UNIVERSITY OF CALIFORNIA SAN DIEGO

Network Designs Via Signaling Dynamics On Geometric Dynamic Graphs

A dissertation submitted in partial satisfaction of the
requirements for the degree
Doctor of Philosophy

in

Bioengineering with a Specialization in Computational Neuroscience

by

Vivek Kurien George

Committee in charge:

Professor Gabriel Silva, Chair
Professor Henry Abarbanel
Professor Gert Cauwenberghs
Professor Timothy Gentner
Professor Shankar Subramaniam

2019

The dissertation of Vivek Kurien George is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

_____

_____

_____

_____

_____

Chair

University of California San Diego

2019

DEDICATION

To Fatima, without whom I would certainly be less. You enrich my life in innumerable ways. I appreciate your sacrifice, patience, kindness, and good-humor. You are the light which feeds the sun.

EPIGRAPH

The endeavor to understand is the first and only basis of virtue.

—Baruch Spinoza

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

Thanks to Dr. Gabriel Silva for giving me a space to think and develop ideas. I am grateful to Dr. Henry Abarbanel for introducing me to the ideas and concepts of data assimilation among a host of others which has enriched my experience. I would also like to thank all the current and past members of the Mathematical Neuroscience Laboratory. The cross-pollination of ideas through our discussions has been a constant source of inspiration.

Chapter 2, is currently being prepared for submission for publication. Vivek George, Francesca Puppo, and Gabriel Silva. This dissertation author was the primary investigator of this paper. The chapter's presentation is as it may appear online.

Chapter 3, is currently being prepared for submission for publication. Vivek George, Arkin Gupta and Gabriel Silva. This dissertation author was the primary author of this paper. The chapter's presentation is as it may appear online.

Chapter 4, in full, is a reprint of Francesca Puppo, Vivek George and Gabriel Silva. This article was published online by the journal Nature Scientific Reports on July 11 2018. This dissertation author was the second author of this paper.

VITA

2006	B. S. in Electrical Engineering *magna cum laude*, Florida Institute of Technology
2006	B. S. in Mathematical Sciences *magna cum laude*, Florida Institute of Technology
2006-2008	Research Engineer
2008-2012	Research and Development Technical Project Manager
2019	Ph. D. in Bioengineering with a Specialization in Computational Neuroscience, University of California, San Diego

PUBLICATIONS

Vivek K. George, Francesca Puppo, Gabriel A. Silva. "An Analysis of Network Dynamics: Application to the C. Elegans Connectome", 2019. [*Manuscript in preparation*]

Vivek K. George, Arkin Gupta, Gabriel A. Silva. "Vector Embedding of Network Dynamics Using Steady-State Statistics", 2019. [*Manuscript in preparation*]

Francesca Puppo, Vivek K. George, Gabriel A. Silva. "An Optimized Structure-Function Design Principle Underlies Efficient Signaling Dynamics in Neurons", *Nature Scientific Reports*, July 2018, 8(1):1-15.

ABSTRACT OF THE DISSERTATION

Network Designs Via Signaling Dynamics On Geometric Dynamic Graphs

by

Vivek Kurien George

Doctor of Philosophy in Bioengineering with a Specialization in Computational Neuroscience

University of California San Diego, 2019

Professor Gabriel Silva, Chair

Artificial neural networks are treated as black boxes. Generally,only the states of a subset of the network are considered to determine its efficacy, while the relationship between a neural network's topology and its function remains under-theorized. For my analysis, I use a new class of event-driven recurrent neural networks—a geometric dynamic network modeled on canonical neurobiological signaling principles that allows to directly encode input data into its evolving dynamics—to forward a new type of machine learning approach. I accomplish this by first, mapping causal neuronal signal flows in the C. elegans connectome to show how the dynamic evolution of signal flows results in a unique internal representation of particular input data. Second, I propose two distinct approaches to determine the upper-bound for the

amount of network dynamics needed for capturing the signaling evolution of the system. Using the upper-bound values, I construct a mathematical object representing the causal neuronal signaling dynamics, and delineate the interaction of sub-sub structures at various scales/heights of sub-graphs. Finally, based on recent theoretical propositions regarding optimal signaling in a geometric dynamic network, I show that neurons modify their axonal morphology so that the propagation time of an action potential, and the membrane's refractory period become balanced. Thus, this work not only lays the foundation to construct and analyze a new class of artificial neural networks whose overall behavior and underlying dynamics are transparently coupled, it also provides fertile grounds for future work on biologically inspired artificial intelligence.

Chapter 1

Introduction

Alan Turing formalized the question “Can Machines Think?” [24]. The wonder evoked by human cognition and the elements which give to it has inspired many researchers. As such there is coupling between neuroscience and artificial intelligence [6]. From study of the nervous system, Warren McCulloch, a neuroscientist, and Walter Pitts, a logician, proposed a logical neuron model (MC model) which enabled the construction of hypothetical neuronal networks with some required properties[12]. They popularized the construction and analysis of networks built from an abstraction of the basic elements [14].

Frank Rosenblatt extended the MC model by proposing the perceptron model [16] which acted as a linear classifier by introducing variable weights between nodes in the network and supervisory feedback. Then multilayer perceptrons with back-propagation [17, 19, 18] were proposed to overcome the limitations of the original perceptron model [14, 13]. Virtually all current widely used methods modify synaptic weights based on some form of back-propogation of errors. Equipped with the universal approximation theorem [4, 7], Turing computability [21] and error based training methods artificial neural networks have found much success in various fields [20, 9, 5].

One way to partition the field of artificial neural networks is by considering spiking

[11, 10, 15, 23] and non-spiking node models. Generally, both approaches modify synaptic weights during training. The non-spiking node models are continuous and differentiable, therefore, amenable to backpropogation of errors, and their output is clearly stated by the activation of some "output" nodes [20]. Spiking neural networks are the more computationally powerful, in terms of number of neurons needed, than their non-spiking counterparts [11], but they are more difficult to train, especially, multi-layer networks.

The focus of this dissertation is to turn around the paradigm of training networks, with respect to output nodes, by focusing on the structure of dynamics within a network rather than the dynamics of particular network structures: instead of analyzing the state of output nodes, we analyze the neuronal signalling pathways carved by signals through a network through the mathematical objects which arise from the causal interactions. We consider the symphony of activity present on the network. Concretely, we view the network as a directed geometric graph, and consider the walks of signals on that graph. Since the specific underlying node models matter little, the perspective taken may have wider applicability.

Thesis Outline

In what follows we provide a brief outline of the contents of each of the following chapters. In Chapter 2 we used the C. Elegans neural connectome as our test bed. It is one of the few organisms whose neuronal connectivity is known and it is amenable to physical experimentation [3, 8]. We constructed a directed geometric graph using the available data for neuron locations, synaptic connectivity, and node type (either inhibitory or excitatory). The node model used was a simplified version of the Geometric Dynamic Percpetron [22]. The simplified node model did not consider variable synaptic weights, it had a fixed refractory period, it generated a traveling pulse (Driac pulse) upon receiving an input during its non-refractory state and it was deterministic in nature. Using these very simple assumptions, we stimulated ASEL with an impulse which resulted

in a rich repertoire of network activity. Due to cycles in the graph the spiking activity eventually seemed to settle into a periodic orbit. Notably, we observed semblances of contralateral firing between sets of ventral and dorsal motor neurons. These observed activation patterns resembles those which are necessary for animal movement. By this observation our interest had peaked and we wanted to delve into how the signals were transformed from stimulus to response. This lead us to consider the theoretical causal neuronal signaling paths which resulted in previously said firing patterns. To do so we introduced the notion of a Temporal Sequence(TS). It is the causal chain of node activations from some set of nodes in the network, to some other set of nodes(not necessarily mutually exclusive). Every motor neuron activation had an associated TS. Because the network activity entered a periodic orbit we were able to decompose the rise of all the spiking network activity into a basis set of TSs. The basis set contains unique TSs, and repeating TSs. By determining the generators producing cyclic sub-sequences and creating a compressed version of the original TSs we were able to classify them as either unique or repeating. TSs which had the same compressed representations were considered repeats, those TS without match were considered unique. We decomposed all the Temporal Walks from ASEL to the VB and DB motor-neurons into the basis sets. Furthermore, we used TSs as the basis of comparison of different C. Elegans networks to determine the efficacy of sub-networks in terms of being able to preserve causal neuronal signaling paths.

In Chapter 3 we proposed methods to determine when to stop collecting dynamics, that is to determine the Steady-State Statistics of the System, and how to the embed dynamics into a well studied mathematical form, a directed acyclic graph, where a plethora of tools may then be applied to further study the structure of said dynamics. In considering a simple classification problem based on the dynamics in a network one will readily see that it is difficult to determine an upper bound for the time interval of dynamics to consider. An arbitrary stopping time can be considered, but that is not satisfactory. In this paper we propose three methods(the best of which has a computational complexity of O(N) where N is the number of time samples considered) to

determine when the network dynamics enter into steady-state, that is a periodic orbit, and the size of the steady-state period. Using the steady-state statistics we propose a graph construction of the dynamics. Our research interest in decomposing the dynamics into its constituent subgraph structures and the interaction of those subgraphs in order to further study the link between network design and dynamics. We used the Weisfeller-Lehman graph relabeling scheme to ascertain the induced subgraph interactions. This approach not only delineated the prevalence of subgraphs and their interactions but resulted in a vector space embedding of the dynamics. A vector embedding enables the use several existing methods as a back-end for classification and even clustering. This work proposes an end-to-end workflow for the analysis of network dynamics for the purposes of machine learning.

In Chapter 4 we experimentally verify a neurophysiologically inspired theoretical local network design rule [22]. Previous works hypothesised that a neuron's physiology is optimized for the minimization of bio-cellular material while balancing the maintenance of temporal precision [2]. This indicates a balance between the neuron's morphology and its signalling capabilities. We analyzed optical data [1] of a set of Basket cell neurons from the rat neocortex. We found a balance between the physical parameters which affects action potential signal latency (from the axon initial segment to axon terminal) and the membrane refractory period of the individual neurons. Our analysis indicates that the morphology of a neuron's axonal arbours preserved the optimal signaling ratio [22]. The optimal ratio is defined as the time it takes for the propagation of an action potential from the axon initial segment to the synaptic terminals divided by the refractory period.

In Chapter 5 we conclude this work by exploring several future directions of investigation.

References

[1] Giorgio A. Ascoli, Duncan E. Donohue, and Maryam Halavi. NeuroMorpho.Org: A Central Resource for Neuronal Morphologies. *Journal of Neuroscience*, 27(35):9247–9251, August 2007.

[2] Julian Budd and Zoltan F. Kisvarday. Communication and wiring in the cortical connectome. *Frontiers in Neuroanatomy*, 6, 2012.

[3] Ann K. Corsi, Bruce Wightman, and Martin Chalfie. A Transparent Window into Biology: A Primer on Caenorhabditis elegans. *Genetics*, 200(2):387–407, June 2015.

[4] G Cybenkot. Approximation by superpositions of a sigmoidal function. page 12.

[5] Jeffrey Dean, Greg Corrado, Rajat Monga, Kai Chen, Matthieu Devin, Mark Mao, Marc\textquotesingle aurelio Ranzato, Andrew Senior, Paul Tucker, Ke Yang, Quoc V. Le, and Andrew Y. Ng. Large Scale Distributed Deep Networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1223–1231. Curran Associates, Inc., 2012.

[6] Demis Hassabis, Dharshan Kumaran, Christopher Summerfield, and Matthew Botvinick. Neuroscience-Inspired Artificial Intelligence. *Neuron*, 95(2):245–258, July 2017.

[7] Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4(2):251–257, January 1991.

[8] Stephen D. Larson, Padraig Gleeson, and André E. X. Brown. Connectome to behaviour: modelling Caenorhabditis elegans at cellular resolution. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 373(1758):20170366, October 2018.

[9] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, May 2015.

[10] Jun Haeng Lee, Tobi Delbruck, and Michael Pfeiffer. Training Deep Spiking Neural Networks Using Backpropagation. *Frontiers in Neuroscience*, 10, 2016.

[11] Wolfgang Maass. Networks of spiking neurons: The third generation of neural network models. *Neural Networks*, 10(9):1659–1671, December 1997.

[12] W. S. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. 1943. *Bulletin of Mathematical Biology*, 52(1-2):99–115; discussion 73–97, 1990.

[13] Marvin Minsky and Seymour Papert. *Perceptrons*. Perceptrons. M.I.T. Press, Oxford, England, 1969.

[14] Marvin L. Minsky. *Computation: Finite and Infinite Machines*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1967.

[15] Danilo J. Rezende, Daan Wierstra, and Wulfram Gerstner. Variational Learning for Recurrent Spiking Networks. In J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, pages 136–144. Curran Associates, Inc., 2011.

[16] F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386–408, 1958.

[17] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1. pages 318–362. MIT Press, Cambridge, MA, USA, 1986.

[18] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, October 1986.

[19] David E. Rumelhart and James L. McClelland. Learning Internal Representations by Error Propagation. In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition: Foundations*. MITP, 1987.

[20] Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural Networks*, 61:85–117, January 2015.

[21] Hava T. Siegelmann and Eduardo D. Sontag. Turing computability with neural nets. *Applied Mathematics Letters*, 4(6):77–80, 1991.

[22] Gabriel A. Silva. The Effect of Signaling Latencies and Node Refractory States on the Dynamics of Networks. *arXiv:1804.07609 [q-bio]*, August 2019. arXiv: 1804.07609.

[23] Amirhossein Tavanaei, Masoud Ghodrati, Saeed Reza Kheradpisheh, Timothee Masquelier, and Anthony S. Maida. Deep Learning in Spiking Neural Networks. *Neural Networks*, 111:47–63, March 2019. arXiv: 1804.08150.

[24] A. M. Turing. Computing Machinery and Intelligence. *Mind, New Series*, 59(236):433–460, 1950.

Chapter 2

Mathematical Structures Describing Signaling Dynamics

2.1 Abstract

We construct and analyze the signaling dynamics on geometric dynamic networks whose elements are nodes and edges which are embedded in spatial coordinates and support signaling. The network's dynamics follow canonical neurobiological principles where the finite velocity for a discrete signal on an edge determines the signaling latency per edge, which combined with the refractory period of a node, results in the Geometric Dynamic Network Signaling Framework (GDNSF)[58]. Using this framework, we construct and analyze Temporal Sequences (TSs)—time-ordered walks of signals on a graph—and we show how perturbations are transformed by the network's structure into subsequent network activity through various Neuronal Signaling Pathways (NSPs) encapsulated through these TSs. Without needing detailed biophysical models, we apply the GDNSF and TSs to analyze the C. elegans connectome[67, 31, 64]. We show that a food stimulus elicits contralateral mid-body motor neuron activations, responsible for the worm's movement[51]. Also, we find that NSPs, in an isolated C. elegans subnetwork[71, 69], are

drastically different from those in the complete network; reminding us that when analyzing biological subnetworks, the whole is more than the sum of its parts[35]. Finally, we elucidate the necessary set of nodes to preserve the NSPs by decomposing the ensemble NSPs into a fundamental set of basis TSs. In sum, we find that C. elegans connectome when considered under the GDNSF, conserves a behaviorally relevant response of the animal, and through TSs we chart the responsible NSPs between stimulus and response. This work sets the grounds to construct and analyze other networks with discrete signaling dynamics. Our case study generates experimentally testable hypothesis and motivates gathering more detailed measurements of the C. elegans' signaling parameters to better inform models and generate experimentally testable hypotheses. Furthermore, we propose to expand the analytical reach of network analysis techniques to include mathematical structures, such as TSs, to augment existing approaches.

2.2 Introduction

Network analysis is a widely used modality to analyze systems[48, 64, 19, 7, 10, 9, 59]. To contextualize the methods developed in this paper we used C. elegans as our model organism since it has been extensively studied[23, 29, 16]. Its connectome, the anatomical links/edges between all constituent neurons, was fully mapped in 1986[67]. Although the organism's connectome is known, how neurons behave in context of the entire network and regulate one another is not fully understood[6, 5], and it is precisely this aspect which we intend to discuss in this paper[38].

Recent computational research efforts are moving past structural analysis[64, 62], and towards understanding dynamics[33, 11, 55, 47, 61, 70, 34]. We consider how concurrent network activity based on neuron-to-neuron interaction gives rise to causal neuronal signaling paths between nodes in the network. To quantify the signaling paths we introduced the notion of Temporal Sequences (TSs). Each TS is a sequence of causal node activations, more formally a walk on a graph. The study of the effects of network geometry, edge lengths based on spatial node

locations, and topology, network connectivity, on dynamics is a key challenge in the analysis of complex geometric networks.

We studied TSs resulting from stimulus at the ASEL neuron (implicated in chemosensation and resulting in forward bouts of movement[60, 65] and the activations of mid-body motor neurons(implicated in movement [30, 69] (Fig. 2.1A). Using TSs we showed qualitative motorneuron neuronal activity patterns associated with animal movement behaviour, performed quantitative comparisons of the dynamics between networks to quantify the differences in signaling paths, and quantified the number of different causally related classes of neurons.

We performed the analysis using a computational model (Online Methods 2.5.4) based on experimentally derived network connectivity[67] and geometry[17, 31]. Rather than building a network from the ground up using biophysical models[61, 57, 34, 37] which is computationally expensive[25], requires estimates of ion-channel parameters based on experimental data[43] and weights, as well as a whole host of other parameters affecting the electrical characteristics of individual neurons[36], our model approaches the node and network construction from a different perspective. We focus on a minimal set of signaling parameters(signal conduction velocity, spatial node locations, node refractory period, and node type which is either inhibitory or excitatory) and assumptions necessary to simulate signaling dynamics[58]. The nodal dynamics use a simplified version of the Geometric Dynamic Perceptron Model[58].

The general aims of this work are to inform C. elegans research, propose TSs as a way of analyzing network dynamics, show how network dynamics can be decomposed into a fundamental set of sequences which describe all the observed activity of the network. Our computational framework can be used to generate experimentally testable hypothesis[21] regarding the concurrent signaling paths supported by the connectome. TSs can also be used to generate a set of candidate sub-networks, that preservers the signaling paths, upon which more computationally expensive models can be implemented. Finally, our future research focuses on extending the analysis of causal dynamics for machine learning. For example, in a general network setting

where we wish to encode images through the causal dynamics, we analyze the statistics of the resultant TSs, and more general mathematical objects, such as, the statistics of causal network motifs[46, 2, 68, 18].

2.3 Results

2.3.1 The role of Network Geometry: Lattice Comparison

Geometric information plays an important role in network dynamics[58]. Endowing edges with signaling delays, due to physical geometry, adds richness to the network dynamics due to the differentials in signal arrival times at nodes. To quantify the the affect of signal delays/geometry we compared the network activity on the geometrically embedded C. elegans connectome (Fig. 2.1A), henceforth referred to as the Full network (Online Methods: 2.5.4), and a latticed version of the C. elegans network, henceforth referred to as the Lattice network (Online Methods: 2.5.4). While the Lattice network's edge connectivity and node types (inhibitory or excitatory) are identical to the Full network, it's edge signal delays are all the same; this extinguishes the affect of physical distances/geometry.

We compared the resulting network activity between the Full and Lattice networks due to a single pulse stimulus of the ASEL neuron (Fig. 2.1B,C). In the Lattice network, we observed that the network activity went through some transient period of node activations, then, eventually, all nodes which will fire, fire at the same time in a periodic manner (Fig. 2.1B). In contrast, although the Full network's node activity exhibited a transient and periodic activity phases, we observed more richness in its network state repertoire (its node activation patterns were richer) (Fig. 2.1C). To quantify the variability in network activity due to geometry, we counted the number of unique network states. We describe the network's state through a vector of nodal states at each discrete simulation time point. Concretely, if the state of node i at time t is given by $y_i(t) = \{0,1\}$, then the state of a network, with N nodes, at time t, can be written as $\mathbf{y}(t) = \{0,1\}^N$. A plot of the

cumulative count of unique network states over the course of the simulation is shown in Figure (2.1D). The Lattice network assumed 12 unique states, while the C. elegans network assumed 280 unique states across the sampling points (we did not consider the permutations of consecutive states). Thus geometry is an important consideration because it increases the dynamical range of the system.

2.3.2 Network Dynamics in Geometric Networks: Qualitative Comparison of Network Activity From the Full and Feed Networks Using Temporal Sequences.

We inspect the signaling dynamics of the network by constructing and analyzing so called "Temporal Sequences" (TSs) (Online Methods: 2.5.1). A TS represents the causal nodal interactions between a set of start nodes and a set of end nodes. Each TS is a walk, of an individual signal, on the graph from some start node to some end node. TSs capture the signaling pathways of the network, as such, they enables us to chart the course of a stimulus or any signal through the connectome. Only a subset of all possible walks on a graph are realizable at any given time. This is due to network parameters such as delays along edges, refractory periods of the nodes, and concurrent network activity. We focused on the TSs which start at the ASEL neuron, and either terminate or traverse the ventral, VB, or dorsal, DB, classes of motor neurons. To illustrate the TSs in the context of the connectome we traced the TSs from ASEL to neurons in the VB and DB classes, the color sequence represents the time course of the signal (Fig. 2.1A).

To better visualize TSs, we created a so called TS plot (Online Methods: 2.5.2). The TS plot shows the evolution of a set of TSs over time. The plot contains node labels on the y-axis, and time on the x-axis. Each curve in the TS plots traces the causal walk of a signal takes on the structural connectome leading to a motor neuron firing. Considering all curves active in a time interval on the TS plots trace the concurrent neuronal signaling pathways leading to subsequent

activations of motor neurons from the neuronal classes of interest. To lessen the visual clutter in the TS plot we do not explicitly mark the nodes on the walk. The activated nodes are generally located at the notches or discontinuities along the curve.

Using TS plots (Fig. 2.2) and histograms (Fig. 2.3) of the node activations, we qualitatively compared network dynamics of the C. elegans connectome, the Full Network, to a sub-network of the C. elegans connectome, henceforth called the Feed Network (Online Methods: 2.5.4). The Feed Network[69] is thought to contain neurons implicated in the feeding behavior of the animal, as such, contains relevant pathways from ASEL neuron to the VB and DB classes of motorneurons. Therefore, we do not show all the TSs in the network, but only those which contain ASEL and the motor neurons from the classes of interest. The locations of the motor neurons(end nodes) on the TS plot are indicated by the dark blue horizontal lines.

Stimulating ASEL in both the Feed and Full networks qualitatively resulted in back and forth motor neuron firings of the VB and DB classes (Figs. 2.3A,C). This is indicative of the contralateral firing patterns[71, 30, 53] expected for animal movement. To uncover how a single pulse stimulation of the ASEL neuron resulted in the motor neuron firings in the VB and DB classes, we decomposed the network activity into TSs (Figs. 2.2A,C).

Next we qualitatively analyzed the network activity which results from disrupting the connectivity of the network through edge randomization (Online Methods: 2.5.4). The randomization procedure we used[26, 22] preserves the spatial locations of the nodes but replaces the original set of edges with a new set which is some subset of the total number of allowed edges. Although the structural backbone of the connectome plays an important role in propagating the stimulus signal to the VB and DB sets of motor neurons, we observed that disruption of network structure through edge randomization resulted in eventual formation of TSs, in both the Feed and Full networks, linking the stimulus at the sensory neuron to motor neurons activations (see Figs. 2.2B,D). While TSs exist which links the stimulus and motor neuron classes, and seem to eventually settle into periodic activity patterns, the synchronized contralateral motor neuron

firings between the VB and Db classes found in the non-randomized networks (Figs. 2.3A,C) are decimated in the randomized Feed and Full networks (Figs. 2.3B,D).

Another notable qualitative difference between the TSs of the edge randomized and original networks are that the TSs enter periodic regime of activity significantly sooner. Additionally, the TSs in the randomized network displayed circuitous and complex neuronal signaling pathways relative to the orderliness of signaling pathways of the non-randomized networks. Further analysis of TSs could elucidate signaling pathways, significant neuronal hubs of activity[62], and other important features.

2.3.3 Quantitative Comparison of Resulting Temporal Sequences By Similarity Measure

Similarity Measure

We introduce a so called Similarity Measure (SM) as a quantitative approach to compare sets of TSs. These TSs can either result from two different networks, or from two perturbations of the same network. We considered the case of TSs resulting from two different networks. The quantitative comparison technique is similar to graph edit distances [56]. In order to calculate the SM a reference network must be chosen. The SM is calculated relative to the length of each TS from the reference network. We then count the number of TSs meeting some matching criteria α, where α is the percent of a sequence from one set matching some sequence from another set.

Concretely, lets say we wish to calculate SM between network $\mathcal{X}$ and network $\mathcal{Y}$, where $\mathcal{X}$ is the reference. The sets X and Y respectively contain the TSs from $\mathcal{X}$ and $\mathcal{Y}$. Let the i^{th} TS of X be denoted x_i, and the j^{th} TS of Y be denoted y_j. Each comparison of the TS is relative to a threshold α which is a percentage of total elements in a particular x_i which in-order matched some y_j (Algorithm 1, getTSSimilarityMeasure). For each $x \in X$, we evaluate:

$$SM(x_i, y_j) \geq \alpha \tag{2.1}$$

With respect to $y \in Y$. When (2.1) is TRUE, a counter, c_α, is incremented, and the next TS from X, x_{i+1} is checked against $y \in Y$. The final value of the counter, c_α, is number of TSs from $\mathcal{X}$ which met the criteria α relative to the TSs from $\mathcal{Y}$. For example, if 4 out of 5 node activations in some x_i found an in-order match to some y_j, and 4 out of 5 was the best match for all $y \in Y$, then we say x_i meets the $\alpha = 0.8$ criteria, and the counter $c_{0.8}$ is incremented by 1. We outline the Similarity Measure in Algorithm (1). The inputs to the algorithms are (X, Y, α). The output of this algorithms is the SM. The SM result can be stated words as follows: Given network $\mathcal{X}$ as the reference, $n_\mathcal{X}$ TSs out of a total of $m_\mathcal{X}$ TSs from network $\mathcal{X}$, found a α match to some subset of TSs of out a total of $m_\mathcal{Y}$ TSs from network $\mathcal{Y}$. Upon completion of this procedure, we can ascertain the percent of TSs from $\mathcal{X}$ finding a c_α match by normalization: $c_\alpha / m_\mathcal{X}$.

Networks Considered For Similarity Measure

Before we discuss the SM results, we describe the various networks we considered and, in brief, how we constructed them. We analyzed the SM between several networks, all derived from the Full network. As discussed in a previous section, the Feed network contains a subset of the vertices of the Full network, and it maintains the edge connectivity between those vertices.

We constructed randomized counterparts of the Feed and the Full networks using two edge randomization procedures. We chose two different edge randomization procedures in the case of results biased by the choice of the randomization procedure. The two edge randomization procedures resulted in varying degrees of edge reorganization. The less drastic network reorganization of the two randomization schemes was the Edge-Swap based randomization (Online Methods: 2.5.4). It is less drastic because only existing edges have some probability of swapping the edge's terminal node [20]. The Gilbert based edge randomization [26] was a more drastic network reorganization because its network construction was only constrained to maintain approximately

the total number of edges as the original network. This was accomplished by setting a probability of connection between two nodes.

Finally, we considered a so called Embedded Random Feed networks (Online Methods: 2.5.4). These network preserves the edge connections of the Full network outside of the Feed subnetwork. The subnetwork within the Full network uses the same randomized edge connection profile as the standalone randomized Feed networks.

Similarity Measure Results

The results of the SM are presented in Table 2.1. The reference networks for each SM are the antecedents in each of the column heading in the table. The top-half of Table 2.1 contains the SM results of the Gilbert based randomization, and the bottom-half of the table contains the SM results of the Edge-Swap based randomization. In each half of the table, we compared the TSs resulting from each of the five network types described earlier. Stimulus was applied at the ASEL sensory neuron in all the networks, and the TSs of interest were all those which traversed the VB and DB sets of motor neurons. A general observed trend was that as the α-criteria increased, the number of TSs meeting the α-criteria decreased. This is expected as the variation in the structural network will cause variation in TSs. Therefore, fewer TSs from more structurally different networks will meet a high α value.

In Table 1, we show that as the structure of networks diverge the SM values decreases. The SM between the reference Feed network and the Full network was greater than the SM between both the Feed and Full and either of their edge randomized counterparts. This is because the Feed network was more directly derived from the Full network. The SM values between the reference Full/Feed networks and their Edge-Swap Randomized networks were higher than the SM between the Full/Feed networks and their Gilbert Randomized networks. This was because the Edge-Swap Random networks preserve various network parameters, such as, out-degree, network connectedness. The edge-swapping procedure is based on flipping a weighted coin for

the chance to swap an edge, therefore, not all edges need actually be different. Additionally, based on the number of iterations one chooses to run the edge-swapping procedure the network will steadily structurally deviate from the original. In contrast the Gilbert based randomization procedure's only constraint is to maintain approximately the same number of edges between the original and the randomized networks.

Although the Feed network was derived from the Full network, the Feed network in isolation only completely preserved 6 out of 124 TSs, the remaining 118 TSs followed different neuronal pathways. Ideally, the TSs from a subnetwork should completely be represented in the full network. A natural question which arises is: Can we build a better subnetwork? Contained in the 697 TSs are all the nodes necessary for a subnetwork. But the sufficient set of nodes are those additional vertexes involved in the neuronal signaling pathways of the concurrent network activity which ensures the the existence of the 697 patterns. Determining the sufficient set of nodes, which require iterative simulation and testing, is out of the scope of this paper.

To bridge the dynamical regime of the isolated Feed network and the Full network one can take several approaches. In the context of the Full network removing all the vertices and edges of the Feed network, except for those associated with ASEL, VB, and DB, and analyzing the resulting TSs will result in the paths which completely lie outside the feed network, under the same necessary and sufficient conditions stated earlier. But removing all the vertices and edges associated with the Feed network from the Full network can drastically affect the overall dynamics of the network due to the absence of a set of concurrent dynamics. To workaround this issue, we incrementally randomize the edges in the Feed subnetwork within the context of the Full network.

We analyzed the TSs resulting from the reference Full network to the Embedded Random Feed network. Relative to the SM values resulting from the comparison of the isolated Feed network (vs. Full network), we observed higher SM values while comparing the reference Full network and the Embedded Random Feed networks (Table 2.1, 5^{th} column). Although there are

no TSs which have a 100% match when comparing the TSs from the reference Full network to the Embedded Random Feed network, across α values the Embedded Random Feed networks better preserved TSs than the reference isolated Feed network relative to the Full network.

Not only do these results paint a cautionary tale for the analysis of isolated subnetworks, but it also provides a network signaling based solution. As such we have a quantitative method for generating another Feed network, one which preserves the observed neuronal signaling pathways of the Full network. The SM procedure coupled with experimental observations and other theoretical considerations can provide a basis for choosing the constituents of a subnetwork.

2.3.4 Decomposing Temporal Sequences Into Their Basis Sequences

We decomposed the complex patterns of network activity into a set of basis TSs. We used the term basis to mean: a set of TSs which is composed of TSs which arise from signals performing One-Time walks on the graph, and TSs which arise from signals who perform Repeated walks on the graph by virtue of traversing cycles in the graph. From the set of One-Time walks and Repeated walks all other observed TSs can be recomposed.

To ascertain the basis set of TSs, we categorized each TS into One-Time Temporal Sequences, and Repeated Temporal Sequences (Online Methods: 2.5.3). The basis set for the Full Network showed that out of the 697 TSs which traversed the VB and DB set of neurons there were 50 One-Time TSs, and 20 Repeated TSs. We show the sequence of node activations which comprise each of the unique TSs in Fig. 2.4B. These patterns of node activity in each of the 50 TSs are unique in that these were no other other TSs out of the 697 which were sub-sequence repetitions of these 50 TSs. In Fig. 2.4A we show the repeated TSs with each of the sequence generators are only shown once. Each of the repeated TSs contained some set of sub-sequences, where each sub-sequence is repeated some number of times. Since there were 50 unique TSs, out of 697 total TSs, the 20 Repeated TSs made up the remaining 647 TSs.

In Figs. 2.4A and 2.4B we show the TS plot of One-Time and Repeated Sequences

Algorithm 1 Similarity Measure Algorithm

```
1: procedure GETSIMILARITYMEASURE(X, Y, α)
2:     for x_i ← 1, |X| do
3:         maxPercentMatch ← 0
4:         percentMatch ← 0
5:         for y_j ← 1, |Y| do
6:             percentMatch ← getTSSimilarityMeasure(x_i, y_j)
7:             if maxPercentMatch < percentMatch then
8:                 maxPercentMatch ← percentMatch
9:                 if maxPercentMatch ≥ α then
10:                    count ← count + 1
11:                    thresholdMet ← 1
12:                    break
13:                end if
14:            end if
15:        end for
16:    end for
17:    return count
18: end procedure

19: procedure GETTSSIMILARITYMEASURE(x_i, y_j)   ▷ Compares the similarity of Temporal
    Sequences x_i and y_j
20:    cardx ← |x_i|                              ▷ Store the number of elements in x_i
21:    cardy ← |y_j|                              ▷ Store the number of elements in y_j
22:    lasty ← 1
23:    count ← 0
24:    for u ← 1, cardx do      ▷ We sweep through all the elements of the TS x_i looking for
    in-order matches of elements in x_i
25:        for w ← lasty, cardy do
26:            if x_i[u] = y_j[w] then
27:                count ← count + 1
28:                lasty ← y_j[w]
29:            else
30:                continue                       ▷ Go to the next iteraction of w for-loop
31:            end if
32:        end for
33:    end for
34:    PercentMatch = count/cardx
35:    return PercentMatch
36: end procedure
```

respectively. While the entirety of the One-Time Sequences are displayed in the figure, we only show the Repeated Sequences with a minimum of their repeating sub-sequences. Together the One-Time and Repeated Sequences form the minimal description of the dynamics and can be used as for further analysis and network construction.

2.3.5 Interacting Motor Neuronal Classes Along Neuronal Signaling Pathways

C. elegans forward and backward movement is thought to result from the synchronized activations of banks of motor neurons[30] which involve the interaction between several classes of neurons[15, 71]. Signaling paths individually and in concert give rise to the neuronal firings necessary for movement. We analyzed the individual TSs to discern the associations between the various classes of mid-body motor neurons in a signaling path by expanding the set of neurons from those we previously considered. We focused on sequences starting at the ASEL neuron, traversing the mid-body motor neurons from the DA, DD, DB, VA, VD, and VB classes.

We were interested in the presence of multiple motor-neuron classes along the signaling pathways relevant to the organisms movement. As a starting point, we show (Fig. 2.5A) the number of TSs which traverse any of the neurons in the classes of interest. We observed that all of the classes implicated in movement were present in at least some of the TSs (Fig. 2.5A). Interestingly, the ventral side neuronal classes had significantly had more TSs traverse their neurons than the dorsal side neuronal classes.

Although the order of classes along the signaling pathways are relevant, they were not considered in this work. In Figs. 2.5B,C,D we show the results for 2, 3, and 4, classes of neurons present in TSs. We tabulated the number of TSs which contained 2, 3, or 4 out of the 6 mid-body motor neuronal classes (Figs. 2.5B,C,D). As the requirement for the number of classes present in each TS increased, the number of TSs which traversed the increasing number of classes decreased. Additionally we observed more TSs traversing the ventral side of neurons than the dorsal side

neurons (Fig. 2.5B).

The fact that there are 3 and 4 class interactions, provides impetus for experimentally testing these neuronal interactions, and elucidating their behavioral consequences (in the movement models of C. elegans is this something that was observed). Out of the possible 6 neuronal classes implicated in movement, TSs only contained at most 4 classes. That is to say, the dynamics on the structural connectome limited the signaling pathways to 4 classes of neurons and no more. This analysis can be expanded to consider permutations of class interactions rather than only considering combinations of interactions in TSs.

2.4 Discussion

While many experimental and analytical techniques to study the functional consequences resulting from C. elegans connectome in particular, and networks in general, focus on structural network statistics[64, 62, 63, 12], or focus on localized circuits to ascertain their function[13, 27, 14, 72], novel approaches are required for network scale analysis of dynamics[33, 11, 70, 30, 1].

We introduced the analysis of the C. elegans connectome using TSs. Our computational analysis revealed that network connectivity and geometry gave rise to orderly motor neuron activation patterns implicated in movement. Moreover, we were able to quantitatively chart the causal neuronal signaling pathways of motor neuron activations. These neuronal signaling pathways arose within the context of concurrent network activity. The analysis showed the existence of multiple neuronal signaling paths outside the Feed Network which transformed the stimulus at the ASEL neuron to activations of the VB and DB classes of motor neurons. These paths outside the Feed Network contain candidate nodes and edges which may be considered to design extended sub-networks.

By analyzing individual TSs we quantified the extent and prevalence of various classes of mid-body motor neurons in individual neuronal signaling pathways. Our results show that

there are up to four classes of neurons implicated in neuronal signaling pathways. These results underscore the need for approaches to analyze network scale concurrent dynamics within the context of a geometric network, as this view gives rise to testable hypothesis.

In order not be bogged down by biophysical complexity, we focused on the simplest neuronal model which preserved key parameters significant in network signaling[58]. Future studies can extend our analysis of C. elegans in several ways such as implementing in a geometric framework more complex neuronal models, using detailed biophysical signaling parameters, or making incremental modifications to the existing assumptions to uncover causal relationships supported by the network's structure. Given the various ways to extend this work, a delicate balance exists between the model's complexity, analytical tractability, and usability. We approached this work with eye to minimize the complexity of network components in order to start developing the framework to transform network dynamics into mathematical structures such as walks on graphs.

TSs based approaches can be implemented various scientific domains where a network abstraction is possible and causal signaling dynamics plays a role. In future work we would like to understand how the contributions of various factors affecting signaling dynamics, such as, edge/node models, network parameters and topology manifests itself in the causal relations between nodes in the networks to uncover how the network behaves at different temporal and spatial scales. Furthermore, the stability and sensitivity of TSs to small perturbations and initial conditions needs further study.

Another crucial challenge for future studies is how to extend the mathematical analysis of TSs. The Similarity Measure is but one type of mathematical analysis of TSs. To gain insights into the structure of dynamics our approach to constructing TSs can be easily transformed to more complex mathematical objects such as a graphs[66, 4] and other topological structures[18].

Although we focused on the C. elegans connectome for this work, the analysis of TSs is an approach agnostic of a specific network model. The causal interactions in a network is a facet of the dynamics, and one which has structure that can be mathematically abstracted in order to be

analyzed using a wide array of tools. A TS is one view of the structure of the dynamics.

2.5 Methods

2.5.1 Temporal Sequence Construction

A Temporal Sequence(TS) describes a walk[20] of a quanta of some signal on the graph. Let us say we have a graph $G = (V,E)$, where $v \in V$ and $E \in V \times V$. At time $t = 0$ lets say that the signal activates node v_0 and after $t = t_f$ steps node v_t is activated, then the sequence of vertices $X_{t_f} = (v_0, v_1 \cdots v_{t_f})$ is the TS starting at v_0(start node) traversing v_{t_f}(end node). Since we are interested in TSs which traverse specific nodes in the network, we specify some beginning vertex(this could be a set of vertices) for a TS and some end/traversed vertex(this could also be a set of vertices). The example we use in this paper is the set of TSs which start at ASEL and traverse any of the neurons in the VB and DB classes. As such, we are no longer interested in a TSs over time, but the TSs which traverse a specific node of interest during some time interval bounded by $t < t_k$, where t_k is the simulation time or time over which we wish to make our observations. The resulting set of TSs represent the parallel paths through the graph traversed by signals beginning at the start node and traversing the end node, over a time interval of interest.

2.5.2 Temporal Sequence Plot construction

To create a visualization the evolution of the Temporal Sequences we created a plot similar to the raster plot called the Temporal Sequence plot (TS plot). It has time steps on the x-axis, and node number on the y-axis. Each colored curve on the plot represents one TS. The height of the curve represents the node which the signal has traversed during the course of the TS, and the corresponding time of node activation is given by the value on the x-axis. Several TSs have sub-sequences which overlap. In Fig. 2.4A we observe that all the Repeated TSs go through the

same set of nodes at stimulus onset, before splitting into various Basis TSs. Note that several TSs share a partial signaling pathway

In order to better visualize the paths through the nodes of interest, we have drawn blue horizontal lines on the TS plots. Each horizontal line represents the height corresponds a neuron from to the VB and DB sets of neurons. In the Feed Network the DB set is the group of blue horizontal lines from node 17 to 23, and the VB set nodes range from 31 to 41. And in the Full Network the DB set is the group of blue horizontal lines from nodes 94 to 100, and VB set nodes range from 248 to 258.

2.5.3 Basis Sequence Construction

The Basis Temporal Sequences of the network dynamics consist of One-time and Repeated Temporal Sequences. We first discuss the definition of each of the constituent sequence type in generality, later with more specificity. Each TS is given a set of alternate descriptions which compress the repetitions within the sequence. Repetitions in the sequence represent a closed walk on the graph. The nodes and signals involved in the closed walk can give rise to repeating network activity. Repeating network activity is manifested in TSs by the repetition of a sub-sequence or a set of sub-sequences within the overall TS. We call the TSs who differ from one another only in the number of repeating sub-sequences as Repeated Temporal Sequences. It is important to node that we do not determine in this work if the Repeated TSs will indefinitely repeat. Given the definition of the Repeating Temporal Sequences, we can readily define One-Time Temporal Sequences as the remaining TSs which are not Repeating TSs.

Let X be the set of all temporal sequences. Next, let R be the set of Repeated TSs in X, where $R \subseteq X$, and finally let U be the set of One-Time Sequences, where $U \subseteq X$. We constrain the sets R and U such that $R \cap U = \emptyset$, that is, no TSs can be both a member of the Repeated TSs as well as the One-Time TSs.

For some $r_i \in R$ and $r_j \in R$, where the only difference between r_i and r_j is in the repetition

of some sub-sequences between the r_i and r_j. Next we will show how to ascertain whether some $x_i \in X$ should be a member of R. Let us assume that x_i contains sub-sequences which repeat in different parts of its sequence. Let $I = \{i_1, i_2 \cdots i_n\}$ be the set of repeating sub-sequences of x_i. Where each $i \in I$ is the sub-sequence but without repeats, a primitive sequence. To account for the repeats, let $K = \{k_1, k_2 \cdots k_n\}$ be the set containing the number of repetitions of each of the primitive sequences in I. As such any $x \in X$ can be re-written as combination of primitive sequences together with its non-repeating sub-sequences. Concretely, given some TS x, we compress all its repeating sub-sequences in the form of $i_j^{k_j}$, where j is the j^{th} sub-sequence in I, which takes on k_j repetitions.

Every TS, $x_i \in X$, which can be derived from changing the number of repetitions of a primitive sequence, to match another sequence, $x_j \in X$ is considered a Repeated TS. Both x_i and x_j are considered elements of R. Contained in the set I are the candidate pattern generators[28, 42] of the network, that is, the sequence of node activations which sustain ongoing network activity, and the realized cycles of the graph. It is not necessary that the dynamics realizes all the possible cycles of the graph. Because the elements of R are attained from observation, that Repeated TSs are not necessarily periodic, in the sense that they are guaranteed to repeat indefinitely.

Given the definition of Repeated Sequences, we can define the set U of One-Time Sequences as $U = \{u | u \in X, u \notin R\}$. The difference between R and U is that for all $x_i \in U$ there is no other sequence $x_j \in X$ such that the difference between x_i and x_j is only in some repetition of a sub-sequence in x_i. Note $u_i \in U$ may contain repeated sub-sequences. Since there are many approaches to ascertain Basis Sequences, we do not provide an explicit algorithm to extract Basis Sequences. We can reconstruct all TSs in X form the the Basis Sequences and knowledge of the number of repetitions of each of the sub-sequences.

2.5.4 Network Construction

Full Network

We represent the structural connectome using a directed geometric graph. A graph G is a pair $G = (V, E)$. Where V is the set of vertices/nodes/neurons, and E is the set of edges/link/axons such that $E \subseteq V \times V$. A geometric graph is a graph whose vertices are embedded in euclidean space. We consider neuronal signaling to be unidirectional, that is signals enter the neuron through the dendritic arbor and exit the neuron through the axonal arbor[32], as such, the resulting network is directed. We used a simplified version of the Geometric Dynamic Perceptron model to describe the dynamics of each of the nodes[58]. In what follows we will discuss relevant network and node parameters.

We constructed the directed adjacency matrix of the Full Network using connectivity information from[67, 31, 3]. Let A be the network adjacency matrix, if there is a connection between neuron i and neuron j, then $A_{ij} = 1$, otherwise $A_{ij} = 0$. Our model only considered chemical connections between neurons, that is connections along axons. We did not consider electrical connections through gap junctions. Next we constructed the distance adjacency matrix using experimentally available data. Out of the 302 neurons which make up the hermaphrodite C. elegans connectome, the 2D location of only 277 neurons are known[17]. Therefore, we calculated the straight line edge lengths between those neurons.

Our interest here is in analyzing the network dynamics supported by the structural connectome, specifically from the perspective of causal neuronal signaling pathways, rather than studying mechanisms underlying the molecular basis for neuronal signaling. Therefore, we assumed that some quanta of charge is transmitted between nodes through all-or-none nodal events[44, 40, 39, 45]. Once a node initiated a signal, we assume it traverses all outgoing edges at a constant conduction velocity. We used a conduction velocity of $80\frac{mm}{s}$[49]. We calculated the signaling time delays along edges through the geometrically derived edge lengths and the

conduction velocity.

All nodes in the network were either inhibitory or excitatory. Only the GABA expressing neurons were considered inhibitory[24]. All other neurons types were considered excitatory[50], including the unknown neuron types. An excitatory neuron sends an excitatory signal through its outgoing edges. An excitatory signal elicits a response from its down stream neuron. That response is two fold, first the downstream neuron becomes refractory, and second the downstream neuron send outgoing signals itself. In contrast, an inhibitory neuron when activated sends an inhibitory signal. An inhibitory signal also elicits a response from its down stream neuron, but that response is only one fold. That is, the downstream neuron becomes refractory, and does not send any outgoing signals. While a node is refractory none of the incoming signals which reach the axon hillock contribute to that nodes firing. The refractory period of all neurons are set to $4ms$[8, 41].

The simplified Geometric Dynamic Perceptron node model takes on two states, it is either active/firing or inactive/waiting. Let the state of node i at time t be given by $y_i(t) = \{0,1\}$. When the node is in the active state $y(t) = 1$, when it is inactive $y(t) = 0$. While node i is initially in an inactive state it can instantly transition to the active state; but when the node i is in the active state it will become refractory for the time interval of the refractory period. The threshold for activation is one incoming signal, that is, when a signal arrives to the axon hillock of node i and it is in the inactive state, instantaneously, an outgoing signal will be generated and transmitted through all its outgoing edges.

To run our simulations, the network activity is initialized to quiescence. We initiate network activity by stimulating the ASEL neuron with a single pulse stimulus, eliciting outgoing signals based on the network connectivity. The simulation is run for approximately $1.5s$ of network activity, broken up into 6000 discrete time steps.

Feed Network

We used the same basic assumptions of the Full Network to construct the Feed Network. The difference between the two networks is that the Feed Network is a subgraph[69] of the Full Network. To generate the connections for the Feed Network we created a new distance adjacency matrix by collecting the rows and columns corresponding to the nodes in the Feed Network.

Lattice Network

We used the adjacency matrix of the Full Network for the Lattice Network. The only difference between the two networks are in the signaling parameters, namely, the refractory periods of each node and the edge lengths. The objective for constructing the Lattice Network was to diminish the influence of variations in edge lengths. To do so the edge signaling delays of the lattice network were uniformly set to $1tu$ (arbitrary time unit), and the refractory periods were set to $0.9tu$. This ratio of refractory period to signaling delay is considered to be near optimal from a local signaling perspective[58].

Edge-Swap Network

The Edge-Swap Network was constructed using the edge-swapping algorithm[54]. The edge-swapping algorithm was used to randomized the edges in the network to construct null networks for comparison. The randomization procedure is as follows: Two edges are randomly selected. If the two edges have two distinct source nodes and two distinct destination nodes, then the the two destination nodes are swapped as long as the new edge does not already exist. If any of the nodes associated with the randomly selected edges are the same, then two other edges are picked randomly, and the procedure is repeated. A network connectedness check is performed to make sure a separate network component is not created during the edge-swapping process. The code for the algorithm can be found at[54].

The algorithm is run for 10 iterations on both the Full Network and the Feed Network,

to generate their randomized versions. This randomization scheme preserves the node degree distribution, by extension the number of edges in the network, and network connectedness. A new distance adjacency matrix is computed given the new connection set.

Gilbert Randomized Network

Here we describe the Gilbert randomization procedure. The difference between between the Full Network and their Gilbert randomized[26, 52, 22] counterparts are in the edge connectivity pattern and new edge lengths. The Gilbert random network is constructed based on the total number of edges in the Full Network. We calculated the probability of edge connection as follows: Let N be the number of nodes in the Full Network, and let e be the number of edges in the Full Network.

$$p = \frac{e}{N(N-1)} \tag{2.2}$$

To create the random adjacency matrix, we create a matrix of random number between 0 and 1, then using p as the threshold value, we set the values of the matrix to 0 if the previous value was greater than p and 1 if the previous value was less than p. This results in an adjacency matrix of 0s and 1s with approximately the same number of edges in the Gilbert Randomized Network as in the Full Network. Using the positional data for the neurons, we calculated a distance adjacency matrix based on the network connectivity.

Embedded Random Feed Network

We embed either (Gilbert or Edge-Swap) edge randomized Feed Network into the Full Network to generate a, so called, Embedded Random Feed Network. To create this network, we replace the rows and columns of Full Network's distance adjacency matrix with the edge randomized Feed Network's distance adjacency matrix. We use the same distance adjacency matrix as in the standalone Edge Randomized Feed Network to make network comparisons of dynamics more congruent.

Table 2.1: Table of Similarity Measures with respect to the different networks and with respect to different randomization schemes. The rows in the leftmost column of the table represent the percent of a target pattern (refer to the matching criteria α in the body) which matched any of the patterns in the source network. The heading for each column are organized as reference network vs target network. The cells in the table represent the number of TSs of the reference network which found a match TS from the target network meeting the corresponding row's matching criteria.

Gilbert Rand Based Network					
Percent match	Feed(124) vs Feed Rand(598)	Feed(124) vs Full(697)	Feed(124) vs Embedded Rand Feed(649)	Full(697) vs Embedded Rand Feed(649)	Full(697) vs Full Rand(665)
0.00%	124	124	124	697	697
25.00%	9	78	9	576	10
50.00%	0	33	0	36	1
75.00%	0	13	0	3	0
100.00%	0	6	0	1	0
Edge Swapped Network					
Percent match	Feed(124) vs Feed Rand(130)	Feed(124) vs Full(697)	Feed(124) vs Embedded Rand Feed(674)	Full(697) vs Embedded Rand Feed(674)	Full(697) vs Full Rand(643)
0.00%	124	124	124	697	697
25.00%	50	78	9	688	16
50.00%	6	33	0	614	2
75.00%	3	13	0	422	0
100.00%	1	6	0	0	0

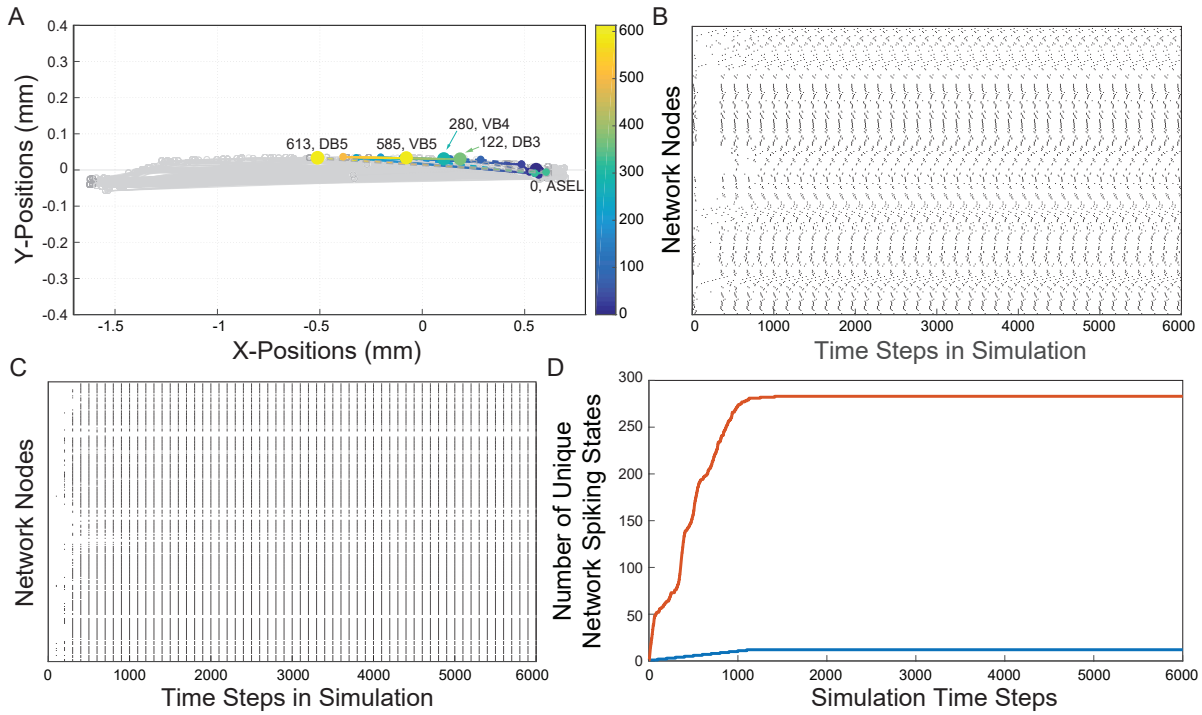

Figure 2.1: Panel A, The reconstruction of the C. elegans connectome. Each gray line is an edge. Each edge represents chemical connections between nodes. The colored dashed lines represents a sample of the traversal of a signal from the sensory neuron ASEL through the interneuron network to the VB and DB classes of motor-neurons. The color gradient indicates the time course as the signal traverses the network. Cold colors represent early in the neuronal signaling pathway and warm colors represent late in the neuronal signaling pathway. The node labels and numbers next to them represent the names of the motor neurons at the terminus of the sequence of activations and the time step when that terminal node was activated. Panel B, The Full Network's spike raster caused by stimulating ASEL. The activity is a direct result of the signaling parameters(signal conduction velocity, refractory period), network connectivity, and the spatial locations of each of the nodes. Each dot represents the activation of a node in the network, the white space represents an inactive state of the node. Each discrete height on y-axis represents a specific node in the network. Panel C, The Lattice Network's spike raster caused by stimulating ASEL. The Lattice Network preserves the signaling parameters and network connectivity, but in effect modifies the spatial locations relative to the Full Network. As a result all edges in the Lattice Network are have uniform delays. The resulting spike raster is significantly less varied than that of the Full Network. We will quantify this previous statement with the next panel. Panel D, Each curve is a cumulative count of the unique number of network states of presented by the Full(red) and Lattice(blue) Networks as they evolve in time. We observe an approximately a 20-fold increase in the number of network states resulting from the dynamics of the Full Network than the states resulting from the Lattice Network.

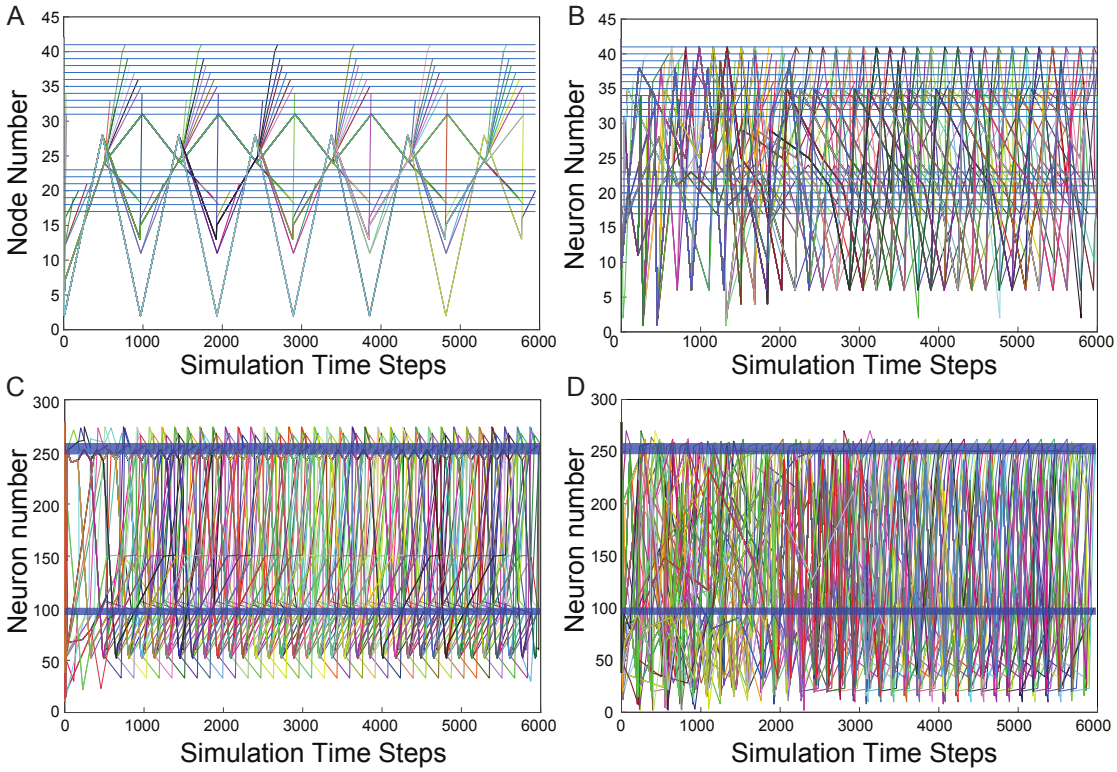

Figure 2.2: TS Plots. All TSs from ASEL to the VB and DB classes. Panel A, TS plot of Feed Network. The activation patterns of neurons in the VB and DB classes are synchronized and staggered eventually giving rise to periodic sequences of node activity. Panel B, TSs of the Gilbert Randomized Feed Network. Synchronized and staggered activations from the VB and DB class are absent. Transient period of network activity is longer than in Panel A. The TSs eventually enter a periodic sequences of node activity. There were significantly more TSs in the randomized network than in Panel A. Panel C, The number of TSs of the Full Network were significantly more than in the Feed Network (Panel A). Furthermore, the number of motor neurons were activated at a faster rate than they were in the Feed Network (Panel A). This indicates that there are many other signaling pathways or more active signaling pathways from ASEL to the VB and DB classes of motor neurons which are either not captured or not as quickly activated by the Feed Network. The TSs of the Full Network seems to enter periodic activity within the first 1000 simulation time steps. Panel D, TSs from Gilbert Randomized Full Network. The TS displayed a significantly longer transient period before the network activity entered a periodic regime. The transient period lasted approximately for the first 2000 steps of network activity. The time to periodic network activity relative to the original networks seems to be an indicator of the extent of random deviation of network connectivity.

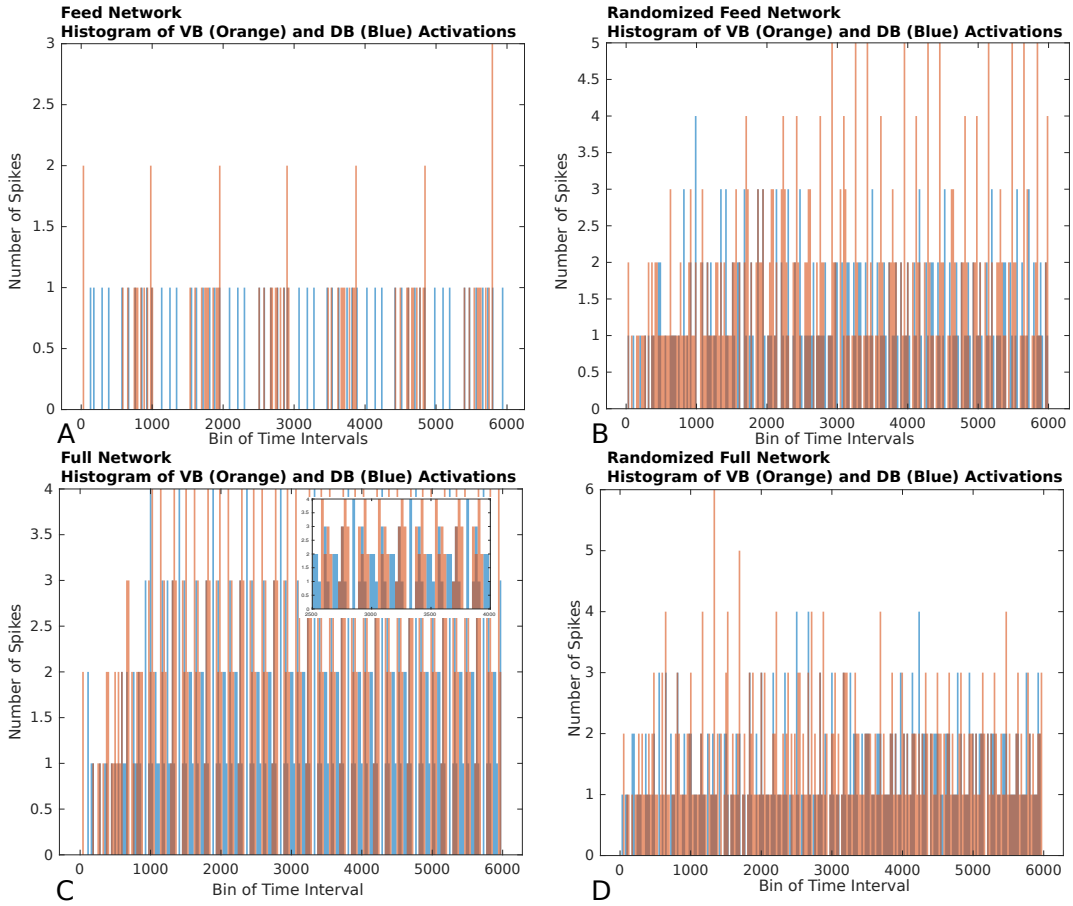

Figure 2.3: Here we show histograms of the number of firings of nodes from the VB and DB classes over discrete time intervals of length 250 steps. Panel A. Histogram of the Feed Network's VB and DB class neurons activations. The network structure supports dynamics in the form of grouped actvivations of neurons from the VB classes, then a group of activations of neurons from the DB classes, and the switch of activity from the VB to DB classes continues. Panel B. Histogram of the Gilbert Randomized Feed Network's VB and DB class neurons activations. The randomization procedure significantly modified the class based neuron activation patterns and decimates the grouped class based activations we observed in Panel A. A semblance of periodic node activations from the VB and DB classes is observed after approximately 3000 time steps. Panel C. Histogram of the Full Network's VB and DB class neurons activations. The C. elegans connectome gives rise to significantly more activity in the VB and DB classes of motor neurons in comparison to the activity in the Feed Network, Panel A. The dynamics of the Full Network also supported alternating VB, DB class activity. Because of the density of node activation we show in the inset a zoomed in view of a sub-interval of activity to better view the VB-DB class activation pattern. Panel D. Histogram of the Gilbert Randomized Full Network's VB and DB class neurons activations. The randomization procedure resulted in the destruction of the class-wise firing patterns.

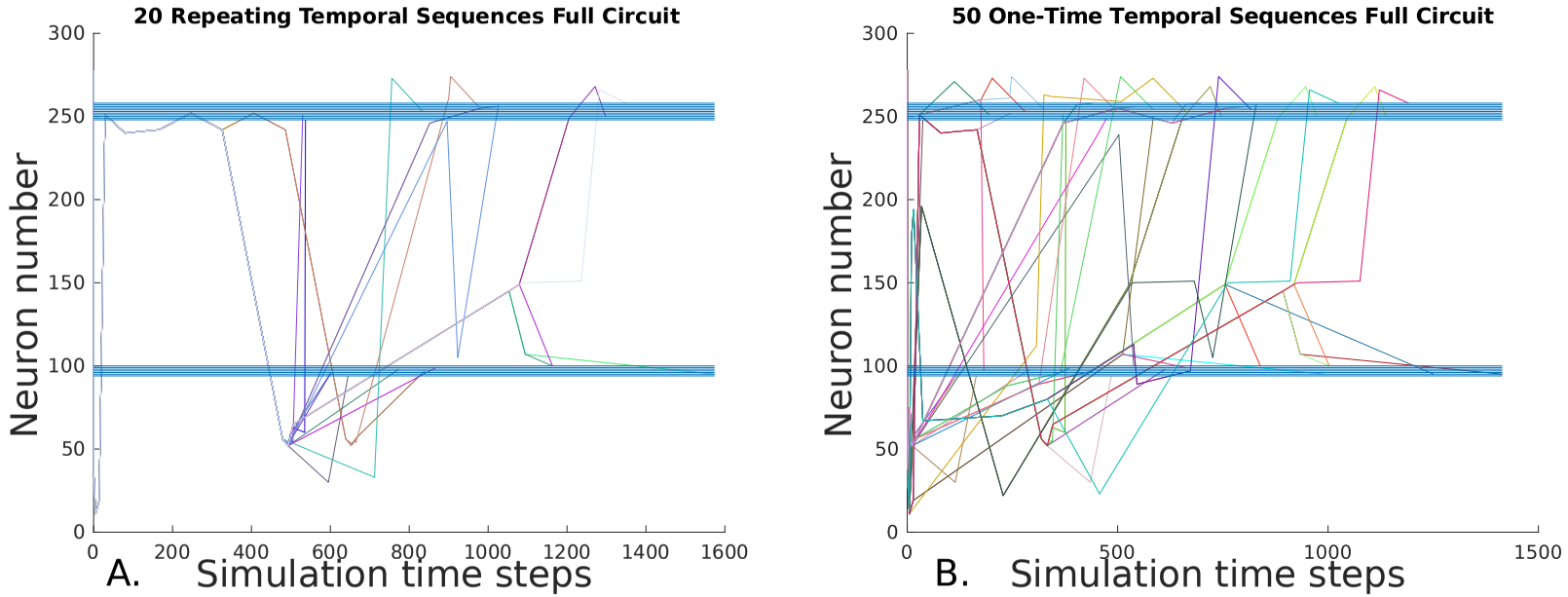

Figure 2.4: We decomposed all the TSs from the Full Network into its basis set. The basis set contains One-Time TSs and Repeated TSs. One time TSs can be construed as the transient activity of the network, and the Repeated TSs can be construed as the steady-state or repeating activity of the network. The horizontal axis represents time, the vertical axis represents the nodes of the network. Each line in the plot shows the traversal of the signal through the network over time. The blue horizontal lines represent the heights of neurons from the VB and DB classes. The DB class neurons are located at approximately y-values of 100. The VB class neurons are located at approximately y-values of 250. Panel A, TS plot of Repeated TSs. These TSs contain sub-sequences, that when repeated, is an exact match for another TS. Panel B, TS plot of One-Time TSs. These TSs are unique neuronal signaling paths present in the dynamics.

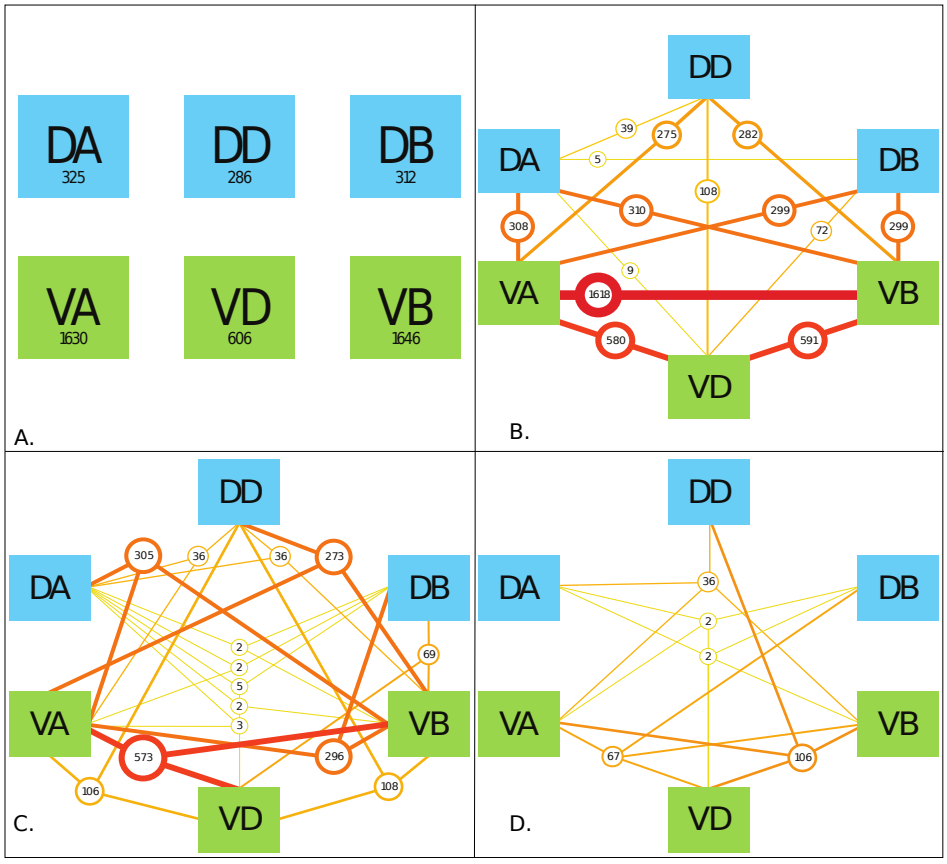

Figure 2.5: Here we expand the motor neuron classes under consideration to include DA, DD, DdB, VA, VD, and VB classes. We color coded the dorsal side classes in blue and the ventral size classes in green. To elucidate the classes implicated in individual TSs, as a result of ASEL stimulation, we counted the number of TSs which contained at least one, two, three and four classes out of the six total classes under consideration. We only show 4 panels because there were no TSs which contained five or more classes in them. Panel A, The values in each of boxes indicate the number of TSs with the specific neuronal class present within the TS. For example, looking at the box labeled DA, at least one of the neurons in the DA class was present in 325 TSs. The ventral classes were over-represented in TSs with single classes present. Panel B, Here we considered the presence of any two classes of motor neurons in individual TSs. The most prevalent two classes present were neurons from the VA-VB classes. Only the DB-DD class interaction was not observed in any of the TSs captured. Panel C, Here we considered three class interactions. The classes present in the TSs are given by the lines connecting the boxes. The number of sequences are given by the circles. Here we observe significantly fewer TSs than in Panel B. Panel D, There are significantly fewer TSs meeting the four present classes criteria. We do not consider the order of present classes because of the much bigger combinatorial space to visualize.

References

[1] Misha B. Ahrens, Michael B. Orger, Drew N. Robson, Jennifer M. Li, and Philipp J. Keller. Whole-brain functional imaging at cellular resolution using light-sheet microscopy. *Nature Methods*, 10(5):413–420, May 2013.

[2] Uri Alon. Network motifs: theory and experimental approaches. *Nature Reviews Genetics*, 8(6):450–461, June 2007.

[3] Z. F. Altun, L.A. Herndon, C.A. Wolkow, C. Crocker, R. Lints, and D. H. Hall. WormAtlas, 2002.

[4] José Ayala and Wolfgang Kliemann. Topological dynamics on finite directed graphs. *arXiv:1501.07509 [math]*, March 2016. arXiv: 1501.07509.

[5] Cornelia I. Bargmann. Beyond the connectome: How neuromodulators shape neural circuits. *BioEssays*, 34(6):458–465, June 2012.

[6] Cornelia I. Bargmann and Eve Marder. From the connectome to brain function. *Nature Methods*, 10(6):483–490, June 2013.

[7] Danielle S. Bassett, Ankit N. Khambhati, and Scott T. Grafton. Emerging Frontiers of Neuroengineering: A Network Science of Brain Connectivity. *Annual Review of Biomedical Engineering*, 19(1):327–352, 2017.

[8] Michael J. Berry and Markus Meister. Refractoriness and Neural Precision. *Journal of Neuroscience*, 18(6):2200–2211, March 1998.

[9] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D. U. Hwang. Complex networks: Structure and dynamics. *Physics Reports*, 424(4):175–308, February 2006.

[10] Stefano Boccaletti, Ginestra Bianconi, Regino Herrero, C.I.del Genio, Jesus Gómez-Gardeñes, Miguel Romance, Irene Sendiña-Nadal, Zhen Wang, and Massimiliano Zanin. The Structure and Dynamics of Multilayer Networks. *Physics Reports*, 544:1–122, November 2014.

[11] Connor Brennan and Alexander Proekt. A quantitative model of conserved macroscopic dynamics predicts future motor commands. *eLife*, 8:e46814, July 2019.

[12] Ed Bullmore and Olaf Sporns. The economy of brain network organization. *Nature Reviews Neuroscience*, 13(5):336–349, May 2012.

[13] Sreekanth H. Chalasani, Nikos Chronis, Makoto Tsunozaki, Jesse M. Gray, Daniel Ramot, Miriam B. Goodman, and Cornelia I. Bargmann. Dissecting a circuit for olfactory behaviour in Caenorhabditis elegans. *Nature*, 450(7166):63–70, November 2007.

[14] M Chalfie, Je Sulston, Jg White, E Southgate, Jn Thomson, and S Brenner. The neural circuit for touch sensitivity in Caenorhabditis elegans. *The Journal of Neuroscience*, 5(4):956–964, April 1985.

[15] Martin Chalfie, John E. Sulston, JOHN G. White, Eileen Southgate, J. Nicol Thomson, and Sydney Brenner. The neural circuit for touch sensitivity in Caenorhabditis elegans. *Journal of Neuroscience*, 5(4):956–964, 1985.

[16] Yongmin Cho, Charles L. Zhao, and Hang Lu. Trends in high-throughput and functional neuroimaging in *Caenorhabditis elegans*: High-throughput and functional neuroimaging in *C . elegans*. *Wiley Interdisciplinary Reviews: Systems Biology and Medicine*, 9(3):e1376, May 2017.

[17] Y. Choe, B. H. McCormick, and W. Koh. Network connectivity analysis on the temporally augmented C. elegans web: A pilot study. In *Soc Neurosci Abstr*, volume 30, 2004.

[18] Carina Curto. What can topology tell us about the neural code? *Bulletin of the American Mathematical Society*, 54(1):63–78, September 2016.

[19] N. Deo. Graph Theory with Applications to Engineering and Computer Science. *Networks*, 5(3):299–300, 1975.

[20] R. Diestel. *Graph Theory*. Electronic library of mathematics. Springer, 2006.

[21] Elizabeth M. DiLoreto, Christopher D. Chute, Samantha Bryce, and Jagan Srinivasan. Novel Technological Advances in Functional Connectomics in C. elegans. *Journal of Developmental Biology*, 7(2), April 2019.

[22] P. Erdős and A. Rényi. On the Evolution of Random Graphs. In *Publication of the Mathematical Institute of the Hungarian Academy of Sciences*, pages 17–61, 1960.

[23] Lise Frézal and Marie-Anne Félix. C. elegans outside the Petri dish. *eLife*, 4, March 2015.

[24] Marie Gendrel, Emily G Atlas, and Oliver Hobert. A cellular and regulatory map of the GABAergic nervous system of \textlessi\textgreaterC. elegans\textless/i\textgreater. *eLife*, 5:1–38, 2016.

[25] Wulfram Gerstner and Richard Naud. How Good Are Neuron Models? *Science*, 326(5951):379–380, October 2009.

[26] E. N. Gilbert. Random Graphs. *Annals of Mathematical Statistics*, 30(4):1141–1144, 1959.

[27] Jesse M. Gray, Joseph J. Hill, and Cornelia I. Bargmann. A circuit for navigation in Caenorhabditis elegans. *Proceedings of the National Academy of Sciences of the United States of America*, 102(9):3184–3191, March 2005.

[28] Pierre A. Guertin. Central Pattern Generator for Locomotion: Anatomical, Physiological, and Pathophysiological Considerations. *Frontiers in Neurology*, 3, February 2013.

[29] D. H. Hall and Z. F. Altun. C. elegans atlas. *C. elegans atlas.*, 2007.

[30] Gal Haspel, Michael J O'Donovan, and Anne C Hart. Motoneurons dedicated to either forward or backward locomotion in the nematode Caenorhabditis elegans. *The Journal of neuroscience : the official journal of the Society for Neuroscience*, 30(33):11151–11156, 2010.

[31] Marcus Kaiser and Claus C. Hilgetag. Nonoptimal Component Placement, but Short Processing Paths, due to Long-Distance Projections in Neural Systems. *PLOS Computational Biology*, 2(7):e95, July 2006.

[32] E.R. Kandel, T.M. Jessell, J.H. Schwartz, S.A. Siegelbaum, and A.J. Hudspeth. *Principles of Neural Science, Fifth Edition*. Principles of Neural Science. McGraw-Hill Education, 2013.

[33] Saul Kato, Harris S. Kaplan, Tina Schrödel, Susanne Skora, Theodore H. Lindsay, Eviatar Yemini, Shawn Lockery, and Manuel Zimmer. Global Brain Dynamics Embed the Motor Command Sequence of Caenorhabditis elegans. *Cell*, 163(3):656–669, October 2015.

[34] Jimin Kim, William Leahy, and Eli Shlizerman. Neural Interactome: Interactive Simulation of a Neuronal System. *Frontiers in Computational Neuroscience*, 13, 2019.

[35] Hiroaki Kitano. Systems Biology: A Brief Overview. *Science*, 295(5560):1662–1664, March 2002.

[36] Christof Koch. *Biophysics of Computation: Information Processing in Single Neurons*. Oxford University Press, October 2004. Google-Books-ID: aeAJCAAAQBAJ.

[37] Nancy J. Kopell, Howard J. Gritton, Miles A. Whittington, and Mark A. Kramer. Beyond the Connectome: The Dynome. *Neuron*, 83(6):1319–1328, September 2014.

[38] Stephen D. Larson, Padraig Gleeson, and André E. X. Brown. Connectome to behaviour: modelling Caenorhabditis elegans at cellular resolution. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 373(1758):20170366, October 2018.

[39] Shawn R. Lockery and Miriam B. Goodman. The quest for action potentials in C. elegans neurons hits a plateau. *Nature neuroscience*, 12(4):377–378, 2009.

[40] Shawn R Lockery, Miriam B Goodman, and Serge Faumont. First report of action potentials in a C. elegans neuron is premature. *Nature neuroscience*, 12(4):365–366; author reply 366, 2009.

[41] S. Mack, E.R. Kandel, T.M. Jessell, J.H. Schwartz, S.A. Siegelbaum, and A.J. Hudspeth. *Principles of Neural Science, Fifth Edition*. Principles of Neural Science. McGraw-Hill Education, 2013.

[42] Eve Marder and Dirk Bucher. Central pattern generators and the control of rhythmic movements. *Current Biology*, 11(23):R986–R996, November 2001.

[43] Stefano Masoli, Martina F. Rizza, Martina Sgritta, Werner Van Geit, Felix Schürmann, and Egidio D'Angelo. Single Neuron Optimization as a Basis for Accurate Biophysical Modeling: The Case of Cerebellar Granule Cells. *Frontiers in Cellular Neuroscience*, 11, March 2017.

[44] Jerry E Mellem, Penelope J Brockie, David M Madsen, and Andres V Maricq. Action potentials contribute to neuronal signaling in C. elegans. *Nature Neuroscience*, 11(8):865–867, August 2008.

[45] Jerry E. Mellem, Penelope J. Brockie, David M. Madsen, and Andres V. Maricq. Reply to "First report of action potentials in a C. elegans neuron is premature". *Nature Neuroscience*, 12(4):366–366, April 2009.

[46] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon. Network Motifs: Simple Building Blocks of Complex Networks. *Science*, 298(5594):824–827, October 2002.

[47] Carolina A. Moreira and Marcus A. M. de Aguiar. Modular structure in C. elegans neural network and its response to external localized stimuli. *Physica A: Statistical Mechanics and its Applications*, 533:122051, November 2019.

[48] Mark Newman. *Networks: An Introduction*. Oxford University Press, March 2010.

[49] Ernst Niebur and Paul Erdo. Theory of the locomotion of nematodes: Control of the somatic motor neurons by interneurons. *Mathematical Biosciences*, 118(1):51–82, 1993.

[50] Laura Pereira, Paschalis Kratsios, Esther Serrano-Saiz, Hila Sheftel, Avi E. Mayo, David H. Hall, John G. White, Brigitte LeBoeuf, L. Rene Garcia, Uri Alon, and Oliver Hobert. A cellular and regulatory map of the cholinergic nervous system of C. Elegans. *eLife*, 4(DECEMBER2015):1–46, 2015.

[51] Beverly J. Piggott, Jie Liu, Zhaoyang Feng, Seth A. Wescott, and X.Z. Shawn Xu. The Neural Circuits and Synaptic Mechanisms Underlying Motor Initiation in C. elegans. *Cell*, 147(4):922–933, November 2011.

[52] Brenton J. Prettejohn, Matthew J. Berryman, and Mark D. McDonnell. Methods for Generating Complex Networks with Selected Structural Properties for Simulations: A Review and Tutorial for Neuroscientists. *Frontiers in Computational Neuroscience*, 5, 2011.

[53] Donald L. Riddle, Thomas Blumenthal, Barbara J. Meyer, and James R. Priess. *Introduction: the Neural Circuit For Locomotion*. Cold Spring Harbor Laboratory Press, 1997.

[54] Mikail Rubinov and Olaf Sporns. Complex network measures of brain connectivity: Uses and interpretations. *NeuroImage*, 52(3):1059–1069, September 2010.

[55] Kaeser M. Sabrin, Yongbin Wei, Martijn van den Heuvel, and Constantine Dovrolis. The hourglass organization of the C. elegans connectome. *bioRxiv*, page 600999, April 2019.

[56] David Sankoff and Joseph B. Kruskal. *Time warps, string edits, and macromolecules: the theory and practice of sequence comparison*. Addison-Wesley Pub. Co., Advanced Book Program, 1983. Google-Books-ID: _e5QAAAAMAAJ.

[57] Gopal P. Sarma, Chee Wai Lee, Tom Portegys, Vahid Ghayoomie, Travis Jacobs, Bradly Alicea, Matteo Cantarelli, Michael Currie, Richard C. Gerkin, Shane Gingell, Padraig Gleeson, Richard Gordon, Ramin M. Hasani, Giovanni Idili, Sergey Khayrulin, David Lung, Andrey Palyanov, Mark Watts, and Stephen D. Larson. OpenWorm: overview and recent advances in integrative biological simulation of Caenorhabditis elegans. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 373(1758):20170382, October 2018.

[58] Gabriel A. Silva. The Effect of Signaling Latencies and Node Refractory States on the Dynamics of Networks. *arXiv:1804.07609 [q-bio]*, August 2019. arXiv: 1804.07609.

[59] Olaf Sporns, Giulio Tononi, and Rolf Kötter. The Human Connectome: A Structural Description of the Human Brain. *PLOS Computational Biology*, 1(4):e42, September 2005.

[60] Hiroshi Suzuki, Tod R. Thiele, Serge Faumont, Marina Ezcurra, Shawn R. Lockery, and William R. Schafer. Functional asymmetry in C. elegans taste neurons and its computational role in chemotaxis. *Nature*, 454(7200):114–117, July 2008.

[61] Balázs Szigeti, Padraig Gleeson, Michael Vella, Sergey Khayrulin, Andrey Palyanov, Jim Hokanson, Michael Currie, Matteo Cantarelli, Giovanni Idili, and Stephen Larson. OpenWorm: an open-science approach to modeling Caenorhabditis elegans. *Frontiers in Computational Neuroscience*, 8, November 2014.

[62] Emma K. Towlson, Petra E. Vértes, Sebastian E. Ahnert, William R. Schafer, and Edward T. Bullmore. The Rich Club of the C. elegans Neuronal Connectome. *Journal of Neuroscience*, 33(15):6380–6387, April 2013.

[63] Martijn P. van den Heuvel and Olaf Sporns. Network hubs in the human brain. *Trends in Cognitive Sciences*, 17(12):683–696, December 2013.

[64] Lav R. Varshney, Beth L. Chen, Eric Paniagua, David H. Hall, and Dmitri B. Chklovskii. Structural properties of the Caenorhabditis elegans neuronal network. *PLoS Computational Biology*, 7(2), 2011.

[65] Lifang Wang, Hirofumi Sato, Yohsuke Satoh, Masahiro Tomioka, Hirofumi Kunitomo, and Yuichi Iino. A Gustatory Neural Circuit of Caenorhabditis elegans Generates Memory-Dependent Behaviors in Na+ Chemotaxis. *Journal of Neuroscience*, 37(8):2097–2111, February 2017.

[66] B Yu Weisfeiler and A A Leman. A Reduction of a Graph to Canonical Form and the Algebra Which Appears Therein. *Nauchno-Technicheskaya Informatsiya*, page 11, 1968.

[67] J. G. White, E. Southgate, J. N. Thomson, and S. Brenner. The structure of the nervous system of the nematode Caenorhabditis elegans. *Phil. Trans. R. Soc. Lond. B*, 314(1165):1–340, November 1986.

[68] Elisabeth Wong, Brittany Baur, Saad Quader, and Chun-Hsi Huang. Biological network motif detection: principles and practice. *Briefings in Bioinformatics*, 13(2):202–215, March 2012.

[69] Jian Xin Xu and Xin Deng. Biological modeling of complex chemotaxis behaviors for C. elegans under speed regulation - A dynamic neural networks approach. *Journal of Computational Neuroscience*, 35(1):19–37, 2013.

[70] Gang Yan, Petra E. Vértes, Emma K. Towlson, Yee Lian Chew, Denise S. Walker, William R. Schafer, and Albert-László Barabási. Network control principles predict neuron function in the Caenorhabditis elegans connectome. *Nature*, 550(7677):519–523, October 2017.

[71] Mei Zhen and Aravinthan D T Samuel. C. elegans locomotion: Small circuits, complex functions. *Current Opinion in Neurobiology*, 33:117–126, 2015.

[72] Mei Zhen and Aravinthan D T Samuel. C. elegans locomotion: Small circuits, complex functions. *Current Opinion in Neurobiology*, 33:117–126, 2015.

Chapter 2, in part, is currently being prepared for submission for publication of the material. Vivek K. George, Francesca Puppo and Gabriel A. Silva. The dissertation author was the primary investigator and author of this paper.

Chapter 3

Embeddings Of Network Dynamics Using Steady State Statistics

3.1 Abstract

Analysis of the dynamics of spatial-temporal neural networks[49, 39] can provide vital information for characterizing and differentiating between different inputs. We distinguish two network's in this work, there is a spatial network which is the connectivity of nodes along edges, and there is the spatial-temporal network which is created by embedding the neuronal dynamics into a graph whose nodes are neuron activation's and edges , acting as axons, connect the neuron's causal activators. A question which arises in the analysis of network dynamics is how much of the dynamics should be captured, especially, when the patterns of activity start to repeat. We propose a summation based method and a string repetition finding based method to ascertain the steady-state statistics of the dynamics, that is the starting point and period of steady-state oscillations. Those statistics provides bounds over or a stopping criteria for data acquisition. The string method is implemented in a linear time algorithm to identify steady state dynamics based on node firing patterns. This is achieved by converting node firings into a string, then using an

efficient string repetition finding algorithm to identify repeats in the string's corresponding to the steady state firing patterns. After ascertaining the steady-state statistics, we map the network dynamics into a graph, then create a vector embedding of the graph based on the interacting sub-graphs. This work acts as a scaffold upon which various aspects discussed can be further developed and tested. For example, the design and testing of networks, node models, turning of parameters, and the creation of more informative embedding techniques.

3.2 Introduction

Modeling physical and information systems as networks is a powerful approach when multiple elements that make up the system interact in non-trivial and complex ways[44, 23]. Network abstractions are used pervasively in science and engineering[8, 3, 52, 27, 28, 2, 22, 15].

Both the structure and dynamics of complex networks are well studied[44, 3, 31]. Work so far has focused on statistics of network structure[24, 48, 14], and signal spreading[46, 43, 21, 20], for example oscillator synchronization, diffusion processes, rumour spreading, and epidemics spreading[32, 4]. The problem of mathematically embedding network dynamics for quantitative comparison is only starting to receive attention[5, 41]. A complex dynamic network consists of at least two topologies, the topology of the network's structure and the topology of the network's dynamics. Structural topology consists of the physical connectivity of the vertices in a network, for example, vertex count, edge count, and degree distribution. In the context of neuroscience[7] neurons which are physically connected are not necessarily causally dependent. The topology of network dynamics is a subset of the structural topology and consists of the nodes and edges resulting from observed causal signaling paths. Analysis of the topology of network dynamics provides a new approach to develop artificial neural networks for machine learning.

Classically, an artificial neural network is represented, or modeled, as a directed graph[34, 50]. The vertices (nodes) are where computations occur, while the edges (links) of the graph have

weights on them that affect the computations. Some vertices receive inputs or signals from the outside world. These are the input nodes. While some vertices are considered output nodes, the subset of nodes from which one 'reads out' the result of the computation by the network. The rest of the vertices reflect hidden nodes within the network. These are the nodes that are actually carrying out the computations. Each vertex has associated with it an activation function that determines if and when the resulting of summating inputs into the vertex produces an activation of that vertex, which in turn results in the vertex signaling and affecting the vertices it connects into. Activation functions can take on many forms. For example, it can be stochastic, such as in Boltzman machines[42, 51], or deterministic, such as various kinds of feedforward networks[34]. The degree or amount that the activation of a vertex influences the vertices it connects into is determined by the value of the weight on the edge between the two. A vertex sums all the weights from upstream vertices and it is this value that becomes the input into the activation function. The greater the weight between two connected nodes, the greater the effect of the upstream node on the activation of the downstream node. The process of learning in the context of artificial neural networks is a progressive adjustment of these weights. The networks are 'trained' through a random search algorithm in order to find a combination of weights such that the output of the network matches as closely as possible a set of inputs. That way, after its trained, when the network encounters an unknown input it is asked to classify, the resultant output will match one of the known outputs the network has been trained on.

In contrast, we are developing an alternative model of an artificial neural network that does not depend on the training and global adjustment of weights[49, 7]. Weights can be randomly initialized and can either be locally updated[10] or remain fixed (as in this work). Inputs (i.e. signals or information) are still encoded in our networks by what subset of vertices are activated and in what order, and there are still summations at individual vertices that trigger an activation function. But the mapping from input to output is a result of the timing of how signals are flowing or spreading through the network, which in our case are geometric. They have an equivalent

physical geometric representation in space, and therefore, the timing of when signals propagating on the edges of the network arrive at their downstream nodes to influence the summation matter. The resultant patterns in the dynamics of the network that emerge through this process are a natural consequence of input signals themselves. There is no training of the weights. And the timing of the signals in how they interact with each other and affect what vertices are activated when is a direct result of how the network is set up in the first place: its physical connectivity, the signaling speed or conduction velocity of signals on the edges, and, importantly, a period of refractoriness for each vertex during which it is not capable of being responding after it has been activated.

To generate an embedding of the network dynamics, first we first need to determine which part of the dynamical regime to capture by finding the steady-state statistics of the system (SS-S) of the dynamics. The SS-S of the system consists of a steady-sate starting point and the steady-state period, The steady-state starting point is the time point when the network's node activity enter a periodic orbit, that is starts to repeat. And the steady-state period is the time interval or the number of steps of one period of steady-state activity. We propose three algorithms to determine the SS-S of the network dynamics, two of complexity $O(n^3)$, and one of complexity $O(n)$, where n is the number of time samples under consideration. With the SS-S in hand, we construct a graph representing the network dynamics. Then using the graph embedding, we construct a vector space embedding[54, 47]. The vector space embedding represents the enumeration of sub-graphs of the same height within the full graph of the network dynamics, and the interaction of those sub-graphs. Both the graph representation and the vector space representation allows us to quantitatively study the interaction between the network's structure, resulting dynamics, and network initial conditions. The results we discuss here represent an important step towards the efficient construction of an end to end machine learning architecture that makes use of our geometric dynamic model capable of adaptive on the fly data encoding and learning.

3.3 Paper Organization

The paper is organized as follows: First we discuss some preliminaries. The Preliminaries section includes the definitions we will use in the Methods sections. Next, we present a mapping from a dynamical description of the network dynamics into symbolic descriptions of network states. Then describe what we mean by the term steady-state statistics. After which we present the data structures which are used as input to the summation and string search methods. We provide a concrete example of an approach to generate the input to the string algorithm.

After the preliminaries, we will introduce the methods to find the steady-state statistics. Using the steady-state statistics, we will discuss how tp transform the observed dynamics of the network into a directed graph. Finally, we propose a path to transform the graph previously constructed in to a vector embedding. This provides a high-dimensional representation of the input embedded in the resulting dynamics.

3.4 Preliminaries

3.4.1 Network Description

The structural neural network is described by a directed graph[44, 23]. We define a graph G as tuple $G = (V, E, l)$, where V is the set of vertices $V = \{v_1, v_2, \ldots, v_n\}$, where n is the number of nodes in the network, E is a set of directed edges $E \subset V \times V$, and l is the label function which maps the vertices to an alphabet $\mathcal{A}$, $l : V \to \mathcal{A}$. Since the inspiration for this work is derived from analysis of biological neural networks, we interchangeably call nodes neurons, and edges axons. Each neuron in G takes on a binary state at a time t. The state of the i^{th} neuron in G is given by $\omega_i(t) = \{0, 1\}$. An axon from neuron i impinging upon neuron j is given by edge $e_{ij} = \left(v_i, v_j\right)$, where $e_{ij} \in E$. Every axon in G has an associated delay, τ_{ij}, which is the time it takes a signal originating at neuron i to reach neuron j. In the general case τ_{ij} can be a function of several

variables, but for the purposes this work it is kept constant, after network initialization.

3.4.2 Dynamical Description

Following the work of Cessac etal.[12, 11], we will attempt to construct a mathematical bridge from a dynamical system description to a symbolic coding description. In this work we only consider the deterministic evolution of the network activity, a probabilistic formulation is left for a future work.

Let $V_k(t)$ be the membrane potential of neuron $k \in 1 \ldots n$ at time t. $V(t) = [V_i(t)]_{i=1}^{n}$ is the vector representation of the membrane potential of the entire system. $V(0)$ is the initial condition of the system. The bounds on the values assumed by the membrane potential is given by $V_k(t) \in [V_{min}, V_{max}]$. The threshold for firing is some value $\Theta \in [V_{min}, V_{max}]$. For a trajectory V, the firing times of neuron k is given by:

$$t_k^{(m)}(V) = inf\left\{t | t > t_k^{(m-1)}(V), V_k(t) \geq \Theta\right\} \tag{3.1}$$

As such, $t_k^{(m)}$ is the m^{th} firing of neuron k. As an initial condition for the set of firing times we set $t_k^{(0)} = -\infty$. After firing, the neuron's membrane potential resets to some value V_{reset}.

$$V_k(t) \geq \Theta \implies \lim_{\Delta \to 0^+} V_k(t + \Delta) = V_{reset} \tag{3.2}$$

As with $V(t)$, $V_{reset} \in [V_{min}, V_{max}]$.

To generate a symbolic coding description, we introduce the notion of a raster plot. A raster plot is a sequence $\widetilde{\omega} = \{\omega(t)\}_{t=0}^{+\infty}$ of vectors $\omega(t)$, where $\omega(t) = [\omega_k(t)]_{k=1}^{n}$. When neuron k fires, t, $\omega_k(t) = 1$, and when neuron k is quiescent $\omega_k(t) = 0$.

For the purpose of generating a raster plot, the membrane potential of a neuron can be decomposed into two intervals, $I_0 = [V_{min}, \Theta]$ and $I_1 = [\Theta, V_{max}]$. If neuron k is quiescent then

$V_k \in I_0$; otherwise, the neuron has fired. Let $\Lambda = \{0,1\}^n$, and let $[\omega_k(t)]_{k=1}^n \in \Lambda$. Each point $V(t)$ corresponds to a firing state $\omega(t)$ whose components are given by:

$$\omega_k(t) = Z[V_k(t)] \tag{3.3}$$

where Z is defined using the indicator function denoted by χ:

$$Z(x) = \chi[x \geq \Theta] \tag{3.4}$$

The dynamic evolution of the system is given by:

$$V(t+\Delta) = F(V(t)) \tag{3.5}$$

Where $F = [F_i]_{i=1}^n$. In the integrating regime of the neuron, the model evolves as follows:

$$F_i(V) = \gamma V_i(1 - Z[V_i]) + \sum_{j=1}^{n} w_{ji} Z[V_j(t - \tau_{ji})] \tag{3.6}$$

Where γ is some decay function[49], and w_{ji} is the coupling strength between neuron i and j, and τ_{ji} as defined earlier is the delay between neurons i and j.

3.4.3 Description of Steady-State Statistics

Here we define the terms starting point and period of steady-state activity of the network dynamics. It is defined as the repeating or periodic set of network states the dynamics settles to after some transient activity. Given our network construction, the regimes of network activity are (1) persistent activity, and (2) neural death, ie, cessation of all activity in the network. Although neural death is a type of steady-state, it is the trivial case in this work. Persistent activity can either eventually enter a periodic orbit of network states, or the network continues to exhibit

non-periodic but persistent activity. In situations where it is difficult to analytically determine a network's steady-state behavior, we forward calculate the network activity to computationally tractable limits, and analyze the resulting dynamics to determine the network's steady-state statistics.

Let us say that the state of the network is given by some function $x(t)$. We say that the network, has entered steady-state when for $\forall t > t_{ss}$, when:

$$x(t_{ss} + c) = x(t_{ss} + c + T) \tag{3.7}$$

where t_{ss} is the steady-state starting point, T is the steady-state period, and c is some arbitrary positive time shift:

The transient phase of $x(t)$ is the evolution of $x(t)$ until it enters a periodic orbit. We define the starting time/time-index of steady-state activity as the end of the transient phase. Once the network activity has entered a periodic orbit, we define the period of steady-state activity as the amount of time/time-indices it takes for the activity to repeat. In large and complex networks, for example one with many nodes, or with some of feedback mechanism which modifies network parameters as a result of network activity, it is quite possible that the forward computation will not have been calculated for a long enough data window for the entire network to exhibit steady-state behavior.

3.4.4 Sum Description

Here we describe the input to the summation methods. We consider the subset $\mathcal{M} \subset \tilde{\omega}$:

$$\mathcal{M} = \{\omega(t) | \exists \omega_i(t) = 1\} \tag{3.8}$$

In words, $\mathcal{M}$ is the set composed of vectors $[\omega_i(t)]_{i=1}^{n}$, whose elements are vectors of node activations. Each vector in the set $\mathcal{M}$ contains at least one node activation. Let $m \in \mathcal{M}$, and

let us preserve the time ordering of the elements of $\mathcal{M}$ by indexing them, where, $m[i]$ is the i^{th} activation vector of $\mathcal{M}$. Note, that at each t, the corresponding vector representation of the slice of $\mathcal{M}$ contains value 1 at the dimension corresponding to the nodes which activated, and 0 in the dimensions corresponding to quiescent nodes. This form of definition opens up the possibility of finding the steady-state statistics in an asynchronous event based simulation framework[6, 53, 37]. We take advantage of the non-uniform sampling in trying to ascertain steady-state statistics in the network setting. To use the summation method with uniform sampling we use $\tilde{\omega}$, instead of $\mathcal{M}$, keeping all the sampling points which populate ω between firings. Analysis in the uniformly sampled setting comes at the computational cost as a function of the sampling rate. For networks with evolving network parameters which affect the sampling rate we may not know apriori the most optimal sampling rate.

Here we make a distinction, not fully developed in this work, which is one between analyzed nodes and observed nodes. The observed nodes are all the nodes in the network for which we have recorded data. The analyzed nodes are a subset of the observed nodes which we may deem a representative set of nodes. We make this distinction because it may not be necessary to analyze all nodes in the network to determine steady-state statistics of the entire network.

3.4.5 String Description

To apply string run finding algorithms, we must convert the dynamic evolution of the system to a string description. We denote the sequence of firings as the set:

$$\tilde{\omega} = \{\ldots \omega(s) \ldots \omega(-1), \omega(0), \omega(1) \ldots \omega(t) \ldots\} \tag{3.9}$$

And we denote the right-sequence of node activations after some initial conditions, for example at $t = 0$, as:

$$\tilde{\omega}^{+} = \{\omega(0), \omega(1) \ldots \omega(t) \ldots\} \tag{3.10}$$

To get a string representation of the sequence of firings, we define a labeling function which maps firings to an alphabet of fixed length strings of size $s \in \mathbb{N}$. The alphabet is denoted as Σ_s. $\mathcal{L} : \omega \to \{\Sigma_s, \varepsilon\}$, that is, $\mathcal{L}$ maps activations of neurons to fixed length strings, Σ_s, and quiescent states to the empty string, ε. Let σ_{s,ω_i} be the symbol associated with the activation of the i^{th} node. For brevity, we will drop the subscript s and assume a fixed length string in place of the symbol. Note that we do not distinguish in the string the time of firing, the string only contains information about the set of node which activated. We define $\mathcal{S}$ as the set of strings representing the evolution of the network. The ordering of this set is based upon the ordering of $\tilde{\omega}$ with respect to time. The input to the string run finding algorithm is

$$\mathcal{S} = \left\{ \ldots \sigma_{\omega(t-1)}, \sigma_{\omega(t)}, \sigma_{\omega(t+1)} \ldots \right\} \tag{3.11}$$

To extend the string description for use with uniform sampling, the labeling function would assign a symbol or a set of symbols to the quiescent states of $\omega(t)$, rather than mapping those states to ε. Uniform sampling would significantly increases run time by a constant multiple c, the number of bits or symbols used to describe each discretized value. Resulting in a time complexity of order $O(cn)$, where n is the number of samples.

3.4.6 String transformation

In this section we present a simple technique to convert a given discrete signal $x(t)$ into a string. We begin by assuming that this system has n nodes which fire. We then enumerate each node arbitrarily, and keep it fixed over the course of the analysis. We start with an empty string, and since our system has information about which node fired, if node i fired at time $t = 1$ we simply append i to the string, as such, we append to the string each subsequent node firing. In the case of multiple node firings at the same time, we generate a new enumeration and append it to the string. Consider the following network with two nodes showing the following firing

pattern (Fig. 3.1) That is, node 1 fires at t_1, node 2 fires at t_2, both fire at t_3 and so on. Now we enumerate the event of the two nodes firing simultaneously by an arbitrary character, say "a", and return the following output: *12a12a212*. By doing so we convert our signal into a string and reduce the problem of finding steady state in a complex signal to only finding a repeating substring. Note that when encountered by events such as simultaneous firing, we use an unused character to represent it and store the mapping in a hash table.

3.5 Methods

3.5.1 Steady-State Detection

To tackle the the issue of finding Steady-State Statistics we propose three approaches, namely, the Signal Summation approach, the Naive String Search approach, and the Efficient String Search Approach.

Signal Summation

Using as input the $\omega(t)$ constructed in Sec. 3.4.4, we describe an algorithm to determine the steady-state statistics of the system (SS-S-S) (Sec. 3.4.3). This approach is based on the network's dynamics from the description in Sec. 3.4.4 of the form $\omega(t)$. The following algorithm is also extendable to the non-uniform sampling setting using $m[i]$, where i is the index. The methods presented in this paper can only determine that SS-S up-to the extent of the forward calculation. The longest period this algorithm can determine is of size equal to half the data.

In the general discrete case, for a system of N nodes, the state of node j is given by $\omega_j(t)$ (eq. 3.3) where $\omega_j(t) \in \{0,1\}$. We assume a node only fires at most once per time point. Each $\omega_j(t)$ is defined over the interval $t \in [t_0, t_f]$, where t_0 is initial observation time, and t_f is the final observation time. The signal $\omega_j(t)$ is comprised of two parts, the transient part, $g_j(t)$, defined over the interval $t \in [t_0, t_{SS_j} - \Delta]$, and the periodic part $h_j(t)$ defined over the interval $[t_{SS_j}, t_f]$,

for some arbitrarily small Δ. The network's activity can be described by the following set of equations:

$$\begin{aligned}
\omega_1(t) &= g_1(t) + h_1(t) \\
&\cdots \\
\omega_j(t) &= g_j(t) + h_j(t) \\
&\cdots \\
\omega_N(t) &= g_N(t) + h_N(t)
\end{aligned} \tag{3.12}$$

Since $h_j(t)$ is periodic, and given the form of $\omega_j(t)$, we can describe it using a sequence of delta functions:

$$h_j(t) = \sum_{m=1}^{r} (\delta(t - mb_1) + \cdots + \delta(t - mb_n)) \in \{0, 1\} \tag{3.13}$$

There are a total of n activation per period. Let b_i be the time of the i^{th} activation in some m^{th} period, where m is a multiple of the fundamental period. Let r be the total number of periods. Note, care must be taken for the final period in the data window, as it is likely to be a fraction of period.

To find the SS-S-S, we first find Steady-state Statistics of $\omega_j(t)$ (SS-S-j). SS-S-j is comprised of the Steady-state Start Point(SS-SP-j) and it's Steady-state Period(SS-P-j). Let t_{SS_j} be SS-SP-j, the start of $h_j(t)$, and SS-P-j is the fundamental period of $h_j(t)$. After all the SS-SP-j's are determined, the latest starting-point of steady-state across all ω_j is used as the Steady-state Start Point of the system (SS-SP-S). We define the Steady-state Period for the system (SS-P-S) as the longest period across all $\omega_j(t)$. Under the non-uniform sampling setting (eq. 3.8) this technique may not capture the correct steady-state statistics. This happens when there is a contraction or expansion of timescales over node firings take place, but the sequence of node firings do not change.

For some i^{th} node in the network, we construct $z_i(k, T)$ to determine the SS-S-i. Here k is

the candidate SS-SP-i, and T is the candidate SS-P-i.

$$z_i(k,T) = \int_k^{k+T} \omega_i(t)\,dt \tag{3.14}$$

The longest period sets the upper-bound for T, while the lower bound of T is fixed to two consecutive time points. Let $t_i \in [t_0, t_f]$ be the time point when $\omega_i(t)$ initiates steady-state behaviour. We assume for all $t > t_i$, the signal $\omega_i(t)$ is in steady-state. Let $\psi_i \in [0, \frac{t_f}{2} - t_0]$ be the period of $\omega_i(t)$'s steady-state. We evaluate $z_i(k,T)$ over the candidate periods $T \in [0, \frac{t_f}{2} - t_0]$ and candidate starting points k. Our goal is to find the correct ψ_i and t_i. The regimes of operation where we evaluate eq. 3.14 are:

1. $k \in [t_0, t_i)$, $T \in [0, \frac{t_f}{2} - t_0]$
2. $k \in [t_i, t_f]$, $T \in [0, \frac{t_f}{2} - t_0]$
   (a) $k \in [t_i, t_f]$, $T \in \{\psi_i, 2\psi_i, \ldots n\psi_i\}$
   (b) $k \in [t_i, t_f]$, $T \notin \{\psi_i, 2\psi_i, \ldots n\psi_i\}$

We define $f(x)$ to be uniformly zero starting at some constant c, when the function stabilizes at the value of zero after some point c, i.e.,

$$\forall x \geq c\,, f(x) = 0 \tag{3.15}$$

We will use the notion of uniformly zero to analyze the results of Eq. 3.14.

To determine the SS-S-i, we look for the smallest k and smallest T where $\frac{\partial z_i(k,T)}{\partial k}$ and $\frac{\partial z_i(k,T)}{\partial T}$ are uniformly zero. Let

$$A = \left\{ (k,T) \mid \forall k > c, \frac{\partial z(k,T)}{\partial k}\Big|_{T \in [0, \frac{t_f}{2} - t_0]} = 0 \right\} \tag{3.16}$$

First, we determine the SS-SP-i (i.e. t_i), by finding the minimum k in A:

$$t_i = \left\{ k \mid \inf_k A \right\} \tag{3.17}$$

We use the minimum k, to find ψ_i by evaluating A for the minimum T:

$$\psi_i = \left\{ T \mid \inf_T A \Big|_{k=t_i} \right\} \tag{3.18}$$

The uniformly zero condition is satisfied for a short trailing set of start times/indices (see Figs: 3.2c, 3.2f, and 3.4b). This is because two adjacent time units can have the same envelop (eq. (3.14)) for some k in A satisfying the uniformly zero condition of eq. 3.16. This issue is mitigated by considering the k infinum of A.

This process results in a set of values for t_i, but only one of those is the SS-SP-i. Taking the infinum removes the false candidate for t_i. Likewise, to find the period we take the infinum of T when we consider the case:

$$\frac{\partial z(k,T)}{\partial k}\Big|_{\substack{k=t_i \\ T\in\{\psi_i, 2\psi_i, \ldots n\psi_i\}}} = 0 \tag{3.19}$$

Once we determine t_i and ψ_i for each node, the maximum values across all nodes are selected for the steady-state statistics of the system.

String Search: Naive $O(n^3)$

The input to the string search uses the construction from sec. 3.4.5. We introduce the string run finding algorithm by first presenting a naive approach. Here we do a brute force search to identify a repeating, terminating substring in our signal-converted string. To do so, we loop over the period of the expected steady state pattern, and for each period we loop over the start time, essentially testing all $n(n+1)/2$ substrings as the candidate steady state pattern. Formally,

given a string of length n we find the pattern using the following psuedo-code:

Algorithm 2 Naive Find Steady-State Statistics

```
1:  procedure FINDSTEADYSTATE
2:      str ← signal
3:      strlen ← length of signal
4:      perlen ← length of candidate period length
5:      index ← length of current position in string
6:      for perlen = 1 to strlen do:
7:          for index = 0 to strlen-1 do:
8:              substring = str[index:index+perlen]
9:              if CheckTerminating(substring, str, index, perlen) then
10:                 return substring
11:             end if
12:         end for
13:     end for
14: end procedure

15: procedure CHECKTERMINATING
16:     str ← signal
17:     substring ← candidate steady state
18:     index ← candidate start point
19:     perlen ← length of candidate period length
20:     while i = index to strlen-perlen do:
21:         if substring != str[i:i+perlen] then
22:             return False
23:         end if
24:         return True
25:     end while
26: end procedure
```

That is, we consider every possible continuous substring and check whether it is the required pattern, and we stop when we find one, yielding the shortest one. A basic analysis of the psuedo-code yields an $O(n^3)$ time complexity. We improve this in the next section.

String Search: Efficient O(n)

The efficient algorithm uses as input $\mathcal{S}$ which is based on the right-sequence of node activations, $\tilde{\omega}^+$ as defined in sec. 3.4.5. Let $x[i]$ denote the i^{th} location in $\mathcal{S}$. The modulus operator

can be used to map from x to $\omega(t)$. In stringology[18, 55, 25, 29], a string, x, of length n, has period, p, if $x[i] = x[i+p]$, for any i, such that, $1 \leq i \leq n-p$. If we were to consider the case where string x can be decomposed into sub-strings u, v, and w, such that $x = uv^k w$, then string u is considered the prefix of x, and string v is the period, repeated k times, and the string w is considered the suffix.

Due to our interest in finding the starting time/time-index and period of steady state, we are concerned with strings which end in repeats. More specifically, strings that end at some location in the period v, in the course of a run([36]). In stringology, a primitive string is a string that is the fundamental period of the repetition (and can not be further compressed into the form v^k). The leftmost primitive string is called maximal, that is, if the primitive string can not be shifted anymore to the left. If the primitive string could be shifted more leftward, a new prefix and an addition to the k^{th} repetition would result, and that point is leftmost maximal. A run is defined as the maximal fractional primitively rooted repetition (example: $12312312312 = (123)^{3\frac{2}{3}}$, where the run is 123), and takes into account the stop of the forward calculation at some fraction of the period. Now we have the stringology terminology to describe our needs. We are interested in strings which end in runs, i.e. periodic sequences. Moreover, we are interested in the leftmost primitively rooted repetition of that last run. That is the starting point of steady-state.

The problem of finding runs in strings has been solved with O(n) time algorithms ([33]), where n is the length of the string. We use the implementation courtesy of Dr. Hideo Bannai of Kyushu University in Japan called runFinder[1]. Modern run finding algorithms[13, 45, 17, 25, 26] rely on building and analyzing data structures which are amenable to finding repetitions in strings. The following is the procedure used in the algorithm:

(1) A suffix array[38] is build using[40].

(2) The Longest Common Prefix is calculated[30].

(3) Lempel-Ziv factorization[56, 16] is used to find all leftmost runs.

(4) Kolpakov and Kucherov's[33] linear time algorithm is used for finding all maximal

repetitions.

Once the maximal runs have been found, we sort through the runs to find those ending at the input string's final character position. If there are multiple candidate runs, we pick the candidate which has the minimum starting position. The primitively rooted repetition is the period of steady-state activity.

3.5.2 Constructing Graph Representation of Network Dynamics

Using the causal node activity, we construct a graph representing the dynamics on the network. The nodes of the graph are (v_i, t_k), where v_i is the neuron which fires at some time t_k. The edges of the graph represent the causal activators[49] of (v_i, t_k). Every time a neuron is activated a new node, (v_{i+1}, t_{k+1}), is added to the graph. This way we are able to distinguish repeated node firings, while still being able to construct subgraphs of neuron-neuron interactions.

The initial conditions and external perturbations to the network are represented by node without causal predecessors. The child nodes are the subsequent nodes which are causally activated. The steady-state statistics of the network dynamics bounds the construction of the graph.

3.5.3 Embedding Dynamics using WL Scheme

To transform the graph data structure to a vector representation we use the Weisfeller-Lehman(WL) subgraph embedding technique[47, 54], a plethora of other data structures and embeddings can be used[9]. Since our interest is in the interaction of nodes and pathways of network dynamics, we chose WL subraph embedding to illustrate the process of transforming neuronal dynamics to a vector embedding.

If one were interested in analyzing the dynamics within the core of the network itself rather than the graphs which are formed due to the interactions with the sensory layer, i.e., the

initial conditions, we must make a conscious effort to blind the algorithm of the sensory nodes. An easy way to do accomplish this is to assign all the sensory nodes the same label. Doing so will make resultant relabeling and therefore subgraphs dependent only upon the patterns of activity within the core of the network. Obfuscation of the input layer will also allow us to compare the ability of the network's dynamics to characterize the inputs rather than using the inputs, through the initial conditions, as the differentiating signature of the dynamics.

3.6 Results

3.6.1 Summation Method, Uniformly Sampled 1 Hz Signal

To illustrate the summation method methods we first show the results on a simple idealize waveform which consists of a $1Hz$ impulse train (fig. 3.2a), sampled at $100Hz$. The signal consists of a sequence of zeros and ones. Fig. 3.2b shows the sum of the spikes over the various candidate periods and starting points, a reflection of eq. 3.14. Fig. 3.2c shows the change in the number of spikes as a function of starting positions, a reflection of eq. 3.16.

The red horizontal lines in fig. 3.2c indicate the candidate periods and starting points which corresponds to the condition eq. 3.15 applied in eq. 3.16.

The candidates periods fall into two categories. Category 1, encompasses the two red horizontal lines which share a start time of $0s$, and category 2 encompasses all the other red horizontal lines which start at start time $1s, 2s$ and $3s$. For the signal's SS-S we choose the earliest start time eq. 3.17, and the shortest period eq. 3.18. The period of $1s$ explains the data from a start time of $0s$ onward. The other red horizontal lines are a result of the envelopes for some range of (k, T) in eq. 3.14 as noted in the methods (sec. 3.5.1), meeting the condition eq. 3.15, but explaining less of the data.

3.6.2 Uniformly Sampled Randomized Pulse Train

We now consider a sequence of impulses which are poisson distributed as shown in Fig. 3.2d. In Fig. 3.2e we observe that there are bands of values, although not horizontally contiguous as is our requirement for steady-state statistics, of start times and period lengths which have the same number of spikes. Our intention for future work is to consider noisy signals, and one way in which we can handle those is by relaxing the criteria on the change in the number of spikes in adjacent starting points.

Given the stochastic nature of firings, Fig. 3.2f no longer exhibits the kinds symmetries which were found in Fig. 3.2c. All the zero differences are relegated to a few starting points at the end of the data-window, and there is no period of negligible size, or easily discerned harmonics thereof. In the case that the system is not actually in a regime of periodic activity, we may end up with candidate periods and starting points due to an unchanging number of spikes in the last few starting points, k, where the uniformly zero criteria is met. One must make sure that the forward calculations are done for long enough such that the system is actually in steady-state. A simple workaround is to consider steady-state statistics such that the forward-calculations are done for a data-window long enough such that the starting point of steady-state must be some fraction of the data-window.

3.6.3 Summation Method, Non-Uniform Sampling, Steady-State Statistics in Network Application

Here we illustrate an application of the steady-state finding algorithms as it pertains to discerning steady-state statistics in network dynamics settings. The network under consideration is illustrated in Fig. 3.3a. We introduce an image of a hand written digit from the MNIST database [35] as binary vector at time $t = 0$ to the input layer nodes of the network, and analyze the resulting dynamics. In order to get a sense of the data we wish to analyze, we create a raster

plot of the activation of the output nodes (Fig. 3.3b).

Following the procedures outlined in the methods, we show the start time index and the size of the periods in units of time indices Figs. 3.4a, 3.4b. Note that each iteration through the method is done on a per node basis, and the steady-state statistics of the system are are taken to be the latest start time and the longest period among all the analyzed nodes.

As with the previous examples, the smallest value on the abscissa where the red horizontal line begins is the start of the steady-state behavior. Each red line, of the set of longest red lines, on the ordinate represents the harmonics of the period of steady-state activity.

Note the switch from using seconds to using time indices. The data fed into the steady-state finding algorithm is no longer uniformly distributed, but used from eq. 3.8. Non-uniform sampling is used here to reduce the computational complexity of the problem by using the network's activity itself as the sampling points. While the start index corresponds to a specific absolute time value, the number of time indices which make up a period corresponds to a range of absolute time values. While evaluating eq. 3.14 for a fixed period, in units of time indices, across start time indices, we observe that the fluctuations in period length measured in absolute time, seconds, between start times. This is due to the way we non-uniformly sample and the fact that the absolute time between adjacent activations vary, therefore moving one time index does not correspond to a move of one absolute time unit. Fig. 3.5a, illustrates the variation in the absolute time for a fixed period as a function of starting time index.

Each horizontal line in Fig. 3.5a corresponds to a harmonic of the period. The fluctuations in a particular horizontal line in Fig. 3.5a correspond to the aforementioned absolute time between the start of the period and end of the period. Fig. 3.5b shows the number of spikes which are present in each harmonic of the period. There is a one-to-one correspondence between Figs. 3.5a and 3.5b in terms of bottom to top ordering of curves. As expected, and needed, once the system is in steady-state the number of spikes in a period stabilizes to the steady-state number of spikes in a period. The number of spikes in a period is unchanging over a period for $t \geq t_{SS}$. Each

successive vertically increasing curve in Fig. 3.5b shows the number of spikes in harmonics of the period.

Issues

Some issues exist in using the summation method in the non-uniform sampling setting, here we discuss a few. This method has the shortfall that variations in network activations which do not cause a reordering of spikes will not be distinguishable, therefore, any drift in spiking activity without causing a reordering of $\mathcal{M}$ will be treated the same.

Since we are using the latest start time and the longest period, and although we are assured that all the other nodes in the network will be in steady-state, we may not capture entire integer multiples of full periods of all of the nodes. There may be fractions of periods, of some nodes, contained in the longest period. If the requirement were to capture integer multiples of the periods, then the greatest common denominator of all the periods may far exceed the observation time of the network.

Another issue which exists with the summation method is that the network states which are under consideration for the determination of steady-state are only the node activations. Cases where nodes activate due to different causal activators but result in the same pattern of network activity can exist, but the summation method is not able to distinguish between the different causal activators. We resolve this issue during post-processing. The one and zeros of the signal are broken up based on the steady-state statistics, and the signals are aligned on a per period basis. Preceding the vector positions of the activations, the causal activators are inserted, and the dimension of the data vector is increased as a result. Finally, the vectors of causal activators, as well activations, are compared for alignment across periods, to make sure that there is no difference between the activators which give rise to the firing's in all the periods until the end of the observation window. When comparing periods, one may not capture integer multiples of a period, therefore, care must be taken to consider the final fractional period.

The final issue with the sum method which we will discuss is that activations in the initial period may be erroneously included in the steady-state period or as the steady-state starting point. This issue arises at it's core due to non-uniform sampling. Because the sum method is comparing a count of activations within a period's time window, and because we slide that window across the candidate starting points, a node activation not really part of the steady-state set of node activations may be inadvertently included. When that erroneously included node activaiton happens to lie close the first actual node activation of the period, where we define close as-a distance in time indices less than the spacing between any of the node activation of the actual period. Outside the first period, this is not a issue which will arise due to the strict requirement that number of spikes in the smallest period from start to finish must be maintained between all consecutive starting points. Alignment of activations, as done to detect difference in causal activators, is one way to detect this issue. When an erroneous spike is detected one solution is to remove the erroneous spike from consideration, by flipping the corresponding bit, and rerunning the algorithm and rechecking for alignment.

3.6.4 Steady-State Statistics using efficient run-finding algorithm

We apply the run-finding algorithm to the strings generated by the introduction of the MNIST images to the network, as in the previous section. Since we have a total of 793 nodes in the network (784 sensory nodes associated with the pixels in each image, 5 core nodes, and 4 leaf nodes), we use a string of 3 symbols. Node 1, will have a label 001, node 2 will have label 002, and so on up to node 793 with label 793. We do not delimit the symbols with commas or other delimiters as those symbols increase the processing time.

Table 3.1 shows a sample output for illustration. For a string of length 8586 symbols, which translates to 2862 node activations, the run-finding algorithm finds a string of length 81 symbols, which is 27 node activations in a steady-state period, and a run of length 6090 symbols, which translates to 2030 node activations. As with the summing method, using the information in

the table below we are able to construct a graph of activations, based on set of causal relationships between node activations up to the point where the dynamics stop producing new information.

3.7 Discussion

In this work we have proposed a path to ascertain vector embeddings from the observed dynamics on a network. To get a complete representation of the information contained in the dynamics one must find the steady-state statistics. Using SS-S as the basis, a graph representation, which is one of perhaps several to describe the dynamics, is generated.

Large and/or more more complex networks, one which includes feedback or variations in the network parameters, pose challenges to finding the steady-state statistics. Two of those challenges are that the network dynamics may not entirely enter steady-state behavior, or the network dynamics may need too much time to enter the steady-state regime. One may only be interested in stability of the network dynamics indicated by periodic activity for some duration of time, we call this transient stability. The methods to find steady-state statistics supports the ability to find repetitions of periods lasting some duration of time. In the case of the string matching method, it is just a filter on the length of runs, and in the case of the summing method, a filter on the length of zero derivatives rather than a filter on the uniformly zero criteria.

Another issue with finding the steady-state statistics on complex networks is that it may be computationally infeasible to measure the dynamics of the entire system. A solution to this problem may be to observe the dynamics at a user defined subset of the network and depending on the network create some confidence measure as to the likelihood of the system being in the steady-state regime based observed subset's dynamics. We have some preliminary results which show that analyzing the dynamics some subset of nodes does in fact result in the same steady-state statistics as analyzing all the nodes in the network(should i put the data in the appendix, i removed it because it was for a different MNIST example, and detracted from the main point of the paper).

In large and complex network analysis, we suspect that the nominal use-case in analyzing networks will be a combination of the two previously discussed topics. One maybe interested in only a subset of the network exhibiting transient stability. When transient stability has not yet been achieved by the network, after some duration of observation time, one may wish to continue the forward computations and along the way check for steady-state behavior using whatever user defined criteria for the amount of steady-state activity which will convince the user that the system is in fact in steady-state, for example some fraction of the captured observations exhibit steady-state behavior. Online methods, those that do not require the recomputing of entire data structures previously computed, would be quite useful to run in parallel with the forward computation. The less computationally efficient summing method can easily be extended into an online algorithms since each additional node activation is does not require the recomputing of previously computed values. The string searching algorithm must be rerun in entirety every time we wish to analyze a new string, even one generated by the addition of one activation to the end of the string. An online algorithm algorithm which requires minimal recomputing would aid in inline network dynamics analysis.

An extension of the application of the string searching algorithm to elucidate steady-state statistics is to include additional observed network states and parameters, for example, causal activators, signal contributions, times between firings, etc. This can be done by extending the library of symbols to be included in the string paying relatively small computational penalty. For example, in the case of adding causal node activators, lets say there are, on average, three activators required per activation, then we will be extending the size of the string by a multiple of three. Any parameter that can be translated into a symbolic representation, which is pretty much all information stored in a computer can be included in the string representation for analysis. It is up to the user to determine the appropriate parameters for inclusion.

Finding approximate periods, that is a minimal number of symbol switches which results in a repeating sub-string is also of interest. This problem has to some extent been resolved in that

algorithms robust to substitution resulting in repeats exist, and algorithms which result in ”fuzzy” repeats exist based on graph edit or Levenstein distance. As such some sources of noise such as aberrant activations or approximate periods can be determined using the string technique.

The vector embedding process of the network activity is agnostic of the steady-state finding process. Any embedding process will need to be informed of how much of the dynamics to include in the embedding.

Since the main focus of this work is to put together a workflow which enables systematic analysis, we are now in a position to evaluate various network models and parameters. The modularized nature of our workflow enables us to test not only various network models, but also node models, more informative embedding schemes, and using other back-end classifiers and techniques to discriminate data, as well as analyze different regimes of network activity based on various definitions of network dynamic stability.

3.8 Conclusion

In the preceding we show how to map neural network dynamics into a vector representation. We propose the steady-state criteria to determine how much of the dynamics to capture, and we outline methods to do so. In circumstance where one is analyzing large or complex network dynamics, our methods are extensible to relaxed constraints, such as, finding transient stability, and analyzing subsets of network dynamics.

Each steady-state statistics finding method provides different boons. The summation method can be extended to the probabilistic setting by relaxing the uniformly zero criteria, whereas, the string search method utilizes an efficient algorithm.

Inspired by theories from neuroscience we believe the graph representation of the network dynamics provides a straightforward mapping which preserves some salient information (references about neural patterns being important). The WL subgraph embedding scheme is used

because we are interested in the interaction of various activity patterns, and at various levels of organization. Specifically, the WL scheme captures, the interaction between nodes, interaction between subgraphs of neighbours, and interaction of subgraphs of larger heights. Although we propose a specific path to ascertain a vector embedding, one is encouraged to build more general mathematical bridges between network activity and forms of representation, such as topological constructs[19] to represent the dynamics.

These tools provide a basis upon which we can analyze different network designs and organizations, compare various node models. The vector representation lets us analyze the results using a plethora of existing tools from algebra, such as norms and differences.

3.9 Appendix: Network Description

3.9.1 Preliminaries

The structure of our network can be described by a simple directed graph[23]. The construction of the network we consider for this work is from[49], whose work also introduces the Geometric Dynamic Perceptron Node model.

We define a graph G as tuple $G = (V, E, l)$, where V is the set of vertices $V = \{v_1, v_2, \ldots, v_n\}$, E is a set of directed edges $E \subset V \times V$, and l is the label function which maps the vertices to an alphabet Σ, $l : V \to \Sigma$. Each node in G takes on a binary state, the state of the i^{th} node in G is $X_i = 0, 1$. Each edge $e_{ij} = \left(v_i, v_j\right)$ in G has an associated delay, τ_{ij}, which is fixed after initialization.

3.9.2 Network Construction

Our network consists of three layers, an input layer, a recurrent/hidden layer, and an output layer, Fig. 3.3a shows a graphical description. The input layer contains 784 nodes, one for each

pixel of the MNIST data set. The recurrent layer contains 5 nodes, and finally the output layer contains 4 nodes. Connectivity of the nodes in each layer are as follows, edges from the input layer to the recurrent layer are connected in a feed-forward manner, that is, there is a directed edge from each node in the input layer to every node in the recurrent layer. Within the recurrent layer, each node is connected to every other node. The nodes between the recurrent layer and the output layer are connected in a feed-forward manner, similar to edges between the input and recurrent layer. Finally, at the output layer there are no edges emanating from any of the output layer nodes.

Each edge has an associated delay. All the edges in the network used for the results section were drawn from a uniform distribution.

3.9.3 Node Model

The threshold for each node in the network is set to an amplitude of 2 units. Once the voltage of the node is above threshold, a signal of amplitude 1 unit is transmitted along all the outgoing edges of that node. Once a signal reaches its destination node along an edge, a decay is applied to that signal. Each individual signal being summed at a node is separately exponentially decayed, using a decay constant of xyx. The synaptic weights are set to unity, the network dynamics are only a function of the edge delays. We set the refractory period of all nodes to $0.0625s$.

3.9.4 Network Initial Conditions

The initial conditions of the network are set in two parts, one is at the input layer, the other is for the rest of the network. At the input layer we select a subset of nodes to be active at time $t = 0$, the initialization can be done in whatever way necessary to sufficiently excite the network such that non-trivial SS exists. Subsequent layers, and all their associated edges are quiescent.

3.9.5 Data structure

Upon stimulating the network at the input layer, we observe the spiking activity in the recurrent and output layers. A spike is the activation of a neuron and corresponds to the change in membrane voltage. In this network, which is a simplification of the neurobiological process, we define a spike to be the transition of the state variable Y from $0 \rightarrow 1 \rightarrow 0$. The spike time is the time at which a spike is generated, and the set of times, S for a node i is written $S_i = \{t|Y_i(t) = 1\}$. We observe that the spike times are points in time not necessarily picked from uniformly sampled values.

Next we place the activity of node i in the context of the entire network. To do so we define the notion of a spike train. Let us denote a set T_i to be the spike train of node i. An element of the set $T_i(t) = 1$ takes on a value of 1 when the associated $Y_i(t) = 1$, and $T_i(t) = 0$ when $Y_k(t) = 1$. At some time t_r, in the case where $Y_i(t_r) = 1$, and $Y_k(t_r) = 1$, we set $T_i(t_r) = 1$. Now for any i, $T_i(t) = \{0, 1\}$.

Table 3.1: A table containing the sample outputs of the steady-state statistics of the system using the string finding method. Each of the entries are in either String Character Units or refer to specific Node Numbers in a sequence. String Character Units refer to the index(rows 1 and 2) or interval(row 3) of a symbol or symbols in a string sequence. When mapping from String Character Units to network node activations care must be taken to account for the length of the node's symbol description, for example, because we are using a network with 793 total nodes are we are using the decimal counting system, therefore, we use three symbols describe every node and each of the String Character Units are divisible by three. Row 4 shows a sequence of node numbers which represent the sequence of node activations in one period of steady-state. We inserted commas in row 4 only to make the nodes involved in the sample period more readable, inserting extraneous symbols into the string finding algorithm increases the run time and may not have enough benefit.

Sample Start Index (String Character Units)	2496
Sample Stop Index (String Character Units)	8586
Sample Period Length (String Character Units)	81
Sample Period (Node Numbers)	786,788,789,785,793,791,792, 790,787,786,789,788,793,785, 791,792,790,787,786,789,788, 785,793,791,792,790,787

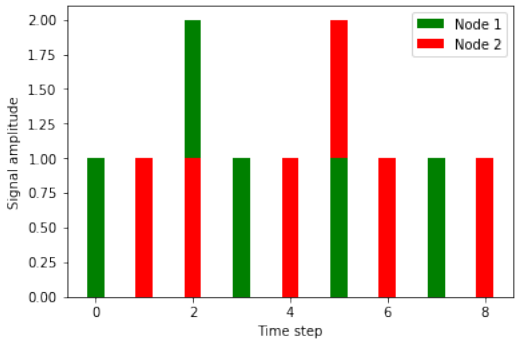

Figure 3.1: A pictorial representation of a set of ordered node activations which we will convert to a string of symbols. The activity of two nodes are presented. The y-axis represents the state of the nodes.

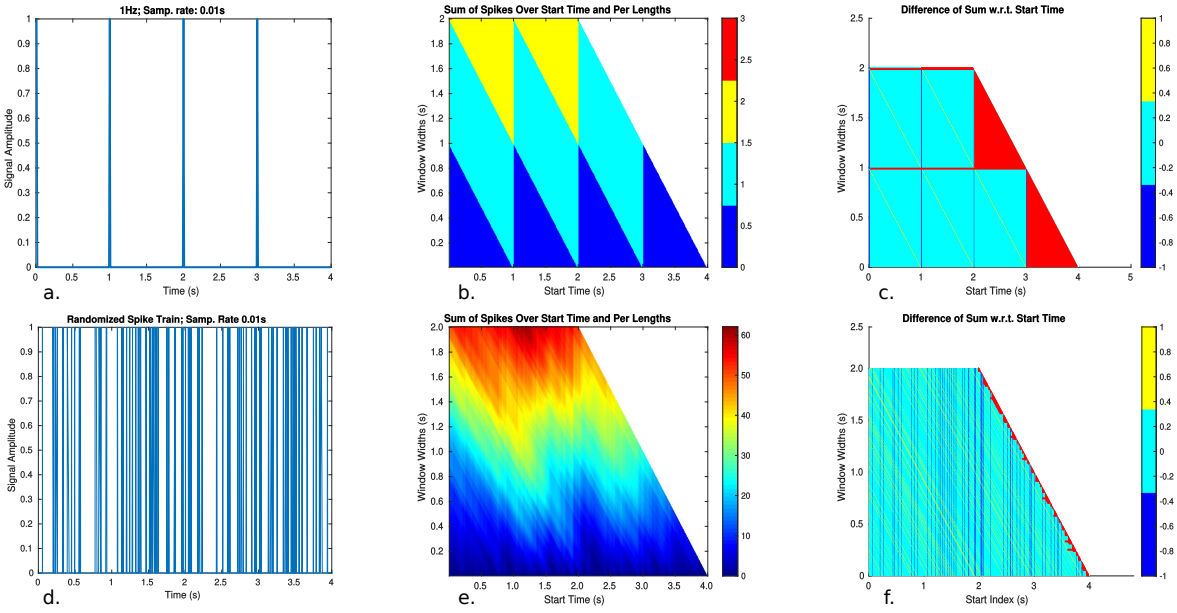

Figure 3.2: (a) An idealized $1Hz$ signal, sampled at $100Hz$, comprised of Dirac pulses to be analyzed by the sum method. The duration of the signal is $4s$. (b) The sum of pulses over the candidate starting points and candidate periods. The values of the heat map are defined over the set of possible starting points and periods, whereas, the white-space corresponds to the starting points and periods which can not be determined given the data. For example, we can not determine periods longer than half the data because we do not observe a repetition. (c) A plot of the difference of the sum of node activations with respect to starting position. The two leftmost horizontal red lines refer to the period and the harmonics of the original signal. The red horizontal lines indicate when the difference is zero from its leftmost end, the start time, to the end of the data. At the leftmost position of the red line is the start time of steady-state. The heat map indicates the values of the differences. (d) A randomized set of Dirac pulses sampled at $100Hz$. (e) The sum of node activations, the symmetries present in the previous examples are absent. (f) There is no dominant start time and harmonic.

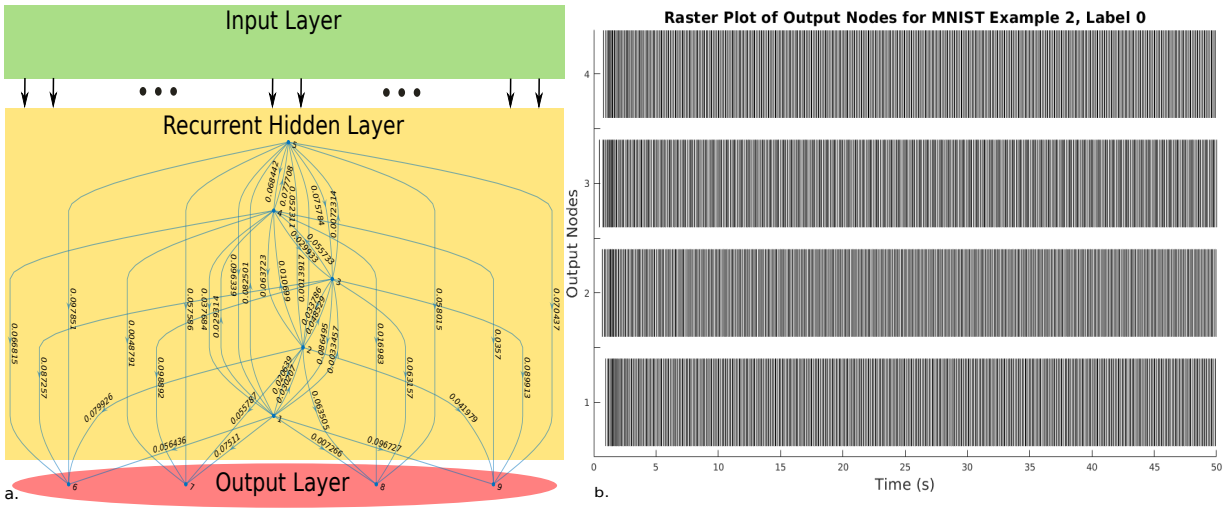

Figure 3.3: (a) Network Diagram. The Input layer consists of 784 nodes. The Recurrent/Hidden layer contains 5 nodes. The Output layer contains 4 nodes. The values of the Input Layer are set based upon the MNIST images. If the pixel contains a value, then the corresponding input node is activated at the initial condition of the network. Feed-forward connections from the Input layer to the Hidden layer, recurrent connections within the Hidden layer, and feed-forward connections from the Hidden layer to the Output layer. A sample of the edge delays are shown. The differentiating factor between the Hidden and Output layers are the directions of connections. (b) The resulting network activity of the nodes in the output layer resulting from stimulus of the input layer, the initial conditions. The network activity is shown as a raster plot. We forward calculate the network dynamics to 50s. We chose this length of time for two reasons. Firstly, the length was long enough to make sure that most initial conditions lead to steady-state activity for our purposes. Secondly, that we are able to observed multiple periods of steady-state activity to make visualization of the steady-state statistics easier.

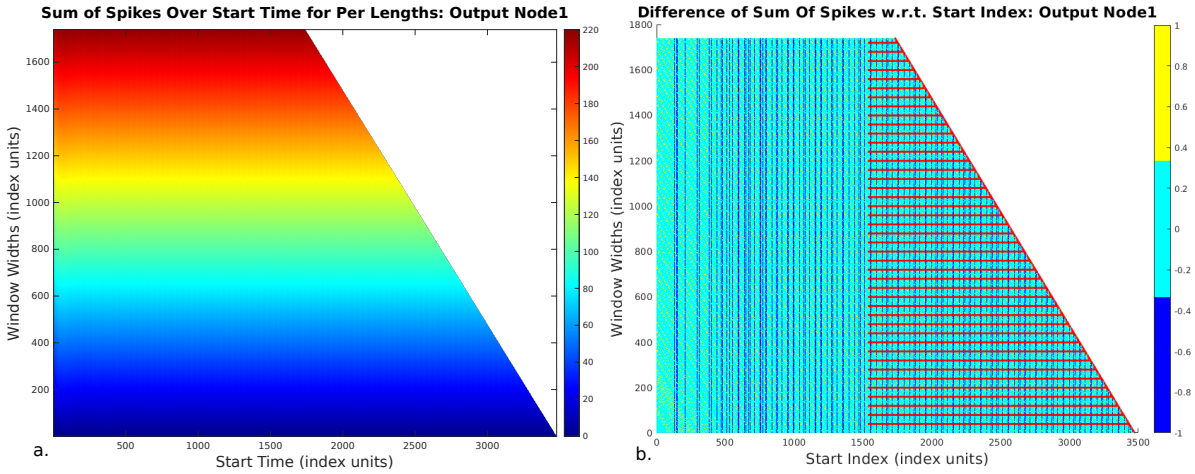

Figure 3.4: (a) The sum of node activations over candidate periods and starting indices for Output Node 1. Note, the change in units from Figure (3.2). Given our approach to the sum method in the non-uniform sampling setting, it no longer necessary to discuss units in terms of absolute time. In the non-uniform sampling setting the candidate periods and starting points are selected swept along the sampling points determined by the system's node activations. For example, a period of length y index units may refer to a range of absolute times. As such, the candidate steady-state statistics are determined based on indices rather than an absolute time measure. The system's steady-state statistics is determined from analysis of all nodes in the network, and selecting the latest starting point and longest period amongst all nodes. As the size of the period increases more node activations are captured. (b) The first finite difference in index units. The result of the difference is given by values of the heat map. The minimum starting point uniformly zero condition, as denoted by the horizontal red lines, is met around time index 1500. The fundamental period of network activity is given by the smallest height red horizontal line, with the minimum starting index. Each harmonic of the fundamental period is given by the incrementally higher horizontal red lines with the same starting indices. The red horizontal lines at the end of the data range of starting points correspond to consecutive candidate periods with the same number of node activations. In this situation, these candidates are easily filtered using the minimum starting point criteria. Light blue heat map values correspond to adjacent starting points whose difference in the number of node activations is 0.

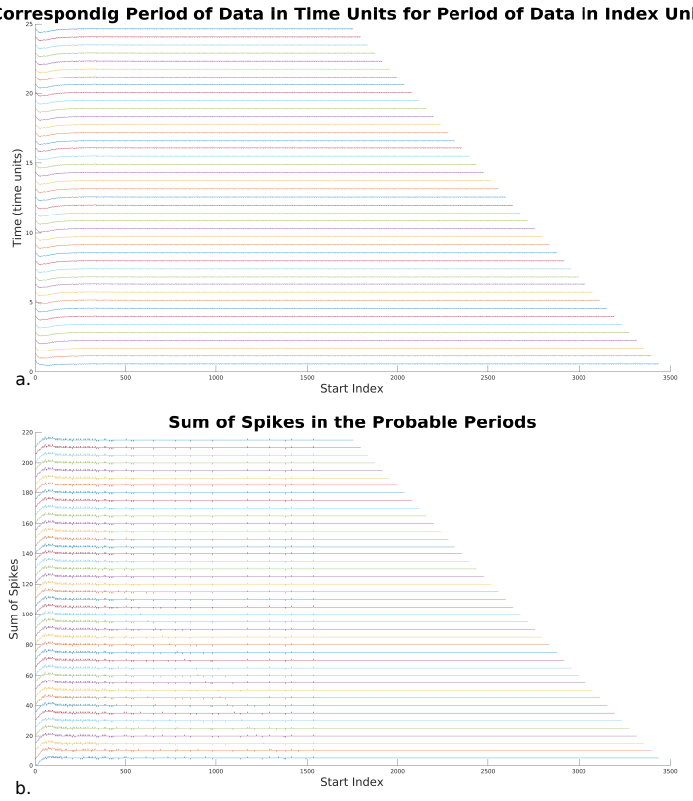

Figure 3.5: These plots are intended to show the correspondence between the time indices and absolute time. (a) The abscissa shows the candidate start indices. The ordinate shows the length of the period in time units of seconds. Each horizontal line corresponds to a previously determined harmonic, the horizontal red lines from Figure (3.4 b) The bottom most horizontal line corresponds to the fundamental period, and each higher one corresponds to the harmonics of the fundamental period. Fluctuations of each horizontal line correspond to the the length of the period in seconds when counting a fixed number of time indices, corresponding the number of time indices which make up a period, from the start index to the end of the period. Because of the non-uniform sampling the fluctuations remain. (b) Here we count the number of node activations in each period as a function of start index, and we can clearly see that after the inception of steady-state dynamics the number of node activations stabilizes to the steady-state value. The ordinate is a count of the number of node activations. The abscissa is the start index. And each horizontal line corresponds to a harmonic, the bottom-most corresponding to the fundamental period, each subsequent one a harmonic.

References

[1] GitHub - hdbn/runfinder: Automatically exported from code.google.com/p/runfinder.

[2] Danielle S. Bassett, Ankit N. Khambhati, and Scott T. Grafton. Emerging Frontiers of Neuroengineering: A Network Science of Brain Connectivity. *Annual Review of Biomedical Engineering*, 19(1):327–352, 2017.

[3] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D. U. Hwang. Complex networks: Structure and dynamics. *Physics Reports*, 424(4):175–308, February 2006.

[4] Stefano Boccaletti, Ginestra Bianconi, Regino Herrero, C.I.del Genio, Jesus Gómez-Gardeñes, Miguel Romance, Irene Sendiña-Nadal, Zhen Wang, and Massimiliano Zanin. The Structure and Dynamics of Multilayer Networks. *Physics Reports*, 544:1–122, November 2014.

[5] Connor Brennan and Alexander Proekt. A quantitative model of conserved macroscopic dynamics predicts future motor commands. *eLife*, 8:e46814, July 2019.

[6] Romain Brette, Michelle Rudolph, Ted Carnevale, Michael Hines, David Beeman, James M. Bower, Markus Diesmann, Abigail Morrison, Philip H. Goodman, Frederick C. Harris, Milind Zirpe, Thomas Natschläger, Dejan Pecevski, Bard Ermentrout, Mikael Djurfeldt, Anders Lansner, Olivier Rochel, Thierry Vieville, Eilif Muller, Andrew P. Davison, Sami El Boustani, and Alain Destexhe. Simulation of networks of spiking neurons: A review of tools and strategies. *Journal of Computational Neuroscience*, 23(3):349–398, December 2007.

[7] Marius Buibas and Gabriel A. Silva. A Framework for Simulating and Estimating the State and Functional Topology of Complex Dynamic Geometric Networks. *Neural Computation*, 23(1):183–214, October 2010.

[8] Ed Bullmore and Olaf Sporns. Complex brain networks: graph theoretical analysis of structural and functional systems. *Nature Reviews Neuroscience*, 10(3):186–198, March 2009.

[9] H. Cai, V. W. Zheng, and K. C. Chang. A Comprehensive Survey of Graph Embedding: Problems, Techniques, and Applications. *IEEE Transactions on Knowledge and Data Engineering*, 30(9):1616–1637, September 2018.

[10] Natalia Caporale and Yang Dan. Spike Timing–Dependent Plasticity: A Hebbian Learning Rule. *Annual Review of Neuroscience*, 31(1):25–46, July 2008.

[11] Bruno Cessac, Hélène Paugam-Moisy, and Thierry Viéville. Overview of facts and issues about neural coding by spikes. *Journal of Physiology-Paris*, 104(1-2):5–18, January 2010.

[12] Bruno Cessac and Thierry Viéville. On Dynamics of Integrate-and-Fire Neural Networks with Conductance Based Synapses. *Frontiers in Computational Neuroscience*, 2, July 2008.

[13] Gang Chen, Simon J. Puglisi, and W. F. Smyth. Fast and Practical Algorithms for Computing All the Runs in a String. In Bin Ma and Kaizhong Zhang, editors, *Combinatorial Pattern Matching*, Lecture Notes in Computer Science, pages 307–315. Springer Berlin Heidelberg, 2007.

[14] Luciano da F. Costa, Francisco A. Rodrigues, Gonzalo Travieso, and P. R. Villas Boas. Characterization of complex networks: A survey of measurements. *Advances in Physics*, 56(1):167–242, January 2007. arXiv: cond-mat/0505185.

[15] Luciano da Fontoura Costa, Osvaldo N. Oliveira Jr, Gonzalo Travieso, Francisco Aparecido Rodrigues, Paulino Ribeiro Villas Boas, Lucas Antiqueira, Matheus Palhares Viana, and Luis Enrique Correa Rocha. Analyzing and modeling real-world phenomena with complex networks: a survey of applications. *Advances in Physics*, 60(3):329–412, June 2011.

[16] M. Crochemore, L. Ilie, and W. F. Smyth. A Simple Algorithm for Computing the Lempel Ziv Factorization. In *Data Compression Conference (dcc 2008)*, pages 482–488, March 2008.

[17] M. Crochemore, C. S. Iliopoulos, M. Kubica, J. Radoszewski, W. Rytter, K. Stencel, and T. Waleń. New simple efficient algorithms computing powers and runs in strings. *Discrete Applied Mathematics*, 163:258–267, January 2014.

[18] Maxime Crochemore, Christophe Hancart, and Thierry Lecroq. Algorithms on Strings by Maxime Crochemore, April 2007.

[19] Carina Curto. What can topology tell us about the neural code? *Bulletin of the American Mathematical Society*, 54(1):63–78, September 2016.

[20] Guilherme Ferraz de Arruda, Francisco A. Rodrigues, and Yamir Moreno. Fundamentals of spreading processes in single and multilayer complex networks. *Physics Reports*, 756:1–59, October 2018.

[21] Manlio De Domenico, Clara Granell, Mason A. Porter, and Alex Arenas. The physics of spreading processes in multilayer networks. *Nature Physics*, 12(10):901–906, October 2016.

[22] N. Deo. Graph Theory with Applications to Engineering and Computer Science. *Networks*, 5(3):299–300, 1975.

[23] R. Diestel. *Graph Theory*. Electronic library of mathematics. Springer, 2006.

[24] Claire Donnat and Susan Holmes. Tracking network dynamics: A survey using graph distances. *The Annals of Applied Statistics*, 12(2):971–1012, June 2018.

[25] Frantisek Franek and Robert Fuller. A note on performance comparisons of various runs programs. page 11.

[26] Robert C G Fuller. Performance comparisons of various runs algorithms. page 60.

[27] Anna Goldenberg, Alice X. Zheng, Stephen E. Fienberg, and Edoardo M. Airoldi. A Survey of Statistical Network Models. *Found. Trends Mach. Learn.*, 2(2):129–233, February 2010.

[28] Thilo Gross and Bernd Blasius. Adaptive coevolutionary networks: a review. *Journal of The Royal Society Interface*, 5(20):259–271, March 2008.

[29] Dan Gusfield. Algorithms on Strings, Trees, and Sequences by Dan Gusfield, May 1997.

[30] Toru Kasai, Gunho Lee, Hiroki Arimura, Setsuo Arikawa, and Kunsoo Park. Linear-Time Longest-Common-Prefix Computation in Suffix Arrays and Its Applications. In G. Goos, J. Hartmanis, J. van Leeuwen, and Amihood Amir, editors, *Combinatorial Pattern Matching*, volume 2089, pages 181–192. Springer Berlin Heidelberg, Berlin, Heidelberg, 2001.

[31] W. O. Kermack and A. G. McKendrick. A Contribution to the Mathematical Theory of Epidemics. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 115(772):700–721, August 1927.

[32] M. Kivela, A. Arenas, M. Barthelemy, J. P. Gleeson, Y. Moreno, and M. A. Porter. Multilayer networks. *Journal of Complex Networks*, 2(3):203–271, September 2014.

[33] R. Kolpakov and G. Kucherov. Finding maximal repetitions in a word in linear time. In *40th Annual Symposium on Foundations of Computer Science (Cat. No.99CB37039)*, pages 596–604, New York City, NY, USA, 1999. IEEE Comput. Soc.

[34] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, May 2015.

[35] Yann LeCun and Corinna Cortes. MNIST handwritten digit database. 2010.

[36] Michael G. Main. Detecting leftmost maximal periodicities. *Discrete Applied Mathematics*, 25(1):145–153, October 1989.

[37] T. Makino. A Discrete-Event Neural Network Simulator for General Neuron Models. *Neural Computing & Applications*, 11(3-4):210–223, June 2003.

[38] U. Manber and G. Myers. Suffix Arrays: A New Method for On-Line String Searches. *SIAM Journal on Computing*, 22(5):935–948, October 1993.

[39] Othon Michail. An Introduction to Temporal Graphs: An Algorithmic Perspective. *arXiv:1503.00278 [cs]*, March 2015. arXiv: 1503.00278.

[40] Yuta Mori. A lightweight suffix-sorting library. Contribute to y-256/libdivsufsort development by creating an account on GitHub, August 2019. original-date: 2015-03-17T15:30:25Z.

[41] Katherine Morrison and Carina Curto. Chapter 8 - Predicting Neural Network Dynamics via Graphical Analysis. In Raina Robeva and Matthew Macauley, editors, *Algebraic and Combinatorial Computational Biology*, pages 241–277. Academic Press, January 2019.

[42] Emre Neftci, Srinjoy Das, Bruno Pedroni, Kenneth Kreutz-Delgado, and Gert Cauwenberghs. Event-driven contrastive divergence for spiking neuromorphic systems. *Frontiers in Neuroscience*, 7, 2014.

[43] M E J Newman. The spread of epidemic disease on networks. page 12.

[44] M. E. J. Newman. The structure and function of complex networks. *SIAM Review*, 45(2):167–256, January 2003. arXiv: cond-mat/0303516.

[45] Simon J Puglisi, William F Smyth, and Munina Yusufu. Fast Optimal Algorithms for Computing All the Repeats in a String. page 9, 2008.

[46] M. Salehi, R. Sharma, M. Marzolla, M. Magnani, P. Siyari, and D. Montesi. Spreading Processes in Multilayer Networks. *IEEE Transactions on Network Science and Engineering*, 2(2):65–83, April 2015.

[47] Nino Shervashidze. Weisfeiler-Lehman Graph Kernels. page 23.

[48] Yutaka Shimada, Yoshito Hirata, Tohru Ikeguchi, and Kazuyuki Aihara. Graph distance for complex networks. *Scientific Reports*, 6:34944, October 2016.

[49] Gabriel A. Silva. The Effect of Signaling Latencies and Node Refractory States on the Dynamics of Networks. *arXiv:1804.07609 [q-bio]*, August 2019. arXiv: 1804.07609.

[50] Amirhossein Tavanaei, Masoud Ghodrati, Saeed Reza Kheradpisheh, Timothee Masquelier, and Anthony S. Maida. Deep Learning in Spiking Neural Networks. *Neural Networks*, 111:47–63, March 2019. arXiv: 1804.08150.

[51] Bruno Umbria Pedroni. *Boltzmann Energetics and Temporal Dynamics of Learning Neuromorphic Systems*. Ph.D., University of California, San Diego, United States – California, 2019.

[52] Stanley Wasserman and Katherine Faust. *Social network analysis: Methods and applications*. Social network analysis: Methods and applications. Cambridge University Press, New York, NY, US, 1994.

[53] Duncan J. Watts. A simple model of global cascades on random networks. *Proceedings of the National Academy of Sciences*, 99(9):5766–5771, April 2002.

[54] B Yu Weisfeiler and A A Leman. A Reduction of a Graph to Canonical Form and the Algebra Which Appears Therein. *Nauchno-Technicheskaya Informatsiya*, page 11, 1968.

[55] Chia-Chun Weng. *Implementing Efficient Algorithms for Computing Runs*. thesis, October 2011.

[56] J. Ziv and A. Lempel. Compression of individual sequences via variable-rate coding. *IEEE Transactions on Information Theory*, 24(5):530–536, September 1978.

Chapter 3, in part, is currently being prepared for submission for publication of the material. Vivek K. George, Arkin Gupta and Gabriel A. Silva. The dissertation author was the primary investigator and author of this paper.

Chapter 4

Local Biological Neural Network Design Principle

SCIENTIFIC REPORTS

OPEN

An Optimized Structure-Function Design Principle Underlies Efficient Signaling Dynamics in Neurons

Francesca Puppo[1,3], **Vivek George**[1,3] **& Gabriel A. Silva**[1,2,3]

Dynamic signaling on branching axons is critical for rapid and efficient communication between neurons in the brain. Efficient signaling in axon arbors depends on a trade-off between the time it takes action potentials to reach synaptic terminals (temporal cost) and the amount of cellular material associated with the wiring path length of the neuron's morphology (material cost). However, where the balance between structural and dynamical considerations for achieving signaling efficiency is, and the design principle that neurons optimize to preserve this balance, is still elusive. In this work, we introduce a novel analysis that compares morphology and signaling dynamics in axonal networks to address this open problem. We show that in Basket cell neurons the design principle being optimized is the ratio between the refractory period of the membrane, and action potential latencies between the initial segment and the synaptic terminals. Our results suggest that the convoluted paths taken by axons reflect a design compensation by the neuron to slow down signaling latencies in order to optimize this ratio. Deviations in this ratio may result in a breakdown of signaling efficiency in the cell. These results pave the way to new approaches for investigating more complex neurophysiological phenomena that involve considerations of neuronal structure-function relationships.

The mechanisms underlying the successful integration and rapid transmission of information in the brain rely on interactions between structural and dynamical properties that begin at the level of the single neuron. The complexity of these interactions are reflected in the wide variability of axon arbor morphologies and dynamical states neurons can take on. A still unsolved fundamental question is what is the relationship between the morphological design principles of individual branching axons and their role in optimizing action potential signaling in the neuron?

Neuronal morphologies are the outcome of complex developmental processes including axon growth, stabilization of synaptic connections and axon pruning[1–3]. These processes are dependent on a multitude of local molecular and cellular mechanisms and conditions[4–6]. Despite the stochastic nature of morphological development, as well as other biological, physical, and molecular constraints, evolutionarily neurons have achieved a degree of common computational efficiency. A growing list of experimental and computational results, including those we present in this paper, suggest that these developmental processes in neurons satisfy a set of specific optimization principles[7–10].

Until recently, the prevailing dominant hypotheses has been that neurons are morphologically designed to optimize for the least amount of cellular material necessary. The logic was that the less amount of material that was used, the greater the conservation of energy and cellular resources. In particular, a number of studies have argued that wiring minimization principles that maximize the conservation of material underlie the morphological design of neurons and even the broader anatomical organization responsible for functional maps in the neocortex[7,10–13], such as the intracortical wiring underlying functional maps in mammalian visual cortex[10,14]. However, more recent work has shown that neurons are not minimized for wiring length but instead are designed somewhere in between the two extremes of minimizing wiring costs versus maximizing action potential conduction velocities. They use more material than minimal construction costs would allow in order to increase conduction velocities that decrease temporal costs, but at the same time they do not signal as fast at they could if the wiring design was optimized strictly for speed, thereby offsetting the material cost[9,15]. Budd *et al.* have recently

[1]Department of Bioengineering, University of California, San Diego, La Jolla, 92093, CA, USA. [2]Department of Neurosciences, University of California, San Diego, La Jolla, 92093, CA, USA. [3]Center for Engineered Natural Intelligence, University of California, San Diego, La Jolla, 92093, CA, USA. Correspondence and requests for materials should be addressed to G.A.S. (email: gsilva@ucsd.edu)

examined the functional consequences and temporal cost of wiring minimization principles. Their results suggest that there exists a trade-off between the time it takes action potentials to reach synaptic terminals (temporal cost) and the amount of cellular material associated with the wiring path length of the neuron's morphology (material cost)[9]. This can be interpreted as a need to balance the advantages to the organism of signaling and processing information as fast as possible, while keeping material and cellular costs within reason.

However, where this spatio-temporal trade-off lies and the design principles being optimized remain unknown. Identifying them is critical to ultimately understanding why neurons are designed the way they are, and the effect they may have on a neuron's ability to represent and process information in the brain. This would also allow many other seemingly unrelated neurobiological results that involve considerations of neuronal structure-function relationships to be understood within this new context. As a result, completely disconnected results, models, and interpretations of data would have an underlying 'constraint' that nature necessitates they conform to. This would have a significant impact on our understanding of the brain as a system.

The work by Budd and colleagues suggested that the spatio-temporal trade-off observed in axon arbors contributes to the maximization of temporal precision, or conversely, the minimization of temporal dispersion in neuronal circuits[9]. Here, we explored the functional role of axon morphologies in the propagation of action potentials. The result of our analysis suggests a new putative underlying optimization principle between temporal and material costs in axon arbors. We investigated the signaling dynamics of action potentials in individual axonal branches using a model that computes a metric associated with the efficiency of signaling in spatial geometric networks. We have recently shown that optimal efficient signaling between connected node pairs in geometric networks is bounded by a ratio that approaches unity, between the signaling latency on the edge and the internal dynamics of the individual nodes[16]. We call this ratio the refraction ratio. It reflects a necessary balance between the internal dynamics of the participating nodes that make up the network, and the dynamics of signaling or information flow on the network. We have previously shown in numerical simulation experiments that a deviation of this principle can result in the complete breakdown of signaling dynamics[17].

In this work, we propose the refraction ratio as a measure of signaling efficiency in axons, and we provide evidence that the design of axonal arborizations are constructed to optimize this ratio. We considered individual Basket cell neurons as geometric axon tree networks with nodes along axonal branching points. We then analyzed the signaling dynamics along individual axonal branches, where a branch is the entire axonal segment connecting the axon initial segment to one of the axon's terminals. Basket cell neurons optimize the ratio between the refractory period of the membrane and action potential conduction latencies. Given a signaling latency along the convoluted path resulting from the morphological geometry of an axon arbor, the ratio of the signaling latency to the membrane refractory period approaches a value of unity for each individual axonal branch. This reflects the theoretically predicted optimal ratio.

One interpretation of our results is that the convoluted paths taken by axons and axonal branches reflect a design compensation by the neuron to slow down signaling latencies in order to optimize the refraction ratio. Because the cell does not have direct control over the biophysics responsible for determining the length and changing character of the membrane refractory period, it uses axonal morphology (structure) to ensure that an optimized refraction ratio is almost always preserved by the time action potentials reach synaptic terminals. An optimized ratio reflects a balance between the temporal constraints associated with the biophysics of the membrane, and the amount of time it takes action potentials (the signals) to travel down the axonal branch.

Competitive Refractory Dynamics Model

We have recently described the construction and theoretical analysis of a framework derived from the canonical neurophysiological principles of spatial and temporal summation that models the competing dynamics of incident signals into nodes along directed edges in a network[16]. We considered how temporal latencies produce offsets in the timing of the summation of incoming discrete events, and the ultimate activation of downstream nodes. Our approach involved the development of a competitive refractory framework. The framework models how the timing of different signals compete to 'activate' nodes they connect into. We use the term activation here to imply an appropriate reaction of that node to the signal it has received. For example, in the case of biological neurons it would be the generation of an action potential that propagates down the axon and axonal arborizations. In an abstract notion of a directed network, it is the generation of a discrete signaling event by the activated node for an excitatory input (or lack of activation for an inhibitory input) and a subsequent period of refractoriness. The interplay between temporal latencies of propagating discrete signaling events on the network relative to the internal dynamics of the individual nodes can have profound effects on the dynamics of the network[17–20]. In a geometric network temporal latencies are due to the relationship between signaling speeds (conduction velocities) and the geometry of the edges on the network (i.e. edge path lengths). The model also assumes that each node has associated with it a refractory period or refractory state. The refractory period is a value that reflects the internal state of the node in response to its activation by an upstream winning node (or set of nodes). It is a period of time during which the node is not capable of responding to other input signals from nodes that connect into it. In other words, it is refractory to such inputs. We did not assume anything about the internal model that produces this refractory state, which could include an internal processing time during which the node is making a decision about how to react.

The refraction ratio. We refer the reader to[16] for a full treatment of the derivations and theoretical analysis. We considered the geometrical construction of a network in the following sense. We assumed that signals or discrete information events (e.g. action potentials in biological neurons) propagate between nodes along directed edges at a finite speed or conduction velocity, resulting in a temporal delay or latency of the signal arriving at the downstream node. Imposing the existence of signaling latencies implies a network that can be mapped to a geometric construction, where individual nodes could be assigned a spatial position in space in R^3. Directed

edges connecting node pairs are allowed to have a convoluted path integral, i.e. a Jordan arc. There is no restriction that edges have to be spatially minimizing straight line edges, geodesics. A signaling latency τ_{ij} defines the ratio between the distance traveled on the edge relative to the speed of the propagating signal. For any set of connected vertex pairs $v_i v_j$, $\tau_{ij} = d_{ij}/s_{ij}$. We then define a refractory period for the vertex v_j by R_j. This reflects the internal dynamics of v_j once a signaling event activates it. For example, the amount of time the internal dynamics of v_j requires to make a decision about an output in response to being activated, or some reset period during which it cannot respond to subsequent arriving input signals. We place no restrictions on the internal processes or time that contribute toward R_j. Once the winning node v_j 'activates', it will become refractory for a period of time R_j. We can then compute at discrete times in parallel for every vertex in the network, i.e. every v_j, which set of nodes that connects into v_j causally activates it. We are able to achieve this by keeping track of the temporal interplay of propagating signaling events on the edges relative to the refractory states of the individual vertices.

A key consideration of this computation was establishing a relationship between R_j and τ_{ij} for a vertex v_j. We defined the refraction ratio between the refractory period R_j and a signaling latency τ_{ij} associated with a discrete signaling event into v_j coming from a vertex v_i on the edge e_{ij} as

$$\Delta_{ij} = \frac{R_j}{\tau_{ij}} = \frac{R_j \cdot s_{ij}}{d_{ij}} \tag{1}$$

where $R_j > 0$. Our determination of which set of v_i will successfully activate v_j emerges from an analysis of this ratio for every vertex connected into v_j.

Under realistic conditions, there is likely to be a temporal offset between when each v_i signals and how far along v_j is in its recovery from its refractory period due to a previous signaling event. Furthermore, each contributing v_i is statistically independent from every other v_i, so that the amount of temporal offset for each v_i vertex signaling v_j will be different. In order to compute these offsets and keep track of the overall dynamics of the network, we index two different notions of time. We defined t_i to be the moment at which v_i initiates a signal along its edge e_{ij} towards v_j. And an observation time T_o, which is the moment at which we observe or measure the state of the network. We can then compute temporal offsets by slightly expanding how we define the refractory period and signaling latency. For the refractory period, we let ϕ_j represent a temporal offset from R_j, such that at T_o.

$$\overline{R}_j = R_j - \phi_j \text{ where } 0 \leq \phi_j \leq R_j \tag{2}$$

We call $\overline{R}_j$ the effective refractory period. It reflects the amount of time remaining in the recovery from R_j at the observation time T_o.

Similarly, the times at which each v_i vertex initiates a signaling event would not be expected to be all the same. At any given arbitrary observation time T_o a signal from any v_i may be traveling part way along e_{ij} at a speed s_{ij}, effectively shortening τ_{ij}. Or it may be delayed in signaling if v_i signals some time after T_o, effectively lengthening τ_{ij}. At T_o we need to take into account the degree of signaling progression for each v_i along its edge. To accomplish this, we extend how we consider a signaling latency in the following way. First, we retain τ_{ij} to represent the absolute temporal delay (latency period) for a signal that travels on the edge e_{ij} for a vertex v_i. Note that the initiation of the signaling event could come before, right at, or after the observation time T_o. We then considered a temporal offset for τ_{ij}, an effective shortening or lengthening of τ_{ij} relative to T_o as follows

$$\overline{\tau}_{ij} = \tau_{ij} + \delta_{ij} \text{ where, } \delta_{ij} \in \mathbb{R} \tag{3}$$

$\delta_{ij} > 0$ represents an effective delay or elongation beyond τ_{ij}. In other words, it represents the vertex v_i initiating a signal at some time after T_o. Values $-\tau_{ij} < \delta_{ij} < 0$ represent an effective shortening of τ_{ij}. This would be the case when v_i had initiated a signal that was traveling part way along the edge e_{ij} towards v_j prior to T_o. When $\delta_{ij} = 0$ it implies that v_i signals exactly at the moment the network is observed. And when $\delta_{ij} = -\tau_{ij}$ it implies that the signal arrives at v_j at the moment the network is observed. Values of $\delta_{ij} < \tau_{ij}$, which result in $\overline{\tau}_{ij} < 0$, represent a signal arriving at v_j *prior* to the observation time T_o.

In the general case then, we were then able to extend Equation 1 to reflect the effective refractory period and effective latency (relative to an observation time T_o) as

$$\Lambda_{ij} = \frac{\overline{R}_j}{\overline{\tau}_{ij}} \tag{4}$$

Efficient signaling using the refraction ratio. We then defined a notion of optimized efficient signaling in the context of the competitive refractory dynamic framework and the refraction ratio. Given an effective refractory period $\overline{R}_j$ and effective latency $\overline{\tau}_{ij}$ along an edge e_{ij}, the condition for the winning vertex v_i from the set of vertices that connect into v_j that achieves activation of v_j is dependent on the $\lim_{\overline{\tau}_{ij} \to \overline{R}_j^+}$ for $\Lambda_{ij} = \overline{R}_j / \overline{\tau}_{ij}$. In other words, the first discrete signal that arrives at v_j after it stops being refractory is the signal that results in its activation. Intuitively, efficient signaling in the context of the framework means that there should be a temporal match between the dynamics of signals traveling on edges relative to how quickly nodes can internally process signals. When a mismatch between these two considerations occurs, it can cause a break down in the signaling dynamics of the network. If signaling speeds s_{ij} are too fast, or equivalently, if the latencies $\overline{\tau}_{ij}$ are too short, compared to the refractory period of the node, the network will not be able to sustain recurrent activity[17]. If s_{ij} is too slow or the set of $\overline{\tau}_{ij}$ too long then the network will be inefficient in the sense that it has the potential for faster dynamic signaling

that is not being realized. Time, as a resource, is being wasted in such a network. In[16] we formalized these concepts by deriving upper and lower bounds for the refraction ratio between connected vertices v_i and v_j. We were able to show that, given a subgraph of the network as a set of nodes v_i with directed edges into a node v_j, the optimal refraction ratio $[\Lambda_{ij}]_{opt}$ is bounded by

$$[\Lambda_{ij}]_{opt} = \lim_{\tau_{ij} \to R_j^+} \Lambda_{ij} \text{ when } \phi_j \text{ and } \delta_{ij} = 0 \qquad \text{[Upperbound]} \tag{5a}$$

$$[\Lambda_{ij}]_{opt} \Rightarrow \lim_{\delta_{ij} \to -\varphi_j^+} \Lambda_{ij} \text{ when } \phi_j = R_j \qquad \text{[Lowerbound]} \tag{5b}$$

where τ_{ij} is the absolute signaling delay on the edge e_{ij} and R_j is the absolute refractory period for v_j. Given these bounds, an optimized refraction ratio is such that

$$\left[\Lambda_{ij}\right]_{opt} = \frac{\overline{R}_j}{\overline{\tau}_{ij}} \to 1. \tag{5c}$$

We refer the reader to[16] for the full derivation and proof.

Refraction ratio analysis in axon arbors. In this paper, we considered the morphological structure of individual Basket cell neurons as geometric networks, and carried out an analysis of the refraction ratio for individual axonal branches given experimentally determined morphologies and computed values of conduction velocities and refractory periods. Specifically, we considered a path consisting of the soma of a neuron and its axon and axonal arborizations. We first mapped the axon arbor network into a tree-graph consisting of a root vertex defined at the axon's initial segment, and terminal vertices at each synaptic terminal (see Methods section). We then evaluated the signaling efficiency at synaptic terminals by computing the refraction ratio at each terminal vertex as a function of the geometry (morphology) of the respective axonal branches. We use the term 'branch' to refer to the sequence of axonal segments connecting the root vertex, i.e. the first vertex at the soma, to a specific terminal vertex at one of the axon's synaptic terminals.

The refraction ratio is a point property, in the sense that it is computed at a specific node in the network. We computed the ratio for a known set of defined nodes corresponding to specific axonal segments along the length of a branch, culminating at the terminal synaptic node. Computing the ratio depends on knowledge or estimation of the conduction velocity of the action potential and the refractory period for the node under consideration. The refractory period or refractory state of individual nodes along specific points in the arbor graph reflect the refractory period of the membrane at the corresponding points along the axon (see the Methods section for full details regarding the estimation of the refractory period). The refraction ratio along the defined node points of the branching path that connects the root vertex to the synaptic terminal vertex is a property of that branch. It depends on the time of propagation of an action potential along that branch (which is a function of the morphology and conduction velocity). Within this context, the refraction ratio determines a gating effect at each node: in setting the rate at which each action potential can be transmitted along an axonal branch, and as a result when synaptic signaling occurs, it dictates the frequency at which each neuron fires.

Results

Graph-Based Model of Geometric Axon Arbor Networks. We carried out our analysis of the refraction ratio in geometric tree networks of morphologically reconstructed axonal arbors. In order to compare our results with those of Budd and colleagues[15], we choose to analyze Basket cell neurons. Specifically, we considered anatomical and morphological data from detailed three-dimensional (3D) reconstructions of complete single neuron axonal arbors of 57 basket cells from the rat neocortex[21–23]. This resulted in a data set consisting of 11,575 independent axonal branches. The morphology of one of the basket cells we analyzed is shown in Fig. 1A.

We used a graph-based modeling approach to map the geometric connectivity of the axon arbor structures. The vertices in the graph reflect sampling points in the original morphological reconstruction of the neurons, and are associated with a physical position in three dimensional space (see NeuroMorpho.Org[22,23]). Figure 1B shows a graphical representation of the axon tree network: the root vertex v_R corresponds to the initial vertex at the axon initial segment where the axon begins at the base of the cell body (soma); branching points along the axons were labeled as branching or bifurcation vertices v_B; synaptic terminals were defined as fixed end terminal vertices v_T. We use the term 'branch' to refer to the axonal segment connecting the root vertex, i.e. the first vertex at the soma, to a specific terminal vertex at one of the axon's synaptic terminals. Vertices in between these two end point vertices reflect axonal segments along the way, defined by the original morphological sampling frequency of the data in the NeuroMorpho database. We call these elementary vertices $v_{e,n}$. We equivalently use the term 'branch' or 'branching path' in some instances in the paper. Within a specific neuron, every branch shares a common subset of vertices, specifically, the vertices for the segments that make up the axon itself from the soma up to where the axonal arborization branches off into individual synaptic branches. A full branching path then, is composed of this common axonal subset plus the unique arborization branch that ends at its respective synaptic terminal. Mathematically, we summarize a specific branch as $p(v_R, v_{T_s})$. For example, in Fig. 1B, a branch is the sequence of axonal segments that connects v_R to the terminal vertex v_{T_1}. A second branch is represented by the sequence of axonal segments that connects v_R to the terminal vertex v_{T_2}. In other words, a branching path $p(v_R, v_{T_s})$ is the path through which an action potential that is generated at the soma propagates to reach a specific synaptic terminal.

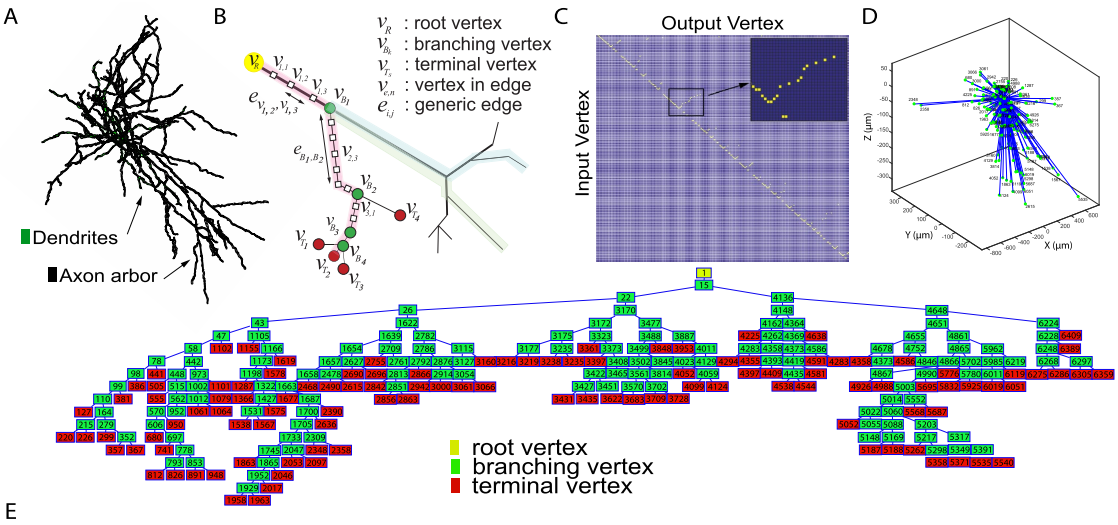

Figure 1. Deriving axonal network tree graphs from morphological reconstructions. (**A**) Representative three dimensional morphological reconstruction of one of the Basket cells in our data set from rat neocortex. Dendritic arbors are in green, axonal arbors in black [data from http://neuromorpho.org/[21–23]]. (**B**) Graph-based model of the geometric network of an axon and its arborizations. Vertices in the graph are labeled as follows: root vertex v_R is located at the initial segment of the axon at the soma; branching vertices v_B corresponding to branch points along the axons; terminal vertices v_T were synaptic terminals of the axon arborizations. Vertices between branch points reflect measurement points in the original reconstruction, and are denoted as $v_{e,n}$, with e identifying the edge and n the specific axonal point within that edge. The computation of the convoluted geometric distance between the root vertex v_R and a selected terminal vertex v_{T_s} was determined by the sum of all edge distances between $v_{e,n}$ pairs forming the total path that connected the two vertices. The total distance approximates well the edge path length integral (see the Methods) of the tortuous path connecting the axon initial segment to a synaptic terminal in a real neuron (see panel A). (**C**) Reduced adjacency matrix mapping the connectivity between branching points along the axon tree of the neuron visualized in A. (**D**) Three dimensional graph of the same axon arbor mapped onto a minimized length network where only root vertex and terminal vertices are considered. The units are microns, as indicated. Metric distances are Euclidean distances from the root vertex at the center of the figure (common point for each branch) to the axonal terminals. The actual terminals are the green points in the figure. The 3D positions of root and terminal vertices (X, Y, Z coordinate values) were obtained from the 3D morphological reconstruction in the NeuroMorpho database. Visually, Euclidean distances are represented by the blue lines connecting the root to the terminal vertices. (**E**) Biograph of the reduced network of the axon arbor in A, described by the adjacency matrix in C. Each labeled box in the graph stands for an identification number of the reported vertex. Root vertex in yellow, branching vertices in green, and terminal vertices in red.

Figure 1C shows the adjacency matrix encoding all the edges for the network graph representation of the neuron. A pair of vertices (v_i, v_j) in the graph are connected by a directed geometric edge e_{ij} that reflects the morphological geometry of the axonal segment linking both vertices. Each edge is associated with a length parameter associated with the convoluted length of the axonal segment in the arbor, computed as the path integral of the corresponding curve. Additional details about network construction, as well as the description of the edge path length integral calculation can be found in the Methods.

Figure 1D plots the three dimensional graph of the axon network. For clarity, this particular plot only shows the root vertex v_R and synaptic terminals v_T. The plot shows the length minimized network, where the distance from root vertex to terminals is the shortest straight line path calculated as the Euclidean distance. The figure highlights the difference between the geodesic (shortest) path distance in the minimized network, relative to the actual morphological paths of the axon arbor (Fig. 1A). Panel E in Fig. 1 shows the biograph of the same axonal network. We observed similar single neuron reconstructions for all the cells in our data set.

Signaling Dynamics in Axon Arbor Networks. We characterized the signaling dynamics on individual axonal arbors by estimating the conduction velocities (signaling speeds) of action potentials, and computing the propagation delays (signaling latencies) for each branch. The axonal branches were defined by fixing the principal vertex v_R and each terminal vertex v_{T_s} in the axon graph for all branching paths $p(v_R,v_{T_s})$ (the subscript s indexes specific vertices - see Methods). Axon conduction velocity is a function of the biophysical and morphological properties of the membrane and axon; it varies with axon thickness, branching, ion channel density and variety, and myelination[24]. We assumed that the neurons we analyzed here were unmyelinated, inferred from their small average diameter (mean 0.63 μm) from direct measurements[25] and theoretical studies[26,27] (see the Methods for a detailed justification and calculations). We then derived the conduction velocity s_{ij} between any two vertices v_i

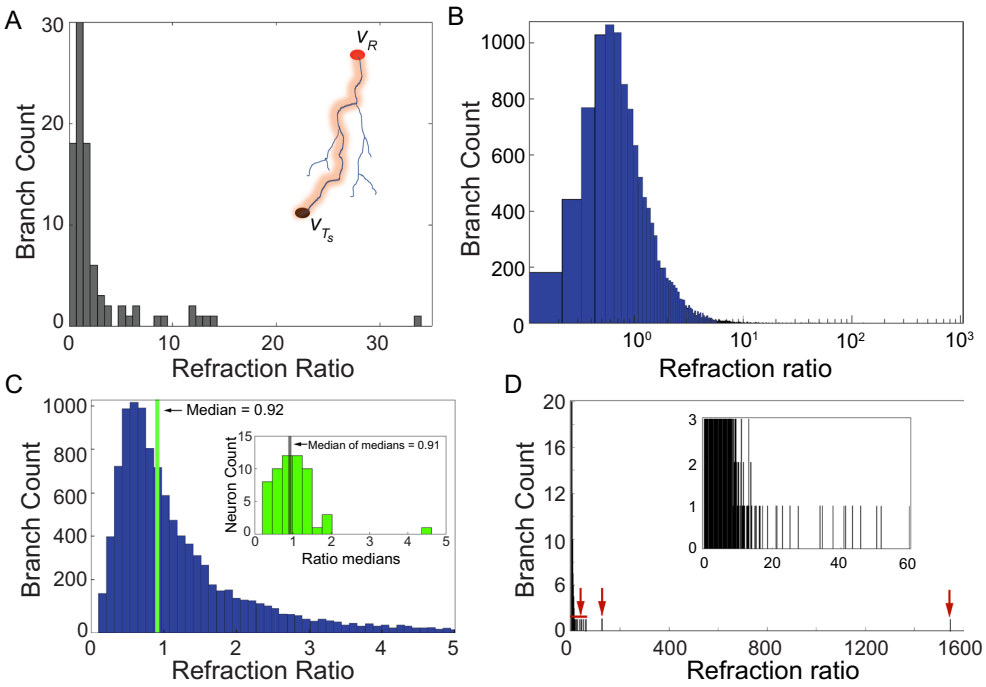

Figure 2. Basket cell neurons display optimized refraction ratios. (**A**) Distribution of the refraction ratio for all branches forming the axon arbor for a single neuron, i.e., for all paths $p(v_R, v_T)$ connecting the root vertex at the initial segment v_R to the synaptic terminals v_T. The inset shows an example of one axonal branch path from this cell. The refraction ratio was computed independently for each axonal branch for all branches of the axon tree. The *x*-axes represents values of the computed refraction ratio, while the *y*-axes shows the number or count of individual axonal branches. (**B**) Distribution of the refraction ratio for the full data set of 57 basket cells. The refraction ratio axis is on a logarithmic scale. All axonal branches (11,575) from all the neurons are statistically independent variables, and the value of the refraction ratio corresponding to each of them was used to build the distribution. (**C**) The peak of the distribution shown in B occurred at 0.56, while the median had a value of 0.92. The median value of all medians computed for each of the 57 neurons in the data set had a value of 0.91 (see the discussion in the main text). Inset: histogram of the median values of the refraction ratio distribution computed for each of the 57 neurons. The computed mean of the median ratio distribution had a value of 1.01 with 49 neurons (86 % of the entire data set) within one standard deviation, 1.01 ± 0.61 ($\mu \pm 1\sigma$). (**D**) Distribution of the refraction ratio for the entire data set (same data shown in panel B) but with *x*-axis extending out to the full range of values in order to show the few outliers (indicated by the red arrows). Inset: zoomed in view of the main plot out to a refraction ratio of 60 which captures the range over which most of the outliers were found. Note how the branch count steeply decreases from the peak. The distribution of refraction ratio values, the number and width of each bin, was obtained by dividing the full range of the computed refraction ratio by the smallest value of the ratio in order to obtain reasonable bin sizes (see Methods).

and v_j as a function of the axonal diameter D according to the relationship in[28]. We estimated an average conduction velocity for each individual axonal branch separately as described in detail in the Methods.

We obtained an average conduction velocity of 0.39 ± 0.16 m/s (Fig. 2 in Methods). Current estimates of mean intracortical axonal conduction velocity in adult cat visual cortex vary in the range 0.1–0.6 m/s[29–31] with similar values reported for horizontal axonal connections in rat neocortex (0.3–0.5 m/s)[25,32,33]. It is important to note that the accuracy of axonal diameter values are dependent on the resolution of the imaging techniques used in[21]. Fractions of microns are not easily or reliably measurable by optical microscopy. This uncertainty on the variability of the diameter data was accounted for in our analysis by computing average conduction velocities.

To estimate the signaling latency τ_{ij} on an edge e_{ij}, we divided the path length $|e_{ij}|$ between vertices v_i to v_j by a conduction velocity s_{ij}. The labels v_i and v_j indicate any vertex pair with v_i connecting into v_j regardless of the type of vertex (i.e. v_R, v_T or $v_{e,n}$). We then computed the total propagation delay along an axonal branching path $p(v_R, v_{T_s})$ as the sum of all temporal delays for each individually computed segment (see Methods).

Refractory Dynamics in Basket Cell Axon Arborizations. We next independently computed the refractory period R_j at the positions on the membrane corresponding to vertices v_j for every axonal segment in the tree network of all 57 neurons. Accurate experimental measurements of sub-type specific axonal membrane refractory periods are difficult to find in the literature. In general, refractory periods are defined as a range of values for whole neurons and almost always without regard for differences along different axonal segments.

Textbook values of the absolute refractory period are typically reported to range between 0.8–1 ms[34]. The relative refractory period typically lasts up to 5 ms. A few studies have investigated the details of the refractory period in the context of specific morphological and biophysical measurements[35–38]. This has allowed other investigators to study the role of axonal refractoriness on the signaling dynamics and information processing along individual axon branches, in response to extracellular activation[39,40]. However, although it is known that membrane refractoriness can change along the length of the axon and its axonal branches, there is limited empirical data available[41]. Recent work by Hu and colleagues on the biophysical and dynamical properties of Basket cells, the same class of neurons we investigated here, allowed us to estimate changes in the refractory period along axon arbors[25]. These authors showed that Basket cells display a supercritical density of sodium ion (Na^+) channels with a specific density profile along the axonal tree, responsible for high reliability fast propagation in nonmyelinated small axons. Additionally, analysis of Na^+ channel gating kinetics and voltage-dependence of activation and inactivation revealed that the inactivation time constant was 2.4-fold faster for axonal versus somatic Na^+ channels. Based on these observations, we estimated individual values of the refractory period R_j at axonal branching points as a function of the distance from the soma. We computed variations in the membrane refractory period using a derived function inferred from actual measurements of channel density variations in these neurons[25] (see Supporting Information).

Basket Cells Axon Branches Approach Near Optimal Values of the Refraction Ratio. The refraction ratio is defined as the numerical ratio between the refractory state R_j of a vertex v_j, and the temporal propagation delay (signaling latency) τ_{ij} of a signal traveling from a vertex v_i that connects into v_j[16]. We calculated the refraction ratio for each of the 11,575 axonal branch paths $p(v_R, v_T)$ from root vertex v_R to terminal vertices v_T independently for the entire data set. To compute the ratio, we used the path length, and calculated τ_{ij} and R_j values. Figure 2A, shows the distribution of the ratio for all branches of the axon arbor of one representative Basket cell from our data set. There is a clear clustering of the ratio near unity for the axonal branches of this neuron. The histogram for the distribution of refraction ratios across the entire data set for all the axonal branches is shown in Fig. 2B. There is a narrow peak in the distribution centered near unity. Figure 2C shows a highly magnified view highlighting the peak of the histogram in panel B. Approximately 76% (8, 795) of the total 11,575 branches had computed refraction ratios within the range of 0.25–1.75, with a median value of 0.92 for the entire distribution.

The value of the refraction ratio for the median of the medians across all 57 neurons had a computed value of 0.91 (inset in Fig. 2C). To compute the median of the medians we took the median value of the refraction ratio for each of the 57 basket cells computed independently from their respective distributions as shown for the representative neuron in panel A, and then obtained the median value of all the individual medians across the entire population. Our rationale for computing the median of the medians was that it provides a normalization against the varying total number of terminal vertices (synapses) across our sample population of 57 neurons. The median of the entire distribution pools all the ratios equally for all computed axonal branches across the whole population, but it does not take into account that different neurons contribute different numbers of ratios to the distribution because they have more (or less) number of axonal branches. This means that neurons that have more branches will have a greater effect on the value of the median of the population, and if they happen to have a disproportionately greater number of outlier ratio values these outliers will skew the value of the median. By computing the median of each cell independently, and then identifying the median of the population of median values, each of the 57 neurons have equal weight or effect on the median of the medians, regardless of the total number of terminal vertices they may have. It effectively filters out or normalizes a disproportionate effect contributed by outlier ratio values. Interestingly, the correspondence of the median of the entire population versus the median of the medians was almost identical (0.92 versus 0.91), adding further weight to the validity of the results.

In addition to the median values being near the theoretically predicted optimal, the distribution of the population of the medians, the degree of clustering near the peak, is equally significant. Because each median was computed independently for each neuron, the population of median values can be treated as its own data set. Of the 57 neurons, 49 of them (86 %) had ratios falling within one standard deviation, within the range 1.01 ± 0.61 ($\mu \pm 1\sigma$). Nearly the entire sampled population, 56 out of 57 neurons, had median values of their ratios falling within two standard deviations, in the range 1.01 ± 1.22 ($\mu \pm 2\sigma$) for the positive values of the ratio (since negative values have no meaning).

Importantly however, we note that some axonal branches did deviate significantly from an optimized refraction ratio, in one case extending out to an extreme value of 1600; see the outliers in Fig. 2B and D. This lends support for the sensitivity of the refraction ratio as a metric of signaling efficiency in neurons. We discuss the possible functional implications of such outliers in the Discussion below.

Morphological Regulation of Signaling Latencies. In order to investigate how the different parameters that define the refraction ratio affected the computed values for our data, we looked more closely at what parameters were having the greatest effect on the axonal branches of one representative neuron. Figure 3A shows a stylized axon arbor and schematically exemplifies observed signaling pathways in our neuron graph model. An action potential is generated at the axon initial segment (v_R, yellow circle) and propagates to individual synaptic vertices (terminal vertices v_T, black circles) along the respective branching paths $p(v_R, v_T)$ of the arbor. Two example branching paths are illustrated in the figure (blue and orange lines), which share the first two bifurcation points (red circles) and corresponding edges. Figure 3B shows the refraction ratios for all the branching paths $p(v_R, v_{T_s})$ of an individual cell ordered by decreasing edge path length (Methods). Similar to the results above, the plot shows how a large subset of the terminal vertices in this neuron fall within a range of 1 ± 0.5 (red horizontal bar in Fig. 3B). The value of the ratio is sensitive to and inversely related to the path length from the root vertex at the soma to the terminal vertices (Fig. 3C). Figure 3D shows the signaling latencies from the root vertex to the terminal vertices for all paths in the arbor. Signaling latencies are sensitive to $|e_{ij}|$, which are determined by the

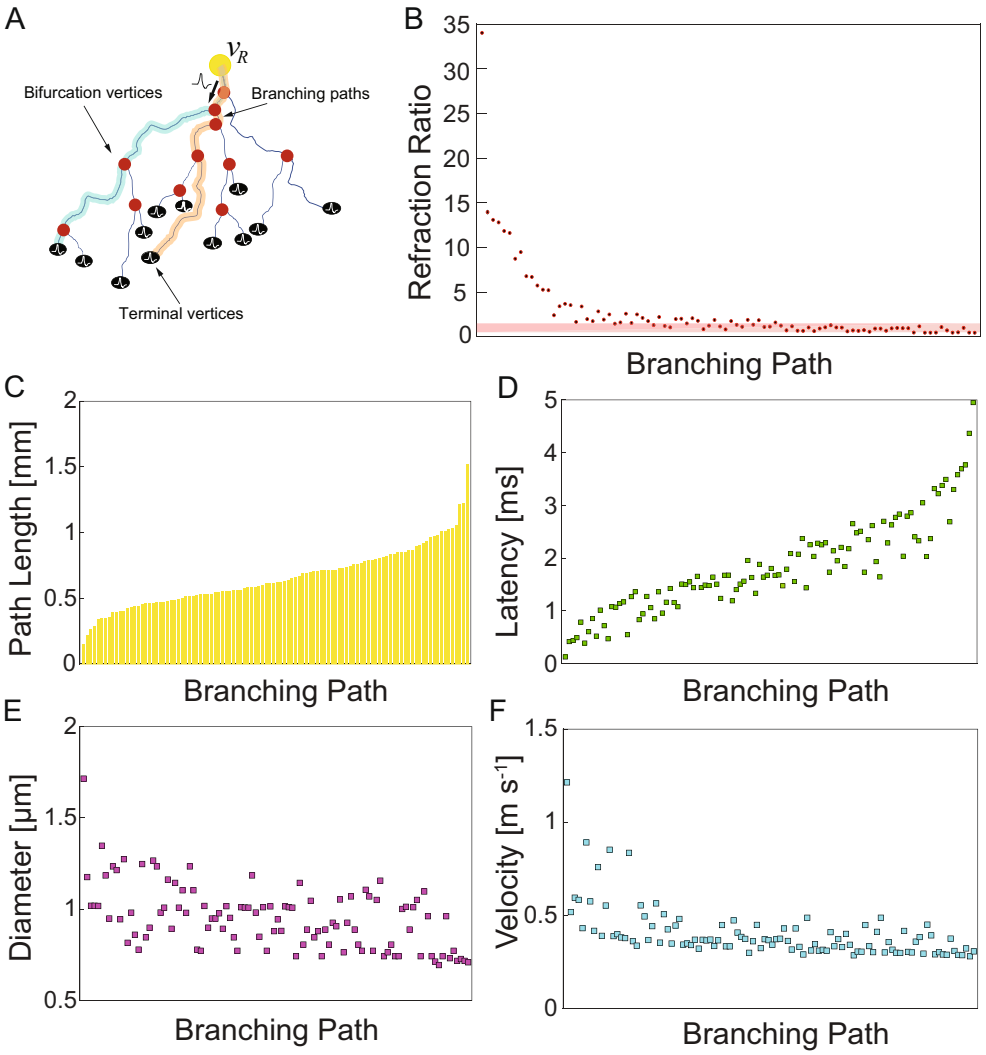

Figure 3. Effect of axonal morphological branch path lengths and conduction velocities on computed refraction ratios. (**A**) Stylized axon arbor of a neuron. Yellow circle: root vertex at the axon initial segment; red circles: bifurcation vertices; black circles: terminal vertices at the axon's synaptic terminals. A branching path $p(v_R, v_{T_s})$ is defined as the axon segment connecting the root vertex v_R to a specific terminal vertex v_{T_s}. The two example branching paths in the figure (blue and orange lines) share the first two bifurcation points and corresponding edges. An action potential is generated at the axon initial segment and propagates from the soma to each synaptic vertices throughout the individual branching paths of the arbor. (**B**) Refraction ratios computed for all branches of the axonal tree of one representative basket cell. Each circle represents one individual path $p(v_R, v_{T_s})$. Data points within the red band are ratios within a range of 0.5 to 1.5, indicating branches that fell near an optimal value of unity. (**C**) Path lengths associated with the axon morphologies responsible for the convoluted edges connecting the root vertex at the initial segment (v_R) to the synaptic terminals (v_T). Each bar represents a different synaptic terminal (branch) of the axon tree. (**D**) Propagation delays (latencies) for the same cell. (**E**) Mean axonal diameter for each branch. The mean diameter for a given branch was calculated as the mean for all measured diameter points for each branch. (**F**) Signaling speeds (conduction velocities) for each branch in this neuron. Although branch path lengths varied significantly, conduction velocities for most branches were within a range of 0.3–0.5 m/s. The data for panels A-D on the *x*-axes were ordered by path length from shortest to longest.

axon's morphology. These results are in agreement with numerical simulations of intracortical axon arbors which suggest that the path length is the main determinant of signaling latency[15]. Axonal branches longer than ~500 μm in this neuron were almost always capable of achieving an optimal ratio despite significant differences in morphological path lengths and signal latencies. We observed similar results in other individual neurons we looked at.

Figure 3E shows the relationship between the mean diameter measured as a function of position along individual branches of the axonal arbor, while Fig. 3F shows the scatter plot of the conduction velocity from the soma

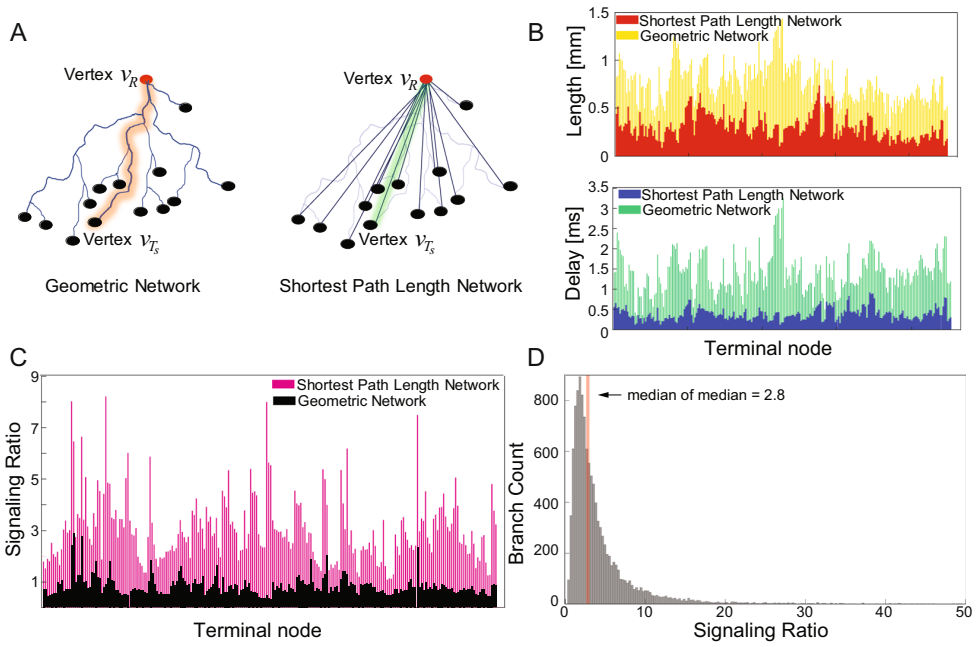

Figure 4. Effect of axonal branch path length on the refraction ratio. (**A**) As a control, we re-computed the refraction ratio for the entire data set with axonal branch path lengths that reflected the shortest Euclidean distance between the root vertex and terminal synaptic vertices. Left: The geometric tree network for one representative neuron where each edge connecting the root vertex to the terminal synaptic vertices were associated with the geometric convoluted path length (see the Methods). Right: the path length minimized tree network for the same neuron, where the length of the edges connecting the vertices were defined by the shortest Euclidean distance. (**B**) (Top) Path length computation in the geometric network (yellow bars) superimposed on the path length minimized network (red bars) for one of the 57 axon arbors. Each bar in the diagram identifies a different axonal branch. (Bottom) Signaling latencies derived for the geometric network (green bars) and the minimized network (blue bars) for the same axon tree. (**C**) Computed refraction ratios for the geometric network versus the minimized network for the same cell as in (**B** and **C**). The height of each of the black bars stands for the refraction ratio calculated as a function of the convoluted geometric path of an individual branch of the axon. (**D**) Histogram showing the global distribution of refraction ratios computed for all 11,575 branches of 57 neurons in our data set when mapped into a length-minimized network. The median value of all the medians computed for length minimized networks for all the neurons was centered at 2.8, indicating a deviation from optimality of the refraction ratio as a result of the perturbed path lengths associated with ignoring the geometry of the axonal morphologies.

to its synaptic terminal along individual axonal branches. In contrast to the effect of the morphology (path length) there is comparatively little variability in the diameter, even though it is a key contributor to the conduction velocity. This is reflected in the computed constant conduction velocity across all branches of the neuron. We also showed that the convergence towards an optimal ratio occurs quickly, within a few branching orders from the soma (see the Supplementary Information).

Perturbations of Morphological Path Lengths Resulted in Deviations of Optimal Ratios. We then investigated the effect that changing the axonal path lengths has on the ability of the neuron to compensate for longer latencies in order to preserve a near optimized refraction ratio at synaptic terminals. We compared ratios for the actual geometric morphology of a representative neuron (Fig. 4A, left) to a length-minimized version of the same cell (Fig. 4A, right) constructed by measuring the Euclidean (shortest) distance between the root vertex at the soma and terminal vertices at the synaptic terminals (cf. Fig. 1D). Figure 4B shows the distribution of lengths (distance of terminal vertex from the root vertex) for all branches in the tree network for this neuron and the distribution of signaling latencies, respectively. Figure 4C shows the refraction ratios for all paths between the soma and synaptic terminals of the same Basket cell calculated for the two different configurations of the network. As would be expected, for the length minimized network all branches have signaling ratios that deviate significantly from optimality. Figure 4D shows the refraction ratio for the entire 57 neurons and all individual axonal branches computed for length minimized tree networks. The population median of the medians of the ratios increased from 0.92 to 2.8. Mechanistically, this is due to poor temporal matching between signaling latencies and refractory recovery times at the synaptic terminals. The path length minimized networks still preserve the relative order of the axonal branches organized by lengths, in the sense that longer branches relative to shorter branches in the morphologically accurate geometric networks will still be longer in the shortest path networks. But the entire distribution of the ratios is skewed.

Discussion

In this work we investigated the relationship between axonal morphology, membrane refractory period, and action potential conduction velocities through an analysis of computed refraction ratios. This analysis allowed us to understand how the interplay between structure and dynamics achieves signaling efficiency in Basket cell neurons. Presumably, either directly or as a consequence of other processes, neurons are optimizing the refraction ratio as a design target.

Of the total 11,575 axon branches we computed refraction ratios for, 76% had a ratio within a range 0.25–1.75. 86% of the neurons had median ratio values within one standard deviation of the median of the population (0.91); a value that approaches the theoretically predicted maximal efficiency. This suggests that while an optimal refraction ratio at synaptic terminals is a local property associated with individual axonal arborizations, it is highly conserved across the entire population of sampled neurons.

In some cases, we also showed that some axonal branches did deviate significantly from an optimized refraction ratio (Fig. 2D). Some amount of variability is to be expected from any analysis of empirically measured experimental data. However, we note that the number of outliers was very small. An interesting question is whether these reflect 'noise' in the data set, essentially true (random) outlier events, or if they instead perhaps exhibit such extreme refraction ratios for a physiological reason. For example, one possibility is that these outliers reflect axonal branches that are designed to filter specific signals at a subset of synaptic terminals. The presence of such outliers also serve the purpose of validating the sensitivity of the refraction ratio in capturing a mismatch between refractory states and signaling latencies within physiologically plausible bounds.

The analysis also suggests that morphological regulation of signaling latencies exerts the greatest control on the refraction ratio. The data in Fig. 3 indicates that the shortest branches are not able to preserve an optimal ratio. This is because the latencies are too short and not able to match the refractory period at such short distances from the soma. One interpretation is that these branches are outliers relative to the rest of the dataset. This is statistically reasonable if we consider the variability of the experimental parameters. Incomplete development of axon arbors or incomplete pruning associated with the young age of the rat from which our data were derived (10–15 days postnatal) could possibly account for the discrepancies in the refraction ratio values. However, axons can also make synaptic contacts at short distances for physiologically functional purposes. Taking this into consideration, these outliers could be possibly serving a physiological role. One possibility is that branches that display non-optimality of the refraction ratio could be playing a different function in the dynamics of the cell, such as the propagation and transmission of signals at lower or higher frequencies. Such a possibility for a functional role of non-optimal ratios is further supported by the fact that seemingly outlier refraction ratios showed similar trends in several neurons. In contrast to the effect of the morphology, there was less variability in the average diameter computed for different branches, even though it is a key contributor to conduction velocity. This is reflected in the computed constant conduction velocities across all branches of the neuron. Conduction velocities and membrane refractory periods are the direct result of membrane biophysics; more specifically, the permeability and channel kinetics of sodium and potassium ion channel populations and the inactivation kinetics of sodium channels, respectively. These are biophysical properties that a neuron does not have control over in a sense. They represent generic cellular and molecular properties that the neuron cannot change. An intriguing interpretation of our results is that neurons may take advantage of the morphological geometry of individual axonal branches in order to manipulate signaling latencies to match the refractory periods at the terminal vertices (synapses). This would represent a mechanism or 'design criteria' to optimize signaling transmission in the cell.

We also tested the hypothesis that Basket cell axon arbors are not minimized for either wiring length or conduction velocity latencies[15]. By wire minimizing the axonal morphology, we were able to show that axonal path length is used to compensate for longer latencies in order to preserve a near optimized refraction ratio at synaptic terminals (Fig. 4). More significant deviations in the refraction ratios will occur when additional effects to axon path lengths and conduction velocities are considered, such as a reduction of the arbor diameter or randomly generated arbor graphs, as some authors have modeled[13,42–44]. In fact, shortest distance wiring path lengths like we computed here reflect the most conservative changes possible, and are therefore the least disruptive to the computation of the refraction ratios. This makes the surprising precision of the refraction ratio results near optimality for the morphologically accurate data even more significant.

Subsequent work should explore if an optimized refraction ratio holds for other types of neurons. We are also beginning to investigate experimentally if this principle holds in individual neurons where morphological reconstructions and detailed empirically measured values of conduction velocities and refractory periods across different segments are obtained in the same cell. Recent advancements in the development of novel electrophysiology techniques able to provide signal detection with high spatio-temporal resolution[45,46] will greatly benifit this type of analysis by reducing the number of assumptions, such as computing rather than measuring refractory periods and conduction velocities that contribute to signaling latencies. Finally, beyond establishing the refraction ratio as the design principle being optimized by neurons are questions about the physiological purpose of preserving an optimized ratio. Presumably neurons are exerting this level of dynamic precision for important neurophysiological reasons that require further investigation. One example could be temporal precision at synaptic terminals and its consequences on signaling integration. This has been previously suggested as one reason why neurons are neither optimized for wiring length or signaling speeds[15].

Methods

Axon Database. We used anatomical and morphological data from detailed three-dimensional (3D) reconstructions of complete axonal arbors of Basket cells from the rat neocortex[21]. Data were obtained from the online digital database NeuroMorpho.Org[23]. The neurons in our dataset were chosen based on the availability of high accuracy measurements of axon thickness or diameter throughout the whole arbor structure. More specifically, the data in the NeuroMorph database is comprised of spatial reconstructions and measurements along the axonal

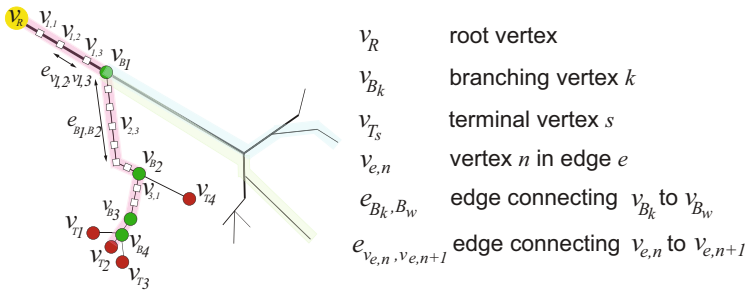

Figure 5. Graph-based model of the geometric network of an axon arbor.

arbor, with an identification number for each sample point, a three dimensional location (x_i, y_i, z_i), and the diameter D_i of the axon at that point. The database also contains indices indicating the physical connectivity between sample points along the arbors, thus providing all the relationships to construct the axonal tree. Different neurons have differing degrees with respect to the completeness of this information. Because of our analysis necessitated high detailed reconstructions we randomly selected Basket cell neurons that had the most complete information available.

Axon Graph Description. *Graph-model of axon arbors.* We used a graph theoretic approach to describe the three dimensional connectivity of single axons. A schematic description of the graph model is shown in Fig. 5. An axon arbor is naturally modeled as a tree-graph. We used the term axon 'branch' or 'branching path' to refer to the axonal segment connecting the axon initial segment to one of the axon's synaptic terminals. In a graph-based model, a 'branch' is composed of multiple axonal segments stringed together by a set of edge connected vertices. For the purposes of our analysis, we needed to distinguish between three classes of vertices and their corresponding edges. For each individual neuron in our dataset, vertices and edges were defined based on specific axonal landmarks. These were the initial axonal segment, axonal branching points or bifurcations, and synaptic terminals. These features mapped to a root vertex v_R, branching or bifurcation vertices v_{B_k} ($k = 1, \ldots, Q - 1$ with $Q - 1$ total number of branching vertices in the selected branch), and terminal vertices v_{T_s} ($s = 1, \ldots, W$, with W total number of synaptic terminals, or branches, in a given neuron), respectively. The vertex pair consisting of the root vertex v_R and a synaptic terminal vertex v_{T_s} connected by the edge $e_{R,V_{T_s}}$ defined a branching path. We equivalently used the label $p(v_R, v_{T_s})$ to indicate a specific branching path. Successive bifurcation vertices, *i.e* one branching point and the next branching point in an axonal arborization, are given by $v_{B_k(i)}$ and $v_{B_{k(i+1)}}$, along with their edges $e_{B_k i, B_k i+1}$, the segment of the axon that connects two bifurcation points. Vertices in between the root and terminal vertex reflect axonal segments along the way, defined by the original morphological sampling frequency of the data in the NeuroMorpho database. We called these elementary vertices. Elementary vertices correspond to neither a branching vertex, a terminal vertex, nor the root vertex. The elementary vertices located in between bifurcation vertices of the same axonal branch are labeled $v_{e,n}$ where $e = 1, ..., Q$ (Q is the total number of edges in a branch), and $n = 1, ..., N - 1$ ($N - 1$ is the total number of nodes along the e^{th} edge). Edges that connect successive elementary vertices $v_{e,n(i)}$ and $v_{e,n(i+1)}$ were labeled $e_{v_{e,n(i)}, v_{e,n(i+1)}}$. Note that $e_{R,V_{T_s}}$ and $e_{B_{k(i)}, B_{k(i+1)}}$ are composed from subsets of elementary edges $e_{v_{e,n(i)}, v_{e,n(i+1)}}$.

Adjacency matrix. From the structural data we built an adjacency matrix that encoded information about the connectivity of the network. The adjacency matrix was organized such that each element represents a connection (graph edge) between an input vertex (row) and an output vertex (column). The axon and axon arborizations are represented by tree-network graphs. The associated adjacency matrix describes the composite hierarchical organization of the tree, where a common axonal segment ('parent segment', identified by all elements on the diagonally-shaped line in the connectivity matrix) gives rise to a number of daughter axonal branchlets.

For the purpose of our analysis we chose to reduce the full adjacency matrix (containing nodes from the sets v_R, v_B, v_T, and $v_{e,n}$) to an adjacency matrix containing vertices from the sets v_R, v_B, and v_T. It is this reduced matrix that we used in our analysis. This reduced adjacency matrix was created for three reasons. First, branching points define the variable paths of action potentials (signals) transmission through the axon. Second, in the more general case of myelinated axons, bifurcation points typically corresponds to unmyeliated locations where a node of Ranvier is present and thus where the signal propagates down the axon by saltatory conduction. This agrees with the methodology of our analysis which focuses on pairs of defined vertices that have an active role in neural signaling. Finally, synaptic terminals define the different paths of signaling in the axonal arbor and are thus necessary for comparisons between selected branches in the tree. And synaptic terminals represent active areas of the axon where synaptic transmission to one or multiple post-synaptic neurons occurs.

Deriving the Path Lengths of Axons and Axonal Arborization Branches. As described, pairs of vertices (v_i, v_j) in the graph were connected by a directed geometric edge $e_{i,j}$. Each vertex in the graph was associated with a physical position in three dimensional space, and each edge followed a convoluted geometric path in

space determined by the actual physical morphology. Given an edge $e_{i,j}$, the convoluted distance, i.e. the physical distance $\tilde{d}_{i,j}$ of the edge between any two vertices v_i and v_j can be defined by the path integral $\tilde{d}_{i,j}$ from v_i to v_j. To compute it, we took into account all segments forming a particular edge, and then calculated the sum of all Euclidean distances between pairs of nodes for a given segment. For example, to calculate the convoluted distance (path integral) $\tilde{d}_{B_k,B_w}$ from branching vertex v_{B_k} to branching vertex v_{B_w}, we approximated the length by:

$$\tilde{d}_{B_k,B_w} = \sum_N \overline{d}_{n_m,n_q} \tag{6}$$

where $\overline{d}_{n_m,n_q}$ is the Euclidean distance $\left(\overline{d}_{n_m,\,n_q} = \sqrt{\left(x_{n_m} - x_{n_q}\right)^2 + \left(y_{n_m} - y_{n_q}\right)^2 + \left(z_{n_m} - z_{n_q}\right)^2}\right)$ computed between pairs of nodes (v_{e,n_m}, v_{e,n_q}), and N is the number of sub-edges e_{n_m,n_q} forming e_{B_k,B_w}.

The tortuous path connecting the root vertex v_R to a selected synaptic terminal v_{T_s} was the sum of all convoluted distances $\tilde{d}_{B_k,B_w}$:

$$\tilde{d}_{R,T_s} = \sum_Q \tilde{d}_{B_k,B_w} \tag{7}$$

where Q indicates the total number of edges e_{B_k,B_w} that compose the selected path $p(v_R,\ v_{T_s})$ in the axonal tree. For example, for the neuron represented by the graph in Fig. 5 (path highlighted in red), the total distance between the root vertex and the synaptic vertex v_{T_3} was computed by summing up all distances of the edges e_{B_k,B_w} between the branching vertices along that path, which in turn were obtained by the sum of all distances of the multiple sub-edges e_{n_m,n_q} they were composed of.

Calculations of Signaling Parameters. Axon conduction velocity is a function of several biophysical and morphological properties, including axon thickness, branching, ion channel density and variety, and myelination[24]. Considering the predominant role that axon thickness and degree of myelination have in defining the conduction velocity of action potential in mammalian axons[47], in this work we derived estimated values of conduction velocity based on only these two structural factors. For our data, we assumed that the arbors of all our basket cell neurons were unmyelinated based on morphological empirical results and geometrically-based models from the literature[25–27]. While it is generally agreed that the conduction velocity of myelinated axons increases linearly with increasing diameter[37,48,49], the situation for non-myelinated axons is less definitive. Theoretical arguments and empirical measurements on large invertebrate axons suggest that the conduction velocity of non-myelinated axons is proportional to the square root of the axonal cross-sectional diameter[27]. However, other results seem to suggest that such a square-root relationship at least in invertebrates oversimplifies the situation[50,51]. In the vertebrate nervous system however very fine non-myelinated axons were found to support a linear relationship[28].

In our data, the diameter of the axons we studied were comparable to the fine non-myelinated axons in[28]. Therefore, we assumed a linear relationship between axonal diameter and conduction velocity (signaling speed) of non-myelinated axons. Hoffmeister *et al.* experimentally derived the linear relationship $s = 0.24 * C$ between the circumference C of non-myelinated fibers and their signaling speed s. Our data had axonal segments expressed as diameters, not circumference. We therefore wrote the equivalent relationship: $s = KD$ with $K = 0.24 * pi = 0.75$, where s is signaling speed, D is the axon diameter, K is a constant, and the relationship between circumference and diameter is $C = 2 * pi * D$. The same relationship has been used in other recent work to investigate the volume-latency trade-off in neocortical axons[52].

We used this relationship to estimate the signaling speed s_{ij} between vertex pairs v_i and v_j:

$$s_{ij} = 0.75D \tag{8}$$

where D is the (constant) diameter of the axonal segment connecting v_i and v_j. If v_i and v_j are consecutive measurement sample vertices between branching points, $v_{e,n}$, D was typically constant because $D_i \sim D_j$ due to the sampling resolution of the original data. If v_i and v_j corresponded to two consecutive branching points, D was calculated as the arithmetic mean of the diameters of all sample points between successive vertices $v_{B_{k(i)}}$ and $v_{B_{k(i+1)}}$. Considering that sampling points along individual edges of the axon were characterized by small diameter changes, the average represented a good approximation of the diameter of that particular axonal segment.

Neurons in our dataset had computed conduction velocity values of 0.39 ± 0.17 m/s ($\mu \pm \sigma$, with μ being the mean computed on the raw distribution of conduction velocities from all 11,575 axonal branches, and σ the standard deviation of their distribution). The histogram in Fig. 6 shows the distribution of signaling speeds independently computed for all branches of the entire data set of 57 basket cells. We also considered the distributions of conduction velocities for individual neurons in order to better characterize the behavior of single cells. For each neuron we computed the mean value of the conduction velocities of all axonal branches. We then calculated the mean value of the 57 computed means, obtaining a mean value of all means of 0.38 m/s.

The signaling latency or propagation delay τ_{ij} of action potentials (signals) traveling between vertices v_i and v_j were calculated as:

$$\tau_{ij} = \frac{d_{ij}}{s_{ij}} \tag{9}$$

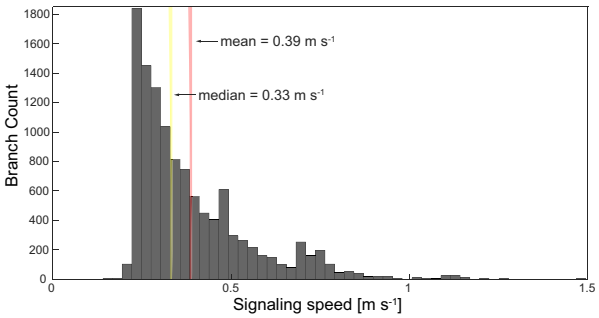

Figure 6. Global distribution of the signaling speeds independently computed for all branches of the 57 basket cells. Signaling speeds of individual axon branches range from 0.15 to 1.49 m/s (mean 0.39 m/s, median 0.33 m/s).

where d_{ij} is the distance measured between vertex i and vertex j as described in the previous section. Because our analysis is highly sensitive to the convoluted path length between vertices, the calculation of signaling speed and propagation delay depended on the kind of vertices that are considered in the arbor graph. For pairs of consecutive vertices $v_{e,n}$ belonging to individual edges between branching vertices, the distance was calculated using the Euclidean distance between them, and the dynamic parameters, conduction velocity and signaling latency, were obtained according to Eqs 10 and 11. On the other hand, for vertices corresponding to branching points or synaptic terminals of the arbors, the dynamics was computed slightly differently taking into consideration that any of these convoluted edges can be decomposed into multiple sub-edges of uniform diameter and conduction velocity. More specifically, for each branch of the axon arbor, that is the path connecting the root vertex v_R to a synaptic terminal v_{T_s}, we calculated the total propagation delay as the sum of all piecewise propagation delays obtained for all edges e_{B_k,B_w} forming that branch:

$$\tau_{R,T_s} = \sum_{Q} \tau_{B_k,B_w} \tag{10}$$

with Q being the total number of edges connecting the initial segment to a synaptic terminal. In turn, each edge in the branch can be decomposed into multiple sub-edges e_{n_m,n_q} along which the signal propagates with a temporal latency defined as:

$$\tau_{B_k,B_w} = \sum_{N} \tau_{e,n_m,n_q} \tag{11}$$

where N is the total number of sub-edges e_{n_m,n_q} in e_{B_k,B_w}. We then used the information about axonal path lengths and latencies to compute the signaling speed. By doing this we avoided averaging different diameters in the chosen path for the calculation of a mean signaling speed. We instead considered multiple conduction velocities obtained for short segments between consecutive edges in the path that had a constant diameter D. We then used the resulting combination of propagation delays for estimating the signaling speed between the axon initial segment and its synaptic terminals. This more accurately reflects the true values of action potential propagation speeds.

Estimation of the Refractory Period. The refractory period was estimated along the entire length of the axon, from the soma (root vertex) to each synaptic terminal (synaptic vertex). Since each path from root vertex to a terminal vertex was defined as a branching path, it is equivalent to saying that the refractory period was estimated for all axonal sampling points (i.e. nodes) along each specific branch, as a function of the distance between root and terminal vertex. We estimated the refractory period from empirical measurements reported in the literature[25]. More specifically, we approximated the decreasing refractory period from soma to synaptic terminals with a curve whose descending slope was inferred from actual measurements of channel density variation and membrane inactivation properties[25]. We set a lower bound to 1 ms, which is about the absolute lower limit for any neuron due to fundamental limitations associated with the kinetics of Na^+ channels. The maximum value for R_j was set to 2.5 ms, which we justify by three specific arguments: first, 2.5 ms represents a reasonable value as the average of the generally accepted range (0–5 ms) associated with neocortical neurons[53–56]; second, work on demyelinated axons and the supercritical Na^+ channel density proposed in[25] implies that higher values would not be physically appropriate; finally, a change of refractory period from 1 ms to 2.5 ms is in agreement with the 2.4-fold increase of activation time for Na^+ channels suggested by the experiments in[25]. A detailed explanation about the rationale for computation of a local refractory period for each vertex in the axon graph is reported in the online Supporting Information.

Computation and Statistics of the Refraction Ratio. We considered the refraction ratio[16] between all pairs vertex pairs (v_R, v_{T_s}): the root and terminal vertices in the network, the first vertex corresponding to the initial axon segment and the second to any synaptic terminal at the end of an axonal arborization. The value of the

refraction ratio was obtained by computing the numerical ratio between the refractory period of any terminal vertex v_{T_s} and the propagation delay signaling latency along the axonal branching path $p(v_R, v_{T_s})$ connecting the root vertex to the terminal vertex. The refraction ratio was computed independently for all reconstructed axon arbor graphs across each of the 57 basket cell neurons in our dataset. This resulted in 11,575 statistically independent refraction ratio values.

Histograms were used to collect, visualize, and analyze the data. At the single neuron level, each histogram described the distribution of the computed refraction ratio for all pairs of root and terminal vertices in the analyzed axon arbor (c.f. Fig. 2A). The value of bin within the distribution (the height of individual plotted bars) reflected the number of branches associated with refraction ratio value falling within a the specific range for the interval defined for that particular bin. For each basket cell we independently (from other neurons) computed the median value of the distribution.

To visualize and compute the statistics on the entire dataset, we constructed a histogram for the raw distribution of refraction ratio values computed for the 11,575 axonal branching paths from each of the 57 neurons (c.f. Fig. 2B). To normalize against varying numbers of terminal vertices across our sample population of 57 neurons, we computed the median value of the median population (see the main text above) (c.f. Fig. 2C).

References

1. Gomez, T. M. & Spitzer, N. C. *In vivo* regulation of axon extension and pathfinding by growth-cone calcium transients. *Nature* **397**, 350–355 (1999).
2. Low, L. K. & Cheng, H.-J. Axon pruning: an essential step underlying the developmental plasticity of neuronal connections. *Philos Trans R Soc Lond B Biol Sci.* **361**, 1531–1544 (2006).
3. Xu, T. *et al.* Rapid formation and selective stabilization of synapses for enduring motor memories. *Nature* **462**, 915–919 (2009).
4. Braitenberg, V. & Schuez, A. *Cortex: Statistics and Geometry of Neuronal Connectivity*. (Heidelberg: Springer, 1998).
5. Kulkarni, G. *et al.* Experimental evidence for unc-6 (netrin) axon guidance by stochastic fluctuations of intracellular unc-40 (dcc) outgrowth activity. *Biology Open* **2**, 1300–1312 (2013).
6. Mortimer, D., Dayan, P., Burrage, K. & Goodhill, G. J. Bayes-Optimal Chemotaxis. *Neural Comput.* **23**, 336–373 (2011).
7. Mitchison, G. Neuronal Branching Patterns and the Economy of Cortical Wiring. *Proc. Biol. Sci* **245**, 151–158 (1991).
8. Cuntz, H., Forstner, F., Borst, A. & Häusser, M. One Rule to Grow Them All: A General Theory of Neuronal Branching and Its Practical Application. *PLOS Computational Biology* **6**, 1–14 (2010).
9. Budd, J. & Kisvárday, Z. Communication and wiring in the cortical connectome. *Front Neuroanat.* **6**, 42 (2012).
10. Chklovskii, D. & Koulakov, A. Maps in the brain: what can we learn from them? *Annu. Rev. Neurosci.* **27**, 369–392 (2002).
11. Cherniak, C. Local optimization of neuron arbors. *Biol. Cybern.* **66**, 503–510 (1992).
12. Chklovskii, D., Schikorski, T. & Stevens, C. Wiring Optimization in Cortical Circuits. *Neuron* **34**, 341–347 (2002).
13. Ahn, Y.-Y., Jeong, H. & Kim, B. Wiring cost in the organization of a biological neuronal network. *Physica A* **367**, 531–537 (2006).
14. Koulakov, A. A. & Chklovskii, D. B. Orientation Preference Patterns in Mammalian Visual Cortex. *Neuron* **29**, 519–527 (1991).
15. Budd, J. M. L. *et al.* Neocortical Axon Arbors Trade-off Material and Conduction Delay Conservation. *PLoS Comput. Biol.* **6**, 1–25 (2010).
16. Silva, G. A. The effect of signal latencies and node refractory states on the dynamics of networks. *arXiv* **2232026** (2018).
17. Buibas, M. & Silva, G. A. A framework for simulating and estimating the state and functional topology of complex dynamic geometric networks. *Neural Comp.* **23**, 183–214 (2011).
18. Manor, Y., Koch, C. & Segev, I. Effect of geometrical irregularities on propagation delay in axonal trees. *Biophy. J.* **60**, 1424–1437 (1991).
19. Holme, P. & Saramaki, J. Temporal networks. *Springer* (2013).
20. George, B. & S., K. Spatio-temporal networks: Modeling and algorithms. *Springer Briefs in Computer Science* (2013).
21. Kawaguchi, Y., Karube, F. & Kubota, Y. Dendritic branch typing and spine expression patterns in cortical nonpyramidal cells. *Cereb. cortex* **16**, 696–711 (2006).
22. Ascoli, G. A. Mobilizing the base of neuroscience data: the case of neuronal morphologies. *Nat. Rev. Neurosci.* **7**, 318–324 (2006).
23. Ascoli, G. A., Donohue, D. E. & Halavi, M. Neuromorpho.org: A Central Resource for Neuronal Morphologies. *J. Neurosci.* **27**, 9247–9251 (2007).
24. Debanne, D. Information processing in the axon. *Nat. Rev. Neurosci.* **5**, 304–316 (2004).
25. Hu, H. & Jonas, P. A supercritical density of Na^+ channels ensures fast signaling in GABAergic interneuron axons. *Nat. Neurosci.* **17**, 686–693 (2014).
26. Waxman, S. & Swadlow, H. Ultrastructure of visual callosal axons in the rabbit. *Exp. Neurol.* **53**, 115–127 (1976).
27. Waxman, S. & Bennet, M. Relative Conduction Velocities of Small Myelinated and Non-myelinated Fibres in the Central Nervous System. *Nature* **239**, 217–219 (1972).
28. Hoffmeister, B., Jänig, W. & Lisney, S. A proposed relationship between circumference and conduction velocity of unmyelinated axons from normal and regenerated cat hindlimb cutaneous nerves. *Neurosci.* **42**, 603–611 (1991).
29. Luhmann, H. J., Greuel, J. M. & Singer, W. Horizontal Interactions in Cat Striate Cortex: II. A Current Source-Density Analysis. *Eur. J. Neurosci.* **2**, 358–368 (1990).
30. Hirsch, J. & Gilbert, C. Synaptic physiology of horizontal connections in the cat's visual cortex. *J. Neurosci.* **11**, 1800–1809 (1991).
31. Bringuier, V., Chavane, F., Glaeser, L. & Frégnac, Y. Horizontal Propagation of Visual Activity in the Synaptic Integration Field of Area 17 Neurons. *Science* **283**, 695–699 (1999).
32. Lohmann, H. & Rörig, B. Long-range horizontal connections between supragranular pyramidal cells in the extrastriate visual cortex of the rat. *J. Comp. Neurol.* **344**, 543–558 (1994).
33. Telfeian, A. & Connors, B. Widely integrative properties of layer 5 pyramidal cells support a role for processing of extralaminar synaptic inputs in rat neocortex. *Neurosci. Lett.* **343**, 121–124 (2003).
34. Kandel, E., Schwartz, J. &Jessel, T. *Principles of neural science*. (4th ed.) (McGraw-Hill, Health Professions Division, New York, USA, 1991).
35. Paintal, A. Effects of temperature on conduction in single vagal and saphenous myelinated nerve fibers of the cat. *J. Physiol.* **180**, 20–49 (1965).
36. Hursh, J. Properties of growing nerve fibers. *Am. J. Physiol.* **127**, 140–153 (1939).
37. Gasser, H. & Grundfest, H. Axon diameters in relation to spike dimensions and conduction velocity in mammalian fibers. *Am. J. Physiol.* 127–393 (1939).
38. Quandt, F. & Davis, F. Action potential refractory period in axonal demyelination: a computer simulation. *Biol. Cybern.* **67**, 545–552 (1992).
39. Lee, D., Lee, S.-G. & Kim, S. A compartment model with variable ion channel density on the propagation of action potentials along a nonuniform axon. *Eur. J. Neurosci. B* **85**, 400 (2012).

40. Kopysova, I. L. & Debanne, D. Critical role of axonal A-type K^+ channels and axonal geometry in the gating of action potential propagation along CA3 pyramidal cell axons: A simulation study. *J. Neurosci.* **18**, 7436–7451 (1998).
41. Raastad, M. & Shepherd, G. Single-axon action potentials in the rat hippocampal cortex. *J. Physiol.* **548**, 745–752 (2003).
42. Holme, P. & Saramäki, J. Temporal networks. *Phys. Rep.* **519**, 97–125 (2012).
43. Sporns, O. Small-world connectivity, motif composition, and complexity of fractal neuronal connections. *Biosystems* **85**, 55–64 (2006).
44. Towlson, E. K., Vértes, P. E., Ahnert, S. E., Schafer, W. R. & Bullmore, E. T. The Rich Club of the C. elegans Neuronal Connectome. *J. Neurosci.* **33**, 6380–6387 (2013).
45. Radivojevic, M. *et al*. Tracking individual action potentials throughout mammalian axonal arbors. *eLife* **6**, 2–23 (2017).
46. Liu, R. *et al*. High Density Individually Addressable Nanowire Arrays Record Intracellular Activity from Primary Rodent and Human Stem Cell Derived Neurons. *Nano Lett.* **17**, 2757–2764 (2017).
47. Waxman, S. Determinants of conduction velocity in myelinated nerve fibers. *Muscle Nerve* **3**, 141–150 (1980).
48. Hursh, J. Conduction velocity and diameter of nerve fibers. *Am. J. Physiol.* **127**, 131–139 (1939).
49. Rushton, W. A. H. A theory of the effects of fibre size in medullated nerve. *J. Physiol.* **115**, 101–122 (1951).
50. Hodes, R. Linear relationship between fiber diameter and velocity of conduction in giant axon of squid. *J. Neurophysiol.* **16**, 145–154 (1953).
51. Paintal, A. The influence of diameter of medullated nerve fibres of cats on the rising and falling phases of the spike and its recovery. *J. Physiol.* **184**, 791–811 (1966).
52. Wang, S. S.-H. *et al*. Functional Trade-Offs in White Matter Axonal Scaling. *J. Neurosci.* **28**, 4047–4056 (2008).
53. Chen, N., Yu, J., Qian, H., Ge, R. & Wang, J.-H. Axons Amplify Somatic Incomplete Spikes into Uniform Amplitudes in Mouse Cortical Pyramidal Neurons. *PLoS ONE* **5**, 1–10 (2010).
54. Khaliq, Z. M. & Raman, I. M. Axonal Propagation of Simple and Complex Spikes in Cerebellar Purkinje Neurons. *J. Neurosci.* **25**, 454–463 (2005).
55. Berry, M. J. & Meister, M. Refractoriness and Neural Precision. *J. Neurosci.* **18**, 2200–2211 (1998).
56. Shu, Y., Duque, A., Yu, Y., Haider, B. & McCormick, D. A. Properties of Action-Potential Initiation in Neocortical Pyramidal Cells: Evidence From Whole Cell Axon Recordings. *J. Neurophysiol.* **97**, 746–760 (2007).

Acknowledgements

This work was supported by a Swiss National Science Foundation (SNSF) Mobility Fellowship (P2ELP2_168553) to FP. Army Research Office (ARO), United States Department of Defense (grant numbers 65375-NS and 63795EGII). And in part by unrestricted funds to the Center for Engineered Natural Intelligence (CENI).

Author Contributions

G.S. conceived the original hypothesis and analyses approach, with additional contributions by F.P. F.P., V.G. and G.S. conceived the project. F.P. and V.G. designed and organized the study. F.P. carried out all the analyses. All authors have contributed in interpreting the results and writing the manuscript.

Additional Information

Supplementary information accompanies this paper at https://doi.org/10.1038/s41598-018-28527-2.

Competing Interests: The authors declare no competing interests.

Publisher's note: Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Supporting Information

An Optimized Structure-Function Design Principle Underlies Efficient Signaling Dynamics in Neurons

Francesca Puppo[1,3], **Vivek George**[1,3], **Gabriel A. Silva**[1,2,3,*]

[1] Department of Bioengineering, University of California, San Diego, La Jolla, 92093 CA
[2] Department of Neurosciences, University of California, San Diego, La Jolla, 92093 CA
[3] Center for Engineered Natural Intelligence, University of California, San Diego, La Jolla, 92093 CA
[*] Email: gsilva@ucsd.edu

Estimation of the Refractory Period

Our rationale for justifying computing a local refractory period for each vertex was based on the following considerations. The inactive state of a neuron is strictly dependent on the recovery time of the cellular membrane after activation. Once an action potential is initiated at one point in the cell body, its propagation along the axon occurs through a chain of depolarization-repolarization membrane processes. An action potential only occurs in a limited area of the cell membrane, but the changes in membrane potential at the edges of this area are enough to initiate another action potential in a neighboring space, which then give rise to another action potential further down the axon, a process which results in the propagation of the action potential. Each single point of the axon and its neighborhood experience initiation of an action potential followed by a period of recovery during which Na^+ channels are in their inactive state, and the axon segment is refractory. Given the spatially local area of how we defined nodes along the axon and axonal arborization (see above sections), we assumed that every vertex in our tree graph model of each neuron could be associated with a locally active-refractory area of membrane along the axon, and that a locally-defined refractory period could be calculated for it.

Our estimation of the refractory period at individual vertices along the axon was based on very recent findings about the biophysical and dynamical properties of basket cells. In the work by Hu and colleagues[1] these authors carried out confocally visualized sub-cellular patch-clamp recordings in rat hippocampal slices at different sites in the soma and axons of basket cells. Their experimental measurements demonstrated a supercritical density of sodium ion (Na^+) channels with a non-uniform density profile along the axonal tree. In particular, sub-cellular mapping revealed a stepwise increase of Na^+ conductance density from the soma ($\rho_{Na} = 2.6$ channels μm^{-2}) to the proximal axon ($\rho_{Na} = 25$ channels μm^{-2} at 50 μm distance from the soma), followed by an additional gradual increase in the distal axon ($\rho_{Na} = 46.1$ channels μm^{-2} at 315 μm distance from the soma). High Na^+ channel density is normally required for action potential generation; in typical neuron axons Na^+ channels are typically concentrated at the axon initial segment where action potential generation occurs. Based on this detailed work however, Na^+ channel distribution differed somewhat in basket cell axons. This supercritical, gradual increase in the density of sodium channels was interpreted as ensuring high reliability and fast propagation of neural signaling in unmyelinated small axons[1]. Additionally, analysis of gating kinetics and voltage-dependence of activation and inactivation revealed that Na^+ channels in basket cell axons differed from somatic channels in their functional properties. In particular, the inactivation time constant was 2.4-fold faster for axonal than for somatic Na^+ channels, thus implying locally reduced refractory periods in the axon with respect to the soma.

Given this information, we decided to approximate the decreasing refractory period from soma to synaptic terminals with a curve whose descending slope was inferred from the actual measurements of channel density variation. Specifically, a lowest and highest bound were defined for the refractory period. The lowest bound was set to 1 ms, which is about the absolute lower limit for any neuron due to fundamental limitations associated with the kinetics of Na^+ channels. The maximum value for R_j was set to 2.5 ms, which we justify by three specific arguments: first, 2.5 ms represents a reasonable value as the average of the generally accepted range (0 - 5 ms) associated with neocortical neurons[2–5]; second, the work on demyelinated axons and the supercritical Na^+ channel density proposed in[1] implies that higher values would not be physically appropriate; finally, a change of refractory period from 1 ms to 2.5 ms is in agreement with the 2.4-fold increase of activation time for Na^+ channels suggested by the experiments in[1]. We thus determined the refractory period for any output vertex v_j of individual axon branches as a function of the distance of the vertex from the soma. Specifically, R_j values steeply decrease from 2.5 ms to $\sim$ 1.7 ms in the first 50 μm distance from the cell main body; it then further gradually reduces at the distal site of the axon until reaching a constant refractory period of 1 ms at a threshold length of 315 μm (Figure 1).

The function used to estimate the varying refractory period along individual branches was constructed from considerations of the experimental curve obtained for the Na^+ channel density variation along basket cell axons from the initial segment to distal parts of the axons (315 um) in[1] (see Figure 1 in[1]). The channel density steeply increased from the soma to the proximal axon out to about 50 μm from a density of 2.6 channels μm^2 to a value of about 25 channels μm^2. We empirically estimated the slope of the curve describing the profile of channel density change within this first section of the axon by taking the slope of the straight line connecting these two points in that data. Beyond about 50 μm the channel density increased less steeply from a value of 25 channels μm^2 to approximately 46.1 channels μm^2 at

315 μm from the soma. We similarly linearly estimated the changing channel density in this second section of the axon by calculating the slope of the line connecting these two points.

We then built the function approximating the gradual decrease of the refractory period along individual branches as a function of the distance from the soma or axon initial segment.

Since no information was reported for distances from the soma larger than 315 μm, and given the gradual decrease in the slope relative to the channel density curve, we assumed that by 315 μm^2 the refractory period nearly reached saturation at a minimal value of 1 ms, and assumed that beyond that distance any further changes in the refractory period were negligible. Moreover, observing that for larger distances the average diameter of axonal arborizations could be assumed to have no relevant variation, we found it to be a reasonable assumption since the geometry does affect the recovery properties of the membrane[6]. We also assumed that the refractory period varies along individual branches from the minimal value of 1 ms to the maximum of 2.5 ms as the inverse of the channel density profile, since as discussed above there is an inverse relationship between the refractory period and channel density. In an ideal scenario empirical data for the refractory period would exist along the continuum of axonal branches. Since this is not the case, we made assumptions that reflect the most current and detailed data available about the refractory period of Basket cell neurons. Importantly, we modeled the refractory period before and independent of calculations of the refraction ratio. In other words, the refractory period model was not adjusted to achieve an optimal ratio.

Trade-Off Conservation at The Synapses

In order to investigate how the dynamic properties vary throughout the axon hierarchy, we calculated the evolution of the ratio along individual axonal paths. We computed the refraction ratio for sequential pairs of vertices where each pair includes the root vertex v_R as source vertex and one of the $Q-1$ branching points v_B as output vertex, where each branching vertex is selected one at a time descending along the axonal branch (or path $p(v_R, v_{T_s})$) to the synaptic terminal (Figure 2 A). In Figure 2B we report the evolving refraction ratio for all branches of a single basket cell (coded as different colors for each branch; Fig.2B). When evaluated this way, there is a converging trend of the refraction ratio towards optimality as it approaches the terminal output vertices. The inset in Figure 2B shows that a high percentage of the total number of branches in the cell have high values of refraction ratio early in the tree, which then converge quickly to near-optimal values as the terminal synaptic vertices for each branch are approached (*c.f.* Figure 2 in the main text).

Figure 3A shows the distribution towards optimal convergence of the refraction ratio for a representative basket cell as a function of the branching order at which convergence towards the optimal range occurs. The branching order corresponds to depth of a selected branching vertex on the axonal path $p(v_R, v_{T_s})$). The vertex depth is here defined by the number of bifurcations that separate that specific branching vertex from the root vertex. For this analysis, we arbitrarily fixed the range $0.5 \leq \Lambda_{ij} \leq 1.5$ as optimality range of convergence. The histogram shows what is the smallest branching order in the axon hierarchy at which the refraction ratio converges. Figure 3B shows the same metric combined for our whole dataset of 57 neurons. More specifically, the histogram shows the number of neurons whose maximum number of branches have ratio converging to optimality as a function of the first branching at which convergence occurs. To build this histogram, we first selected, for each neuron in the data set, the branches whose ratio at synapses falls within an optimal range (here arbitrarily set to $0.5 \leq \Lambda_{ij} \leq 1.5$); then, for each branch we determined the branching order in the hierarchy at which convergence first occurs, as described for data in Figure 3A; finally, for each of the 57 cells, we calculated the order of bifurcation that maximizes the number of converging branches. We obtained 57 different orders, one for each neuron, and used them to build the here reported distribution.

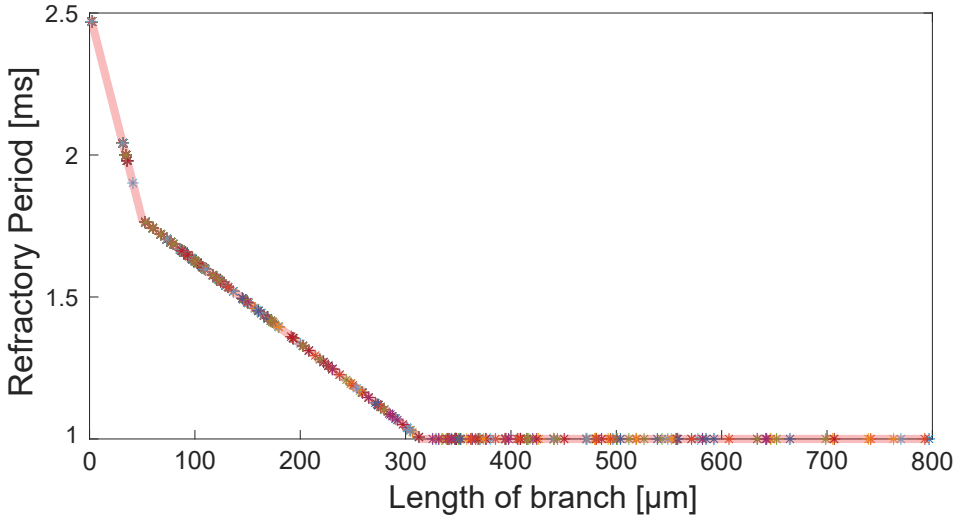

Figure 1. Estimation of the refractory period for an individual basket cell. R_j is plotted as a function of the path length. Its values steeply decrease from 2.5 ms to $\sim$ 1.7 ms in the first 50 μm distance from the cell main body; then, R_j gradually reduces at the distal site of the axon until reaching a constant refractory period of 1 ms at a threshold length of 315 μm.

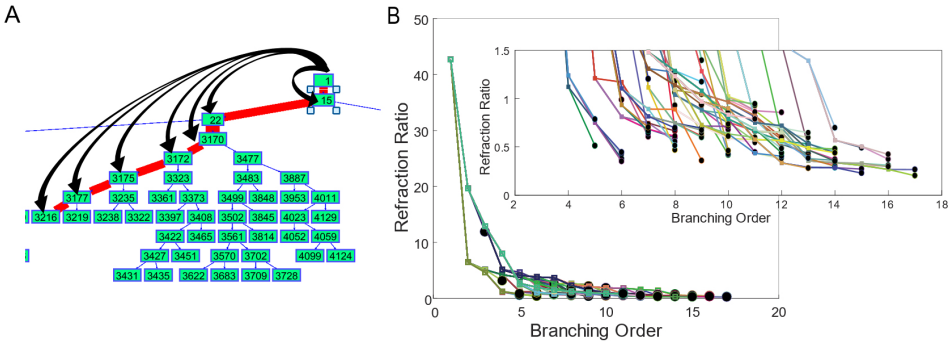

Figure 2. Convergence of the refraction ratio towards an optimal range for one representative basket cell neuron. (A) Schematic representation of the different pairs of vertices for which the evolving ratio was computed, starting from the root vertex v_R at the soma and ending at the synaptic terminals v_{T_s} for all paths $p(v_R, v_{T_s})$. (B) Evolving signaling ratio for all paths in the axon arbors of the same cell. Different colors represent different branches. The x-axes shows the order of bifurcation (depth of the tree) in the hierarchical structure from the origin to the terminal vertex. The y-axes shows the value of the computed refraction ratios. Inset: zoomed in view of the same data. We assumed an optimal refraction ratio range to be within 0.5 and 1.5 for this analysis.

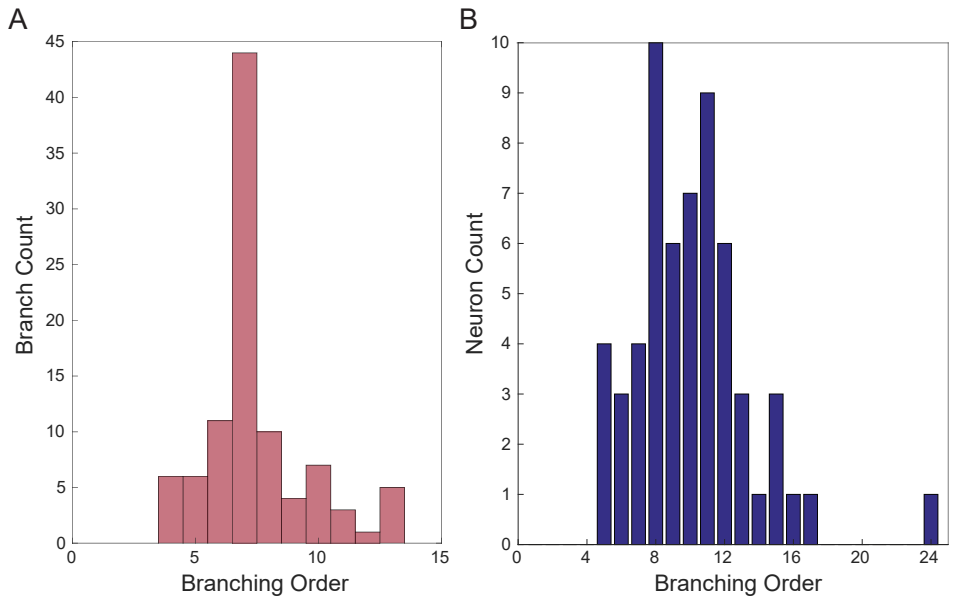

Figure 3. Branching order counts at which an optimal ratio is achieved. (A) Histogram showing the distribution of branching orders (depth in the tree-network) for a single neuron at which an optimal ratio is reached. We assumed the same optimal ratio 0.5 and 1.5 as in the rest of the analysis. (B) Data showing the results of the same analyis as for the representative neuron in panel A, but plotting for each neuron only the branch order that displayed the maximum number of axonal branches that achieved the optimal ratio at that particular branch order (see text).

References

1. Hu, H. & P., J. A supercritical density of Na^+ channels ensures fast signaling in GABAergic interneuron axons. *Nat. Neurosci.* **17**, 686–693 (2014).

2. Chen, N., Yu, J., Qian, H., Ge, R. & Wang, J.-H. Axons Amplify Somatic Incomplete Spikes into Uniform Amplitudes in Mouse Cortical Pyramidal Neurons. *PLoS ONE* **5**, 1–10 (2010).

3. Khaliq, Z. M. & Raman, I. M. Axonal Propagation of Simple and Complex Spikes in Cerebellar Purkinje Neurons. *J. Neurosci.* **25**, 454–463 (2005).

4. Berry, M. J. & Meister, M. Refractoriness and Neural Precision. *J. Neurosci.* **18**, 2200–2211 (1998).

5. Shu, Y., Duque, A., Yu, Y., Haider, B. & McCormick, D. A. Properties of Action-Potential Initiation in Neocortical Pyramidal Cells: Evidence From Whole Cell Axon Recordings. *J. Neurophysiol.* **97**, 746–760 (2007).

6. Paintal, A. The influence of diameter of medullated nerve fibres of cats on the rising and falling phases of the spike and its recovery. *J. Physiol.* **184**, 791–811 (1966).

Chapter 4, in full, is a reprint of *An Optimized Structure-Function Design Principle Underlies Efficient Signaling Dynamics in Neurons*, Puppo F, George V, Silva G. This article appeared in the Nature Scientific Reports 2018, volume 8:10460. This dissertation author was the second on this paper.

Chapter 5

Conclusions and Future Work

The main thrust of this work was to takes steps towards constructing generalized learning architectures based on principles from the theory of operation of the brain. Although this is a lofty problem and the contributions of this work are modest steps towards that end, this work lays an important foundation. Despite considerable advances in the field of artificial neural networks and its ever increasing application space, the brain is still a formidable learning machine. How the brain computes and operates continues to inspire scientists, engineers and artists. In search for the underlying principles upon which the brain operates there are some experimental guides[4], but there is also much that is unknown[7]. The goal of this dissertation is not only to use principles from neuroscience to inform new machine learning algorithms, but also to contribute to neuroscience literature.

In the first part of this dissertation, we investigated the dynamics supported by the C. elegans connectome through the study of Temporal Sequences(TSs). TSs are causal neuronal signaling pathways of a network which arise from walks of signals on the network. The neuronal signaling pathways are constrained by the concurrent network dynamics. The mathematical study of structure of the dynamics, whether it may be TSs, graphs, vectors or other mathematical structures one may construct, is potentially more informative than only studying the evolution of

a nodal state variable, for example, recording the membrane voltage of a set of neurons in the network. In a more sophisticated analysis one may merge the evolution of salient state variables and the structure of the dynamics.

We constructed a geometric directed network of the C. elegans connectome using experimentally derived physiological data[10, 9, 3]. To simulate network dynamics we implemented a simplified version of the so-called Geometric Dynamic Perceptron Model[8] to describe nodal behaviour. The refractory nature of the individual neurons lead to non-trivial neuronal signaling pathways, i.e. TSs. These TSs are non-trivial because of network signaling contention. Not all possible paths supported by the structural graph must be realizable concurrently, as such, there is an order to the possible causal interactions. Only observing the responses, in terms of node activity, of a subset of the nodes in the network deprives one of elucidating and studying the intricate paths carved through the network by signals.

From a neuroscience perspective, our network simulation results showed that upon stimulus of the chemosensory neuron ASEL, the dynamics on both the entire connectome and in a subset of the connectome, resulted in what looked like contraleteral activations of the ventral(VB) and dorsal (DB) classes of motor neurons. The contralateral motor-neuron activation pattern, is a general neuron activity pattern which is implicated in animal movement[10, 5]. Through the analysis of TSs we were able to elucidate the network paths which gave rise to the observed node activations. Then we recast the use of TSs to compare networks through quantifying the difference between networks on the basis of their causal signaling paths. As an example we compared TSs from the sub-networks to the TSs from the full connectome. This was interesting because one may reasonably want to design a sub-network such that at the least every neuronal signaling path within the sub-network should also exist in the full network, but not necessarily vice versa. By comparing TSs from the two networks we were able to quantify the preservation of neuronal signaling pathways.

For any scientist looking to study sub-networks, our results provided a cautionary note, but

also a partial remedy. We found that the neuronal signaling on sub-network outside the context of the full-network behaves differently than the neuronal signaling of the full network. We provided a measure (the Similarity Measure) to determine how differently. The remedy to picking a good representative sub-network lies in approaching the problem of sub-network generation by using computationally cheap approaches to study the preservation of some macroscopic behaviours, for example, making sure that TSs between the network are statistically similar and preserve neuron activation rates. That is how much of the TSs from the entire connectome formed the same pathways as those in the sub-network and maintaining the neuron activation rates between the two networks.

By analyzing the TSs for repeating sub-sequences we can hypothesize the pattern generating circuits supported by the structural connectome in the presence of concurrent network signaling dynamics. The analysis of sequence repetitions also provided a method to decompose the rich signaling dynamics observed on the network into a fundamental set of neuronal signaling pathways. We called this fundamental set the set of Basis Sequences. Basis sequences provide a minimal descriptions of TSs and from the canonical signaling pathways on the connectome.

This work has some profound implications for machine learning. Existing artificial neural network approaches focus on minimizing cost functions relative the activations of some node in an output layer. We upend that paradigm by proposing the analysis of TSs to analyze and categorize dynamics on a network. Since the work on C. elegans was focused on neuronal signaling pathways we constructed TSs. We again note that a TS is only one view of the network dynamics, there are several other mathematical structures which can be built and analyzed. Furthermore, through the analysis of the structure of the dynamics we increase the mathematical expressiveness of the network dynamics.

Our second work focused on how much of the networks dynamics one must capture to then further analyze and how to preserve the concurrency of network signaling pathways in a mathematical object. In a machine learning paradigm which depends on the collection of analysis

of network statistics from a dynamical system it is unclear what are the physical observational limits after which the collection of new observations prove to be less informative. One answer to this quandary is to wait for long time until the network activity reaches steady-state in the same sense that one considers steady-state solutions to differential equations. Even if one waited for a long time for the network activity to reach steady-state, since our analysis is predicated upon the construction of mathematical objects representing the dynamics we may still want to know how much of the network dynamics in steady-state one must capture.

We proposed two distinct methods to determine the steady-state statistics, that is, start of periodic network activity and the period of the repeating activity. One method was based upon the sum of node activations over time, and the other was based on finding repetitions in strings representing network states. The string repetition method is the more computationally efficient of the two. It has a computational complexity of $O(N)$, where N is the number of samples. Although the string method is efficient, it is not robust to noise, in that, the computational complexity diminishes when considering almost periodic strings. The summation method more easily can be extended to account for noise, albeit, in an approximate way. This is something we have left for the future since the current networks under consideration have no need for it.

An opportunity for enhancement of the string method is in extending it to consider strings in an online way. The goal would be that once the steady-state statistics were found for N samples, then if m additional samples were considered, the resulting complexity should be $O(m)$, rather than $O(N+m)$. One approach to reduce the complexity of an online approach is to minimize the reconstruction of fundamental data structures used for the string search.

Upon determining the system's steady-state start time and period, we were interested in developing a mathematical description encompassing the ensemble dynamics which could also delineated the local structures arising in the dynamics. We used a directed acyclic graph to capture the causal signaling activity in the network. Given that the signaling activity is taking place on the structural backbone of the network, the resulting graph description involves the

repetition of labels. To accommodate this need we used a technique commonly used to describe chemical molecules, but extended the idea to a directed graph setting. Although several tools exist to measure distances between graphs[2], machine learning software packages for the most part operate on vectors. Once the graph was constructed, we created a vector space embedding[6] of the graph which counted the number of subgraphs of various heights present in the graph, and in that process inherently captured the interaction of subgraphs. Not only does the vector space embedding allow the use of a multitude of classifiers, but it also provides a test bed to study the relationships between the network's topology, node models and dynamics in order to inform the elusive search for structure-function relationships. Research within this end-to-end framework is an area of active inquiry.

Our final work in this dissertation experimentally verifies a theoretical neuronal design principle[8] underlying axonal development and morphology. We found that there was a balance between the membrane refractory periods and axonal signaling delay. A guiding learning principle from neuroscience used in machine learning is spike time dependent plasticity. This is one of the few established local learning rules with experimental data to provide guidance on its functional form[1]. This work is the first of many experimental works necessary for its characterization. In the future we intend to extend the Hebbian learning framework to include this phenomena as a local learning rule and to construct and optimize structural networks. On the experimental side we intend to study whether neuronal signaling modifications are associated with brain disorders and other conditions.

References

[1] G. Bi and M. Poo. Synaptic modification by correlated activity: Hebb's postulate revisited. *Annual Review of Neuroscience*, 24:139–166, 2001.

[2] Claire Donnat and Susan Holmes. Tracking network dynamics: A survey using graph distances. *The Annals of Applied Statistics*, 12(2):971–1012, June 2018.

[3] Marcus Kaiser and Claus C. Hilgetag. Nonoptimal Component Placement, but Short Processing Paths, due to Long-Distance Projections in Neural Systems. *PLOS Computational Biology*, 2(7):e95, July 2006.

[4] Wolfgang Maass. Searching for principles of brain computation. *Current Opinion in Behavioral Sciences*, 11:81–92, October 2016.

[5] Beverly J. Piggott, Jie Liu, Zhaoyang Feng, Seth A. Wescott, and X.Z. Shawn Xu. The Neural Circuits and Synaptic Mechanisms Underlying Motor Initiation in C. elegans. *Cell*, 147(4):922–933, November 2011.

[6] Nino Shervashidze, Pascal Schweitzer, Erik Jan van Leeuwen, Kurt Mehlhorn, and Karsten M. Borgwardt. Weisfeiler-Lehman Graph Kernels. *J. Mach. Learn. Res.*, 12:2539–2561, November 2011.

[7] Gabriel A. Silva. The Need for the Emergence of Mathematical Neuroscience: Beyond Computation and Simulation. *Frontiers in Computational Neuroscience*, 5, 2011.

[8] Gabriel A. Silva. The Effect of Signaling Latencies and Node Refractory States on the Dynamics of Networks. *arXiv:1804.07609 [q-bio]*, August 2019. arXiv: 1804.07609.

[9] Lav R. Varshney, Beth L. Chen, Eric Paniagua, David H. Hall, and Dmitri B. Chklovskii. Structural properties of the Caenorhabditis elegans neuronal network. *PLoS Computational Biology*, 7(2), 2011.

[10] J. G. White, E. Southgate, J. N. Thomson, and S. Brenner. The structure of the nervous system of the nematode Caenorhabditis elegans. *Phil. Trans. R. Soc. Lond. B*, 314(1165):1–340, November 1986.