SCALING PROPERTIES FOR ARTIFICIAL NEURAL NETWORK MODELS OF A SMALL NERVOUS SYSTEM

Quilee Simeon*
{qsimeon}@mit.edu

Leandro R. Venâncio†
{lrvenan}@mit.edu

Edward S. Boyden‡ **Guangyu R. Yang**‡
{edboyden, yanggr}@mit.edu

Aran Nayebi§
{anayebi}@mit.edu
Department of Brain and Cognitive Sciences, MIT¶

ABSTRACT

The nematode worm *C. elegans* provides a unique opportunity for exploring *in silico* data-driven models of a whole nervous system, given its transparency and well-characterized nervous system facilitating a wealth of measurement data from wet-lab experiments. This study explores the scaling properties that may govern learning the underlying neural dynamics of this small nervous system by using artificial neural network (ANN) models. We investigate the accuracy of self-supervised next time-step neural activity prediction as a function of data and models. For data scaling, we report a monotonic log-linear reduction in mean-squared error (MSE) as a function of the amount of neural activity data. For model scaling, we find MSE to be a nonlinear function of the size of the ANN models. Furthermore, we observe that the dataset and model size scaling properties are influenced by the particular choice of model architecture but not by the precise experimental source of the *C. elegans* neural data. Our results fall short of producing long-horizon predictive and generative models of *C. elegans* whole nervous system dynamics but suggest directions to achieve those. In particular our data scaling properties extrapolate that recording more neural activity data is a fruitful near-term approach to obtaining better predictive ANN models of a small nervous system.

1 INTRODUCTION AND RELATED WORK

1.1 INTRODUCTION

Exploring neural system dynamics is crucial in neuroscience and artificial intelligence (AI). This intersection has spurred the evolution of artificial neural network (ANN) models, inspired by biological neural systems. ANNs offer the potential to emulate diverse animal behaviors, providing advantages like detailed specification, causal manipulability, and increasing analytical accessibility, reflecting key aspects of biological nervous systems (Yamins & DiCarlo, 2016; Yamins et al., 2014). The nematode *Caenorhabditis elegans* (*C. elegans*) is an exemplary model in this context, offering a valuable platform for comparing real and artificial neural dynamics.

*First author.
†Second author.
‡Equal contribution.
§Last author.
¶Affiliation of all authors.

1.2 *C. elegans* AS A MODEL SYSTEM

C. elegans is an excellent model organism for neural dynamics research due to its well-mapped connectome and capabilities for non-invasive neuronal activity tracking via advanced imaging techniques (Leifer et al., 2011; Nguyen et al., 2016). The organism's compact size, transparency, and well-annotated genome simplify intricate optical measurements and deep insights into neural activity. NeuroPAL, a multicolor atlas, allows precise *in vivo* neuron identification, enhancing the capabilities for measurement and analysis of the *C. elegans* nervous system (Yemini et al., 2021).

We formulate the problem of *in silico* nervous system modeling as a teacher-student framework a real biological neural network (that of *C. elegans*, in our case) is the teacher, ANN models are the students, and self-supervised next time-step neural activity prediction is the curriculum. We consider instances of this nervous system to be phenotypically matched animals (in our case, adult hermaphrodite worms).

1.3 SELF-SUPERVISED NEURAL ACTIVITY PREDICTION

Predicting future neural activity based on historical neural data is not new but the machine learning approach to do it has seen increasing adaption ever since advancements advancements in models like LSTMs demonstrating some success in mammals (Pandarinath et al., 2018). In *C. elegans*, the simplified behavioral repertoire and consistent biology offer a unique setting for in-depth model analysis. Self-supervised learning, predicting future states from intrinsic neural patterns, reduces dependence on behaviorally annotated data. While acknowledging the importance of behavior in neural dynamics, our study concentrates on the inherent predictability within neural activity, exploring how neural dynamics can be predicted without direct behavioral reference, similar to how large language models (LLMs) uncover intricate structures in language data (Radford et al., 2019; Brown et al., 2020).

1.4 NEURAL NETWORK SCALING PROPERTIES

Research into ANNs' scaling properties has shown that improvements in model size, data volume, and computational resources significantly enhance performance (Kaplan et al., 2020b; Hoffmann et al., 2022). The relationship between data size and model capacity is critical in optimizing model performance. However, this relationship in the context of predicting neural dynamics in biological organisms like *C. elegans* is not well-explored. Our study aims to fill this gap by analyzing the impact of data volume, model architecture, and size on ANN performance in neural activity prediction in *C. elegans*. These insights are crucial for optimizing experimental and modeling strategies in neuroscience, contributing to the development of more accurate predictive models for biological nervous systems.

2 METHODS

2.1 NEURAL ACTIVITY DATA

Data sources. We obtained 8 open-source datasets (Atanas et al., 2023; Randi et al., 2022; Yemini et al., 2021; Uzel et al., 2022; Kaplan et al., 2020a; Skora et al., 2018; Nichols et al., 2017; Kato et al., 2015) measuring neural activity in *C. elegans* (see Table 1 for download links and the associated publications). These datasets, each recorded under varying experimental conditions, quantify neural activity through the measurement of changes in calcium fluorescence ($\Delta F/F_0$) within subsets of the worm's 302 neurons. The 8 experimental datasets contained different numbers of recorded instances of the nervous system (i.e. worms) as well as variable lengths of time for neural activity measurements (Figure 1A). The experimental conditions were variable across datasets, ranging from freely moving (Atanas et al., 2023), immobilized (Uzel et al., 2022), and asleep (Nichols et al., 2017) states, to optogenetically stimulated scenarios (Randi et al., 2022) (see Figure 8 for more summary statistics about the experimental datasets). However, our modeling strategy is agnostic to the potentially differing conditions and protocols under which the data was acquired as long as the measured system (*C. elegans*) and metric ($\Delta F/F_0$) is consistent.

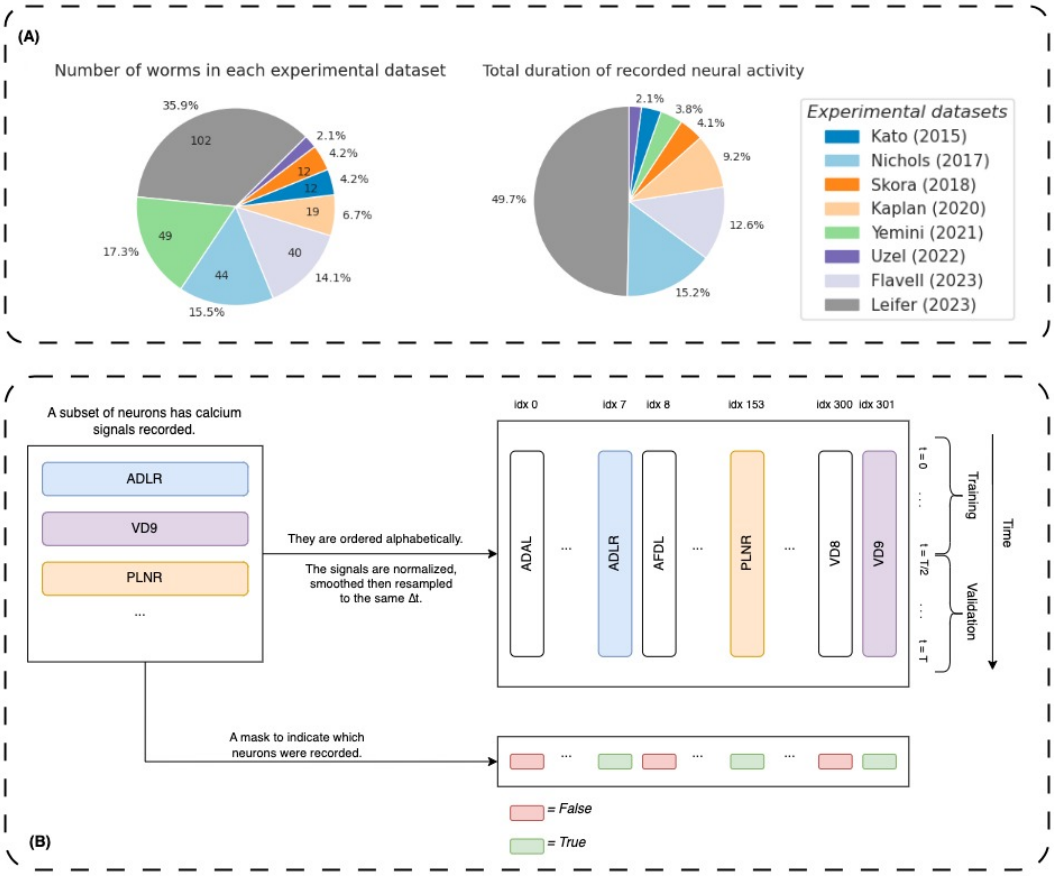

Figure 1: Worm neural datasets curation. Eight open-source *C. elegans* neural activity datasets were downloaded, preprocessed and assimilated. (A) The distribution of the number of worms and the total recording time in each dataset. (B) The neural activity data of all worms is organized a standard format involving a muti-dimensional time series and a boolean feature mask.

Standard data format. Each dataset $\mathcal{D}^n$ includes individual recordings from n worms, each consisting of neural activity and a mask indicating the subset of the 302 neurons that were measured and labelled. This mask ensures models are trained only on neural activity recorded from NeuroPAL labelled neurons (Figure 1B). Specifically:

$$\mathcal{D}^n = \{\mathbf{X}^1, \mathbf{X}^2, \ldots, \mathbf{X}^n\} \times \{\mathbf{y}^1, \mathbf{y}^2, \ldots, \mathbf{y}^n\}, \quad n = |\mathcal{D}^n|$$

Each worm, indexed by k, has a data matrix $\mathbf{X}^k \in \mathbb{R}^{302 \times T_k}$ and a binary vector $\mathbf{y}^k \in \{0,1\}^{302}$ specifying which neurons were recorded and labelled. Each row $\mathbf{x}_i^k$ of the matrix $\mathbf{X}^k$ contains the time series of neural activity for the ith neuron for T_k time steps. The rows of $\mathbf{X}^k$ and are ordered according to the alphabetical canonical names of the neurons [1] with rows ordered analogously to $\mathbf{y}^k$.

Preprocessing. The data, denoted as $\mathbf{X}^k$, is processed from the original raw data (see Table **??** for details about the individual experimental sources). First, we normalized the calcium data of each worm by z-scoring the full time series independently for each neuron. We then smoothed the signal using a causal exponential kernel (smoothing parameter $\alpha = 0.5$). Finally, the neural data was resampled to a fixed time step interval ($\Delta t \approx 0.667s$). Figure 7 steps through the preprocessing pipeline for a handful of neurons from 1 worm in the Kato dataset (Kato et al., 2015).

[1] https://www.wormatlas.org/NeuronNames.htm

Train-Test split. For each worm's neural activity data matrix $\mathbf{X}^k$, we performed a temporal split to create a training set $\mathbf{X}^k_{\text{train}}$ and a testing set $\mathbf{X}^k_{\text{test}}$. A balanced 50:50 split was adopted, allocating the first half of the neural activity recording to the training set and the second half to the testing set. One might create equally sized train and test sets containing n_{s} sequences of length L by sampling their start indices uniformly from (or equidistantly within) $[0, \lfloor T/2 \rfloor - L - 1)$ and $[\lceil T/2 \rceil, T - L - 1]$ for train and test, respectively

Amount of Data. The ability to vary the amount of training data is central to our investigation of the effects of data scaling on the ability of self-supervised models to do future neural activity prediction. However, we are data constrained in our setting since the collective dataset of all worms $\mathcal{D}_{ALL}$ has only 284 worms (see Figure 6 for the distribution of neurons in this dataset).

Source Datasets. The experimental sources contributing to $\mathcal{D}_{ALL} = \mathcal{D}^{284}$ are Kato $\mathcal{D}_{Kato}$ ($|\mathcal{D}_{Kato}| = 12$), Nichols $\mathcal{D}_{Nichols}$ ($|\mathcal{D}_{Nichols}| = 44$), Skora $\mathcal{D}_{Skora}$ ($|\mathcal{D}_{Skora}| = 12$), Kaplan $\mathcal{D}_{Kaplan}$ ($|\mathcal{D}_{Kaplan}| = 19$), Yemini $\mathcal{D}_{Yemini}$ ($|\mathcal{D}_{Yemini}| = 49$), Uzel $\mathcal{D}_{Uzel}$ ($|\mathcal{D}_{Uzel}| = 6$), Flavell $\mathcal{D}_{Flavell}$ ($|\mathcal{D}_{Flavell}| = 42$), Leifer $\mathcal{D}_{Leifer}$ ($|\mathcal{D}_{Leifer}| = 103$).

Mixed Datasets. To create mixed datasets combining worms from the different experimental sources, we will randomly sample from the combined pool of all available worms $\mathcal{D}_{ALL}$.

Let $i \in [8] := \{1, 2, \ldots, 8\}$ index into the list of experimental sources (sorted by publication date[2]): [Kato, Nichols, ..., Leifer]. We denote a mixed worm dataset containing n_w worms from any combination of experimental sources as $\mathcal{D}^{n_w}$.

$$\mathcal{D}^{n_w} = \bigcup_{i=1}^{8} \mathcal{D}_i^{(k_i)} \quad \text{s.t.} \quad \sum_{i=1}^{8} k_i = n_w$$

Here, $\mathcal{D}_i^{(k_i)}$ stands for k_i worms sampled specifically from the experimental dataset indexed by i (Figure 2A). Note that there are multiple assignments that can achieve a dataset with n_w worms. Therefore, D^{n_w} is a random variable. Our mixed dataset sampling process is akin to sampling from a multinomial distribution where the probabilities correspond to the proportion of available worms from each experimental source:

$$\mathcal{D}^{n_w} \sim \text{Multinomial}(n_w; p_1, \ldots, p_8)$$

where p_i reflects the relative contribution of the experimental dataset indexed by i to the pool.

This methodical approach allows us to create increasingly larger mixed datasets up to the largest one containing all worms $\mathcal{D}_{ALL} := \mathcal{D}^{284}$ (for which there is only 1 possible assignment). The result is a series of mixed datasets, each with a unique composition of worms, yet collectively spanning the full range of neural dynamics present in the collective data. The mixed datasets $\mathcal{D}^{n_w}$ thus represent a diverse cross-section of neural activities encompassing variations in experimental conditions.

We could also generate subsets from any single experimental dataset using this approach by simply restricting our random sampling to that source. This allows us to create increasingly larger data subsets constrained to a particular experimental context.

Scaling Datasets. The data scaling experiment we describe later will test models across discrete mixed dataset sizes ranging from $\mathcal{D}^1$ to $\mathcal{D}^{284}$. The size and composition of each dataset is determined by the multinomial sampling process described earlier. For any discrete dataset size n_w desired it is possible possible to perform bootstrapping by randomly selecting an assignment from all possible assignments that yield the desired size n_w.

Synthetic Datasets. We created synthetic datasets that mirror the complexity and challenges of the processed real *C.elegans* neural activity data, such as partial observability, sparsity, and noise, with 200 synthetic 'worms' each having 1500 time steps of activity from 50 randomly chosen neurons. These datasets allow for robust model validation against known dynamics, offering clear benchmarks for performance evaluation.

[2]See Table 1.

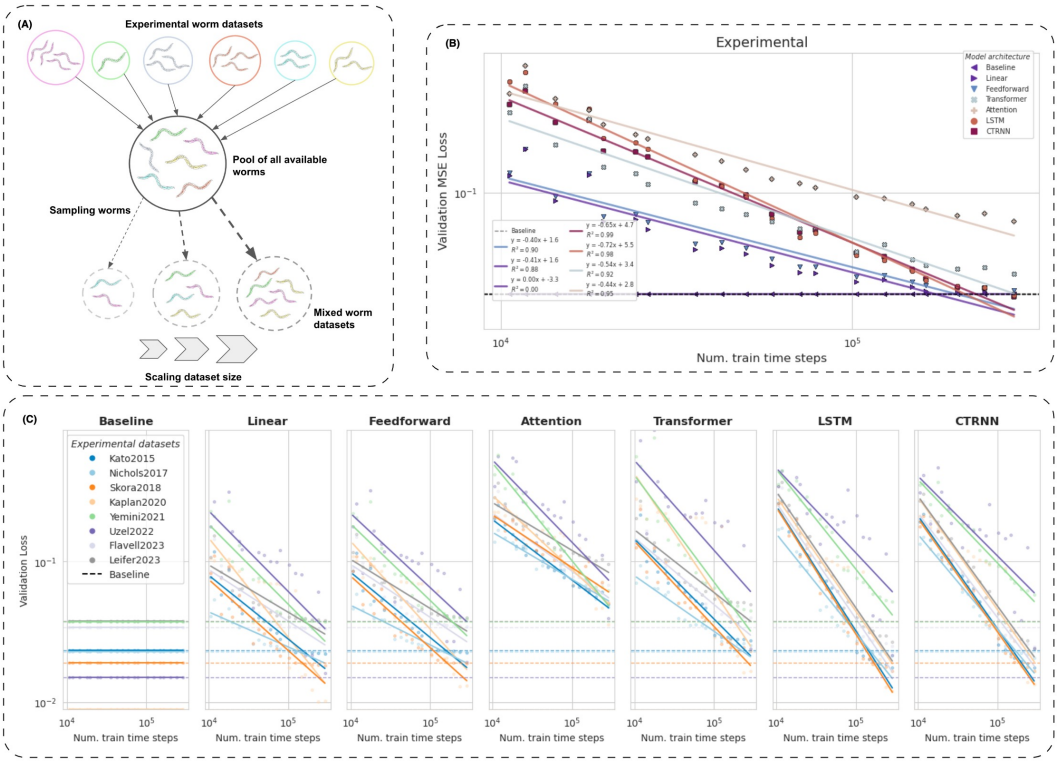

Figure 2: Sampling datasets and dataset scaling experiment. (A) Creating mixed worm datasets of various sizes by sampling from the pool of all available worms across different experimental datasets. (B) The validation loss from optimizing for next time step prediction is a decreasing function of the number of training time steps and depends on the model architecture. All models are approximately matched in size at 580K (K=thousand) trainable parameters. (C) Within a model class, the data scaling slopes exhibit remarkable similarity across experimental dataset sources, despite the diversity of the experimental and behavioral conditions.

The **Sines** dataset models each neuron's activity as an independent sinusoid with a random phase but consistent frequency across worms, exploring an assumption of neuron-specific 'fingerprints' in the network dynamics. Figure 9 shows the simulated 'neural' data from two 'worms' in the Sines dataset, highlighting the consistent frequency of two pairs of neurons 'observed' in both. This dataset tests models on their ability to capture simple, uncoupled oscillatory patterns, a task suited for recurrent models like LSTMs.

Conversely, the **Random Walk** dataset challenges models with inherently unpredictable neural activity simulated through a random walk process, aiming to benchmark against the naive predictor's performance which is provably the best estimator in this case. This setup serves as a critical test of model implementations, ensuring no model unjustifiably surpasses this baseline.

2.2 Model Structure

Model architectures. Our study utilizes three distinct classes of neural networks to harness different inductive biases for the prediction of future neural activity in *C. elegans*. These include Long-Short Term Memory (LSTM) networks, Transformer networks, and Feedforward networks. These architectures were chosen to represent a fundamental set of mechanisms—recurrence, attention, and feedforward processing—allowing us to assess the impact of structural and mechanistic differences on the task at hand.

Shared model structure. Each architecture is implemented within a common structural framework comprising an embedding block, a hidden 'core' module, and a linear readout layer to enable a

consistent training and evaluation procedure. This shared structure is modular allowing for the comparison of different ANN architectures by substituting only the core module (Figure 3A).

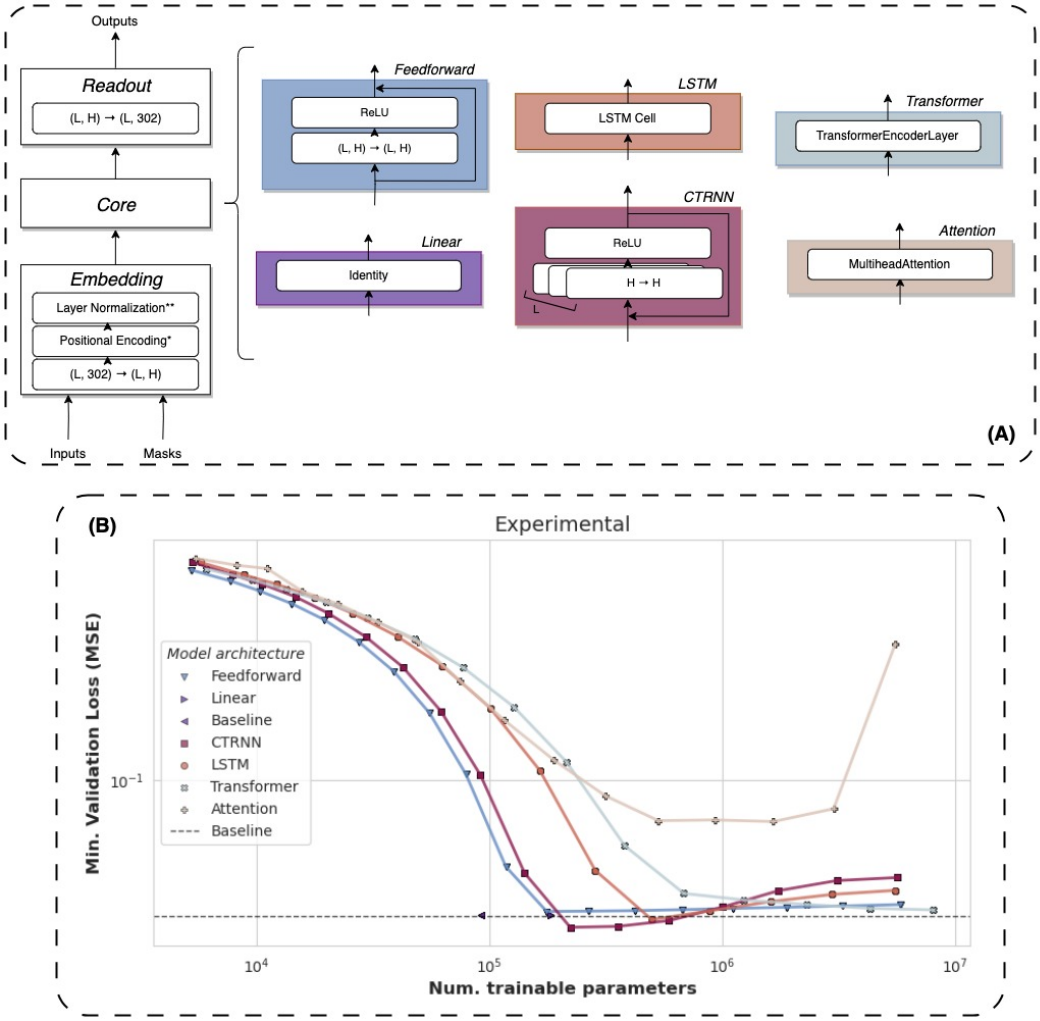

Figure 3: Model architecture and model scaling experiment. (A) The six (6) model classes we investigate share a common backbone but have differentiated core architectures: Feedforward, Linear, LSTM, CTRNN, Transformer, and Attention. Asterisks indicate modules appear in some architectures but not others. *Positional encoding present here in the Transformer and Attention models. **Layer normalization absent here in the Feedforward and Attention models. (B) The next time step prediction validation accuracy (inverse relation to loss) improves non-linearly with increasing the model sizes with some model architectures scaling better than others. Only the recurrent (CTRNN and LSTM) models beat the baseline loss. All models are trained and validated on the fixed largest training and validation datasets, respectively, made from $\mathcal{D}_{ALL}$.

1. *Embedding:* The basic embedding layer linearly projects from the 302-dimensional neural state space to a higher or lower H-dimensional latent space. Certain architectures like the Transformer may add a positional encoding and other architectures may apply layer normalization to stabilize the learning process.

2. *Core:* The core module is architecture-specific and constitutes the primary computational engine of the model. It is restricted to use a single layer to maintain simplicity and facilitate interpretability. Besides the single layer restriction, we place no constraints on the class of computations the core is allowed to use (e.g. recurrence, parallel-processing, etc.), making

our approach highly modular and scalable. The Baseline and LinearRegression prediction models are 'shallow' models, which means that they lack the 'core' module.

3. *Output Mapping:* The final component of the model is a linear projection from the latent space back to the original neural state space. The output of this layer is the predicted future neural activity $\hat{\mathbf{X}} = f_{ho}(\mathbf{Z})$, where f_{ho} is the linear transformation from the hidden to the output space.

Causal Predictions and Temporal Memory. Our models are tasked with making causal predictions, where future predictions do not rely on future inputs. We use a causal attention mask in the Transformer and Attention models. The LSTM and CTRNN models are inherently causal by definition. The Linear and Feedforward models lack access to temporal context beyond the current time step (i.e. they process each time point in a sequence independently). This essentially restricts the computation of the Linear and Feedforward models to feature regression (linear and nonlinear, respectively), providing a baseline for the importance of temporal information in self-supervised neural prediction.

Baseline Model. We use the naive predictor which posits that the next neural state will be identical to the current one as our baseline model. This baseline is a commonly used one for time series prediction tasks and it is the known optimal predictor for a random walk. Despite its simplicity, this baseline is not trivial to beat in the context of neural activity data which can often resemble, at a first-order approximation, a random process. Beating this baseline requires our ANN models to uncover and leverage complex, higher-order structures in the neural activity data beyond what is expected from a purely stochastic process. To maintain consistency with our models super-structure (Figure 3A, left), our Baseline model is implemented as'thunk' model class with no trainable parameters that simply copies its masked input as its output.

2.3 Training Objective and Loss Function

Training Objective. The models are trained under a self-supervised objective to predict the 1-time step shifted sequence of neural activity given a input sequence of neural activity of length L. The training objective is simply to minimize the mean-squared error (MSE) between the predicted and true neural activity sequences. The loss function further incorporates the boolean neuron mask to ensure that only neurons with measured data contribute to the loss computation. The mean-squared error (MSE) loss function with the boolean mask is defined as:

$$\mathcal{L}(\mathbf{X}, \hat{\mathbf{X}}; \mathbf{y}) = \frac{1}{N \times L} \sum_{i=1}^{N} \sum_{t=0}^{L-1} \mathbf{y} \odot (\mathbf{X}_i(t+1) - \hat{\mathbf{X}}_i(t+1))^2,$$

where $\mathbf{X}_i(t+1)$ is the true activity of the ith neuron at time $t+1$, $\hat{\mathbf{X}}_i(t+1)$ is the predicted activity at time $t+1$, $\mathbf{y} \in \{0,1\}^{302}$ is the boolean feature mask indicating the presence of data for neuron i, L is the sequence length used for training the model, and $N = \mathbb{1}^\top \mathbf{y}$ is the number of masked neurons (i.e. the number of labelled neurons with data).

Data Sampling and Model Evaluation. We construct the training and validation sets by sampling from each worm $n_s = 32$ sequences of length $L = 180$ time steps according to the method in subsection 2.1 **Train-Test split**. For example, a dataset $\mathcal{D}^{n_w}$ containing n_w worms would produce training and validation sets each with $n_w \times 32$ sequences. Train and validation data loaders use a batch size of 128.

Training Protocol. Models are trained up to a maximum of 500 epochs using the AdamW optimizer, with an initial learning rate of 0.001. A learning rate scheduler reduces the rate upon a validation loss plateau, with a decay factor of 0.1. Early stopping with a patience of 100 epochs is employed for efficiency. Training for all experiments was run with the same computing resources and device specifications (1 NVIDIA A100 80GB GPU).

3 RESULTS

3.1 DATA SCALING

3.1.1 MIXED DATASET SCALING

Objective. To assess how increasing the amount of training data influences the self-supervised next time step neural activity prediction.

Approach. We trained models of every architecture on incrementally larger training sets from a sequence of mixed worm datasets ranging in size from 1 to 284 worms, sampled according to subsection 3.1.1 **Mixed Dataset Scaling**. At each training set size, the models were evaluated against the same, fixed, maximum sized validation set made from $\mathcal{D}_{ALL}$. Since there are multiple possible combinatorial assignments for a mixed dataset $\mathcal{D}^{n_w}$ of size $n_w < 284$ worms, we plot the independent variable as the number time-steps in the train dataset. We controlled for the hidden size of each model architecture class so that all models were approximately matched at 0.58M (M=million) trainable parameters (see Table 2).

Results. The results shown in Figure 2B indicate that, at a fixed model size 0.580 million trainable parameters, the CTRNN models scale to scale the best (slope $= -0.65$) with dataset size, whereas the Feedforward models scale the worst (slope $= -0.40$), out of the architecture classes investigated.

3.1.2 INDIVIDUAL DATASET SCALING

Objective. To determine if models trained on mixed datasets maintained consistent scaling properties when evaluated on the individual experimental source datasets.

Approach. Utilizing the best model from the mixed dataset scaling experiment at each training dataset size, we evaluated on the largest validation set made from each of the experimental sources (refer to last paragraph of subsection 2.1 **Mixed Datasets**).

Results. Figure 2C.

3.2 MODEL SCALING

Objective. To determine the effect of model complexity, as determined by the number of trainable parameters and architecture, on the performance of self-supervised neural activity prediction in *C. elegans*.

Approach. We varied the hidden size of the 'core' architecture as a knob with which to vary the number of parameters of our models. Since each architecture (Linear, Feedforward, CTRNN, LSTM, Transformer, Attention) has a different number of trainable parameters than the others at any given hidden size, we plot the number of trainable parameters as the independent variable. The various sized models of each architecture/class were trained on the same, fixed maximum sized training dataset made from $\mathcal{D}_{ALL}$. We also validated our model scaling on the synthetic datasets described in subsection 2.1 **Synthetic Datasets** (Figure 5).

Results. Figure 3B shows the results of increasing the model size for the different architecture classes. As discussed in subsection 2.2 **Shared model structure**, the Baseline and Linear models have no hidden size dimension to vary. The Baseline model simply has no trainable parameters at all, whereas the Linear model has no trainable parameters in its 'core' module – which is just an Identity layer (Figure 3, left).

- The Baseline model achieved a minimum validation loss of 0.035413 which, by definition, is the baseline loss.
- The Linear model achieved a minimum validation loss of 0.035328.
- The Feedforward model achieved its minimum validation of 0.036377 at a hidden size of 214 corresponding to 176,210 trainable parameters.
- The Attention model achieved its minimum validation loss of 0.072782 at a hidden size of 572 corresponding to 1,658,530 trainable parameters.

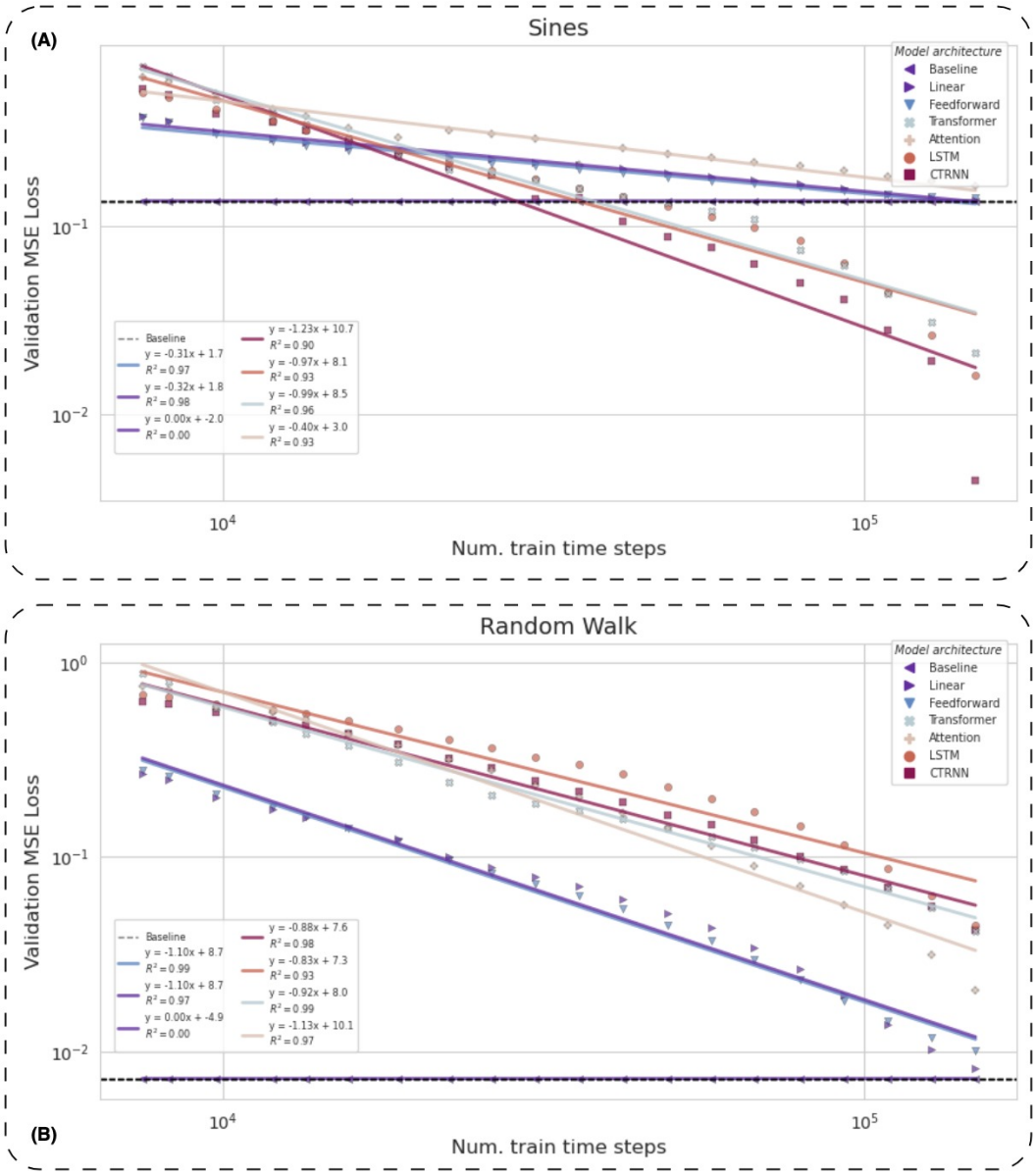

Figure 4: Dataset size scaling on synthetic datasets. All models are approximately matched in size at 580K (K=thousand) trainable parameters. (A) Dataset size scaling of the ANN models used in 2B on the Sines dataset. Beyond a modest dataset size threshold, the sequence models – recurrent and transformer – outperform the naive predictor baseline loss. (B) Dataset size scaling of the ANN models used in 2B on the Random Walk dataset. As expected, no model beats the naive predictor baseline loss since that is the optimal predictor for a random walk.

- The Transformer model achieved its minimum validation loss of 0.036321 at a hidden size of 2,124 corresponding to 55,468,562 trainable parameters.
- The LSTM model achieved its minimum validation loss of 0.034167 at a hidden size of 214 and corresponding to 498,280 trainable parameters.
- The CTRNN model achieved its minimum validation loss of 0.032193 at a hidden size of 214 corresponding to 222,434 trainable parameters.

A qualitative comparison of the auto-regressive generation capability of some of the trained models is presented Figure 10.

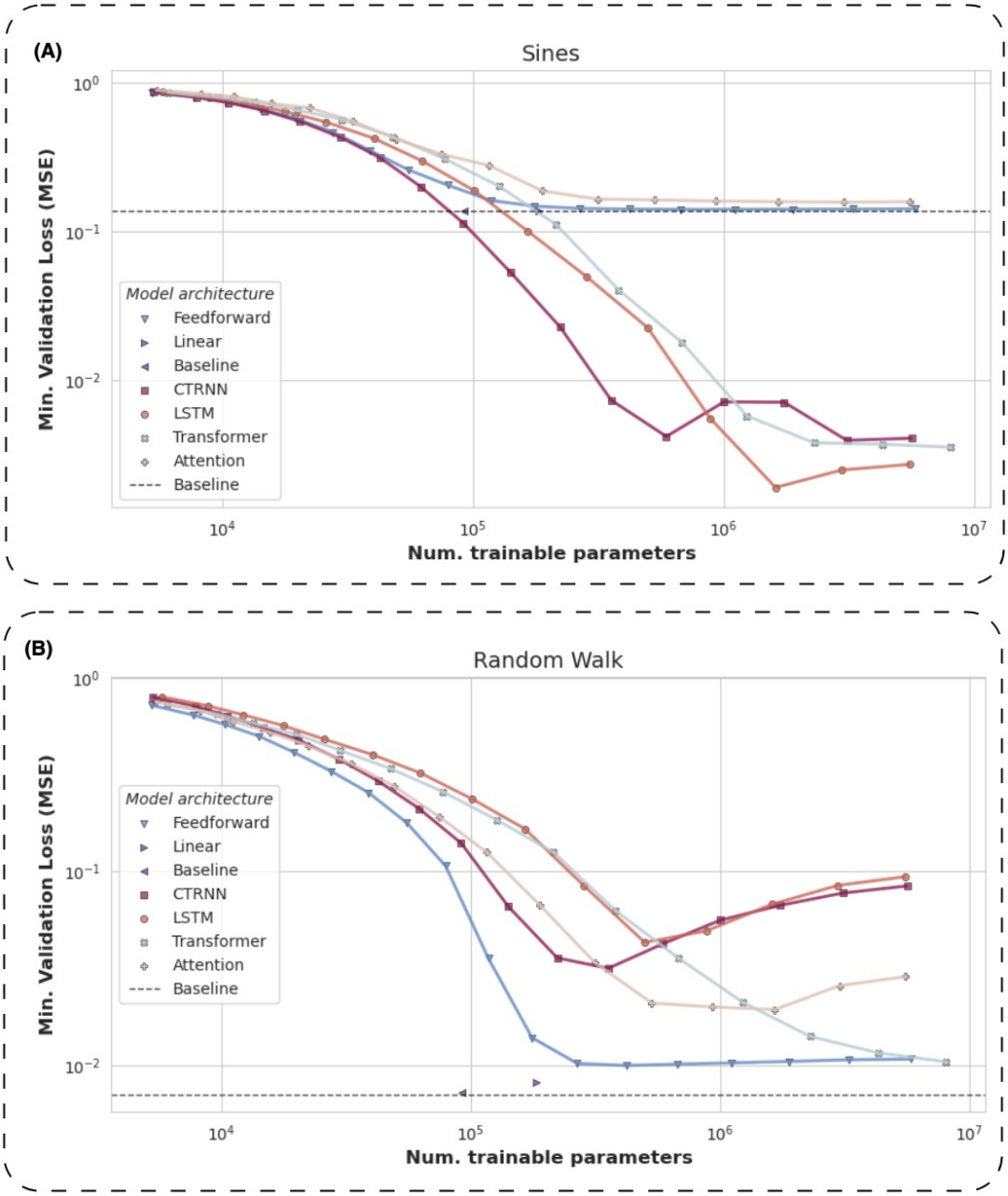

Figure 5: Model size scaling on synthetic datasets. The training and validation sets are the fixed largest possible ones made from $\mathcal{D}_{ALL}$. (A) Model size scaling of the ANN models used in 3B on the Sines dataset. Beyond modest model size threshold, the sequence models – recurrent and transformer – outperform the naive predictor baseline loss. (B) Model size scaling of the ANN models used in 3B on the Random Walk dataset. No model can beat the naive predictor baseline loss, even after training on all the available data.

3.3 Synthetic Experiments

Objective. To investigate the degree to which the scaling properties observed for the real worm datasets are a function of the underlying system (*C elegans*) versus of the networks being trained.

Approach. We trained the same ANN classes and sizes used to model the real *C. elegans* neural datasets (sections 3.1 and 3.2) on two synthetic neural datasets generated by known dynamical systems: a sinusoidal oscillator (Sines) and a stochastic process (Random Walk).

3.3.1 DATASET SIZE SCALING

Results. Figure 4.

3.3.2 MODEL PARAMETERS SCALING

Results. Figure 5.

4 DISCUSSION

This study's exploration into the scaling properties of ANNs in predicting neural activity within *C. elegans* reveals critical insights into data-driven modeling of biological neural networks. Our results demonstrate that the predictive accuracy of ANNs is significantly influenced by both the volume of training data and the complexity of the model. We observed a logarithmic decrease in mean-squared error with increased training data, a consistency that persisted across various experimental datasets. This suggests a pivotal role for data volume in enhancing model performance, with further gains possible through optimal model complexity.

Model architecture emerged as a decisive factor; recurrent models, such as LSTMs and CTRNNs, outperformed others, underscoring the importance of temporal dynamics in neural activity prediction. The experiments indicated that different datasets, despite their diverse origins, share common underlying dynamics, as evidenced by the similar scaling slopes in prediction accuracy. This finding substantiates the pooling of data from multiple sources to improve model robustness.

Challenges encountered include determining appropriate model sizes for varying data volumes and incorporating behavioral contexts into predictions, which present promising avenues for future research. Our work thus far has not fully realized the long-horizon predictive capabilities for *C. elegans* neural dynamics. However, it paves the way for more comprehensive models by highlighting the potential benefits of incorporating richer datasets and nuanced model architectures.

Future efforts will focus on refining models to capture the complexity of neural dynamics more accurately. This includes considering the integration of behaviorally-annotated data to provide additional context and investigating architectures capable of leveraging larger datasets without succumbing to the diminishing returns of over-complexity. Extension of these approaches to more complex nervous systems could offer valuable comparative insights and further the understanding of neural dynamics prediction.

In conclusion, this research contributes foundational knowledge towards the development of ANN models that more accurately reflect the intricacies of biological neural networks, bridging the gap from model organisms to broader biological contexts.

REFERENCES

Adam A Atanas, Jungsoo Kim, Ziyu Wang, Eric Bueno, Mccoy Becker, Di Kang, Jungyeon Park, Talya S Kramer, Flossie K Wan, Saba Baskoylu, Ugur Dag, Elpiniki Kalogeropoulou, Matthew A Gomes, Cassi Estrem, Netta Cohen, Vikash K Mansinghka, and Steven W Flavell. Brain-wide representations of behavior spanning multiple timescales and states in c. elegans. *Cell*, 186(19): 4134–4151.e31, sep 2023. doi: 10.1016/j.cell.2023.07.035.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, et al. Training compute-optimal large language models. *arXiv [cs.CL]*, 2022. URL `http://arxiv.org/abs/2203.15556`. arXiv:2203.15556.

Harris S. Kaplan, Oriana Salazar Thula, Niklas Khoss, and Manuel Zimmer. Nested neuronal dynamics orchestrate a behavioral hierarchy across timescales. *Neuron*, 105(3):562–576, 2020a.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv [cs.LG]*, 2020b. URL `http://arxiv.org/abs/2001.08361`. arXiv:2001.08361.

Saul Kato, Harris S. Kaplan, Tina Schrödel, Susanne Skora, Theodore H. Lindsay, Eviatar Yemini, Shawn Lockery, and Manuel Zimmer. Global brain dynamics embed the motor command sequence of caenorhabditis elegans. *Cell*, 163(3):656–669, oct 2015. doi: 10.1016/j.cell.2015.09.034.

Andrew M Leifer, Christopher Fang-Yen, Marc Gershow, Mark J Alkema, and Aravinthan D T Samuel. Optogenetic manipulation of neural activity in freely moving caenorhabditis elegans. *Nat. Methods*, 8(2):147–152, feb 2011. doi: 10.1038/nmeth.1554.

Jeffrey P. Nguyen, Frederick B. Shipley, Ashley N. Linder, George S. Plummer, Mochi Liu, Sagar U. Setru, Joshua W. Shaevitz, and Andrew M. Leifer. Whole-brain calcium imaging with cellular resolution in freely behaving caenorhabditis elegans. *Proceedings of the National Academy of Sciences*, 113(8):E1074–E1081, 2016. doi: 10.1073/pnas.1507110112.

Annika LA Nichols, Tomáš Eichler, Richard Latham, and Manuel Zimmer. A global brain state underlies c. elegans sleep behavior. *Science*, 356(6344):eaam6851, 2017.

Chethan Pandarinath, Daniel J. O'Shea, Jasmine Collins, Rafal Jozefowicz, Sergey D. Stavisky, Jonathan C. Kao, Eric M. Trautmann, et al. Inferring single-trial neural population dynamics using sequential auto-encoders. *Nature Methods*, 15(10):805–815, 2018.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8), 2019.

Francesco Randi, Anuj K. Sharma, Sophie Dvali, and Andrew M. Leifer. Neural signal propagation atlas of c. elegans. *arXiv preprint arXiv:2208.04790*, 2022.

Susanne Skora, Fanny Mende, and Manuel Zimmer. Energy scarcity promotes a brain-wide sleep state modulated by insulin signaling in c. elegans. *Cell reports*, 22(4):953–966, 2018.

Kerem Uzel, Saul Kato, and Manuel Zimmer. A set of hub neurons and non-local connectivity features support global brain dynamics in c. elegans. *Current Biology*, 32(16):3443–3459, 2022. doi: 10.1016/j.cub.2022.06.039.

Daniel L K Yamins and James J DiCarlo. Using goal-driven deep learning models to understand sensory cortex. *Nat. Neurosci.*, 19(3):356–365, mar 2016.

Daniel L K Yamins, Ha Hong, Charles F Cadieu, Ethan A Solomon, Darren Seibert, and James J DiCarlo. Performance-optimized hierarchical models predict neural responses in higher visual cortex. *Proc. Natl. Acad. Sci. U. S. A.*, 111(23):8619–8624, jun 2014.

E. Yemini, A. Lin, A. Nejatbakhsh, E. Varol, R. Sun, G.E. Mena, A.D.T. Samuel, L. Paninski, V. Venkatachalam, and O. Hobert. Neuropal: A multicolor atlas for whole-brain neuronal identification in c. elegans. *Cell*, 184:272–288.e11, 2021. doi: 10.1016/j.cell.2020.12.012.

A Appendix

A.1 Hugging Face Dataset

We have made our combined $C.elegans$ neural dataset $\mathcal{D}_{\text{ALL}}$ conveniently available as an open-source dataset on Hugging Face: `https://huggingface.co/datasets/qsimeon/celegans_neural_data`.

Table 1: Open-source *C. elegans* neural activity datasets. This table summarizes key details such as publication references, data sources, content format, worm counts, and neuron labeling statistics.

Paper Link	Database Link	Files w/ Data	Num. worms	Mean # neurons ID'd/recorded	Num. ID'd neurons (min, max)
tinyurl.com/Leifer2023	osf.io/e2syt/	exported_data.tar.gz	103	69/122	(26, 102)
tinyurl.com/Flavell2023	wormwideweb.org	YYYY-MM-DD-*.json/h5	42	88/136	(64, 115)
tinyurl.com/Uzel2022	osf.io/3vkxn/	Uzel_WT.mat	6	50/138	(46, 58)
tinyurl.com/Yemini2021	zenodo.org/records/3906530	_Activity_OH*.mat	49	110/125	(33, 179)
tinyurl.com/Kaplan2020a	osf.io/9nfhz/	Neuron2019_Data_.mat	19	36/114	(23, 51)
tinyurl.com/Skora2018	osf.io/za3gt/	WT_.mat	12	46/129	(39, 55)
tinyurl.com/Nichols2017	osf.io/kbf38/	let.mat	44	34/108	(23, 43)
tinyurl.com/Kato2015	osf.io/2395t/	WT_Stim.mat	12	42/127	(31, 51)

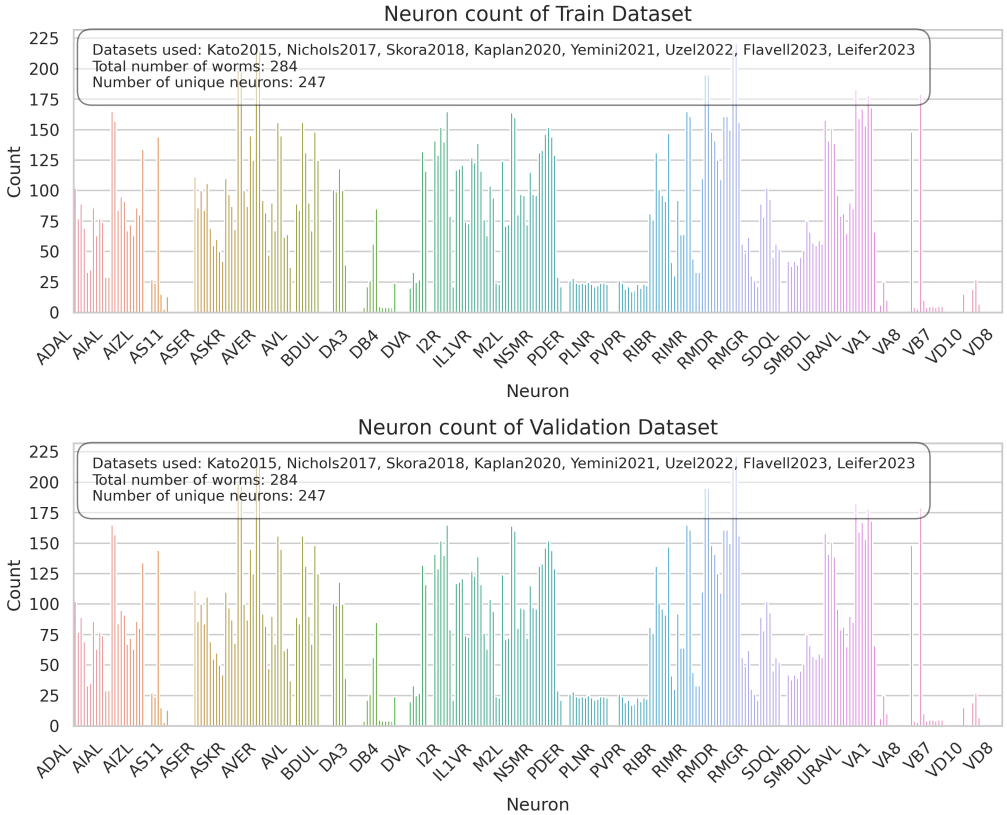

Figure 6: Distribution of neurons in $\mathcal{D}_{ALL}$. Since the train-validation split is along the temporal (not feature) dimension, the neuron distribution is the same in the train and validation sets. The combined dataset has 284 worms from the 8 experimental dataset sources, with 247 out the 302 neuron classes in $C.elegans$ represented (i.e. recorded in at least 1 worm). Of the recorded neurons, some are over-represented whereas others have been recorded in only 1 worm.

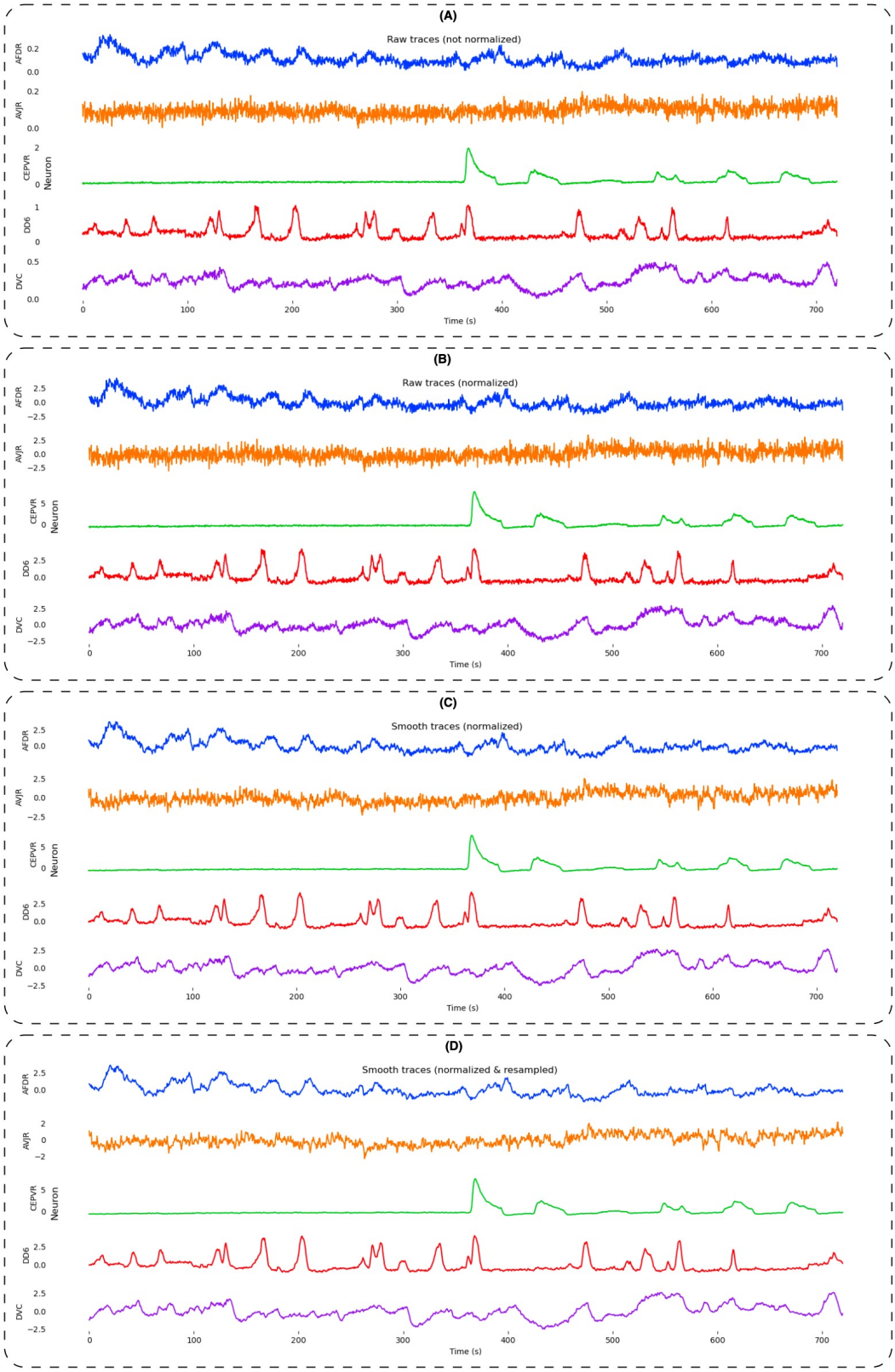

Figure 7: Example of preprocessing worm neural data. We use the example of the Kato (2015) dataset to illustrate steps of the preprocessing the neural activity data. Starting with the raw calcium fluorescence signals (A); we first standardize or z-score each neuron independently (B); then smooth using a causal filter (C); and finally resample to a fixed time step (D).

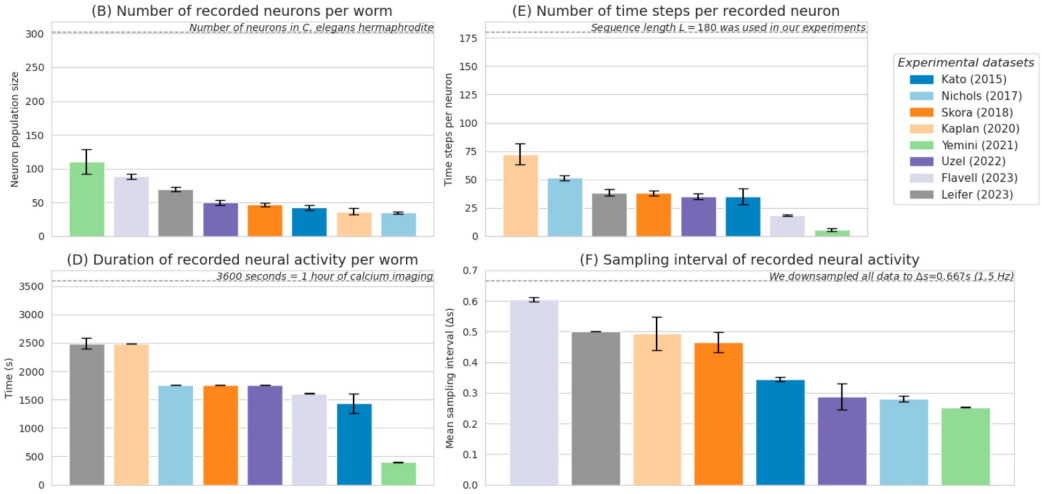

Figure 8: Experimental datasets supplementary summary statistics.

Table 2: Model Hidden Sizes and Parameter Counts. A comparison of the size in number of trainable parameters for different model architectures. In Table 2A, the models are matched for the hidden size of their internal 'core' module to compare their parameter counts. In Table 2B, the models are approximately matched for their parameters count to compare their hidden sizes.

A: Parameter Counts for Hidden Size 300			
Model Name	**Model Alias**	**Hidden Size**	**Parameter Count**
FeatureFFNN	Feedforward	300	272702
NetworkCTRNN	CTRNN	300	363302
NetworkLSTM	LSTM	300	904802
NeuralTransformer	Transformer	300	1268402
PureAttention	Attention	300	543602

B: Matched Parameter Count (Approx. 0.580M)			
Model Name	**Model Alias**	**Hidden Size**	**Parameter Count**
FeatureFFNN	Feedforward	516	580286
NetworkCTRNN	CTRNN	408	582110
NetworkLSTM	LSTM	234	582260
NeuralTransformer	Transformer	196	584186
PureAttention	Attention	312	580310

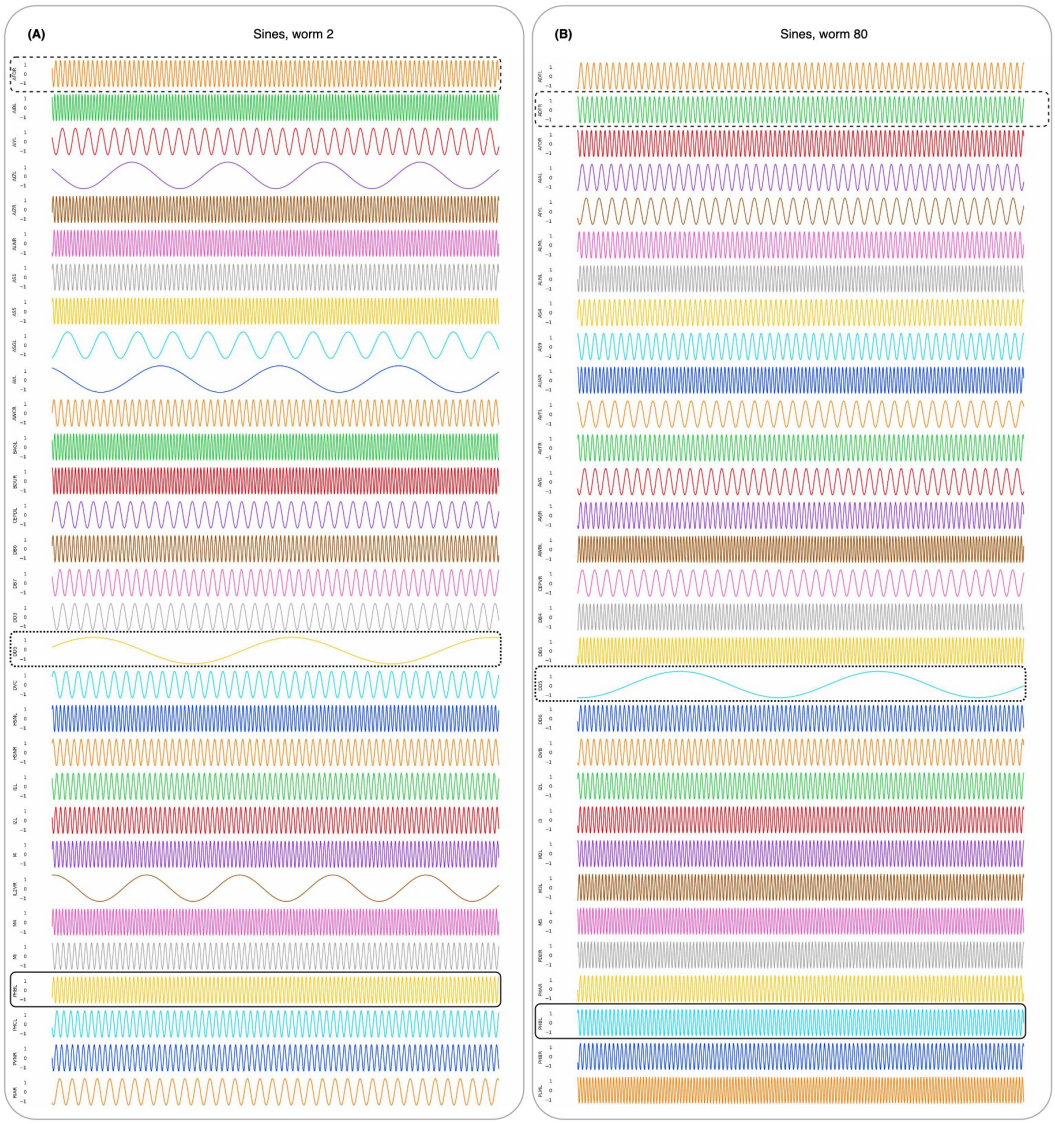

Figure 9: Synthetic neural data from 2 worms in the Sines dataset. Synthetic neural data from worm #2 (A) and worm #80 (B) is generated for a random subset of 50 neuron classes in each animal. Three neuron classes that are observed in both worms are circled in pairs to show that a neuron's frequency 'signature' is maintained across instances of the dynamical system (i.e. across animals)

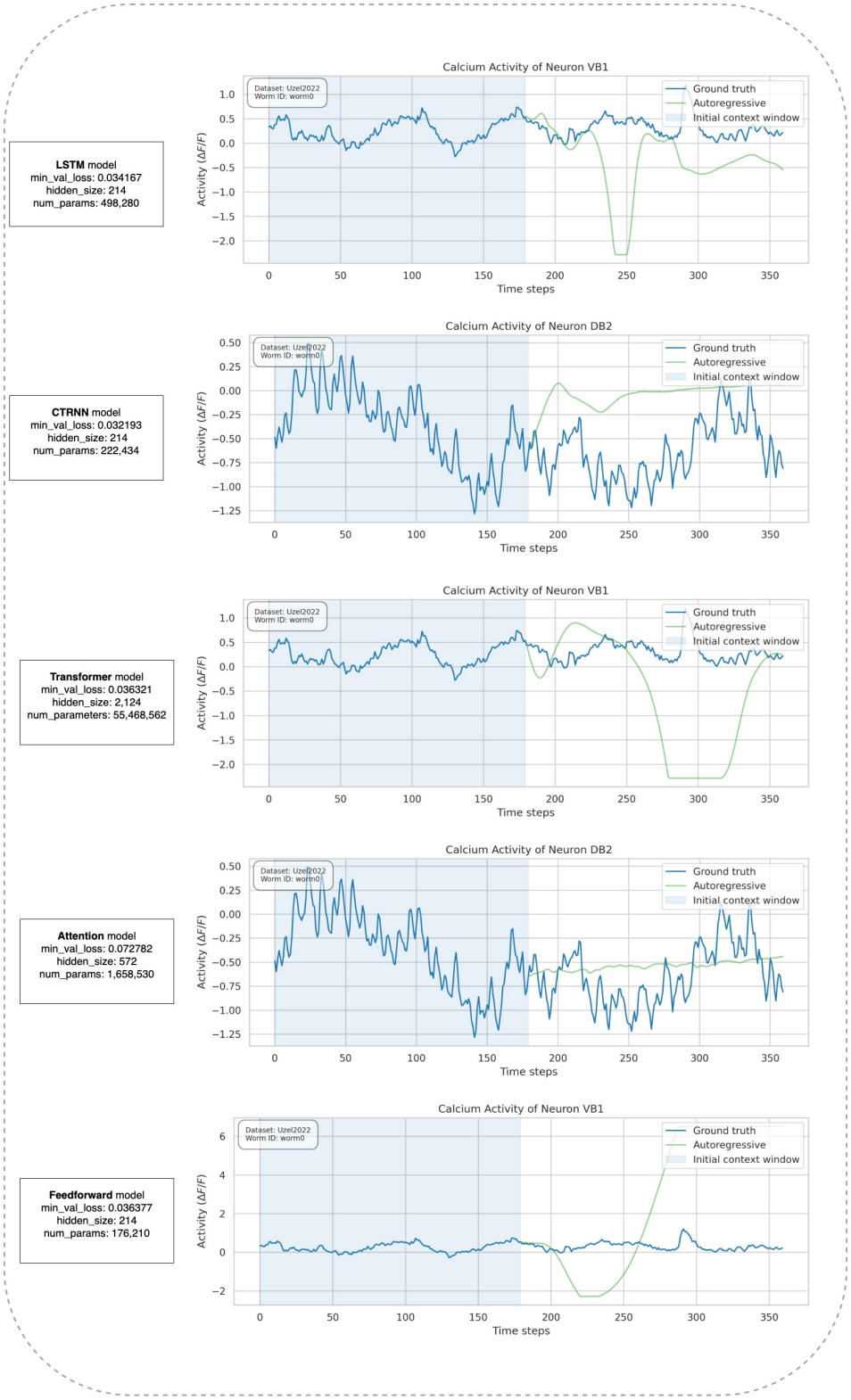

Figure 10: Auto-regressive generation of trained models on $\mathcal{D}_{Uzel}$. Models attaining the lowest validation loss for some architecture classes in Figure 3B are seeded with a length $L = 180$ time step sequence from the validation set of worm #0 in $\mathcal{D}_{Uzel}$ as context. The models are then made to autoregressively generate the next 180 time-steps by repeatedly predicting 1-time step ahead and appending this prediction to the end of a sliding context window.