ElectroPhysiomeGAN: Generation of Biophysical Neuron Model Parameters from Recorded Electrophysiological Responses

Jimin Kim[1], Qiang Liu[3,*], and Eli Shlizerman[1,2,*]

[1]Department of Electrical and Computer Engineering, University of Washington
[2]Department of Applied Mathematics, University of Washington
[3]Department of Neuroscience, City University of Hong Kong
[*]Corresponding Authors

Abstract

Recent advances in connectomics, biophysics, and neuronal electrophysiology warrant modeling of neurons with further details in both network interaction and cellular dynamics. Such models may be referred to as ElectroPhysiome, as they incorporate the connectome and individual neuron electrophysiology to simulate neuronal activities. The nervous system of *C. elegans* is considered a viable framework for such ElectroPhysiome studies due to advances in connectomics of its somatic nervous system and electrophysiological recordings of neuron responses. In order to achieve a simulated ElectroPhysiome, the set of parameters involved in modeling individual neurons need to be estimated from electrophysiological recordings. Here, we address this challenge by developing a novel deep generative method called ElectroPhysiomeGAN (EP-GAN), which once trained, can instantly generate parameters associated with the Hodgkin-Huxley neuron model (HH-model) for neurons with graded potential response. The method combines Generative Adversarial Network (GAN) architecture with Recurrent Neural Network (RNN) Encoder and can generate an extensive number of parameters (>170) given the neuron's membrane potential responses and steady-state current profiles. We validate our method by estimating HH-model parameters for 200 synthetic neurons with graded membrane potential followed by 9 experimentally recorded neurons (where 6 of them newly recorded) in the nervous system of *C. elegans*. Compared to other methods, EP-GAN is advantageous in both accuracy of generated parameters and inference speed. In addition, EP-GAN preserves performance when provided with incomplete membrane potential responses up to 25% and steady-state current profiles up to 75%. EP-GAN is designed to leverage the generative capability of GAN to align with the dynamical structure of HH-model, and thus able to achieve such performance.

1 Introduction

Models of the nervous system aim to achieve biologically detailed simulations of large-scale neuronal activity through the incorporation of both structural connectomes (connectivity maps) and individual neural dynamics. The nervous system of *Caenorhabditis elegans* (*C. elegans*) is considered a framework for such a model as the connectome of its somatic nervous system for multiple types of interaction is mapped [1, 2, 3]. In addition to the connectome, advances in electrophysiological methodology allow the recording of whole-cell responses of individual neurons. These advances provide biophysically relevant details of individual neuro-dynamical properties and warrant a type of model for the *C. elegans* nervous system incorporating both the connectomes and individual biophysical processes of neurons. Such a model could be referred to as *ElectroPhysiome*, as it incorporates a layer of individual neural dynamics on top of the layer of inter-cellular interactions facilitated by the connectome.

The development of models that are further biophysically descriptive for each neuron, i.e., modeling

neurons using the Hodgkin-Huxley type equations (HH-model) require fitting a large number of parameters. For a typical single neuron, these parameters could be tuned via local optimizations of individual ion channel parameters estimated separately to fit their respective *in-vivo* channel recordings such as activation/inactivation curves [4, 5, 6, 7, 8, 9]. Such method requires multiple experiments to collect each channel data and when such experiments are infeasible, the parameters are often estimated through hand-tuning. In the context of developing the ElectroPhysiome of *C. elegans*, the method would have to model approximately 300 neurons each including an order of hundreds of parameters associated with up to 15 to 20 ionic current terms (with some of them having unknown ion channel composition), which would require large experimental studies [7]. Furthermore, the fitted model may not be the unique solution as different HH-parameters can produce similar neuron activity [10, 11, 12, 13]. As these limitations also apply for general neuron modeling tasks beyond *C. elegans* neurons, there has been an increasing search for alternative fitting methods requiring less experimental data and manual interventions.

A promising direction in associating model parameters with neurons has been the simultaneous estimation of all parameters of an individual neuron given only electrophysiological responses of cells, such as membrane potential responses and steady-state current profiles. Such an approach requires significantly less experimental data per neuron and offers more flexibility with respect to trainable parameters. The primary aim of this approach is to model macroscopic cell behaviors in an automated manner. Indeed, several methods adopting the approach have been introduced. Buhry et al 2012 and Laredo et al 2022 utilized the Differential Evolution (DE) method to simultaneously estimate the parameters of a 3-channel HH-model given a whole-cell membrane potential responses recording [14, 15]. Naudin et al 2022 further developed the DE approach and introduced the Multi-objective Differential Evolution (DEMO) method to estimate 22 HH-parameters of 3 non-spiking neurons in *C. elegans* given their whole-cell membrane potential responses and steady-state current profiles [16]. The study was a significant step toward modeling whole-cell behaviors of *C. elegans* neurons in a systematic manner. From statistical standpoint, Wang et al 2022 used the Markov-Chain-Monte-Carlo method to obtain the posterior distribution of channel parameters for HH-models featuring 3 and 8 ion channels (2 and 9 parameters respectively) given the simulated membrane potential responses data [17]. From an analytic standpoint, Valle et al 2022 suggested an iterative gradient descent based method that directly manipulates HH-model to infer 3 conductance parameters and 3 exponents of activation functions given the measurements of membrane potential responses [18]. Recent advances in machine learning gave rise to deep learning based methods which infer steady-state activation functions and posterior distributions of 3-channel HH-model parameters inferred by an artificial neural network model given the membrane potential responses data [19, 20].

While these methods suggest that simultaneous parameter estimation from macroscopic cell data is indeed possible through a variety of techniques, it is largely unclear whether they can be extended to fit more detailed HH-models featuring a large number of parameters [7]. Furthermore, for most of the above methods, the algorithms require an independent (from scratch) optimization process for modeling each individual neuron, which leads to expensive computation when scaling them to a large number of neurons.

Here we propose a new data-driven model approach that aims to address these aspects for the class of non-spiking neurons, which constitute the majority of neurons in *C. elegans* nervous system [21]. Specifically, we develop a deep generative neural network model (GAN) combined with a recurrent neural network (RNN) encoder called ElectroPhysiomeGAN (EP-GAN), which directly maps electrophysiological recordings of a neuron, e.g., membrane potential responses and steady-state current profiles, to HH-model parameters of arbitrary dimensions (Figure 1). EP-GAN can be trained with simulation data informed by a generic HH-model encompassing a large set of arbitrary ionic current terms and thus can generalize its modeling capability to multiple neurons. Unlike

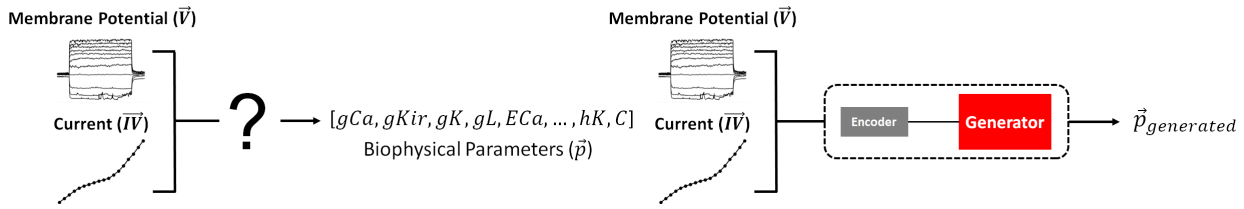

Figure 1: **Estimation of HH-model parameters from membrane potential and steady-state current profiles.** Given the membrane potential responses (V) and steady-state current profiles (IV) of a neuron, the task is to predict biophysical parameters of Hodgkin-Huxley type neuron model (Left). We use Encoder-Generator approach to predict the parameters (Right)

typical GAN architecture trained solely with adversarial losses, we propose to implement additional supervised loss for reconstructing the given membrane potential responses and current profiles from generated parameters, thus improving the accuracy of the generative model. In addition, due to RNN component of EP-GAN, the approach supports input data with missing features such as incomplete membrane potential responses and current profiles.

We validate our method to estimate HH-model parameters of 200 synthetic non-spiking neurons and apply to three previously recorded non-spiking neurons of *C. elegans*, namely RIM, AFD, and AIY. Studies have shown that membrane potential responses of these neurons can be well modeled with typical HH-model formulations with 22 parameters [16, 22]. We show that when trained with a more detailed HH-model consisting of 15 ionic current terms resulting with 176 trainable parameters, EP-GAN can produce parameters modeling their membrane potential responses with better accuracy in reconstruction of membrane potential and significantly faster speed than existing algorithms such as Multi-Objective DE and Genetic Algorithms. Through ablation studies on input data, we show that EP-GAN preserves its prediction capability even when provided with incomplete membrane potential responses and current profiles. We also perform ablation studies on the model architecture components to elucidate each component contributions toward the accuracy of the predicted parameters. To further test EP-GAN, we estimate HH-model parameters for 6 newly recorded non-spiking *C. elegans* neurons: AWB, AWC, URX, RIS, DVC, and HSN, whose membrane potential responses were not previously modeled.

Our results suggest that EP-GAN can learn a translation from electrophysiologically recorded responses and propose projections of them to parameter space. EP-GAN method is currently limited to non-spiking neurons in *C. elegans* as it was designed and trained with HH-model describing the ion channels of these neurons. EP-GAN applications can be potentially extended toward resolving neuron parameters in other organisms since non-spiking neurons are found within animals across different species [23, 24, 25, 26, 27, 28, 29, 30].

2 Results

We evaluate EP-GAN with respect to 4 current evolutionary algorithms introduced for general parameter estimation: NSGA2, DEMO, GDE3, and NSDE. Specifically, NSGA2 is a variant of the Genetic Algorithm utilizing a non-dominated sorting survival strategy, and is a commonly used benchmark algorithm for multi-objective optimization problems including HH-model fitting [31, 32, 33]. DEMO, GDE3 and NSDE are variants of multi-objective differential evolution (DE) algorithms that combine DE mutation with pareto based ranking and crowding distance sorting applied in NSGA2's survival strategy [34, 35, 36]. These methods have been proposed as more effective methods than direct DE for estimation of HH-model parameters [16, 37, 38, 39]. In particular, DEMO has been successfully applied to estimate HH-model parameters for non-spiking neurons in *C. elegans* and we include additional comparison studies with EP-GAN in the sub-section *Additional Comparisons with DEMO*

Method	HH simulations	Overall Error	RIM	AFD	AIY
DEMO	32k (11k · 3)	28.1mV	40.1mV	10.6mV	33.5mV
GDE3	32k (11k · 3)	23.4mV	28.4mV	14.1mV	27.8mV
NSDE	32k (11k · 3)	20.9mV	19.9mV	11.8mV	30.9mV
NSGA2	32k (11k · 3)	11.2mV	13.9mV	10.2mV	9.6mV
EP-GAN	**32k (32k · 1)**	**6.7mV**	**7.5mV**	**6.4mV**	**6.2mV**

Table 1: **The lowest membrane potential responses errors achieved by each method for validation scenarios.** HH simulations represent the total number of simulations each method used for training. For all scenarios, the error represents the average RMSE between ground truth and predicted membrane potential responses in each time point across all membrane potential responses traces.

[16].

All 4 methods support multi-objective optimization over large parameter space allowing them to have similar training setups as EP-GAN. All 4 methods were implemented in Python where DEMO uses the algorithm proposed in [16] whereas NSGA2, GDE3 and NSDE were implemented using Pymoo package [40]. For the HH-model to be estimated, we use the formulation introduced in [7]. The model features 15 ion channels that were found in *C. elegans* and other organisms expressing homologous channels and is considered the most detailed neuron model for the organism. The model has a total of **216** parameters, of which 176 have approximate ranges with lower and upper bounds that can be inferred from the literature [8, 22, 41]. We thus target these 176 parameters as trainable parameters for all methods.

Validation Scenarios. We validate EP-GAN trained on 32,000 (32k) training samples (i.e., 32k HH simulated neurons) by modeling 200 synthetic neurons. These neurons are randomly simulated using HH-model and divided into two groups - 100 neurons sampled from training data and other 100 neurons from test data (i.e. not part of the training set). At each training epoch, EP-GAN generates parameters for the neurons being evaluated and for each neuron, the parameters which achieved the lowest RMSE error (detailed descriptions of its calculation provided in supplementary) of membrane potential responses w.r.t. ground truth are reported at the end of training. These results in EP-GAN scoring overall error of **5.84mV** and **5.81mV** for neurons sampled from training data and testing data respectively (Table S1).

We then model 3 experimentally recorded non-spiking neurons of *C. elegans* - RIM, AFD, and AIY, and compare the performance of EP-GAN with NSGA2, DEMO, GDE3, and NSDE. Recorded data for these neurons is publicly available and their modeling descriptions were elaborated by previous works, making them common benchmarks [16, 22, 42]. For all methods, the total training data size of 32,000 (32k HH simulated neurons) is used. i.e., EP-GAN is given 32k training samples from synthetic training scenario, and for NSGA2, DEMO, GDE3, and NSDE, each algorithm is allocated with up to 11k HH simulations for each neuron during its search phase, thus adding up to total of 32k simulations for all three neurons. For each search phase of NSGA2, DEMO, GDE3 and NSDE, the parameter set candidates are recorded at each iteration, and the candidate achieving the lowest membrane potential responses error is reported at the end of the search phase. For Multi-objective DE methods (DEMO, GDE3 and NSDE2), we follow the same configurations used in the literature to set their parameters and optimization scheme [16]. Specifically, we use the root-mean-square error normalized to the noise level for the objective functions for both membrane potential responses and steady-state current and set crossover parameter CR and scale factor F to 0.3 and 1.5 respectively. For all 4 methods, NP is set to 600 with total 18 iterations (i.e., $\sim 11k$ HH simulations).

Figure 2 shows comparison between reconstructed membrane potential responses of predicted parameters vs ground truth membrane potential responses for all 4 methods. We include recon-

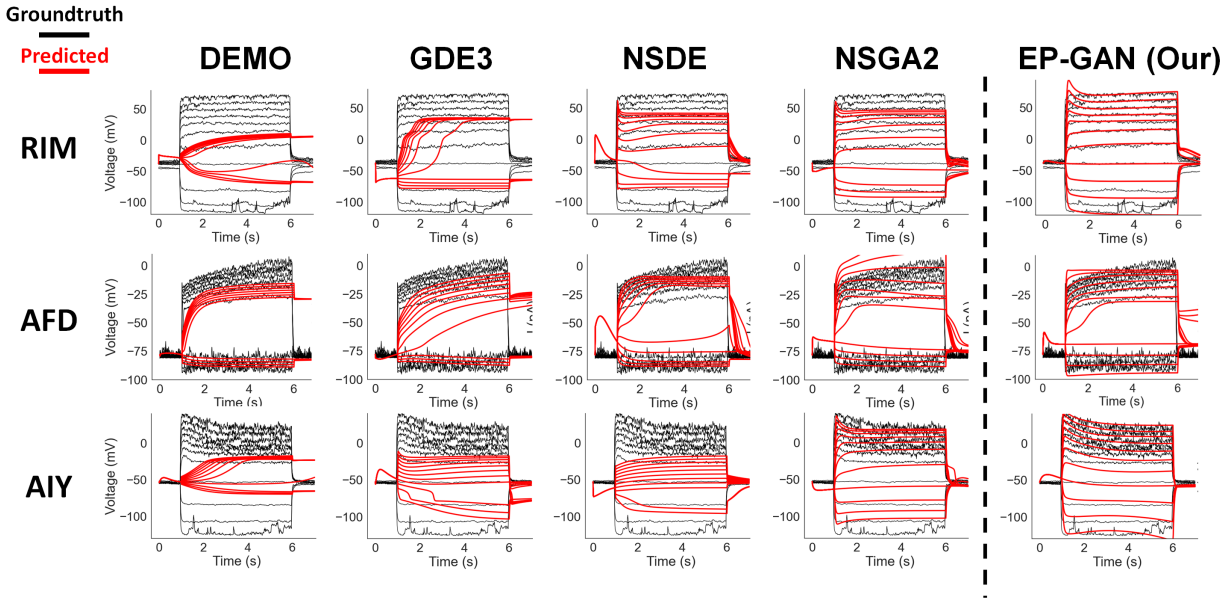

Figure 2: **Comparison of membrane potential responses reconstructed from predicted parameters (validation scenarios)**. Each row and column corresponds to predicted neuron and method respectively. For each neuron we show ground truth membrane potential responses (Black) against the reconstructed membrane potential responses (red) from the predicted parameters.

structed steady-state current profiles in supplement Figure S1. Figure 2 illustrates that EP-GAN can reconstruct membrane potential responses closer to ground truth responses than those of NSGA2, DEMO, GDE3 and NSDE. Indeed, when comparing the overall RMSE error (Table 1) between ground truth and reconstructed membrane potential responses, EP-GAN overall error (6.7mV) is 40% lower than that of NSGA2 (11.2mV) followed by NSDE (20.9mV), GDE3 (23.4mV) and DEMO (28.1mV). The overall error of 6.7mV is also in the similar ballpark of $\sim$ 5.8mV recorded during synthetic validation. Among the three neurons used for prediction, EP-GAN showed the best accuracy for AIY neuron (6.2mV), followed by AFD (6.4mV) and RIM (7.5mV).

Ablation Studies To test the robustness of EP-GAN when incomplete input data is given, we provide the model with membrane potential responses and steady-state current profiles with miss-

Input Data Ablation	Overall Error	RIM	AFD	AIY
EP-GAN (25% V)	32.5mV	38.3mV	23.9mV	35.4mV
EP-GAN (50% V)	28.0mV	43.0mV	12.4mV	28.6mV
EP-GAN (75% V)	16.8mV	23.4mV	13.4mV	13.6mV
EP-GAN (25% IV)	9.6mV	10.8mV	10.3mV	7.7mV
EP-GAN (50% IV)	9.4mV	10.4mV	10.7mV	7.1mV
EP-GAN (75% IV)	9.4mV	9.3mV	11.4mV	7.6mV
EP-GAN (Full)	**6.7mV**	**7.5mV**	**6.4mV**	**6.2mV**
Loss Ablation	Overall Error	RIM	AFD	AIY
EP-GAN (A)	21.2mV	26.5mV	19.1mV	17.9mV
EP-GAN (A, IV)	18.1mV	15.6mV	17.1mV	21.5mV
EP-GAN (A, IV, V)	**6.7mV**	**7.5mV**	**6.4mV**	**6.2mV**

Table 2: **Ablation studies**. **Top**: membrane potential responses errors achieved for EP-GAN when provided with incomplete input data. **Bottom**: membrane potential responses errors achieved for EP-GAN upon using only adversarial loss (A) and using adversarial + current reconstruction loss (A, IV) and all three loss components (A, IV, V)

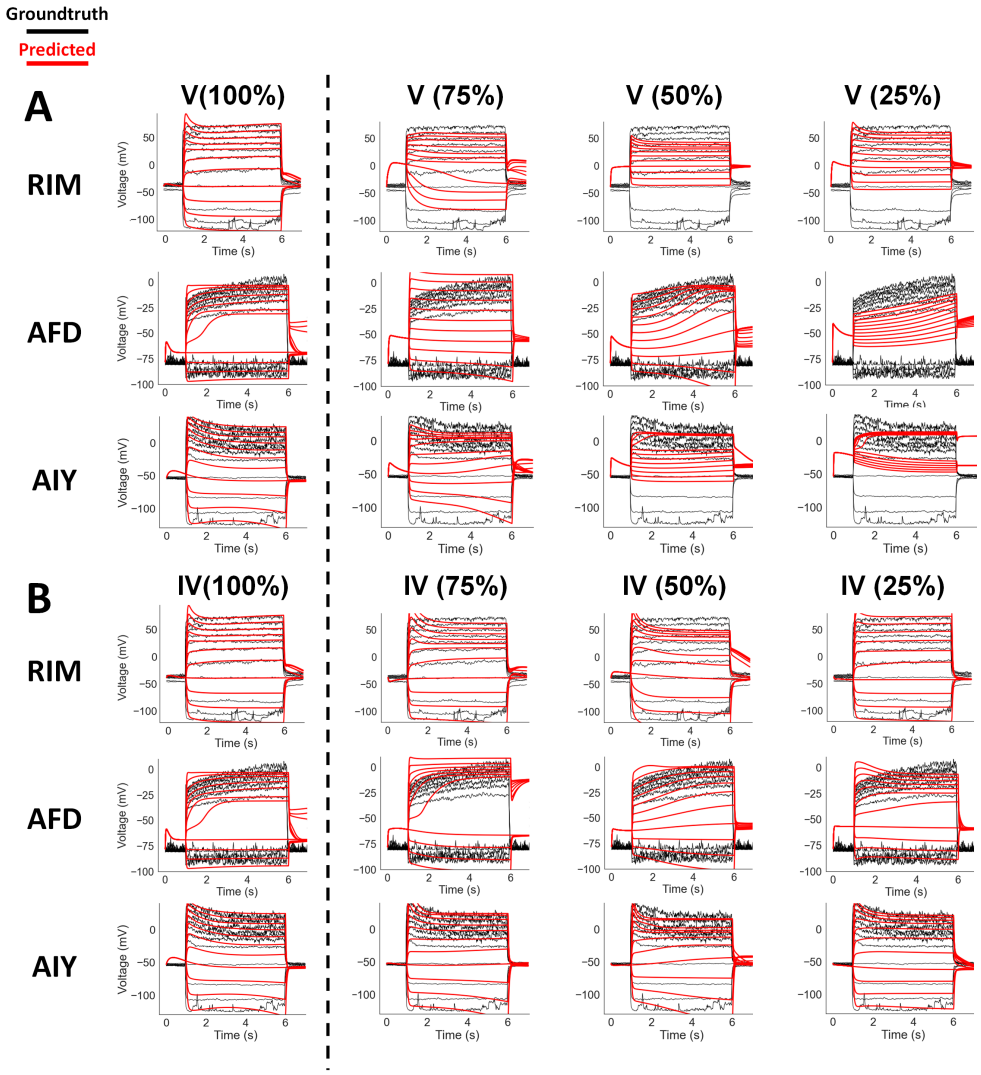

Figure 3: **Input data ablation on EP-GAN**. **A**: Reconstructed membrane potential responses when given with incomplete membrane potential responses data. Percentages in parenthesis represent the remaining portion of input membrane potential responses trajectories. **B**: Reconstructed membrane potential responses when given with incomplete current profile.

ing data points. For each membrane potential responses and current profile, the data is reduced by 25%, 50%, and 75% each. For membrane potential responses data, the ablation is done on stimulus space where a 50% reduction corresponds to removing half of the membrane potential responses traces each associated with a stimulus. For steady-state current profile, we remove the first n-data points where they are instead extrapolated using linear interpolation with existing data points.

Our results show that the quality of the predicted parameters depends on membrane potential responses (Figure 3A, Table 1 Row 2). In particular, EP-GAN preserves reasonable accuracy up to a 25% reduction in membrane potential responses but becomes less accurate when it increases to 50%. With respect to steady-state current profiles, those have an effect on the accuracy (Figure 3B, Table 2 Row 1) but the effect is more minor than that of reducing membrane potential responses and the error doesn't vary when scale is reduced. The results imply that the membrane potential responses play a major role in the performance of the model.

We also performed ablation studies on model architecture by removing each loss component of the Generator module, allowing us to evaluate the relative contribution of each loss to accuracy. From

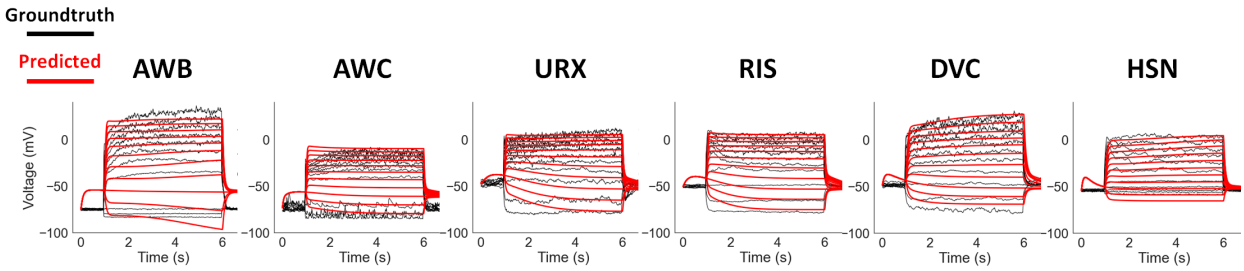

Figure 4: **Illustration of membrane potential responses reconstructed from predicted parameters (prediction scenarios)**. Membrane potential responses trajectories reconstructed from EP-GAN predicted parameters (red) overlaid on top of the ground truth recording membrane potential responsess (black).

Method	HH simulations	AWB	AWC	URX	RIS	DVC	HSN
DEMO	32k (5.4k · 6)	62.6mV	14.7mV	13.0mV	27.3mV	23.7mV	29.5mV
GDE3	32k (5.4k · 6)	54.3mV	21.1mV	16.9mV	27.9mV	33.9mV	21.8mV
NSDE	32k (5.4k · 6)	31.3mV	21.0mV	20.6mV	20.8mV	27.6mV	18.6mV
NSGA2	32k (5.4k · 6)	44.6mV	15.9mV	14.8mV	21.6mV	20.8mV	12.7mV
EP-GAN	**32k (32k · 1)**	**7.7mV**	**7.4mV**	**4.7mV**	**4.2mV**	**4.1mV**	**4.0mV**

Table 3: **The lowest membrane potential responses errors achieved by each method for prediction scenarios.** HH simulations represent the total number of simulations each method used for training.

Table 2 bottom, we see that removing the membrane potential loss term (V) results in a significant loss in performances for all three neurons as expected, where the reduction for AIY neuron is the most prominent. Upon removing the steady-state current reconstruction loss term (IV) in addition to membrane potential reconstruction loss, we see further reduction in overall performance. These results highlight the significance of the reconstruction losses in aligning the Generator to produce the desired outputs.

Prediction Scenarios. To further test if EP-GAN can generalize to other testing scenarios, we applied EP-GAN on 6 additional neurons - AWB, AWC, URX, RIS, DVC, and HSN, which recording data is novel and these neurons were not previously modeled. Similar to modeling RIM, AFD, and AIY, EP-GAN with identical training data, size of 32k, was used where membrane potential responses were rearranged for each training sample according to neuron specific current-clamp protocol. For NSGA2, DEMO, GDE3 and NSDE, each algorithm was given 5.4k HH simulations for each neuron, adding up to 32k total simulations with 6 neurons. From Figure 4, we observe that EP-GAN is indeed able to predict parameters that closely recover ground truth membrane potential responses for all 6 neurons. Upon inspecting the RMSE error, EP-GAN significantly outperforms NSGA2, DEMO, GDE3 and NSDE with 50% - 80% less error on average, and also records smaller overall error of 5.35mV than EP-GAN error for validation scenarios ($\sim$ 6.7mV) and on synthetic data ($\sim$ 5.8mV). These results indicate that the predictive capability of EP-GAN can generalize to a novel set of neurons with a relatively small training data size of 32k without need for retraining if same format

Method	HH simulations	Data Generation	Training	Inference (3 neurons)
DEMO	32k (11k · 3)	N/A	N/A	18h (5.8h · 3)
GDE3	32k (11k · 3)	N/A	N/A	18h (5.8h · 3)
NSDE	32k (11k · 3)	N/A	N/A	18h (5.8h · 3)
NSGA2	32k (11k · 3)	N/A	N/A	18h (5.8h · 3)
EP-GAN	**32k (32k · 1)**	**$\sim$10h**	**6.3h**	**1.4e-6h (0.005s)**

Table 4: **Inference Time Comparison between methods for validation scenarios.** For EP-GAN, a training consists of 100 epochs.

of recordings is used.

Parameter Inference Time. In addition to its capability to generalize over multiple neurons, we evaluated EP-GAN for scalability by comparing its overall inference time to other methods. Indeed, for estimation tasks involving many neurons, it is essential that the method is scalable so that the predictions are done within a reasonable time. In Table 4 we list the computation times involved in each approach for validation scenarios. We show that while EP-GAN requires time for initial data generation and training, it is of orders of magnitude faster in the actual estimation process. As an example, given a hypothetical task of modeling all 279 somatic neurons in the *C. elegans* nervous system, it would take DEMO, GDE3, NSDE and NSGA2 more than 67 days (assuming 11k HH simulations per neuron) whereas, for EP-GAN, given that all neurons have identical recording protocol, the process would be done within a day under the similar training setup. For larger number of neurons, computational requirement of existing methods would grow linearly while EP-GAN would require constant time to complete the inference. This is due to EP-GAN being a neural network which supports parallel processing of inputs, allowing it to take multiple neuron profiles and output corresponding parameters in a single forward pass [43].

Additional Comparisons with DEMO. EP-GAN aim and primary application is similar to the algorithm introduced in [16] which utilizes DEMO (DE for Multi-objective Optimization). Both methods aim to estimate HH-model parameters for non-spiking neurons of *C. elegans*. DEMO is applied to 3 channels HH equations ($I_{Ca,p} + I_{Kir} + I_{K,t} + I_L$-model and $I_{Ca,t} + I_{Kir} + I_{K,p} + I_L$-model) consisting of 22 parameters to model RIM, AFD and AIY. This is much lower dimension model than we use previously. In addition to validating EP-GAN on more detailed model, we evaluate its performance on the same model for which DEMO was applied. We observe that for all HH simulation sizes used for training (32k, 64k, 96k), EP-GAN achieves membrane potential response errors on par with those obtained with DEMO (Table S2). While the best overall error for EP-GAN (4.9mV) is higher than that of DEMO (3.2mV), EP-GAN requires significantly less number of HH simulations (96k $\cdot$ 2 (EP-GAN) vs 12,000k $\cdot$ 3 (DEMO)). To further investigate the correlation between the number of HH simulations and DEMO performance, we re-evaluate DEMO with respect to the detailed HH-model of [7] when given with 64k HH simulations ($\sim$ 107 iterations, $NP = 600$) for each neuron instead of 11k and 5.4k simulations conisdered in validation and prediction scenarios. From Table S3, we see that while DEMO overall error is improved, EP-GAN trained with a single training set of 32k HH simulations maintains better results for both validation (6.7mV (EP-GAN) vs 26.8mV (DEMO)) and prediction scenarios (5.35mV (EP-GAN) vs 24.2mV (DEMO)). These results suggest that EP-GAN can be applied to different HH-models and generate parameters with consistent accuracy.

3 Methods

We divide the method section into two parts. In the first part we describe the detailed architecture of EP-GAN including its sub-modules with the protocol for training. In the second part, we describe the dataset and experimental protocol of novel neuron recording of AWB, AWC, URX, RIS, DVC, and HSN from *C. elegans* nervous system.

3.1 Architecture of EP-GAN

Deep Generative Model for Parameter Prediction. EP-GAN receives neuron recording data such as membrane potential responses and steady-state current profiles and generates a parameter set that is associated with them in terms of simulating the inferred HH-model and comparing the simulated outcome with the inputs (Figure 1,5). We choose a deep generative model approach, specifically Generative Adversarial Network (GAN) as a base architecture of EP-GAN. The key

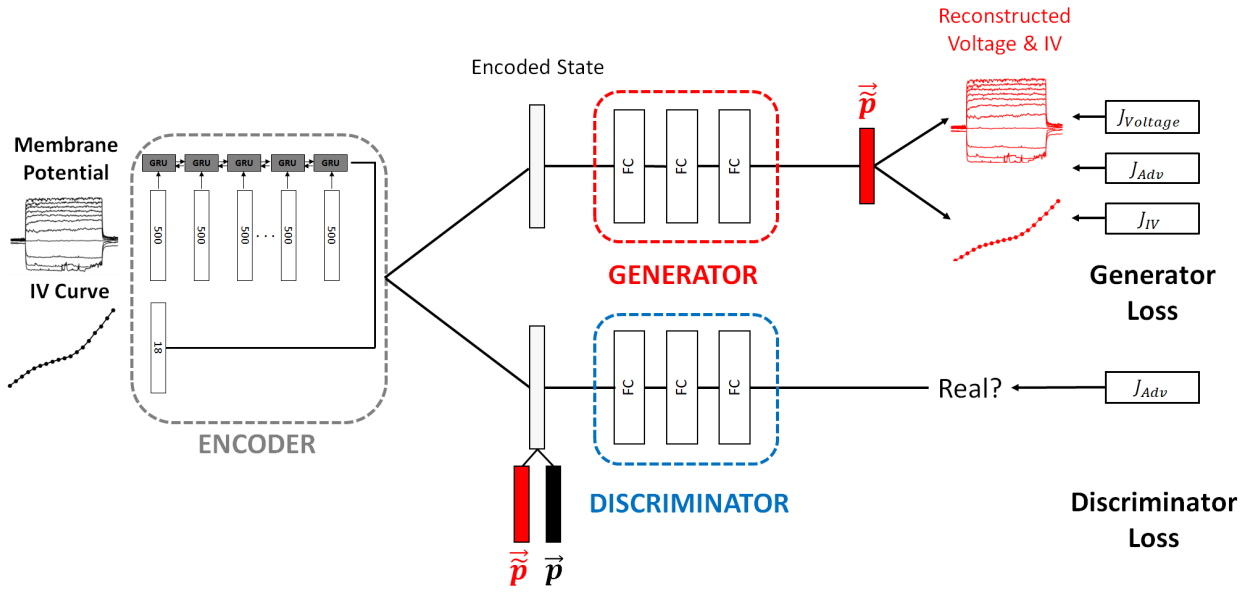

Figure 5: **Architecture of EP-GAN**. The architecture consists of Encoder, Generator and Discriminator. Encoder compresses the membrane potential responses into a 1D vector (i.e., latent space) which is then concatenated with 1D steady-state current profile to be used as an input to both generator and discriminator. Generator translates the latent space vector into a vector of parameters $\vec{\tilde{p}}$ and the Discriminator outputs a scalar measuring the similarity between generated parameters ($\vec{\tilde{p}}$) and ground truths ($\vec{p}$). The Generator is trained with adversarial loss supplemented by reconstruction losses for both membrane potential responses and steady-state current profiles. The Discriminator is trained with discriminator adversarial loss only. Generator and Discriminator follow the architecture of Wasserstein GAN with gradient penalty (WGAN-GP) for more stable learning.

advantage of GAN is that it is designed to generate artificial data that closely resembles the real data. Such generative nature of GAN is advantageous for addressing the multi-modal nature of our problem, where the parameter solution is not guaranteed to be unique for a given neuron recording. Indeed, several computational works attempting to solve inverse HH-model have pointed out the ill-posed nature of the parameter solutions [5, 10, 11, 12, 13, 18, 44, 45]. Our approach is therefore leveraging GAN to learn a *domain of parameter sets* compatible with a neuron recordings instead of mapping onto a single solution. GAN consists of two separate networks, Generator and Discriminator. The goal of the Generator is to generate outputs that are indistinguishable from real data whereas the Discriminator's goal is to distinguish outputs that are generated by the generator against real data. Throughout training, the Generator and the Discriminator compete with each other (zero-sum game) until they both converge to optimal states [46]. The particular architecture we use is Wasserstein GAN with gradient penalty (WGAN-GP), a variant of GAN architecture offering more stable training and faster convergence [47].

Encoder Module. In addition to Generator and Discriminator, we implement Encoder module which pre-processes the input membrane potential responses for Generator and Discriminator. (Figure 5 left). Specifically, Encoder serves two roles: i) Compression of membrane potential responses traces along the stimulus space, thus reducing its dimension from 2-dimensional to 1-dimensional, and ii) Translating membrane potential responses traces into a latent space encoding a meaningful internal representation for the Discriminator and Generator. The Encoder module uses Gated Recurrent Units (GRU) architecture, a variant of Recurrent Neural Network (RNN) to perform this task [48]. Each input sequence to a GRU cell at step t corresponds to the entire membrane potential response of length 500 (i.e., 500 time points) associated with a stimulus. The output of the Encoder is a latent space vector of length 500 encoding membrane potential responses information. The latent space vector is then concatenated with a 1D vector storing current profile information and fed to both Generator and Discriminator.

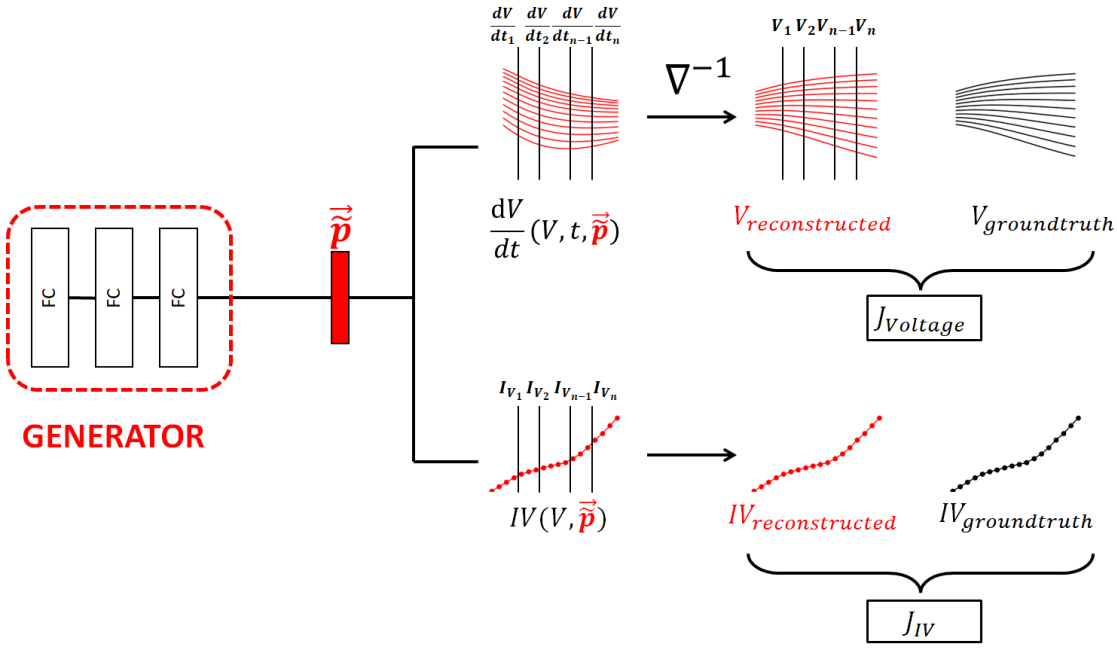

Figure 6: **Description of membrane potential responses and current reconstruction losses for the Generator**. Generated parameter vector $\vec{\tilde{p}}$ is used to evaluate membrane potential responses derivatives dV/dt at n time points sampled with fixed interval given the ground truth $\vec{V}$ at those time points. The evaluated membrane potential responses derivatives are then used to reconstruct $\vec{V}$ using the inverse gradient operation ∇^{-1}. The reconstructed $\vec{V}$ is then compared with ground truth $\vec{V}$ to evaluate the membrane potential responses reconstruction loss $J_{\vec{V}}$. Current reconstruction is computed in a similar way via evaluating the currents at defined membrane potential responses steps V given generated parameters $\vec{\tilde{p}}$ as inputs.

Discriminator Module. The goal of the Discriminator is given the input membrane potential responses and current profiles, to distinguish generated parameters from real ground truth parameters. The Discriminator receives as input the latent space vector from Encoder concatenated with generated or ground truth parameter vector and outputs a scalar representing the relative distance between two parameter sets (Figure 5, Eqn 3.1). Such a quantity is called Wasserstein distance or Wasserstein loss and differs from vanilla GAN Discriminator which only outputs 0 or 1. Wasserstein loss is known to remedy several common issues that arise from vanilla GAN such as vanishing gradient and mode collapse, leading to more stable training [47]. To further improve the training of WGAN architecture we supplement Wasserstein loss with a gradient penalty term, which ensures that the gradients of the Discriminator's output with respect to the input has unit norms [49]. This condition is called Lipschitz continuity and prevents Discriminator outputs from having large variations when there is only small variations in the inputs [49, 50]. Combined together, the Discriminator is trained with the following loss:

$$J_D = \mathbb{E}[D(\vec{\tilde{p}})] - \mathbb{E}[D(\vec{p})] + \lambda \mathbb{E}[(||\nabla_{\hat{p}} D(\hat{p})||_2 - 1)^2] \tag{1}$$

where $\mathbb{E}[D(\vec{\tilde{p}})]$ and $\mathbb{E}[D(\vec{p})]$ are the mean values of Discriminator outputs with respect to generated samples $\vec{\tilde{p}}$ and real samples $\vec{p}$ respectively and $\lambda \mathbb{E}[(||\nabla_{\hat{p}} D(\hat{p})||_2 - 1)^2]$ is the gradient penalty term modulated by λ where $\hat{p} = t\vec{\tilde{p}} + (1-t)\vec{p}$ is the interpolation between generated and real samples with $0 \leq t \leq 1$.

Generator Module. Being an adversary network of Discriminator, the goal of Generator is to fool the Discriminator by generating parameters that are indistinguishable from the real parameters. The Generator receives as input the concatenated vector from the Encoder and outputs a parameter vector (Figure 5). The module consists of 4 fully connected layers. Each parameter in the

output vector is scaled between -1 and 1 which is then scaled back to the parameters' original scales. The module is trained using 3 loss terms: i) Generator adversarial loss, ii) Membrane potential responses reconstruction loss and iii) Current reconstruction loss as follows:

$$J_G = -\mathbb{E}[D(\vec{\tilde{\boldsymbol{p}}})] + J_{\vec{\boldsymbol{V}}} + J_{\boldsymbol{I}\vec{\boldsymbol{V}}} \tag{2}$$

$$J_{\vec{\boldsymbol{V}}} = \sum_{i=1}^{n} |\vec{\boldsymbol{V}}_{groundtruth} - \vec{\boldsymbol{V}}_{reconstructed}| \tag{3}$$

$$J_{\boldsymbol{I}\vec{\boldsymbol{V}}} = \sum_{i=1}^{n} |\boldsymbol{I}\vec{\boldsymbol{V}}_{groundtruth} - \boldsymbol{I}\vec{\boldsymbol{V}}_{reconstructed}| \tag{4}$$

where $-\mathbb{E}[D(\vec{\tilde{\boldsymbol{p}}})]$ is Generator adversarial loss which is reciprocal of the mean Discriminator outputs with respect to generated samples and $J_{\vec{\boldsymbol{V}}}$ and $J_{\boldsymbol{I}\vec{\boldsymbol{V}}}$ are L_1 regression loss for reconstructed membrane potential responses and current profiles respectively. It's important to note that $J_{\vec{\boldsymbol{V}}}$ and $J_{\boldsymbol{I}\vec{\boldsymbol{V}}}$ are part of Generator's computation graph and thus force Generator to optimize them on top of adversarial loss (Figure 6). The composite loss function of Generator makes EP-GAN a "physics-informed" neural network as HH-model itself becomes part of the training process. Such networks have shown to be more data efficient during training as they don't rely solely on training data to learn effective strategy [51, 52]. The mathematical description of membrane potential responses and current reconstruction from generated parameter set $\vec{\tilde{\boldsymbol{p}}}$ is as follows:

$$\vec{\boldsymbol{V}}_{reconstructed}(t) = \nabla^{-1}\left(\frac{dV}{dt}(V, t, \vec{\tilde{\boldsymbol{p}}})\right),\ \boldsymbol{I}\vec{\boldsymbol{V}}_{reconstructed}(V) = IV(V, \vec{\tilde{\boldsymbol{p}}}) \tag{5}$$

Here $\frac{dV}{dt}(V, t, \vec{\tilde{\boldsymbol{p}}})$ is the right-hand-side function of HH-model which computes the membrane potential responses derivative at time t given the membrane potential responses V and parameter set $\vec{\tilde{\boldsymbol{p}}}$ and $IV(V, \vec{\tilde{\boldsymbol{p}}})$ is the function that evaluates neuron's current I given the membrane potential responses V and parameter set $\vec{\tilde{\boldsymbol{p}}}$. Membrane potential responses are reconstructed by first evaluating their derivatives at regularly sampled time points which are followed by inverse derivative operation ∇^{-1} which uses cumulative summation to approximate $\vec{\boldsymbol{V}}$ at sampled time points [53]. The total number of time points to be sampled can be adjusted to increase the accuracy of the membrane potential responses reconstruction in exchange for the increased computational cost. The current profile is reconstructed by directly evaluating $IV(V, \vec{\boldsymbol{p}})$ which uses generated parameter set $\vec{\tilde{\boldsymbol{p}}}$ over the range of membrane potential responses values. We show in Table 2 that reconstruction losses are essential for the accuracy of predicted parameters.

Generating Training Data. For a successful training of a neural network model, the training data must be of sufficient number of samples, denoised, and diverse. To ensure these conditions are met with a simulated dataset, we employ a two-step process for generating training data (Figure 7). In the first step, each parameter set is randomly generated by uniformly sampling of each parameter within a given range. The range is determined according to the biologically feasible ranges reported by the literature [8, 22, 41]. In the second step, membrane potential responses and current traces are simulated for each sampled parameter set followed by imposing constraints on each of them (See supplementary for numerical simulation protocol). For the membrane potential response, the constraint is such that the variances between membrane potential responses trajectories are above a certain value. For the IV profile, an upper bound and lower bound on the y-axis (current axis) are set such that a certain proportion of data points must fall within two boundaries. Parameter sets that do not satisfy these constraints are then removed from the training set. The constraints serve two purposes: i) remove parameter sets that result in non-realistic membrane potential responses/current profiles from the training set and ii) serve as an initial data augmentation process for the model. The

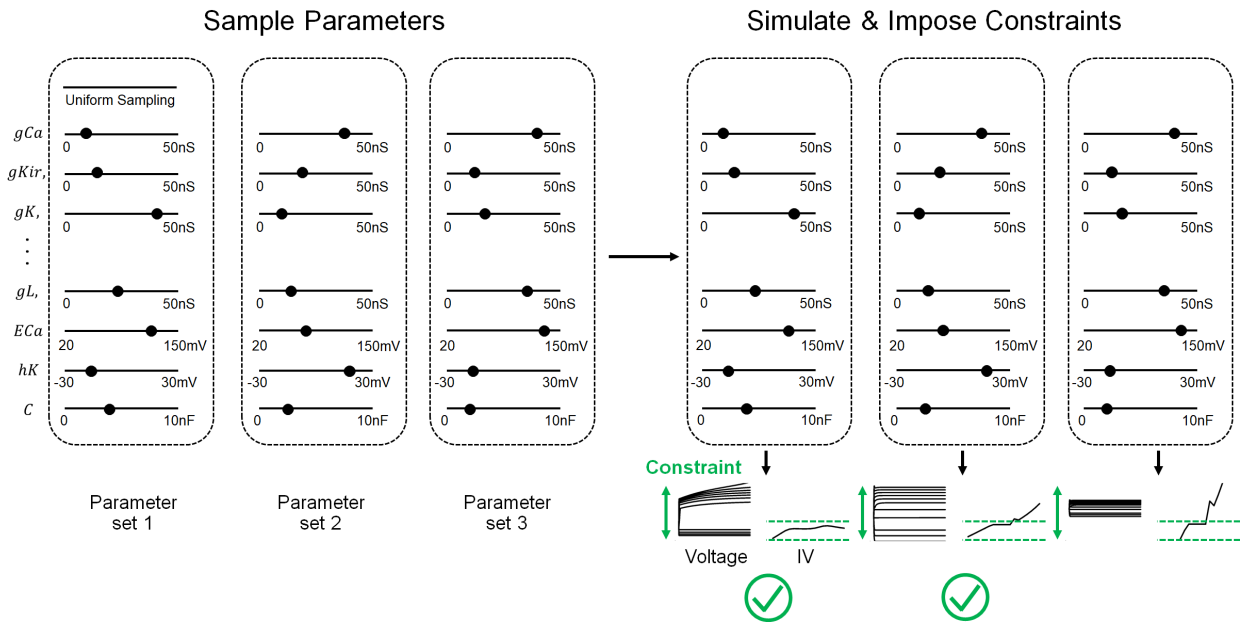

Figure 7: **Generating training data**. Each parameter is initially sampled from biologically plausible ranges using uniform sampling. A parameter set consists of 176 parameters spanning 15 known ion channels in *C. elegans*. Once parameter sets are generated, membrane potential responses and current profiles are evaluated for each set to ensure they satisfy the predefined constraints such as minimum offset between membrane potential responses traces and minimum and maximum bounds for steady-state current traces. Only parameter sets that meet the constraints are included in the training set.

constraints can also be adjusted if deemed necessary for the improvement of the training process. Once constraints are applied, Gaussian noise is added to the membrane potential responses training data to mimic the measurement noises in experimental membrane potential responses recording data.

3.2 Experimental Protocol

***C. elegans* Culture and Strains.** All animals used in this study were maintained at room temperature (22-23°C) on nematode growth medium (NGM) plates seeded with E. coli OP50 bacteria as a food source [54]. Strains used in this study were: CX7893 kyIs405 (AWB), CX3695 kyIs140 (AWC), ZG611 iaIs19 (URX), EG1285 lin-15B(n765);oxIs12 (RIS), UL2650 leEx2650 (DVC) and CX4857 kyIs179 (HSN).

Electrophysiology. Electrophysiological recordings were performed on young adult hermaphrodites (∼3-days old) at room temperature as previously described [8]. The gluing and dissection were performed under an Olympus SZX16 stereomicroscope equipped with a 1X Plan Apochromat objective and widefield 10X eyepieces. Briefly, an adult animal was immobilized on a Sylgard-coated (Sylgard 184, Dow Corning) glass coverslip in a small drop of DPBS (D8537; Sigma) by applying a cyanoacrylate adhesive (Vetbond tissue adhesive; 3M) along one side of the body. A puncture in the cuticle away from the incision site was made to relieve hydrostatic pressure. A small longitudinal incision was then made using a diamond dissecting blade (Type M-DL 72029 L; EMS) along the glue line adjacent to the neuron of interest. The cuticle flap was folded back and glued to the coverslip with GLUture Topical Adhesive (Abbott Laboratories), exposing the neuron to be recorded. The coverslip with the dissected preparation was then placed into a custom-made open recording chamber (∼1.5 ml volume) and treated with 1 mg/ml collagenase (type IV; Sigma) for ∼10 s by hand pipetting. The recording chamber was subsequently perfused with the standard extracellular solution using a custom-made gravity-feed perfusion system for ∼10 ml.

All electrophysiological recordings were performed with the bath at room temperature under an upright microscope (Axio Examiner; Carl Zeiss, Inc) equipped with a 40X water immersion lens and 16X eyepieces. Neurons of interest were identified by fluorescent markers and their anatomical positions. Preparations were then switched to the differential interference contrast (DIC) setting for patch-clamp. Electrodes with resistance (RE) of 15-25 MΩ were made from borosilicate glass pipettes (BF100-58-10; Sutter Instruments) using a laser pipette puller (P-2000; Sutter Instruments) and fire-polished with a microforge (MF-830; Narishige). We used a motorized micromanipulator (PatchStar Micromanipulator; Scientifica) to control the electrodes back filled with standard intracellular solution. The standard pipette solution was (all concentrations in mM): [K-gluconate 115; KCl 15; KOH 10; MgCl2 5; CaCl2 0.1; Na2ATP 5; NaGTP 0.5; Na-cGMP 0.5; cAMP 0.5; BAPTA 1; Hepes 10; Sucrose 50], with pH adjusted with KOH to 7.2, osmolarity 320–330 mOsm. The standard extracellular solution was: [NaCl 140; NaOH 5; KCL 5; CaCl2 2; MgCl2 5; Sucrose 15; Hepes 15; Dextrose 25], with pH adjusted with NaOH to 7.3, osmolarity 330–340 mOsm. Liquid junction potentials were calculated and corrected before recording. Whole-cell current clamp and voltage-clamp experiments were conducted on an EPC-10 amplifier (EPC-10 USB; Heka) using PatchMaster software (Heka). Two-component capacitive compensation was optimized at rest, and series resistance was compensated to 50%. Analog data were filtered at 2 kHz and digitized at 10 kHz. Current-injection and voltage-clamp steps were applied through the recording electrode.

4 Discussion

In this work, we introduce a novel deep generative method and system called ElectroPhysiomeGAN (EP-GAN), for estimating Hodgkin-Huxley model (HH-model) parameters given the recordings of neurons with graded potential (non-spiking). The proposed system encompasses RNN Encoder layer to process the neural recordings information such as membrane potential responses and steady-state current profiles and Generator layer to generate large number of HH-model parameters. The system can be trained entirely on simulation data informed by an arbitrary HH-model. When applied to neurons in *C. elegans*, EP-GAN generates parameters of HH-model which membrane potential responses is closer to test membrane potential responses than previous methods such as Differential Evolution and Genetic Algorithms. The advantage of EP-GAN is in both accuracy and inference speed achieved through less number of simulations than the previous methods and generic such that it doesn't depend on the number of neurons for which inference is to be performed [16]. In addition, the method also preserves performance when provided with input data with partial information such as missing membrane potential responses (up to 25%) or steady-state current traces (up to 75%).

While EP-GAN is a step forward toward ElectroPhysiome model of *C. elegans*, its inability to support neurons with spiking membrane potential responses remains a limitation. The reason stems from the fact that neurons with spiking membrane potential responses are rare during the generation of training data of HH 15 ionic channels model without Na channels, making their translation strategies to parameter space difficult to learn. A similar limitation is present with bi-stable membrane potential responses, e.g., AWB and AWC, although at lesser extent. While these profiles are not as rare as spiking membrane potential responses, their relative sparseness in the training data tends to cause lower quality of predicted parameters. Indeed, previous studies of *C. elegans* nervous system found that the majority of neurons exhibit graded membrane potential response instead of spiking [7, 21]. Furthermore, the limitation could also lie within the current architecture of EP-GAN as it currently processes data directly without a component that discerns and processes spiking membrane potential responses. Improving the sampling strategy for training data alongside enhancement of network architecture could address these limitations in the future.

As discussed in Methods section, it's worth noting that EP-GAN does not necessarily recover

the ground truth parameters that are associated with the input membrane potential responses and steady-state current profiles. This is mainly due to the fact that there may exist multiple parameter regimes for the HH-model which support the given inputs [5, 10, 11, 12, 13, 18, 44, 45]. The parameters generated by a single forward pass of EP-GAN (i.e., a single flow of information from the input to the output) could thus be interpreted as a one time sampling from such a regime and a small perturbation to inputs may result in a different set of parameters. Such sensitivity to perturbation could be adjusted by supplementing the training samples or inputs with additional recording data (e.g., multiple recording data per neuron).

As a neural network architecture, EP-GAN allows extensions and modifications to accommodate different configurations of the problem. For instance, update to HH-model would only require retraining of the network without changes to its architecture. Extending the inputs to include additional data, e.g., channel activation profiles, can also be done in a straightforward manner by concatenating them to the input vectors of the encoder network.

Despite its primary focus on *C. elegans* neurons, we believe EP-GAN and its future extension could be viable for modeling a variety of neurons in other organisms. Indeed, there are increasing advances in resolving connectomes of more complex organisms and techniques for large-scale neural activities [55, 56, 57, 58]. As neurons in these organisms can be described by a generic HH-model or similar differential equation model, EP-GAN is expected to be applicable and make contributions toward developing the biologically detailed nervous system models of neurons in these organisms.

Acknowledgements This work was supported in part by National Science Foundation grant CRCNS IIS-2113003 (JK,ES), Washington Research Fund (ES), CRCNS IIS-2113120 (QL), Kavli NSI Pilot Grant and Hong Kong Research Grants Council RGC/ECS grant (CityU 21103522) (QL), and Chan Zuckerberg Initiative (to Cori Bargmann). Authors also acknowledge the partial support by the Departments of Electrical Computer Engineering (JK,ES), Applied Mathematics (ES), the Center of Computational Neuroscience (ES), and the eScience Institute (ES,JK) at the University of Washington. In addition, we thank Cori Bargmann and Ian Hope for C. elegans strains. Some strains were provided by the CGC, which is funded by NIH Office of Research Infrastructure Programs (P40 OD010440).

References

[1] 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.

[2] Lav R Varshney, Beth L Chen, Eric Paniagua, David H Hall, and Dmitri B Chklovskii. Structural properties of the caenorhabditis elegans neuronal network. *PLoS computational biology*, 7(2):e1001066, 2011.

[3] 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.

[4] Alan L Hodgkin and Andrew F Huxley. A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of physiology*, 117(4):500, 1952.

[5] Allan R Willms. Neurofit: software for fitting hodgkin–huxley models to voltage-clamp data. *Journal of neuroscience methods*, 121(2):139–150, 2002.

[6] Allan R Willms, Deborah J Baro, Ronald M Harris-Warrick, and John Guckenheimer. An improved parameter estimation method for hodgkin-huxley models. *Journal of computational neuroscience*, 6:145–168, 1999.

[7] Martina Nicoletti, Alessandro Loppini, Letizia Chiodo, Viola Folli, Giancarlo Ruocco, and Simonetta Filippi. Biophysical modeling of c. elegans neurons: Single ion currents and whole-cell dynamics of awcon and rmd. *PloS one*, 14(7):e0218738, 2019.

[8] Qiang Liu, Philip B Kidd, May Dobosiewicz, and Cornelia I Bargmann. C. elegans awa olfactory neurons fire calcium-mediated all-or-none action potentials. *Cell*, 175(1):57–70, 2018.

[9] Jingyuan Jiang, Yifan Su, Ruilin Zhang, Haiwen Li, Louis Tao, and Qiang Liu. C. elegans enteric motor neurons fire synchronized action potentials underlying the defecation motor program. *Nature communications*, 13(1):2783, 2022.

[10] Eve Marder and Jean-Marc Goaillard. Variability, compensation and homeostasis in neuron and network function. *Nature Reviews Neuroscience*, 7(7):563–574, 2006.

[11] Eve Marder and Adam L Taylor. Multiple models to capture the variability in biological neurons and networks. *Nature neuroscience*, 14(2):133–138, 2011.

[12] Astrid A Prinz, Cyrus P Billimoria, and Eve Marder. Alternative to hand-tuning conductance-based models: construction and analysis of databases of model neurons. *Journal of neurophysiology*, 2003.

[13] Astrid A Prinz, Dirk Bucher, and Eve Marder. Similar network activity from disparate circuit parameters. *Nature neuroscience*, 7(12):1345–1352, 2004.

[14] Laure Buhry, Michele Pace, and Sylvain Saïghi. Global parameter estimation of an hodgkin–huxley formalism using membrane voltage recordings: Application to neuro-mimetic analog integrated circuits. *Neurocomputing*, 81:75–85, 2012.

[15] Juan Luis Jiménez Laredo, Loïs Naudin, Nathalie Corson, and Carlos M Fernandes. A methodology for determining ion channels from membrane potential neuronal recordings. In *Applications of Evolutionary Computation: 25th European Conference, EvoApplications 2022, Held as Part of EvoStar 2022, Madrid, Spain, April 20–22, 2022, Proceedings*, pages 15–29. Springer, 2022.

[16] Loïs Naudin, Juan Luis Jiménez Laredo, Qiang Liu, and Nathalie Corson. Systematic generation of biophysically detailed models with generalization capability for non-spiking neurons. *PloS one*, 17(5):e0268380, 2022.

[17] Y Curtis Wang, Johann Rudi, James Velasco, Nirvik Sinha, Gideon Idumah, Randall K Powers, Charles J Heckman, and Matthieu K Chardon. Multimodal parameter spaces of a complex multi-channel neuron model. 2022.

[18] Jemy A Mandujano Valle and Alexandre L Madureira. Parameter identification problem in the hodgkin-huxley model. *Neural Computation*, 34(4):939–970, 2022.

[19] Pedro J Gonçalves, Jan-Matthis Lueckmann, Michael Deistler, Marcel Nonnenmacher, Kaan Öcal, Giacomo Bassetto, Chaitanya Chintaluri, William F Podlaski, Sara A Haddad, Tim P Vogels, et al. Training deep neural density estimators to identify mechanistic models of neural dynamics. *Elife*, 9:e56261, 2020.

[20] Lautaro Estienne. Towards an hybrid hodgkin-huxley action potential generation model. In *2021 XIX Workshop on Information Processing and Control (RPIC)*, pages 1–6. IEEE, 2021.

[21] Miriam B Goodman, David H Hall, Leon Avery, and Shawn R Lockery. Active currents regulate sensitivity and dynamic range in c. elegans neurons. *Neuron*, 20(4):763–772, 1998.

[22] Loïs Naudin, Nathalie Corson, MA Aziz-Alaoui, Juan Luis Jimenez Laredo, and Thibaut Démare. On the modeling of the three types of non-spiking neurons of the caenorhabditis elegans. *International Journal of Neural Systems*, 31(02):2050063, 2021.

[23] Kristin Koch, Judith McLean, Ronen Segev, Michael A Freed, Michael J Berry, Vijay Balasubramanian, and Peter Sterling. How much the eye tells the brain. *Current biology*, 16(14):1428–1434, 2006.

[24] Alan Roberts and Brian MH Bush. *Neurones without impulses: their significance for vertebrate and invertebrate nervous systems*, volume 6. Cambridge University Press, 1981.

[25] RE Davis and AO Stretton. Passive membrane properties of motorneurons and their role in long-distance signaling in the nematode ascaris. *Journal of Neuroscience*, 9(2):403–414, 1989.

[26] Ralph E Davis and AO Stretton. Signaling properties of ascaris motorneurons: graded active responses, graded synaptic transmission, and tonic transmitter release. *Journal of Neuroscience*, 9(2):415–425, 1989.

[27] M Burrows, GJ Laurent, and LH Field. Proprioceptive inputs to nonspiking local interneurons contribute to local reflexes of a locust hindleg. *Journal of Neuroscience*, 8(8):3085–3093, 1988.

[28] Gilles Laurent and Malcolm Burrows. Distribution of intersegmental inputs to nonspiking local interneurons and motor neurons in the locust. *Journal of Neuroscience*, 9(9):3019–3029, 1989.

[29] Gilles Laurent and M Burrows. Intersegmental interneurons can control the gain of reflexes in adjacent segments of the locust by their action on nonspiking local interneurons. *Journal of Neuroscience*, 9(9):3030–3039, 1989.

[30] Loïs Naudin. Biological emergent properties in non-spiking neural networks. *AIMS Mathematics*, 7(10):19415–19439, 2022.

[31] Kalyanmoy Deb, Samir Agrawal, Amrit Pratap, and Tanaka Meyarivan. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: Nsga-ii. In *Parallel Problem Solving from Nature PPSN VI: 6th International Conference Paris, France, September 18–20, 2000 Proceedings 6*, pages 849–858. Springer, 2000.

[32] Etay Hay, Sean Hill, Felix Schürmann, Henry Markram, and Idan Segev. Models of neocortical layer 5b pyramidal cells capturing a wide range of dendritic and perisomatic active properties. *PLoS computational biology*, 7(7):e1002107, 2011.

[33] Werner Van Geit, Erik De Schutter, and Pablo Achard. Automated neuron model optimization techniques: a review. *Biological cybernetics*, 99:241–251, 2008.

[34] Tea Robič and Bogdan Filipič. Differential evolution for multiobjective optimization. In *Evolutionary Multi-Criterion Optimization: Third International Conference, EMO 2005, Guanajuato, Mexico, March 9-11, 2005. Proceedings 3*, pages 520–533. Springer, 2005.

[35] Saku Kukkonen and Jouni Lampinen. Gde3: The third evolution step of generalized differential evolution. In *2005 IEEE congress on evolutionary computation*, volume 1, pages 443–450. IEEE, 2005.

[36] Rakesh Angira and BV Babu. Non-dominated sorting differential evolution (nsde): An extension of differential evolution for multi-objective optimization. In *IICAI*, pages 1428–1443, 2005.

[37] Timothy H Rumbell, Danel Draguljić, Aniruddha Yadav, Patrick R Hof, Jennifer I Luebke, and Christina M Weaver. Automated evolutionary optimization of ion channel conductances and kinetics in models of young and aged rhesus monkey pyramidal neurons. *Journal of computational neuroscience*, 41:65–90, 2016.

[38] J Christopher Octeau, Mohitkumar R Gangwani, Sushmita L Allam, Duy Tran, Shuhan Huang, Tuan M Hoang-Trong, Peyman Golshani, Timothy H Rumbell, James R Kozloski, and Baljit S Khakh. Transient, consequential increases in extracellular potassium ions accompany channelrhodopsin2 excitation. *Cell reports*, 27(8):2249–2261, 2019.

[39] Laure Buhry, Audrey Giremus, Eric Grivel, Sylvain Saïghi, and Sylvie Renaud. New variants of the differential evolution algorithm: application for neuroscientists. In *2009 17th European Signal Processing Conference*, pages 2352–2356. IEEE, 2009.

[40] Julian Blank and Kalyanmoy Deb. Pymoo: Multi-objective optimization in python. *IEEE Access*, 8:89497–89509, 2020.

[41] Eugene M Izhikevich. *Dynamical systems in neuroscience*. MIT press, 2007.

[42] Loïs Naudin, Juan Luis Jiménez Laredo, and Nathalie Corson. A simple model of nonspiking neurons. *Neural Computation*, 34(10):2075–2101, 2022.

[43] Jinming Zou, Yi Han, and Sung-Sau So. Overview of artificial neural networks. *Artificial neural networks: methods and applications*, pages 14–22, 2009.

[44] Ashley E Raba, Jonathan M Cordeiro, Charles Antzelevitch, and Jacques Beaumont. Extending the conditions of application of an inversion of the hodgkin–huxley gating model. *Bulletin of mathematical biology*, 75:752–773, 2013.

[45] Loïs Naudin. Different parameter solutions of a conductance-based model that behave identically are not necessarily degenerate. *Journal of Computational Neuroscience*, pages 1–6, 2023.

[46] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

[47] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223. PMLR, 2017.

[48] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[49] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. *Advances in neural information processing systems*, 30, 2017.

[50] Aladin Virmaux and Kevin Scaman. Lipschitz regularity of deep neural networks: analysis and efficient estimation. *Advances in Neural Information Processing Systems*, 31, 2018.

[51] George Em Karniadakis, Ioannis G Kevrekidis, Lu Lu, Paris Perdikaris, Sifan Wang, and Liu Yang. Physics-informed machine learning. *Nature Reviews Physics*, 3(6):422–440, 2021.

[52] Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics*, 378:686–707, 2019.

[53] James Stewart, Daniel K Clegg, and Saleem Watson. *Calculus: early transcendentals*. Cengage Learning, 2020.

[54] Sydney Brenner. The genetics of caenorhabditis elegans. *Genetics*, 77(1):71–94, 1974.

[55] Paul Brooks, Andrew Champion, and Marta Costa. Mapping of the zebrafish brain takes shape. *Nature Methods*, pages 1–2, 2022.

[56] Michael Winding, Benjamin D Pedigo, Christopher L Barnes, Heather G Patsolic, Youngser Park, Tom Kazimiers, Akira Fushiki, Ingrid V Andrade, Avinash Khandelwal, Javier Valdes-Aleman, et al. The connectome of an insect brain. *Science*, 379(6636):eadd9330, 2023.

[57] Seung Wook Oh, Julie A Harris, Lydia Ng, Brent Winslow, Nicholas Cain, Stefan Mihalas, Quanxin Wang, Chris Lau, Leonard Kuan, Alex M Henry, et al. A mesoscale connectome of the mouse brain. *Nature*, 508(7495):207–214, 2014.

[58] Nicholas James Sofroniew, Daniel Flickinger, Jonathan King, and Karel Svoboda. A large field of view two-photon mesoscope with subcellular resolution for in vivo imaging. *elife*, 5:e14472, 2016.