estimator $q_\phi(\theta)$, we used the distribution overlap coefficient (OVL) (Pastore and Calcagnì (2019); Tolley et al. (2024)) which is calculated as:

$$\text{OVL}_k(q_\phi, \hat{q}_\phi) = \int \min\Big(q_\phi(\theta[k]), \hat{q}_\phi(\theta[k])\Big) d\theta \tag{4.8}$$

where $q_\phi(\theta[k])$ and $\hat{q}_\phi(\theta[k])$ are two different distributions we seek to compare on the marginal distribution for the k-th parameter. To calculate the OVL_k numerically we used an evenly spaced grid of size 30^d over $d \in \mathbb{N}_+$ parameters. The OVL varies on a scale of [0,1] such that 1 indicates complete overlap, and 0 indicates no overlap.

4.4 Results

4.4.1 BRC reveals that NMDA receptors are more efficient than AMPA receptors in solving working memory tasks

We applied biophysical reservoir computing (BRC) to study how synapse location and receptor time constants of extrinsic inputs impacted a network's ability to solve a simplified working memory task. We tested 4 extrinsic input variants using the biophysical reservoir computing (BRC) framework, where the properties of task-relevant extrinsic inputs were fixed (Figure 4.1A), and intrinsic biophysics and connectivity were freely optimized (Figure 4.1C). In each network variant, the synapse location of extrinsic inputs to excitatory neurons was either the soma or the 3rd apical segment (furthest from the soma), and the synapse time constants were either slow ("NMDA-like" $\texttt{NMDA}_{\texttt{dend}}$ and $\texttt{NMDA}_{\texttt{soma}}$) or fast ("AMPA-like" $\texttt{AMPA}_{\texttt{dend}}$ and $\texttt{AMPA}_{\texttt{soma}}$).

We trained each of the 4 extrinsic input variants on the working memory task (see "Working memory task" section of Methods), and examined the resulting population errors.

Figure 4.2B plots the average population error for each extrinsic input variant over 10 epochs. As shown, $\texttt{AMPA}_{\texttt{dend}}$ was the only network that failed to reach the target error threshold of 1.0, and both of the NMDA conditions reached the target threshold faster than the $\texttt{AMPA}_{\texttt{soma}}$ condition. More specifically, both $\texttt{NMDA}_{\texttt{dend}}$ and $\texttt{NMDA}_{\texttt{soma}}$ converged to the target error within 5 epochs, while $\texttt{AMPA}_{\texttt{soma}}$ converged to the error threshold within 8 epochs. Further, for networks with inputs on the dendrites, NMDA receptors are required for learning that task, as networks with dendritic AMPA receptors were unable to learn the task. For task-related inputs on the soma, either NMDA or AMPA receptors were sufficient for performing the working memory task, but NMDA outperformed the AMPA receptors.

To gain mechanistic insight into how the network learned the task, we also examined the resulting local synaptic connectivities after training in each extrinsic input variant that successfully learned the tasks ($\texttt{NMDA}_{\texttt{dend}}$, $\texttt{NMDA}_{\texttt{soma}}$, $\texttt{AMPA}_{\texttt{soma}}$) to assess if and how the connectivities needed to be set up to solve the working memory task. By plotting histograms of the learned synaptic weights of local connections, it is possible to identify distinct network configurations (local synaptic connectivity patterns) for each extrinsic input variant when their corresponding histograms are non-overlapping. Differences between parameter distributions were quantified using the overlap coefficient (OVL, see "Overlap coefficient" section of Methods), where an OVL of 0.0 indicates a large separation between distributions, and an OVL of 1.0 indicates complete overlap. Figure 4.2C plots the distribution of learned synaptic strengths for all local recurrent connections in the successfully trained extrinsic input variants ($\texttt{NMDA}_{\texttt{dend}}$ (dark blue), $\texttt{NMDA}_{\texttt{soma}}$ (red), and $\texttt{AMPA}_{\texttt{soma}}$ (orange)).

As shown, the extrinsic input variants produced highly distinct network configurations. Synaptic targets of connections described below are abbreviated with E_s (excitatory soma), E_d (excitatory dendrite), and I (inhibitory). Across the learned local connection weights, every variant exhibited a high separation from the other two variants for at least one

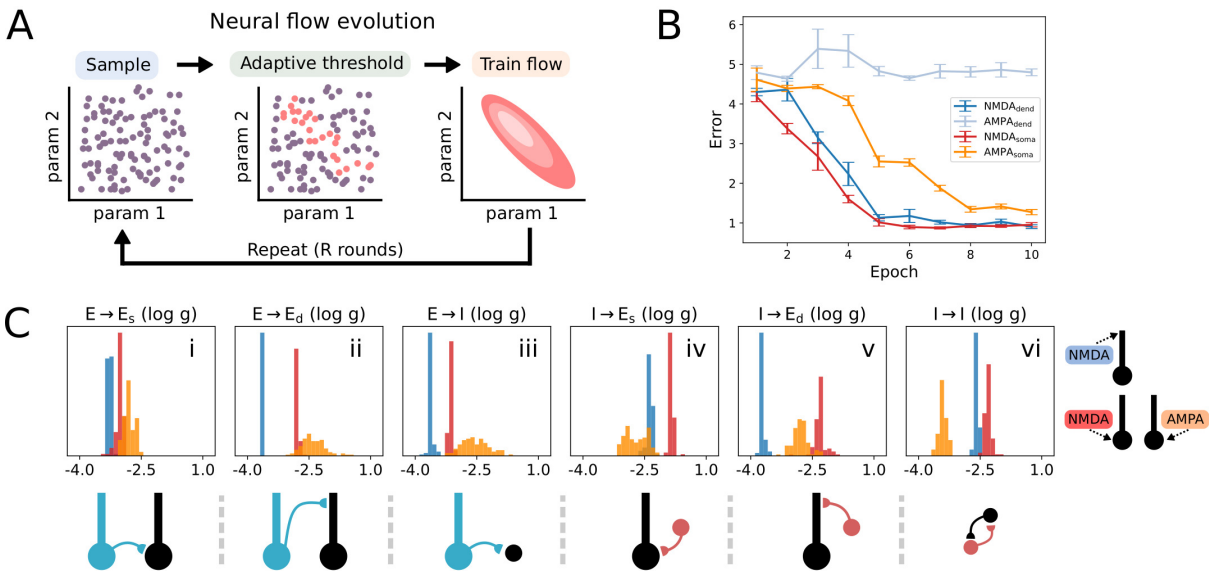

Figure 4.2: **Neural flow evolution. A:** Schematic of the evolutionary algorithm used to optimize the parameters of the biophysical reservoir computer to perform a working memory task. The initial population of parameters is created by uniformly sampling the parameter space. Each parameter set is simulated and assigned a fitness score for its performance on the working memory task. An adaptive threshold is calculated based on the quantile of the current population's error. A neural density estimator is trained to approximate the distribution of parameters that remain after thresholding. A new generation is created by sampling from the trained neural density estimator. **B**: Optimization error for $\texttt{NMDA}_{\texttt{dend}}$ (dark blue), $\texttt{AMPA}_{\texttt{dend}}$ (light blue), $\texttt{NMDA}_{\texttt{soma}}$ (red) and $\texttt{AMPA}_{\texttt{soma}}$ (orange) extrinsic input variants. NMDA inputs at either the soma or dendrite rapidly converged to the target error level of 1.0 compared to the AMPA inputs. While $\texttt{AMPA}_{\texttt{soma}}$ eventually converges to the target error level, $\texttt{AMPA}_{\texttt{dend}}$ remained at chance level error. **C(i-vi):** Optimized connectivity parameters for the successfully trained network variants $\texttt{NMDA}_{\texttt{dend}}$, $\texttt{NMDA}_{\texttt{soma}}$, and $\texttt{AMPA}_{\texttt{dend}}$. Values correspond to synaptic conductance in nanosiemens. Connections shown included E → E_s (i), E → E_d (ii), E → I (iii), I → E_s (iv), I → E_d (v), and I → I (vi)

synapse. The only connection which exhibited low separation for all 3 variants was E → E_s (Figure 4.2C(i)).

We were specifically interested in understanding if learned network configurations were different with respect to extrinsic input location (soma vs dendrite), and time constants (AMPA vs NMDA). Starting with extrinsic input location, we can compare the dendritic input variant $\texttt{NMDA}_{\texttt{dend}}$, to the somatic input variants $\texttt{NMDA}_{\texttt{soma}}$ and $\texttt{AMPA}_{\texttt{soma}}$. For the connections E → E_d (Figure 4.2D(ii)) and I → E_d (Figure 4.2D(v)), $\texttt{NMDA}_{\texttt{dend}}$ exhibited much lower weights compared to either $\texttt{NMDA}_{\texttt{soma}}$ or $\texttt{AMPA}_{\texttt{soma}}$ such that the histograms had

no overlap (OVL=0.0 for $\texttt{NMDA}_{\texttt{dend}}$ vs $\texttt{NMDA}_{\texttt{soma}}$ or $\texttt{AMPA}_{\texttt{soma}}$). It is interesting to observe that these strongly decreased connections occurred at the dendrite, suggesting that reducing the strength of local dendritic inputs is necessary when delivering task-relevant extrinsic inputs to the dendrite.

For comparing network configurations with respect to extrinsic input time constants (NMDA vs AMPA), we observe that networks trained with $\texttt{NMDA}_{\texttt{dend}}$ and $\texttt{NMDA}_{\texttt{soma}}$ produced substantially lower synaptic strengths for all excitatory connections compared to $\texttt{AMPA}_{\texttt{soma}}$ (Figure 4.2C(i-iii)). We additionally observe that nearly all inhibitory connections for $\texttt{NMDA}_{\texttt{dend}}$ and $\texttt{NMDA}_{\texttt{soma}}$ are higher than $\texttt{AMPA}_{\texttt{soma}}$ (Figure 4.2C(iv-vi)), with the exception of I $\rightarrow$ E_{d} (Figure 4.2C(v)) where $\texttt{NMDA}_{\texttt{dend}}$ has substantially lower weights compared to the other extrinsic input variants as described above, suggesting this connection is less useful for learning. We additionally note a large separation between both NMDA variants and $\texttt{AMPA}_{\texttt{soma}}$ for the I $\rightarrow$ I connections (Figure 4.2C(vi)). In fact, this was the only connection in which $\texttt{AMPA}_{\texttt{soma}}$ had an OVL of 0.0 with both of the NMDA network variants. Previous studies have implicated I $\rightarrow$ I connections as being important for working memory tasks (Kim and Sejnowski (2021)), potentially suggesting a relationship between NMDA transmission and inhibitory connectivity for the ability to perform working memory tasks. However, we caution overinterpretation of the optimized parameter values in the context of neuroscience. Our goal was not to construct a model that directly explains how real brains function, but instead identify computational principles inspired by the brain. Nevertheless, the BRC framework could be used in follow up studies to characterize I $\rightarrow$ I connections when they are fixed to specific values in the same way that extrinsic inputs are treated in this study.

In summary, training the BRC with neural flow evolution revealed that the synapse location and receptor time constants of task-relevant extrinsic inputs have a strong impact on the learned network configurations necessary for working memory. Further, networks with $\texttt{AMPA}_{\texttt{dend}}$ (i.e. fast time constants delivered at the dendrite), were completely

unsuccessful in learning the working memory task, despite the intrinsic biophysics and connectivity being freely optimized. While there are potential computational benefits of dendritic information processing, we demonstrate a constraint on task-related dendritic function, such that NMDA receptors are required to transmit cue information through dendrites for working memory, and overall NMDA seems to be more efficient than AMPA for working memory tasks.

For the remaining sections, we will attempt to understand how these differences emerge with analysis techniques traditionally employed by systems neuroscientists, along with a description of their biological significance. We note that while we adopt tools from experimental neuroscience, our ability to make inferences from simulated neural activity are much stronger than in vivo activity, as we have direct access to any measurable property of all neurons simulated.

4.4.2 Successfully trained networks produce distinct and persistent changes in network-level spiking activity

In the previous sections, we examined the impact of variants in extrinsic cue input location and time constant on training in the working memory task by inspecting errors in task performance, and the learned connection strengths of the local network. We now move to analyze the cell spiking activity during the working memory task. Spiking activity is essential to characterize when attempting to understand how a biophysical neural network solves a computational task, as spikes indicate the exact moment at which communication between cells is occurring. Similar to experimental neuroscience, by inspecting ongoing spiking activity, we will uncover differences in spiking mechanisms underlying the ability to maintain persistent representations in the BRC network, and identify which aspects of spiking activity indicate good performance.

Empirical studies have shown that working memory processes in frontal cortical networks require both 1) changes in spiking activity that are distinct for different sensory-

mediated extrinsic inputs, and 2) a persistent change in spiking activity that lasts after the extrinsic input ends (Wang (2021); Vugt et al. (2020)). As such, we were interested in examining how differences in extrinsic inputs (and their optimized intrinsic parameters) impact spiking dynamics to mediate task performance, and if the principles that emerge were consistent with the prior empirical studies. We specifically asked if synapse location and receptor time constants of extrinsic inputs influence network-level spiking response before and after the stimulus, and if the optimized intrinsic parameters for each extrinsic input lead to differences in overall spiking activity throughout the simulation.

Figure 4.3 and Figure 4.4 plot spiking activity in each neuron (panels B and D) and the corresponding network output (panels A and C) for a single trial across all task conditions (panels i-iv). Network outputs were produced by a learned set of linear weights (see "Working memory task" set of Methods, outputs for a held out validation set are shown). The simulations were generated from the best performing parameter set identified during optimization. In Figure 4.5A and B below, we provide a more detailed view by plotting the spiking response of individual neurons across all task conditions.

The networks that were successfully trained to perform the working memory task ($\texttt{NMDA}_{\texttt{dend}}$ in Figure 4.3A, $\texttt{NMDA}_{\texttt{soma}}$ and $\texttt{AMPA}_{\texttt{soma}}$ in Figure 4.4A and C) all show network outputs (colored lines) that closely match the target signal (black lines). In contrast, $\texttt{AMPA}_{\texttt{dend}}$, which was not successfully trained, produces outputs with small fluctuations that are uncorrelated with the target signal (Figure 4.3C).

Inspection of the spike raster plots reveals why $\texttt{AMPA}_{\texttt{dend}}$ exhibits poor performance on the task (Figure 4.3D). Despite the presence of persistent spiking activity at baseline in excitatory (blue) neurons, activity of the input neurons (green) during the cue stimulus period (250-275 ms) fails to produce any time-locked change that lasts to the end of the 1 second simulation. A small increase in inhibitory (red) neuron spiking aligned to the cue stimulus is observed, but this change is transient as inhibitory neurons primarily exhibit stochastic spiking due to the noisy background input. We hypothesize that strong and

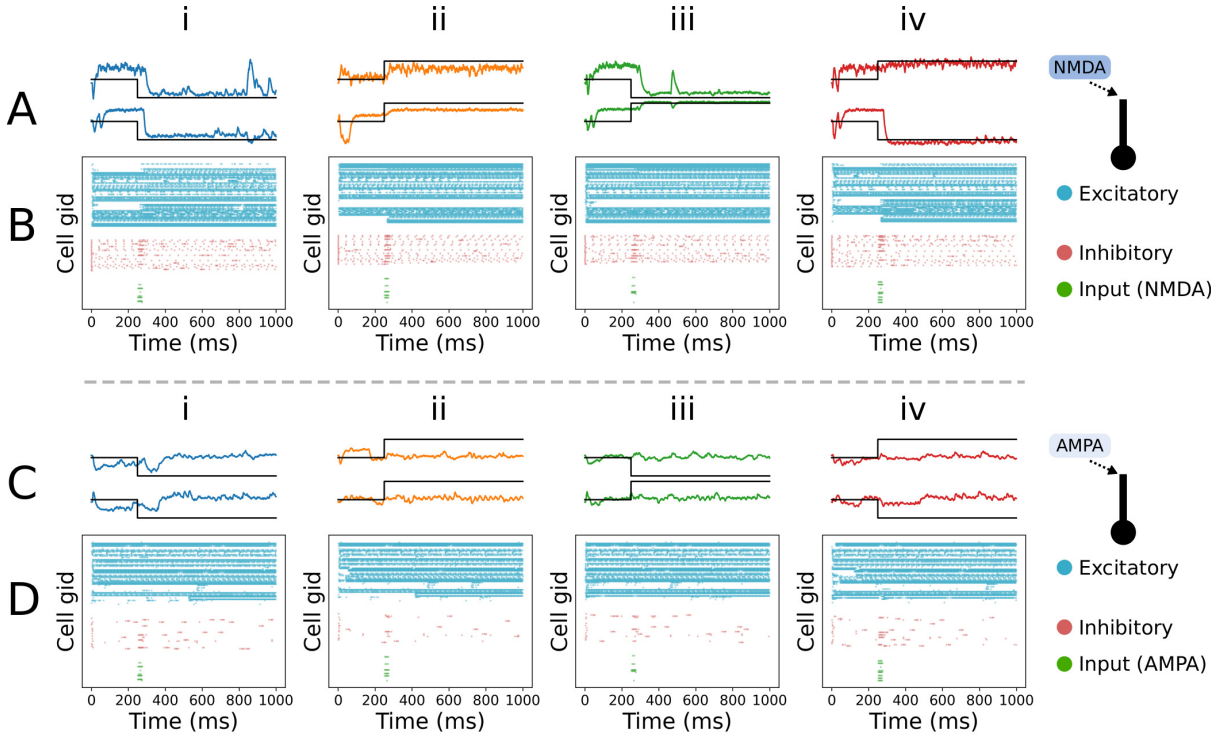

Figure 4.3: **Network-level spiking response to dendritic inputs. A (i-iv):** Outputs of the trained biophysical reservoir computer with dendritic NMDA inputs ($\texttt{NMDA}_{\texttt{dend}}$). Panels i-iv correspond to the 4 possible types of input stimuli. Model predictions (colored lines) are overlaid on the target output (black lines). As shown, the $\texttt{NMDA}_{\texttt{dend}}$ network was successfully trained to perform the working memory task and maintains outputs in close proximity to the target. **B (i-iv)**: Spike rasters for the simulations underlying the outputs shown in panel A. Input neurons (green) spike briefly during the cue stimulus period (250-275 ms) triggering an immediate change in the spiking of both excitatory (blue) and inhibitory (red) neurons. The change in spiking is unique for all 4 task conditions and persists for the remainder of the 1 second simulation. **C (i-iv)**: Outputs of the biophysical reservoir computer with dendritic AMPA inputs ($\texttt{AMPA}_{\texttt{dend}}$). The optimizer was unsuccessful in training the network with $\texttt{AMPA}_{\texttt{dend}}$ (simulations from the best scoring model are shown). Model outputs hover near zero indicating that activity from the network is not distinct across the 4 task conditions, and does not persistently change after the cue inputs. **D (i-iv)**: Spike rasters for the simulations underlying the outputs shown in panel B. Activity of input neurons at 250-275 ms produces no visible change in spiking of excitatory neurons, and a minimal increase in spiking of inhibitory neurons. The change in spiking does not persist after the cue stimulus period, and is not visibly distinct for the 4 task conditions.

synchronous inhibition is key to resetting the spiking activity in the excitatory neurons, permitting the formation of newly activated clusters that encode the memory. This strong and synchronous inhibition does not occur in this case of $\texttt{AMPA}_{\texttt{soma}}$ (Figure 4.3D), in

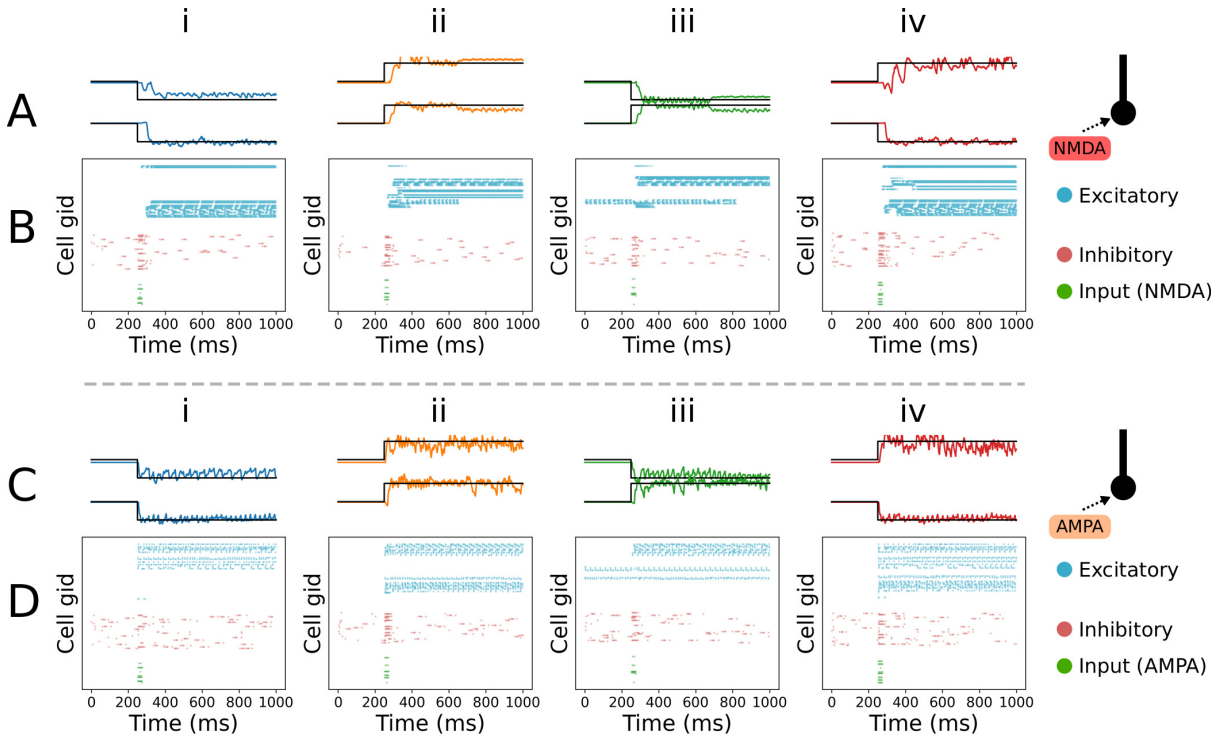

Figure 4.4: **Network-level spiking response to somatic inputs.** Figure panels follow the same structure as Figure 4.3. **A-B**: Model outputs (A) and spike rasters (B) are shown for the network trained with $\texttt{NMDA}_{\texttt{soma}}$ inputs. Across all task conditions (i-iv), the network produces a robust response to the cue input, with distinct clusters of excitatory neurons that maintain persistent firing to the end of the simulation. **C-D**: Model outputs (C) and spike rasters (D) are shown for the network trained with $\texttt{AMPA}_{\texttt{soma}}$ inputs. Similar to $\texttt{NMDA}_{\texttt{soma}}$ (panels A-B), the network exhibits clustered excitatory neuron activity that is distinct for each task condition.

contrast to $\texttt{NMDA}_{\texttt{dend}}$ and $\texttt{AMPA}_{\texttt{soma}}$ which both exhibit a strong recruitment of inhibitory neuron spiking that is time-locked to the cue input (Figure 4.3B and Figure 4.4B and C).

We also observe that the excitatory neurons in $\texttt{AMPA}_{\texttt{dend}}$ seem to have no clear change in spiking aligned to the cue stimulus (Figure 4.3D). This is in contrast to the $\texttt{NMDA}_{\texttt{dend}}$ network variant where in addition to a cue-aligned change in both excitatory and inhibitory neuron spiking, there is relatively higher inhibitory neuron spiking at baseline (Figure 4.3B red dots prior to 250 ms). Figure 4.5B(ii) shows this effect with a spike raster for a single inhibitory neuron, and demonstrates that despite the neuron being selectively activated by input neurons during different trial conditions (colored bars), this activation has no sustained impact in the post-stimulus period.

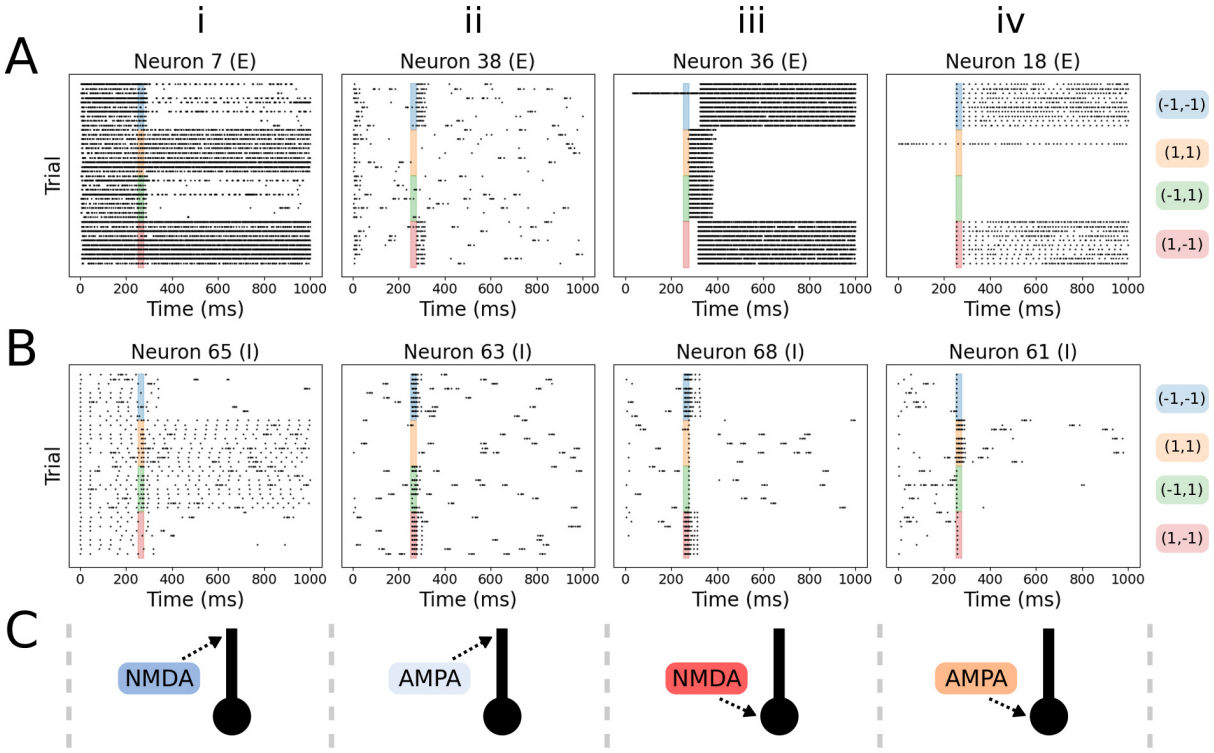

Figure 4.5: **Dendritic NMDA input triggers sustained persistent firing. A**: Raster plots for individual excitatory neurons during the working memory task. Colored bars indicate the task condition (10 trials per condition). As shown, the successfully trained networks with $\texttt{NMDA}_{\texttt{dend}}$ (i), $\texttt{NMDA}_{\texttt{soma}}$ (iii), and $\texttt{AMPA}_{\texttt{soma}}$ (iv) all exhibit condition-dependent spiking activity. In the network with $\texttt{AMPA}_{\texttt{soma}}$ (ii), which was not successfully trained to perform in the working memory task, the exemplar excitatory neuron exhibits a condition-dependent increase in firing rate when the cue input is active (increased for blue/red, no increase for orange/green). However, the change in activity does not persist past the cue stimulation period (250-275 ms). **B:** Raster plots for individual inhibitory neurons. As shown, inhibitory neurons in successfully trained networks (i, iii, and iv) exhibit condition-specific changes in activity that are sustained after the cue stimulation period. However, in the unsuccessfully trained network with $\texttt{AMPA}_{\texttt{soma}}$ (ii), the inhibitory neuron only shows spiking in response to background noise that does not depend on the task condition. **C**: Schematic of the cue synapse identity and location for each corresponding column.

In the successfully trained networks ($\texttt{NMDA}_{\texttt{dend}}$, $\texttt{NMDA}_{\texttt{soma}}$, and $\texttt{AMPA}_{\texttt{soma}}$), it is clear that the spike rasters demonstrate persistent spiking with distinct responses for each of the trial conditions that lasts to the end of the simulation (Figure 4.3B and Figure 4.4B and D). While the spike rasters for $\texttt{NMDA}_{\texttt{dend}}$ exhibits elevated activity across all neurons in the network (Figure 4.3B), the cue inputs are successfully able to produce lasting changes in both excitatory and inhibitory neuron activity, with distinct clusters entering

an "on" state in the post-stimulus period. In contrast, spike rasters for $\texttt{NMDA}_{\texttt{soma}}$ and $\texttt{AMPA}_{\texttt{soma}}$ exhibit less activity across the entire network, in favor of smaller clusters of excitatory neurons that are strongly activated in response to the cue inputs (Figure 4.4B and D). Since all extrinsic input variants are optimized with the same parameter ranges for local connectivity and biophysics, the elevation in baseline activity for $\texttt{NMDA}_{\texttt{dend}}$ and $\texttt{AMPA}_{\texttt{dend}}$ suggests that the cell and network adaptations necessary for propagating inputs from the dendrites to the soma make the neurons more sensitive to the noisy background inputs, leading to higher spiking at baseline. For example, the changes in biophysics necessary for dendritic spikes may make the neuron more excitable, whereas the biophysics of $\texttt{NMDA}_{\texttt{soma}}$ and $\texttt{AMPA}_{\texttt{soma}}$ are able to suppress responses to noisy inputs, and still produce persistent spiking in response to strong and synchronous cue inputs.

Figure 4.5 highlights the diversity of single neuron responses across the trained networks. While the spike raster of $\texttt{NMDA}_{\texttt{dend}}$ in Figure 4.3B clearly demonstrates cue neurons triggering persistent spiking, Figure 4.5A(i) and B(i) reveals that persistent firing in excitatory and inhibitory neurons can be turned off by the cue inputs. This is an important observation that is similar to modeling work in Papoutsi et al. (2014) which also describes a biophysical mechanism in which excitatory inputs can terminate persistent firing. One of the open questions in working memory research is how/why the prefrontal cortex maintains the same overall level of spiking activity when comparing rest to working memory periods (Wang (2021)). The ability of $\texttt{NMDA}_{\texttt{dend}}$ to turn off persistent spiking in specific neurons offers a possible explanation, such that different sets of neurons are switched on/off under different task conditions, but the overall number of "on" neurons remains the same.

Across the successfully trained networks, inhibitory neurons exhibit low firing rates relative to excitatory neurons (Figure 4.3B and Figure 4.4B and D, red dots), except for the initial burst of spikes time-locked to the cue input. In these network-level raster plots, it is not visually apparent if the inhibitory neurons exhibit task condition specific

changes in firing activity. However, visualizing spike rasters at the single neuron level demonstrates that the inhibitory neurons still do exhibit lasting changes in spiking that are specific to the task condition (Figure 4.5B(i, ii, and iv)).

4.4.3 Both intrinsic bursting and network connections contribute to working memory performance

As described previously, one reasonable hypothesis on why NMDA receptors are useful for working memory is that they are able to trigger persistent spiking activity in single neurons that is maintained during the memory period (Wang (2021); Wang et al. (2013)). Due to the slow time constants of NMDA receptors, extrinsic inputs equipped with NMDA are capable of producing sustained depolarization in the target neurons, with a burst of spiking activity even in the absence of inputs from the local network. Therefore, a simple solution to the working memory task would be for cue evoked extrinsic inputs to act like an "on" or "off" switch, where the input triggers a long burst of spiking in target neurons. If cell ion channels are set up such that the burst does not terminate, then this would be a situation where single neuron biophysics (and not network-level interactions) are sufficient for persistent spiking activity underlying working memory. Further, even if the target neuron is unable to maintain persistent spiking independently, a long burst of spikes could more effectively excite neurons in the local network, and reciprocally excite target neurons to produce persistent spiking.

To examine this possibility further, we assessed if the variants in task-relevant extrinsic inputs (and their optimized intrinsic biophysics) produced distinct burst responses in isolated (**disconnected**) excitatory neurons (Figure 4.6A). We further asked if the burst responses of isolated neurons were predictive of task performance (i.e. task errors quantified in Figure 4.2B).

Here, we removed the local connectivity to distinguish how much of the persistent spiking observed in the connected network in Figure 4.5 was due to inputs from the local

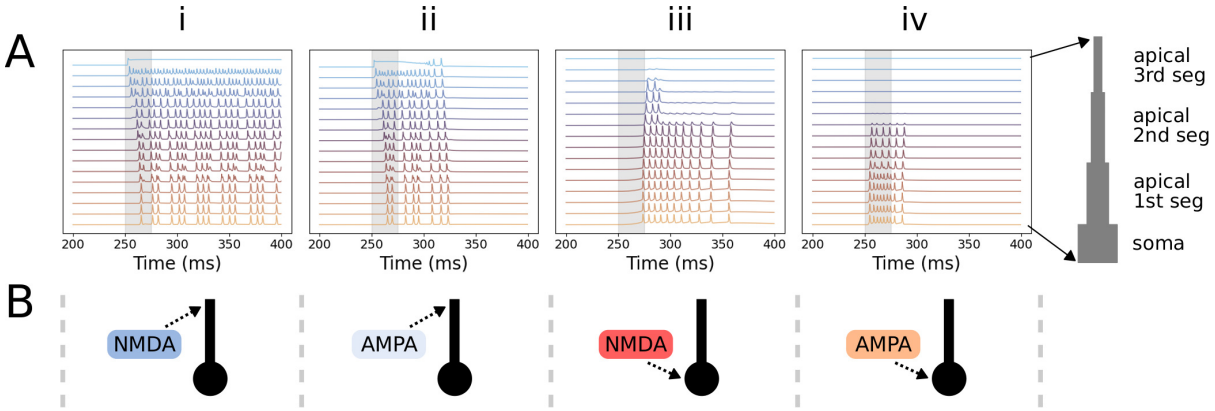

Figure 4.6: **Extrinsic input variants generate diverse lengths of spike bursts.** **A**: Membrane potentials of excitatory neurons were recorded in every compartment. Panels indicate the extrinsic input variant simulated, including: $\texttt{NMDA}_{\texttt{dend}}$ (i), $\texttt{AMPA}_{\texttt{dend}}$ (ii), $\texttt{NMDA}_{\texttt{soma}}$ (iii), $\texttt{AMPA}_{\texttt{soma}}$ (iv). The simulations were run with all recurrent connections removed, such that the only synaptic input comes from cue neurons. The 25 ms cue stimulus period is plotted in grey. Panel iv includes a schematic which maps membrane potential traces to each compartment. As shown, all extrinsic input variants initiate a burst of spikes that starts in the cue period (grey bar), and continues for varying amounts of time beyond the end of the cue period. Inputs to the dendrite generate bursts of spikes that propagate via dendritic spikes down to the soma (panel i and ii). Inputs to the soma also generate bursts of spikes, but they fail to propagate to the top of the dendrite (iii and iv). **B**: Schematic of the extrinsic input variant for each corresponding column.

network vs a self-generated burst of spiking due to the cell's own biophysics. During these simulations, the membrane potential was recorded in every compartment of excitatory neurons. We defined a burst as spiking activity that continues autonomously after the cessation of cue inputs. As shown in Figure 4.6A, the $\texttt{NMDA}_{\texttt{dend}}$ (panel i) and $\texttt{NMDA}_{\texttt{soma}}$ (panel iii) excitatory neurons produce robust bursts of spiking lasting $>$ 100 ms, which is substantially longer than the 25 ms cue stimulus period (grey bar). In $\texttt{NMDA}_{\texttt{dend}}$ (panel i) the spiking is initiated in the dendrite by a sustained depolarization of the 3rd apical segment (top-most voltage trace), with dendritic spikes propagating down the apical dendrite to trigger somatic spiking (bottom-most trace). $\texttt{NMDA}_{\texttt{soma}}$ (panel iii) exhibits back propagating action potentials originating from the soma, but they do not completely ascend to the top of the apical dendrite.

Despite the fast time constants of AMPA receptors, extrinsic AMPA inputs to the dendrite and soma were also able to generate single neuron spike bursts (Figure 4.6A(ii

and iv)). In contrast to the burst responses triggered by NMDA inputs, $\texttt{AMPA}_{\texttt{dend}}$ (panel ii) and $\texttt{AMPA}_{\texttt{soma}}$ (panel iv) exhibit substantially shorter bursts of ~ 40 ms and ~ 60 ms respectively. Interestingly, while $\texttt{AMPA}_{\texttt{soma}}$ produced the shortest burst length of ~ 40 ms (Figure 4.6A(iv)), it is clear that when simulated in a connected network, the neuron can produce sustained firing in response to cue inputs (Figure 4.5A(iv)). This suggests that the ability of $\texttt{AMPA}_{\texttt{soma}}$ to maintain unique representations of cue information highly depends on local recurrent connections in the network, and cannot be solely explained by the bursting properties of individual neurons. Further, despite $\texttt{AMPA}_{\texttt{dend}}$ having the same recurrent connections and a similar burst response to cue inputs (Figure 4.6A(ii)), condition-specific persistent firing does not arise.

Overall, these results demonstrate that the variants of task-relevant extrinsic inputs do indeed produce different bursting activity in isolated neurons, with NMDA receptors producing longer bursts when synapses are located either at the dendrite or soma. Further, despite dendritic AMPA receptors being capable of generating bursting activity that persists beyond the cue stimulus period, the $\texttt{AMPA}_{\texttt{dend}}$ network was unable to learn the working memory task. These results suggest that single-neuron firing properties (i.e. bursting) are not entirely predictive of persistent-firing activity that maintain task-relevant information. In the context of scaling biophysical network models to solve more difficult cognitive tasks, these results indicate that researchers should not necessarily begin by optimizing single neuron biophysics for desired response patterns, but instead should optimize cell and network level properties jointly.

4.4.4 Dendritic AMPA receptors are insufficient for establishing stable attractor dynamics

An increasingly popular way to analyze the dynamics of a simulated neural system is to study how the computations correspond to trajectories that evolve upon a low-dimensional neural manifold (Gallego et al. (2017); Langdon and Engel (2025)). One

common dynamical systems-based theory that seeks to explain working memory posits that neural networks flexibly create fixed point attractors which pull neural trajectories along manifolds towards their basin of attraction (Compte (2006)). Memories correspond to neural trajectories that remain in the basin of attraction for a sustained period of time. Figure 4.7B provides a schematic relevant to the working memory task in this study, and shows how basins of attraction corresponding to each task condition occupy distinct regions of the state space. Under this framework, the ability of a network to perform a working memory task relates to its ability to establish temporally stable basins of attraction that are separated in the state space.

We first asked if the location and time constants of extrinsic synapses impacted the complexity (i.e. dimensionality) of network-level neural dynamics by quantifying its dimensionality with principal component analysis (PCA). In the context of this study, dimensionality is a measure of describing the complexity of the manifold that neural trajectories traverse during the working memory task. Numerous studies have suggested that dimensionality reflects distinct computational processes (Jazayeri and Ostojic (2021); Altan et al. (2021); Cunningham and Yu (2014)). Given the striking differences in task performance and network-level spiking across extrinsic input variants (Figure 4.3 and Figure 4.4), we reasoned that the dimensionality may explain these differences with respect to the complexity of their associated neural manifolds.

Explained variance of the principal components (PCs) was used to estimate the dimensionality of neural activity. Figure 4.7A plots the variance explained for all 4 variants of extrinsic inputs. Despite the failure of $\texttt{AMPA}_{\texttt{dend}}$ to perform the working memory task (Figure 4.2B and Figure 4.3C), the variance explained for each PC is not a large outlier, and instead fits between the curves of $\texttt{NMDA}_{\texttt{soma}}$ and $\texttt{AMPA}_{\texttt{soma}}$. There is a general trend for AMPA inputs exhibiting slightly higher dimensionality, with $\texttt{AMPA}_{\texttt{dend}}$ and $\texttt{AMPA}_{\texttt{soma}}$ reaching 90% variance explained with 7 and 8 PCs respectively, in contrast to $\texttt{NMDA}_{\texttt{dend}}$ and $\texttt{NMDA}_{\texttt{soma}}$ with 4 and 6 PCs respectively. In summary we find that PCA-

estimated dimensionality does not directly relate to performance on the working memory task, but instead seems to reflect activity produced by networks trained with either NMDA or AMPA for extrinsic inputs.

Previous researchers have shown that in vivo, the dimensionality of neural activity is strongly modulated by the behavioral task considered, and consequently the extrinsic inputs driving activity in the recorded neural population (Badre et al. (2021); Cunningham and Yu (2014)). Here, we are analyzing the dimensionality of neural activity driven by 4 distinct cue inputs (representing a 2-dimensional target signal) in addition to noise. While the minimal dimensionality of neural activity needed to solve the task is 2 dimensions (or even 1 dimension if the 4 basins of attraction are mapped to different positions on a single axis), it is clear across all extrinsic input variants that the neural activity is explained by substantially more than 2 PCs (the lowest dimensionality was $\texttt{NMDA}_{\texttt{dend}}$ with 4 PCs explaining 90% of the variance), potentially suggesting that more complex activity patterns are necessary to solve the task in the presence of background noise.

We then asked if the extrinsic input variants impacted 1) the creation and strength of multiple basins of attraction, and 2) the path of a neural trajectory towards these basins of attraction. A "strong" basin of attraction is characterized by trajectories that exhibit low-variance clusters concentrated around the basins of attraction. It is also desirable for these basins of attraction to be spatially separated, as basins that are overlapping would prevent the network from distinguishing different cue stimuli. As such, we developed a simple measure called "attractor strength coefficient" (ASC), which is calculated as $ASC = \bar{\sigma}_{\text{intra}}/\bar{\delta}_{\text{inter}}$ where $\bar{\sigma}_{\text{intra}}$ is the average intra-cluster standard deviation in PCA space for each task condition, and $\bar{\delta}_{\text{inter}}$ is the average distance between the median coordinates of all clusters. ASC values are bounded on the range $(0, \infty)$, where values closer to zero indicate the desirable properties of low intra-cluster variance, and high distances between the clusters. ASC values reported here are for trajectories limited to PC1-PC3 to match what is shown visually in Figure 4.7D, however we found the results

hold when ASC was calculated on PC1-PC8 (8 components explains > 90 % variance for all network variants).

Figure 4.7C plots neural trajectories over time for all 4 extrinsic input variants (panels i-iv). The entire 1 second simulation is shown, with the beginning marked by a black star, and the end marked by a colored circle corresponding to each task condition. Cue inputs with $\texttt{NMDA}_{\texttt{dend}}$ (panel i, ASC=0.066) and $\texttt{NMDA}_{\texttt{soma}}$ (panel iii, ASC=0.056) produced a clear separation in trajectories at the start of the trial, and quickly settled into distinct regions of the state space indicating the presence of unique basins of attraction. The final positions of the trajectories produced by $\texttt{NMDA}_{\texttt{soma}}$ (panel iii) are generally more separated than $\texttt{NMDA}_{\texttt{dend}}$ (panel i), however trajectories for both exhibit tight clustering around their basins of attraction. In contrast, $\texttt{AMPA}_{\texttt{soma}}$ (panel iv, ASC=0.120) produces trajectories that are clearly separable, however they visibly exhibit high variance around their basin of attraction relative to the distance between the start and end of the simulation. One possible explanation for the high variance is that persistent firing is being maintained by recurrent connections between neurons, rather than persistent bursting of individual neurons as suggested by Figure 4.5A(iv) and B(iv). Inspecting the learned local connectivity further supports this hypothesis, as $\texttt{AMPA}_{\texttt{soma}}$ exhibits higher weights for all excitatory connections compared to $\texttt{NMDA}_{\texttt{dend}}$ and $\texttt{NMDA}_{\texttt{soma}}$ (Figure 4.2C(i-iii) orange histogram). A potential consequence of these weaker (high variance) basins of attraction, is sensitivity to perturbation, as sufficient noise could push the neural trajectory into a different basin causing the network to "forget" information about the cue stimulus.

The trajectories of the unsuccessfully trained $\texttt{AMPA}_{\texttt{dend}}$ (ASC=5.527) network collapse into an identical basin of attraction across all task conditions ((Figure 4.7C(ii))). Interestingly, the trajectories are clearly distinguishable at the start of the simulation, indicating that the failure of $\texttt{AMPA}_{\texttt{dend}}$ to perform the working memory task is not related to its ability to produce a unique stimulus response, but instead reflects its inability to create unique and stable attractors.

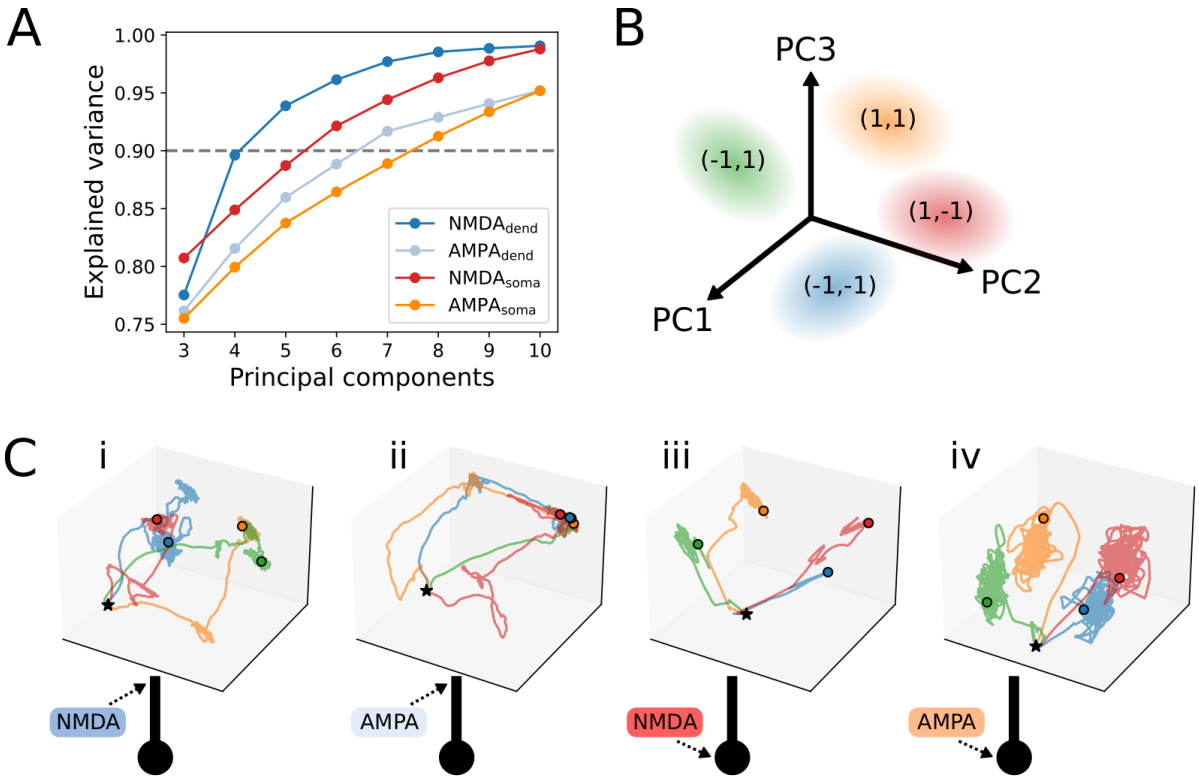

Figure 4.7: **Synaptic receptor identity and location alters attractor dynamics.** Network activity was analyzed with principal component analysis (PCA) to assess the impact of the different types of cue inputs on network dynamics during the working memory task. PCA was only applied to the firing rates of excitatory neurons. **A:** Cumulative variance explained for simulations with dendritic $\texttt{NMDA}_{\texttt{dend}}$ and $\texttt{AMPA}_{\texttt{dend}}$, and somatic $\texttt{NMDA}_{\texttt{soma}}$ and $\texttt{AMPA}_{\texttt{soma}}$ cue inputs. Networks with NMDA exhibit lower dimensionality compared to networks with AMPA inputs, as $\texttt{NMDA}_{\texttt{dend}}$ and $\texttt{NMDA}_{\texttt{soma}}$ reach 90% explained variance (dashed grey line) with fewer components than $\texttt{AMPA}_{\texttt{dend}}$ and $\texttt{AMPA}_{\texttt{soma}}$. **B**: Schematic of PCA figures shown in panel C. Colored ovals represent basins of attraction which pull neural trajectories into different regions of the state space depending on the task condition. **C**: Neural trajectories for PC1-PC3 are shown for the 4 extrinsic input variants. Each trajectory corresponds to the entire 1 second simulation. The beginning of the simulation is marked by a black star, and the end of the simulation is marked by a colored dot. The color of the trajectory corresponds to the different trial conditions.

Overall, analysis of PCA trajectories demonstrated that location and time constants of extrinsic synapses (and associated changes to intrinsic parameters), dramatically alter neural trajectories produced during the working memory task. A potentially important observation is that NMDA synapses at either the dendrite or soma produce strong basins of attraction (Figure 4.7C(i and iii)), which is supported by their relatively small ASC values (0.066 and 0.056 for $\texttt{NMDA}_{\texttt{dend}}$ and $\texttt{NMDA}_{\texttt{soma}}$ compared to 0.120 and 5.526 for $\texttt{AMPA}_{\texttt{dend}}$ and

$\texttt{AMPA}_{\texttt{soma}}$). This result is not apparent from inspection of network-level spiking, as spike rasters for $\texttt{NMDA}_{\texttt{dend}}$ (Figure 4.3B) are highly distinct from those of $\texttt{NMDA}_{\texttt{soma}}$ (Figure 4.4B), with elevated firing activity throughout the simulation. In contrast, despite the spike rasters of $\texttt{NMDA}_{\texttt{soma}}$ appearing more similar to $\texttt{AMPA}_{\texttt{soma}}$ with relatively low baseline spiking activity and sharply defined clusters (Figure 4.4B and D), analysis of the neural trajectories potentially suggests the attractors produced by $\texttt{AMPA}_{\texttt{soma}}$ are less stable than those produced by $\texttt{NMDA}_{\texttt{soma}}$ (Figure 4.7C(iii and iv)).

We note that some of these results may depend upon the choice of using PCA for dimensionality reduction. An important flaw of analyzing neural trajectories with PCA is that the analysis does not directly account for noise or time-dependencies (dynamics), and instead treats every time point independently with the goal of capturing the covariance structure in the data (Cunningham and Yu (2014); Pellegrino et al. (2024)). While these oversimplifications make PCA an imperfect fit for estimating dimensionality and state-space trajectories in neural data, PCA remains one of the most popular choices for dimensionality reduction of neuroscience, allowing this work to be more directly compared to existing publications. Further, given striking differences in PCA-estimated neural trajectories across extrinsic input variants, it is unlikely that our conclusions would change given an alternative dimensionality reduction technique.

For the goal of guiding the construction/training of biophysical network models, these results suggest attractor dynamics measured by ASC could be a good indicator of network-level activity that should be promoted during training, potentially by direct incorporation into the loss function. In contrast, PCA-estimated dimensionality did not seem to directly relate to task performance given the similarity between $\texttt{AMPA}_{\texttt{dend}}$ and $\texttt{AMPA}_{\texttt{soma}}$ (Figure 4.7A), and instead indicated a general trend of NMDA network variants exhibiting lower dimensional neural activity.

4.4.5 Magnesium block supplements slow NMDA kinetics to produce stable attractors

The previous sections have demonstrated that NMDA inputs can produce stable attractors across the two different synapse locations (dendrite vs soma), whereas AMPA receptors failed to produce stable attractors through dendritic inputs. While the slow time constants of NMDA receptors are an important feature that could contribute to this difference, they also possess a magnesium ion block, which is only removed with depolarization of the postsynaptic cell (Destexhe et al. (1998)). This magnesium ion block is theorized to function as a coincidence detector, however its contribution to attractor dynamics is unclear.

To further identify what dynamic properties of AMPA and NMDA receptors contribute to working memory task performance, we swapped the rate constants of the NMDA and AMPA receptors. This produced two modified receptors including a fast "NMDA-like" receptor ($\texttt{NMDA}^{\texttt{fast}}_{\texttt{dend}}$), and a slow "AMPA-like" receptor ($\texttt{AMPA}^{\texttt{slow}}_{\texttt{dend}}$). Importantly, $\texttt{NMDA}^{\texttt{fast}}_{\texttt{dend}}$ possessed fast kinetics with the characteristic magnesium block, whereas $\texttt{AMPA}^{\texttt{slow}}_{\texttt{dend}}$ possessed slow kinetics without the magnesium block. Figure 4.8A (pink) demonstrates that $\texttt{NMDA}^{\texttt{fast}}_{\texttt{dend}}$ was unsuccessfully trained on the working memory task, suggesting that the magnesium block alone does not convey an advantage for performing the task. In contrast the slow kinetics of $\texttt{AMPA}^{\texttt{slow}}_{\texttt{dend}}$ did lead to an improvement of task performance (Figure 4.8B, purple), however the network did not converge to the target error threshold of 1.0. Further, neural trajectory analysis of $\texttt{AMPA}^{\texttt{slow}}_{\texttt{dend}}$ (Figure 4.8E) reveals no distinguishable attractor dynamics, despite the reduction in error. This result suggests that slow receptor kinetics combined with the magnesium block enable dendritic inputs to produce stable attractor dynamics, as shown in the unmodified $\texttt{NMDA}_{\texttt{dend}}$ (Figure 4.8A (blue line) and Figure 4.7C).

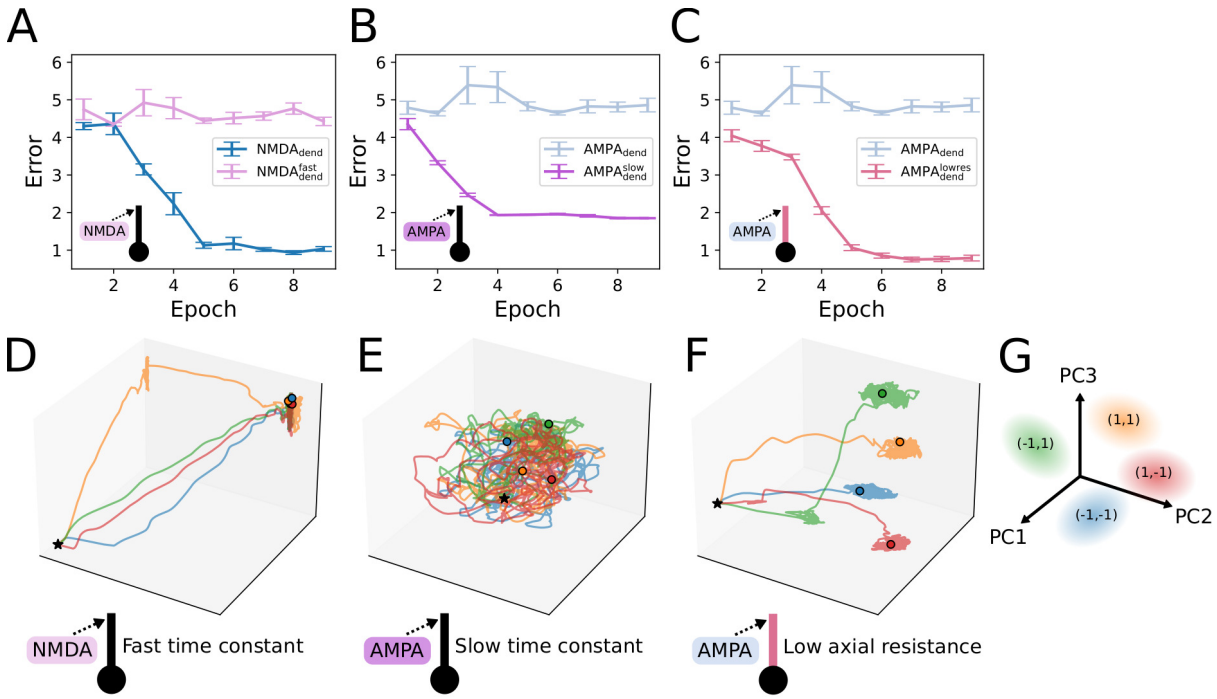

Figure 4.8: **Dendritic inputs with slow rate constant or magnesium block alone do not promote attractor dynamics. A:** NMDA receptors were modified by setting their rate constants equal to the default AMPA rate constants. This creates a fast "NMDA-like" receptor ($\texttt{NMDA}^{\texttt{fast}}_{\texttt{dend}}$, pink) that possesses a magnesium block. Compared to the successfully trained $\texttt{NMDA}_{\texttt{dend}}$ (blue), the modified $\texttt{NMDA}^{\texttt{fast}}_{\texttt{dend}}$ was unable to learn the working memory task. **B:** Similar modification as **A** to create a slow "AMPA-like" receptor (purple). The slow receptor kinetics did permit the network to reduce error on the working memory task, but it failed to reach the target error threshold of 1.0. The unsuccessfully trained $\texttt{AMPA}_{\texttt{dend}}$ (light blue) is overlaid. **C** $\texttt{AMPA}_{\texttt{dend}}$ (light blue) was retested in a network where the axial resistivity was reduced from 300 Ωcm to 10 Ωcm ($\texttt{AMPA}^{\texttt{lowres}}_{\texttt{dend}}$, violet red). This change allowed the network to be successfully trained with dendritic AMPA receptors and reach the target error threshold. **D**: Neural trajectories associated with $\texttt{NMDA}^{\texttt{fast}}_{\texttt{dend}}$ (panel A) are shown. Black stars indicate the start of the cue stimulus, colored lines indicate each task condition, and colored dots indicate the end of the 1 second simulation (same as Figure 4.7C). As shown, $\texttt{NMDA}^{\texttt{fast}}_{\texttt{dend}}$ fails to produce a stable attractor, such that all trajectories collapse to the same region of the state space. **E:** Neural trajectories for $\texttt{AMPA}^{\texttt{slow}}_{\texttt{dend}}$ (panel B) indicate that no stable attractor is produced, and instead all trajectories overlap. **F**: Neural trajectories for $\texttt{AMPA}^{\texttt{lowres}}_{\texttt{dend}}$ (panel C) indicate the presence of distinct and stable attractor dynamics. **G**: Legend for panels D-F labeling principal components 1-3 (axes) and task conditions (colored ovals)
.

4.4.6 Dendritic AMPA receptors fail to produce attractor dynamics due to requirement to produce dendritic spikes

As a final investigation, we sought to identify which properties of the dendrite were responsible for the inability of $\texttt{AMPA}_{\texttt{soma}}$to perform the working memory task. The most likely candidate is the axial resistivity, which leads to the attenuation of dendritic inputs in a purely passive dendrite. As observed in Figure 4.6A(i-ii), cue inputs arriving at the 3rd apical segment are propagated to the soma through dendritic action potentials. One potential issue with this constraint is that active dendrites conducive to action potentials raise the overall excitability of the neurons (e.g. increased spiking in Figure 4.3 compared in Figure 4.4). We hypothesized that $\texttt{AMPA}_{\texttt{soma}}$ inputs are unable to overcome this increase in activity to produce stable attractor dynamics. To test if the requirement for active dendrites was indeed harming performance of dendritic AMPA inputs, we decreased the axial resistivity of the apical dendrite from 300 Ωcm to 10 Ωcm ($\texttt{AMPA}^{\texttt{lowres}}_{\texttt{dend}}$). As shown in Figure 4.8C, this change allowed $\texttt{AMPA}^{\texttt{lowres}}_{\texttt{dend}}$ (violet red line) to rapidly reach the target error threshold, and produce distinct and persistent attractors for each task condition (Figure 4.8F). While further analysis will be necessary to narrow down the exact impact of active dendrites on the overall excitability of a neuron, the results clearly demonstrate that the changes in intrinsic biophysics that yield active dendrites can be harmful to task performance when task information is relayed through dendritic AMPA receptors. The unique properties of NMDA receptors (i.e. slow rate constants and magnesium block) are both essential to overcoming the associated changes in dendritic biophysics, and produce stable attractor dynamics conducive to working memory.

4.5 Discussion

4.5.1 Summary

This study introduces biophysical reservoir computing (BRC) as a framework to discover mechanistic insights that guide the construction and training of biophysically-detailed network models to solve cognitive tasks. To address the challenges of training biophysically-detailed network models, we developed a novel optimization algorithm, named "neural flow evolution", which combines evolutionary algorithms with neural density estimators. Neural flow evolution benefits from the robust parameter search capabilities afforded by evolutionary algorithms, and the ability of neural density estimators to represent complex, high-dimensional distributions.

Training the BRC network to solve a simplified working memory task led to the following mechanistic insights with respect to extrinsic cue inputs: 1) NMDA receptors are the most effective for working memory for both somatic and dendritic synapse locations, exhibiting improved training efficiency and attractor dynamics, 2) dendritic AMPA receptors are insufficient for producing persistent spiking and associated attractor dynamics that are necessary for working memory, and 3) the combination of NMDA receptors' unique properties, specifically their slow rate constants and magnesium block, are both required for working memory when extrinsic inputs arrive at the dendrite.

We additionally "opened the black box" (Sussillo and Barak (2012)) of the BRC, and characterized how the trained networks solved the working memory task. Since the BRC is composed of biophysically-detailed neurons, we were able to use analysis techniques common to systems neuroscience for this goal. We revealed that while NMDA receptors at the dendrite and soma are equally efficient at solving the working memory task, however, networks which delivered extrinsic cue inputs through the dendrite exhibited higher spiking activity at baseline. This difference suggests that the dendritic ion channel configurations necessary to relay cue information to the soma makes the neurons more excitable and

sensitive to noisy inputs, and may explain why networks with AMPA inputs to the dendrite failed to learn the task. The spiking analysis also suggested that a synchronous burst of inhibitory neuron spiking was essential for resetting network dynamics, and subsequently activating persistent spiking in a new set of neurons which encoded the memory. We additionally revealed that single neuron burst properties were not predictive of working memory task performance, indicating that a combination of intrinsic bursting and local network connectivity underlie successfully solving the working memory task. Lastly, by examining the learned synaptic weights of the recurrent local connectivity, we revealed that the different types of extrinsic cue inputs produce highly distinct network configurations, suggesting an important role of network-level interactions in solving the working memory task.

These results are in agreement with neuroscience literature, indicating that the observed importance of NMDA receptors for working memory is not a byproduct of biological constraints, but instead reflects a real and generalizable functional benefit. Since the learned cell and network configurations in the BRC framework are not forced to match real brain activity, these results suggest that the importance of NMDA for working memory is a general computational principle that is applicable to both real and artificial neural systems.

4.5.2 Related work and future directions

Jaxley

This study directly builds upon the software package Jaxley, which was created to directly address the challenges of training biophysical neural models by solving the complex differential equation representing biophysical models using advanced numerical computing software. These numerical computing platforms are commonly used by artificial intelligence (AI) researchers to develop deep neural networks. Specifically, Jaxley employs Jax (Bradbury et al. (2021)), a high-performance numerical computing platform that

permits auto-differentiation for backpropagation, code compilation, and parallelized computations that can be accelerated with GPU computing (Deistler et al. (2024)). It is important to note that Jaxley uses standard techniques to simulate biophysical models, namely differential equation solvers like those in NEURON (Hines and Carnevale (1997)). However, implementing these models and solvers in Jax provides the software and hardware advantages that have historically been reserved for the development and application of deep neural networks, advantages which have significantly contributed to the rapid growth of deep learning in recent years.

In Deistler et al. (2024), they train a biophysically detailed model to perform a working memory task using gradient descent with backpropagation. There are several major differences that distinguish our work from the example in this publication which we outline below:

The first major difference is that the composition of our network model was focused on neurons with active dendrites that transmitted extrinsic inputs via dendritic spikes, a feature that is suggested to convey significant computational benefits to cell and network-level function (see "Experimental studies on the computational benefits of dendrites" below). Specifically, our network model contains distinct populations of excitatory and inhibitory neurons. The excitatory neurons possess an elongated apical dendrite, a characteristic feature of cortical pyramidal neurons often implicated in top-down (feedback) information processing (Harris and Shepherd (2015); Schuman et al. (2021); Fişek et al. (2023)). Our network model additionally uses AMPA, NMDA, and GABAa synapses for local cell-cell connections, as well AMPA and NMDA synapses for extrinsic task-relevant inputs. Mathematically, these synapses are not differentiable, preventing the use of backpropagation for gradient-based optimization of network parameters. The non-differentiability of the synapses in our model is an important limitation of our model as it prevents the use of gradient descent with backpropagation, but it permits an investigation into more biologically realistic synapses with distinct computational roles as

suggested by neuroscience literature. Instead, the network in Deistler et al. (2024) uses a differentiable synapse model for recurrent local connections that permits gradient-based optimization. The weights of these synapses were optimized such that they could produce either excitatory or inhibitory conductances, whereas our model enforces a strict separation between excitatory and inhibitory neurons. The extrinsic cue inputs in Deistler et al. (2024) were also distinct from our model, as they employed direct current injection to the soma of neurons, whereas our network model used a dedicated population of cue neurons with a gaussian receptive field that transmitted cue information with either AMPA or NMDA synapses.

The second major difference is the working memory task used. In our study we focus on a highly simplified working memory task in favor of thoroughly characterizing individual biophysical details. Deistler et al. (2024) trains their model on significantly more challenging tasks (delayed match to sample and evidence integration), to test hypotheses on different forms of neural coding.

The third major difference is our use of reservoir computing, where the parameters for individual neurons and synapses are shared across the entire network. While this potentially limits the computational capabilities of the biophysical network model, it provides a strong constraint for identifying generically useful biophysical properties. For example, our study revealed that NMDA receptors are generically useful for producing attractor dynamics, however their properties make them uniquely well-suited for transmitting information through dendrites.

It is important to emphasize that Jaxley's ability to perform gradient-based optimization in network models with biophysically detailed neurons containing non-linear dynamics is a major advance offered by the platform, and has the potential to massively increase the complexity of tasks that biophysical models can be trained to perform. Our choice to use AMPA and NMDA synapses for the extrinsic cue input and multicompartment active pyramidal neuron dendrites was motivated by having a closer connection to neuroscience

literature. An important future investigation will be to implement differentiable synapses in the BRC framework with equivalent time constants for AMPA and NMDA synapses, and a differentiable implementation of the magnesium block in the NMDA synapses. Employing differentiable synapses would allow a further investigation into the importance of NMDA for working memory as it would enable the use of gradient-based optimization, potentially expanding the complexity of working memory tasks that can be investigated. A complementary future direction will be to adapt surrogate gradient techniques from spiking neural networks (Neftci et al. (2019)) to perform gradient-based training in Jaxley with non-differentiable synapse models. Alternatively, FORCE training (Nicola and Clopath (2017); Sussillo and Abbott (2009)), another method popular for task-trained RNNs and SNNs, could be adapted using Jaxley, as it does not require tracking gradients backwards through time, but instead updates model parameters mid-simulation. While non-differentiable synapses prevent backpropagation through time, FORCE training could still work with multicompartment models as it only requires gradient calculations for the current time step (i.e. recursive least squares).

Task-trained neural models without dendrites

Prior to the introduction of Jaxley, researchers persisted in creating well-trained recurrent neural network (RNN) and spiking neural network (SNN) models of working memory using more simplified non-biophysical neuron elements. For example, researchers have developed methods to transfer learned synaptic weights of a trained RNN to a SNN (Kim and Sejnowski (2021)). Outside of working memory, researchers have successfully trained SNNs to perform a wide variety of behaviorally relevant cognitive tasks (Wang et al. (2023); Maes et al. (2020); Nicola and Clopath (2017)). While SNNs remain simplified relative to biophysical models, they represent a much more biologically plausible architecture compared to RNNs and modern deep learning systems.

An active area of research with RNNs is studying how a single network can be trained

to perform multiple tasks (Driscoll et al. (2024); Yang et al. (2019)). These studies have demonstrated that trained RNNs exhibit a form of compositionality, that is they learn dynamical motifs that are flexibly recombined to solve tasks with overlapping computational demands. One future direction of the BRC framework would be to study the emergence of compositionality in biophysical neural models trained to perform multiple tasks. A valuable point of comparison would be to train biophysical models and RNNs to perform the same array of behavioral tasks, and assess differences in how they compose dynamical motifs. Recently developed methods for comparing dynamical systems may prove instrumental for this goal (Ostrow et al. (2023); Guilhot et al. (2024)).

Biophysical neural models with dendrites

Task-training of biophysical neural models is a relatively recent development given the high computational cost of simulation and training. Instead, biophysical neural models have been more frequently used to model specific brain regions by highly constraining model parameters and activity patterns to experimental measurements. Nevertheless, there are several biophysical neural modeling studies worth mentioning due to their implications for working memory in networks with dendrites.

With respect to biophysical models with morphologically simplified dendrites, Gollo et al. (2009) showed how at the single-neuron level, active dendrites dramatically expanded the dynamic range of neuron's firing rate in response to synaptic inputs. In a separate study using networks of dendrite-equipped neurons, Papoutsi et al. (2014) demonstrated that dendritic non-nonlinearities reduced the number of interconnected neurons necessary for the flexible emergence and termination of persistent firing activity. The source of these benefits was largely linked to NMDA mediated dendritic spikes in layer 5 pyramidal neurons, and demonstrated a novel mechanism in which excitatory synaptic inputs to these neurons were able to terminate persistent activity. Building from these results, Stefanos et al. (2019) showed that large networks of dendrite-equipped biophysical neurons could

maintain multiple stable representations of extrinsic inputs, and that NMDA mediated persistent activity was crucial for the network to achieve multistable attractor dynamics. The key difference from our study and Stefanos et al. (2019) is that they highly constrain model parameters and activity to closely match experimental measurements of real cortical circuits. Accordingly, their model agrees with the result in our study that NMDA is required under biological constraints observed in the neocortex. Our use of the BRC framework reveals a more generalized result that NMDA is generically useful for working memory, even when local connectivity and cell biophysics are freely optimized with potentially non-biological network configurations.

Experimental studies on the computational benefits of dendrites

Unlike the neural modeling studies described previously, there are significantly fewer examples of experimental studies investigating the role of dendrites in working memory. However, numerous experimental studies have proposed a diverse set of alternative computational benefits that are unique to dendrites. These proposals include computational benefits such as compartmentalization (Polsky et al. (2004)), coincidence detection (Larkum et al. (1999); Xu et al. (2012)), modulation of sensory perception (Takahashi et al. (2016)), and learning (McBride et al. (2008)).

Despite the fact that all of these theories concern neurons in a highly interconnected network, the computational benefits of these theories primarily concern functions at the level of single neurons (see Jones and Kording (2021)). Experimental studies examining the relationship between dendrites and their impact on network level dynamics are significantly fewer in number due to the technical challenges of performing such experiments (Poirazi and Papoutsi (2020)). In future work, the BRC framework may prove to be a highly useful tool for isolating the computational benefits of dendrites beyond the simple working memory task presented here, and propose new experiments that take into account the network-level functioning of dendrite-equipped neurons.

4.5.3 Limitations

There are several limitations to the present study that provide ample opportunity for follow up research.

The first limitation is the simple morphology used for the apical dendrite. As stated in the introduction, a primary motivation of this study was understanding which modeling choices made sense when expanding task-trained RNNs and SNNs to include dendrites. Our results revealed the dendritic NMDA receptors are uniquely suited for transmitting extrinsic cue information, and ultimately recruit distinct clusters of persistently firing neurons. However, we did not observe a unique benefit of transmitting cue information through dendrites, and instead demonstrated the dendritic AMPA receptors were insufficient to promote attractor dynamics. It is important to emphasize that these results do not suggest dendrites are not computationally useful for other tasks. Many of the theories of dendritic function emphasize highly elaborated dendritic arbors which serve to process incoming information separately from the soma. A valuable expansion of the BRC framework would be the inclusion of morphologically complex dendrites to directly test more sophisticated theories of dendritic function.

The second limitation is the simplicity of the cognitive task used. We focused on a basic working memory which required maintaining persistent representations of 4 distinct stimuli. In general, the cognitive tasks used in experimental settings are highly simplified with respect to the behavioral demands of the real world (Huk et al. (2018); Maselli et al. (2023); Krakauer et al. (2017)). While our simplified working memory task is not representative of the cognitive demands of naturalistic behavior, we argue that persistent representations are a fundamental building block required for a vast array of more complex functions. Further, the simplified working memory task was surprisingly sufficient to demonstrate the fundamental role of NMDA receptors in the context of persistent representations. It is highly possible that the network implemented will be insufficient for more complex cognitive tasks, which may require additional biophysical

properties, optimization strategies, and a significantly better understanding of how these computations are performed in real neural circuits.

The third limitation is the exclusive use of AMPA and GABAa receptors for recurrent local connectivity. This choice was made early in the study, as inclusion of recurrent NMDA synapses was found to produce runaway excitation, a problem that may reflect the relatively small number of neurons in the biophysical reservoir. However, experimental studies have demonstrated that NMDA and AMPA receptors are both involved in the onset and maintenance of persistent firing underlying cortical working memory (Vugt et al. (2020); Self et al. (2012)). Follow up studies will be necessary to elucidate the role of recurrent local NMDA connections, and if their inclusion can mitigate the issues we report here with signaling cue information through dendritic AMPA synapses.

The fourth limitation is the lack of heterogeneity in cell-type parameters of the network. It is well documented that cell heterogeneity plays a crucial role in the computational properties of biologically-plausible neural networks (Perez-Nieves et al. (2021); Gast et al. (2024); Lengler et al. (2013); Mejias and Longtin (2014)). However, in the context of working memory, Gast et al. (2024) demonstrates with SNNs that cell heterogeneity (e.g. varied spike thresholds) actually impairs the ability of networks to encode and maintain information about external stimuli. Cell heterogeneity was shown to increase the dimensionality of neural activity, making these networks more suitable as function generators that continuously transform inputs to a new target signal. A future direction will be to explore if this tradeoff between function generation and persistent representations holds in network models composed of biophysically detailed neurons, and if heterogeneity in dendritic parameters confer unique computational properties. One possibility is that the presence of connection heterogeneity (i.e. between cell connection probabilities) in biophysically detailed neurons with dendritic synapses provides similar benefits as cell heterogeneity in SNNs.

As described above, the decision to use evolutionary algorithms combined with neural

density estimation (see "Neural flow evolution" in Methods) was largely due to challenges we faced with existing optimization techniques. Previous work, including our own, has shown that neural density estimation is particularly effective for parameter estimation in biophysical models (Tolley et al. (2024); Gonçalves et al. (2020)). Traditionally this takes the form of likelihood-free Bayesian inference, where the estimator learns the posterior distribution $p(\theta \mid x)$ over model parameters θ conditioned on simulated outputs x. While these methods have been effective for fitting biophysical models to data, we found that conditional density estimation was numerically unstable in the setting of training the BRC to solve the working memory, with the majority of parameter samples falling outside the support of the prior distribution. Instead, we used neural density estimation in a simpler form to approximate an unconditional distribution $p(\theta)$. We note that previous studies have proposed integrating probability distributions with evolutionary algorithms as a method to generate offspring from the parent population (Kern et al. (2004)). To our knowledge, this is the first study to propose using neural density estimation for this purpose. Follow up work will be necessary to benchmark this approach against standard evolutionary algorithms, as well as traditional density estimation techniques like kernel density estimation.

4.6 Supplemental Data

Table 4.1: **Simulation parameter ranges for neural flow evolution**

Channel conductance ($\bar{g}$)	**Range**	**Transform**
E_{soma} Leak (S/cm^2)	(-9, -2)	exponential
E_{soma} Km (S/cm^2)	(-9, -2)	exponential
E_{soma} CaL (S/cm^2)	(-9, -2)	exponential
E_{soma} CaT (S/cm^2)	(-9, -2)	exponential
E_{dend} Leak (S/cm^2)	(-9, -2)	exponential
E_{dend} Km (S/cm^2)	(-9, -2)	exponential
E_{dend} CaL (S/cm^2)	(-9, -2)	exponential
E_{dend} CaT (S/cm^2)	(-9, -2)	exponential
I Leak (S/cm^2)	(-9, -2)	exponential
I Km (S/cm^2)	(-9, -2)	exponential
I CaL (S/cm^2)	(-9, -2)	exponential
I CaT (S/cm^2)	(-9, -2)	exponential

Connection probability	**Range**	**Transform**	**Synapse**
E→E_{soma}	(0.0, 0.6)	linear	AMPA
E→E_{dend}	(0.0, 0.6)	linear	AMPA
E→I	(0.0, 0.6)	linear	AMPA
I→E_{soma}	(0.0, 0.6)	linear	GABAa
I→E_{dend}	(0.0, 0.6)	linear	GABAa
I→I_{dend}	(0.0, 0.6)	linear	AMPA

Connection strength ($\bar{g}$)	**Range**	**Transform**	**Synapse**
E→E_{soma} (μS)	(-4, 1)	exponential	AMPA
E→E_{dend} (μS)	(-4, 1)	exponential	AMPA
E→I (μS)	(-4, 1)	exponential	AMPA
I→E_{soma} (μS)	(-4, 1)	exponential	AMPA
I→E_{dend} (μS)	(-4, 1)	exponential	AMPA
I→I_{dend} (μS)	(-4, 1)	exponential	AMPA
Cue→E_{soma} (μS)	(-4, 1)	exponential	AMPA
Cue→E_{dend} (μS)	(-4, 1)	exponential	AMPA
Cue→E_{soma} (μS)	(-4, 1)	exponential	NMDA
Cue→E_{dend} (μS)	(-4, 1)	exponential	NMDA
Cue→I (μS)	(-4, 1)	exponential	AMPA

The prior support and transform function for the parameters of examples shown in the text. For neural density estimation, all parameters were transformed such that the minimum and maximum of the **Range** column was on the interval $[0, 1]$.

Acknowledgments

We would like to thank Katharina Duecker, Jason Ritt, and Christopher Moore for feedback on early versions of this manuscript.

References

Abbott, L. F. and Dayan, P. (2001). *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems*. Computational Neuroscience Series. MIT Press, Cambridge, MA, USA.

Altan, E., Solla, S. A., Miller, L. E., and Perreault, E. J. (2021). Estimating the dimensionality of the manifold underlying multi-electrode neural recordings. *PLoS Computational Biology*, 17(11):e1008591.

Arthur, B. J., Kim, C. M., Chen, S., Preibisch, S., and Darshan, R. (2023). A scalable implementation of the recursive least-squares algorithm for training spiking neural networks. *Frontiers in Neuroinformatics*, 17. Publisher: Frontiers.

Badre, D., Bhandari, A., Keglovits, H., and Kikumoto, A. (2021). The dimensionality of neural representations for control. *Current Opinion in Behavioral Sciences*, 38:20–28.

Bittner, K. C., Andrasfalvy, B. K., and Magee, J. C. (2012). Ion Channel Gradients in the Apical Tuft Region of CA1 Pyramidal Neurons. *PLOS ONE*, 7(10):e46652. Publisher: Public Library of Science.

Bradbury, J., Frostig, R., Hawkins, P., Johnson, M. J., Leary, C., Maclaurin, D., Necula, G., Paszke, A., VanderPlas, J., Wanderman-Milne, S., and Zhang, Q. (2021). JAX: Autograd and XLA. *Astrophysics Source Code Library*, page ascl:2111.002. ADS Bibcode: 2021ascl.soft11002B.

Compte, A. (2006). Computational and *in vitro* studies of persistent activity: Edging towards cellular and synaptic mechanisms of working memory. *Neuroscience*, 139(1):135–151.

Cunningham, J. P. and Yu, B. M. (2014). Dimensionality reduction for large-scale neural recordings. *Nature Neuroscience*, 17(11):1500–1509. Publisher: Nature Publishing Group.

Damicelli, F., Hilgetag, C. C., and Goulas, A. (2022). Brain connectivity meets reservoir computing. *PLOS Computational Biology*, 18(11):e1010639. Publisher: Public Library of Science.

Deistler, M., Kadhim, K. L., Pals, M., Beck, J., Huang, Z., Gloeckler, M., Lappalainen, J. K., Schröder, C., Berens, P., Gonçalves, P. J., and Macke, J. H. (2024). Differentiable simulation enables large-scale training of detailed biophysical models of neural dynamics. Pages: 2024.08.21.608979 Section: New Results.

D'Esposito, M. (2007). From cognitive to neural models of working memory. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 362(1481):761–772. Publisher: Royal Society.

Destexhe, A., Mainen, Z. F., and Sejnowski, T. J. (1998). Kinetic models of synaptic transmission. *Methods in neuronal modeling*, 2:1–25. Publisher: MIT press Cambridge, MA.

Driscoll, L. N., Shenoy, K., and Sussillo, D. (2024). Flexible multitask computation in recurrent networks utilizes shared dynamical motifs. *Nature Neuroscience*, 27(7):1349–1363. Publisher: Nature Publishing Group.

Erhan, D., Courville, A., Bengio, Y., and Vincent, P. (2010). Why does unsupervised pre-training help deep learning? pages 201–208. JMLR Workshop and Conference Proceedings.

Eshraghian, J. K., Ward, M., Neftci, E. O., Wang, X., Lenz, G., Dwivedi, G., Bennamoun, M., Jeong, D. S., and Lu, W. D. (2023). Training spiking neural networks using lessons from deep learning. *Proceedings of the IEEE*, 111(9):1016–1054. Publisher: IEEE.

Fişek, M., Herrmann, D., Egea-Weiss, A., Cloves, M., Bauer, L., Lee, T.-Y., Russell, L. E., and Häusser, M. (2023). Cortico-cortical feedback engages active dendrites in visual cortex. *Nature*, 617(7962):769–776. Publisher: Nature Publishing Group.

Gallego, J. A., Perich, M. G., Miller, L. E., and Solla, S. A. (2017). Neural Manifolds for the Control of Movement. *Neuron*, 94(5):978–984.

Gast, R., Solla, S. A., and Kennedy, A. (2024). Neural heterogeneity controls computations in spiking neural networks. *Proceedings of the National Academy of Sciences*, 121(3):e2311885121. Publisher: Proceedings of the National Academy of Sciences.

Gilhooly, K. J. (2004). Working memory and planning. In *The Cognitive Psychology of Planning*. Psychology Press. Num Pages: 7.

Gollo, L. L., Kinouchi, O., and Copelli, M. (2009). Active Dendrites Enhance Neuronal Dynamic Range. *PLOS Computational Biology*, 5(6):e1000402. Publisher: Public Library of Science.

Gonçalves, P. J., Lueckmann, J.-M., Deistler, M., Nonnenmacher, M., Öcal, K., Bassetto, G., Chintaluri, C., Podlaski, W. F., Haddad, S. A., Vogels, T. P., Greenberg, D. S., and Macke, J. H. (2020). Training deep neural density estimators to identify mechanistic models of neural dynamics. *eLife*, 9:e56261. Publisher: eLife Sciences Publications, Ltd.

Guilhot, Q., Wójcik, M., Achterberg, J., and Costa, R. P. (2024). Dynamical similarity analysis can identify compositional dynamics developing in RNNs. arXiv:2410.24070 [cs].

Hanin, B. and Rolnick, D. (2018). How to Start Training: The Effect of Initialization and Architecture. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Harris, K. D. and Shepherd, G. M. G. (2015). The neocortical circuit: themes and variations. *Nature Neuroscience*, 18(2):170–181. Number: 2 Publisher: Nature Publishing Group.

Hasson, U., Nastase, S. A., and Goldstein, A. (2020). Direct Fit to Nature: An Evolutionary Perspective on Biological and Artificial Neural Networks. *Neuron*, 105(3):416–434.

Publisher: Elsevier.

Hines, M. L. and Carnevale, N. T. (1997). The NEURON Simulation Environment. *Neural Computation*, 9(6):1179–1209. Conference Name: Neural Computation.

Huk, A., Bonnen, K., and He, B. J. (2018). Beyond Trial-Based Paradigms: Continuous Behavior, Ongoing Neural Activity, and Natural Stimuli. *Journal of Neuroscience*, 38(35):7551–7558. Publisher: Society for Neuroscience Section: TechSights.

Janocha, K. and Czarnecki, W. M. (2017). On Loss Functions for Deep Neural Networks in Classification. arXiv:1702.05659 [cs].

Jazayeri, M. and Ostojic, S. (2021). Interpreting neural computations by examining intrinsic and embedding dimensionality of neural activity. *Current Opinion in Neurobiology*, 70:113–120.

Jones, I. S. and Kording, K. P. (2021). Might a Single Neuron Solve Interesting Machine Learning Problems Through Successive Computations on Its Dendritic Tree? *Neural Computation*, 33(6):1554–1571.

Kern, S., Müller, S. D., Hansen, N., Büche, D., Ocenasek, J., and Koumoutsakos, P. (2004). Learning probability distributions in continuous evolutionary algorithms – a comparative review. *Natural Computing*, 3(1):77–112.

Khan, A., Sohail, A., Zahoora, U., and Qureshi, A. S. (2020). A survey of the recent architectures of deep convolutional neural networks. *Artificial Intelligence Review*, 53(8):5455–5516.

Kim, R., Li, Y., and Sejnowski, T. J. (2019). Simple framework for constructing functional spiking recurrent neural networks. *Proceedings of the National Academy of Sciences*, 116(45):22811–22820. Publisher: Proceedings of the National Academy of Sciences.

Kim, R. and Sejnowski, T. J. (2021). Strong inhibitory signaling underlies stable temporal dynamics and working memory in spiking neural networks. *Nature Neuroscience*,

24(1):129–139. Publisher: Nature Publishing Group.

Krakauer, J. W., Ghazanfar, A. A., Gomez-Marin, A., MacIver, M. A., and Poeppel, D. (2017). Neuroscience Needs Behavior: Correcting a Reductionist Bias. *Neuron*, 93(3):480–490.

Langdon, C. and Engel, T. A. (2025). Latent circuit inference from heterogeneous neural responses during cognitive tasks. *Nature Neuroscience*, 28(3):665–675. Publisher: Nature Publishing Group.

Larkum, M. E. (2022). Are Dendrites Conceptually Useful? *Neuroscience*, 489:4–14.

Larkum, M. E., Zhu, J. J., and Sakmann, B. (1999). A new cellular mechanism for coupling inputs arriving at different cortical layers. *Nature*, 398(6725):338–341. Publisher: Nature Publishing Group.

Larochelle, H., Bengio, Y., Louradour, J., and Lamblin, P. (2009). Exploring strategies for training deep neural networks. *Journal of machine learning research*, 10(1).

Lengler, J., Jug, F., and Steger, A. (2013). Reliable Neuronal Systems: The Importance of Heterogeneity. *PLOS ONE*, 8(12):e80694. Publisher: Public Library of Science.

Maes, A., Barahona, M., and Clopath, C. (2020). Learning spatiotemporal signals using a recurrent spiking network that discretizes time. *PLOS Computational Biology*, 16(1):e1007606. Publisher: Public Library of Science.

Major, G., Larkum, M. E., and Schiller, J. (2013). Active Properties of Neocortical Pyramidal Neuron Dendrites. *Annual Review of Neuroscience*, 36(Volume 36, 2013):1–24. Publisher: Annual Reviews.

Maselli, A., Gordon, J., Eluchans, M., Lancia, G. L., Thiery, T., Moretti, R., Cisek, P., and Pezzulo, G. (2023). Beyond simple laboratory studies: Developing sophisticated models to study rich behavior. *Physics of Life Reviews*, 46:220–244.

McBride, T. J., Rodriguez-Contreras, A., Trinh, A., Bailey, R., and DeBello, W. M.

(2008). Learning Drives Differential Clustering of Axodendritic Contacts in the Barn Owl Auditory System. *Journal of Neuroscience*, 28(27):6960–6973. Publisher: Society for Neuroscience Section: Articles.

Mejias, J. F. and Longtin, A. (2014). Differential effects of excitatory and inhibitory heterogeneity on the gain and asynchronous state of sparse cortical networks. *Frontiers in Computational Neuroscience*, 8. Publisher: Frontiers.

Murray, J. D., Jaramillo, J., and Wang, X.-J. (2017). Working Memory and Decision-Making in a Frontoparietal Circuit Model. *Journal of Neuroscience*, 37(50):12167–12186. Publisher: Society for Neuroscience Section: Research Articles.

Narkhede, M. V., Bartakke, P. P., and Sutaone, M. S. (2022). A review on weight initialization strategies for neural networks. *Artificial Intelligence Review*, 55(1):291–322.

Neftci, E. O., Mostafa, H., and Zenke, F. (2019). Surrogate Gradient Learning in Spiking Neural Networks: Bringing the Power of Gradient-Based Optimization to Spiking Neural Networks. *IEEE Signal Processing Magazine*, 36(6):51–63.

Nicola, W. and Clopath, C. (2017). Supervised learning in spiking neural networks with FORCE training. *Nature Communications*, 8(1):2208. Publisher: Nature Publishing Group.

Oláh, V. J., Wu, J., Kaczmarek, L. K., and Rowan, M. J. (2025). Input-specific gating of NMDA amplification via HCN channels in mouse L2/3 pyramidal neurons. *eLife*, 13. Publisher: eLife Sciences Publications Limited.

Ostrow, M., Eisen, A., Kozachkov, L., and Fiete, I. (2023). Beyond Geometry: Comparing the Temporal Structure of Computation in Neural Circuits with Dynamical Similarity Analysis. arXiv:2306.10168 [q-bio].

Pagkalos, M., Chavlis, S., and Poirazi, P. (2023). Introducing the Dendrify framework for

incorporating dendrites to spiking neural networks. *Nature Communications*, 14(1):131. Publisher: Nature Publishing Group.

Palmer, L. M., Shai, A. S., Reeve, J. E., Anderson, H. L., Paulsen, O., and Larkum, M. E. (2014). NMDA spikes enhance action potential generation during sensory input. *Nature Neuroscience*, 17(3):383–390. Publisher: Nature Publishing Group.

Pals, M., Macke, J. H., and Barak, O. (2024). Trained recurrent neural networks develop phase-locked limit cycles in a working memory task. *PLOS Computational Biology*, 20(2):e1011852. Publisher: Public Library of Science.

Papamakarios, G. (2019). Neural Density Estimation and Likelihood-free Inference. arXiv:1910.13233 [stat].

Papoutsi, A., Sidiropoulou, K., and Poirazi, P. (2014). Dendritic Nonlinearities Reduce Network Size Requirements and Mediate ON and OFF States of Persistent Activity in a PFC Microcircuit Model. *PLOS Computational Biology*, 10(7):e1003764. Publisher: Public Library of Science.

Pascanu, R. and Jaeger, H. (2011). A neurodynamical model for working memory. *Neural Networks*, 24(2):199–207.

Pastore, M. and Calcagnì, A. (2019). Measuring Distribution Similarities Between Samples: A Distribution-Free Overlapping Index. *Frontiers in Psychology*, 10.

Pazzini, R., Kinouchi, O., and Costa, A. A. (2021). Neuronal avalanches in Watts-Strogatz networks of stochastic spiking neurons. *Physical Review E*, 104(1):014137. Publisher: American Physical Society.

Pellegrino, A., Stein, H., and Cayco-Gajic, N. A. (2024). Dimensionality reduction beyond neural subspaces with slice tensor component analysis. *Nature Neuroscience*, 27(6):1199–1210. Publisher: Nature Publishing Group.

Perez-Nieves, N., Leung, V. C. H., Dragotti, P. L., and Goodman, D. F. M. (2021). Neural

heterogeneity promotes robust learning. *Nature Communications*, 12(1):5791. Publisher: Nature Publishing Group.

Poirazi, P. and Papoutsi, A. (2020). Illuminating dendritic function with computational models. *Nature Reviews Neuroscience*, 21(6):303–321. Publisher: Nature Publishing Group.

Polsky, A., Mel, B. W., and Schiller, J. (2004). Computational subunits in thin dendrites of pyramidal cells. *Nature Neuroscience*, 7(6):621–627. Bandiera_abtest: a Cg_type: Nature Research Journals Number: 6 Primary_atype: Research Publisher: Nature Publishing Group.

Pospischil, M., Toledo-Rodriguez, M., Monier, C., Piwkowska, Z., Bal, T., Frégnac, Y., Markram, H., and Destexhe, A. (2008). Minimal Hodgkin–Huxley type models for different classes of cortical and thalamic neurons. *Biological cybernetics*, 99:427–441. Publisher: Springer.

Rhodes, B. J., Bullock, D., Verwey, W. B., Averbeck, B. B., and Page, M. P. A. (2004). Learning and production of movement sequences: Behavioral, neurophysiological, and modeling perspectives. *Human Movement Science*, 23(5):699–746.

Rhodes, P. (2006). The Properties and Implications of NMDA Spikes in Neocortical Pyramidal Cells. *Journal of Neuroscience*, 26(25):6704–6715. Publisher: Society for Neuroscience Section: Articles.

Salimans, T. and Kingma, D. P. (2016). Weight Normalization: A Simple Reparameterization to Accelerate Training of Deep Neural Networks. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc.

Schulz, J. M., Knoflach, F., Hernandez, M.-C., and Bischofberger, J. (2018). Dendrite-targeting interneurons control synaptic NMDA-receptor activation via nonlinear 5-GABAA receptors. *Nature Communications*, 9(1):3576. Publisher: Nature Publishing Group.

Schuman, B., Dellal, S., Prönneke, A., Machold, R., and Rudy, B. (2021). Neocortical Layer 1: An Elegant Solution to Top-Down and Bottom-Up Integration. *Annual review of neuroscience*, 44:221–252.

Self, M. W., Kooijmans, R. N., Supèr, H., Lamme, V. A., and Roelfsema, P. R. (2012). Different glutamate receptors convey feedforward and recurrent processing in macaque V1. *Proceedings of the National Academy of Sciences*, 109(27):11031–11036. Publisher: Proceedings of the National Academy of Sciences.

Shanahan, M. (2008). Dynamical complexity in small-world networks of spiking neurons. *Physical Review E*, 78(4):041924. Publisher: American Physical Society.

Shorten, C. and Khoshgoftaar, T. M. (2019). A survey on Image Data Augmentation for Deep Learning. *Journal of Big Data*, 6(1):60.

Soo, W. W. M., Goudar, V., and Wang, X.-J. (2023). Training biologically plausible recurrent neural networks on cognitive tasks with long-term dependencies. Pages: 2023.10.10.561588 Section: New Results.

Sporns, O. and Zwi, J. D. (2004). The small world of the cerebral cortex. *Neuroinformatics*, 2(2):145–162.

Stefanos, S. S., Papoutsi, A., and Poirazi, P. (2019). Structured connectivity exploits NMDA non-linearities to enable flexible encoding of multiple memoranda in a PFC circuit model. Pages: 733519 Section: New Results.

Sussillo, D. and Abbott, L. F. (2009). Generating Coherent Patterns of Activity from Chaotic Neural Networks. *Neuron*, 63(4):544–557.

Sussillo, D. and Barak, O. (2012). Opening the Black Box: Low-Dimensional Dynamics in High-Dimensional Recurrent Neural Networks. *Neural Computation*, 25(3):626–649. Publisher: MIT Press.

Takahashi, N., Oertner, T. G., Hegemann, P., and Larkum, M. E. (2016). Active cortical

dendrites modulate perception. *Science*, 354(6319):1587–1590. Publisher: American Association for the Advancement of Science.

Tanaka, G., Yamane, T., Héroux, J. B., Nakane, R., Kanazawa, N., Takeda, S., Numata, H., Nakano, D., and Hirose, A. (2019). Recent advances in physical reservoir computing: A review. *Neural Networks*, 115:100–123. Publisher: Elsevier.

Taylor, L. and Nitschke, G. (2018). Improving Deep Learning with Generic Data Augmentation. In *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1542–1547.

Tolley, N., Rodrigues, P. L. C., Gramfort, A., and Jones, S. R. (2024). Methods and considerations for estimating parameters in biophysically detailed neural models with simulation based inference. *PLOS Computational Biology*, 20(2):e1011108. Publisher: Public Library of Science.

Vugt, B. v., Kerkoerle, T. v., Vartak, D., and Roelfsema, P. R. (2020). The Contribution of AMPA and NMDA Receptors to Persistent Firing in the Dorsolateral Prefrontal Cortex in Working Memory. *Journal of Neuroscience*, 40(12):2458–2470. Publisher: Society for Neuroscience Section: Research Articles.

Vyas, S., Golub, M. D., Sussillo, D., and Shenoy, K. V. (2020). Computation Through Neural Population Dynamics. *Annual Review of Neuroscience*, 43(Volume 43, 2020):249–275. Publisher: Annual Reviews.

Wang, G., Sun, Y., Cheng, S., and Song, S. (2023). Evolving Connectivity for Recurrent Spiking Neural Networks. *Advances in Neural Information Processing Systems*, 36:2991–3007.

Wang, M., Yang, Y., Wang, C.-J., Gamo, N. J., Jin, L. E., Mazer, J. A., Morrison, J. H., Wang, X.-J., and Arnsten, A. F. T. (2013). NMDA Receptors Subserve Persistent Neuronal Firing during Working Memory in Dorsolateral Prefrontal Cortex. *Neuron*, 77(4):736–749. Publisher: Elsevier.

Wang, Q., Ma, Y., Zhao, K., and Tian, Y. (2022). A Comprehensive Survey of Loss Functions in Machine Learning. *Annals of Data Science*, 9(2):187–212.

Wang, X.-J. (2021). 50 years of mnemonic persistent activity: quo vadis? *Trends in Neurosciences*, 44(11):888–902. Publisher: Elsevier.

Watts, D. J. and Strogatz, S. H. (1998). Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440–442. Publisher: Nature Publishing Group.

Xu, N.-l., Harnett, M. T., Williams, S. R., Huber, D., O'Connor, D. H., Svoboda, K., and Magee, J. C. (2012). Nonlinear dendritic integration of sensory and motor input during an active sensing task. *Nature*, 492(7428):247–251. Publisher: Nature Publishing Group.

Yang, G. R., Joglekar, M. R., Song, H. F., Newsome, W. T., and Wang, X.-J. (2019). Task representations in neural networks trained to perform many cognitive tasks. *Nature Neuroscience*, 22(2):297–306. Publisher: Nature Publishing Group.

Yang, G. R. and Molano-Mazón, M. (2021). Towards the next generation of recurrent network models for cognitive neuroscience. *Current Opinion in Neurobiology*, 70:182–192.

Yu, X. and Gen, M. (2010). *Introduction to evolutionary algorithms*. Springer Science & Business Media.

Yu, Y., Si, X., Hu, C., and Zhang, J. (2019). A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures. *Neural Computation*, 31(7):1235–1270.

Zhang, H. and Vargas, D. V. (2023). A Survey on Reservoir Computing and its Interdisciplinary Applications Beyond Traditional Machine Learning. *IEEE Access*, 11:81033–81070. Conference Name: IEEE Access.

CHAPTER 5

Conclusions

5.1 Summary

The previous chapters have demonstrated that biophysical models combined with deep learning are uniquely suited to understand the mechanistic origins of emergent properties in neural systems. The novel scientific contributions in this thesis specifically focus on connecting cell-level properties to network-level dynamics in the form of 1) MEG/EEG neural oscillations (Chapter 3) using the Human Neocortical Neurosolver (HNN) modeling software (Chapter 2), and 2) the cognitive function of working memory using a biophysical reservoir computing framework that incorporates characteristic properties of cortical cells and circuits (Chapter 4). It is important to highlight that several of the results presented were only identifiable because of the unique combination of biophysical modeling and deep learning. This combination has largely been facilitated by the rise of open-source scientific tools in the neuroscience community (Chapter 2), allowing for the rapid integration of biophysical models and deep learning algorithms.

As experimental and computational technology improves, the scientific insights gained from biophysical models of neural networks will continue to accelerate. Below I describe what I see as the most promising directions for the field to pursue with deep learning

assisted biophysical modeling in the scientific, technological, and clinical research domains.

5.2 Untapped potential of combining physics-based models and NeuroAI

The combination of artificial intelligence (AI) and neuroscience has the potential to be a bidirectional relationship in which 1) AI is used to investigate neuroscience questions, and 2) neuroscience findings are used to improve AI systems. Researchers have unified around the term "NeuroAI" to distinguish the unique problems of working at the intersection of neuroscience and AI. We note that AI is an expansive category that encompasses machine learning, and deep learning as a further subset. My thesis work suggests that a promising avenue of NeuroAI research is using AI techniques (i.e. deep learning) to help distill neuroscience knowledge from physics-based models.

The major techniques presented in this thesis, namely surrogate modeling and SBI, are clear examples where AI is employed to gain scientific insights from physics-based models. Together these techniques address the challenges of physics-based simulations with respect to their computational cost (through surrogate modeling) and complex parameter-output mappings (through SBI). A future research direction is the creation of a unified AI system that simultaneously captures the forward and inverse relationships of a physics-based model for broad classes of neuroscience problems. A potential deep learning framework for achieving this goal is foundation modeling.

Foundation models traditionally refer to deep neural networks (i.e. transformers) that have been trained on a vast corpus of data in a specific domain. In the domain of language modeling, models like ChatGPT (Radford et al. (2018)), LLaMA (Touvron et al. (2023)), and DeepSeek (DeepSeek-AI et al. (2025)) have seen widespread adoption outside of the research community. In light of the success of large language models, researchers in numerous domains, including neuroscience, are looking to see if a similar inflection point

can be reached (Wang and Chen (2025); Azabou et al. (2023); Cui et al. (2024)).

One path towards improving foundation models is the direct integration of physics-based simulations in the training data. Simulations are a natural fit for the data requirements of foundation models, as the computational cost of simulation is the only limit to the size of the training dataset that can be generated. Examples of such an approach have already appeared in the domains of economics (Chen et al. (2023)), chaos theory (Lai et al. (2025)), and robotics (Mahler et al. (2016)). The lofty goal of such approaches is for the foundation models to achieve high-performance on problems not present in the training dataset, otherwise known as generalization. In the context of neuroscience, an EEG foundation model trained with this approach would ideally impose inductive biases that equip the model with an internal understanding of generative mechanisms underlying EEG signals. An example of generalization would be the ability to predict seizure onset without recordings of seizures being explicitly present in the training data. The presumption is that when the model takes neural recordings as input, the internal representation is directly related to the biophysical mechanisms learned from the simulated training data.

5.3 Technical hurdles for future development of physics-based NeuroAI

To fully realize the potential of physics-based foundation models, it will still be necessary to address the practical problems presented by biophysical modeling for targeted scientific questions. The tools presented in this thesis demonstrate how deep learning can be used to address many of the shortcomings of biophysical modeling. Nevertheless, there remain several technical hurdles for the use of biophysical models, and their potential integration with foundation models.

Fitting models to data or desired emergent properties is far from a solved problem. In

the context of biophysical modeling, parameter fitting techniques for black-box simulators (i.e. no access to gradients) remains an active area of research, particularly for high-dimensional parameter spaces (Papenmeier et al. (2025); Zhou et al. (2024)). Gradients are useful for parameter fitting because they can indicate the direction a parameter should be changed to produce a desired output. A recent major advance on this front is implementing biophysical models in software that tracks gradients during simulations (Deistler et al. (2024)). This approach allows the use of gradient-based optimization techniques for training biophysical models. The potential impact of this advance is significant, as gradient-based optimization underlies the training of nearly every deep learning architecture. However, even when biophysical models are implemented in software such that the gradients are tracked during simulation, the associated loss landscape can be highly complex, leading to the failure of standard techniques like gradient descent compared to computationally expensive techniques like evolutionary algorithms (Hazelden et al. (2023); Wang et al. (2023)). Research on hybrid methods that combine gradient information with evolutionary algorithms may be a promising path towards dealing with challenging loss landscapes imposed by biophysical models (Yang et al. (2022); Maheswaranathan et al. (2019)). Alternatively, online optimization techniques like those used in spiking neural networks (i.e. FORCE training Nicola and Clopath (2017)) may prove valuable now that gradient information is accessible in biophysical models.

Another bottleneck for neuroscience foundation models is the availability of experimental data. While simulated datasets may provide useful inductive biases, in vivo recordings of neural activity are still essential for real-world applications of neuroscience foundation models. Fortunately, technology for recording neural activity in numerous modalities is advancing rapidly, with tools like Neuropixels increasing the amount of neural data collected in a single experiment by multiple orders of magnitude (Juavinett et al. (2019); Steinmetz et al. (2021); Paulk et al. (2022)). There are also organizations like the Allen Institute (de Vries et al. (2023); Hawrylycz et al. (2014)) and the Blue Brain Project

(Markram (2006)) conducting monumental projects with the goal of curating massive neuroscience specific datasets in subdomains such as physiology, behavior, genetics, molecular biology, development, and connectivity. It is important to note that even with access to a limitless set of simulated neural data, it is currently unknown which features and representations should be extracted to identify computational principles of the brain and their associated neural mechanisms. Many neuroscientists consider computations in the brain to be best represented as a dynamical system which evolves upon a low-dimensional manifold (Shenoy et al. (2013); O'Shea et al. (2022); Gallego et al. (2017); Langdon and Engel (2025)). Therefore, integration of diverse neural datasets, both experimentally recorded and simulated, could be achieved through the appropriate representation of neural dynamics. Techniques that compare neural activity as a function of their underlying dynamical mechanisms remains an active area of research that may help in this regard (Guilhot et al. (2024); Ostrow et al. (2023)).

An important limitation that challenges the construction of physics-based neuroAI is the true complexity of biological systems. A major goal of this thesis is arguing for techniques such as biophysical modeling which are able to represent complex neural systems. However, these models still invoke strong theoretical assumptions that are only partially consistent with available data. For example, in the cortical column model "HNN" presented in Developing widely accessible and sustainable neuro-modeling tools (Chapter 2), the extrinsic sources that drive activity in the network are divided into "feedforward" drives that arrive at proximal dendritic locations, and "feedback" drives that arrive as distal feedback locations (Neymotin et al. (2020); Jas et al. (2023)). While these input pathways reflect experimental studies which characterize a canonical neocortical microcircuit (Harris and Shepherd (2015)), these pathways primarily summarize the largest trends in anatomical connectivity from tract tracing studies. In reality, there are always numerous exceptions to the connectivity rules suggested by computational models, and it is not always apparent if these exceptions are functionally important. For the future goal

of merging artificial intelligence and physics-based models, it will be crucial to be aware of the assumptions imposed by models, and critically evaluate if the inductive biases they introduce help or harm task performance.

5.4 Digital twins as the future of neural modeling

A digital twin refers to the simulation of a real-world object or system, and can be used for testing and monitoring the system in a virtual environment. Digital twins are used in numerous domains including construction (Opoku et al. (2021)), aerospace engineering (Xiong and Wang (2022)), meteorology (Li et al. (2023)), agriculture (Pylianidis et al. (2021)), and biology (Gerach et al. (2021)).

In the field of biology, there has been significant advancement of patient-specific digital twins of the heart, enabling high-fidelity monitoring of cardiovascular issues, and simulation-informed decision making on optimal treatments (Gerach et al. (2021)). There are several similarities between computational models of the brain and heart that suggest a congruency in the construction of digital twins. However, unlike the brain, the correct functioning of the heart is undisputed. The evaluation and benchmarking of a heart digital twin directly relates to if the heart is pumping blood with enough pressure to reach the vital organs of the body. In the case of the brain, there are an overwhelming number of functions with varying levels of overlap in terms of when these functions are employed, and which structures are involved.

Despite the challenges of developing digital twins of the brain, there have been efforts at developing clinically relevant biophysical models for a variety of therapeutic domains. For example, Geerts et al. (2012) has developed a thalamocortical model of schizophrenia, and demonstrated its ability to predict the effectiveness of drugs in a prospective clinical trial. Similarly, Guet-McCreight et al. (2024) has developed a detailed model of cortical circuits to study EEG biomarkers of depression, and can be used for optimal dose predictions. In

the domain of epilepsy, Dollomaja et al. (2025) has developed a whole-brain model that is based on techniques from dynamic causal modeling, and has the potential to be used for personalized epilepsy treatments.

Given the diversity of brain function and recording modalities, the future success of digital twin technology will require specifying a neural function as the target of modeling, and choosing biophysical details at all spatial scales based on the target. Rather than building a generalized digital twin of numerous brain functions, researchers should focus on building digital twins targeting functions like working memory, planning, or attention. For each target function, the biological details included in a patient-specific brain model may change based on their importance for the overall function, along with the data that is necessary to fit a model to that patient. Supporting this view, Faingold (2004) describes how neuropsychiatric drug development may benefit from viewing drug action as impacting emergent properties on brain networks, rather than their cellular-level impact in isolation. Given the highly complex and challenging nature of experiments studying emergent properties of neural systems, biophysical modeling with deep learning is well-posed to assist in these efforts.

This function-informed modeling moves away from the highly popular framework introduced by David Marr, where he proposes that mechanistic properties should not be compared at different spatial scales, and instead theories regarding neural function should be strictly confined within their respective levels (Marr (1982)). Indeed, the computational neuroscience community is beginning to challenge the framework of Marr's levels (Postle (2024); Pillow (2024)). It is important to note that this modeling approach is not a recent development in the field. Arguably the majority of biophysical modeling research uses targeted emergent properties to constrain modeling choices. As a less common but valuable alternative, bottom-up modeling efforts such as the Blue Brain Project (Markram (2006)) push the limits of biophysical neural modeling with a similar departure from Marr's levels, but with the goal of more generalized models that can be used to

investigate multiple scientific questions. While the scientific challenges associated with these approaches are distinct, they both benefit from the technological advancements described in this thesis, and emphasize the value of linking cell-level biophysical properties to network-level function.

With the next generation of deep learning tools, the full potential of biophysical neural modeling will be realized through novel mechanistic insights that were previously unattainable. These insights will not be derived from simulation alone. The next major step for biophysical modeling will be using these tools to accelerate the theory-experimentation loop via close interaction of modelers and experimentalists. The technological and scientific advancements necessary for this goal will be significant, but it is a challenge the neuroscience community is well-equipped to overcome.

References

Azabou, M., Arora, V., Ganesh, V., Mao, X., Nachimuthu, S., Mendelson, M., Richards, B., Perich, M., Lajoie, G., and Dyer, E. (2023). A Unified, Scalable Framework for Neural Population Decoding. *Advances in Neural Information Processing Systems*, 36:44937–44956.

Chen, H., Cheng, Y., Liu, Y., and Tang, K. (2023). Teaching economics to the machines. *Available at SSRN*, 4642167.

Cui, W., Jeong, W., Thölke, P., Medani, T., Jerbi, K., Joshi, A. A., and Leahy, R. M. (2024). Neuro-GPT: Towards A Foundation Model For EEG. In *2024 IEEE International Symposium on Biomedical Imaging (ISBI)*, pages 1–5. ISSN: 1945-8452.

de Vries, S. E., Siegle, J. H., and Koch, C. (2023). Sharing neurophysiology data from the Allen Brain Observatory. *eLife*, 12:e85550. Publisher: eLife Sciences Publications, Ltd.

DeepSeek-AI, Liu, A., Feng, B., Xue, B., Wang, B., Wu, B., Lu, C., Zhao, C., Deng, C., Zhang, C., Ruan, C., Dai, D., Guo, D., Yang, D., Chen, D., Ji, D., Li, E., Lin, F., Dai, F., Luo, F., Hao, G., Chen, G., Li, G., Zhang, H., Bao, H., Xu, H., Wang, H., Zhang, H., Ding, H., Xin, H., Gao, H., Li, H., Qu, H., Cai, J. L., Liang, J., Guo, J., Ni, J., Li, J., Wang, J., Chen, J., Chen, J., Yuan, J., Qiu, J., Li, J., Song, J., Dong, K., Hu, K., Gao, K., Guan, K., Huang, K., Yu, K., Wang, L., Zhang, L., Xu, L., Xia, L., Zhao, L., Wang, L., Zhang, L., Li, M., Wang, M., Zhang, M., Zhang, M., Tang, M., Li, M., Tian, N., Huang, P., Wang, P., Zhang, P., Wang, Q., Zhu, Q., Chen, Q., Du, Q., Chen, R. J., Jin, R. L., Ge, R., Zhang, R., Pan, R., Wang, R., Xu, R., Zhang, R., Chen, R., Li, S. S., Lu, S., Zhou, S., Chen, S., Wu, S., Ye, S., Ye, S., Ma, S., Wang, S., Zhou, S., Yu, S., Zhou, S., Pan, S., Wang, T., Yun, T., Pei, T., Sun, T., Xiao, W. L., Zeng, W., Zhao, W., An, W., Liu, W., Liang, W., Gao, W., Yu, W., Zhang, W., Li, X. Q., Jin, X., Wang, X., Bi, X., Liu, X., Wang, X., Shen, X., Chen, X., Zhang, X., Chen, X., Nie, X., Sun, X., Wang, X., Cheng, X., Liu, X., Xie, X., Liu, X., Yu, X., Song, X., Shan, X.,

Zhou, X., Yang, X., Li, X., Su, X., Lin, X., Li, Y. K., Wang, Y. Q., Wei, Y. X., Zhu, Y. X., Zhang, Y., Xu, Y., Xu, Y., Huang, Y., Li, Y., Zhao, Y., Sun, Y., Li, Y., Wang, Y., Yu, Y., Zheng, Y., Zhang, Y., Shi, Y., Xiong, Y., He, Y., Tang, Y., Piao, Y., Wang, Y., Tan, Y., Ma, Y., Liu, Y., Guo, Y., Wu, Y., Ou, Y., Zhu, Y., Wang, Y., Gong, Y., Zou, Y., He, Y., Zha, Y., Xiong, Y., Ma, Y., Yan, Y., Luo, Y., You, Y., Liu, Y., Zhou, Y., Wu, Z. F., Ren, Z. Z., Ren, Z., Sha, Z., Fu, Z., Xu, Z., Huang, Z., Zhang, Z., Xie, Z., Zhang, Z., Hao, Z., Gou, Z., Ma, Z., Yan, Z., Shao, Z., Xu, Z., Wu, Z., Zhang, Z., Li, Z., Gu, Z., Zhu, Z., Liu, Z., Li, Z., Xie, Z., Song, Z., Gao, Z., and Pan, Z. (2025). DeepSeek-V3 Technical Report. arXiv:2412.19437 [cs].

Deistler, M., Kadhim, K. L., Pals, M., Beck, J., Huang, Z., Gloeckler, M., Lappalainen, J. K., Schröder, C., Berens, P., Gonçalves, P. J., and Macke, J. H. (2024). Differentiable simulation enables large-scale training of detailed biophysical models of neural dynamics. Pages: 2024.08.21.608979 Section: New Results.

Dollomaja, B., Wang, H. E., Guye, M., Makhalova, J., Bartolomei, F., and Jirsa, V. K. (2025). Virtual epilepsy patient cohort: Generation and evaluation. *PLOS Computational Biology*, 21(4):e1012911. Publisher: Public Library of Science.

Faingold, C. L. (2004). Emergent properties of CNS neuronal networks as targets for pharmacology: application to anticonvulsant drug action. *Progress in Neurobiology*, 72(1):55–85.

Gallego, J. A., Perich, M. G., Miller, L. E., and Solla, S. A. (2017). Neural Manifolds for the Control of Movement. *Neuron*, 94(5):978–984.

Geerts, H., Spiros, A., Roberts, P., Twyman, R., Alphs, L., and Grace, A. A. (2012). Blinded Prospective Evaluation of Computer-Based Mechanistic Schizophrenia Disease Model for Predicting Drug Response. *PLOS ONE*, 7(12):e49732. Publisher: Public Library of Science.

Gerach, T., Schuler, S., Fröhlich, J., Lindner, L., Kovacheva, E., Moss, R., Wülfers, E. M.,

Seemann, G., Wieners, C., and Loewe, A. (2021). Electro-Mechanical Whole-Heart Digital Twins: A Fully Coupled Multi-Physics Approach. *Mathematics*, 9(11):1247. Number: 11 Publisher: Multidisciplinary Digital Publishing Institute.

Guet-McCreight, A., Mazza, F., Prevot, T. D., Sibille, E., and Hay, E. (2024). Therapeutic dose prediction of 5-GABA receptor modulation from simulated EEG of depression severity. *PLOS Computational Biology*, 20(12):e1012693. Publisher: Public Library of Science.

Guilhot, Q., Wójcik, M., Achterberg, J., and Costa, R. P. (2024). Dynamical similarity analysis can identify compositional dynamics developing in RNNs. arXiv:2410.24070 [cs].

Harris, K. D. and Shepherd, G. M. G. (2015). The neocortical circuit: themes and variations. *Nature Neuroscience*, 18(2):170–181. Number: 2 Publisher: Nature Publishing Group.

Hawrylycz, M., Ng, L., Feng, D., Sunkin, S., Szafer, A., and Dang, C. (2014). The Allen Brain Atlas. In Kasabov, N., editor, *Springer Handbook of Bio-/Neuroinformatics*, pages 1111–1126. Springer, Berlin, Heidelberg.

Hazelden, J., Liu, Y. H., Shlizerman, E., and Shea-Brown, E. (2023). Evolutionary algorithms as an alternative to backpropagation for supervised training of Biophysical Neural Networks and Neural ODEs.

Jas, M., Thorpe, R., Tolley, N., Bailey, C., Brandt, S., Caldwell, B., Cheng, H., Daniels, D., Pujol, C. F., Khalil, M., Kanekar, S., Kohl, C., Kolozsvári, O., Lankinen, K., Loi, K., Neymotin, S., Partani, R., Pelah, M., Rockhill, A., Sherif, M., Hamalainen, M., and Jones, S. (2023). HNN-core: A Python software for cellular and circuit-level interpretation of human MEG/EEG. *Journal of Open Source Software*, 8(92):5848.

Juavinett, A. L., Bekheet, G., and Churchland, A. K. (2019). Chronically implanted Neuropixels probes enable high-yield recordings in freely moving mice. *eLife*, 8:e47188.

Lai, J., Bao, A., and Gilpin, W. (2025). Panda: A pretrained forecast model for universal representation of chaotic dynamics. arXiv:2505.13755 [cs].

Langdon, C. and Engel, T. A. (2025). Latent circuit inference from heterogeneous neural responses during cognitive tasks. *Nature Neuroscience*, 28(3):665–675. Publisher: Nature Publishing Group.

Li, X., Feng, M., Ran, Y., Su, Y., Liu, F., Huang, C., Shen, H., Xiao, Q., Su, J., Yuan, S., and Guo, H. (2023). Big Data in Earth system science and progress towards a digital twin. *Nature Reviews Earth & Environment*, 4(5):319–332. Publisher: Nature Publishing Group.

Maheswaranathan, N., Metz, L., Tucker, G., Choi, D., and Sohl-Dickstein, J. (2019). Guided evolutionary strategies: augmenting random search with surrogate gradients. In *Proceedings of the 36th International Conference on Machine Learning*, pages 4264–4273. PMLR. ISSN: 2640-3498.

Mahler, J., Pokorny, F. T., Hou, B., Roderick, M., Laskey, M., Aubry, M., Kohlhoff, K., Kröger, T., Kuffner, J., and Goldberg, K. (2016). Dex-Net 1.0: A cloud-based network of 3D objects for robust grasp planning using a Multi-Armed Bandit model with correlated rewards. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1957–1964.

Markram, H. (2006). The Blue Brain Project. *Nature Reviews Neuroscience*, 7(2):153–160. Publisher: Nature Publishing Group.

Marr, D. (1982). *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. Henry Holt and Co., Inc., USA.

Neymotin, S. A., Daniels, D. S., Caldwell, B., McDougal, R. A., Carnevale, N. T., Jas, M., Moore, C. I., Hines, M. L., Hämäläinen, M., and Jones, S. R. (2020). Human Neocortical Neurosolver (HNN), a new software tool for interpreting the cellular and network origin of human MEG/EEG data. *eLife*, 9:e51214. Publisher: eLife Sciences

Publications, Ltd.

Nicola, W. and Clopath, C. (2017). Supervised learning in spiking neural networks with FORCE training. *Nature Communications*, 8(1):2208. Publisher: Nature Publishing Group.

Opoku, D.-G. J., Perera, S., Osei-Kyei, R., and Rashidi, M. (2021). Digital twin application in the construction industry: A literature review. *Journal of Building Engineering*, 40:102726.

Ostrow, M., Eisen, A., Kozachkov, L., and Fiete, I. (2023). Beyond Geometry: Comparing the Temporal Structure of Computation in Neural Circuits with Dynamical Similarity Analysis. arXiv:2306.10168 [q-bio].

O'Shea, D. J., Duncker, L., Goo, W., Sun, X., Vyas, S., Trautmann, E. M., Diester, I., Ramakrishnan, C., Deisseroth, K., Sahani, M., and Shenoy, K. V. (2022). Direct neural perturbations reveal a dynamical mechanism for robust computation. Pages: 2022.12.16.520768 Section: New Results.

Papenmeier, L., Poloczek, M., and Nardi, L. (2025). Understanding High-Dimensional Bayesian Optimization. *arXiv preprint arXiv:2502.09198*.

Paulk, A. C., Kfir, Y., Khanna, A. R., Mustroph, M. L., Trautmann, E. M., Soper, D. J., Stavisky, S. D., Welkenhuysen, M., Dutta, B., Shenoy, K. V., Hochberg, L. R., Richardson, R. M., Williams, Z. M., and Cash, S. S. (2022). Large-scale neural recordings with single neuron resolution using Neuropixels probes in human cortex. *Nature Neuroscience*, 25(2):252–263. Publisher: Nature Publishing Group.

Pillow, J. W. (2024). Cross Talk opposing view: Marr's three levels of analysis are not useful as a framework for neuroscience. *The Journal of Physiology*, 602(9):1915–1917.

Postle, B. R. (2024). Has AI "killed" Marr's 'three'-levels framework? PubPub.

Pylianidis, C., Osinga, S., and Athanasiadis, I. N. (2021). Introducing digital twins to

agriculture. *Computers and Electronics in Agriculture*, 184:105942.

Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I. (2018). Improving language understanding by generative pre-training. Publisher: San Francisco, CA, USA.

Shenoy, K. V., Sahani, M., and Churchland, M. M. (2013). Cortical Control of Arm Movements: A Dynamical Systems Perspective. *Annual Review of Neuroscience*, 36(1):337–359.

Steinmetz, N. A., Aydin, C., Lebedeva, A., Okun, M., Pachitariu, M., Bauza, M., Beau, M., Bhagat, J., Böhm, C., Broux, M., Chen, S., Colonell, J., Gardner, R. J., Karsh, B., Kloosterman, F., Kostadinov, D., Mora-Lopez, C., O'Callaghan, J., Park, J., Putzeys, J., Sauerbrei, B., van Daal, R. J. J., Vollan, A. Z., Wang, S., Welkenhuysen, M., Ye, Z., Dudman, J. T., Dutta, B., Hantman, A. W., Harris, K. D., Lee, A. K., Moser, E. I., O'Keefe, J., Renart, A., Svoboda, K., Häusser, M., Haesler, S., Carandini, M., and Harris, T. D. (2021). Neuropixels 2.0: A miniaturized high-density probe for stable, long-term brain recordings. *Science*, 372(6539):eabf4588. Publisher: American Association for the Advancement of Science.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., and Azhar, F. (2023). Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Wang, G., Sun, Y., Cheng, S., and Song, S. (2023). Evolving Connectivity for Recurrent Spiking Neural Networks. *Advances in Neural Information Processing Systems*, 36:2991–3007.

Wang, R. and Chen, Z. S. (2025). Large-scale foundation models and generative AI for BigData neuroscience. *Neuroscience Research*, 215:3–14.

Xiong, M. and Wang, H. (2022). Digital twin applications in aviation industry: A review. *The International Journal of Advanced Manufacturing Technology*, 121(9):5677–5692.

Yang, S., Tian, Y., He, C., Zhang, X., Tan, K. C., and Jin, Y. (2022). A Gradient-Guided Evolutionary Approach to Training Deep Neural Networks. *IEEE Transactions on Neural Networks and Learning Systems*, 33(9):4861–4875.

Zhou, M., Cui, M., Xu, D., Zhu, S., Zhao, Z., and Abusorrah, A. (2024). Evolutionary optimization methods for high-dimensional expensive problems: A survey. *IEEE/CAA Journal of Automatica Sinica*, 11(5):1092–1105. Publisher: IEEE.

Appendix A

Connectivity tutorial

The relationship between network connectivity and emergent neural oscillations is an important an unresolved problem in neuroscience. The original HNN model implemented cell connections in an all-to-all pattern, where the synapse strength decayed with increasing distance between cells. My contribution to HNN-core is designing a simple interface for users to modify and add connections to the HNN model, notably enabling sparse connectivity to connect subsets of neurons.

This tutorial can be found on the HNN-core examples documentation page:

https://jonescompneurolab.github.io/hnn-core/stable/auto_examples/index.html

plot_connectivity

This example demonstrates how to modify the network connectivity.

[ ]:
```
import os.path as op
```

Let us import `hnn_core`.

[ ]:
```
import hnn_core
from hnn_core import jones_2009_model, simulate_dipole
```

To explore how to modify network connectivity, we will start with simulating the evoked response from the `evoked example <sphx_glr_auto_examples_plot_simulate_evoked.py>`, and explore how it changes with new connections. We first instantiate the network. (Note: Setting `add_drives_from_params=True` loads a set of predefined drives without the drives API shown previously).

[ ]:
```
net_erp = jones_2009_model(add_drives_from_params=True)
```

Instantiating the network comes with a predefined set of connections that reflect the canonical neocortical microcircuit. `net.connectivity` is a list of dictionaries which detail every cell-cell, and drive-cell connection. The weights of these connections can be visualized with :func:`~hnn_core.viz.plot_connectivity_weights` as well as :func:`~hnn_core.viz.plot_cell_connectivity`. We can search for specific connections using `pick_connection` which returns the indices of `net.connectivity` that match the provided parameters.

[ ]:
```
from hnn_core.viz import plot_connectivity_matrix, plot_cell_connectivity
from hnn_core.network import pick_connection

print(len(net_erp.connectivity))

conn_indices = pick_connection(
    net=net_erp, src_gids='L5_basket', target_gids='L5_pyramidal',
    loc='soma', receptor='gabaa')
conn_idx = conn_indices[0]
print(net_erp.connectivity[conn_idx])
plot_connectivity_matrix(net_erp, conn_idx)

# Note here that ``'src_gids'`` is a `set` object
# The `.pop()` method can be used to sample a random element
```

```
src_gid = net_erp.connectivity[conn_idx]['src_gids'].copy().pop()
fig = plot_cell_connectivity(net_erp, conn_idx, src_gid)
```

Data recorded during simulations are stored under :class:~hnn_core.Cell_Response. Spiking activity can be visualized after a simulation is using :meth:~hnn_core.Cell_Response.plot_spikes_raster

[ ]:
```
dpl_erp = simulate_dipole(net_erp, tstop=170., n_trials=1)
net_erp.cell_response.plot_spikes_raster()
```

We can also define our own connections to test the effect of different connectivity patterns. To start, `net.clear_connectivity()` can be used to clear all cell-to-cell connections. By default, previously defined drives to the network are retained, but can be removed with `net.clear_drives()`. `net.add_connection` is then used to create a custom network. Let us first create an all-to-all connectivity pattern between the L5 pyramidal cells, and L2 basket cells. :meth:hnn_core.Network.add_connection allows connections to be specified with either cell names, or the cell IDs (gids) directly.

[ ]:
```
def get_network(probability=1.0):
    net = jones_2009_model(add_drives_from_params=True)
    net.clear_connectivity()

    # Pyramidal cell connections
    location, receptor = 'distal', 'ampa'
    weight, delay, lamtha = 1.0, 1.0, 70
    src = 'L5_pyramidal'
    conn_seed = 3
    for target in ['L5_pyramidal', 'L2_basket']:
        net.add_connection(src, target, location, receptor,
                           delay, weight, lamtha, probability=probability,
                           conn_seed=conn_seed)

    # Basket cell connections
    location, receptor = 'soma', 'gabaa'
    weight, delay, lamtha = 1.0, 1.0, 70
    src = 'L2_basket'
    for target in ['L5_pyramidal', 'L2_basket']:
        net.add_connection(src, target, location, receptor,
                           delay, weight, lamtha, probability=probability,
                           conn_seed=conn_seed)
    return net


net_all = get_network()
dpl_all = simulate_dipole(net_all, tstop=170., n_trials=1)
```

We can additionally use the `probability` argument to create a sparse connectivity pattern instead of all-to-all. Let's try creating the same network with a 10% chance of cells connecting to each

other.

[ ]:
```
net_sparse = get_network(probability=0.1)
dpl_sparse = simulate_dipole(net_sparse, tstop=170., n_trials=1)
```

With the previous connection pattern there appears to be synchronous rhythmic firing of the L5 pyramidal cells with a period of 10 ms. The synchronous activity is visible as vertical lines where several cells fire simultaneously Using the sparse connectivity pattern produced a lot more spiking in the L5 pyramidal cells.

[ ]:
```
net_all.cell_response.plot_spikes_raster()
net_sparse.cell_response.plot_spikes_raster()
```

We can plot the sparse connectivity pattern between cell populations.

[ ]:
```
conn_indices = pick_connection(
    net=net_sparse, src_gids='L2_basket', target_gids='L2_basket',
    loc='soma', receptor='gabaa')

conn_idx = conn_indices[0]
plot_connectivity_matrix(net_sparse, conn_idx)
```

Note that the sparsity is in addition to the weight decay with distance from the source cell.

[ ]:
```
src_gid = net_sparse.connectivity[conn_idx]['src_gids'].copy().pop()
plot_cell_connectivity(net_sparse, conn_idx, src_gid=src_gid)
```

In the sparse network, there still appears to be some rhythmicity where the cells are firing synchronously with a smaller period of 4-5 ms. As a final step, we can see how this change in spiking activity impacts the aggregate current dipole.

[ ]:
```
import matplotlib.pyplot as plt
from hnn_core.viz import plot_dipole
fig, axes = plt.subplots(2, 1, sharex=True, figsize=(6, 6),
                         constrained_layout=True)

window_len = 30  # ms
scaling_factor = 3000
dpls = [dpl_erp[0].smooth(window_len).scale(scaling_factor),
        dpl_all[0].smooth(window_len).scale(scaling_factor),
        dpl_sparse[0].smooth(window_len).scale(scaling_factor)]

plot_dipole(dpls, ax=axes[0], layer='agg', show=False)
axes[0].legend(['Default', 'Custom All', 'Custom Sparse'])
net_erp.cell_response.plot_spikes_hist(
    ax=axes[1], spike_types=['evprox', 'evdist'])
```

Appendix B

Firing pattern tutorial

An advantage of using detailed biophysical models over abstracted models is the ability to access activity at the level of individual synapses and dendrites. To support investigations centered on microscale properties of neurons, I created an interface to allow users to easily record membrane potentials and synaptic currents. The recordings are enabled with a single keyword argument to the simulate function, and all the outputs are stored in a structure that maps each recording to its associated compartment on the neuron model.

This tutorial can be found on the HNN-core examples documentation page:

https://jonescompneurolab.github.io/hnn-core/stable/auto_examples/index.html

plot_firing_pattern

[ ]:
```
# Authors: Mainak Jas <mjas@harvard.mgh.edu>
#          Nick Tolley <nick nicholas_tolley@brown.edu>

import os.path as op
import tempfile
```

Let us import `hnn_core`.

[ ]:
```
import hnn_core
from hnn_core import read_spikes, jones_2009_model, simulate_dipole
```

Now let's build the network. We have used the same weights as in the `evoked example <sphx_glr_auto_examples_plot_simulate_evoked.py>`.

[ ]:
```
import matplotlib.pyplot as plt

net = jones_2009_model()
```

`net` does not have any driving inputs and only defines the local network connectivity. Let us go ahead and first add a distal evoked drive. We need to define the AMPA and NMDA weights for the connections. An "evoked drive" defines inputs that are normally distributed with a certain mean and standard deviation.

[ ]:
```
weights_ampa_d1 = {'L2_basket': 0.006562, 'L2_pyramidal': 7e-6,
                   'L5_pyramidal': 0.142300}
weights_nmda_d1 = {'L2_basket': 0.019482, 'L2_pyramidal': 0.004317,
                   'L5_pyramidal': 0.080074}
synaptic_delays_d1 = {'L2_basket': 0.1, 'L2_pyramidal': 0.1,
                      'L5_pyramidal': 0.1}
net.add_evoked_drive(
    'evdist1', mu=63.53, sigma=3.85, numspikes=1, weights_ampa=weights_ampa_d1,
    weights_nmda=weights_nmda_d1, location='distal',
    synaptic_delays=synaptic_delays_d1, event_seed=274)
```

The reason it is called an "evoked drive" is it can be used to simulate waveforms resembling evoked responses. Here, we show how to do it with two proximal drives which drive current up the dendrite and one distal drive which drives current down the dendrite producing the negative deflection.

[ ]:
```
weights_ampa_p1 = {'L2_basket': 0.08831, 'L2_pyramidal': 0.01525,
                   'L5_basket': 0.19934, 'L5_pyramidal': 0.00865}
synaptic_delays_prox = {'L2_basket': 0.1, 'L2_pyramidal': 0.1,
                        'L5_basket': 1., 'L5_pyramidal': 1.}

# all NMDA weights are zero; pass None explicitly
net.add_evoked_drive(
    'evprox1', mu=26.61, sigma=2.47, numspikes=1, weights_ampa=weights_ampa_p1,
    weights_nmda=None, location='proximal',
    synaptic_delays=synaptic_delays_prox, event_seed=544)
```

Now we add the second proximal evoked drive and simulate the network dynamics with somatic voltage recordings enabled. Note: only AMPA weights differ from first.

[ ]:
```
weights_ampa_p2 = {'L2_basket': 0.000003, 'L2_pyramidal': 1.438840,
                   'L5_basket': 0.008958, 'L5_pyramidal': 0.684013}
# all NMDA weights are zero; omit weights_nmda (defaults to None)
net.add_evoked_drive(
    'evprox2', mu=137.12, sigma=8.33, numspikes=1,
    weights_ampa=weights_ampa_p2, location='proximal',
    synaptic_delays=synaptic_delays_prox, event_seed=814)

dpls = simulate_dipole(net, tstop=170., record_vsec='soma')
```

Here, we explain more details about the data structures and how they can be used to better interpret the data. The cell IDs (gids) uniquely define neurons in the network and are stored in the :class:`~hnn_core.Network` object as a dictionary

[ ]:
```
gid_ranges = net.gid_ranges
print(net.gid_ranges)
```

Simulated voltage in the soma and other cell sections are stored in :class:`~hnn_core.CellResponse` as a dictionary. The CellResponse object stores data produced by individual cells including spikes, voltages and currents.

[ ]:
```
trial_idx = 0
vsec = net.cell_response.vsec[trial_idx]
print(vsec.keys())
```

We can plot the firing pattern of individual cells by indexing with the gid

[ ]:
```
gid = 170
plt.figure(figsize=(4, 4), constrained_layout=True)
plt.plot(net.cell_response.times, vsec[gid]['soma'])
plt.title('%s (gid=%d)' % (net.gid_to_type(gid), gid))
plt.xlabel('Time (ms)')
plt.ylabel('Voltage (mV)')
plt.show()
```

Also, we can plot the spikes in the network and write them to text files. Note that we can use formatting syntax to specify the filename pattern with which each trial will be written ('spk_1.txt', 'spk_2.txt, ...). To read spikes back in, we can use wildcard expressions.

[ ]:
```
net.cell_response.plot_spikes_raster()
with tempfile.TemporaryDirectory() as tmp_dir_name:
    net.cell_response.write(op.join(tmp_dir_name, 'spk_%d.txt'))
    cell_response = read_spikes(op.join(tmp_dir_name, 'spk_*.txt'))
cell_response.plot_spikes_raster()
```

We can additionally calculate the mean spike rates for each cell class by specifying a time window with `tstart` and `tstop`.

[ ]:
```
all_rates = cell_response.mean_rates(tstart=0, tstop=170,
                                     gid_ranges=net.gid_ranges,
                                     mean_type='all')
trial_rates = cell_response.mean_rates(tstart=0, tstop=170,
                                       gid_ranges=net.gid_ranges,
                                       mean_type='trial')
print('Mean spike rates across trials:')
print(all_rates)
print('Mean spike rates for individual trials:')
print(trial_rates)
```

Finally, we can plot the soma voltage along with the spiking activity with raster plots and histograms for the pyramidal cells.

[ ]:
```
fig, axes = plt.subplots(3, 1, figsize=(5, 7), sharex=True)

for idx in range(10):  # only 10 cells per cell-type
    gid = gid_ranges['L2_pyramidal'][idx]
    axes[0].plot(net.cell_response.times, vsec[gid]['soma'], color='g')
    gid = gid_ranges['L5_pyramidal'][idx]
    axes[0].plot(net.cell_response.times, vsec[gid]['soma'], color='r')
net.cell_response.plot_spikes_raster(ax=axes[1])
net.cell_response.plot_spikes_hist(ax=axes[2],
                                   spike_types=['L5_pyramidal',
                                                'L2_pyramidal'])
```

Appendix C

HNN animation tutorial

One of the most challenging aspects of working with detailed biophysical models is creating appropriate visualizations that capture important aspects of the simulation. To better understand the connection between single neuron spiking and network level dynamics, I created a visualization tool which animates HNN simulations by plotting the membrane potential over time on top of a 3-dimensional rendering of the model. These animations have proven to be highly effective in workshops and classroom settings for teaching the community about fundamentals of HNN simulations. One of the most striking outcomes of visualizing HNN simulations in this way is observing the spatial organization of spiking activity, specifically its tendency to spread across the cortical column in waves of activity. Cortical traveling waves are a well-documented emergent phenomenon observed in vivo, and these animations have provided direct motivation for new research directions linking HNN to traveling wave dynamics.

This tutorial can be found on the HNN-core examples documentation page:

https://jonescompneurolab.github.io/hnn-core/stable/auto_examples/index.html

plot_hnn_animation

```
[ ]: # Author: Nick Tolley <nicholas_tolley@brown.edu>
```

First, we'll import the necessary modules for instantiating a network and running a simulation that we would like to animate.

```
[ ]: import os.path as op

import hnn_core
from hnn_core import jones_2009_model, simulate_dipole, read_params
from hnn_core.network_models import add_erp_drives_to_jones_model
```

We begin by instantiating the network. For this example, we will reduce the number of cells in the network to speed up the simulations.

```
[ ]: net = jones_2009_model(mesh_shape=(3, 3))

# Note that we move the cells further apart to allow better visualization of
# the network (default inplane_distance=1.0 μm).
net.set_cell_positions(inplane_distance=300)
```

The :class:`hnn_core.viz.NetworkPlotter` class can be used to visualize the 3D structure of the network.

```
[ ]: from hnn_core.viz import NetworkPlotter

net_plot = NetworkPlotter(net)
net_plot.fig
```

We can also visualize the network from another angle by adjusting the azimuth and elevation parameters.

```
[ ]: net_plot.azim = 45
net_plot.elev = 40
net_plot.fig
```

Next we add event related potential (ERP) producing drives to the network and run the simulation (see `evoked example <sphx_glr_auto_examples_plot_simulate_evoked.py>` for more details). To visualize the membrane potential of cells in the network, we need use `simulate_dipole(..., record_vsec='all')` which turns on the recording of voltages in all sections of all cells in the network.

[ ]:
```
add_erp_drives_to_jones_model(net)
dpl = simulate_dipole(net, tstop=170, record_vsec='all')
net_plot = NetworkPlotter(net)  # Reinitialize plotter with simulated network
```

Finally, we can animate the simulation using the `export_movie()` method. We can adjust the xyz limits of the plot to better visualize the network.

[ ]:
```
net_plot.xlim = (400, 1600)
net_plot.ylim = (400, 1600)
net_plot.zlim = (-500, 1600)
net_plot.azim = 225
net_plot.export_movie('animation_demo.gif', dpi=100, fps=30, interval=100)
```

Appendix D

Simulate beta tutorial

One of the most useful aspects of open-source scientific code is the ability to directly build on upon the results in a publication, rather than attempt to translate the methodology to a custom code environment. While the latter is useful for assessing reproducibility, it is often subject to mistakes and creates an unnecessary block to beginning a new scientific investigation. In the context of HNN-core, I have created a framework for loading in base model classes associated with scientific publications using HNN-core. The default network `jones_2009_model()` is used in the majority of the tutorials, and corresponds to the network model from Jones et al. (2009). To further demonstrate the utility of this framework, I adapted the model from Law et al. (2022) so that is could be loaded in one line using `law_2022_model()`, and created a tutorial demonstrating the principle simulation results concerning the interaction of MEG/EEG beta events and event related potentials.

This tutorial can be found on the HNN-core examples documentation page:

https://jonescompneurolab.github.io/hnn-core/stable/auto_examples/index.html

References

Jones, S. R., Pritchett, D. L., Sikora, M. A., Stufflebeam, S. M., Hämäläinen, M., and Moore, C. I. (2009). Quantitative Analysis and Biophysically Realistic Neural Modeling of the MEG Mu Rhythm: Rhythmogenesis and Modulation of Sensory-Evoked Responses. *Journal of Neurophysiology*, 102(6):3554–3572. Publisher: American Physiological Society.

Law, R. G., Pugliese, S., Shin, H., Sliva, D. D., Lee, S., Neymotin, S., Moore, C., and Jones, S. R. (2022). Thalamocortical Mechanisms Regulating the Relationship between Transient Beta Events and Human Tactile Perception. *Cerebral Cortex*, 32(4):668–688.

plot_simulate_beta

[ ]:
```
# Authors: Nick Tolley <nicholas_tolley@brown.edu>

from hnn_core import simulate_dipole, law_2021_model, jones_2009_model
from hnn_core.viz import plot_dipole
```

We begin by instantiating the network model from Law et al. 2021 [1]_.

[ ]:
```
net = law_2021_model()
```

The Law 2021 model is based on the network model described in Jones et al. 2009 [2]_ with several important modifications. One of the most significant changes is substantially increasing the rise and fall time constants of GABAb-conductances on L2 and L5 pyramidal. Another important change is the removal of calcium channels from basal dendrites and soma of L5 pyramidal cells specifically. We can inspect these properties with the `net.cell_types` attribute which contains information on the biophysics and geometry of each cell.

[ ]:
```
net_jones = jones_2009_model()

jones_rise = net_jones.cell_types['L5_pyramidal'].synapses['gabab']['tau1']
law_rise = net.cell_types['L5_pyramidal'].synapses['gabab']['tau1']
print(f'GABAb Rise (ms): {jones_rise} -> {law_rise}')

jones_fall = net_jones.cell_types['L5_pyramidal'].synapses['gabab']['tau2']
law_fall = net.cell_types['L5_pyramidal'].synapses['gabab']['tau2']
print(f'GABAb Fall (ms): {jones_fall} -> {law_fall}\n')

print('Apical Dendrite Channels:')
print(net.cell_types['L5_pyramidal'].sections['apical_1'].mechs.keys())
print("\nBasal Dendrite Channels ('ca' missing):")
print(net.cell_types['L5_pyramidal'].sections['basal_1'].mechs.keys())
```

A major change to the Jones 2009 model is the addition of a Martinotti-like recurrent tuft connection [3]_. This new connection originates from L5 basket cells, and provides GABAa inhibition on the distal dendrites of L5 pyramidal cells.

[ ]:
```
print('Recurrent Tuft Connection')
print(net.connectivity[16])
```

The remaining changes to the connectivity was the removal of an L2_basket -> L5_pyramidal GABAa connection, and replacing it with GABAb.

[ ]:
```
print('New GABAb connection')
print(net.connectivity[15])

print('\nConnection Removed from Law Model')
print(net_jones.connectivity[10])
```

To demonstrate sensory depression, we will add the drives necessary to generate and ERP similar to `evoked example <sphx_glr_auto_examples_plot_simulate_evoked.py>`, but modified to reflect the parameters used in Law et al. 2021. Specifically, we are considering the case where a tactile stimulus is delivered at 150 ms. 25 ms later, the first input to sensory cortex arrives as a proximal drive to the cortical column. Proximal drive corresponds to projections from the direct thalamic nuclei. This is followed by one :term:`distal` drive representing projections from indirect thalamic nuclei, and a final late proximal drive. It is important to note that the parameter values for each are different from previous examples of the evoked response. This reflects the altered network dynamics due to the changes described above.

[ ]:
```
def add_erp_drives(net, stimulus_start):
    # Distal evoked drive
    weights_ampa_d1 = {'L2_basket': 0.0005, 'L2_pyramidal': 0.004,
                       'L5_pyramidal': 0.0005}
    weights_nmda_d1 = {'L2_basket': 0.0005, 'L2_pyramidal': 0.004,
                       'L5_pyramidal': 0.0005}
    syn_delays_d1 = {'L2_basket': 0.1, 'L2_pyramidal': 0.1,
                     'L5_pyramidal': 0.1}
    net.add_evoked_drive(
        'evdist1', mu=70.0 + stimulus_start, sigma=0.0, numspikes=1,
        weights_ampa=weights_ampa_d1, weights_nmda=weights_nmda_d1,
        location='distal', synaptic_delays=syn_delays_d1, event_seed=274)

    # Two proximal drives
    weights_ampa_p1 = {'L2_basket': 0.002, 'L2_pyramidal': 0.0011,
                       'L5_basket': 0.001, 'L5_pyramidal': 0.001}
    syn_delays_prox = {'L2_basket': 0.1, 'L2_pyramidal': 0.1,
                       'L5_basket': 1., 'L5_pyramidal': 1.}

    # all NMDA weights are zero; pass None explicitly
    net.add_evoked_drive(
        'evprox1', mu=25.0 + stimulus_start, sigma=0.0, numspikes=1,
        weights_ampa=weights_ampa_p1, weights_nmda=None,
        location='proximal', synaptic_delays=syn_delays_prox, event_seed=544)

    # Second proximal evoked drive. NB: only AMPA weights differ from first
    weights_ampa_p2 = {'L2_basket': 0.005, 'L2_pyramidal': 0.005,
                       'L5_basket': 0.01, 'L5_pyramidal': 0.01}
    # all NMDA weights are zero; omit weights_nmda (defaults to None)
```

```
    net.add_evoked_drive(
        'evprox2', mu=135.0 + stimulus_start, sigma=0.0, numspikes=1,
        weights_ampa=weights_ampa_p2, location='proximal',
        synaptic_delays=syn_delays_prox, event_seed=814)

    return net
```

A beta event is created by inducing simultaneous proximal and distal drives. The input is just strong enough to evoke spiking in the L2 basket cells. This spiking causes GABAb mediated inhibition of the network, and ultimately suppressed sensory detection.

[ ]:
```
def add_beta_drives(net, beta_start):
    # Distal Drive
    weights_ampa_d1 = {'L2_basket': 0.00032, 'L2_pyramidal': 0.00008,
                       'L5_pyramidal': 0.00004}
    syn_delays_d1 = {'L2_basket': 0.5, 'L2_pyramidal': 0.5,
                     'L5_pyramidal': 0.5}
    net.add_bursty_drive(
        'beta_dist', tstart=beta_start, tstart_std=0., tstop=beta_start + 50.,
        burst_rate=1., burst_std=10., numspikes=2, spike_isi=10,
        n_drive_cells=10, location='distal', weights_ampa=weights_ampa_d1,
        synaptic_delays=syn_delays_d1, event_seed=290)

    # Proximal Drive
    weights_ampa_p1 = {'L2_basket': 0.00004, 'L2_pyramidal': 0.00002,
                       'L5_basket': 0.00002, 'L5_pyramidal': 0.00002}
    syn_delays_p1 = {'L2_basket': 0.1, 'L2_pyramidal': 0.1,
                     'L5_basket': 1.0, 'L5_pyramidal': 1.0}

    net.add_bursty_drive(
        'beta_prox', tstart=beta_start, tstart_std=0., tstop=beta_start + 50.,
        burst_rate=1., burst_std=20., numspikes=2, spike_isi=10,
        n_drive_cells=10, location='proximal', weights_ampa=weights_ampa_p1,
        synaptic_delays=syn_delays_p1, event_seed=300)
    return net
```

We can now use our functions to create three distinct simulations: 1) beta event only, 2) ERP only, and 3) beta event + ERP.

[ ]:
```
beta_start, stimulus_start = 50.0, 125.0
net_beta = net.copy()
net_beta = add_beta_drives(net_beta, beta_start)

net_erp = net.copy()
net_erp = add_erp_drives(net_erp, stimulus_start)

net_beta_erp = net_beta.copy()
```

```
net_beta_erp = add_erp_drives(net_beta_erp, stimulus_start)
```

And finally we simulate. Note that the default simulation time has been increased to 400 ms to observe the long time course over which beta events can influence sensory input to the cortical column.

[ ]:
```
dpls_beta = simulate_dipole(net_beta, tstop=400)
dpls_erp = simulate_dipole(net_erp, tstop=400)
dpls_beta_erp = simulate_dipole(net_beta_erp, tstop=400)
```

By inspecting the activity during the beta event, we can see that spiking occurs exclusively at 50 ms, the peak of the gaussian distributed proximal and distal inputs. This spiking activity leads to sustained GABAb mediated inhibition of the L2 and L5 pyrmaidal cells. One effect of this inhibition is an asymmetric beta event with a long positive tail.

[ ]:
```
import matplotlib.pyplot as plt
import numpy as np
fig, axes = plt.subplots(4, 1, sharex=True, figsize=(7, 7),
                         constrained_layout=True)
net_beta.cell_response.plot_spikes_hist(ax=axes[0], show=False)
axes[0].set_title('Beta Event Generation')
plot_dipole(dpls_beta, ax=axes[1], layer='agg', tmin=1.0, color='b', show=False)
net_beta.cell_response.plot_spikes_raster(ax=axes[2], show=False)
axes[2].set_title('Spike Raster')

# Create a fixed-step tiling of frequencies from 1 to 40 Hz in steps of 1 Hz
freqs = np.arange(10., 60., 1.)
dpls_beta[0].plot_tfr_morlet(freqs, n_cycles=7, ax=axes[3])
```

Next we will inspect what happens when a sensory stimulus is delivered 75 ms after a beta event. Note that the delay time for a tactile stimulus at the hand to arrive at the cortex is roughly 25 ms, which means the first proximal input to the cortical column occurs ~100 ms after the beta event.

[ ]:
```
dpls_beta_erp[0].smooth(45)
fig, axes = plt.subplots(3, 1, sharex=True, figsize=(7, 7),
                         constrained_layout=True)
plot_dipole(dpls_beta_erp, ax=axes[0], layer='agg', tmin=1.0, color='r',
            show=False)
axes[0].set_title('Beta Event + ERP')
net_beta_erp.cell_response.plot_spikes_hist(ax=axes[1], show=False)
axes[1].set_title('Input Drives Histogram')
net_beta_erp.cell_response.plot_spikes_raster(ax=axes[2], show=False)
axes[2].set_title('Spike Raster')
```

To help understand the effect of beta mediated inhibition on the response to incoming sensory stimuli, we can compare the ERP and spiking activity due to sensory input with and without a beta event. The sustained inhibition of the network ultimately depresses the sensory response which is associated with a reduced ERP amplitude

[ ]:
```
dpls_erp[0].smooth(45)
fig, axes = plt.subplots(3, 1, sharex=True, figsize=(7, 7),
                         constrained_layout=True)
plot_dipole(dpls_beta_erp, ax=axes[0], layer='agg', tmin=1.0, color='r',
            show=False)
plot_dipole(dpls_erp, ax=axes[0], layer='agg', tmin=1.0, color='b', show=False)
axes[0].set_title('Beta ERP Comparison')
axes[0].legend(['ERP + Beta', 'ERP'])
net_beta_erp.cell_response.plot_spikes_raster(ax=axes[1], show=False)
axes[1].set_title('Beta + ERP Spike Raster')
net_erp.cell_response.plot_spikes_raster(ax=axes[2], show=False)
axes[2].set_title('ERP Spike Raster')
plt.show()
```

0.1 References

.. [1] Law, R. G., Pugliese, S., Shin, H., Sliva, D. D., Lee, S., Neymotin, S., Moore, C., & Jones, S. R. (2021). Thalamocortical mechanisms regulating the relationship between transient beta events and human tactile perception. BioRxiv, 2021.04.16.440210. https://doi.org/10.1101/2021.04.16.440210 .. [2] Jones, S. R., Pritchett, D. L., Sikora, M. A., Stufflebeam, S. M., Hämäläinen, M., & Moore, C. I. (2009). Quantitative Analysis and Biophysically Realistic Neural Modeling of the MEG Mu Rhythm: Rhythmogenesis and Modulation of Sensory-Evoked Responses. Journal of Neurophysiology, 102(6), 3554–3572. https://doi.org/10.1152/jn.00535.2009 .. [3] Silberberg, G., & Markram, H. (2007). Disynaptic Inhibition between Neocortical Pyramidal Cells Mediated by Martinotti Cells. Neuron, 53(5), 735–746. https://doi.org/10.1016/j.neuron.2007.02.012