UNIVERSITY OF CALIFORNIA SAN DIEGO

Towards Generalizable Machine Learning in Neuroscience using Graph Neural Networks

A thesis submitted in partial satisfaction of the requirements for the degree of Master of Science

in

Physics

by

Paul Y. Wang

Committee in charge:

Professor Gabriel Silva, Chair
Professor Henry Abarbanel, Co-chair
Professor Javier Duarte

2021

The thesis of Paul Y. Wang is approved, and it is acceptable in quality and form for publication on microfilm and electronically.

University of California San Diego

2021

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

In hindsight, my life has been defined by singular moments of spontaneity- whether deciding to apply for doctorate programs or deciding to graduate with a masters. While seemingly random, I firmly believe that the trajectory of my life has been strongly influenced by the excellent mentors whom I have had the utmost pleasure of meeting.

I thank William Fitzpatrick, my music teacher, for teaching me everything I know about how to learn and think. His insightful stories and profound statements on music led to my endearing passion for discovery and disdain for rote thinking. Stumbling into the field of physics was indeed a serendipitous moment; however, I strongly believe that the mentality instilled by Professor Fitzpatrick culminated in my innate curiosity in physics. To Professor Fitzpatrick: thank you very much for dedicating your life to teaching not only music but also how to think, grow, and explore!

I enrolled into college as an engineering physics major- essentially a double major between electrical engineering- where I absolutely abhorred my electrical engineering courses. Joining Ivan K. Schuller's lab was a purely random event: he was the only professor who replied back to my emails. Yet, my time with Professor Schuller was some of the most enriching and rewarding times of my life. Working with his group everyday was a pure joy in both scientific discovery and collaborative thinking. As a result, I delved deeply into experimental condensed matter physics research through which I discovered my passion for research and met some of the most diverse and extraordinary groups of individuals.

My research as an undergraduate culminated in a total of 6 coauthored papers, an achievement I attribute completely to the excellent mentors and collaborators I encountered in Ivan's group. First, I would like to thank Ivan for teaching me how to be a scientist with integrity foremost. Next, I must thank Dr. Ilya Valmianski who is easily the most brilliant individual I have met. Needless to say, Ilya has become a good friend, and I continue to look to him for advice. As my most frequent and prolific collaborator, I thank Dr. Javier del Valle who served as a mentor for most of my undergraduate career. Javier has been a most outstanding coauthor and, in my mind, has been the perfect teammate. Finally, I thank Professor Gabriel Ramirez who was instrumental in teaching me the publishing process. They say that it takes a village to raise a child; however, I believe that it also takes a village to raise a scientist! Along this line, there are many other individuals from Ivan's group who I wish to acknowledge including (not exclusively): Dr. Yoav Kalcheim, Dr. Jaime Wampler, Dr. Christian Urban, Dr. Stefan Guenon, Dr. Nikita Butakov. Ultimately, working with Ivan's group was nothing less than a pleasure and a true highlight of my academic career.

Graduate school is without a doubt the hardest endeavor I have ever attempted. I would like to extend my deepest thanks to the graduate advising team who tirelessly advocate for graduate students. Furthermore, I thank the physics faculty both at UCSD and other universities for writing and freely distributing their incredible wisdom and knowledge. From Professor Gabriel Silva's lab, I thank Dr. Vivek George for fostering a creative environment and introducing me to the paper upon which I built my research project. Together, we brainstormed more ideas than we could possibly tackle in a Ph.D. Finally, I would like to extend my deepest thanks to my chair, Professor Gabriel Silva, for providing excellent feedback during the publication process and for unequivocally supporting my academic career.

Chapter 1 and 2, in full, is a reprint of the material as it appears in "Generalizable Machine Learning in Neuroscience using Graph Neural Networks" in Frontiers in Artificial

Intelligence 2021. Wang, Paul Y.; Sapra, Sandalika; George, Vivek K.; Silva, Gabriel A., Frontiers Media, 2021. The thesis author was the primary investigator and author of this paper.

Appendix A, in full, is a reprint of the supplemental material for "Generalizable Machine Learning in Neuroscience using Graph Neural Networks" in Frontiers in Artificial Intelligence 2021. Wang, Paul Y.; Sapra, Sandalika; George, Vivek K.; Silva, Gabriel A., Frontiers Media, 2021. The thesis author was author of this supplemental material section.

VITA

Education

2014 – 2018

Bachelor of Science, Engineering Physics

University of California San Diego

2018 – 2021

Master of Science, Physics

University of California San Diego

Publications

1. *Origin of the current-driven breakdown in vanadium oxides: Thermal versus electronic*; I. Valmianski**, P. Y. Wang**, S. Wang, Juan Gabriel Ramirez, S. Guenon, Ivan K. Schuller. *Physical Review B* 98.19 (2018): 195144.
2. *Broadband Electrically Tunable Dielectric Resonators Using Metal-Insulator Transitions in VO_2*; N. Butakov, M. Knight, T. Lewi, P. Iyer, D. Higgs, H. Chorsi, J. Trastoy, J. Del Valle, I. Valmianski, C. Urban, Y. Kalchheim, **P. Wang**, P. Hon, I. Schuller, J. Schuller. *ACS Photonics 5.10 (2018): 4056-4060.*
3. *Subthreshold Firing in Mott Nanodevices*; J. del Valle, P. Salev, F. Tesler, N. Vargas, Y. Kalcheim, **P. Wang**, J. Trastoy, M. Lee, G. Kassabian, J. Ramirez, M. Rozenberg, I. Schuller. *Nature,* 569(7756), p.388.
4. *Thermally Reconfigurable Meta-Optics*; T. Lewi, N. Butakov, H. Evans, M. Knight, P. Iyer, D. Higgs, H. Chorsi, J. Trastoy, J. Del Valle, I. Valmianski, C. Urban, Y. Kalcheim, **P. Wang**, P. Hon, I. Schuller, J. Schuller *IEEE Photonics Journal* 11.2 (2019): 1-16.
5. *Reconfigurable semiconductor Mie-resonant meta-optics;* T. Lewi, N. Butakov, P. Iyer, H. Evans, D. Higgs, H. Chorsi, J. Trastoy, J. Del Valle Granda, I. Valmianski, C. Urban, Y. Kalcheim, **P. Wang**, I. Schuller, J. Schuller. *Proc. SPIE 11080, Metamaterials, Metadevices, and Metasystems* 2019, 110802P (5 September 2019)
6. *Generalizable Machine Learning in Neuroscience using Graph Neural Networks*; **P. Y. Wang**, S. Sapra, V. George, G. A. Silva. Accepted, *Frontiers in Artificial Intelligence*.
7. *Filament nucleation and growth in Mott insulators*; J. del Valle, R. Rocco , N. M. Vargas, P. Salev , Y. Kalcheim , P. Lapa , C. Adda, M. Lee , **P. Y. Wang** , L. Fratino , M. J. Rozenberg, I. K. Schuller. In Review.

ABSTRACT OF THE THESIS

Towards Generalizable Machine Learning in Neuroscience

by

Paul Y. Wang

Master of Science in Physics

University of California San Diego, 2021

Professor Gabriel Silva, Chair
Professor Henry Abarbanel, Co-Chair

Machine learning and neuroscience have enjoyed a golden era of prosperity over the past decade as the perfect confluence of technological advances have enabled extraordinary experiments and discovery. Though tightly intertwined in the past, advances in both fields have largely diverged such that the application of deep learning techniques to microscopic neural systems remains relatively unexplored. In this thesis, I present work bridging recent advances in machine learning and neuroscience. Specifically, relying on recent advances in whole-brain imaging, we examined the performance of deep learning models on microscopic neural dynamics and resulting emergent behaviors using calcium imaging data from the nematode C. elegans. We show that neural networks perform remarkably well on both neuron-level dynamics prediction and behavioral state classification. In addition, we compared the performance of structure agnostic neural networks and graph neural networks to investigate if graph structure can be exploited as a favourable inductive bias. To perform this experiment, we designed a graph neural network which explicitly infers relations between neurons from neural activity and leverages the inferred graph structure during computations. In our experiments, we found that graph neural networks generally outperformed structure agnostic models and excel in generalization on unseen C. elegans worms. These results imply a potential path to generalizable machine learning in neuroscience where pre-trained models are evaluated on unseen individuals.

1. Background

1.1. Introduction

Constructing generalizable models in neuroscience poses a significant challenge because systems in neuroscience are typically complex in the sense that dynamical systems composed of numerous components collectively participate to produce emergent behaviors. Analyzing these systems can be difficult because they tend to be highly non-linear in how they interact, can exhibit chaotic behaviors, and are high-dimensional by definition. As such, indistinguishable macroscopic states can arise from numerous unique combinations of microscopic parameters, i.e. parameters relevant to lower scales of organization. Thus, bottom-up approaches to modeling neural systems often fail since a large number of microscopic configurations can lead to the same observables (Golowasch et al. (2002); Prinz et al. (2004)).

Because neural systems are highly degenerate and complex, their analysis is not amenable to many conventional algorithms. For example, observed correlations between individual neurons and behavioral states of an organism may not generalize to other organisms or even to repeated trials in the same individual (Fregnac (2017); Churchland et al. (2010); Goldman et al. (2001)). Hence, individual variability of neural dynamics remains poorly understood and a fundamental obstacle to model development as evaluation on unseen individuals often leads to subpar results. Nevertheless, neural systems exhibit universal behavior: organisms behave similarly. Motivated by the need for robust and generalizable analytical techniques, researchers recently applied tools from dynamical systems analysis to simple

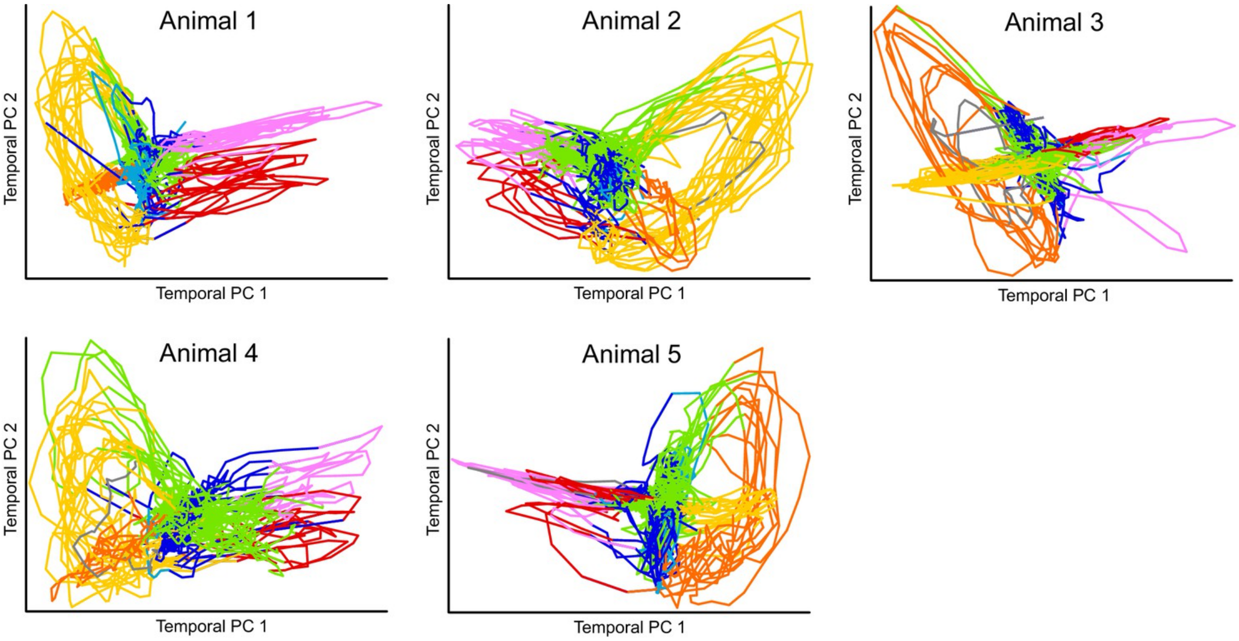


Figure 1.1: Time derivatives of calcium traces projected onto each individual organism's principal components. Distinct loops correspond to manifolds in latent space where colors correspond to behavior assigned in Kato et al. Reproduced with permission from Brennan and Proekt (2019). Copyright has been obtained.

organisms in hopes of discovering a universal organizational principle underlying behavior. These studies, made possible by advances in whole-brain imaging, reveal that neural dynamics live on low-dimensional manifolds which map to behavioral states (Prevedel et al. (2014); Kato et al. (2015)). This discovery implies that although microscopic neural dynamics differ between organisms, a macroscopic/global universal framework may enable generalizable algorithms in neuroscience. Nevertheless, the need for significant hand-engineered feature extraction in these studies underscores the potential of deep learning models for scalable analysis of neural dynamics.

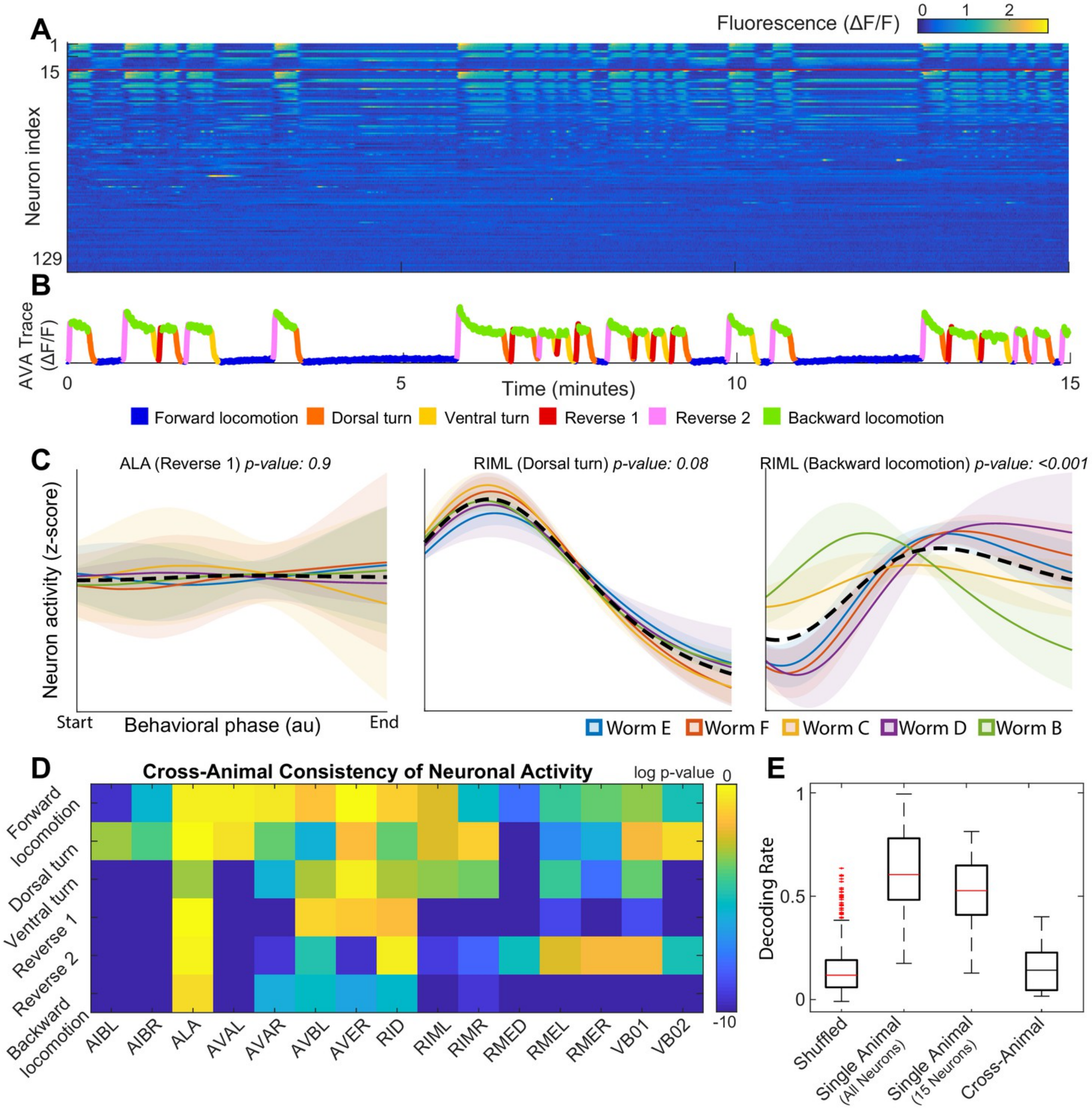


Figure 1.2: (A) Calcium signals recorded in one animal for∼15 minutes by Kato et al. (2015). Each row represents a single neuron. The top 15 rows (above the red line) correspond to neurons unambiguously identified in all animals (shared neurons). (B)Sample trace with corresponding behavioral state colored. (C) Neural dynamics of two neurons for specific behavior states. Colored solid lines are the mean activity for each animal, and the black dashed line is the mean activity for all animals. Shaded colored regions show 95% confidence intervals. (D)Probabilities that neural dynamics from different individuals were drawn from the same distribution. (E) Attempt by Brennan and Proekt (2019) to decode onset of backwards locomotion using neural dynamics for each animal and averaged neural dynamics across other four animals. Reproduced with permission from Brennan and Proekt (2019). Copyright has been obtained.

1.2. Universality/Generalizability in C. Elegans

The motor action sequence of C. elegans is one of the only systems for which experiments on whole-brain microscopic neural activity may be performed and readily analyzed. As such, numerous efforts have focused on building models that can accurately capture the hierarchical nature of neural dynamics and resulting locomotive behaviors (Sarma et al. (2018); Gleeson et al. (2018)). Taking advantage of this, Kato et al. (2015) investigated neural dynamics corresponding to a pirouette, a motor action sequence in which worms switch from forward to backward crawling, turn, and then continue forward crawling. Their analysis showed that most variations (~65%) in neural dynamics can be expressed by three components found through principal component analysis (PCA) and that neural dynamics in the resulting latent space trace cyclical trajectories on well-defined low dimensional manifolds corresponding to the motor action sequence (Figure 1.1). By identifying individual neurons, an experimental feat, these authors further determined that these topological structures in latent space were universally found among all five worms imaged in their study.

Following Kato et al. (2015), the authors published several studies focusing on global organizational principles of C. Elegans behavior (Nichols et al. (2017); Kaplan et al. (2020); Skora et al. (2018). Building on two of these works, Brennan and Proekt (2019) found consistent differences between each individual's neural dynamics, precluding the use of established dimensional reduction techniques. For example, among 15 neurons uniquely identified among all 5 worms, only 3 neurons displayed statistically consistent behavior (Figure 1.2D). Examples of inconsistent behavior for unequivocally identified neurons (ALA and RIML) are shown in Figure 1.2C where the average of ALA's activity fails to resemble the behavior of any worm and where

RIML's activity is consistent among all animals during dorsal turns, but inconsistent during reverse crawling. Resulting from these discrepancies, topological structures identified by performing PCA on each worm's neural activity were no longer observed when data from all worms was pooled together.

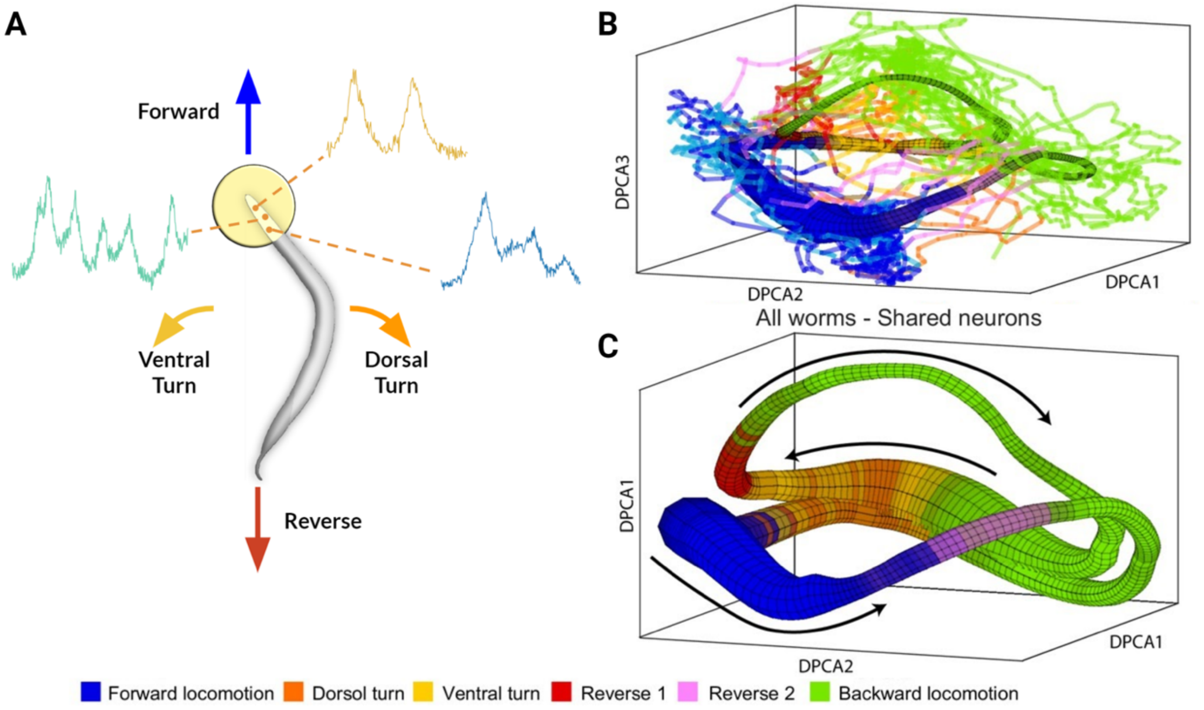


Figure 1.3: (A) Rendering of calcium imaging experiment where activity of neurons in the head of the worm is recorded. Coloured arrows show main motor action behavioral states. (B) and (C) Resulting manifold from Brennan and Proekt (2019). (B) Manifold constructed from activity of four worms with coloured lines indicating neural activity of fifth worm.(C)Manifold constructed from neural activity of uniquely identified neurons (n=15) shared among all 5 worms. Black arrows correspond to cyclical transition of motor action sequence and colors correspond to motor action states. Modified with permission from Brennan and Proekt (2019). Copyright has been obtained.

To address this issue, Brennan and Proekt (2019) introduced a new algorithm, asymmetric diffusion map modeling (ADMM), which maps the neural activity of any worm to an universal manifold (Figure 1.3). To achieve this, ADMM first performs time-delay embedding of neural activity into phase space. Next, a transition probability matrix is constructed by calculating distances between points in phase space using a Gaussian kernel centered on the subsequent timestep. Finally, this asymmetric diffusion map is used to construct a manifold representative of neural activity. Contrasting conventional dimensional reduction techniques, ADMM allowed quantitative modeling by mapping neural activity from the manifold and enabled the prediction of motor action states up to 30 s ahead. Despite its success, the algorithm heavily relies on hyperparameters, such as embedding parameters, which are difficult to justify and tune.

1.3. Deep Learning in Neuroscience and Graph Neural Networks

With the success of convolutional neural networks, researchers successfully applied deep learning to numerous domains in neuroscience (Glaser et al. (2019) including MRI imaging (Lundervold and Lundervold (2019)) and connectomes (Brown and Hamarneh (2016)) where algorithms can predict disorders such as autism (Brown et al. (2018)). Similarly, brain-computer interfaces (BCI) are a well-studied field related to our work as they focus on decoding macroscopic variables from measurements of neural activity. These studies generally involve fMRI or EEG data, which characterize neural activity on a population level, to varying amounts of success (Bashivan et al. (2015); Kwak et al. (2017); Mensch et al. (2017); Makin et al. (2020)). Regardless, a challenge for the field is developing generalizable algorithms to individuals unseen during training (Zhanget al. (2019)).

Although the application of deep learning to neuroscience has proved successful, a resounding question is whether or not a more suitable inductive bias exists that may further guide the development of deep learning models. Historically, the recent resurgence of interest in deep learning can be attributed to the development of the convolutional neural network which seeks to exploit translational invariance within the data. Building upon this, Cohen and Welling (2016) proposed that invariances within a dataset should motivate the construction of group equivariant neural networks where operations within the neural network are specifically chosen to be equivariant to the assumed group symmetry. Originally, Cohen and Welling (2016) developed equivariant neural networks for the *p4m* and *p4* group. Subsequently, equivariant neural networks were developed for the SO(3) group (Cohen et al. (2018)) and gauge transformations (Haan et al. (2020)).

Returning to the subject of neuroscience, the question remains whether or not a suitable inductive bias exists. Biophysically, neural systems can be modeled as a graph wherein vertices and edges correspond to neurons and synapses. Thus, when designing neural networks for machine learning neural systems, we can consider the inherent graph structure of the system and design an inductive bias equivariant to symmetries of the graph (generally node permutations or the S_n group). Yet, substantial challenges bar the simple formulation of a neural system as a graph. First, neural systems can become extremely large resulting in high computational requirements. Second, imaging techniques measure neural activities at varying levels of granularity. For instance, techniques such as EEG or fMRI measurements measure population level dynamics where the structure of interactions is not explicitly known. Finally, physical connections between neurons (the connectome) have only been mapped for few organisms, and the relationship between functional and physical connectivity remains tenuous. With these challenges in mind, we consider whether or not a graph can serve as a favourable inductive bias when machine learning on neural activity measured from C. Elegans worms. Fortunately, the connectome of C. Elegans is relatively small, neuron-level measurements can be readily acquired through calcium imaging, and the physical connectome has been completely mapped in numerous studies. Thus, we hypothesize that designing an equivariant neural network for node permutations may serve as a favourable inductive bias.

Machine learning on non-Euclidean space remains a substantial challenge in machine learning; however, recent developments with graph neural networks have significantly advanced techniques when machine learning on graph-structured data. These graph neural networks combine message passing computational methods with the universal approximator ability of neural networks, circumventing issues with stability which plagued previous message passing algorithms. Specifically, graph neural networks (GNNs) are a class of neural networks that explicitly use graph structure during computations through message passing algorithms where features are passed along edges between nodes and then aggregated for each node (Scarselli et al. (2009); Gilmer et al. (2017); Battaglia et al. (2018)). To demonstrate the equivariance of GNNs to node permutations, Kipf and Welling (2016) proved that one-hop message passing approximates spectral convolutions on graphs. Their work prompted a resurgence of interest that have led to numerous breakthroughs in studying the representational power of GNNs. Most notably, Xu et al. (2018) showed that the discriminative power of GNNs can match the lauded Weisfeiler-Lehman isomorphism test, potentially outperforming other machine learning techniques for graphs. In comparison, Dehmamy et al. (2019) demonstrated that GNNs are extremely sensitive to design choices when learning graph moments. Nevertheless, from an applied perspective, GNNs have been widely successful in a wide variety of domains including relational inference (Kipf et al. (2018); Lowe et al. (2020); Raposo et al. (2017)), node classification (Kipf and Welling (2016); Hamilton et al. (2017)), point cloud segmentation (Wang et al., 2019), and traffic forecasting (Yu et al. (2018); Li et al. (2018)).

While we use the physical connectome in some of our experiments, we primarily focus on inferred connectivity computed through relational inference to design the structure of our graph. In particular, relational inference remains a longstanding challenge with early works in neuroscience seeking to quantify correlations between neurons (Granger (1969)). Modern approaches to relational inference employ graph neural networks as their explicit reliance on graph structure forms a relational inductive bias (Battaglia et al. (2016); Battaglia et al. (2018)). In particular, our model is inspired by the Neural Relational Inference model (NRI) which uses a variational autoencoder for generating edges and a decoder for predicting trajectories of each object in a system (Kipf et al. (2018)). By inferring edges, the NRI model explicitly captures interactions between objects and leverages the resulting graph as an inductive bias for various machine learning tasks. This model was successfully used to predict the trajectories of coupled Kuramoto oscillators, particles connected by springs, the pick and roll play from basketball, and motion capture visualizations. Subsequently, the authors developed Amortized Causal Discovery, a framework based on the NRI model which infers causal relations from time-dependent data (Lowe et al. (2020)).

Aside from our work, several studies have successfully applied GNNs on various tasks such as annotating cognitive state (Zhang and Bellec (2019)), and several frameworks based on graph neural networks have been proposed for analyzing fMRI data (Li and Duncan (2020); Kim and Ye (2020)).

1.4. Acknowledgements

Chapter 1, in full, is a reprint of the material as it appears in “Generalizable Machine Learning in Neuroscience using Graph Neural Networks” in Frontiers in Artificial Intelligence 2021. Wang, Paul Y.; Sapra, Sandalika; George, Vivek K.; Silva, Gabriel A., Frontiers Media, 2021. The thesis author was the primary investigator and author of this paper.

2. Enabling Generalizable Machine Learning through Graph Neural Networks

2.1. Introduction

In this work, we examine the performance and generalizability of deep learning models applied to the neural activity of C. elegans (round worm/nematode). In particular, C. elegans is a canonical species for investigating microscopic neural dynamics because it remains the only organism whose connectome (the mapping of all 302 neurons and their synaptic connections) is completely known and well-studied (White et al. (1986); Bargmann and Marder (2013); Varshney et al. (2011); Cook et al. (2019)). Furthermore, the transparent body of these worms allows for calcium imaging of whole brain neural activity which remains the only imaging technique capable of spatially resolving the dynamics of individual neurons (Wen and Kimura, 2020). Leveraging these characteristics and insight gained from previous studies, we developed deep learning models that bridge recent advances in neuroscience and deep learning. Specifically, we first demonstrate state-of-the-art performance for classifying motor action states, e.g. forward and reverse crawling, of C. elegans from calcium imaging data acquired in previous works. Next, we examine the generalization performance of our deep learning models on unseen worms both within the same study and in worms from a separate study published years later. We then show that graph neural networks exhibit a favourable inductive bias for analyzing both higher-order function and microscopic/neuron-level dynamics in C. elegans.

2.2. Model

In this section, we first present the general framework of our behavioral state classification and trajectory prediction models. Next, we detail the implementation of our neural network models.

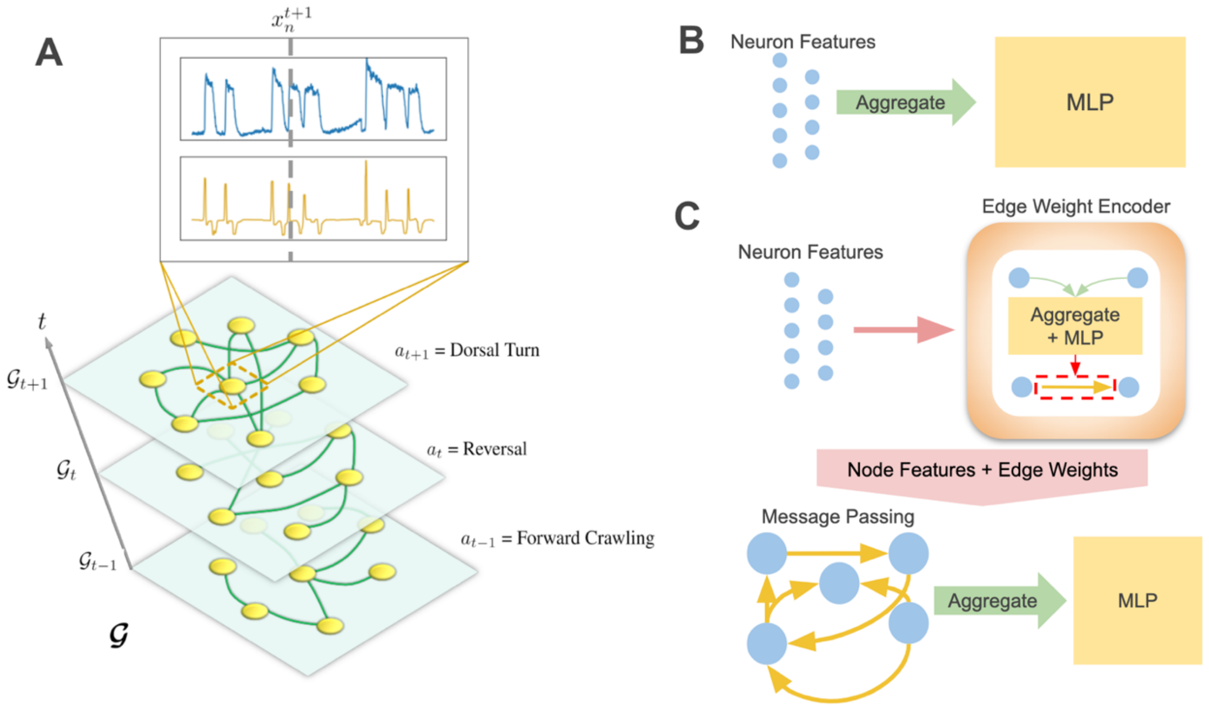


Figure 2.1: (A) Visualization of temporal graph. Inset shows x_n plotted against t where the top is the calcium trace, and the bottom is its derivative. The dashed line intercepts the feature vectors at $t' = t + 1$ and denotes x_n^{t+1}. (B) and (C) are simplified visualizations of the MLP and GNN models respectively.

2.2.1. Framework

We define the set of trajectories (calcium imaging traces) for each worm as $\mathbf{X}_\alpha = \{\mathbf{x}_1, \ldots, \mathbf{x}_\mathrm{n}, \ldots, \mathbf{x}_\mathrm{N}\}_\alpha$ where α denotes the label of the individual, n the name of the neuron, N

the total number of neurons, and $\boldsymbol{v}_n$ the feature vector of the neuron. In our case, $\mathbf{x}_{\mathrm{n}} \in \mathbb{R}^{T\times 2}$ corresponds to time-dependent normalized calcium traces and their derivatives for each neuron where T is the total number of timesteps. Likewise, $x_{n,t} \in \mathbb{R}^2$ corresponds to the features of neuron n at timestep t. Finally, the behavioral states of an individual are encoded as $\boldsymbol{a}_{\alpha} = (a_1, \ldots, a_t, \ldots, a_T)_{\alpha}$ where a behavioral state a is assigned for each timestep t.

Separate models were developed for each task: behavioral state classification and trajectory prediction. In both cases, data from a worm α is structured as a temporal graph $\boldsymbol{\mathcal{G}}_{\alpha} = (\mathcal{G}_1, \ldots, \mathcal{G}_t, \ldots, \mathcal{G}_T)_{\alpha}$ (Figure 2.1A) where each timestep is represented by a static graph whose nodes correspond to neurons. Following the notation above for worm α, the trajectories of each neuron's calcium traces are encoded as node features $\boldsymbol{x}_n$, and the behavioral state of the worm is interpreted as a graph feature a_t. For behavioral state classification, our model consists of the following:

$$\boldsymbol{H}_{\alpha,t} = f(\boldsymbol{X}_{\alpha,t}) \tag{1}$$

$$\boldsymbol{p}_{\alpha,t} = softmax(\boldsymbol{H}_{\alpha,t}) \tag{2}$$

$$\hat{\mathrm{a}}_{\alpha,\mathrm{t}} = \mathrm{argmax}(\boldsymbol{p}_{\alpha,\mathrm{t}}) \tag{3}$$

where $\boldsymbol{X}_{\alpha,t}$ corresponds to node feature vectors for worm α at timestep t, f is an universal approximator/neural network model (described in the next section), $\boldsymbol{H}_{\boldsymbol{\alpha},\boldsymbol{t}} \in \mathbb{R}^k$ corresponds to embedded features, $\boldsymbol{p}_{\alpha,t}$ is the probability that the worm is in one of k motor states (Figure 2.2D), and $\hat{a}_{\alpha,t}$ is the most probable/predicted state.

For trajectory prediction, we developed a Markovian model for inferring trajectories of a consecutive timestep:

$$\boldsymbol{H}_{\alpha,t} = f(\boldsymbol{X}_{\alpha,t}) \tag{4}$$

$$\widehat{\boldsymbol{X}}_{\alpha,\mathrm{t}+1} = \boldsymbol{X}_{\alpha,t} + \boldsymbol{H}_{\alpha,t} \tag{5}$$

where f is the same as before, $\boldsymbol{H}_{\boldsymbol{\alpha},\boldsymbol{t}}$ is the predicted change of the trajectory and can be interpreted as $\Delta\widehat{\boldsymbol{X}}_{\alpha,\mathrm{t}}$, and $\widehat{\boldsymbol{X}}_{\alpha,\mathrm{t}+1}$ is the predicted value of the subsequent timestep. When predicting multiple timesteps, the predicted value of the previous timestep is substituted for $\boldsymbol{X}_{\alpha,t}$. We also experimented with non-Markovian models like recurrent neural networks (RNNs) for which a hidden state is included for each timestep.

The structure of our framework allows us to substitute various models for f. While we include results from several neural networks, we focus on two representative models: a multi-layer perceptron (MLP) agnostic to graph structure (Figure 2.1B) and a graph neural network (GNN) which explicitly computes on an inferred graph (Figure 2.1C).

2.2.2. Neural Network Models $\boldsymbol{f}$: MLP and GNN

Our MLP model aggregates (sums or concatenates) the features of a graph and feeds the aggregated features into a 2-layer MLP neural network:

$$\boldsymbol{H}_{out} = g_{graph:mlp}\big(aggregation(x_1, \dots, x_n, \dots, x_N)\big) \tag{6}$$

where $g_{graph:mlp}$ is a 2-layer MLP. Contrasting the MLP model, our GNN relies on message passing between connected nodes and contains an encoder for edge weights A_{ij}:

$$\boldsymbol{V} = g_{node}(\boldsymbol{X}) \tag{7}$$

$$E_{ij} = g_{edge}\left(aggregation\big(\boldsymbol{v}_i, \boldsymbol{v}_j\big)\right) \tag{8}$$

$$A_{ij} = sigmoid\big(E_{ij}\big) \tag{9}$$

where in equation (7), $\boldsymbol{V} = (\boldsymbol{v}_1, \dots, \boldsymbol{v}_n, \dots, \boldsymbol{v}_N)$ corresponds to the embedding of each node's features through the MLP g_{node}. Next, the edge embedding E_{ij} is computed by aggregating all pairs of node embeddings followed by the MLP g_{edge}. Finally, applying the sigmoid function to the edge embedding E_{ij} produces edge weights A_{ij} normalized between 0 and 1. $\boldsymbol{A}$ can be interpreted as an inferred weighted adjacency matrix where A_{ij} denotes the edge weight between nodes i and j such that $i = j$ denotes a self-edge. The edge weights either dynamically change in each timestep's inferred graph $\mathcal{G}_t$ or remain fixed for the whole temporal graph $\boldsymbol{\mathcal{G}}_\alpha$ of an individual worm. If the edges are static for the temporal graph, the aggregation step in equation (8) also averages hidden features across all timesteps such that $\boldsymbol{V} = \frac{1}{T}\sum_{t=1}^{T} g_{node}(\boldsymbol{X}_t)$. Note that in this case, the edge encoder is given all timesteps $\boldsymbol{X}_\alpha$ in equation (1) and equation (4) instead of just one timestep.

After edges are encoded, the GNN performs a message passing (10) and aggregation step (11):

$$\boldsymbol{M} = \boldsymbol{AX} \tag{10}$$

$$\boldsymbol{H}_{out} = g_{graph:gnn}\big(aggregation(\boldsymbol{M})\big) \tag{11}$$

As mentioned before, our MLP and GNN models can be substituted for f in equation (1) and equation (4). Depending on the task, the dimension of H_{out} for the MLP (6) and GNN (11) models differs. For behavioral state classification, $\boldsymbol{H}_{out} \in \mathbb{R}^k$ whereas for trajectory prediction, $dim(\boldsymbol{H}_{out}) = dim\big(\boldsymbol{X}_{\alpha,t}\big)$ such that $\boldsymbol{H}_{out} \in \mathbb{R}^{N\times 2}$.

Theoretically, an arbitrary number of message passing steps can be implemented; however, we did not find any improvements when using more than one step. In addition, we find that performance improves when using concatenation instead of summation during the aggregation step.

2.3. Experiments

Our experiments were performed with data acquired in Kato et al. (2015) and Nichols et al. (2017). We summarize various details about the data in this section; however, we direct the reader to each respective publication for specific experimental details.

2.3.1. Calcium Imaging

Kato et al. (2015) showed that neural activity corresponding to the motor action sequence lives on low dimensional manifolds. To record neuron level dynamics, they performed whole-brain genetically encoded Ca^{2+} imaging with single-cell-resolution and measured ~100 neurons for around 18 minutes. They then normalized each calcium trace by peak fluorescence and identified neurons using spatial position and previous literature (Altun et al., 2002–2020). Aside from imaging freely moving worms, the authors also examined robustness of topological features to sensory stimuli changes, hub neuron silencing, and immobilization. For simplicity, we limited our experiments to data collected on freely moving worms.

Nichols et al. (2017) focused on differences in neural activity of C. elegans while awake or asleep and studied two different strains of worms, n2 (11 total worms) and npr1 (10 total worms). Because experiments in both studies were performed by the same group, most experimental procedures were similar, allowing us to easily process data to match the Kato dataset. While this dataset includes imaging data of each worm during quiescence, for consistency with the Kato dataset, we only included data before sleep was induced. Furthermore, we pooled results for both strains of worms as we did not notice any statistically relevant differences between them.

2.3.2. Dataset Enlargement

Although our data for each worm is relatively small (~3000-4000 timesteps), our datasets contained calcium traces from numerous worms. In total, 5 worms were measured in Kato et al. (2015) and 21 worms were measured in Nichols et al. (2017). Taking advantage of the large number of worms measured, we experimented with dataset enlargement where our models were trained on pooled data from different numbers of worms in the Kato dataset. Similarly, we pooled data from all 21 worms from the Nichols dataset; however, we use this dataset only during evaluation- i.e. the model never sees this dataset in training. In this way, we define the "seen" population as worms whose data was seen in training and the "unseen" population as worms the model did not see during training. More details about how datasets were used in our experiments can be found in the following section.

To perform dataset enlargement, we separately trained the models on each worm in the seen population for each epoch. In other words, we independently optimized the loss function for each worm in every epoch. We followed this procedure such that batch normalization was separately performed on each worm's features. This technique was motivated by experiments where batch normalization on data from individual worms improved both test set and generalization accuracy. In contrast, performing batch normalization on pooled data from all worms greatly decreased model performance.

2.3.3. Data Processing

We normalized the calcium trace of each neuron and its derivative to [0,1]. Normalization was performed for the entire recorded calcium trace of a worm instead of within each batch because the relative magnitudes of the traces have been found to contain graded information about the worm's behavioral state (e.g. crawling speed).

For the seen population, we separated each calcium trace of approximately 3000-4000 timesteps into batches of 8 timesteps where each timestep corresponds to roughly 1/3 of a second. We chose batch sizes of 8 timesteps because visualization of calcium traces showed that most local variations occur within this time frame. Moreover, 8 timesteps roughly corresponds to 3 seconds which is about the amount of time a worm needs to execute a behavioral change. Finally, the batches were shuffled before being divided into 10 folds later used for cross-validation, ensuring that each fold is representative across the whole dataset.

When evaluating on the unseen population, we treat the data differently for each task. For behavioral classification, we infer the behavioral state of the system using data from one timestep. As such, we do not split the data and simply run the model separately on each timestep of the worm's calcium traces. In contrast, for trajectory prediction, we split the calcium traces into batches of 16 timesteps and evaluate the model on all batches.

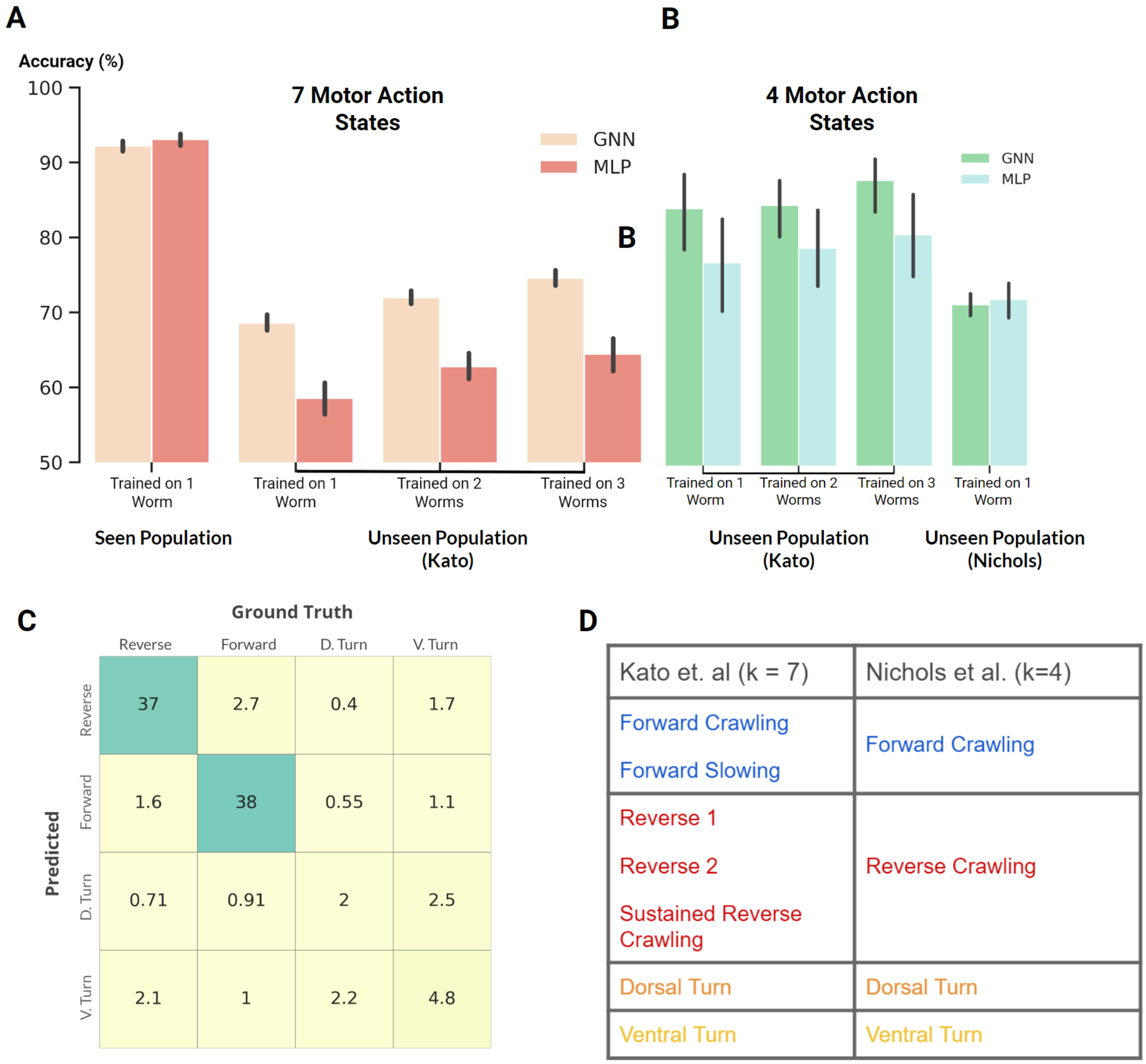
To compare with previous works, we performed our experiments on uniquely identified neurons between the datasets that we investigated. Identifying specific neurons is an experimental challenge, and as such, only a small fraction of neurons were unequivocally labeled. A total of 15 neurons were uniquely identified between all 5 worms measured in the Kato dataset: (AIBL, AIBR, ALA, AVAL, AVAR, AVBL, AVER, RID, RIML, RIMR, RMED, RMEL, RMER, VB01, VB02). In addition, the Nichols dataset contained data from 21 worms with 3 uniquely identified neurons shared among all worms in both datasets: (AIBR, AVAL, VB02).

2.4. Results

Following Brennan and Proekt (2019), we used data from Kato et al. (2015) for training/evaluating our models and data from Nichols et al. (2017) as an extended evaluation set. Because whole brain imaging is incredibly difficult, our datasets were relatively small. To address this, we experimented with dataset enlargement by combining data from multiple worms in the Kato dataset during model training. For all experiments, we performed 10-fold cross validation on all permutations of worms in our training set. More details, along with supplemental experiments, can be found in Appendix A.

Table 2.1: Classification Accuracy of Forward and Reverse Crawling

	Seen Population	Unseen Population (Kato)	Unseen Population (Nichols)
Brennan and Proekt (2019)	83	81	----
SVM	98.8 ± .4	82.8 ± 7.6	79.0 ± 11.7
MLP	99.3 ± .6	93.9 ± 10.3	88.9 ± 11.4
GNN (Connectome)	99.5 ± .6	96.8 ± 4.3	85.5 ± 12.9
GNN	**99.5 ± .5**	**97.7 ± 3.1**	**95.5 ± 6.4**



Kato et. al (k = 7)	Nichols et al. (k=4)
Forward Crawling Forward Slowing	Forward Crawling
Reverse 1 Reverse 2 Sustained Reverse Crawling	Reverse Crawling
Dorsal Turn	Dorsal Turn
Ventral Turn	Ventral Turn

Figure 2.2: (A; B) Classification accuracy of our GNN and MLP models where black vertical lines show statistical spread. (A): Classification of 7 motor action states within the Kato dataset. (B): Classification of 4 motor action states on both the Kato and Nichols datasets. (C) Confusion matrix. Percent occurrence of predicted states against labeled states when evaluating on the Nichols dataset. (D) Mapping of behavioral states between the Kato and Nichols dataset.

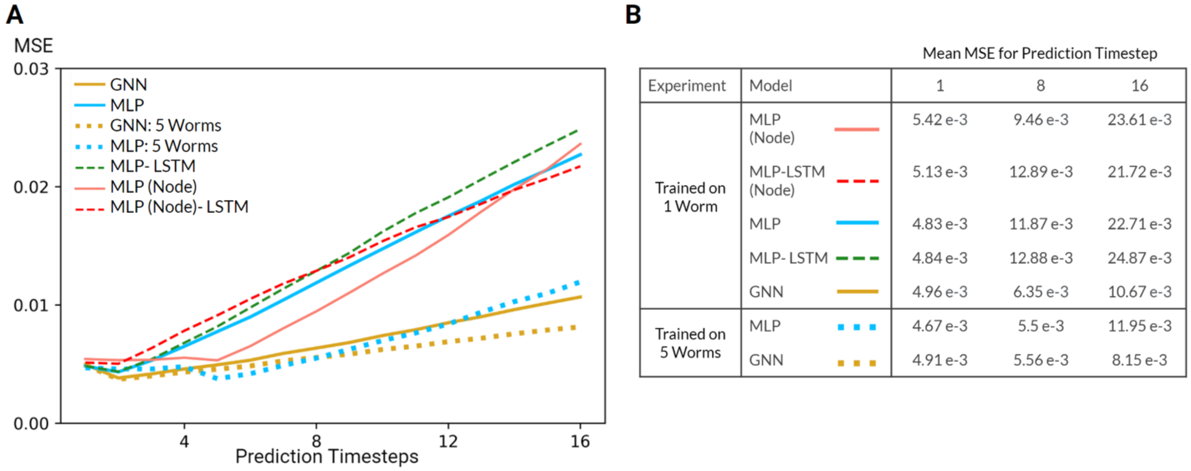
2.4.1. Behavioral State Classification

Our first experiment compared the performance of our models to state-of-the-art results reported in Brennan and Proekt (2019). Specifically, this experiment involved the classification of only two motor action states, forward and reverse crawling. Along with our models described above, we also experimented with a support vector machine (SVM) and a GNN which computes with edges derived from the physical connectome (White et al., 1986). In particular, we incorporated the connectome into our model to investigate whether physical/structural connections between neurons can serve as a favourable inductive bias for our GNN. Our results are shown in Table 2.1 where “Seen Population” denotes test set accuracy after training on the same worm and “Unseen Population” denotes evaluation/generalization accuracy on worms unseen during training.

Our deep learning models clearly outperformed the SVM and state-of-the-art results, demonstrating the ability of our models to successfully classify behavioral states and generalize to other worms. Interestingly, the SVM matched the performance of our deep learning models on the seen population; however, its generalization performance on unseen individuals was significantly worse than our deep learning models. As such, the SVM distinctly illustrates challenges of individual variability for model development in neural systems despite the simplicity of our experiments which involve the same set of unequivocally identified neurons. Similarly, our GNN using edges derived from the connectome performed well on the seen population but generalized worse than when using inferred edges. We hypothesize that the detrimental effect of using the connectome may be attributed to the distinction between inferred/functional and structural connectivity. In particular, the connectome maps physical connections between neurons which is generally conserved between different individuals. In contrast, individual variability of neural activity implicitly implies that the inferred/functional connectivity is unique to individuals (See A.4.3).

Following the previous experiment, we applied our MLP and GNN models to the harder task of classifying all behavioral states labeled in the Kato dataset (Figure 2.2A). Within this dataset, 7 states were labeled: Forward Crawling, Forward Slowing, Reverse 1, Reverse 2, Sustained Reverse Crawling, Dorsal Turn, and Ventral Turn. In comparison to the Kato dataset, only 4 states were labeled in the Nichols dataset: reverse crawling, forward crawling, ventral turn, and dorsal turn. For compatibility, we mapped the 7 states of the Kato dataset to 4 states of the Nichols dataset when using the Nichols dataset as an extended evaluation set (Figure 2.2D).

Despite the harder task of classifying 7 states, our models achieved a classification accuracy of ~92% on the same worm (Figure 4A). Moreover, our GNN trained on three worms in the Kato dataset generalized with an accuracy of 87% (Figure 2.2B) when classifying 4 states on the remaining unseen worms. This substantially exceeds the performance of our MLP model and Brennan and Proekt (2019) who report an 81% cross-animal accuracy on two states. Nevertheless, both MLP and GNN models generalized equally well (~70%) to the 21 unseen worms of the Nichols dataset. These experiments consistently demonstrate that our GNN exceeds the performance of state-of-the-art techniques and also often exceeds the performance of our baseline MLP model.



Experiment	Model		Mean MSE for Prediction Timestep 1	8	16
Trained on 1 Worm	MLP (Node)	—	5.42 e-3	9.46 e-3	23.61 e-3
	MLP-LSTM (Node)	- - -	5.13 e-3	12.89 e-3	21.72 e-3
	MLP	—	4.83 e-3	11.87 e-3	22.71 e-3
	MLP- LSTM	- - -	4.84 e-3	12.88 e-3	24.87 e-3
	GNN	—	4.96 e-3	6.35 e-3	10.67 e-3
Trained on 5 Worms	MLP	▪ ▪ ▪	4.67 e-3	5.5 e-3	11.95 e-3
	GNN	▪ ▪ ▪	4.91 e-3	5.56 e-3	8.15 e-3

Figure 2.3: (A) Mean squared error (MSE) of the GNN and various MLP models evaluated on the Nichols dataset. All models were trained using data from one worm or five worms in the Kato Dataset. (B) Table of mean MSE values for all models for 1, 8, and 16 timesteps.

2.4.2. Neuron-Level Trajectory Prediction

For trajectory prediction, we predicted each neuron's calcium trace and its derivative (normalized to [0,1]) for 8 timesteps during training (seen population) and 16 timesteps during evaluation/validation (unseen population). While training our Markovian models, scheduled sampling was performed to minimize the accumulation of error (Bengio et al., 2015). When evaluating on the unseen population, the model was given one timestep as the initial condition after which the model predicts 16 timesteps. In addition to our Markovian models, we also experimented with RNN implementations trained with burn-in periods of four timesteps (12 timesteps during training and 20 timesteps during evaluation). Our experiments primarily focused on generalization performance of our models on the extended evaluation/Nichols dataset (Figure 2.3).

Predicting neuron-level trajectory using deep learning is fairly novel since advances in whole-brain imaging are recent and limited to few organisms. Nevertheless, neural systems generically fall under the category of dynamical systems where each neuron is described by a differential equation such that neural activity can be modeled as a system of coupled differential equations. Under this formulation, the task of trajectory prediction involves learning the underlying physical laws in order to predict the time evolution of the system. To quantify the predictive power of our models, we evaluated the mean squared error (MSE) of each prediction timestep relative to the true trajectory. In the context of our Markovian model, this metric measures the error of the predicted transition matrix which time evolves the state of the system and, by extension, demonstrates the ability of our models to learn the underlying physical laws of the dynamical system.

Several challenges limited the predictive power of our models. Most prominently, our system is inherently non-linear and potentially chaotic, a fact further exacerbated by the nature of calcium imaging which is notoriously noisy and an indirect measurement of neural activity. In addition, our datasets are relatively small in spite of our dataset enlargement technique. Resulting from these challenges, the performance of our model is poor, especially in comparison to that of models in data assimilation which leverage apriori knowledge of the dynamical system (Meliza et al. (2014); Moye and Diekman (2018)). Nevertheless, inspecting the MSE as a function of prediction step (Figure 2.3) reveals that our models are able to learn how the system transitions up to a short timescale. Moreover, increasing the number of worms included during training (dataset enlargement) also improved the generalization performance of our MLP and GNN models. Perhaps most surprising, our Markovian GNN outperformed all MLP models and their derived RNN variants. We attribute this result to the largely deterministic nature of neural dynamics, characterized by sparse bifurcations on the latent manifold, and the inductive bias of GNNs. As a result, given 1 timestep, our GNN outperformed all other models including RNN variants which were given 4 burn-in timesteps. Therefore, we conclude that our GNN displays a favorable inductive bias in contrast to graph-agnostic models for the task of predicting microscopic dynamics.

2.5. Discussion

For both tasks, our GNN consistently matched or exceeded our MLP model which we accredit to its favourable inductive bias. Kato et al. (2015) established that projecting neural dynamics onto three principal components for each worm reveals universal topological structures; however, attempts to project neural dynamics onto shared principal components of all worms failed to display any meaningful structure. Thus, variability in each worm's neural activity, corresponding to low dimensional manifolds in latent space, is represented by different linear combinations of neurons. In other words, relevant topological structures in latent space are loosely related by linear transformations of node features. We speculate that our GNN's performance stems from its explicit structure of message passing along inferred edges which is analogous to learning linear transformations of node features (see equation (10)). Based on our experimental results, we further speculate that this inductive bias proves favourable on both microscopic and macroscopic machine learning tasks in neural systems.

Interestingly, our model's performance was not significantly impacted by using 3 neurons (~1% of all neurons) instead of 15 (~5% of all neurons). This is not surprising because neurons strongly coupled to the motor action sequence retain most information (Gao and Ganguli, 2015), a fact consistent with Brennan and Proekt (2019) who found that strategically choosing 1 neuron retains~75% of the information contained in the larger set of 15 neurons.

Finally, as a critical question, we ask whether our model's performance stems from choosing a stereotyped organism that is well studied and biologically simple, or if our results imply a path towards generalizable/universal machine learning in neural systems. While the neurophysiology of C. elegans is quite complex, the motor action sequence we studied is relatively simple, especially in comparison toother organisms and cognitive functions. Moreover, organisms are adaptive and capable of learning new behavior, a fact not represented in our dataset. However, a recent astounding study (Gallego et al. 2020) measured neural dynamics in monkeys trained to perform action sequences and determined that learned latent dynamics live in low-dimensional manifolds that were conserved throughout the length of the study. By aligning latent dynamics, their model accurately decoded the action of monkeys up to two years after the model was trained despite changes in biology (e.g. neuron turnover, adaptation to implants). Consequently, we posit that techniques similar to those used in our model may broadly apply to more complex organisms and functions.

2.6. Conclusion

In this study, we examined the ability of neural networks to classify higher-order function and predict neuron level dynamics. In addition, inspired by global organizational principles of behavior discovered in previous studies, we demonstrated the ability of neural networks to generalize to unseen organisms. Specifically, we first showed that our models exceed the performance of previous studies in behavioral state classification of C. elegans. Next, we found that a simple MLP performs remarkably well on unseen organisms. Nevertheless, our graph neural network, which explicitly learns linear transformations of node features, matched or exceeded the performance of graph agnostic models in all experiments. These experiments demonstrate that our models are capable of successful evaluation on unseen organisms, both within the same study, and in a separate experiment spaced years apart. Finally, our results show that dataset enlargement through the inclusion of more individuals can significantly improve generalization performance in microscopic neural systems.

We note that our results of generalization on both higher-order functions and neuron-level dynamics (macroscopic and microscopic) suggests wide applicability of our technique to numerous machine learning tasks in neuroscience and hierarchical dynamical systems. A promising research direction is the hierarchical relationship between neuron-level and population-level dynamics. Breakthroughs in this direction may inform machine learning models working with population-level functional and imaging techniques, such as EEG or fMRI, which are readily available and widespread. In addition, in this study, we only focused on simple machine learning tasks and imaging data taken under similar experimental conditions. Further studies using GNNs may involve more complex tasks such as those involving graded information in neural dynamics, changes in sensory stimuli, acquisition of learned behaviors, and higher-order functions comprised of complicated sequences of behavior. From a machine learning perspective, the development of a recurrent graph neural network for the edge encoder with a suitable attention mechanism may aid model generalization. Additional work is also needed in examining and improving model performance on arbitrary sets of neurons as neuron identification is experimentally challenging and limited to small systems.

2.7. Acknowledgements

Chapter 2, in full, is a reprint of the material as it appears in “Generalizable Machine Learning in Neuroscience using Graph Neural Networks” in Frontiers in Artificial Intelligence 2021. Wang, Paul Y.; Sapra, Sandalika; George, Vivek K.; Silva, Gabriel A., Frontiers Media, 2021. The thesis author was the primary investigator and author of this paper.

Appendix A. Supplemental for Chapter 2

A.1 Model Selection

The two final models included in the main text were chosen for their performance and simplicity. Nevertheless, we experimented with numerous established models which were easily substituted for f. For GNNs, we primarily used the excellent Pytorch Geometric library (Fey and Lenssen, 2019). Tested modules include the GIN-0/GIN-ϵ (Xu et al., 2018), Graph Sage (Hamilton et al., 2017), GAT (Velickovic et al., 2018), and Global Attention (Li et al., 2015). In particular, we expected the GIN to outperform the other modules because its expressiveness has been shown to aid transfer learning (Hu et al., 2020); however, because our edges are not explicitly known, we essentially applied the GIN on a fully connected graph. Under this formulation, the GIN-0 simply symmetrizes node features after a message passing step which is similar to the aggregation step of our MLP. We also found that the GIN-was prone to overfitting. Finally, we tested the GAT which is similar to our model when edges are dynamically inferred each timestep. As a result, we found that the GAT performs equally well on trajectory prediction but performs slightly worse on behavioral state classification.

A.2 Model Implementation

A.2.1 Neural Networks

The two-layer MLP corresponding to g in the main text comprised of linear layers followed by elu activation functions. We also applied batch norm on the output of the two layers. The Node

MLP in the main text refers to individual MLPs for each node. To construct RNN variants, we added an LTSM unit before the MLP. We performed some minor hyperparameter optimization as our combinatorial cross-validation was computationally expensive. Overall, we found our models relatively robust to different hyperparameters. For trajectory prediction, we used hidden layers with 256 dimensions. On the other hand, for behavioral state classification, we used hidden layers with 16 dimensions. Furthermore, we determined that dynamic edges evaluation worked better for trajectory prediction; however, globally evaluated edges for each worm resulted in better performance for behavioral state classification. Finally, for trajectory prediction, we chose to optimize the mean square error (MSE). For behavioral state classification, we optimize the negative loglikelihood (NLL).

A.2.2 Support Vector Machine

For the SVM, we used a linear SVM module from sci-kit learn (Linear SVC). Although SVMs with linear kernels are significantly less expressive than that with non-linear kernels, we chose a linear kernel as our test set accuracy implies that behavioral states are linearly separable. The SVM was trained with the same loss function until a tolerance of 1E-5 was achieved.

A.3 Experimental Procedures

For the extended evaluation set, we chose prelethargus data where 4 states were labeled: reverse, forward, dorsal turn, and ventral turn. For compatibility with the training dataset, we mapped reverse 1, reverse 2, and sustained reverse crawling to the reverse state. Similarly, we mapped forward crawling and forward slowing to forward. In addition to the 7 or 4 labeled states, there was another labeled state for unknown behavior or quiescence. This state comprised a very small portion of our data, and during training and evaluation, we ignore the result when the target is unknown. For all experiments in the main text, we perform 10-fold cross validation on all possible permutations of worms in our training set (Kato dataset). For example, on our experiments trained on two worms, the possible permutations of worms are the following: {(1, 2), (1, 3), (1, 4), (1, 5), (2, 3), (2, 4), (2, 5), (3, 4),(3, 5), (4, 5)}. Experiments labeled with "Train on 2 worms" involved models trained separately on each of these permutations. Each permutation then involved 10-fold cross validation where the test set was left out when performing hyperparameter optimization. In particular, for our experiments on behavioral state classification, we used 1 fold as the test/ "leave-out" set and 1 fold for the validation set which was used for optimization and as a metric for stopping training. On the other hand, our experiments on trajectory prediction were focused primarily on generalization performance instead of test set accuracy so we used 1fold as the validation set and evaluated on all worms in the extended validation set (Nichols dataset). As a note, we also attempted experiments where data from the extended dataset was used as a validation set. Under this condition, we found that the MLP performed significantly better; however, we were concerned that the MLP was overfitting to the validation set so we chose not to include those results. We performed our experiments on with an Intel i9 9900k CPU and Nvidia GeForce RTX 2080Ti graphics card. Since our models are relatively simple, we were able to train the model on data from one worm in one batch. Nevertheless, the number of worms and cross-validation procedure was very computationally expensive. As such, training and evaluating each model required roughly a week or two of continuous computation. For optimization, we used the Adams optimizer with a learning rate of10^{-3}. We decayed the learning rate with by a factor of .25 if the loss did not improve after 50 epochs. We then trained for 800epoch and saved the model with the lowest validation loss. For scheduled sampling (used during trajectory prediction), we adopted a linear decay which terminated at 300 timesteps.

A.4 Additional Experiments

We performed numerous experiments to verify our results and examine the performance of our model on diverse machine learning tasks. We did not perform rigorous cross validation for the following experiments.

A.4.1 Experiments without AVA

Referees of Brennan and Proekt (2019) were concerned with behavioral state classification where AVA neurons were included. In particular, these neurons were used by Kato et al. (2015) to define behavioral state through trajectory clustering in latent space. Referees commented that classifying behavioral states with neurons used to define those states was akin to circular reasoning. We would like to note that Kato et al. (2015) verified their assigned behavioral states through recorded videos, minimizing risks that assigned behavioural states differ from reality. Nevertheless, we followed Brennan and Proekt (2019) and performed an experiment excluding AVA neurons in which we found no noticeable difference in model performance.

A.4.2 One-hot encoding of edges

To enforce a sparsity on the edges, we experimented with one-hot encoding by adding a scaling factor within the softmax. We found that our GNN achieved similar test accuracies as in the main text. However, our GNN failed to generalize well to unseen worms. Following our discussion in the main text, we believe that one-hot encoding was detrimental to generalization because it effectively results in a permutation matrix which simply permutes node features. This is counter to previous studies where topological structures are related by more general linear transformations.

A.4.3 Comparison of inferred edges to known connectome

Inferring the connectivity between neurons in neural systems remains a key challenge in neuroscience. Because C. Elegans is among few organisms whose connectome mostly or completely known, we decided to compare the inferred edges of our model to the connectome of C. Elegans. Ultimately, we found no similarities between our inferred edges and the connectome. In neuroscience, two types of connectivity are defined: structural and functional. Structural connectivity refers to physical connections between neurons whereas functional/effective connectivity corresponds to observed connections (Horwitz, 2003). The exact methods for determining either metrics remains heavily contested. Regardless, in the context of C. Elegans, each worm generally has the same structural connectivity; however, differences in neural activity implies a different functional connectivity exists for unique individuals. Since the connectome relates to the structural connectivity, we believe that our inferred edges are a poor proxy for the connectome. On a more abstract level, our graph neural network works with a subset of neurons such that a inferred edge may not correspond to a direct correlation, but may rather represent higher order correlations with unseen neurons.

A.5 Acknowledgements

Appendix A, in full, is a reprint of the supplemental material for “Generalizable Machine Learning in Neuroscience using Graph Neural Networks” in Frontiers in Artificial Intelligence 2021. Wang, Paul Y.; Sapra, Sandalika; George, Vivek K.; Silva, Gabriel A., Frontiers Media, 2021. The thesis author was author of this supplemental material section.

References

[Dataset] Altun, Z. F., Herndon, L. A., Wolkow, C. A., Crocker, C., Lints, R., and Hall, D. H. (2002-2020). Worm atlas

Bargmann, C. I. and Marder, E. (2013). From the connectome to brain function. Nature methods10, 483

Bashivan, P., Rish, I., Yeasin, M., and Codella, N. (2015). Learning representations from eeg with deep recurrent-convolutional neural networks. arXiv preprint arXiv:1511.06448https://arxiv.org/abs/1511.06448

Battaglia, P.W., Hamrick, J.B., Bapst, V., Sanchez-Gonzalez, A., Zambaldi, V., Malinowski, M., Tacchetti, A., Raposo, D., Santoro, A., Faulkner, R. and Gulcehre, C. (2018). Relational inductive biases, deep learning, and graph networks. arXiv https://arxiv.org/pdf/1806.01261.pdf

Battaglia, P. W., Pascanu, R., Lai, M., Rezende, D., & Kavukcuoglu, K. (2016). Interaction networks for learning about objects, relations and physics. In Advances in neural information processing systems. 4502–4510

Bengio, S., Vinyals, O., Jaitly, N., and Shazeer, N. (2015). Scheduled sampling for sequence prediction with recurrent neural networks. In Advances in Neural Information Processing Systems. 1171–1179

Brennan, C. and Proekt, A. (2019). A quantitative model of conserved macroscopic dynamics predicts future motor commands. Elife8, e46814

Brown, C. J. and Hamarneh, G. (2016). Machine learning on human connectome data from mri. arXiv preprint arXiv:1611.08699https://arxiv.org/abs/1611.08699

Brown, C. J., Kawahara, J., and Hamarneh, G. (2018). Connectome priors in deep neural networks to predict autism. In2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018) (IEEE),110–113

Churchland, M. M., Cunningham, J. P., Kaufman, M. T., Ryu, S. I., and Shenoy, K. V. (2010). Cortical preparatory activity: representation of movement or first cog in a dynamical machine? Neuron, 68,387–400

Cohen, T. and Welling, M. (2016). Group equivariant convolutional networks. In International conference on machine learning. 2990–2999

Cohen, T. S., Geiger, M., Köhler, J., & Welling, M. (2018). Spherical cnns. arXiv preprint arXiv:1801.10130.

Cook, S.J., Jarrell, T.A., Brittin, C.A., Wang, Y., Bloniarz, A.E., Yakovlev, M.A., Nguyen, K.C., Tang, L.T.H., Bayer, E.A., Duerr, J.S. and Bülow, H.E. (2019). Whole-animal connectomes of both caenorhabditis elegans sexes. Nature, 571, 63–71 de Haan, P., Weiler, M., Cohen, T., & Welling, M. (2020). Gauge Equivariant Mesh CNNs Anisotropic convolutions on geometric graphs.

Dehmamy, N., Barab´asi, A.-L., and Yu, R. (2019). Understanding the representation power of graph neural networks in learning graph topology. In Advances in Neural Information Processing Systems.15413–15423

Fregnac, Y. (2017). Big data and the industrialization of neuroscience: A safe roadmap for understanding the brain? Science 358, 470–477

Gallego, J. A., Perich, M. G., Chowdhury, R. H., Solla, S. A., and Miller, L. E. (2020). Long-term stability of cortical population dynamics underlying consistent behavior. Nature neuroscience 23, 260–270

Gao, P. and Ganguli, S. (2015). On simplicity and complexity in the brave new world of large-scale neuroscience. Current opinion in neurobiology 32, 148–155

Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O., and Dahl, G. E. (2017). Neural message passing for quantum chemistry. In Proceedings of the 34th International Conference on Machine Learning-Volume70. 1263–1272

Glaser, J. I., Benjamin, A. S., Farhoodi, R., and Kording, K. P. (2019). The roles of supervised machine learning in systems neuroscience. Progress in neurobiology 175, 126–137

Gleeson, P., Lung, D., Grosu, R., Hasani, R., and Larson, S. D. (2018). c302: a multiscale framework for modelling the nervous system of caenorhabditis elegans. Philosophical Transactions of the Royal Society B: Biological Sciences373, 20170379

Goldman, M., Golowasch, J., Marder, E., and Abbott, L. (2001). Global structure, robustness, and modulation of neuronal models. Journal of Neuroscience21, 5229–5238

Golowasch, J., Goldman, M. S., Abbott, L., and Marder, E. (2002). Failure of averaging in the construction of a conductance-based neuron model. Journal of neurophysiology87, 1129–1131

Granger, C. W. (1969). Investigating causal relations by econometric models and cross-spectral methods. Econometrica: journal of the Econometric Society, 424–438

Hamilton, W., Ying, Z., and Leskovec, J. (2017). Inductive representation learning on large graphs. In Advances in neural information processing systems. 1024–1034

Kaplan, H. S., Thula, O. S., Khoss, N., and Zimmer, M. (2020). Nested neuronal dynamics orchestrate a behavioral hierarchy across timescales. Neuron105, 562–576

Kato, S., Kaplan, H.S., Schrödel, T., Skora, S., Lindsay, T.H., Yemini, E., Lockery, S. and Zimmer, M. (2015). Global brain dynamics embed the motor command sequence of caenorhabditis elegans. Cell163, 656–669

Kim, B.-H. and Ye, J. C. (2020). Understanding graph isomorphism network for brain mr functional connectivity analysis. Frontiers in Neuroscience14, 630. doi:10.3389/fnins.2020.00630

Kipf, T., Fetaya, E., Wang, K.-C., Welling, M., and Zemel, R. (2018). Neural relational inference for interacting systems. In International Conference on Machine Learning. 2688–2697

Kipf, T. N. and Welling, M. (2016). Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907

Kwak, N.-S., Muller, K.-R., and Lee, S.-W. (2017). A convolutional neural network for steady state visual evoked potential classification under ambulatory environment. PloS one12, e0172578

Li, X. and Duncan, J. (2020). Brain gnn: Interpretable brain graph neural network for fmri analysis. bioRxiv https://www.biorxiv.org/content/10.1101/2020.05.16.100057v1

Li, Y., Yu, R., Shahabi, C., and Liu, Y. (2018). Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. International Conference on Learning Representations

Lundervold, A. S. and Lundervold, A. (2019). An overview of deep learning in medical imaging focusingon mri. Zeitschrift fur Medizinische Physik29, 102–127

Lowe, S., Madras, D., Zemel, R., and Welling, M. (2020). Amortized causal discovery: Learning to infer causal graphs from time-series data. https://arxiv.org/abs/2006.10833

Makin, J. G., Moses, D. A., and Chang, E. F. (2020). Machine translation of cortical activity to text with an encoder–decoder framework. Tech. rep., Nature Publishing Group

Meliza, C. D., Kostuk, M., Huang, H., Nogaret, A., Margoliash, D., and Abarbanel, H. D. (2014). Estimating parameters and predicting membrane voltages with conductance-based neuron models. Biological cybernetics108, 495–516

Mensch, A., Mairal, J., Bzdok, D., Thirion, B., and Varoquaux, G. (2017). Learning neural representations of human cognition across many fmri studies. In Advances in neural information processing systems.5883–5893

Moye, M. J. and Diekman, C. O. (2018). Data assimilation methods for neuronal state and parameter estimation. The Journal of Mathematical Neuroscience8, 11

Nichols, A. L., Eichler, T., Latham, R., and Zimmer, M. (2017). A global brain state underlies c. elegans sleep behavior.Science 356

Prevedel, R., Yoon, Y.G., Hoffmann, M., Pak, N., Wetzstein, G., Kato, S., Schrödel, T., Raskar, R., Zimmer, M., Boyden, E.S. and Vaziri, A.. (2014). Simultaneous whole-animal 3d imaging of neuronal activity using light-field microscopy. Nature methods11, 727–730

Prinz, A. A., Bucher, D., and Marder, E. (2004). Similar network activity from disparate circuit parameters. Nature neuroscience7, 1345–1352

Raposo, D., Santoro, A., Barrett, D., Pascanu, R., Lillicrap, T., and Battaglia, P. (2017). Discovering objects and their relations from entangled scene representations. International Conference on Learning Representations

Sarma GP, Lee CW, Portegys T, Ghayoomie V, Jacobs T, Alicea B, Cantarelli M, Currie M, Gerkin RC, Gingell S, Gleeson P. (2018). Openworm: overview and recent advances in integrative biological simulation of caenorhabditis elegans. Philosophical Transactions of the Royal Society B373, 20170382

Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., and Monfardini, G. (2009). The graph neural network model. IEEE Transactions on Neural Networks20, 61–80

Skora, S., Mende, F., and Zimmer, M. (2018). Energy scarcity promotes a brain-wide sleep state modulated by insulin signaling in c. elegans. Cell reports22, 953–966

Varshney, L. R., Chen, B. L., Paniagua, E., Hall, D. H., and Chklovskii, D. B. (2011). Structural properties of the caenorhabditis elegans neuronal network. PLoS Comput Biol7, e1001066

Wang, Y., Sun, Y., Liu, Z., Sarma, S. E., Bronstein, M. M., and Solomon, J. M. (2019). Dynamic graph cnn for learning on point clouds. Acm Transactions On Graphics (tog) 38, 1–12

Wen, C. and Kimura, K. D. (2020). How do we know how the brain works?-analyzing whole brain activities with classic mathematical and machine learning methods. Japanese Journal of Applied Physics59, 030501

White, J. G., Southgate, E., Thomson, J. N., and Brenner, S. (1986). The structure of the nervous system of the nematode caenorhabditis elegans. Philos Trans R Soc Lond B Biol Sci314, 1–340

Xu, K., Hu, W., Leskovec, J., and Jegelka, S. (2018). How powerful are graph neural networks? International Conference on Learning Representations

Yu, B., Yin, H., and Zhu, Z. (2018). Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. Proceedings of the 27th International Joint Conference on Artificial Intelligence, 3634–3640

Zhang, X., Yao, L., Wang, X., Monaghan, J., Mcalpine, D., and Zhang, Y. (2019). A survey on deep learning based brain computer interface: Recent advances and new frontiers. arXiv preprint arXiv:1905.04149https://arxiv.org/abs/1905.04149

Zhang, Y. and Bellec, P. (2019). Functional annotation of human cognitive states using graph convolutionnetworks. International conference on machine learning

Fey, M. and Lenssen, J. E. (2019). Fast graph representation learning with PyTorch Geometric. In ICLR Workshop on Representation Learning on Graphs and Manifolds

Horwitz, B. (2003). The elusive concept of brain connectivity. Neuroimage19, 466–470

Li, Y., Tarlow, D., Brockschmidt, M., and Zemel, R. (2015). Gated graph sequence neural networks. arXiv preprint arXiv:1511.05493https://arxiv.org/abs/1511.05493

Hu, W., Liu, B., Gomes, J., Zitnik, M., Liang, P., Pande, V. and Leskovec, J., 2020, April. Strategies For Pre-training Graph Neural Networks. In International Conference on Learning Representations (ICLR).