

---

# NOBLE – Neural Operator with Biologically-informed Latent Embeddings to Capture Experimental Variability in Biological Neuron Models

---

Luca Ghafourpour<sup>1,2</sup> Valentin Duruisseaux<sup>2\*</sup> Bahareh Tolooshams<sup>3,4\*</sup> Philip H. Wong<sup>5</sup>  
Costas A. Anastassiou<sup>5,6</sup> Anima Anandkumar<sup>2</sup>

<sup>1</sup>ETH Zürich <sup>2</sup>California Institute of Technology <sup>3</sup>University of Alberta  
<sup>4</sup>Alberta Machine Intelligence Institute (Amii) <sup>5</sup>Cedars-Sinai Medical Center  
<sup>6</sup>Archimedes AI, Athena Research Center

## Abstract

Characterizing the cellular properties of neurons is fundamental to understanding their function in the brain. In this quest, the generation of bio-realistic models is central towards integrating multimodal cellular data sets and establishing causal relationships. However, current modeling approaches remain constrained by the limited availability and intrinsic variability of experimental neuronal data. The deterministic formalism of bio-realistic models currently precludes accounting for the natural variability observed experimentally. While deep learning is becoming increasingly relevant in this space, it fails to capture the full biophysical complexity of neurons, their nonlinear voltage dynamics, and variability. To address these shortcomings, we introduce NOBLE, a neural operator framework that learns a mapping from a continuous frequency-modulated embedding of interpretable neuron features to the somatic voltage response induced by current injection. Trained on synthetic data generated from bio-realistic neuron models, NOBLE predicts distributions of neural dynamics accounting for the intrinsic experimental variability. Unlike conventional bio-realistic neuron models, interpolating within the embedding space offers models whose dynamics are consistent with experimentally observed responses. NOBLE enables the efficient generation of synthetic neurons that closely resemble experimental data and exhibit trial-to-trial variability, offering a 4200× speedup over the numerical solver. NOBLE is the first scaled-up deep learning framework that validates its generalization with real experimental data. To this end, NOBLE captures fundamental neural properties in a unique and emergent manner that opens the door to a better understanding of cellular composition and computations, neuromorphic architectures, large-scale brain circuits, and general neuroAI applications.

## 1 Introduction

Hundreds of distinct neuronal cell types co-exist and compute within neural circuits, yet how they shape cognitive functions remains essentially unanswered [1–5]. This is particularly true in the human brain, where access and monitoring capabilities are severely limited compared to animal models. Over the past decade, multimodal cellular datasets that integrate electrophysiology, morphology, and transcriptomics have emerged for human cell types [6–10].

---

\*These authors contributed equally to this work.

While integrating across the different modalities remains a challenge, clear differences in gene expression, morphology, and electrophysiology are evident across cell types. However, understanding how these differences impact brain processing is crucial, e.g. to uncover how expression of specific genes relates to neurological diseases.

Cellular models representing multiple data modalities are invaluable as they offer a degree of control and perturbations that are experimentally impossible (e.g., [11–13]). Recently, evolutionary multi-objective optimization algorithms [14] have been used to generate and validate bio-realistic models of neurons in the form of 3D multi-compartment partial differential equation (PDE) models that mirror both their shape and ion channel expression, shaping their electrical properties [12, 13, 15, 16]. Yet, such models are deterministic and fail to capture the intrinsic variability observed experimentally, where identical input to the same neuron often results in different electrophysiological responses. One approach is to generate families of models, referred to as "hall-of-fame" (HoF) models [12, 13, 17] to represent a single cell. While each HoF model is distinct and reproduces the electrophysiological features of parts of an experiment, the ensemble of deterministic HoF models is used as a collective representation that captures both the main features as well as their variability in an experiment [12, 13]. Typically, neurons exhibit highly nonlinear behavior, necessitating equally complex models rendering the optimization computationally demanding (i.e. requiring about 600k CPU core hours per single-neuron model [12, 13]). Yet, even tiny perturbations of the model parameters lead to large deviations from experimental data [18]. Other approaches have explored capturing variability through introducing stochasticity in neuron models [19–21]. However, the synthetic injection of white noise is non-mechanistic and introduces perturbations that can lead to unrealistic predictions [22–24]. In summary, capturing the nature and variability of neurons is a challenge with existing computational techniques.

The challenges of scalability and the computational cost associated with traditional numerical modeling approaches, such as numerical integrators and evolutionary optimization algorithms, have led the scientific community to explore the use of machine learning to accelerate simulations by learning underlying relationships between variables directly from experimental and synthetic data. While neural networks have been used successfully for many applications, they learn mappings between finite-dimensional vectors, which can limit their ability to model physical phenomena that are better described using functions in infinite-dimensional spaces and functional relationships between them [25]. As a result, neural networks can overfit to the training discretization and suffer from limited out-of-distribution capabilities. Neural operators [25–27] are a principled way to generalize neural networks to learn operators mapping functions to functions, with a universal operator approximation property [28]. A variety of neural operators have been proposed, such as the Fourier Neural Operator (FNO) [29, 30].

Machine learning approaches have been applied to model single-cell electrophysiology, primarily for point-neuron systems such as FitzHugh–Nagumo [31, 32] and Hodgkin–Huxley [33]. Conventional neural networks [34–36] and physics-informed neural networks [37–42] successfully reproduced their dynamics but remain highly specific to deterministic formulations and require retraining for each new stimulus. More recently, neural operators demonstrated strong potential for learning the governing dynamics of Hodgkin–Huxley systems [43], though the study was limited to simplified data, without capturing biological variability. Related works like NeuPRINT [44] captured biological neuronal variability, but on slower in vivo 2-photon calcium imaging data and models population-level fluorescence dynamics, rather than fast intracellular voltage dynamics of individual neurons.

We build on these advances and address current limitations to enable deeper insights into brain function and neuroAI.

**Contributions.** We introduce NOBLE (Neural Operator with Biologically-informed Latent Embeddings), a neural operator framework for learning the nonlinear somatic dynamics across a population of HoF models for a single neuron (Figure 1). NOBLE is the first scaled-up deep learning framework whose performance is validated with experimental human cortex data. Rather than training a separate independent surrogate for each HoF model, NOBLE learns a single neural operator that maps from a continuous latent space of user-defined, interpretable neuron characteristics to an ensemble of somatic voltage responses induced by current injection. This latent space is constructed using an embedding strategy informed by the specified characteristics of the neuron models.



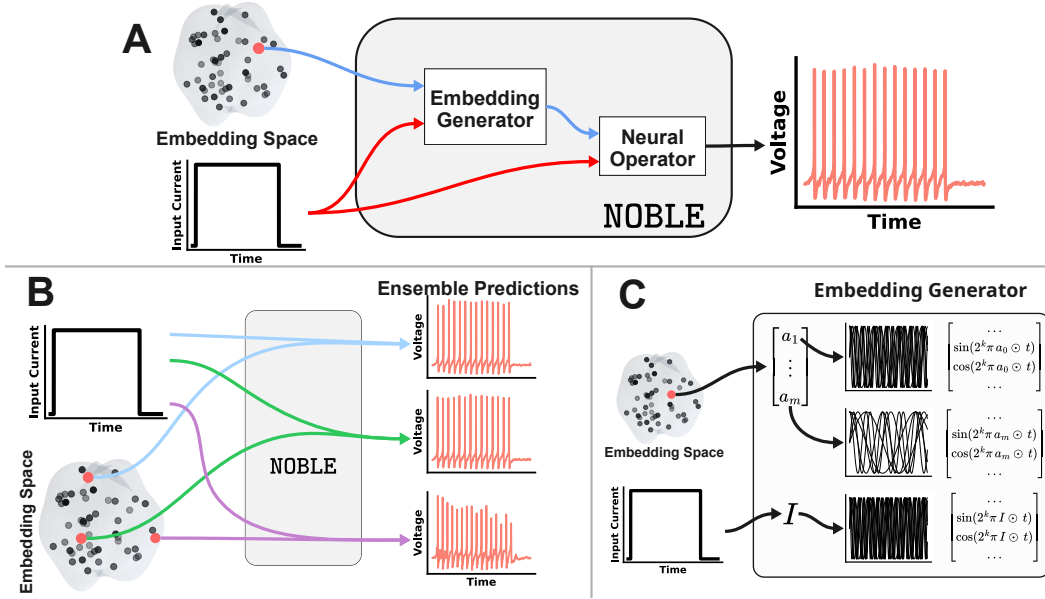


Figure 1: The Neural Operator with Biologically-informed Latent Embeddings (NOBLE) framework. **A)** In NOBLE, a current injection and neuron model features are first encoded using the proposed embedding strategy, before passing through a neural operator to produce a prediction for the somatic voltage response. **B)** NOBLE can be queried in parallel with different model latent representations to produce ensemble predictions. **C)** The proposed embedding in NOBLE encodes specified neuron features and the input current as a stack of trigonometric time-series, as described in Section 3.4.

As an example application, we train and evaluate NOBLE on a parvalbumin-positive (PVALB) neuron dataset generated using 50 HoF PVALB models (Figure 2). We show that a single NOBLE model accurately captures both subthreshold and spiking dynamics across all 50 HoF models (in-distribution) as well as 10 unseen HoF models (out-of-distribution) while achieving a significant speedup of 4200x over the numerical solver used to generate the dataset (Figure 3). In addition, the NOBLE predictions across 16 electrophysiological features of interest (including spike count, amplitude, and width) remain within the variability observed in experimental data (Figure 5B). Additional ablation studies confirm that biologically informed embeddings are critical for capturing both firing and non-firing dynamics, and demonstrate that we can enhance performance on targeted features without compromising overall dynamics by introducing a feature-specific fine-tuning approach.

We further show that NOBLE can successfully generate novel bio-realistic neuron models by sampling and interpolating within the latent space of models. The dynamics of novel neuron models generated by NOBLE align both with previously unseen HoF PVALB models and experimentally observed somatic responses. In contrast, direct interpolation between the parameters of bio-realistic PDE-based neuron models fails due to the sensitivity and nonlinearity of the underlying PDEs [45, 46] (Figure 4). We also successfully instantiate and train an additional NOBLE on a vasoactive intestinal peptide (VIP) interneuron to demonstrate the generalizability of our embedding framework. Owing to NOBLE’s ability to generate novel bio-realistic neuron models, ensemble predictions are no longer constrained to the original 50 HoF models used for training. We demonstrate that NOBLE can produce somatic voltage responses for an arbitrary number of biologically plausible neurons by predicting responses to input stimuli across a larger set of models (Figure 5). The results showcase NOBLE’s ability to accurately capture a broad range of neuron dynamics while enabling dense interpolation across model space. By unlocking the efficient, unlimited generation of diverse yet realistic neurons from a continuous embedding, NOBLE offers a scalable alternative to computationally intensive, scale-limited evolutionary approaches, laying the foundation for brain-scale neural circuit modeling.

Finally, the biologically-informed latent representation of the neuron models together with the capability of NOBLE to generate arbitrarily many new bio-realistic neuron models also offers further insight into the behavior of neural dynamics. We can use NOBLE to obtain the somatic responses to current injections on a fine grid in the model latent representation space, and consequently construct heat maps and surface plots to better understand how neuron features used for the model latent representation affect any electrophysiological feature of interest.

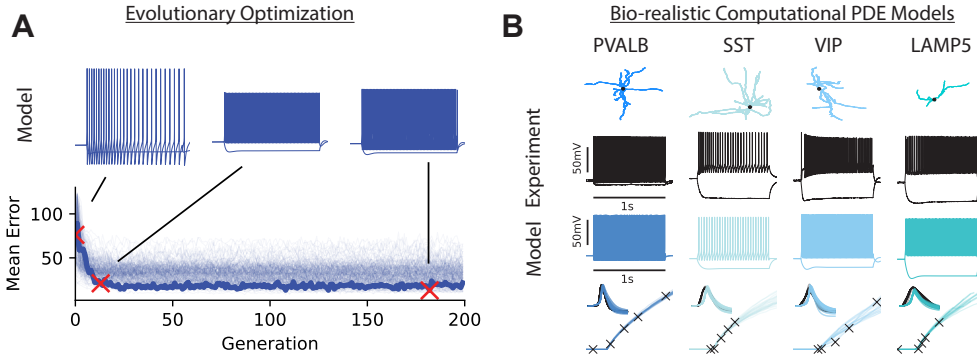


Figure 2: Creation of Bio-realistic PDE-based Neuron Models. **A)** Evolutionary optimization process for a neuron of interest, with voltage responses sampled at different generations (top) and the error history with other experimental neurons overlaid in the background (bottom). **B)** Sample HoF models of various inhibitory cell-types, showing morphology (top), experimental voltage traces (2nd row), simulated voltage traces (3rd row), and spike waveform and frequency-current curves (bottom).

## 2 Background on Bio-Realistic Neuron Modeling

We create bio-realistic PDE-based neuron models (based on the cable equation [47]) using actual reconstructions of neuron morphologies from human cortical data [6, 8, 9] (Figure 2). We instantiate these models using a framework [45] built on the NEURON simulation environment [46], which uses a spatial discretization to simulate the models as a system of coupled ordinary differential equations. We place ion channels in an “all-active” configuration [12, 13], where active ion channels are distributed along both somatic and dendritic compartments along the neuron morphology. For each experimental neuron, models are generated using a multi-objective evolutionary optimization framework [13] to find ion conductance parameters replicating a standard set of electrophysiological features from patch clamp recordings (Figure 2A). We adopt a two-stage optimization strategy, first fitting passive subthreshold responses, followed by capturing the active dynamics above the spiking threshold and the full frequency-current curve of each neuron. After 250 generations of evolutionary optimization, the models that best minimize the mean z-score error between simulated and actual experimental electrophysiological features are selected as HoF models (Figure 2B). More details about the electrophysiological features of interest are provided in Appendix B.

To illustrate the proposed approach, we consider a randomly selected PVALB human cortical neuron, for which we created 60 HoF models. PVALB neurons are fast-spiking inhibitory interneurons regulating high-frequency gamma oscillations (30-80Hz) [48] and their dysfunction has been associated with cognitive impairments such as schizophrenia and Alzheimer’s disease [49, 50]. We also consider the class VIP of inhibitory interneurons, known for its disinhibitory role in cortical circuits [51–53].

## 3 Method

### 3.1 Subsampling

NOBLE utilizes the notable property of neural operators of training on low-resolution data while reserving the capability to generate dynamics at higher resolution. In this regard, we subsample the reference HoF simulations in time. To avoid discarding high-resolution information necessary for capturing neuron features of interest, we analyze how these features are affected by different subsampling factors and strategies, in particular via the discrepancy between HoF simulations and experimental data. We consider (i) low-pass filtering followed by decimation in time, (ii) low-pass filtering followed by truncation in the frequency domain, (iii) truncation in the frequency domain, and (iv) decimation in time without filtering. Across neuron features, we observe no consistent differences in performance between these strategies and thus opt for low-pass filtering followed by decimation in time. Our analysis (see Appendix C.1) reveals that 3× subsampling preserves the fidelity of extracted features, within the bounds of the intrinsic discrepancy between HoF simulations and experimental data, and without inducing notable aliasing. For the HoF simulations, we consider time series of 515ms with a timestep of 0.02ms. For such signals, the subsampling reduces the sequence length from 25,750 to 8,583, substantially decreasing the computational load without compromising biological realism.

### 3.2 Current Amplitude Sampling

We consider square DC step current inputs, which are widely used in electrophysiological experiments and common for characterizing neuron behavior. We sample the current amplitudes from a skew-normal distribution whose support matches the experimentally validated range of the HoF models,  $I \in [-0.11, 0.28]\text{nA}$ . To effectively capture the highly nonlinear dynamics around the spiking threshold (0 to  $0.05\text{nA}$ ) where neural responses transition abruptly from being non-spiking to spiking, the mode of our sampling distribution is strategically located within this peri-threshold window. To address the greater learning challenge posed by the high-frequency components of depolarizing, spiking responses (characterized by features such as spike width, latency to first spike, and spike count), we deliberately use a heavier positive tail in our sampling distribution. This ensures the model encounters numerous examples of spike onset and complex spiking patterns during training while still covering the full input range. The distribution of square-pulse amplitudes is shown in Figure 8.

### 3.3 Neural Operators for Neuron Dynamics Simulation

We choose to use neural operators as they offer clear advantages for modeling complex dynamics (Appendix C.3). Among neural operators, the Fourier Neural Operator (FNO) [29, 30] is very efficient as it leverages fast Fourier transforms on equidistant grids, which aligns naturally with our setting where both experimental recordings and PDE simulations are sampled at constant timesteps. FNOs provide a principled and efficient framework for modeling neuronal dynamics by learning mappings from input currents to voltage responses across a broad family of neuron models and current injections. Unlike conventional neural networks that operate on vector inputs and outputs of fixed sizes, the FNO learns operators, that is, mappings between functions. By operating in the frequency domain, the FNO efficiently captures global, nonlinear, and high-frequency components of voltage responses. These properties allow the model to generalize across different temporal resolutions, input currents, and neuron types, enabling the accurate simulation of unseen configurations without retraining.

### 3.4 Embedding Strategy for Neuron-Model Variability

NOBLE learns a single neural operator that maps from a continuous latent space of user-defined, interpretable neuron characteristics to an ensemble of somatic voltage responses induced by current injection. The frequency-current (F-I) curve is a useful electrophysiological descriptor that summarizes cellular excitability by relating injected current amplitude to the neuron’s firing rate [54]. Differences between HoF parameterizations manifest as shifts in key features of this curve: the threshold current  $I_{thr}$  (the minimum amplitude that elicits spiking) and the local slope  $s_{thr}$  at  $I_{thr}$  (the rate of increase in the firing rate upon spiking). Figure 3A displays examples of F-I curves for different HoF PVALB models, illustrating how variability in  $I_{thr}$  and  $s_{thr}$  can represent the trial-to-trial intrinsic variability observed when a single neuron is repeatedly recorded under the same current injection.

Using this observation, we propose representing a given neuron model by its threshold current  $I_{thr}$  and local slope  $s_{thr}$ , that is, using  $(I_{thr}, s_{thr})$ . We propose to use this representation as part of a NeRF-style (Neural Radiance Field) embedding [55], where input features are encoded using sine and cosine functions. More precisely, a feature  $p$  is encoded as a stack of trigonometric time-series

$$\gamma(p, t) = [\sin(2^0 \pi p \odot t), \cos(2^0 \pi p \odot t), \dots, \sin(2^{K-1} \pi p \odot t), \cos(2^{K-1} \pi p \odot t)], \quad (1)$$

for some integer  $K > 0$ , where the frequencies are modulated by the feature  $p$ . Here  $t$  denotes the discretized time coordinates and  $\odot$  indicates element-wise multiplication with appropriate broadcasting.

The use of sine and cosine functions for encoding features is particularly synergistic with FNOs, which operate in the frequency domain to learn mappings between functions. FNOs leverage the Fourier transform to represent and manipulate data as sums of sine and cosine functions, effectively learning complex patterns by capturing interactions among frequency components. NeRF-style encodings lead to a representation of the input features that aligns naturally with the spectral approach of FNOs, enhancing their ability to learn high-frequency dynamics. In this context, the sinusoidal embeddings can be thought of as a form of spectral lifting, translating low-dimensional inputs into a richer representation in the frequency domain that FNOs can more efficiently process.

We encode separately the model features  $I_{thr}$  and  $s_{thr}$ , and the amplitude of the current injection, and stack the resulting embeddings as input channels. To compress the large range of  $I_{thr}$  and  $s_{thr}$  values into a manageable scale for embedding, we normalize them to  $[0.5, 3.5]^2$ , supporting more distinct feature space representations of HoF models. Figure 10 displays the latent representations in normalized  $(I_{thr}, s_{thr})$ -space of the 60 HoF PVALB models used in our numerical experiments.

### 3.5 NOBLE: Neural Operator with Biologically-informed Latent Embeddings

We introduce the Neural Operator with Biologically-informed Latent Embeddings (NOBLE), for modeling neuronal voltage dynamics in response to current injections. NOBLE offers a scalable alternative to computationally intensive numerical solvers for biophysically detailed, PDE-based neuron models. It learns a direct mapping from input currents and a continuous, interpretable latent space of neuron features, to the resulting voltage traces (Figure 1A). A key feature of NOBLE is its use of biologically-informed embeddings, which enables interpretability and generalization across biological neuron models. At its core, NOBLE is based on a neural operator, whose discretization invariance allows NOBLE to learn on low-resolution data and infer somatic voltage dynamics at higher resolutions. By combining a neural operator with the proposed continuous interpretable embedding, NOBLE learns a continuous operator over the space of bio-realistic neuron models.

The proposed NOBLE framework offers key advantages that set it apart from existing approaches:

- NOBLE provides a unified framework that learns ensemble dynamics directly, enabling it to generate diverse, biophysically plausible membrane potentials for the same input. Conditioned on a particular electrophysiological feature, it produces one realization of the intrinsic variability observed in biological neurons. This stands in contrast to previous deep learning approaches, which are inherently deterministic and produce a single trace for each input, failing to capture the trial-to-trial variability observed experimentally. To account for different neuronal behaviors, such models must be retrained for each variation, resulting in inefficiency and fragmentation.
- Through the latent embedding space of electrophysiological features, NOBLE can interpolate between known HoF models to produce new, bio-realistic neuronal responses. This capability is significant because HoF models are restricted to the finite set discovered by evolutionary optimization, and direct interpolation between their parameters does not result in realistic traces. As shown in Figure 4, interpolation in NOBLE’s latent space consistently produces valid, bio-realistic responses, whereas interpolations between PDE parameters do not.
- NOBLE can rapidly generate arbitrarily many distinct neuron models by sampling points within this continuous latent embedding space and producing the corresponding dynamics. This enables a single model to capture both spiking and subthreshold behaviors beyond the finite set of HoF models, providing an effectively infinite ensemble of bio-realistic responses that remain consistent with the variability observed in biological neurons. This is distinct from previous deep-learning methods, which were limited to predicting either spiking or subthreshold regimes in isolation.
- The bio-informed latent space of NOBLE, combined with its ability to generate unlimited realistic neuron models, enables fine-grained exploration of neural dynamics. By sampling models across this space, NOBLE can produce somatic responses for different latent features, and reveal how they influence electrophysiological behavior via visualizations (e.g. heat maps and surface plots).

## 4 Results

### 4.1 Experimental Setup

For evaluating NOBLE, we focus on the PVALB neuron example introduced in Section 2, and further assess the framework’s generality using a VIP neuron. In the PVALB setting, NOBLE receives as input the applied current injection  $I$ , together with stacked embeddings of  $I$  (with  $K = 9$  different frequencies) and of the normalized model features  $(I_{thr}, s_{thr})$  associated to a neuron model  $\text{HoF}_\ell$  (with  $K = 1$  frequency). NOBLE then outputs a corresponding somatic voltage response. The current injections are square DC steps with an activation duration of 400ms, consistent across all stimuli used for training and testing. We have access to 60 HoF models, where 50 are used during training,  $\{\text{HoF}^{train}\}$ , and the remaining 10 HoF models  $\{\text{HoF}^{test}\}$  are used for testing. Figure 10 displays the latent representations in normalized  $(I_{thr}, s_{thr})$ -space of these HoF models. For more details on the data generation, see Appendix D.1.

We use a 1D FNO (implemented as in the NeuralOperator library [56]) with 12 layers, each with 24 hidden channels and 256 Fourier modes. The resulting NOBLE with 1.8M trainable parameters is trained in PyTorch for 300 epochs using the Adam optimizer with learning rate 0.004, and the ReduceLROnPlateau scheduler with factor 0.4 and patience 4. The training minimizes the relative L4 error, while the performance metrics are reported using the relative L2 error for interpretability.

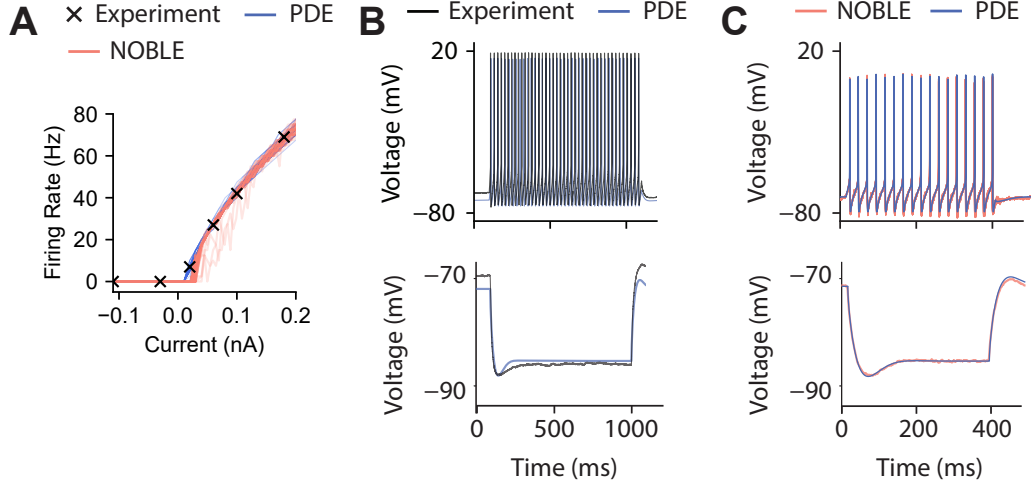


Figure 3: **A)** F-I curves from experimental recordings, PDE simulations, and NOBLE predictions on  $\{\text{HoF}^{\text{train}}\}$ . For one HoF model, **B)** compares experimental voltage responses with PDE simulations at a current injection of  $0.1\text{nA}$  (top) and  $-0.11\text{nA}$  (bottom), and **C)** compares the corresponding PDE simulations with the NOBLE predictions for the same HoF model.

To better evaluate physiologically meaningful behavior, we also report errors on four key electrophysiological features: `spikecount`, `AP1_width`, `mean_AP_amplitude`, and `steady_state_voltage` (see Appendix B for definitions of the features). For benchmarking, we compare NOBLE predictions against numerical simulations obtained from HoF models and experimental data since the HoF models were optimized to produce the closest approximations to experimental recordings and capture the biological variability required for a meaningful benchmark. Further details on the evaluation and evaluation metrics are provided in Appendix D.2.

The PyTorch codes used for our implementation of NOBLE and the numerical experiments are based on the NeuralOperator library [56], and are made available at [github.com/neuraloperator/noble](https://github.com/neuraloperator/noble).

#### 4.2 Testing on HoF Models Included in the Training Set

We first validate that the trained NOBLE can accurately reproduce the somatic voltage responses of the training  $\{\text{HoF}^{\text{train}}\}$  models when tested on current injections not seen during training. Figure 3C shows that the voltage traces exhibit minimal differences, confirming that NOBLE generalizes well to unseen inputs. This is supported by a relative L2 test error of 2.18% with the  $\{\text{HoF}^{\text{train}}\}$  models. Figure 3B also shows that the numerical solver outputs align closely with experimental recordings, and together with Figure 3C, indicates that NOBLE inherits this agreement and captures physiologically meaningful dynamics. Note that the available experimental recordings correspond to stimuli with activation durations of 1s. NOBLE also achieves errors of 3% for `spikecount`, 32.8% for `AP1_width`, 8.54% for `mean_AP_amplitude`, and 0.83% for `steady_state_voltage`. To further assess NOBLE’s applicability across neuron types, we trained it on a VIP neuron using the same architecture and observed similarly strong performance (see Appendix D.3 for more details).

The relatively higher error on `AP1_width` arises from how the feature is computed: it measures the width of the first spike at half amplitude, where the half level is defined between the spike peak and the subsequent after-hyperpolarization minimum. If the predicted peak or minimum is slightly misaligned relative to the ground truth, the half-voltage reference shifts, and the measured width corresponds to a different portion of the trace. This sensitivity makes `AP1_width` less stable to small deviations, so its relative error should be interpreted with caution compared to the other features.

We also generate the F-I curves using the trained NOBLE for  $\{\text{HoF}^{\text{train}}\}$  and compare them with the reference F-I curves produced by the numerical solver for the same HoF models. As shown in Figure 3A, the curves from both methods remain close overall, although for 3 out of the 50 HoF models the NOBLE predictions show larger deviations in firing rate between  $0.0 - 0.1\text{nA}$ .



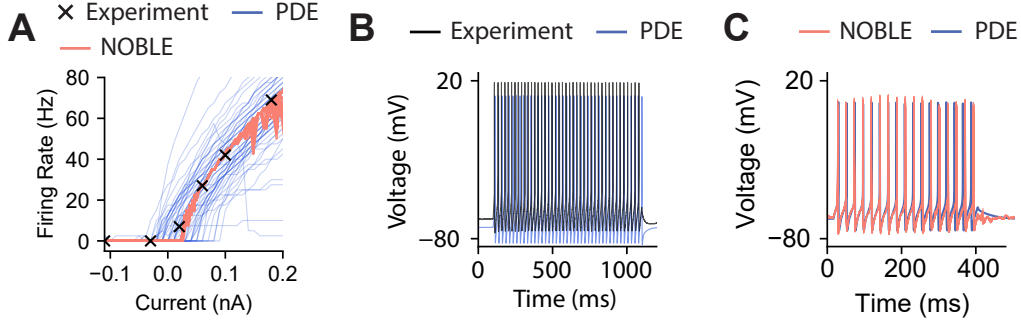


Figure 4: **A)** F-I curves from experimental recordings, PDE simulations, and NOBLE predictions on 50 interpolated HoF models. **B)** Experimental response vs. numerical simulation after interpolating in between PDE parameterizations. **C)** Numerical simulation of  $\text{HoF}_k^{\text{test}}$  vs. distribution of 50 NOBLE predictions after interpolating within the model latent space near the  $(I_{\text{thr}}, s_{\text{thr}})$  features of  $\text{HoF}_k^{\text{test}}$ .

### 4.3 Interpolating Between Models

We now test NOBLE’s ability to interpolate between HoF models within the convex hull  $\mathcal{CH}_{\text{train}}$  of the  $\{\text{HoF}^{\text{train}}\}$  models used for training. Consider a PDE model  $\text{HoF}_k^{\text{test}}$  excluded during training.

We first randomly sample 50 unseen synthetic models from a local neighborhood in the model latent space near the defining  $(I_{\text{thr}}, s_{\text{thr}})$  features of  $\text{HoF}_k^{\text{test}}$  (Figure 11 illustrates this neighborhood). Figure 4C shows that NOBLE accurately captures neuronal dynamics when interpolating within the latent space, as it produces a distribution of voltage responses consistent with those of the previously unseen  $\text{HoF}_k^{\text{test}}$ . In addition, Figure 4A shows that the F-I curves generated by NOBLE remain biophysically meaningful and closely aligned with experimentally observed neuronal behavior.

On the other hand, the parameterizations of the HoF models obtained using multi-objective evolutionary algorithms lack a consistent structure that would enable meaningful interpolation to discover new bio-realistic models. This is illustrated in Figure 4B, where the prediction made by interpolating in between PDE parameterizations deviates significantly from experimental data. This can also be observed in Figure 4A, where the F-I curves obtained by numerically solving PDE models constructed from slightly perturbed parameterizations of  $\text{HoF}_k^{\text{test}}$  deviate markedly from experimental data.

Interpolating within the latent embedding space enables NOBLE to efficiently generate novel neuron models at scale, while providing up to  $4,200\times$  faster model predictions than the numerical solver (see Appendix E). Yet, constructing the initial set of bio-realistic HoF models remains time-consuming, prompting the question of how much diversity is truly needed for robust generalization. To examine this, we varied the number of HoF models included in  $\{\text{HoF}^{\text{train}}\}$  while keeping the dataset size fixed. Performance on voltage traces remained largely stable, but spike-related features degraded markedly as model diversity decreased, highlighting the importance of experiencing sufficient biophysical variability during training. Further details of this analysis are provided in Appendix D.6.

### 4.4 Ensemble Predictions

We now examine how the single trained NOBLE can be used for ensemble predictions. Given a current injection, we run 50 inferences in parallel of NOBLE for the  $\{\text{HoF}^{\text{train}}\}$  models to produce 50 somatic responses. In Figure 5A (left, middle), we compare these 50 predictions with the corresponding 50 numerical solver simulations from  $\{\text{HoF}^{\text{train}}\}$ . We see that the distribution of curves is very similar.

Since NOBLE enables interpolation between the HoF models used for training, it can generate novel bio-realistic neuron models and produce voltage responses for any neuron model whose latent space representation lies within the convex hull  $\mathcal{CH}_{\text{train}}$  of the training set  $\{\text{HoF}^{\text{train}}\}$ . We demonstrate this by querying 200 novel models whose features are sampled randomly within  $\mathcal{CH}_{\text{train}}$ . Figure 5A (right) shows that the distribution of curves remains very similar, but the additional samples provide a denser coverage of the response space while maintaining bio-realism, with no artifacts or implausible predictions. We include additional ensemble analyses in Appendix D.5. There, we first validate NOBLE on the test models  $\{\text{HoF}^{\text{test}}\}$ , where predicted and ground-truth responses again show close agreement (Figure 12). Second, we examine local perturbations in the latent embedding space by sampling 50 synthetic models from a small circle around a held-out test model (Figure 11). The resulting responses (Figure 13) resemble small perturbations around the unseen ground-truth trace, demonstrating NOBLE’s ability to generalize to unseen models and the smoothness of the latent space.



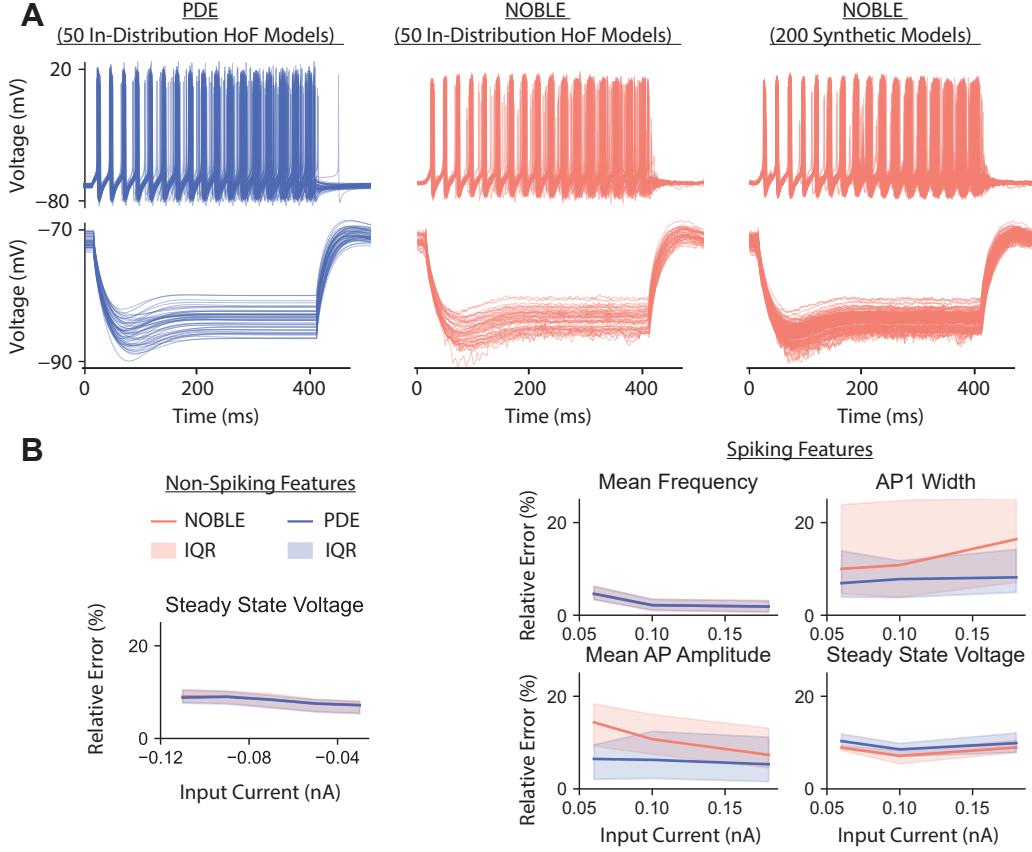


Figure 5: **A)** Distributions of somatic voltage traces across HoF and synthetic models for current injections of 0.1 nA (top) and -0.11 nA (bottom). **B)** Relative errors of ensemble predictions from PDE simulations and NOBLE models on  $\{\text{HoF}^{\text{train}}\}$  compared to experimental data across key features.

Figure 5B shows that NOBLE’s ensemble predictions for the full collection of HoF models achieve comparable accuracy to the HoF models themselves when evaluated on the four key electrophysiological features relative to experimental data. Here, we report the `mean_frequency`, representing the average firing rate, to account for the difference in stimulus activation durations between the experimental recordings and NOBLE, since `spikecount` is a nonlinear function of time and cannot be directly rescaled. These results demonstrate NOBLE’s ability to faithfully represent a diverse set of bio-realistic neuron models while enabling dense interpolation across the model space. However, experimental recordings for this neuron are limited to a single trace across nine amplitudes. Consequently, our comparison is against a single realization rather than a distribution. In practice, experimental features exhibit trial-to-trial variability, which NOBLE is designed to capture, but cannot be directly validated here due to limited experimental data availability.

#### 4.5 Choice of Biologically-Informed Latent Embeddings

Feature embeddings are central to NOBLE as they enable generalization across and in-between biological neuron models. Here, we have chosen  $I_{thr}$  and  $s_{thr}$  for their strong biological interpretability, as a natural 2D representation of firing and non-firing dynamics. To quantify the importance of these embeddings, we conduct an ablation study evaluating NOBLE with lower-dimensional embeddings, as well as without embeddings. We also consider a higher-dimensional embedding that includes `AHP_depth`, selected for its large variation across intracell HoF models and its low correlation with  $I_{thr}$  and  $s_{thr}$ . Results are reported in Table 1 in terms of the relative L2 error of predicted voltage traces and the four key electrophysiological features on the test set with  $\{\text{HoF}^{\text{train}}\}$  models.

Table 1: Relative L2 error of NOBLE on voltage traces and the four key features, when trained with different embedded features. Results are evaluated on the test set with  $\{\text{HoF}^{\text{train}}\}$  models.

Features embedded	Voltage	Steady state voltage	Spikecount	AP1 width	Mean AP amplitude
None	12.1%	1.31%	Never fires	Never fires	Never fires
$s_{thr}$	2.83%	1.33%	4.9%	233%	13%
$I_{thr}$	2.73%	1.20%	4.4%	107%	14%
$s_{thr}, I_{thr}$	<b>1.92%</b>	<b>1.02%</b>	<b>3.1%</b>	27%	<b>8.9%</b>
$s_{thr}, I_{thr}, \text{AHP\_depth}$	2.16%	1.04%	3.3%	<b>22%</b>	9.5%

Without embeddings, NOBLE fails to predict any firing responses, indicating that embeddings are necessary to capture spiking behavior. With a single embedded feature, either  $s_{thr}$  or  $I_{thr}$ , NOBLE achieves similar accuracy on voltage traces and `steady_state_voltage`, but errors on spike-related features remain large, with  $I_{thr}$  providing better accuracy for `AP1_width`. As discussed earlier, this feature is particularly sensitive to small misalignments in spike peak and after-hyperpolarization minima, which makes its relative error less robust as a metric. Embedding both  $s_{thr}$  and  $I_{thr}$  yields the best overall performance across all features as well as the voltage trace. Extending the embedding space with `AHP_depth` slightly improves `AP1_width`, but reduces accuracy for the other features. These results show that feature embeddings are important for NOBLE to capture both firing and non-firing dynamics, and more broadly for its ability to generalize across diverse neuron models.

While we only considered constructing the latent space from electrophysiological features, NOBLE can readily incorporate additional modalities such as gene expression profiles from patch-sequencing data [57]. Learning joint embeddings across modalities could yield a unified latent space linking gene expression, electrophysiology, and morphology, providing a means to test hypotheses that are infeasible experimentally, such as how genes associated with neurological diseases [50] influence neuronal dynamics. In doing so, NOBLE would pave the way for improved statistical analysis, more reliable uncertainty quantification, and robust predictive modeling of neuronal behavior.

#### 4.6 Feature Specific Physics-Informed Fine-Tuning of NOBLE

To preserve overall neural dynamics while improving accuracy on a single specific feature, NOBLE can be fine-tuned with a weighted composite loss  $\mathcal{L}(\lambda) = \mathcal{L}_{\text{data}} + \lambda \mathcal{L}_F$ , where  $\mathcal{L}_F$  penalizes deviations in feature  $F$ , and  $\lambda$  controls its influence. We illustrate this by fine-tuning NOBLE to improve `sag_amplitude` accuracy, a feature reflecting the hyperpolarization-activated cation channel (Ih). In the human cortex, Ih expression varies with cortical depth, making `sag_amplitude` a relevant physiological marker. We fine-tune on 19,600 subthreshold stimuli with negative amplitudes. Even without a feature-specific loss, fine-tuning reduces the L2 feature error from 70% to 19.2%, and incorporating the feature loss  $\mathcal{L}_F$  further lowers it to 9.6% while preserving overall signal fidelity. These results show that NOBLE can be selectively refined to prioritize biophysical features of interest without compromising overall performance. Further details are provided in Appendix D.7.

## 5 Conclusion

We introduced NOBLE, a neural operator framework for learning the nonlinear somatic dynamics across a population of HoF models for a single neuron. Rather than training separate surrogates for each bio-realistic model, NOBLE learns a single neural operator that captures the inherent variability observed in experimental neuron recordings by mapping biologically interpretable embeddings to voltage responses from current injections. Demonstrated on PVALB and VIP neurons, NOBLE correctly captured the diverse neuron dynamics with a 4200x speedup over traditional solvers, while maintaining accuracy across key electrophysiological features. Importantly, our work is among the first to benchmark a deep learning based method for predicting membrane potential responses to intracellular current injections against experimental data from the human cortex. NOBLE also allows for generating novel, bio-realistic neuron models through interpolation in the latent space, which is not feasible with HoF models. This allows for realistic and efficient ensemble predictions beyond the original set of HoF models. NOBLE’s interpretable latent space also offers new insights into how neuron characteristics affect neuron dynamics. NOBLE opens a pathway toward modeling larger-scale brain circuits and leveraging multimodal latent spaces to determine relationships between gene expression, electrophysiology, and morphology, as discussed in Appendix F.

## Acknowledgements

L.G. was responsible for the complete technical implementation of this work. C.A.A. and A.A. conceptualized this work. L.G., V.D., and B.T. jointly developed the methodology of the novel NOBLE framework. P.H.W. and C.A.A. provided neuroscience-specific domain expertise, contextualizing the relevance and impact of this work within the broader field of neuroscience. P.H.W. supplied the biophysical PDE models and developed the multi-objective optimization pipeline. L.G. and P.H.W. produced the figures. L.G., V.D., B.T., P.H.W., and C.A.A. co-wrote the manuscript. C.A.A. and A.A. provided supervision and editorial comments.

## Funding

A.A. is supported by the Bren Endowed Chair, ONR (MURI grant N00014-23-1-2654), and the AI2050 Senior Fellow program at Schmidt Sciences. C.A.A. is supported by the National Institutes of Health R01 - NS120300 and R01 - NS130126. P.H.W. is supported by the National Institutes of Health R01 - NS130126.

## References

- [1] Rodney J Douglas, Christof Koch, Misha Mahowald, Kevan AC Martin, and Humbert H Suarez. Recurrent excitation in neocortical circuits. *Science*, 269(5226):981–985, 1995.
- [2] György Buzsáki. Neural syntax: cell assemblies, synapsembles, and readers. *Neuron*, 68(3): 362–385, 2010.
- [3] Liqun Luo. Architectures of neuronal circuits. *Science*, 373(6559):eabg7285, 2021.
- [4] Kimberly Siletti, Rebecca Hodge, Alejandro Mossi Albiach, Ka Wai Lee, Song-Lin Ding, Lijuan Hu, Peter Lönnerberg, Trygve Bakken, Tamara Casper, Michael Clark, et al. Transcriptomic diversity of cell types across the adult human brain. *Science*, 382(6667):eadd7046, 2023.
- [5] Xi-Han Zhang, Kevin M Anderson, Hao-Ming Dong, Sidhant Chopra, Elvisha Dhamala, Prashant S Emani, Mark B Gerstein, Daniel S Margulies, and Avram J Holmes. The cell-type underpinnings of the human functional cortical connectome. *Nature Neuroscience*, 28(1): 150–160, 2025.
- [6] Jim Berg, Staci A Sorensen, Jonathan T Ting, Jeremy A Miller, Thomas Chartrand, Anatoly Buchin, Trygve E Bakken, Agata Budzillo, Nick Dee, Song-Lin Ding, et al. Human neocortical expansion involves glutamatergic neuron diversification. *Nature*, 598(7879):151–158, 2021.
- [7] Thomas Chartrand, Rachel Dalley, Jennie Close, Natalia A Goriounova, Brian R Lee, Rusty Mann, Jeremy A Miller, Gabor Molnar, Alice Mukora, Lauren Alfiler, et al. Morphoelectric and transcriptomic divergence of the layer 1 interneuron repertoire in human versus mouse neocortex. *Science*, 382(6667):eadf0805, 2023.
- [8] Nathan W Gouwens, Staci A Sorensen, Fahimeh Baftizadeh, Agata Budzillo, Brian R Lee, Tim Jarsky, Lauren Alfiler, Katherine Baker, Eliza Barkan, Kyla Berry, et al. Integrated morphoelectric and transcriptomic classification of cortical gabaergic cells. *Cell*, 183(4): 935–953, 2020.
- [9] Brian R Lee, Rachel Dalley, Jeremy A Miller, Thomas Chartrand, Jennie Close, Rusty Mann, Alice Mukora, Lindsay Ng, Lauren Alfiler, Katherine Baker, et al. Signature morphoelectric properties of diverse gabaergic interneurons in the human neocortex. *Science*, 382(6667): eadf6484, 2023.
- [10] Susan M Sunkin, Lydia Ng, Chris Lau, Tim Dolbeare, Terri L Gilbert, Carol L Thompson, Michael Hawrylycz, and Chinh Dang. Allen brain atlas: an integrated spatio-temporal portal for exploring the central nervous system. *Nucleic acids research*, 41(D1):D996–D1008, 2012.
- [11] Michael W Reimann, Costas A Anastassiou, Rodrigo Perin, Sean L Hill, Henry Markram, and Christof Koch. A biophysically detailed model of neocortical local field potentials predicts the critical role of active membrane currents. *Neuron*, 79(2):375–390, 2013.

- [12] Anatoly Buchin, Rebecca de Frates, Anirban Nandi, Rusty Mann, Peter Chong, Lindsay Ng, Jeremy Miller, Rebecca Hodge, Brian Kalmbach, Soumita Bose, et al. Multi-modal characterization and simulation of human epileptic circuitry. *Cell reports*, 41(13), 2022.
- [13] Anirban Nandi, Thomas Chartrand, Werner Van Geit, Anatoly Buchin, Zizhen Yao, Soo Yeun Lee, Yina Wei, Brian Kalmbach, Brian Lee, Ed Lein, et al. Single-neuron models linking electrophysiology, morphology, and transcriptomics across cortical cell types. *Cell reports*, 40(6), 2022.
- [14] Kaisa Miettinen. *Nonlinear multiobjective optimization*, volume 12. Springer Science & Business Media, 1999.
- [15] Shaul Druckmann, Yoav Banitt, Albert A Gidon, Felix Schürmann, Henry Markram, and Idan Segev. A novel multiple objective optimization framework for constraining conductance-based neuron models by experimental data. *Frontiers in neuroscience*, 1:56, 2007.
- [16] Werner Van Geit, Michael Gevaert, Giuseppe Chindemi, Christian Rössert, Jean-Denis Courcol, Eilif B Muller, Felix Schürmann, Idan Segev, and Henry Markram. Bluepyopt: leveraging open source software and cloud infrastructure to optimise model parameters in neuroscience. *Frontiers in neuroinformatics*, 10:17, 2016.
- [17] Clayton P Mosher, Yina Wei, Jan Kamiński, Anirban Nandi, Adam N Mamelak, Costas A Anastassiou, and Ueli Rutishauser. Cellular classes in the human brain revealed in vivo by heartbeat-related modulation of the extracellular action potential waveform. *Cell reports*, 30(10):3536–3551, 2020.
- [18] Pablo Achard and Erik De Schutter. Complex parameter landscape for a complex neuron model. *PLoS computational biology*, 2(7):e94, 2006.
- [19] Liam Paninski, Jonathan Pillow, and Jeremy Lewi. Statistical models for neural encoding, decoding, and optimal stimulus design. *Progress in brain research*, 165:493–507, 2007.
- [20] William D O’Neill, James C Lin, and Ying-Chang Ma. Estimation and verification of a stochastic neuron model. *IEEE transactions on biomedical engineering*, (7):654–666, 2007.
- [21] Shinsuke Koyama and Robert E Kass. Spike train probability models for stimulus-driven leaky integrate-and-fire neurons. *Neural computation*, 20(7):1776–1795, 2008.
- [22] Elad Schneidman, Barry Freedman, and Idan Segev. Ion channel stochasticity may be critical in determining the reliability and precision of spike timing. *Neural computation*, 10(7):1679–1703, 1998.
- [23] John A White, Jay T Rubinstein, and Alan R Kay. Channel noise in neurons. *Trends in neurosciences*, 23(3):131–137, 2000.
- [24] Joshua H Goldwyn and Eric Shea-Brown. The what and where of adding channel noise to the hodgkin-huxley equations. *PLoS computational biology*, 7(11):e1002247, 2011.
- [25] Kamyar Azizzadenesheli, Nikola Kovachki, Zongyi Li, Miguel Liu-Schiaffini, Jean Kossaifi, and Anima Anandkumar. Neural operators for accelerating scientific simulations and design. *Nature Reviews Physics*, pages 1–9, 2024.
- [26] Nikola Kovachki, Zongyi Li, Burigede Liu, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Learning maps between function spaces with applications to pdes. *Journal of Machine Learning Research*, 24(89):1–97, 2023.
- [27] Julius Berner, Miguel Liu-Schiaffini, Jean Kossaifi, Valentin Duruisseaux, Boris Bonev, Kamyar Azizzadenesheli, and Anima Anandkumar. Principled approaches for extending neural architectures to function spaces for operator learning. 2025. URL <https://arxiv.org/abs/2506.10973>.
- [28] Nikola Kovachki, Samuel Lanthaler, and Siddhartha Mishra. On universal approximation and error bounds for Fourier neural operators. *J. Mach. Learn. Res.*, 22(1), 2021. ISSN 1532-4435.

- [29] Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. *arXiv preprint arXiv:2010.08895*, 2020.
- [30] Valentin Duruisseaux, Jean Kossaifi, and Anima Anandkumar. Fourier neural operators explained: A practical perspective, 2025. URL <https://arxiv.org/abs/2512.01421>.
- [31] Richard FitzHugh. Mathematical models of threshold phenomena in the nerve membrane. *The bulletin of mathematical biophysics*, 17:257–278, 1955.
- [32] Jinichi Nagumo, Suguru Arimoto, and Shuji Yoshizawa. An active pulse transmission line simulating nerve axon. *Proceedings of the IRE*, 50(10):2061–2070, 1962.
- [33] 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.
- [34] Johann Rudi, Julie Bessac, and Amanda Lenzi. Parameter estimation with dense and convolutional neural networks applied to the fitzhugh–nagumo ode. In *Mathematical and Scientific Machine Learning*, pages 781–808. PMLR, 2022.
- [35] Pavel V Kuptsov, Nataliya V Stankevich, and Elmira R Bagautdinova. Discovering dynamical features of hodgkin–huxley-type model of physiological neuron using artificial neural network. *Chaos, Solitons & Fractals*, 167:113027, 2023.
- [36] Wei-Hung Su, Ching-Shan Chou, and Dongbin Xiu. Deep learning of biological models from data: applications to ode models. *Bulletin of mathematical biology*, 83:1–19, 2021.
- [37] M. Raissi, P. Perdikaris, and G.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. ISSN 0021-9991. doi: 10.1016/j.jcp.2018.10.045.
- [38] Maziar Raissi, Paris Perdikaris, and George E. Karniadakis. Physics informed deep learning (part i): Data-driven solutions of nonlinear partial differential equations. *ArXiv*, abs/1711.10561, 2017.
- [39] Maziar Raissi, Paris Perdikaris, and George E. Karniadakis. Physics informed deep learning (part ii): Data-driven discovery of nonlinear partial differential equations. *ArXiv*, abs/1711.10566, 2017.
- [40] Yan Barbosa Werneck, Rodrigo Weber dos Santos, Bernardo Martins Rocha, and Rafael Sachetto Oliveira. Replacing the fitzhugh-nagumo electrophysiology model by physics-informed neural networks. In *International Conference on Computational Science*, pages 699–713. Springer, 2023.
- [41] Matteo Ferrante, Andera Duggento, and Nicola Toschi. Physically constrained neural networks to solve the inverse problem for neuron models. *arXiv preprint arXiv:2209.11998*, 2022.
- [42] Himanshu Pandey, Anshima Singh, and Ratikanta Behera. An efficient wavelet-based physics-informed neural networks for singularly perturbed problems. *arXiv preprint arXiv:2409.11847*, 2024.
- [43] Edoardo Centofanti, Massimiliano Ghiotto, and Luca F Pavarino. Learning the hodgkin–huxley model with operator learning techniques. *Computer Methods in Applied Mechanics and Engineering*, 432:117381, 2024.
- [44] Lu Mi, Trung Le, Tianxing He, Eli Shlizerman, and Uygur Sümbül. Learning time-invariant representations for individual neurons from population dynamics. *Advances in Neural Information Processing Systems*, 36:46007–46026, 2023.
- [45] Sergey L Gratiy, Yazan N Billeh, Kael Dai, Catalin Mitelut, David Feng, Nathan W Gouwens, Nicholas Cain, Christof Koch, Costas A Anastassiou, and Anton Arkhipov. Bionet: A python interface to neuron for modeling large-scale networks. *PloS one*, 13(8):e0201630, 2018.

- [46] Michael L Hines and Nicholas T Carnevale. Neuron: a tool for neuroscientists. *The neuroscientist*, 7(2):123–135, 2001.
- [47] Christof Koch. *Biophysics of computation: information processing in single neurons*. Oxford university press, 2004.
- [48] Tae Kim, Stephen Thankachan, James T McKenna, James M McNally, Chun Yang, Jee Hyun Choi, Lichao Chen, Bernat Kocsis, Karl Deisseroth, Robert E Strecker, et al. Cortically projecting basal forebrain parvalbumin neurons regulate cortical gamma band oscillations. *Proceedings of the National Academy of Sciences*, 112(11):3535–3540, 2015.
- [49] Guillermo Gonzalez-Burgos, Raymond Y Cho, and David A Lewis. Alterations in cortical network oscillations and parvalbumin neurons in schizophrenia. *Biological psychiatry*, 77(12):1031–1040, 2015.
- [50] Mariano I Gabitto, Kyle J Travaglini, Victoria M Rachleff, Eitan S Kaplan, Brian Long, Jeanelle Ariza, Yi Ding, Joseph T Mahoney, Nick Dee, Jeff Goldy, et al. Integrated multimodal cell atlas of alzheimer’s disease. *Nature Neuroscience*, 27(12):2366–2383, 2024.
- [51] Alvar Prönnke, Bianca Scheuer, Robin J Wagener, Martin Möck, Mirko Witte, and Jochen F Staiger. Characterizing vip neurons in the barrel cortex of vipcre/tdtomato mice reveals layer-specific differences. *Cerebral cortex*, 25(12):4854–4868, 2015.
- [52] Julia Veit, Gregory Handy, Daniel P Mossing, Brent Doiron, and Hillel Adesnik. Cortical vip neurons locally control the gain but globally control the coherence of gamma band rhythms. *Neuron*, 111(3):405–417, 2023.
- [53] Hyun-Jae Pi, Balázs Hangya, Duda Kvitsiani, Joshua I Sanders, Z Josh Huang, and Adam Kepecs. Cortical interneurons that specialize in disinhibitory control. *Nature*, 503(7477):521–524, 2013.
- [54] Alan L Hodgkin. The local electric changes associated with repetitive action in a non-medullated axon. *The Journal of physiology*, 107(2):165, 1948.
- [55] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.
- [56] Jean Kossaifi, Nikola Kovachki, Zongyi Li, David Pitt, Miguel Liu-Schiaffini, Robert Joseph George, Boris Bonev, Kamyar Azizzadenesheli, Julius Berner, Valentin Duruisseaux, and Anima Anandkumar. A library for learning neural operators, 2024.
- [57] Cathryn R Cadwell, Athanasia Palasantza, Xiaolong Jiang, Philipp Berens, Qiaolin Deng, Marlene Yilmaz, Jacob Reimer, Shan Shen, Matthias Bethge, Kimberley F Tolias, et al. Electrophysiological, transcriptomic and morphologic profiling of single neurons using patch-seq. *Nature biotechnology*, 34(2):199–203, 2016.
- [58] Zongyi Li, Hongkai Zheng, Nikola Kovachki, David Jin, Haoxuan Chen, Burigede Liu, Kamyar Azizzadenesheli, and Anima Anandkumar. Physics-informed neural operator for learning partial differential equations. *ACM/JMS Journal of Data Science*, 1(3):1–27, 2024.
- [59] Ryan Y. Lin, Julius Berner, Valentin Duruisseaux, David Pitt, Daniel Leibovici, Jean Kossaifi, Kamyar Azizzadenesheli, and Anima Anandkumar. Enabling automatic differentiation with mollified graph neural operators, 2025.
- [60] Adarsh Ganeshram, Haydn Maust, Valentin Duruisseaux, Zongyi Li, Yixuan Wang, Daniel Leibovici, Oscar Bruno, Thomas Hou, and Anima Anandkumar. Fc-pino: High precision physics-informed neural operators via fourier continuation, 2025.
- [61] Vignesh Gopakumar, Stanislas Pamela, Lorenzo Zanisi, Zongyi Li, Anima Anandkumar, and MAST Team. Fourier neural operator for plasma modelling. *arXiv preprint arXiv:2302.06542*, 2023.



- [62] Thorsten Kurth, Shashank Subramanian, Peter Harrington, Jaideep Pathak, Morteza Mardani, David Hall, Andrea Miele, Karthik Kashinath, and Animashree Anandkumar. FourCastNet: Accelerating global high-resolution weather forecasting using adaptive Fourier neural operators. 2022. doi: 10.48550/arXiv.2208.05419.
- [63] Gege Wen, Zongyi Li, Qirui Long, Kamyar Azizzadenesheli, Anima Anandkumar, and Sally M. Benson. Real-time high-resolution CO<sub>2</sub> geological storage prediction using nested Fourier neural operators. *Energy Environ. Sci.*, 16:1732–1741, 2023. doi: 10.1039/D2EE04204E.
- [64] Zongyi Li, Nikola Borislavov Kovachki, Chris Choy, Boyi Li, Jean Kossaifi, Shourya Prakash Otta, Mohammad Amin Nabian, Maximilian Stadler, Christian Hundt, Kamyar Azizzadenesheli, and Anima Anandkumar. Geometry-informed neural operator for large-scale 3D PDEs, 2023.

## A Related Works in Machine Learning for Single-Cell Electrophysiology

Earlier applications of machine learning to single-cell electrophysiology focused on directly learning the dynamics of canonical point-neuron models like FitzHugh-Nagumo [31, 32] and Hodgkin-Huxley [33]. Fully connected and convolutional neural networks were trained to reproduce FitzHugh-Nagumo dynamics [34], while ResNet-based multilayer perceptrons showed promise in learning Hodgkin-Huxley dynamics [35, 36]. Physics-informed neural networks (PINNs) extend conventional neural networks by introducing prior knowledge about the underlying dynamical system [37–39]. This formulation was used to predict FitzHugh-Nagumo dynamics [40] and to learn the Hodgkin-Huxley model ionic conductances from simulated voltage recordings [41]. Further refinements with PINNs incorporated wavelet bases to capture localized multiscale dynamics and compute derivatives analytically, improving both accuracy and convergence speed when training on the FitzHugh-Nagumo model [42]. While these methods demonstrate capabilities in capturing the core spiking dynamics of these simplified models, their ability to accurately represent the full spectrum of electrophysiological behavior, particularly the highly nonlinear onset of firing, remains largely untested. Moreover, as function approximators, they necessitate retraining for each new input stimulus, significantly limiting their practical utility.

To address some of these limitations, Centofanti et al. [43] explored using operator learning approaches for forward simulations of the Hodgkin-Huxley model. Among other approaches, FNOs showed promising results by demonstrating a strong capacity for learning the governing operator of this biophysical system. However, this work still exhibits key limitations and a limited scope: (1) it relies on relatively simple simulated data from a point-neuron model, (2) it does not explicitly attempt to capture the full spectrum of electrophysiological dynamics, particularly the highly nonlinear onset of firing, and (3) its formulation on a single operator inherently lacks the capacity to represent the trial-to-trial variability observed in biological recordings. Related work such as NeuPRINT [44] also leverages deep learning to capture trial-to-trial variability. However, it operates on slower in vivo 2-photon calcium imaging data and models population-level fluorescence dynamics, rather than the fast intracellular voltage dynamics of individual neurons.

## B Electrophysiological Features

For electrophysiological feature extraction and metrics, we use code from the Electrophys Feature Extraction Library (eFEL) available at

<https://github.com/BlueBrain/eFEL>

The formulas, codes, and more details about each electrophysiological feature can be found at

<https://efel.readthedocs.io/en/latest/eFeatures.html>

We list below 16 important electrophysiological features and metrics of interest when constructing neuron models (where AP denotes action potential and AHP denotes after-hyperpolarization):

- **AHP\_depth**: Relative voltage values at the first AHP
- **AHP\_time\_from\_peak**: Time between AP peaks and first AHP depth
- **AHP1\_depth\_from\_peak**: Voltage difference between the first AP peak and first AHP depth
- **AP1\_peak**: The peak voltage of the first AP
- **AP1\_width**: Width of first spike at half spike amplitude, with the spike amplitude taken as the difference between the minimum between two peaks and the next peak

- `decay_time_constant_after_stim`: The decay time constant of the voltage right after the stimulus
- `depol_block`: Check for a depolarization block. Returns 1 if there is a depolarization block or a hyperpolarization block, and returns 0 otherwise.
- `inv_first_ISI`: 1.0 over first interspike interval; returns 0 when no interspike interval
- `mean_AP_amplitude`: The mean of all of the AP amplitudes
- `mean_frequency`: The mean frequency of the firing rate
- `sag_amplitude`: The difference between the minimal voltage and the steady state at the end of the stimulus
- `spikecount`: Number of spikes in the trace, including outside of stimulus interval
- `steady_state_voltage`: The average voltage after the stimulus
- `steady_state_voltage_stimend`: The average voltage during the last 10% of the stimulus duration.
- `time_to_first_spike`: Time from the start of the stimulus to the maximum of the first peak
- `voltage_base`: The average voltage during the last 10% of time before the stimulus

## C Method

### C.1 Impact of Subsampling on Neuron Features

We present the results of the analysis conducted to determine the maximum subsampling factor that preserves the fidelity of extracted neuron features, mentioned in Section 3.1. The results are displayed in Figures 6 and 7 for the low-pass filtering followed by decimation in time subsampling strategy.

We first computed the relative error between the raw, non-subsampled HoF model voltage responses and the experimental data across all amplitudes. For each amplitude, we identified the minimum relative error across all HoF models, and then aggregated these minima to compute the mean and standard deviation. These serve as a reference for the inherent *worst-case* discrepancy between simulations and experimental recordings in the absence of any subsampling. We visualize the mean as a solid black line and the standard deviation as dotted black lines.

Next, we repeated a similar analysis to quantify the additional relative errors introduced by subsampling. For each subsampling factor, we calculated the relative error between the subsampled and original HoF responses for each amplitude. These errors were then averaged across all HoF models, and the distribution of these averages is summarized using the mean (solid line), standard deviation (shaded region), and min/max (error bars).

This study shows that for most electrophysiological features, subsampling introduces negligible additional error. The most sensitive features were AP1\_Width and AP1\_Peak, which exhibited noticeable deviations at higher subsampling rates. To ensure we remain within the bounds of the intrinsic simulation-experiment discrepancy, we adopt a conservative downsampling factor of  $3\times$ , which maintains fidelity while reducing computational load.

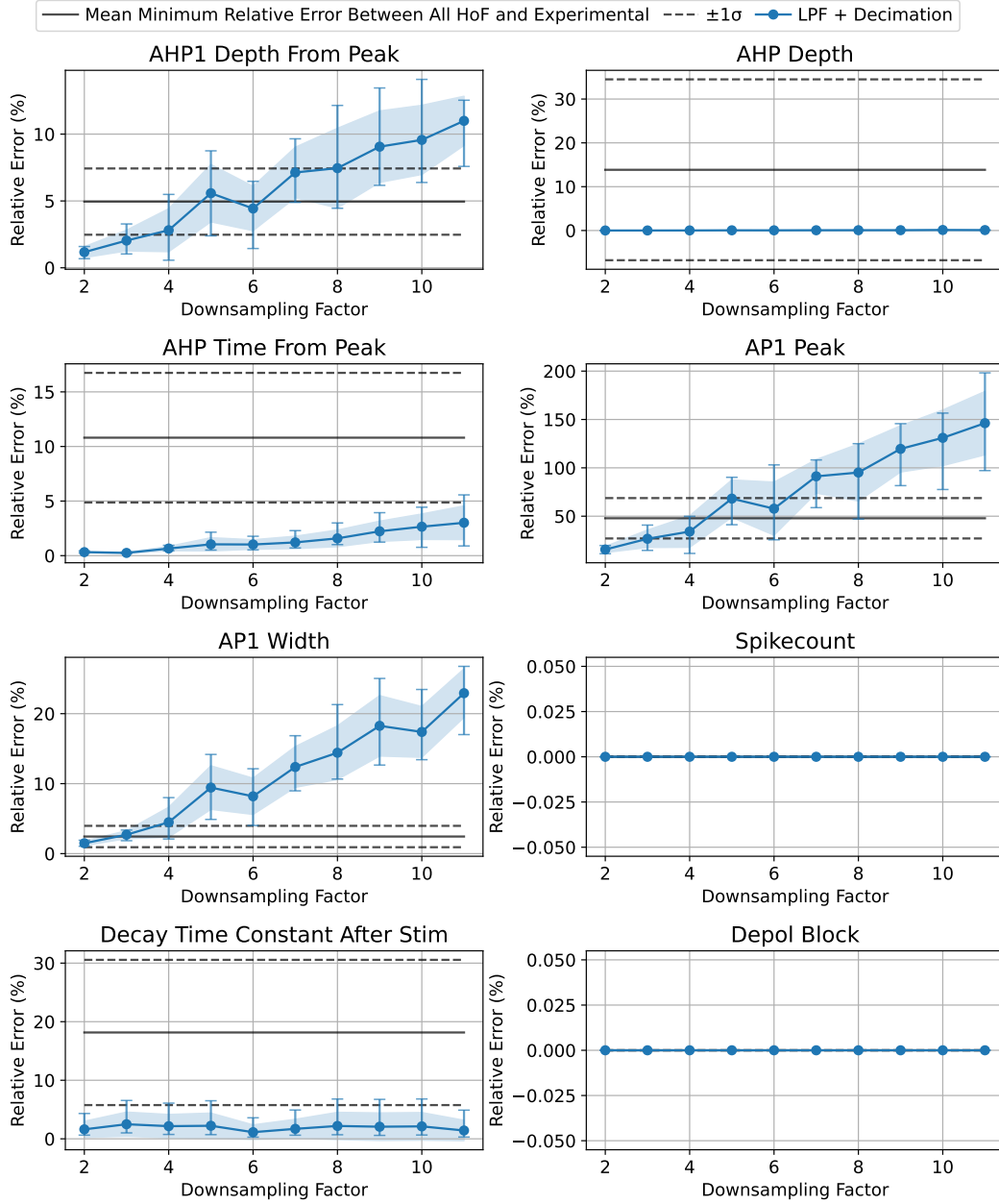


Figure 6: Analysis of the relative errors introduced in neuron feature computation as a function of subsampling factor, using low-pass filtering followed by decimation in time. The solid and dotted black lines indicate the mean and standard deviation, respectively, of the minimum relative error between non-subsampled HoF and experimental responses. The solid blue line, shaded region, and error bars represent the mean, standard deviation, and minimum–maximum statistics of the relative error between non-subsampled and subsampled HoF responses.

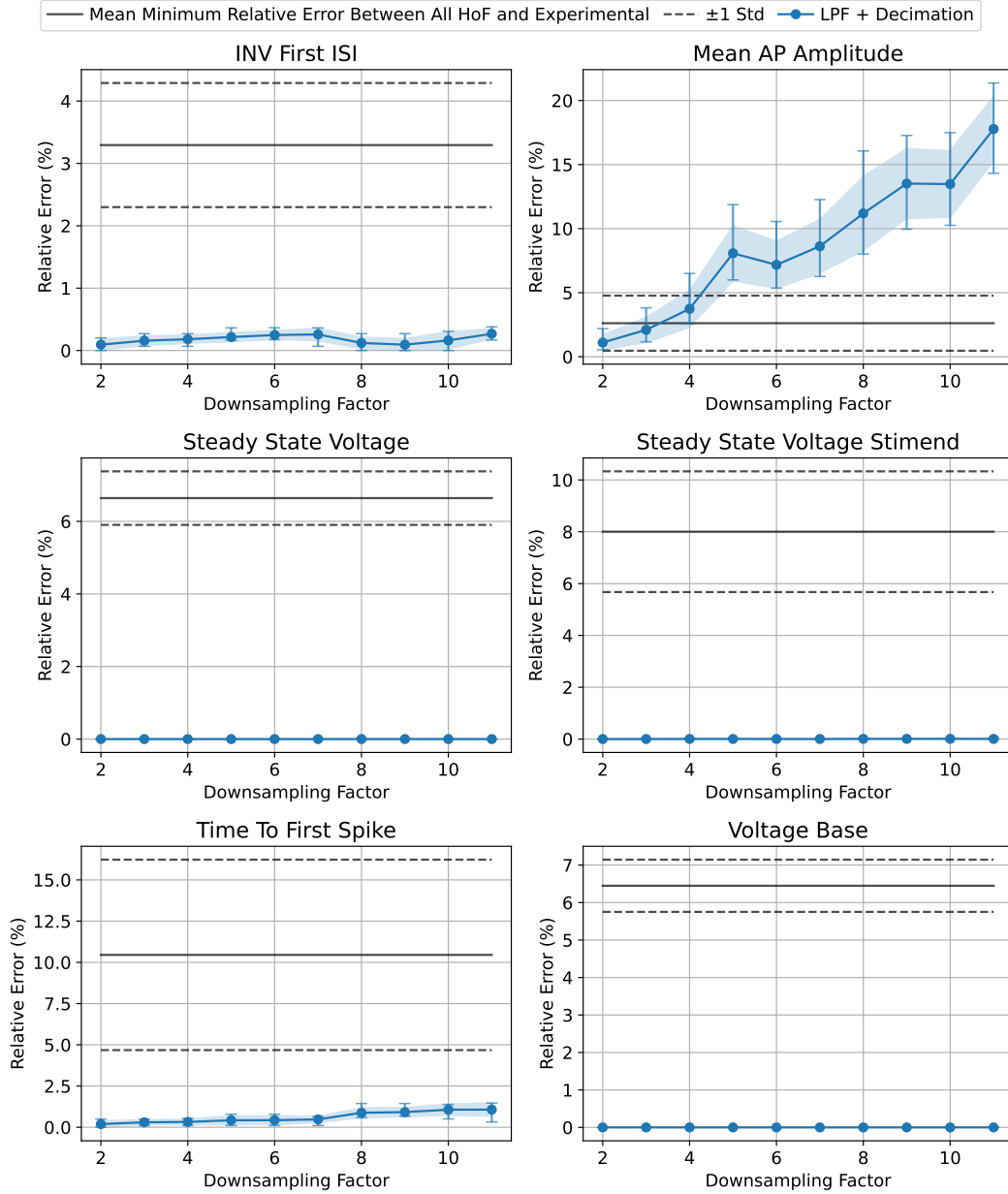


Figure 7: Analysis of the relative errors introduced in neuron feature computation as a function of subsampling factor, using low-pass filtering followed by decimation in time. The solid and dotted black lines indicate the mean and standard deviation, respectively, of the minimum relative error between non-sampled HoF and experimental responses. The solid blue line, shaded region, and error bars represent the mean, standard deviation, and minimum–maximum statistics of the relative error between non-sampled and subsampled HoF responses.

## C.2 Input Current Amplitude Distribution

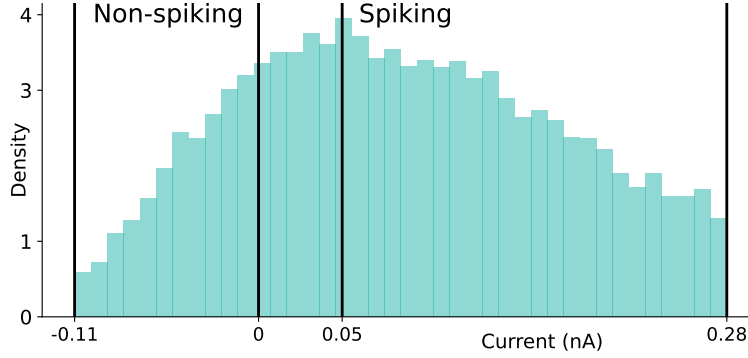


Figure 8: Distribution of square-pulse amplitudes in  $[-0.11, 0.28]$  nA considered. There is a spiking threshold (between 0 to 0.05 nA) where neuron responses transition from being non-spiking to spiking.

## C.3 The Fourier Neural Operator Architecture

Neural operators [25–27] are a principled way to generalize neural networks to learn operators mapping functions to functions, with a universal operator approximation property [28]. Neural operators compose linear integral operators  $\mathcal{K}$  with pointwise nonlinear activation functions  $\sigma$  to approximate highly nonlinear operators. More precisely, we define the neural operator

$$\mathcal{G}_\theta = \mathcal{Q} \circ \sigma(W_L + \mathcal{K}_L + b_L) \circ \cdots \circ \sigma(W_1 + \mathcal{K}_1 + b_1) \circ \mathcal{P} \quad (2)$$

where  $\mathcal{P}$ ,  $\mathcal{Q}$  are the pointwise neural networks that encode the lower dimension function into a higher-dimensional space and vice versa. The model stacks  $L$  layers of  $\sigma(W_l + \mathcal{K}_l + b_l)$  where  $W_l$  are pointwise linear operators (matrices),  $\mathcal{K}_l$  are integral kernel operators,  $b_l$  are bias terms, and  $\sigma$  are fixed activation functions. The parameters  $\theta$  consists of all the parameters in  $\mathcal{P}$ ,  $\mathcal{Q}$ ,  $W_l$ ,  $\mathcal{K}_l$  and  $b_l$ . Kossaifi et al. [56] maintain a comprehensive open-source PyTorch library for learning neural operators, which serves as the foundation for our implementation. Prior knowledge of the relevant physics laws and differential equations can also be incorporated as additional loss terms during training, to supplement or replace reference data, as done with physics-informed neural operators [58–60].

A variety of neural operators have been proposed, such as the Fourier Neural Operator (FNO) [29, 30], and successfully applied to a wide range of problems [61–63]. A FNO is a neural operator using Fourier integral operator layers, which are defined via

$$(\mathcal{K}(\phi)v_t)(x) = \mathcal{F}^{-1}\left(R_\phi \cdot (\mathcal{F}v_t)\right)(x) \quad (3)$$

where  $R_\phi$  is the Fourier transform of a periodic function  $\kappa$  parameterized by  $\phi$ . On a uniform mesh, the Fourier transform  $\mathcal{F}$  can be implemented using the fast Fourier transform (FFT).

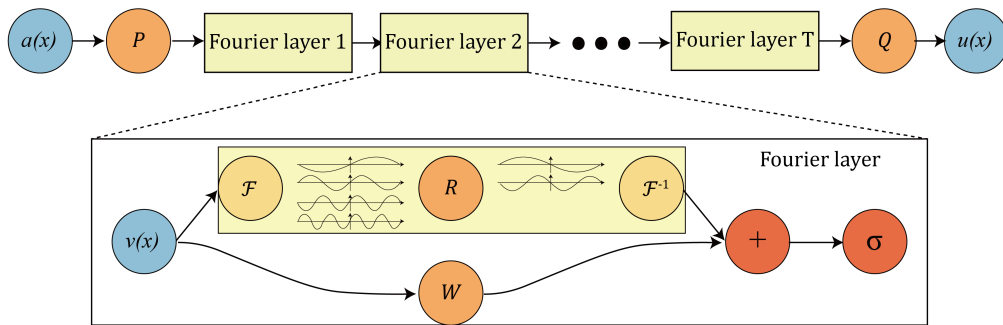


Figure 9: The Fourier Neural Operator (FNO) architecture (extracted from [29]).



## D Experiments

### D.1 Dataset

We have access to 60 HoF models  $\{\text{HoF}\}$  obtained using a multi-objective evolutionary optimization strategy. We use 50 of them during training,  $\{\text{HoF}^{\text{train}}\}$ , and keep the remaining 10  $\{\text{HoF}^{\text{test}}\}$  for testing.

Figure 10 displays the latent representations in normalized  $(\bar{I}_{thr}, \bar{S}_{thr})$ -space of these HoF models.

The training dataset is composed of 75,600 samples, where the current injections are sampled as described in Section 3.2, each of which is associated randomly to one of  $\{\text{HoF}^{\text{train}}\}$ . The samples are generated using a numerical solver [45] built on the NEURON simulation environment [46].

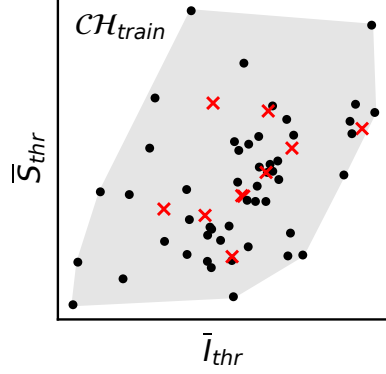


Figure 10: Latent representations of neuron models in the normalized  $(\bar{I}_{thr}, \bar{S}_{thr})$ -space. Black dots indicate the 50 training HoF models  $\{\text{HoF}^{\text{train}}\}$ , and red crosses the 10 test HoF models  $\{\text{HoF}^{\text{test}}\}$  excluded during training.

### D.2 Evaluation and Evaluation Metrics

NOBLE is trained using the relative L4 error, computed via

$$\text{Relative Lp error}(x, y; \epsilon) = \frac{\|x - y\|_p}{\|y\|_p + \epsilon} = \frac{(\sum_i |x_i - y_i|^p)^{1/p}}{(\sum_i |y_i|^p)^{1/p} + \epsilon}. \quad (4)$$

The relative L4 error was selected as the training loss after a preliminary training study on a small dataset, where it consistently preserved spike-related features, especially amplitudes and widths, more effectively than the relative L2 error. While this choice ensured the model captured the electrophysiological details most relevant to our setting, NOBLE is compatible with any other loss function that may be more suitable in different contexts.

Although NOBLE is trained to minimize the relative L4 error, we report results using the relative L2 error, as it provides a more common and interpretable measure of accuracy. We first evaluate performance on voltage traces. Note that even small temporal shifts between predicted and ground-truth voltage responses can result in large relative errors. To better evaluate physiologically meaningful behavior, we also report errors on four key electrophysiological features:

- `spikecount`: number of spikes in the trace
- `AP1_width`: width of the first spike at half amplitude
- `mean_AP_amplitude`: mean amplitude of all action potentials
- `steady_state_voltage`: average voltage after the stimulus

For benchmarking, we compare NOBLE predictions against numerical simulations obtained from HoF models and experimental data. HoF models were optimized to produce the closest approximations to experimental recordings and capture the biological variability required for a meaningful benchmark. In contrast, existing machine learning methods, such as the ones discussed in the introduction, are not designed to reproduce such variability and are therefore unsuitable as baselines. Furthermore, since the PDE solvers used to generate the dataset are themselves approximations with non-negligible error, driving prediction error below the solver–experiment gap risks overfitting to the solver rather than improving alignment with real recordings. Therefore, we tuned hyperparameters only up to the solver–experiment error level, as marginal gains on PDE data are unlikely to yield meaningful improvements relative to experimental recordings.

### D.3 Testing on VIP Neuron HoF Models Included in the Training Set

To assess NOBLE’s ability to perform well on different neuron models, we trained it on a VIP neuron using the same architecture as in the PVALB case with 1.8M trainable parameters. The model was optimized for 450 epochs with the Adam optimizer with learning rate 0.004 and the ReduceLROnPlateau scheduler with factor 0.8 and patience 4, minimizing the relative L4 error. For the embeddings, the neural operator in NOBLE takes the stacked embeddings of the normalized model features  $I_{thr}$  and  $s_{thr}$  associated with  $\text{HoF}_\ell$  (with  $K = 1$  frequency) and  $I$  (with  $K = 11$  different frequencies).

The trained model achieves a relative L2 error of 2.5% on voltage traces and relative L2 errors of 9.0% for spikecount, 20% for AP1\_width, 10% for mean\_AP\_amplitude, and 0.99% for steady\_state\_voltage.

These results are comparable to those obtained for PVALB, indicating that NOBLE, with the same latent embedding space, also performs well when trained on different neuron types.

### D.4 Neighborhood Considered in Interpolation Experiments

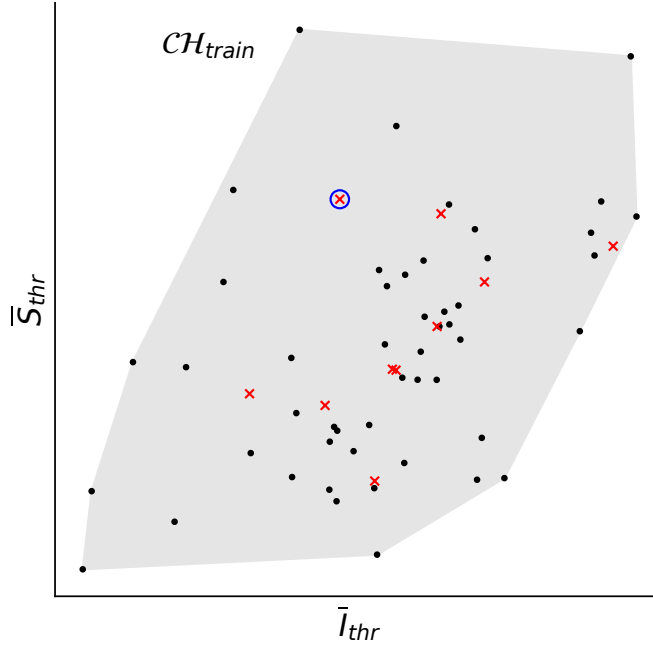


Figure 11: Latent representations in normalized  $(I_{thr}, s_{thr})$ -space of  $\{\text{HoF}^{train}\}$  (black dots) and  $\{\text{HoF}^{test}\}$  (red crosses) models. The latter lie in the convex hull  $\mathcal{CH}_{train}$  of the  $\{\text{HoF}^{train}\}$  models. In the interpolation experiment of Section 4.3, we construct a small neighborhood around a given  $\text{HoF}_k^{test}$  that defines a region of latent space not encountered during training, and sample 50 unseen models from this neighborhood. The boundary of this neighborhood is shown (blue circle) for an example  $\text{HoF}_k^{test}$  model.

## D.5 Auxiliary Ensemble Prediction Figures

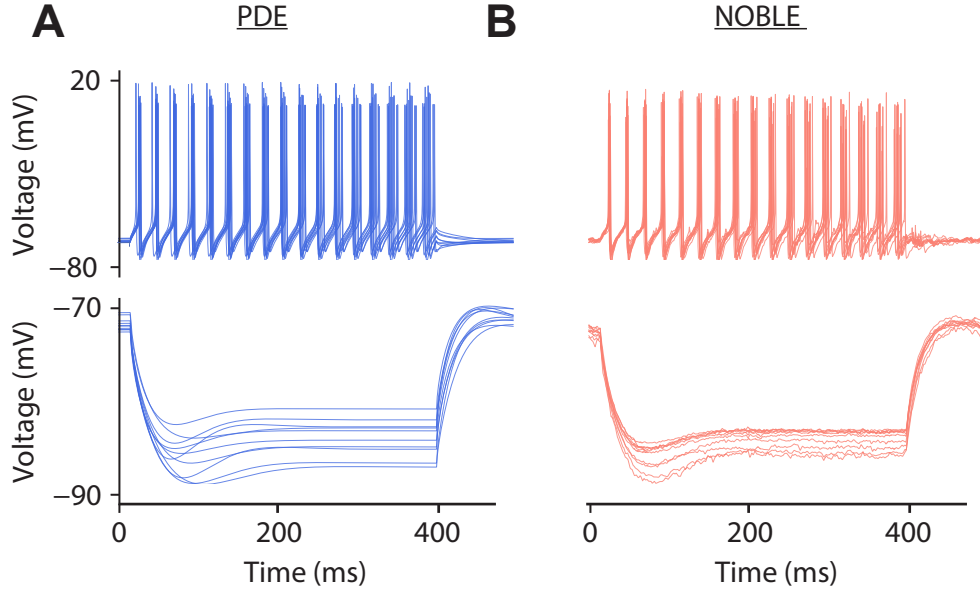


Figure 12: Distribution of somatic voltage traces across HoF models  $\{\text{HoF}^{test}\}$  for current injections of  $0.1\text{nA}$  (top row) and  $-0.11\text{nA}$  (bottom row). **A)** Ground truth voltage responses from HoF simulations, **B)** NOBLE predictions.

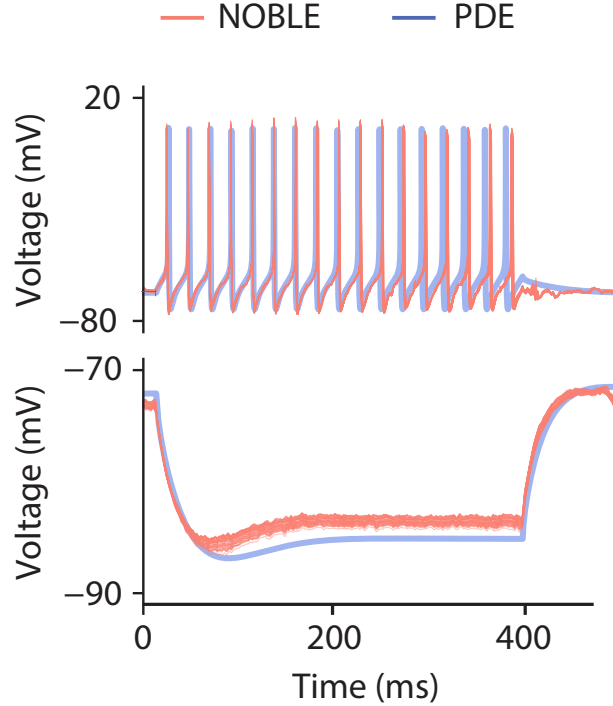


Figure 13: Distribution of somatic voltage traces across 50 synthetic HoF models sampled from a small circle centered on a HoF model in  $\{\text{HoF}^{test}\}$  not experienced during training, as shown in Figure 11. Results are shown for current injections of  $0.1\text{nA}$  (top) and  $-0.11\text{nA}$  (bottom). The ground truth voltage response from the HoF simulation not experienced during training is shown in blue, and the 50 synthetic NOBLE predictions in orange.

## D.6 On the HoF Models Availability Requirement

Training NOBLE on 75,600 samples required approximately 4 days on a 64GB NVIDIA Tesla P100 GPU (300 epochs). Once trained, for a given input current, NOBLE can synthesize arbitrarily many voltage responses of synthetic HoF models almost instantaneously by interpolating within  $\mathcal{CH}_{train}$  (Figure 10). Thus, NOBLE amortizes the high upfront cost of bio-realistic neuron model generation and enables scalable response synthesis at negligible inference cost. To investigate how much model diversity is needed during training for effective amortization, we varied the number of HoF models used to construct the training set  $\{\text{HoF}^{train}\}$  while keeping the total number of samples fixed. Results are summarized in Table 2 in terms of the relative L2 error of predicted voltage traces and the four key electrophysiological features when evaluated on the  $\{\text{HoF}^{test}\}$  models.

Table 2: Predictive performance of NOBLE on voltage traces and the four key electrophysiological features using the relative L2 error metric when the training set  $\{\text{HoF}^{train}\}$  is constructed by including varying numbers of models. Results are evaluated on  $\{\text{HoF}^{test}\}$ .

#HoFs included	Voltage	Steady state voltage	Spikecount	AP1 width	Mean AP amplitude
50	11.7%	2.0%	<b>9.2%</b>	<b>350%</b>	<b>14%</b>
40	10.9%	<b>1.8%</b>	19%	920%	17%
30	10.9%	1.9%	20%	3004%	20%
20	<b>10.6%</b>	1.9%	45%	1698%	20%

The relative L2 error on voltage traces is largely insensitive to HoF diversity, whereas spike-related features, particularly spikecount and AP1\_width, degrade substantially as diversity decreases. As discussed earlier, AP1\_width is particularly sensitive to small misalignments in spike peak and after-hyperpolarization minima, which can shift the half-voltage reference and lead to large relative errors even when the underlying traces are close. This makes AP1\_width less reliable for direct comparison than other features. Overall, these findings indicate that while NOBLE amortizes the cost of HoF generation, effective bio-realistic synthesis still requires sufficient model diversity to learn robust representations of neural dynamics. Depending on the purpose of the study and the solver–experiment error gap, the number of HoF models required for training can be adjusted accordingly. Moreover, for any given electrophysiological feature of particular interest, further fine-tuning can be used to refine predictions and improve generalization, as we discuss next.

## D.7 Additional Information on Fine-Tuning

Suppose the objective is to capture overall neural dynamics while placing particular emphasis on one specific electrophysiological feature. In this setting, the loss function can be designed to prioritize the feature of interest. Let  $F$  denote a feature computed on the ground-truth signal and  $\hat{F}$  the corresponding feature computed from NOBLE’s output. A feature-specific loss can then be defined as  $\mathcal{L}_F = \|F - \hat{F}\|$ , which directly penalizes deviations in the feature of interest.

To illustrate this, we fine-tune the previously trained NOBLE to enhance sag\_amplitude predictive performance. This feature is particularly relevant as it reflects the presence of the hyperpolarization-activated cation channel (Ih). In human cortex, the expression of Ih varies with cortical depth, making sag\_amplitude a useful marker. Although the broader role of Ih in shaping neuronal and network properties is not yet fully understood, it is thought to regulate neural excitability and coincidence detection.

We start from a pretrained NOBLE model and further optimize the weights of the neural operator using the feature-specific loss. Relying solely on a feature-specific loss, such as  $\mathcal{L}_{sag}$ , risks causing NOBLE to overfit to sag\_amplitude, improving that metric while degrading performance on other features and overall voltage trace fidelity. To mitigate this issue, one possible strategy is to introduce an anchor loss, as proposed in PINO [58], which penalizes deviations from the pretrained operator during fine-tuning. Combining the anchor loss with  $\mathcal{L}_{sag}$  could constrain optimization so that gains on a single feature do not come at the expense of overall signal fidelity, thereby encouraging balanced gains across all electrophysiological features.

Another approach, which we adopt, is to define a composite loss that encourages further accuracy on the voltage traces while prioritizing the feature of interest, `sag_amplitude`:

$$\mathcal{L}(\lambda) = \mathcal{L}_{\text{data}} + \lambda \mathcal{L}_{\text{sag}}.$$

To evaluate this approach, we construct a smaller dataset of 19,600 stimulus waveforms with negative non-zero amplitudes, since this regime elicits non-firing responses for which the `sag_amplitude` can be reliably computed. We then fine-tune the pretrained NOBLE model by minimizing the relative L4 error for  $\mathcal{L}_{\text{data}}$  and the relative L2 error for  $\mathcal{L}_{\text{sag}}$ , using the Adam optimizer with learning rate of 0.0005, and the ReduceLROnPlateau scheduler with factor 0.4 and patience 12. The results, reported in Table 3, summarize the relative L2 error of predicted voltage traces and `sag_amplitude` on the test set of  $\{\text{HoF}^{\text{train}}\}$ .

Table 3: Predictive performance of NOBLE fine-tuned on `sag_amplitude`. Metrics are reported as relative L2 errors on voltage traces and on `sag_amplitude`, with the training set  $\{\text{HoF}^{\text{train}}\}$  constrained to samples with negative non-zero amplitudes. Results are evaluated on the test set of  $\{\text{HoF}^{\text{train}}\}$ . Here,  $\lambda$  denotes the weighting factor of the feature-specific loss in the composite loss.

$\lambda$	Before optimization		Epoch 100		Epoch 200		Epoch 300	
	Voltage	Sag amplitude	Voltage	Sag amplitude	Voltage	Sag amplitude	Voltage	Sag amplitude
0	0.14%	69.6%	0.064%	22.2%	<b>0.041%</b>	20.3%	0.043%	19.2%
25	0.14%	69.6%	<b>0.055%</b>	<b>13.2%</b>	0.047%	<b>12.2%</b>	<b>0.035%</b>	<b>9.6%</b>

Even without an additional `sag_amplitude` loss, fine-tuning on the restricted non-firing regime of the dataset improves both trace prediction and `sag_amplitude` performance, since in this setting every sample is subthreshold and the feature can be computed consistently.

Including the feature-specific loss provides a further improvement of approximately 10% on `sag_amplitude`, while maintaining overall signal fidelity. Note that both settings converged after 300 epochs, ensuring that the comparison is fair.

These results demonstrate that prioritizing a feature through the loss function can yield targeted improvements without sacrificing global accuracy.

## E Comparison of Sample Generation Time

The trained NOBLE generates predictions significantly faster than the reference numerical solver from [45].

We record the time necessary to generate 10,000 predictions on a workstation equipped with a single NVIDIA RTX 4090 GPU (24GB VRAM), an AMD Ryzen 9 7900X CPU, and 64GB of system RAM. The numerical solver only generates a single prediction at a time and takes roughly 36,200 seconds to generate 10,000 predictions. When doing one inference at a time with NOBLE (i.e. batch size of 1), we generate 10,000 predictions in 157 seconds, i.e. a speedup of approximately 230× compared to the solver.

In addition, NOBLE can easily be accelerated on a single GPU by generating multiple predictions at the same time. In particular, with a batch size of 1000, NOBLE generates 10,000 predictions in 8.59 seconds, i.e. a speedup of approximately 4200× compared to the solver.

## F Scope and Future Directions

NOBLE successfully captures the dynamics across HoF models but is currently limited to single-neuron settings. A natural next step is to extend NOBLE to multi-neuron configurations with time-varying stimuli, enabling applications such as neuron classification and predicting multi-neuron dynamics. The embedding space used in our experiments is low-dimensional, constructed from two interpretable features derived from a biological neuron model’s F-I curve. Extending this to a learnable, higher-dimensional continuous embedding space represents a promising direction for future work. Beyond this, NOBLE could also integrate additional modalities such as gene expression or morphology to build a unified latent space linking molecular, electrophysiological, and structural characteristics, as discussed in the main text.

Although NOBLE demonstrates strong performance in modeling nonlinear neuronal dynamics, our study makes a few deliberate scope choices. These do not represent inherent limitations of the framework but rather natural starting points, each of which can be extended with minimal or no modifications.

- **Input currents:** We restricted our attention to square-pulse DC step currents as these are widely used in electrophysiological experiments and represent a common protocol for characterizing neuron behavior. However, the NOBLE framework is not specific to these types of input currents: time-varying inputs can be incorporated directly by including examples during training. In such cases, the amplitude embedding can be removed or adapted (e.g., embedding a function of the amplitude, such as a moving-average modulation or the maximum amplitude).
- **Choice of operator learning architecture:** We used the FNO primarily for its computational efficiency and strong generalizability. Moreover, since FNOs use the FFT and thus require inputs and outputs on equidistant grids, they align naturally with our data, where both simulations and experimental recordings are sampled at constant timesteps. For non-uniformly spaced data, alternative neural operators such as geometry-informed neural operators (GINOs) [64, 59] could be used within the same framework.
- **Training cost vs. efficiency:** Training NOBLE on 75,600 samples for 300 epochs took approximately four days on a 64GB NVIDIA Tesla P100 GPU, which is small compared to the  $\sim 600,000$  CPU hours required to generate the HoF models via evolutionary optimization. However, once trained, NOBLE enables fast inference and the instantaneous generation of infinitely many bio-realistic voltage traces through latent space interpolation, capabilities not possible with the original PDE models.
- **Neuron populations considered:** We focused on inhibitory neurons (PVALB and VIP), which show strong heterogeneity in morphology, gene expression, and electrophysiology, making them a stringent test for generalization. However, NOBLE is not restricted to inhibitory neurons and can naturally extend to excitatory types. It can also be applied to larger populations or multiple neurons within a family by expanding the latent embedding space with additional electrophysiological features that capture intracellular variability.



## NeurIPS Paper Checklist

### 1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [\[Yes\]](#)

Justification: The abstract and introduction clearly introduce the contributions made in the paper, and references the figures with the experimental results supporting the claims made.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

### 2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [\[Yes\]](#)

Justification: Limitations of the proposed approach are discussed in the conclusion and Appendix F.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

### 3. Theory assumptions and proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: The paper does not provide any new theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

#### 4. Experimental result reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: All the details of the framework used are provided in the text. The training and architecture hyperparameters are provided in the text. We also properly credited the NeuralOperator library, the numerical solver used, and the eFEL library used for computing electrophysiological features. The PyTorch codes used for our implementation of NOBLE and the numerical experiments are made available at [github.com/neuraloperator/noble](https://github.com/neuraloperator/noble).

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

## 5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The PyTorch codes used for our implementation of NOBLE and the numerical experiments are made available at [github.com/neuraloperator/noble](https://github.com/neuraloperator/noble).

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

## 6. Experimental setting/details

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: Section 4.1 provides all the training details and architecture hyperparameters needed to understand and replicate the results. The PyTorch codes used for our implementation of NOBLE and the numerical experiments are also made available at [github.com/neuraloperator/noble](https://github.com/neuraloperator/noble).

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

## 7. Experiment statistical significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: Error bars are included in figures for appropriate numerical experiments, and explained in the corresponding caption.

Guidelines:

- The answer NA means that the paper does not include experiments.

- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

#### 8. Experiments compute resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: The computing resources used are mentioned in Section 4.1, together with a comparison of running times between several approaches in ??.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

#### 9. Code of ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: [Yes]

Justification: We have reviewed the NeurIPS Code of Ethics and verified that we are compliant with the code of conduct. We also made sure to preserve anonymity during the review process.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

#### 10. Broader impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: We address the societal impact of NOBLE towards tackling fundamental questions in neuroscience and potential applications.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

## 11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: The paper poses no risks of misuse and does not necessitate requiring any usage guidelines or restrictions to access the model or implementing safety filters.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

## 12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We have properly credited and cited the paper for the NeuralOperator library, the papers for the numerical solver used, and the appropriate links to the eFEL library used for computing electrophysiological features.

Guidelines:

- The answer NA means that the paper does not use existing assets.

- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, [paperswithcode.com/datasets](https://paperswithcode.com/datasets) has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

### 13. New assets

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: The paper does not release new assets.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

### 14. Crowdsourcing and research with human subjects

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

### 15. Institutional review board (IRB) approvals or equivalent for research with human subjects

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

**16. Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.