BIOLOGICAL CONNECTOMES AS A REPRESENTATION FOR THE ARCHITECTURE OF ARTIFICIAL NEURAL NETWORKS

Samuel Schmidgall
George Mason University
sschmidg@gmu.edu

Catherine Schuman
University of Tennessee, Knoxville
cschuman@utk.edu

Maryam Parsa
George Mason University
mparsa@gmu.edu

ABSTRACT

Grand efforts in neuroscience are working toward mapping the connectomes of many new species, including the near completion of the *Drosophila melanogaster*. It is important to ask whether these models could benefit artificial intelligence. In this work we ask two fundamental questions: (1) where and when biological connectomes can provide use in machine learning, (2) which design principles are necessary for extracting a good representation of the connectome. Toward this end, we translate the motor circuit of the *C. Elegans* nematode into artificial neural networks at varying levels of biophysical realism and evaluate the outcome of training these networks on motor and non-motor behavioral tasks. We demonstrate that biophysical realism need not be upheld to attain the advantages of using biological circuits. We also establish that, even if the exact wiring diagram is not retained, the architectural statistics provide a valuable prior. Finally, we show that while the *C. Elegans* locomotion circuit provides a powerful inductive bias on locomotion problems, its structure may hinder performance on tasks unrelated to locomotion such as visual classification problems.

INTRODUCTION

Artificial neural networks (ANNs) are typically designed to have highly general architectures that work well on a wide variety of problems, depending primarily on the fine-tuning of weights across very large amounts of data to solve a given problem. This differs from nature, where much of animal behavior is encoded in the genome and decoded via highly structured brain connectivity that allows the rapid development of useful behaviors without much experience (Zador (2019)). These structured neural architectures have been fine-tuned over millions of years by evolution to bias the organism toward behavior which is useful for survival.

This process is not too unfamiliar, as decades of artificial intelligence research has produced many models of representing data that provide strong inductive biases on a variety of task domains. Convolutional neural networks (CNNs) perform particularly well on image learning problems, with even randomly initialized CNNs can provide excellent results in certain image learning settings (Ulyanov et al. (2018)). Recurrent Neural Networks (RNNs) are particularly well-suited for variable input and variable time based problems and Long-Short Term Memory networks (LSTMs) present recurrent structure that is well-suited for handling much larger time spans than the RNNs (Lipton et al. (2015)).

A recent grand endeavor in neuroscience has led to the mapping of the first completed connectome of a nematode species, the *Caenorhabditis Elegans* (*C. Elegans*). For the first time an opportunity has arisen for translating connectomes from neuroscience into the architecture of an artificial neural network. However, it is uncertain as to whether this representation could provide value to machine learning. Furthermore, it is uncertain as to how the architecture should model the synaptic and neuronal dynamics to best represent the connectome.

A complimentary work explores designing an ANN architecture that resembles *C. Elegans*-like microcircuits by combining discrete-time integrator and oscillation units (Bhattasali et al. (2022)). However, (a) this work did not use the *C. Elegans* locomotion circuit, (b) the dynamics of the pre-

sented architecture differs from the *C. Elegans* locomotion circuit (*e.g. the use of sinusoidal input*), and (c) did not provide understanding of how to better utilize exact connectome data. Additionally, it did not provide either improvements nor limitations to connectome-based approaches. **The goal of our paper is not to design an architecture that works well on arbitrary problems, but to discern where and when biological connectomes have use for machine learning problems and how to properly harness connectome data for learning applications.**

To begin answering these questions, we translate the locomotion circuit of the *C. Elegans* nematode into artificial neural networks with neurons models at varying levels of biophysical realism. We use these models to solve (1) a common continuous control problem that resembles nematode locomotion in critical ways yet remains different from the details of the *C. Elegans* body plan, and (2) learning problems that have little relationship to the function of the *C. Elegans* locomotion circuit. For the nematode-like learning problem, we show that the *C. Elegans* connectome provides a powerful inductive bias that enables improved performances over randomly connected neural networks. However, when training on learning problems with little relationship to nematode-like locomotion (e.g. image classification), the performance is dramatically hindered compared with more densely connected *general* architectures.

The primary contributions of this work are as follows:

1. Results demonstrating that, while biophysical realism does indeed improve performance, it is not a necessary condition for attaining the benefits of using biological connectomes.

2. Establishing that the exact wiring diagram provides the most beneficial inductive bias, but also that retaining the architectural statistics of the biological connectome *without* the precise connective patterns still proves to provide valuable priors.

3. A set of experiments demonstrating that while using biological connectomes as an architectural representation may provide a beneficial inductive bias on problems related to the function of that circuitry, it may not be beneficial *in general* on problems unrelated to its function in nature.

4. Open-source software for automatically converting connectome models into artificial neural networks that can be used for solving machine learning problems. This software includes connectomes from the *C. Elegans*, which are studied in this work, and various regions of the *Drosophila Melanogaster* brain, which provide exciting opportunities for future work. It also provides a general structure allowing for novel connectomes to be converted.

We pay close attention to the training methodology (*e.g. choice of optimization, population sizes that reflect natural C. Elegans populations*) and architectural design features (*e.g. neuron model, synapse weight signs*) to remain as close to biology as possible while still providing useful theoretical advancements for machine learning. We believe this work provides meaningful progress toward an understanding of how connectomes may be useful and how to best harness their capabilities.

Background and Methodology

A short introductory background on the *C. Elegans* is important for understanding the results of this work, which is provided below. Here we discuss what the *C. Elegans* is, how its motor circuit works to produce locomotion, and the population and evolutionary dynamics of *C. Elegans* colonies.

The Caenorhabditis Elegans

In the laboratory setting, *C. Elegans* are one of the simplest organisms to study and thus the nematode nervous system has served as an integral model in neuroscience over the past few decades. In addition, *C. Elegans* was the first organism to have its full genome sequenced and still remains as the only organism species which has its *entire* neuronal wiring diagram published to date (Cook et al. (2019)).

The body and neural circuitry of Caenorhabditis Elegans. *C. Elegans* is a free-living transparent nematode that thrives in temperate soil environments. Its cylindrical worm-like body spans a mere 1

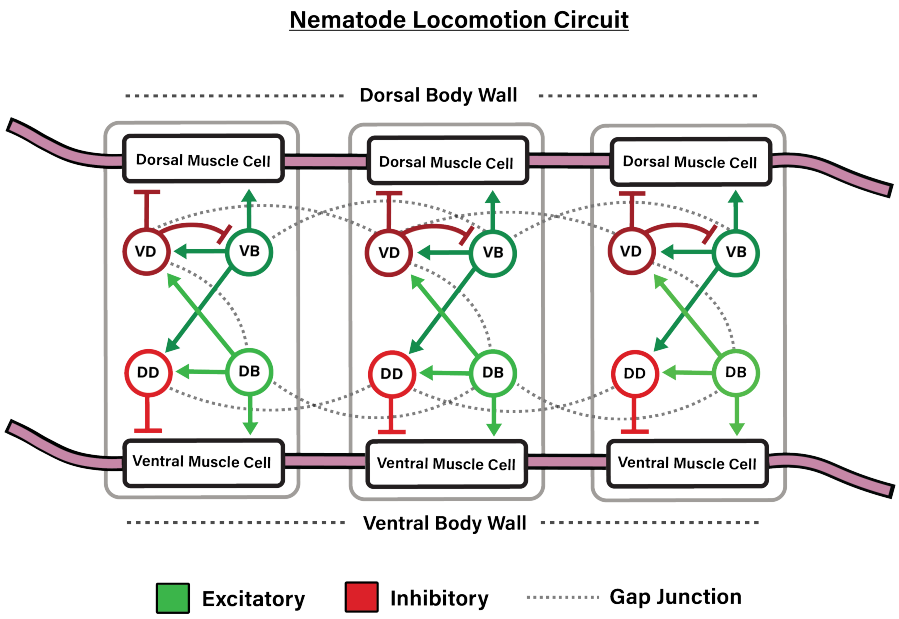

Figure 1: Graphical depiction of a segment of the *C. Elegans* locomotion circuit, which is characterized by the depicted microcircuits repeated down the body of the nematode. This circuit propagates waves down the nematode body which oscillate between innervating ventral and dorsal sides of the muscle body wall.

mm in length and is comprised of 959 somatic cells, 302 of which are neurons [1] (White et al. (1986)). Its exterior is unsegmented and bilaterally symmetric (*i.e. worm-like*), consisting of a simple set of anatomical structures: a mouth, intestines, gonad, and connective tissue, see Fig. 2.

Despite the incredible simplicity of these organisms, the nematode's neuronal dynamics closely resemble that of organisms with more complex nervous systems, and thus has served as a valuable model for neuroscience. Neurons in the nematode nervous system are generally separated into three categories based on their characteristic function and relationship to other neurons: sensory, motor and interneurons (Cook et al. (2019)). In this work we focus on *C. Elegans* motor neuron circuitry and apply this toward solving locomotion problems.

Locomotion in Caenorhabditis Elegans. Among the few cells in the *C. Elegans* connectome, 75 are motor neurons and 95 are body wall muscle neurons. Motor neurons are stimulated by sensory and interneurons, and are the primary driver of muscle neurons activity. These neural microcircuit patterns are repeated along the body in six repeating units of 12 motor neurons and 12 muscle neurons, see Fig 1 (Haspel & O'Donovan (2011); Haspel & O'Donovan (2012); Zhen & Samuel (2015)). These patterns produce muscle wave propagations that travel down the body and produce sinusoidal-like locomotion patterns that allow for a surprisingly wide-range of behaviors (Gray et al. (2005)). Within each microcircuit, forward locomotion is accomplished through the coordination of two classes of motor neurons, B and D, which are further characterized by their location along the body's radial axis, dorsal (D-) and ventral (V-). The B-type motor neuron class is excitatory, meaning it makes its outgoing neurons *more likely* to fire, and acts to innervate both muscle body wall neurons along its respective radial axis and D-type neurons on the opposing radial axis (e.g. ventral to dorsal) (Wen et al. (2012)). D-type motor neurons are inhibitory, making their outgoing connections *less likely* to fire, and inhibit both the firing of their respective muscle body wall neurons and, in the case of VD neurons, the excitatory neuron along its own radial axis (Wen et al. (2012)).

Population dynamics of Caenorhabditis Elegans. Population size estimates range from 750 to 12,000 depending on the geographical location of the population, the time of year, and the avail-

[1] These numbers were measured for the hermaphrodite nematode, with male nematode cell counts being generally larger (1031 somatic cells and 385 neurons).

ability of nutrients (Frézal & Félix (2015)). *C. Elegans* populations consist of both males and hermaphrodites, however, the vast majority of individuals are *hermaphrodites* (roughly 99.9% of the species), meaning each organism has both male and female reproductive organs (Chasnov (2013); Frézal & Félix (2015)). Reproduction is primarily propagated via hermaphroditic self-fertilization, where the hermaphrodite nematode produces offspring using its own reproductive material. As is discussed below, we take these principles of *C. Elegans* population dynamics (e.g. hermaphrodite self-fertilization) and model them through the use of genetic algorithms.

Genetic algorithms reflect Caenorhabditis Elegans evolutionary dynamics. Genetic algorithms are a biologically-inspired optimization process which frames itself around an abstraction of the process of natural selection (Holland (1992); Luke (2013); De Jong (2017)). This algorithm works by (a) maintaining a population of organisms which (b) undergo a selection process based on their performance on a given task, and (c) the top performing organisms repopulate the subsequent organism pool. Selection in embodied learning problems is typically carried out through the use of a fitness function, which is a metric for determining the quality of an organism's behavior over the course of its lifetime. In a genetic algorithm, typically only the *top-N* highest fitness achieving organisms survive to the next generation (referred to as *elite selection*). To expand the population after the selection process, each of the remaining organisms are taken and mutated slightly to create the next generation of the population. This process reflects the evolutionary and reproductive dynamics of the *C. Elegans* relatively well.

Solving Learning Problems with Biological Connectomes

Training dynamics. Our experimental results are collected using a genetic algorithm where replication is based on self-fertilization, or asexual reproduction, with each child in the population having only one parent (*i.e. no crossover*). This is to best mirror the reproductive and evolutionary patterns of *C. Elegans* populations. Additionally, we set the population size of the genetic algorithm to 750, matching *C. Elegans* population ranges present in nature (Barrière & Félix (2005)).

Connectome Representation. Dale's law states that in biological circuits, the sign of the weight of a synapse (excitatory or inhibitory) does not change. We wish to respect Dale's law (Eccles (1976)) by preserving excitatory and inhibitory function of synapses in the connectome model, synaptic weights are represented on the *log-scale* as follows: $w_{i,j} = s_{i,j}\exp(w_{i,j}^{log})$. Genetic algorithm mutations occur on the log weights $w_{i,j}^{log}$, that appropriately bound $\exp(w_{i,j}^{log})$ above zero, which, when multiplied by the original weight sign $s_{i,j}$, preserves the excitatory or inhibitory function of the synapse as is specified in the connectome model. This preservation has been shown to play significant role in retaining the functional properties of oscillatory circuits (Bhattasali et al. (2022)).

Experiments

A body design resembling Caenorhabditis Elegans locomotion

We begin our investigation of the *C. Elegans* connectome on a body design, Swimmer, that is commonly used for continuous control in the Deepmind Control Suite (Tassa et al. (2018)) and OpenAI Gym (Brockman et al. (2016)). Much like in Bhattasali et al. (2022), we choose Swimmer instead of a more accurate neuromechanical model e.g. Izquierdo & Beer (2015); Sarma et al. (2018) to demonstrate the use of connectomes on problems of interest to the machine learning community.

The Swimmer body is comprised of 6 cylindrical links that are connected via 5 articulated joints and looks similar to a worm, see Fig. 2. Instead of pushing off of solid ground, the Swimmer body is encompassed in simulated fluid which it must push against to generate forward momentum and is rewarded proportional to its current velocity along the x-axis. The ideal solution in this environment is indeed very much like *C. Elegans* locomotion, with the Swimmer body propagating a wave down its body starting from the head ending at the tail.

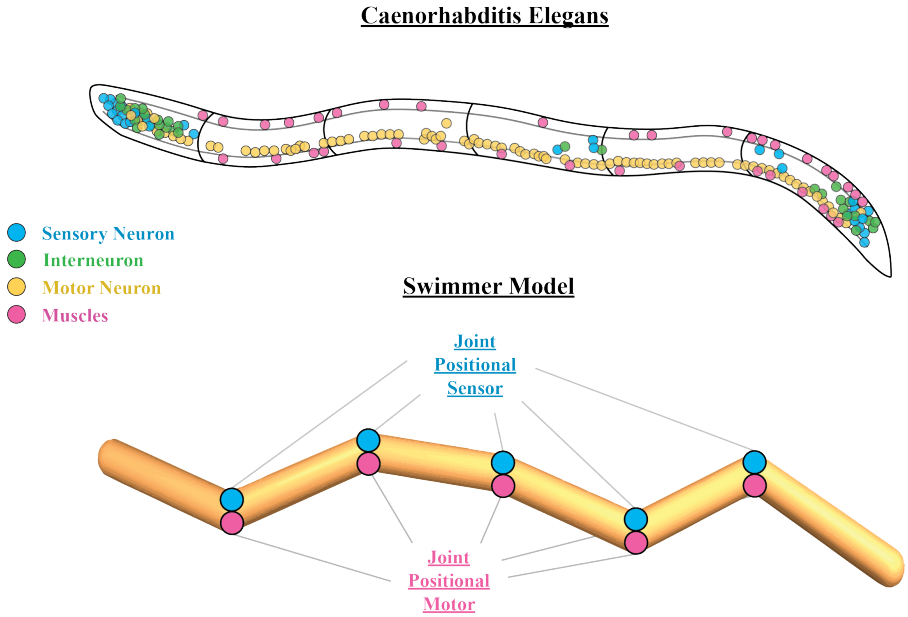

Figure 2: (Top) Graphical depiction of the *Caenorhabditis Elegans* nematode body and its corresponding neuronal placements categorized into sensory (*blue*), motor (*yellow*), muscle (*pink*), and interneurons (*green*). (Bottom) Depiction of the 6-jointed Swimmer body design along with its joint positional sensors and its joint positional motors highlighted blue and pink to draw relationship to *C. Elegans* sensory neurons and muscle neurons respectively.

VARYING BIOPHYSICAL REALISM

Described in this experiment are three neuron models with varying degrees of biophysical realism presented in order of *least* to *most* biophysically accurate. We wish to determine whether we can deviate from biological neuron dynamics to still realize the benefits of the *C. Elegans* connectome.

Artificial Neuron Model. The artificial neuron, which is the neuron model most commonly used in machine learning applications, applies an arbitrary non-linearity to the sum of weighted inputs from pre-neuron i to post-neuron j via $\boldsymbol{z}_j(t) = \sigma(\boldsymbol{w}_{i,j} \cdot \boldsymbol{z}_i(t))$, where $\boldsymbol{w}_{i,j}$ represents the weight from neuron i to j, $\boldsymbol{z}_j(t)$ represents the activity of neuron j at time t, and σ represents a typically smooth and continuous non-linear activation function. While the original activations used in artificial neural networks are more biological in nature (McCulloch & Pitts (1943)), these were slowly replaced over time with smoother time-invariant functions due to the use of differentiation as the medium of optimization, which tends to converge faster with smooth gradient landscapes (Sharma et al. (2017); Nwankpa et al. (2018)). In our experiments we use a ReLU nonlinearity, which resembles the more biologically descriptive neuron models in that it has an *all-or-nothing* firing pattern, producing either a zero or some quantity *x*.

Adaptive Leaky Integrate-and-Fire Neuron Model. The Leaky Integrate-and-Fire (LIF) neuron is characterized via a set of equations which aim to represent the artificial neuron activation function in a more biologically realistic manner while remaining computationally efficient (Lapique (1907); Tuckwell (1988); Abbott (1999)). The LIF equations are as follows:

$$\boldsymbol{v}_j(t + \Delta\tau) = \gamma \boldsymbol{v}_j(t) + \boldsymbol{w}_{i,j}(t)\boldsymbol{s}_i(t), \tag{1}$$

$$\boldsymbol{s}_j(t) = H(\boldsymbol{v}_j(t)) = \begin{cases} 0 & \boldsymbol{v}_j(t) < v_{th} \\ 1 & \boldsymbol{v}_j(t) \geq v_{th} \end{cases}. \tag{2}$$

In this model, activity is integrated into the neuron membrane potential $\boldsymbol{v}_j$ and retained across time, and, once the activity exceeds a threshold value $\boldsymbol{v}_j(t) \geq v_{th}$, a binary action potential, $\boldsymbol{s}_j(t) = 1$, is propagated and the membrane potential is reset to zero. The "leaky" component of the LIF refers to the membrane potential ($\boldsymbol{v}_j$ for a given neuron j) being slowly decayed, or leaked, over time by a constant factor $0 < \gamma < 1$. This leaking property more accurately represents the passive ion diffusion that takes place in biological neurons, and has important consequences for learning.

While the LIF model is certainly closer to biology than the artificial neuron model, it falls short in biophysical representation capacity in several ways. The most significant disadvantage is that it cannot capture neuronal adaptation, which prevents it from representing measured spike trains with constant input current. Neuronal adaptation is the process by which, in the presence of a constant current, the time between action potentials increases over time. This process can be incorporated very simply into the LIF model, becoming the Adaptive LIF (ALIF), if we let each neuron have a local trace, $v_{adp}(t)$, which accumulates action potentials as follows: $v_{adp}(t) = \gamma_s v_{adp}(t) + \boldsymbol{\beta s}(t)$, where γ_s controls the adaptation decay and $\boldsymbol{\beta}$ determines the affect action potentials have on the new threshold. Finally, $v_{adp}(t)$ is incorporated into the action potential firing threshold which becomes $\boldsymbol{v}_j(t) > v_{th}$ becomes $\boldsymbol{v}_j(t) > v_{th} + v_{adp}(t)$.

However, like the LIF model, the ALIF model still struggles to capture fundamental behaviors present in biological neurons (Izhikevich (2004)) and thus falls short in providing thorough biophysical accuracy.

Izhikevich Neuron Model. Many accurate biophysical neurons models, such as the Hodgkins-Huxley, are computationally prohibitive and can only simulate several neurons at real-time speed, preventing their use in computationally intensive applications. To get around this problem, while still maintaining the important properties of more complicated biophysical models, the Izhikevich neuron model (Izhikevich (2003)) effectively balances realism and tractability. This is accomplished via the following set of differential equations:

$$v_i(t + \Delta\tau) = v_i(t) + 0.04v_i^2 + 5v_i + 140 - u_i + \sum_j W_{i,j}(t)s_j(t). \tag{3}$$

$$u_i(t + \Delta\tau) = u_i(t) + a(bv_i - u_i). \tag{4}$$

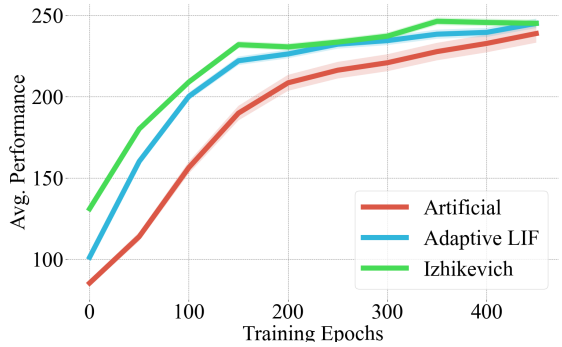

Figure 3: Performance across

In these equations, a spike is fired ($s_j(t) = 1$) when $v_i(t) \geq 30$ mV, upon which $v \leftarrow c$ and $u \leftarrow u + d$. While seemingly non-biologically descriptive at first, the Izhikevich neuron model is actually the product of rigorously fitting the parameters of a quadratic integrate-and-fire equation to a wide range of *in-vitro* recordings of different cortical neuron dynamics. This allows the model to be both highly biophysically accurate and computationally efficient. A selection of the values of the four variables *a, b, c, and d* determine the spiking and bursting dynamics of the neuron. Additional training details for all three neuron models are provided in the Appendix.

Results. The premise of the following experiment is simple: We will compare the performance of the *C. Elegans* locomotion circuit with varying degrees of realism trained on the Swimmer locomotion problem. The weights of these circuits are trained using a genetic algorithm which resembles *C. Elegans* evolutionary dynamics as is described both in *Methods* and in-depth in the *Appendix*. The results of these experiments are shown in Fig. 3.

Similar to the artificial circuits used in Bhattasali et al. (2022), we do not observe much additional performance growth relative to the network prior with additional training, especially with the more biologically realistic neuron models. We can see that the three networks converge to similar final values, with the more biophysically accurate neuron models performing slightly better than the artificial neuron model. The primary difference between these models is in their *initial average performance*, with the Izhikevich neuron model beginning at ~ 130, the ALIF around ~ 100 and the Artificial neuron model around ~ 50. The most important finding of these graphs however, is that biophysical realism is not necessary to realize the potential of biological circuits.

ARCHITECTURAL STATISTICS CONSERVE BENEFITS OF INDUCTIVE PRIORS

While a comprehensive map of the *C. Elegans* neural connections is available in its entirety, this organism is an exception – it is the only organism with a complete connectomic mapping. Nonetheless, while no other organism has a complete mapping, many organisms have *approximate connectome statistics* available (*e.g. number of neurons, number of inhibitory/excitatory synapses*). To motivate the use of connectomes, it would also be necessary to motivate the use of **connectome statistics** since outside of the *C. Elegans* this constitutes the majority of our knowledge of the connectomes of other organisms.

Toward this end, we design an experiment where the **essential statistics** of the *C. Elegans* locomotion circuit are used to construct a sparse Multi-Layer Perceptron (MLP). We then compare the performance of the *C. Elegans* approximate MLP with a fully connected MLP and a sparsely connected MLP with random architectural statistics. The performance is compared on the Swimmer task in order to determine how much architectural bias the *C. Elegans* circuit retains.

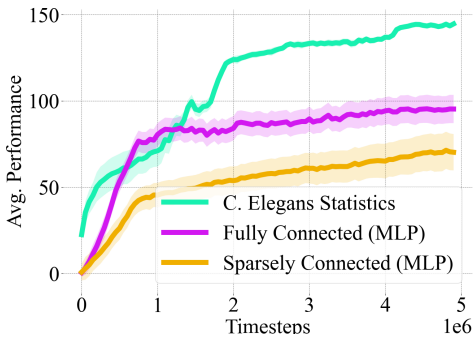

Figure 4: Performance across three network architectures. The first of which has the connectivity statistics of the C. ELegans locomotion circuit. The second is a fully connected MLP. The final curve represents a sparsely connected MLP with randomly sampled sparsity.

The *C. Elegans* statistics network is a sparse artificial neural network which has the same number of neurons and synapses as the original *C. Elegans* connectome, where the excitatory and inhibitory neuron ratios are preserved but the precise circuitry is not. The fully connected neural network is a two-hidden-layer neural network with the same number of neurons as the *C. Elegans* network, however the neurons have a fully connected structure between layers (i.e. an MLP). The final network is also a two-hidden-layer MLP, however the layer between hidden neurons has a sparsely connected topology, where the sparsity and excitatory-inhibitory ratios are randomly sampled from a uniform distribution.

Fig. 4 shows the performance across the three compared network architectures. Both the fully connected and sparsely connected MLPs begin with a performance around 0, but the approximate *C. Elegans* network begins with a performance around 22, retaining a fraction of the locomotion circuit inductive bias. The *C. Elegans* network obtains a final performance of 146.4 ± 3.5 compared with the fully connected MLP around 96.2 ± 13.6 and the sparsely connected MLP around 63.7 ± 17.2. It is clear that the precise circuitry of the *C. Elegans* connectome remains superior. However, these results indicate that, **while the precise connectome may not be known, approximate architectural statistics can still provide useful information for solving a problem.** Nonetheless, more investigation needs to be done around why connectome statistics still provide an inductive bias despite the network microcircuits being degraded.

THE LIMITATIONS OF ARCHITECTURAL PRIORS

We have thus far demonstrated that the *C. Elegans* connectome model may provide useful behavioral priors on *a particular task* that is closely related to the purpose of that connectome. In this case, we showed that the *C. Elegans* connectome provides a powerful inductive prior on a swimmer locomotion task that requires oscillatory wave propagation motion – exactly what the *C. Elegans* locomotion circuit is designed to do. However, we wish to investigate to what extent these priors prove useful on tasks that do not resemble their biological purpose. How well would the *C. Elegans* connectome perform on a legged locomotion problem or on an image learning problem?

To determine this, we examine the performance of the *C. Elegans* connectome on a locomotion problem from the same benchmark suite, as well as a classification problem that is completely outside the domain of the *C. Elegans* architecture design. For the locomotion problem, we evaluate the performance of the connectome on the Half Cheetah task (Tassa et al. (2018)), which consists of a 2D planar robot with a total of 7 links, 2 legs and a torso, where each leg has a total of 3 actuated joints. The objective is to run forward with the highest velocity possible. For the image learning problem we evaluate the performance of the connectome on MNIST (LeCun (1998)), a 28x28 dimensional handwritten digit classification problem with 10 digit classes.

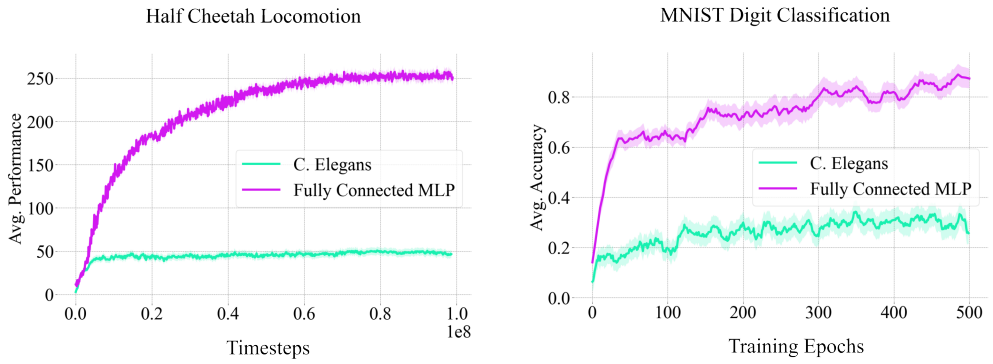

Figure 5: Training performance curves on (a) a reinforcement learning problem, the Half Cheetah locomotion benchmark (Tassa et al. (2018)), and (b) an image learning problem, MNIST (LeCun (1998)). It can be seen that the C. Elegans network structure hinders performance on both tasks and converges early in the training process, whereas the MLP performs relatively well on both tasks.

In Fig. 5 we can see that on both the Half Cheetah and MNIST problems, the C. Elegans connectome not only does not outperform a standard MLP like in previous problems, but also performs dramatically worse than it. There is a characteristic deterioration in performance growth after a small amount of training time. This lack of performance growth may be a product of the *C. Elegans* dependence on structure priors at birth versus on intra-lifetime learning like in higher order mammals, thus developing an architecture that biases toward *innate knowledge* over *learning capability*. While this innate bias is helpful for the Swimmer locomotion problem, as these results demonstrate, this innate bias is in fact a limitation when using the connectome on problems that do not resemble nematode locomotion. Stated more clearly, the architecture constrains the set of possible behaviors, in turn biasing the network to the subset of behaviors that the architecture was designed to perform (e.g. *oscillations*). These findings demonstrate that connectomes must be applied in the right context to realize the benefits of their architectural priors.

DISCUSSION AND FUTURE WORK

In this work we discerned where and when biological connectomes can prove advantageous for machine learning. We showed that for problems resembling the function of the connectome (*e.g. oscillatory locomotion for the C. Elegans)*, there exists a strong and beneficial inductive bias. However, for problems not resembling the function of the connectome, its structure constrains the model's representation, and degrades training performance. We showed that increasing biological realism

improves the priors of the connectome, but that realism is not necessary for extracting the benefits of the network structure. Finally, we demonstrated that architectural statistics conserve some benefits of the inductive priors of the architecture.

One of the primary limitations of using biological connectomes is that it requires a degree of understanding in systems neuroscience to enable an effective implementation. To alleviate this difficulty, we have developed open-source software tools to automatically convert connectome models into artificial neural networks that can be readily used for solving optimization problems[2]. This includes a process of finding a good set of *initial shared weights* for the excitatory and inhibitory connections, as well as tuning the neuron parameters for arbitrary learning problems. We hope this software will help further research at the intersection of connectomics and machine learning, particularly for exploring the connectome of the *Drosophila melanogaster*, which includes a visual processing system (Takemura (2015)) that may lead to new and powerful architectural discoveries for visual learning problems.

We believe that connectomes may provide structure that promotes stable bounds for models of synaptic plasticity, which have seen a recent resurgence for use in deep learning applications (Miconi et al. (2020); Najarro & Risi (2020); Schmidgall et al. (2021); Schmidgall & Hays (2022a)). As this work has demonstrated, the connectome seems to provide a strong inductive bias on both the initial behavior and the set of possible behaviors that the network can produce. The use of structured connectomes could prevent highly dynamic weights from deteriorating behavior. This differs from a more general architecture where the behavior would change much more dramatically with small changes in weights (Schmidgall & Hays (2022b)). These perturbations become more dramatic when weights are changing simultaneously and independently.

We believe that the incredible sparsity of biological connectomes (*3.2% for the C. Elegans*, Cook et al. (2019)) will significantly decrease the amount of energy necessary to deploy deep learning models, perhaps through the use of neuromorphic hardware (Young et al. (2019); Zhu et al. (2020); Schuman et al. (2022)). These connectomes may lead to more robust and resilient neural systems that sidestep many of the adversarial drawbacks of highly general network structures (Guo et al. (2018); Schuman et al. (2020)). Finally, we believe that the use of biological connectomes in machine learning may lead to profound advancements in both our understanding of the role of neural architecture and in discovering new neural architectures. We believe that our work is a step toward bridging the gap between our understanding of systems neuroscience and artificial intelligence.

REPRODUCIBLITY

To ensure reproducibility we have included an exhaustive list of hyperparameters for both the GA, ES, and neuron models in every experiment. A detailed explanation of the training process is also described in the Appendix. Finally, a software toolkit for using connectomes as neural architecture is released along with the paper.

REFERENCES

Larry F Abbott. Lapicque's introduction of the integrate-and-fire model neuron (1907). *Brain research bulletin*, 50(5-6):303–304, 1999.

Antoine Barrière and Marie-Anne Félix. Natural variation and population genetics of caenorhabditis elegans. *WormBook*, 2005:1–19, 2005.

Nikhil X Bhattasali, Anthony M Zador, and Tatiana A Engel. Neural circuit architectural priors for embodied control. *arXiv preprint arXiv:2201.05242*, 2022.

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

Jeffrey R Chasnov. The evolutionary role of males in c. elegans. In *Worm*, volume 2, pp. e21146. Taylor & Francis, 2013.

[2]The link will be attached here upon acceptance.

Steven J Cook, Travis A Jarrell, Christopher A Brittin, Yi Wang, Adam E Bloniarz, Maksim A Yakovlev, Ken CQ Nguyen, Leo T-H Tang, Emily A Bayer, Janet S Duerr, et al. Whole-animal connectomes of both caenorhabditis elegans sexes. *Nature*, 571(7763):63–71, 2019.

Kenneth De Jong. Evolutionary computation: a unified approach. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pp. 373–388, 2017.

John Carew Eccles. From electrical to chemical transmission in the central nervous system: the closing address of the sir henry dale centennial symposium cambridge, 19 september 1975. *Notes and records of the Royal Society of London*, 30(2):219–230, 1976.

Lise Frézal and Marie-Anne Félix. The natural history of model organisms: C. elegans outside the petri dish. *elife*, 4:e05849, 2015.

Jesse M Gray, Joseph J Hill, and Cornelia I Bargmann. A circuit for navigation in caenorhabditis elegans. *Proceedings of the National Academy of Sciences*, 102(9):3184–3191, 2005.

Yiwen Guo, Chao Zhang, Changshui Zhang, and Yurong Chen. Sparse dnns with improved adversarial robustness. *Advances in neural information processing systems*, 31, 2018.

Charles R Harris, K Jarrod Millman, Stéfan J Van Der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J Smith, et al. Array programming with numpy. *Nature*, 585(7825):357–362, 2020.

Gal Haspel and Michael J O'Donovan. A perimotor framework reveals functional segmentation in the motoneuronal network controlling locomotion in caenorhabditis elegans. *Journal of Neuroscience*, 31(41):14611–14623, 2011.

Gal Haspel and Michael J O'Donovan. A connectivity model for the locomotor network of caenorhabditis elegans. In *Worm*, volume 1, pp. 125–128. Taylor & Francis, 2012.

John H Holland. Genetic algorithms. *Scientific american*, 267(1):66–73, 1992.

Eugene M Izhikevich. Simple model of spiking neurons. *IEEE Transactions on neural networks*, 14(6):1569–1572, 2003.

Eugene M Izhikevich. Which model to use for cortical spiking neurons? *IEEE transactions on neural networks*, 15(5):1063–1070, 2004.

Eduardo J Izquierdo and Randall D Beer. An integrated neuromechanical model of steering in c. elegans. In *ECAL 2015: the 13th European Conference on Artificial Life*, pp. 199–206. MIT Press, 2015.

Louis Lapique. Recherches quantitatives sur l'excitation electrique des nerfs traitee comme une polarization. *Journal of Physiology and Pathololgy*, 9:620–635, 1907.

Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

Zachary C Lipton, John Berkowitz, and Charles Elkan. A critical review of recurrent neural networks for sequence learning. *arXiv preprint arXiv:1506.00019*, 2015.

Sean Luke. *Essentials of metaheuristics*, volume 2. Lulu Raleigh, 2013.

Warren S McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, 1943.

Thomas Miconi, Aditya Rawal, Jeff Clune, and Kenneth O Stanley. Backpropamine: training self-modifying neural networks with differentiable neuromodulated plasticity. *arXiv preprint arXiv:2002.10585*, 2020.

Elias Najarro and Sebastian Risi. Meta-learning through hebbian plasticity in random networks. *Advances in Neural Information Processing Systems*, 33:20719–20731, 2020.

Chigozie Nwankpa, Winifred Ijomah, Anthony Gachagan, and Stephen Marshall. Activation functions: Comparison of trends in practice and research for deep learning. *arXiv preprint arXiv:1811.03378*, 2018.

Tim Salimans, Jonathan Ho, Xi Chen, Szymon Sidor, and Ilya Sutskever. Evolution strategies as a scalable alternative to reinforcement learning. *arXiv preprint arXiv:1703.03864*, 2017.

Gopal P Sarma, Chee Wai Lee, Tom Portegys, Vahid Ghayoomie, Travis Jacobs, Bradly Alicea, Matteo Cantarelli, Michael Currie, Richard C Gerkin, Shane Gingell, et al. Openworm: overview and recent advances in integrative biological simulation of caenorhabditis elegans. *Philosophical Transactions of the Royal Society B*, 373(1758):20170382, 2018.

Samuel Schmidgall and Joe Hays. Learning to learn online with neuromodulated synaptic plasticity in spiking neural networks. *bioRxiv*, 2022a.

Samuel Schmidgall and Joe Hays. Stable lifelong learning: Spiking neurons as a solution to instability in plastic neural networks. In *Neuro-Inspired Computational Elements Conference*, pp. 1–7, 2022b.

Samuel Schmidgall, Julia Ashkanazy, Wallace Lawson, and Joe Hays. Spikepropamine: Differentiable plasticity in spiking neural networks. *Frontiers in neurorobotics*, pp. 120, 2021.

Catherine D Schuman, J Parker Mitchell, J Travis Johnston, Maryam Parsa, Bill Kay, Prasanna Date, and Robert M Patton. Resilience and robustness of spiking neural networks for neuromorphic systems. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–10. IEEE, 2020.

Catherine D Schuman, Shruti R Kulkarni, Maryam Parsa, J Parker Mitchell, Bill Kay, et al. Opportunities for neuromorphic computing algorithms and applications. *Nature Computational Science*, 2(1):10–19, 2022.

Sagar Sharma, Simone Sharma, and Anidhya Athaiya. Activation functions in neural networks. *towards data science*, 6(12):310–316, 2017.

Shin-ya Takemura. Connectome of the fly visual circuitry. *Microscopy*, 64(1):37–44, 2015.

Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, et al. Deepmind control suite. *arXiv preprint arXiv:1801.00690*, 2018.

Henry Clavering Tuckwell. *Introduction to theoretical neurobiology: linear cable theory and dendritic structure*, volume 1. Cambridge University Press, 1988.

Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Deep image prior. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 9446–9454, 2018.

Quan Wen, Michelle D Po, Elizabeth Hulme, Sway Chen, Xinyu Liu, Sen Wai Kwok, Marc Gershow, Andrew M Leifer, Victoria Butler, Christopher Fang-Yen, et al. Proprioceptive coupling within motor neurons drives c. elegans forward locomotion. *Neuron*, 76(4):750–761, 2012.

John G White, Eileen Southgate, J Nichol Thomson, Sydney Brenner, et al. The structure of the nervous system of the nematode caenorhabditis elegans. *Philos Trans R Soc Lond B Biol Sci*, 314 (1165):1–340, 1986.

Steven R Young, Pravallika Devineni, Maryam Parsa, J Travis Johnston, Bill Kay, Robert M Patton, Catherine D Schuman, Derek C Rose, and Thomas E Potok. Evolving energy efficient convolutional neural networks. In *2019 IEEE International Conference on Big Data (Big Data)*, pp. 4479–4485. IEEE, 2019.

Anthony M Zador. A critique of pure learning and what artificial neural networks can learn from animal brains. *Nature communications*, 10(1):1–7, 2019.

Mei Zhen and Aravinthan DT Samuel. C. elegans locomotion: small circuits, complex functions. *Current opinion in neurobiology*, 33:117–126, 2015.

Jiadi Zhu, Teng Zhang, Yuchao Yang, and Ru Huang. A comprehensive review on emerging artificial neuromorphic devices. *Applied Physics Reviews*, 7(1):011312, 2020.

A Appendix

A.1 Hardware Specifications

Training Device: 2021 Apple MacBook Pro

Operating System: macOS Monteray v12.3

Processor: Apple M1 Max with 10-core CPU (8 Performance and 2 Efficiency)

Installed Physical Memory (RAM): 64 GB of Unified Memory (LPDDR5)

A.2 Training details

Inputs were mapped into the locomotion circuit motor neurons (or the hidden neurons for MLPs) via a pre-trained non-linear encoder for each of the three networks (Artificial, ALIF, Izikevich) and for the three compared connectivity statistics networks. Outputs were mapped from the muscle body-wall neuron spiking activity to motor actions (or classification labels) via a pre-trained non-linear encoder. Additionally, for all networks, the initial weights and neuron model constants were pretrained, like in previous work (Bhattasali et al. (2022)), and these initial weights were individually fine-tuned during the GA training. Pretraining was accomplished using an Evolutionary Strategies optimizer (Salimans et al. (2017)) for 1000 epochs, with details below.

Neural network, genetic algorithm, and evolutionary strategies code was implemented using the Numpy Python library (Harris et al. (2020)).

A.3 Hyperparameters

A.3.1 Genetic Algorithm

Varying Biophysical Realism

Population Size: 750

Elite Selection Size: 8

Mutation Rate (Encoder/Decoder): 0.01

Mutation Rate (Connectome Weights): 0.01

Mutation Rate Decay (Encoder/Decoder): 0.997

Mutation Rate Decay (Connectome Weights): 0.997

Architectural Statistics Conserve Benefits of Indudctive Priors

Population Size: 750

Elite Selection Size: 8

Mutation Rate (Encoder/Decoder): 0.01

Mutation Rate (Connectome Weights): 0.02

Mutation Rate Decay (Encoder/Decoder): 0.997

Mutation Rate Decay (Connectome Weights): 0.997

Limitations of Architectural Priors

Population Size: 750

Elite Selection Size: 8

Mutation Rate (Encoder/Decoder): 0.02

Mutation Rate (Connectome Weights): 0.02

Mutation Rate Decay (Encoder/Decoder): 0.998

Mutation Rate Decay (Connectome Weights): 0.998

A.3.2 EVOLUTIONARY ALGORITHM

Population Size: 200

Mutation Rate (Encoder/Decoder): 0.1

Mutation Rate (Shared, Connectome Weights): 0.1

Learning Rate (Encoder/Decoder): 0.1

Learning Rate (Shared, Connectome Weights): 0.1

Mutation Rate Decay (Encoder/Decoder): 0.999

Mutation Rate Decay (Shared, Connectome Weights): 0.999

Learning Rate Decay (Encoder/Decoder): 0.999

Learning Rate Decay (Shared, Connectome Weights): 0.999

A.3.3 NEURON MODELS

Adaptive Leaky Integrate-and-Fire Neuron Model

Voltage Decay: $e^{-1/20}$

Adaptation Time Constant: 0.5

Adaptive Threshold Decay: $e^{-1/10}$

Connectome Weight Initialization Sample (Excitatory): $w_{i,j} = e^{-1.0}$

Connectome Weight Initialization Sample (Inhibitory): $w_{i,j} = e^{-1.5}$

Encoder/Decoder Weight Initialization Sample: $w_{i,j} \sim \mathrm{U}(-0.3, 0.3)$

Izhikevich Neuron Model

Initial Parameter *a*: 0.02

Initial Parameter *b*: 0.25

Initial Parameter *c*: -58

Initial Parameter *d*: 0

Spike Firing Threshold: 30

Synaptic Delay: 1 ms

Current Scalar 20.0

Voltage Constant Scalar 0.2

Adaptive Threshold Decay: $e^{-1/10}$

Connectome Weight Initialization Sample (Excitatory): $w_{i,j} = e^{2.0}$

Connectome Weight Initialization Sample (Inhibitory): $w_{i,j} = e^{1.0}$

Encoder/Decoder Weight Initialization Sample: $w_{i,j} \sim \mathrm{U}(-0.3, 0.3)$

Artificial Neuron Model

Connectome Weight Initialization Sample (Excitatory): $w_{i,j} = e^{-2.5}$

Connectome Weight Initialization Sample (Inhibitory): $w_{i,j} = e^{-3.0}$

Encoder/Decoder Weight Initialization Sample: $w_{i,j} \sim \mathrm{U}(-0.3, 0.3)$