Rapid, interpretable, data-driven models of neural dynamics using Recurrent Mechanistic Models

Thiago B. Burghi*[1] Maria Ivanova[2] Ekaterina Morozova[2] Huaxinyu Wang[1]
Eve Marder[2] Timothy O'Leary[1]

October 2024

Abstract

Obtaining predictive models of a neural system is notoriously challenging. Detailed models suffer from excess model complexity and are difficult to fit efficiently. Simplified models must negotiate a tradeoff between tractability, predictive power and ease of interpretation. We present a new modelling paradigm for estimating predictive mechanistic-like models of neurons and small circuits that navigates these issues using methods from systems theory. The key insight is that membrane currents can be modelled using two scalable system components optimized for learning: linear state space models, and nonlinear artificial neural networks (ANNs). Combining these components, we construct two types of membrane currents: lumped currents, which are more flexible, and data-driven conductance-based currents, which are more interpretable. The resulting class of models — which we call Recurrent Mechanistic Models (RMMs) — can be trained in a matter of seconds to minutes on intracellular recordings during an electrophysiology experiment, representing a step change in performance over previous approaches. As a proof-of-principle, we use RMMs to learn the dynamics of two groups of neurons, and their synaptic connections, in the Stomatogastric Ganglion (STG), a well-known central pattern generator. We show that RMMs are efficiently trained using teacher forcing and multiple-shooting. Due to their reliability, efficiency and interpretability, RMMs enable qualitatively new kinds of experiments using predictive models in closed-loop neurophysiology and online estimation of neural properties in living preparations.

A major obstacle to progress in neuroscience is the difficulty of reconciling models with data. This difficulty is particularly acute in neural circuits that have non-trivial intrinsic dynamics, where the interplay of membrane currents over many timescales dictate circuit-level properties. Biophysically detailed models suffer from excessive complexity and are notoriously difficult to fit to data [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]. On the other hand, purely data-driven models may fit data very well, but their general purpose structure limits mechanistic insights [11, 12, 13, 14, 15, 16, 17, 18], and they can give spurious predictions outside of the strict domain of the data they are trained with. At the heart of these challenges is a paradox: the basic physics of neurons and networks is very well understood, yet the models we derive from first principles cannot leverage this knowledge to make good data-driven predictions. On top of this paradox, most existing methods share the common drawback of being costly to constrain in terms of data, computational resources, and time.

In this work we present a new, efficient data-driven modelling approach that navigates the middle ground between mechanistic detail and black-box models from machine learning. Our ultimate goal is to reliably estimate mechanistic neuronal behaviour during time frame of an experiment. This goal is both pragmatic and scientific: obtaining a predictive model during the lifetime of an experiment opens the door to real-time

[1]Department of Engineering, University of Cambridge, UK. * Corresponding author. E-mail: tbb29@cam.ac.uk
[2]Department of Biology, Brandeis University, Waltham, MA.

estimation and closed-loop control of a living neural circuit. Secondly, this goal provides a strong test of our modelling assumptions, and of the efficiency and robustness of model training.

It is important to recognise existing approaches that can successfully fit experimental data and predict neural dynamics in certain settings, and contrast these with our approach and requirements. Existing approaches comprise two broad classes. The first assumes relatively simple neural dynamics and uses phenomenological models to predict spiking, to remarkable effect in some cases [1, 12, 11, 15]. However, this approach cannot account for rich intrinsic dynamics seen in circuits such as central pattern generators. Secondly, phenomenological models do not account for mechanistic effects of membrane conductances, limiting interpretability and preventing their use in closed loop, predictive control experiments with dynamic clamp and neuromodulators. The second class of approaches attempts to constrain large, detailed models to data. This requires extremely precise measurements, "clean" preparations, detailed knowledge about a given neuron's ion channel types, and large computational resources. Importantly, such approaches are not yet practical to use during the lifetime of an experiment where neuronal dynamics are uncertain [19, 2, 3, 4, 5, 6, 7, 8, 10]. Furthermore, it is not clear that these models capture 'true' parameters as opposed to plausible parameter subspaces of mis-specified models [20], which raises questions about their parsimony, interpretability and predictive power.

To address these difficulties, we introduce Recurrent Mechanistic Models (RMMs), a novel modelling framework that leverages the predictive power of machine learning while retaining biophysical, mechanistic intuition. RMMs, which can be framed as class of recurrent neural networks (RNNs) with biophysically motivated constraints, use a combination of linear time-invariant (LTI) state space models and artificial neural networks (ANNs) to capture the dynamics of intrinsic and synaptic ionic currents. Exploiting this model structure, we show that RMMs can be primed for different purposes in which different levels of interpretability are required. RMMs can predict the trajectories of ionic conductances when a detailed model is warranted, effectively acting as a quickly-trainable surrogate for conductance-based models. In more general cases, including most experimental settings where specific ionic currents are unknown or uncertain, we show how the level of detail in RMMs can be systematically reduced so as to capture lower-resolution aspects of neuronal dynamics such as dynamic current-voltage (IV) relations at specific timescales.

A key advantage of RMMs is that they can be reliably trained in a matter of seconds to minutes in a consumer-grade desktop computer, enabling their use during an electrophysiology experiment. Rapid training is achieved with teacher forcing [21], which allows RMMs to be trained in a non-recurrent fashion that exploits massive parallelization. We show how to improve on teacher forcing results using a recurrent method, multiple shooting (for a technical exposition, see [22]), that can be tuned so as to trade off speed of training for prediction accuracy, while avoiding the well-known problem of exploding gradients [23].

To provide a realistic experimental test of this methodology, we estimate the dynamics of coupled neurons in the Stomatogastric Ganglion, a well-studied Central Pattern Generator (CPG) in the crab *cancer borealis*. This circuit provides an ideal testbed for our application, as the dynamics of its neurons are complex and its function is dictated by the action of neuromodulators, which shape membrane conductance properties. We find that RMMs provide excellent predictive power during the lifetime of an experiment, and can recover features of underlying membrane conductances, enabling mechanistic interpretation.

Results

Our results introduce Recurrent Mechanistic Models (RMMs), and demonstrate that they can reliably infer and interpret complex neuronal dynamics, while also being fast enough to generate predictions during electrophysiological experiments.

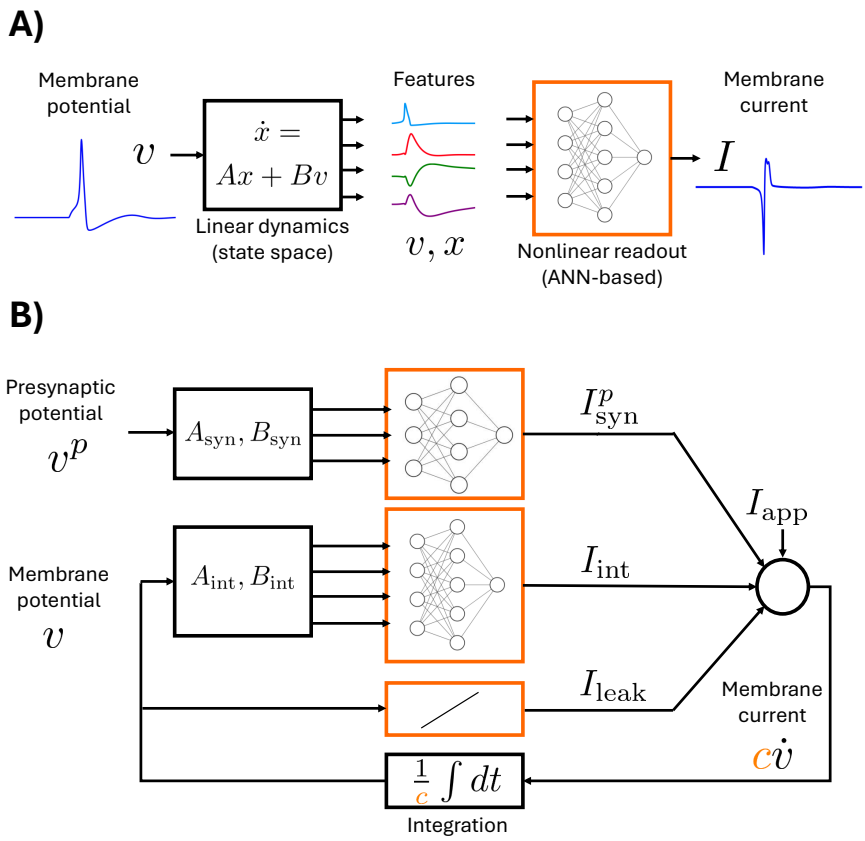

Figure 1: The basic structure of Recurrent Mechanistic Models. **A)** A state space system in series with an artificial neural network is used to transform membrane voltages into membrane currents. State-space systems are systematically designed so that features are dynamically rich (see Methods). Modelling flexibility can be traded off for interpretability by imposing architectural constraints on the ANN. **B)** A neuronal model is obtained by integrating applied, intrinsic and synaptic membrane currents. The blocks depicted in orange are those containing the main trainable parameters of the model, which also include the membrane capacitance, c. While state space models can in principle be learned, biophysical intuition can be used to construct them and keep them fixed, leading to faster training times (see Methods). Dots indicate time derivatives; in practice, the model is implemented in discrete-time according to Equations (1)-(6).

Recurrent Mechanistic Models

RMMs are set apart from other types of neuron models due to the use of linear time-invariant (LTI) state space systems in conjunction with artificial neural networks (ANNs) to model the ionic currents flowing through the neuronal membrane. The basic idea is illustrated in Figure 1 A: a state space system produces a set of dynamical features of the membrane voltage signal, and the features are recombined by a readout function, constructed with ANNs, to yield the membrane current. The state-space systems are systematically designed to have orthogonal convolution kernels, resulting in dynamically rich features for the ANNs to recombine (see Methods). As the results will show, modelling flexibility can be traded off for interpretability by imposing architectural constraints on the readout function. A schematic of a RMM for a neuron with one synaptic input is shown in Figure 1 B. It is straightforward to recognise that the schematic is equivalent to the circuit diagram of a single-compartment conductance-based model [24], with intrinsic, synaptic, leak and applied currents summed in parallel and integrated to compute membrane voltage.

We now briefly describe the basic mathematical structure of RMMs; omitted details can be found in Methods. Since RMMs are learned from sampled data, we use discrete-time notation. The discretized derivative voltage derivative is denoted by

$$\Delta v_t := \frac{v_{t+1} - v_t}{\delta} \tag{1}$$

where δ is the sampling period used when collecting data in an experiment. The dynamics of a single neuron in our modelling paradigm is given by

$$c\Delta v_t = -I_{\text{int},t} - \sum_p I^p_{\text{syn},t} - I_{\text{leak},t} + I_{\text{app},t} \tag{2}$$

where, at time index t, the current $I_{\text{int},t}$ denotes the total intrinsic (ionic) current flowing through the neuron's membrane, $I^p_{\text{syn},t}$ denotes the total synaptic current due to the pth presynaptic neuron, $I_{\text{leak},t}$ denotes the leak current, and $I_{\text{app},t}$ denotes the applied current.

The total intrinsic current $I_{\text{int},t}$ models the sum of all ionic currents flowing through the neuron's membrane. In general, the intrinsic current is given by

$$I_{\text{int},t} = \sum_{i=1}^{m} \varphi_i(x_t, v_t; \theta_i) \tag{3a}$$

where each φ_i is an ANN-based nonlinear readout function, parameterized by θ_i, and x_t is the vector-valued state of a linear time-invariant system given by

$$x_{t+1} = Ax_t + Bv_t \tag{3b}$$

with A and B constant matrices of the appropriate dimension.

Each learnable ANN-based readout φ_i in (3a) models an individual component of the total intrinsic current, whose level of mechanistic detail is determined by the parametric form of φ_i. To model electrical current components whose mechanisms are unknown, uncertain, or too complex to deduce, we employ **lumped currents**. Their models do not single out individual ionic types, and are constructed with a multi-layer perceptron (MLP, i.e, a black-box feed-forward ANN) according to

$$\varphi(x_t, v_t\,; \theta) = \text{mlp}(h(x_t, v_t)\,; \theta) \tag{4}$$

where $\text{mlp}(\,\cdot\,; \theta)$ is a single-output MLP, and $h(x_t, v_t)$ is a fixed layer used to normalize, select, and possibly threshold the inputs v_t and x_t. To model electrical current components whose ionic mechanisms are better understood, we employ **data-driven conductances**. Their models make use of mechanistic priors, such as the reversal potential, or the type of gating dynamics of a current (activation, inactivation, or both). A typical data-driven conductance-based current is given by

$$\varphi(x_t, v_t\,; \theta) = g_t(v_t - E) \tag{5a}$$

with

$$g_t = \bar{g}\,\text{mlp}_+(h_+(x_t, v_t)\,; \theta)^\intercal \text{mlp}_-(h_-(x_t, v_t)\,; \theta) \tag{5b}$$

where E and $\bar{g}$ are the reversal potential and the maximal conductance (about which we might have prior knowledge). Here, the MLPs and their fixed input layers are constrained to model activation $(+)$ and inactivation $(-)$ gating variables, and the inner product of the MLP outputs is interpreted as a normalized conductance. For more details on (4) and (5), see *constructing ANNs* in Methods.

The justification for this particular model structure is based on practical as well as theoretical considerations. From a pragmatic viewpoint, using a single state space system to model the recurrent part of the intrinsic current dynamics greatly facilitates the choice of initial model parameters; in addition, we have empirically found a systematic choice for A and B such that state space parameters do not need to be learned to achieve high predictive performance (see *constructing linear systems* in Methods). State space models also have a smaller number of parameters when compared to equivalent LTI representations such as temporal convolutions. Finally, the structure of RMMs allows exploiting massive paralellization during training, since the readouts φ_i of every intrinsic current component share the same "reservoir" of internal states x_t (see *training algorithms* in Methods).

From the theoretical point of view, we are supported by the fact that any reasonable model of an ionic current (or conductance) must have the so-called Fading Memory property [25]. This means that the mapping from voltage to current (or conductance) can be approximated arbitrarily well by a so-called Wiener model: a linear-time-invariant (LTI) system with a nonlinear static readout function [25, 26]. As pointed out in [13], fading memory can be empirically observed in the input-output behaviour of ionic currents under voltage-clamp. Intuitively, fading memory holds in a population of ionic currents if clamping the membrane voltage to a particular value results in a unique steady-state current, regardless of the value of the voltage prior to clamping. It is the very reason for which neuronal excitability can be studied using steady-state IV curves.

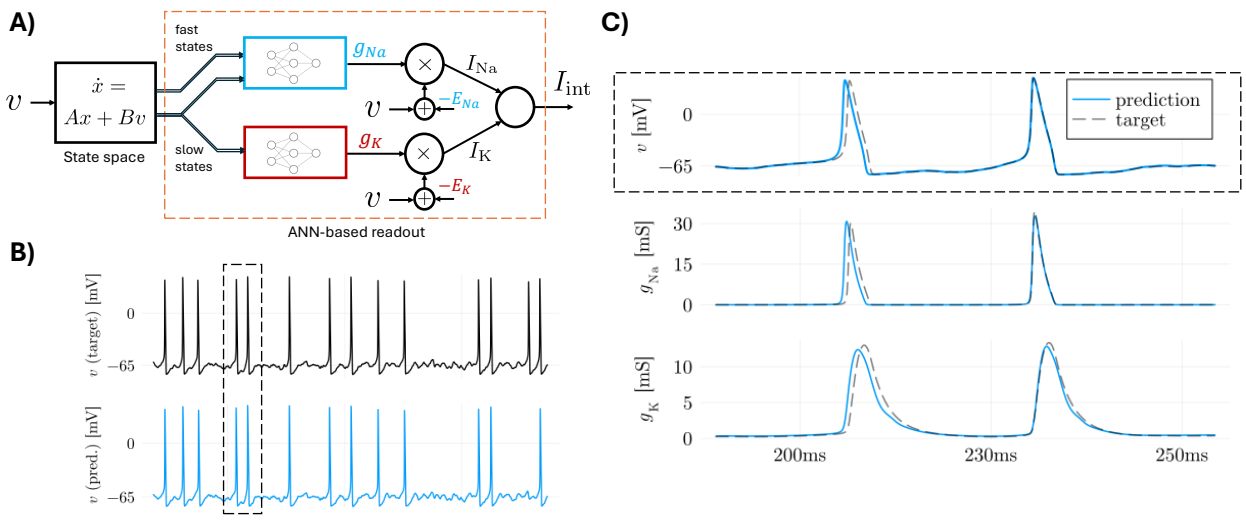

Figure 2: An appropriately constrained RMM can faithfully mimic the classical Hodgkin-Huxley conductance-based model [27]. **A)** A six-dimensional state space system is designed whose state components evolve in fast and slow timescales roughly corresponding to those of the spiking dynamics of the original HH model. The ANN-based readout (cf. Figure 1) is constrained to promote interpretability: sodium and potassium conductances are obtained as outputs of MLPs, and the corresponding ionic currents are obtained by multiplying those data-driven conductances by a difference in potential ($v - E_{\text{ion}}$), with E_{ion} a prior reversal potential. To ensure the model learns the correct currents, fixed layers selecting fast and slow states are used. **B)** Target voltage trace (top) resulting from simulating the original HH model given a noisy applied current, and corresponding RMM prediction (bottom) on held-out test data. **C)** (top) zoom of the dashed region in (B) shows accuracy of spike prediction. (middle, bottom) The RMM accurately recovers conductance trajectories during spiking. For modelling and training details, see Methods.

The total synaptic current $I^p_{\text{syn},t}$ caused by a presynaptic neuron with membrane potential v^p_t is modelled analogously to the intrinsic current, the only difference being the use of v^p_t instead of v_t in (3b). Finally, the leak current is given by a linear-in-the-parameters model

$$I_{\text{leak},t} = s_+(\theta_{\text{leak},1})v_t + \theta_{\text{leak},2} \tag{6}$$

where the softplus function $s_+(\cdot) = \log(1 + \exp(\cdot))$ is used to keep the leak conductance positive throughout training.

RMMs can mimic traditional conductance-based models

The first question we ask is whether RMMs have the modelling capacity to faithfully mimic the dynamical behaviour of conductance-based models, which are widely used in computational studies. Using data-driven conductances (5), we designed an RMM with currents modelling the sodium and the potassium currents of the Hodgkin-Huxley neuron [27]. The structure of the total intrinsic current is illustrated in Figure 2 A, and the parameters are described in detail in *Hodgkin-Huxley RMM* in Methods. In constructing this model, we exploit *timescale separation*: the state space system is defined with three "fast" states evolving roughly in the timescales of sodium activation (0.1, 0.2, and 0.5 milliseconds), and three "slow" states corresponding to sodium inactivation and potassium activation (2.0, 4.0, and 7.0 milliseconds). While the fast state components are fed to the sodium activation MLP, the slow state components are fed to both the sodium inactivation and the potassium inactivation MLPs (no other priors are imposed on the inputs of the MLPs). Figure 2 B and (C) show that RMMs can indeed stand in for conductance-based models: after training the RMM with the HH model's voltage and conductance simulated time series, not only can the RMM faithfully predict the HH model's voltage, but it can also predict the temporal evolution of the sodium and potassium conductances.

Experimental validation of the RMM approach

While data-driven conductances (Eq. (5)) can make RMMs fully interpretable, in most experimental scenarios the precise ion channel makeup of a given neuron is not readily available, providing poor biophysical priors to work with. In addition, quantitatively predicting the response of neuronal circuits *in vitro* and *in vivo* often requires using multi-compartment models whose parameters are difficult if not impossible to estimate based on a single voltage signal. This section shows how single-compartment RMMs can address these problems through judicious use of lumped currents (Eq. (4)).

As a proof-of-principle that RMMs can be used to estimate the behaviour of neurons with complex morphology and uncertain biophysics *solely from applied current and voltage measurements*, we performed experiments to predict neuronal dynamics in a well-studied system: the Stomatogastric Ganglion (STG) of the crab *cancer borealis*. We use the experimental setup illustrated in Figure 3 A: after pharmacologically blocking glutamatergic synapses of the STG, a simple feed-forward network is obtained where the Lateral Pyloric (LP) neuron is synaptically driven by two Pyloric Dilator (PD) neurons via cholinergic synapses. Both PD neurons are strongly coupled to the Anterior Burster (AB) neuron, a bursting pacemaker. As an approximation, one can think of this as a feedforward circuit where a strongly coupled subcircuit consisting of the two PD neurons and the AB neuron drives the single LP neuron. The only data used for training our models in this section consists of the applied currents and voltage signals from the LP neuron and one of the PD neurons. The 'true' PD-AB dynamics in particular are thus extremely complex, owing to the morphology of the cells as well as the fact that the PD measurement signal accounts for the combined dynamics of three coupled neurons.

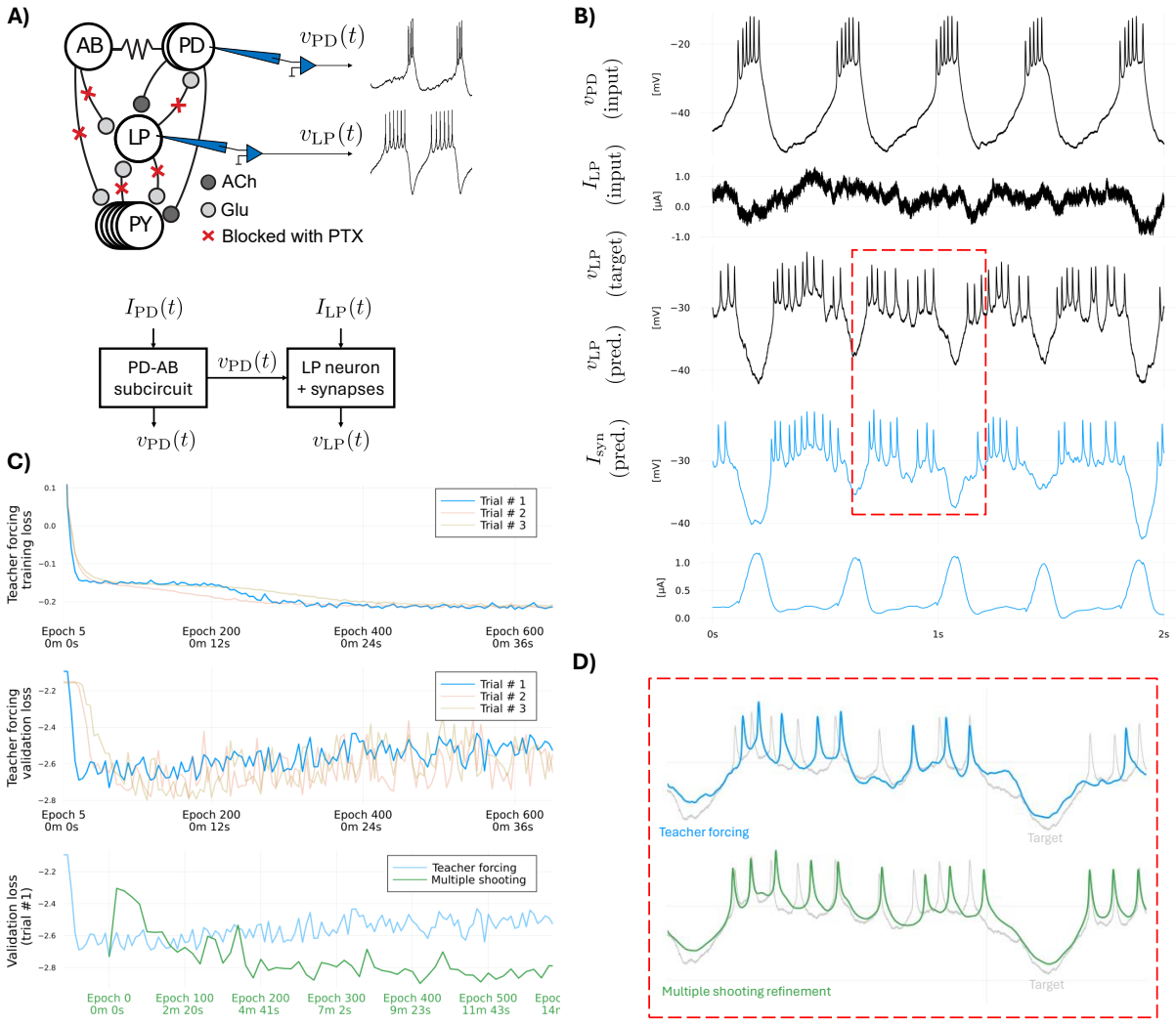

Figure 3: Estimating a RMM model of a neuron with complex spatiotemporal dynamics. **A)** Experimental setup used to generate data for fitting an RMM: the STG circuit is bathed in PTX, which blocks Glutamate synapses but not Acetylcholine synapses; intracellular recordings are taken of *one* of the PD neurons and the LP neuron; the resulting system is a feedforward interconnection with the measured presynaptic PD neuron voltage driving the LP neuron. **B)** Example voltage traces showing held-out data (three top panels) and the predicted voltage and synaptic currents from an RMM trained with teacher forcing (see Methods). **C)** Evolution of the loss functions during training for three different trial datasets; each dataset consists of around 2 seconds of data recorded with a sampling rate of 0.1 ms. The highlighted Trial 1 yields the model used an example in (B). Top: loss function value during training with teacher forcing, evaluated with held-out data; middle: validation loss function value using models trained with teacher forcing; bottom: multiple shooting improves the performance of the best model obtained with teacher forcing. **D)** Example traces demonstrating the improvement obtained with multiple shooting. For a description of teacher forcing and multiple shooting, as well as the loss function used for model validation, see Methods.

Rapid prediction of total intrinsic and synaptic currents

To test the predictive power of RMMs constructed exclusively with lumped currents, we trained models to predict the responses of the LP neuron in Figure 3 A. The LP neuron RMM is constructed according

to the diagram in Figure 1 B, with lumped ANN readouts given by (4) for both intrinsic and synaptic currents (for details, see *LP neuron RMM* in Methods). To generate a dynamically rich, time-varying input to LP, we injected noisy currents given by an Ornstein-Uhlenbeck process, while monitoring both LP and PD membrane potentials. After fitting an RMM we analysed the predictive performance of the model on held-out test data (Figure 3 B) and found good quantitative agreement between predicted LP membrane voltage (orange traces) and measured LP membrane voltage. The model also enabled us to infer the total synaptic current I_{syn}, which varied as expected with PD bursts to generate intermittent pauses in LP spiking.

Over multiple fitting runs with validation on held-out data, we found that good steady-state prediction error is achieved independently of initial conditions and datasets (Figure 3C). In Figure 3 C (top), we plotted losses for three different trials, with datasets collected from the same preparation at different times. Teacher forcing losses decrease and stabilize rapidly, under a minute. These results thus demonstrate a crucial feature of RMMs: the fact that reasonably good prediction accuracy, under large system uncertainty, can be achieved efficiently, with prediction error asymptoting in several tens of seconds on modest hardware (see Methods).

The learning performance of RMMs is a result of using an extremely efficent training method based on teacher forcing (Methods). As with all machine learning methods, the decrease of training loss during teacher forcing does not guarantee a decrease in predictive power, as shown in Figure 3 (C, middle), where validation losses are plotted for models obtained in different epochs of training (see Methods for a description of training vs. validation losses). Thus, while teacher forcing quickly trains the model, it also leads to overfitting, making early stopping or strong regularisation essential. In our results, ANNs are regularized with an L2 penalty.

If necessary, predictive accuracy can be enhanced further using a generalisation of teacher forcing called multiple-shooting (see Methods). The result of this additional training on predictive accuracy is shown in Figure 3 (C, bottom), while Figure 3 D shows how spiking traces from the zoomed in portion of A are improved with this extra step of training. Starting from a model trained with teacher forcing, multiple shooting improves model performance at the cost of longer training times, up to several minutes on this dataset. The tradeoff between predictive accuracy and training time can therefore be dictated by the requirements of the experiment, and can yield very good results in minutes.

Predicting and interpreting endogenous bursting

We next used the RMM framework to learn an interpretable single-compartment model of the bursting PD-AB subcircuit, often called the pacemaker kernel, in the same experimental setup of Figure 3 A. Mechanistically, bursting emerges from a complex combination of sodium, potassium, and calcium ion channels which are unevenly distributed in the system. Despite the complexity, modelling the pacemaker kernel with a single burst-excitable compartment is made possible by the strong electrical coupling between AB and PD neurons, which allows us to assume that the slow component of the bursting wave is roughly the same across those neuron's membranes. However, quickly fitting mechanistic models of the pacemaker kernel remains difficult. In particular, the sodium channels driving intra-burst spikes are located in a region of the PD neuron's axon far from the recording site in the soma, as illustrated in Figure 4 A. As a result, fast intra-burst spikes in the recordings are heavily filtered versions of the PD neuron's spikes (see [28, 29] for details).

To deal with this challenging scenario, we used a combination of lumped and mechanistic data-driven currents to learn the dynamics of the AB-PD pacemaker, as depicted in Figure 4 B, and asked whether the resulting RMM would yield the appropriate intrinsic mechanism for generating endogenous bursting. The RMM is constructed with three mechanistic currents representing the aggregate dynamics of calcium currents (I_{Ca}), potassium (including calcium-activated) currents (I_{K}), and hyperpolarization-induced currents (I_{H}). To model the intra-burst spiking dynamics while constraining the RMM to a single compartment, a lumped current (I_{axon}) using fast states of the state space model is used (see *PD neuron RMM* in Methods).

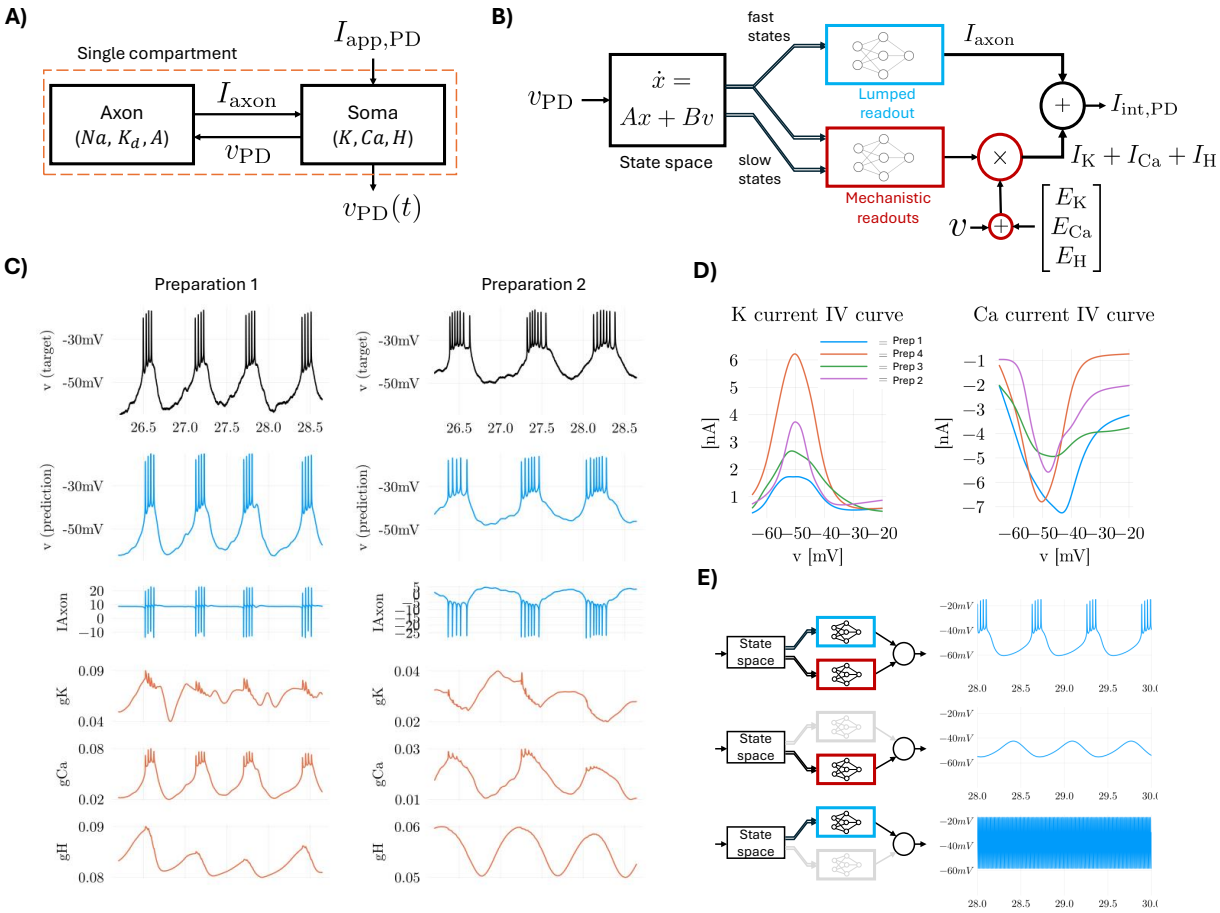

Figure 4: Training a RMM with lumped and mechanistic branches to learn endogeneous bursting rhythms. **A)** a diagram representing the dynamics of the AB-PD pacemaker, simplified here as a single PD cell [29]. Fitting traditional mechanistic models to the system is challenging: a single-compartment conductance-based model cannot quantitatively predict the membrane voltage at the soma (where recordings are made), since fast sodium and potassium currents are located in a distant part of the axon. **B)** To quickly obtain a predictive single-compartment model, we employ an RMM with a mix of lumped and mechanistic currents. The lumped current, which does not distinguish between ion channels and does not require mechanistic structure, is used to predict the fast axonal current. To model the calcium and potassium channels of the soma, we employ three data-driven mechanistic currents to capture the aggregate dynamics of potassium, calcium, and hyperpolarization-induced currents. **C)** Validation of two RMM models on held-out membrane voltage data (black traces) from two different PD neuron preparations. The models, which were trained on current-clamp data (see *PD neuron RMM* in Methods) predict the membrane voltage well. In addition, the model allows interpreting its internal variables as individual currents and conductances. **D)** Steady-state IV curves of four RMMs fit to different PD neuron preparations show that RMMs are able to learn the basic mechanisms of bursting based on calcium and potassium currents. **E)** Removing the fast lumped current from the RMM uncovers the slow wave dynamics underlying burst excitability; conversely, removing the slow mechanistic currents results in fast tonic spiking due to the lumped axonal current. All models shown in this figure were trained with teacher forcing (see methods) over the course of 1000 epochs, corresponding to about four minutes of training.

Figure 4 C, top, shows example voltage traces of two PD neuron recordings. It can be seen that even though the recordings are done in the same type of cell, the features of the recorded voltage traces are reasonably different, due to the intrinsic variability between individual preparations. The voltage traces shown were held out from training and used to validate the pacemaker RMMs, whose predictions are shown in blue in Figure 4 C. The RMMs, which were trained in around four minutes (see *PD neuron RMM* in Methods) predict the target dataset well. In addition to predicting the target voltages, the structure of the RMM allows us recover the lumped membrane current, representing the fast axonal current, as well as the data-driven conductances associated to the aggregate calcium, potassium, and H currents. With minimal a priori information, the RMM is able to learn the functional role of the three types of currents: the aggregate potassium current activates following a burst, hyperpolarizing the membrane; the aggregate calcium current drives the upstroke of the slow bursting wave; and the aggregate H current activates following hyperpolarization, helping to initiate the burst.

Using the same training hyperparameters, we validated models on four different preparations of the PD neuron. In figure Figure 4 D we plot the steady-state IV curves of the aggregate potassium and calcium currents, obtained from the two RMMs of figure Figure 4 C, as well as two other validated models (not shown). The steady-state IV curves all overlap in the range usually associated to the activation of calcium currents in PD neurons of the STG. It should be mentioned that since the models were trained on current-clamp data, it is expected that the steady-stateo IV curves shown in the figure are not as quantitatively accurate as the current-clamp voltage predictions of Figure 4 C (blue trace). However, they do point to the fact that RMMs can learn the basic mechanisms of neuronal excitability without the need for voltage-clamping the cells.

To further investigate whether RMM predictions at the level of individual conductances and currents are dynamically meaningful, we probed the behaviour of the RMM after removing the lumped axonal fast current from the model. Figure Figure 4 E shows the result (the model is that of preparation 1). In the top panel, it can be seen that with both lumped and mechanistic currents, and no external excitation, the RMM generates an endogenous bursting rhythm. Removing the fast lumped current results in a slow wave, shown in the middle panel, proving the point that the mechanistic currents alone are sufficient to drive the endogenous rhythm. Finally, removing the mechanistic currents and their slow dynamics results in a very fast spiking rhythm, shown in the bottom panel, caused by the fast lumped current.

Discussion

Unlike some areas of physical sciences, where quantitative predictions can determine the validity of a theory, model predictions in neuroscience often have a more equivocal character. A poor fit to data can imply incorrect modelling assumptions, low quality data, inappropriate fitting methods, or any of these issues simultaneously. This ambiguity persists even when modelling reduced preparations consisting of a small number of neurons, and where tight control of experimental conditions and accurate measurements are possible. These challenges have preoccupied neurophysiologists for decades, and our limited success in handling them has hindered our ability to apply vast accumulated knowledge of low-level physiology to systems-level questions.

The approach we have introduced here does not solve all these problems, but it does permit theoretically grounded predictive models to be inferred during the course of an experiment, which drastically enhances our ability to solidify theories of neural circuit function, and to use these theories to practical ends. Our goal was to find a model that captures what we know about biophysics without assuming more than can be measured, and to find a means of constraining the model to produce good predictive accuracy on a timescale of minutes.

Achieving this goal forced us to sacrifice a modelling formalism that is often placed among the most successful theoretical developments in all of neuroscience: the detailed Hodgkin-Huxley form of conductance based modelling [27]. This may seem tantamount to abandoning any semblance of 'mechanism' in our modelling approach. Most mechanistic models are conductance-based and strive to account for ion channel gating dynamics and even the spatial distribution of channels in complex morphologies. Such models are fundamental for understanding how low level biophysics shapes the dynamic properties of neurons [30, 31, 32, 29] and in some cases constitute predictive models that fit data very well [10]. However, in general, the empirical success of this modelling paradigm has been modest, especially when applied to more typical neural circuits that possess many tens of ionic conductance types, complex morphologies and physiology that hinders precise experimental measurements. Constraining the parameters of detailed models under such conditions is notoriously difficult [33].

In addition to the practical difficulty of parameter estimation in detailed models, there is a deeper, scientific problem with attempts to capture detail. Regardless of their detail, models likely omit unknown structural and biophysical properties that are impractical to measure [34]. In other words, the strict form of a detailed conductance based model that supposedly describes a given type of neuron is almost certainly wrong, and the parts of it that are right might constrain the model's dynamics in unwanted ways, and even introduce artefacts [35]. This simultaneously impacts predictive power and interpretability: inferred parameter values of a misspecified model, however precise, may have no sensible physical meaning.

Our approach acknowledges our ignorance of the extant pool of conductances and synaptic inputs in a preparation, their spatial distribution, phosphorylation state, splice variants and the remaining plethora structural variation that is known to exist but cannot be measured. Instead, we only model what we can measure, representing this information in terms of known biophysics, such as early and late IV relationships, and, where possible, inward and outward current components. Ultimately these are the features of the underlying biophysics that dictate a neuron's dynamics, and in this sense, they are the most parsimonious, mechanistic inference we can be confident of in a typical experimental setting.

Even though the experimental system we chose is well characterised, with a relatively simple connectivity, unobserved inputs and dynamics in the STG abound in the data. Although we blocked fast inhibitory transmission with PTX, there are numerous modulatory inputs coming from the front-end ganglia that are still present in the preparation that affect several intrinsic currents [36] in a manner that depends on the wider state of the preparation [37]. Gap junctions between the membranes of different cells introduce further uncertainty [38]. PD is tightly electrically coupled to AB, but it is also loosely coupled to LPG, which in turn is connected to VD, and several other cells. LP is connected to IC and multiple PY cells. Furthermore, the strength of gap junctions might be susceptible to modulation [39]. These various inputs and perturbations are not accounted for in the model because we are unable to measure them directly, it is therefore far from trivial to be able to predict dynamics in this circuit with any degree of accuracy.

Methods

We begin with a detailed exposition of the technical aspects of the modelling and training methodology; abstract details are made concrete in the sections describing the HH, LP and PD neuron RMMs. Results in this paper were obtained with code written in the Julia language, using the Flux.jl package to implement gradient descent and automatic differentiation routines [40].

Constructing ANNs

The MLPs used to model lumped readouts in (4) are given by the chained composition of mappings

$$\mathrm{mlp}(\cdot\,;\,\theta) := W_{L+1} \bigcirc_{\ell=1}^{L} \sigma_L(W_\ell \cdot + b_\ell) \tag{7}$$

where L is the number of MLP layers, W_ℓ are weight matrices, b_ℓ are bias vectors, and the vector-valued function $\sigma(\cdot)$ is an elementwise continuous sigmoidal mapping with the appropriate dimensions. In lumped readouts, all activation functions are given by $\sigma_\ell = \tanh$, and the weight matrices are initialized following the Nguyen-Widrow method [41], with biases spread out evenly from -1 to $+1$. The MLPs used to model mechanistic readouts in (5) are also given by (7), but their weights are constrained so as to implement monotonically increasing or decreasing mappings, depending on whether the MLP is used to model the activation (mlp_+) or the inactivation (mlp_-) of a current. All the weights of activation MLPs are given by $W_i = (\tilde{W}_i)_+$, where $(\cdot)_+ = \log(1 + \exp(\cdot))$ is the softplus function and $\tilde{W}_i$ are the weight matrices actually being trained. The weights of inactivation MLPs are defined similarly, except for W_L, which is given by $W_L = -(\tilde{W}_L)_+$. In mechanistic readouts, σ_1 to σ_{L-1} are given by tanh, while σ_L is given by the logistic function (this ensures conductances are always positive); the trainable $\tilde{W}$ are initialized following the Glorot (uniform) method [42] with a gain of 0.5; weights are initialized evenly from -1 to $+1$.

In both lumped and mechanistic readouts, the fixed layers $h_i(x_t, v_t)$ are used for three key functions: to ensure that the MLP inputs are inside the interval $[-1, 1]$ (this is assumed by the weight initialization heuristics above); to pass only parts of the state vector x_t (and possibly omit v_t) as inputs to the MLP; and to bias the inputs of the MLP so that the corresponding current activates mostly when the membrane voltage is within a certain range. The latter two functions implement biophysical priors related to the speed and activation range of membrane currents. Mathematically, the fixed layer of each readout (we omit the subscript i) is given by the affine mapping

$$h(v_t, x_t) = \begin{bmatrix} C_v v_t + \beta_v \\ C_x x_t + \beta_x \end{bmatrix}$$

with the C and β fixed variables determined before training (the first row above is dropped if v_t is not an input to the MLP). The matrix C_x has a single unit element in each row; it can be used to drop parts of x_t associated to fast or slow dynamics (in code, this is implemented by selecting the appropriate rows of x_t). The C and β variables are computed using the training data as follows: if no priors on the membrane current activation range are imposed, then given a continuous voltage recording, denoted by $\bar{v}_t$, a sequence of "forced" states $\bar{x}_t$ is obtained by simulating (3b) with the data $\bar{v}_t$, and discarding transients. Then C_x and β_x are found so that $\min_t C_x \bar{x}_t + \beta_x = -1$ and $\max_t C_x \bar{x}_t + \beta_x = 1$, and analogously for C_v and β_v. If a mechanistic prior exists dictating that a given membrane current activates mostly on a voltage range given by $[\upsilon_{\min}, \upsilon_{\max}]$, then a bias on the values of C and β is introduced. A sequence of "forced" states $\tilde{x}_t$ is now obtained by simulating (3b) with $\tilde{v}_t = \mathrm{sat}_{\upsilon_{\min},\upsilon_{\max}}(\bar{v}_t)$, where $\mathrm{sat}_{\upsilon_{\min},\upsilon_{\max}}(\cdot)$ saturates $\bar{v}_t$ at the endpoints of the prior voltage range. Then C_x and β_x are found as before, and C_v and β_v are found such that $C_v \upsilon_{\min} + \beta_v = -1$ and $C_v \upsilon_{\max} + \beta_v = 1$. Notice that since the weights of $\sigma_1 = \tanh$ are initialized with the assumption that its inputs remain within -1 and 1, the procedure above ensures that whenever v_t is outside of the prior activation range, the inputs to the MLP will fall on the saturated part of tanh, implementing the desired prior on the activation range. Notice also that for this procedure to succeed, it is crucial that the state space matrices A and B are initialized with reasonable values, since the state space system must be simulated prior to learning. The next section describes how this can be achieved.

In mechanistic data-driven currents, it is possible to gain further modelling capacity modifying (5) as

$$\varphi(x_t, v_t\,;\theta) = \bar{g}\left(\mathrm{mlp}_+(h_+(x_t, v_t)\,;\theta) \odot \mathrm{mlp}_-(h_-(x_t, v_t)\,;\theta)\right)^{\intercal} (\mathbf{1}v_t - E) \tag{8}$$

where $\odot$ denotes the elementwise (Hadamard) product, $\mathbf{1}$ represents the all-ones vector, and now E is a vector of Nernst potentials. This can be used when the Nernst potential of a current is uncertain, as is often the case with Calcium currents; the Nernst vector E acts as a further set of weights of the data-driven mechanistic current (it has the same dimension as the output dimension of the activation and inactivation MLPs).

Constructing linear systems

The state space matrices A and B in the total intrinsic current dynamics (3) are constructed systematically, in two steps. First, we choose a sequence of n positive time constants $\tau_1, \tau_2, \ldots, \tau_n$ believed to play a role in the collective dynamics of the intrinsic currents; second, we use the sequence $\{\tau_j\}$ to construct a discrete-time state space system with *orthogonal impulse responses* [43] (convolution kernels) whose output provides a rich set of signals for ANNs to recombine. The matrices for synaptic currents are constructed analogously.

The simplest way to choose the sequence $\{\tau_j\}$ is to exploit prior knowledge of the time constant functions defining the gating kinetics of an existing conductance-based model. For a gating variable w, this is usually given by an equation of the type

$$\tau_w(v)\dot{w} = -w + w_\infty(v)$$

with $\tau_w(v)$ the time constant function of w. The fastest and slowest time constants are chosen as the minimum and the maximum time constants of the model given the observed voltage range $V_{\text{obs}} = [v_{\min}, v_{\max}]$ of the data. Mathematically,

$$\tau_1 = \min_{w,v \in V_{\text{obs}}} \tau_w(v), \quad \tau_n = \max_{w,v \in V_{\text{obs}}} \tau_w(v),$$

Intermediate time constants are chosen incrementally between τ_1 and τ_n so as to roughly represent the range of values observed in $\tau_w(v)$. In practice, because the time constant functions of ion channels often overlap, the sequence $\{\tau_j\}$ will be similar for different models. As an illustration, in Fig. S1 we plot the time constant functions of the LP neuron model of [44] which were used to guide the choice of $\{\tau_j\}$ for the LP and PD data-driven models in this paper.

Once a sequence $\{\tau_j\}$ is given, we compute a sequence of discrete-time system poles (eigenvalues of A) by applying the well-known stability-preserving transformation

$$\lambda_j = \exp(-\delta/\tau_j) \tag{9}$$

where δ is the sampling period of the data and λ_j are the discrete-time poles (see [45] for sampled-data system theory). Those poles are in turn used to construct A and B so as to realize a linear system with orthogonal impulse responses (also known as convolution kernels). To construct a system with no timescale separation between states, this is accomplished by defining $A = A_n$ and $B = B_n$, where A_n and B_n are given by the iterative procedure

$$\begin{aligned}
&A_1 = \lambda_1, \quad B_1 = \sqrt{1-\lambda_1^2}, \quad C_1 = B_1, \quad D_1 = -\lambda_1 \\
&A_i = \begin{bmatrix} A_{i-1} & 0 \\ \sqrt{1-\lambda_i^2}C_{i-1} & \lambda_i \end{bmatrix}, \quad B_i = \begin{bmatrix} B_{i-1} \\ \sqrt{1-\lambda_i^2}D_{i-1} \end{bmatrix} \\
&C_i = \begin{bmatrix} -\lambda_i C_{i-1} & \sqrt{1-\lambda_i^2} \end{bmatrix}, \quad D_i = -\lambda_i D_{i-1}
\end{aligned} \tag{10}$$

for $i = 1, \ldots, n$ (see [43, Chapter 2] for details). The above procedure results in a system whose state components (the elements of x_t) follow distinctly different trajectories given sudden changes in the input; we have found empirically that this results in significantly better predictive models when compared to simpler

choices of A and B. To illustrate this point, in Figure Figure S2 we contrast the "spike responses" of a diagonal state space system with $A = \text{diag}(\lambda_1, \ldots, \lambda_n)$, $B = ((1-\lambda_1), \ldots, (1-\lambda_n))^\intercal$ (a set of first order filters) to the responses of system with identical eigenvalues λ_j, but constructed according to (10); the responses of the latter overlap less in time, providing ANNs with a richer set of input signals. Notice that for slow state components, a spike is well approximated by the impulse function, and hence spike responses of slow states will be near orthogonal (modulo a constant dependent on the mean of the spiking signal).

Whenever timescale separation between components of the state vector x_t is desired, the procedure is slightly modified. To construct A and B in those cases, we first partition the time constant sequence $\{\tau_j\}_{j=1,\ldots,n}$ into non-intersecting subsequences $\{\tau_j\}_{j=1,\ldots,m_1}$, ..., $\{\tau_j\}_{j=m_s,\ldots,n}$ where $s > 0$ is the number of desired timescale partitions. Then, we define A as a block-diagonal matrix whose diagonal (matrix) elements are each constructed according to (10) using the corresponding subsequences of discrete-time poles $\{\lambda_j\}_{j=1,\ldots,m_1}$, etc. This modification ensures that the state vector partitions $[x_1, \ldots, x_{m_1}]$, etc., do not share dynamical modes in the same timescales.

Training RMMs

In this section we describe how rapid training can be achieved with two related training methods that avoid well-known difficulties commonly encountered when training mechanistic models [46]. We consider a dataset given by $\{I_{\text{app},t}, \bar{v}_t\}_{t\in[0,T]}$ consisting of measured applied current $I_{\text{app},t}$ and voltage targets $\bar{v}_t$. The main bottleneck is that of so-called *exploding gradients* [23, 47], which occur when one considers the supervised learning paradigm using the cost function

$$C(\theta, T) = \frac{1}{T}\sum_{t=1}^{T} \|\bar{v}_t - v_t\|^2 + \rho\|\theta\|^2 \tag{11}$$

where $\bar{v}_t$ is the measured voltage, v_t is the θ-dependent voltage estimate from (2), and ρ is a regularization constant. When dealing with spiking signals, a naive attempt to minimize (11) with gradient-based methods leads to a blow-up in $\nabla_\theta C(\theta, T)$ as the dataset horizon T increases. Gradient explosion occurs due to the high sensitivity of spiking models to changes in their parameters [46], which in turn is due to the existence of bifurcations in neural dynamics [30]. The two methods outlined below adapt the techniques of teacher forcing [21] and multiple-shooting [22] to the context of RMMs, eliminating exploding gradients altogether.

Teacher forcing with backpropagation

Rather than using the cost (11), we can formulate a supervised learning problem in terms of

$$\mathcal{C}(\theta, T) = \frac{1}{T}\sum_{t=0}^{T-1} \|I_{\text{app},t} - \hat{I}_{\text{app},t}(\theta)\|^2 + \rho\|\theta\|^2 \tag{12}$$

where $I_{\text{app},t}$ is the measured applied current, $\hat{I}_{\text{app},t}$ is a θ-dependent applied current estimate, and ρ a regularization constant. The idea, which has first been applied to conductance-based models in [19] (see [48] for a theoretical analysis), is to use voltage measurements $\bar{v}_t$ to estimate the applied current $I_{\text{app},t}$, rather than the other way around. By manipulating (2), we find that the current estimate is given by

$$\hat{I}_{\text{app},t} = -c\Delta\bar{v}_{t+1} + \hat{I}_{\text{int},t} + \sum_{p\in\mathcal{P}} \hat{I}^p_{\text{syn},t} + \hat{I}_{\text{leak},t} \tag{13}$$

where

$$\hat{I}_{\text{int},t} = \sum_i \varphi_i(\hat{x}_t, \bar{v}_t; \theta_{\text{int}}) \tag{14a}$$

$$\hat{x}_{t+1} = A_{\text{int}}\hat{x}_t + B_{\text{int}}\bar{v}_t \tag{14b}$$

is used to compute the internal current estimate, with $\hat{I}^p_{\text{syn},t}$ and $\hat{I}_{\text{leak},t}$ being computed analogously using the measured $\bar{v}^p_t$ and $\bar{v}_t$, respectively. In machine learning terms, the formulation (12) is an instance of *teacher forcing* [21], since training is carried out by replacing v_t by the measured (teacher) signal $\bar{v}_t$ in the model dynamics. The formulation has roots in the *equation-error* method of system identification [49].

Teacher forcing lends itself exceptionally well for the purposes of training RMMs, and in case the state space matrices in (14b) remain fixed (non-trainable), massively parallelized computations in GPUs can be exploited to their fullest potential. To see why, assume A and B are fixed a priori, and let

$$\bar{v}_{0:T} = \begin{bmatrix} \bar{v}_0 & \bar{v}_1 & \dots & \bar{v}_T \end{bmatrix}$$

denote a row vector containing all voltage measurements. The teacher-forced state estimates (14b), which need to be computed only once as they do not involve model parameters, can also be put in matrix form as

$$\hat{x}_{0:T} = \begin{bmatrix} \hat{x}_0 & \hat{x}_1 & \dots & \hat{x}_T \end{bmatrix}$$

with $\hat{x}_0$ a "warmed-up" initial condition[1]. By precomputing $\hat{x}_{0:T}$, minimizing (12) in θ reduces to the problem of training a feed-forward ANN with inputs and targets given by large matrices, enabling massive parallelization. Indeed, (12) and (13) can be rewritten as

$$\mathcal{C}(\theta, T) = \frac{1}{T}\|\Delta\bar{v}_{0:T-1} - \Delta\hat{v}_{0:T-1}(\theta)\|^2 + \rho\|\theta\|^2 \tag{15}$$

with $\Delta\bar{v}_{0:T-1}$ a row vector of target ANN outputs, and $\Delta\hat{v}_{0:T-1}$ being the output of a feed-forward ANN with matrix inputs $\bar{v}_{0:T-1}$, $\hat{x}_{0:T-1}$, $\hat{z}^p_{0:T-1}$, and $I_{\text{app},0:T-1}$. Notice that in practice, the capacitance c is lumped into the output layers of the φ sub-networks, and recovered as the inverse of the ANN input weight corresponding to I_{app}.

Computing the gradient of (15) can be done efficiently with backpropagation, which for moderate number of layers in the model's MLPs causes no exploding nor vanishing gradients[2]. Notice that this "static" version of teacher forcing relies on two facts: first, the fact that bounds on physiological time constants are well-known, which allow fixing A and B in a reasonable manner; and second, the fact that A can be trivially constrained to be stable (Hurwitz), a property that ensures that the states $\hat{x}_t$ and $\hat{s}^p_t$ remain bounded for any possible measurements $\bar{v}_t$ and $\bar{v}^p_t$.

Multiple shooting with backpropagation through time

Teacher forcing comes with the downside that since the model is trained to predict the applied current, it might not (and in general there is not guarantee that it will) predict voltage satisfactorily. Although in our results we have shown that teacher forcing yields very good results on experimental data, better models can

[1] Warming up consists in choosing $\hat{x}_0$ by simulating the state space model with a reserved part of the voltage data: suppose the initial segment of the dataset is given by $\{\bar{v}_t\}_{t\in[-N,-1]}$; then this simply means computing (14b) for $t = -N, \dots, -1$.

[2] In practice, additional tricks used to train feed-forward neural networks with gradient descent can be employed to improve generalization of the trained model. Two of those tricks are *mini-batching*, which divides the dataset into multiple batches so as to perform more frequent descent steps, and *shuffling*, which randomly permutes the columns of input and output matrices at the beginning of each training epoch. The joint application of such tricks has come to be known as *stochastic gradient descent.*

be found by exploiting methods that acknowledge the the recurrence of RMMs. This can be achieved flexibly by the method of multiple shooting [22], which we adapt to RMMs in this section, and which generalizes the teacher forcing scheme presented earlier.

In multiple shooting, a dataset of length T is divided into N smaller intervals of length T_s, with N the largest integer such that $NT_s \leq T$. To construct a training cost function, the model being fit is simulated N times, with the initial conditions of each simulation defined independently. Rigorously, for each $n = 0, 1, \ldots, N-1$ and each $t = 0, 1, \ldots, T_s - 1$, we define the inputs

$$I^{(n)}_{\text{app},t} = I_{\text{app},t+nT_s}$$

and simulate the system to obtain N state vectors, which we denote by

$$x^{(n)}_t := (v^{(n)}_t \,;\, x^{(n)}_t \,;\, s^{(1,n)}_t \,;\, s^{(2,n)}_t \,;\, \ldots)$$

and which are initialized according to

$$x^{(n)}_0 = \xi^{(n)} \tag{16}$$

where the $\xi^{(n)}$ are arbitrary initial conditions. In practice, and especially if the A and B matrices are reasonably chosen, good initial conditions are given by the measurements and the teacher-forced states, i.e.

$$\xi^{(n)} = (\bar{v}_{nT_s} \,;\, \hat{x}_{nT_s} \,;\, \hat{s}^1_{nT_s} \,;\, \hat{s}^2_{nT_s} \,;\, \ldots) \tag{17}$$

with $\hat{x}_{nT_s}$ and $\hat{s}^p_{nT_s}$ computed from (14).

The key feature of multiple shooting is that the $\xi^{(n)}$ are trained alongside θ so as to minimize the discrepancy between the state at the end of the nth simulation, $x^{(n)}_{T_s}$, and the initial condition of the $(n+1)^{\text{th}}$ simulation, $\xi^{(n+1)}$. This is accomplished by solving the optimization problem given by

$$\min_{\theta,\xi} \quad \sum_{n=0}^{N-1} C_n(\theta, \xi^{(n)}) + \rho \|x^{(n)}_{T_s} - \xi^{(n+1)}\|^2 + \rho\|\theta\|^2 \tag{18a}$$

where

$$C_n(\theta, \xi^{(n)}) = \frac{1}{T_s} \sum_{t=0}^{T_s-1} \|\bar{v}_{t+1+nT_s} - v^{(n)}_{t+1}\|^2 \tag{18b}$$

are the cost functions associated to each of the N simulations. The problem (18) is an unconstrained optimization problem on the parameters θ and initial conditions $\xi^{(n)}$; it can be solved efficiently by means of gradient descent. When training with multiple shooting, whether A and B are trainable or not, computing the gradient of the cost function in (18a) requires performing backpropagation through time over the T_s timesteps of each multiple-shooting window; in addition, to avoid exploding gradients one must keep the window small enough. The size of the shot size T_s is a hyperparameter of the problem; larger shot sizes may yield better predictions at the cost of slower learning.

It can be seen from (18) that when initial conditions are chosen according to (17), multiple shooting is a generalisation of teacher forcing. The latter is recovered by simply choosing $N = T$ and $T_s = 1$. By increasing T_s, the multiple shooting cost function becomes that used in backpropagation through time, which is recovered with $N = 1$ and $T_s = T$. Multiple shooting can also be significantly sped up by exploiting parallelization. Letting

$$V^{(n)}_t = \begin{bmatrix} v_{t+nT_s} & v_{t+2nT_s} & \ldots & v_{t+(N-1)T_s} \end{bmatrix}, \quad X^{(n)}_t = \begin{bmatrix} x_{t+nT_s} & x_{t+2nT_s} & \ldots & x_{t+(N-1)T_s} \end{bmatrix}$$

and similarly defining S_t and S^p_t, the model can be simulated in parallel on each of the N multiple shooting intervals by computing (2)-(6) blockwise, that is, replacing v_t by $V^{(n)}_t$, x_t by $X^{(n)}_t$, and so on. It follows

that in multiple shooting there is a natural tradeoff between the speed of training (enabled by large N and massive parallelization) and the quality of predictions, which tends to increase (up to a bound determined by exploding gradients) with longer intervals T_s (see [22], [47] for a formal analysis).

Validating RMMs

All models in this paper were validated on held-out training data as follows: a "snapshot" of models during training was saved so that each training run yielded around 100 snapshot models; after training, snapshot models were simulated using (2) and the validation applied current (this was done using parallel computing to speed up the validation process), and the best model was chosen as the result of the training procedure. This was done as a means to prevent overfitting, and to recover loss curves. Validation of all 100 snapshot models did not exceed two minutes for any of the results in this paper.

The best snapshot model was determined by comparing its predicted spike times (including intra-burst spikes) to the validation voltage spike times. This was done by computing a validation cost given by

$$\mathcal{V}(\theta, T) = \frac{1}{T}\sum_{t=0}^{T-1}(F(\bar{v}_t) - F(v_t))^2$$

where v_t is the predicted voltage trace, $\bar{v}_t$ is the target validation voltage trace, and $F(\cdot)$ is a nonlinear filter designed to extract a smoothed spike train out of the voltage traces. The nonlinear filter F applies, in succession: a bandpass 6th order Butterworth filter with cutoff frequencies at 1/50 and 1/2 kHz (which filters out most of the slow components of the voltage traces); a relu nonlinearity with threshold set at 2 mV (which transforms the filtered spike train into a sequence of approximate impulse functions); and a Gaussian smoother with standard deviation between 200 and 300 mV which smooths the approximate impulse train. Notice comparing smoothed spike trains is an established methodology to validate statistical integrate-and-fire models [12].

Hogkin-Huxley RMM

In Figure 2, to learn the HH model with a RMM, the Hodgkin-Huxley model with standard parameters [27, 30] was simulated with the forward-euler discretization scheme and simulation timestep of 0.01 ms. The simulated voltage was downsampled to $\delta = 0.05$ ms and corrupted by white Gaussian noise of standard deviation equal to 0.1 mV, resulting in the voltage target $\bar{v}_t$ used for training and validation; this was done to simulate realistic experimental conditions. Two training datasets of 25 seconds each were combined for training purposes: in the first dataset, the HH model is excited in current-clamp mode with the applied current given by the discretized Ornstein-Uhlenbeck (OU) process

$$\begin{aligned} I_{\text{app},t} &= z_t + I_0 \\ z_{t+1} &= -\bar{\alpha} z_t + \bar{\sigma} w_t \end{aligned} \tag{19}$$

with $I_0 = 0.0$, $\bar{\alpha} = 0.2$, and $\bar{\sigma} = 1$. In the second dataset, the same OU process is used with $\bar{\alpha} = 0.02$ and $\bar{\sigma} = 0.6$. To train the RMM so that it mimics the HH model as well as possible, we have added two components to the training cost functions above, corresponding to the ionic conductance trajectories:

$$\mathcal{C}_{HH}(\theta, T) = \frac{1}{T}\sum_{t=0}^{T-1}\|I_{\text{app},t} - \hat{I}_{\text{app},t}(\theta)\|^2 + \gamma \sum_{\text{ion}=\text{Na,K}}\|\bar{g}_{\text{ion},t} - \hat{g}_{\text{ion},t}(\theta_{\text{ion}})\|^2 + \rho\|\theta\|^2 \tag{20}$$

where $\bar{g}_{\text{ion},t}$ are the HH model's conductance trajectories, $\hat{g}_{\text{ion,t}}$ are teacher-forced versions of the conductance in (5b), and $\gamma > 0$ sets a tradeoff between voltage trace prediction and conductance prediction (we used

$\gamma = 10$). Solving the problem (20) provides a systematic means to initialize a conductance-based RMM so that it has the same behaviour as a traditional conductance-based model; further training the RMM is quick, while further training a CB model is generally not. Training was performed with $\rho = 0.01$, and the cost was minimized with the ADAM routine with step size 0.01 and momentum parameters $\beta_1 = 0.9$, $\beta_2 = 0.95$.

LP neuron RMM

To construct the LP neuron RMM, the intrinsic and synaptic currents were defined based on the sequence of continuous time constants $\tau_i^{\text{int}} = 0.05 \times 1.3^i$, $i = 0, 1, \ldots, 35$, and $\tau^{\text{syn}} = 0.05 \times 1.3^i, \quad i = 14, 15, \ldots, 29$, which were chosen pragmatically so as to ensure that all timescales usually found in LP model are represented; see Figure S1 for representative time constant functions found in an LP conductance-based model [44]. Using those time constants, state space systems are constructed according to the orthogonalisation procedure described in *constructing linear systems*. The model is constructed with a single lumped readout for the total intrinsic and total synaptic currents, that is

$$I_{\text{int},t} = \text{mlp}_{\text{int}}(h_{\text{int}}(x_t, v_t), \theta)$$
$$I_{\text{syn},t} = \text{mlp}_{\text{syn}}(h_{\text{syn}}(x_t^{\text{syn}}, v_t), \theta^{\text{syn}})$$

with mlp given by (7). The fixed affine layers h_{int} and h_{syn} are only used to normalize v_t and x_t so that $-1 < v_t, \|x_{i,t}\|_\infty, \|x_{i,t}^{\text{syn}}\|_\infty < 1$ for all t. The MLPs were chosen with $L_{\text{int}} = 3$ layers and $L_{\text{syn}} = 2$ layers, respectively, with ten units in any given layer.

In Figure 3, training and validation data for the LP neuron consisted in voltage $\bar{v}_t$ and applied current $I_{\text{app},t}$ obtained experimentally according to the *electrophysiology* section. The particular dataset used to train and validate the RMM used $I_{\text{app},t}$ given by the discretized OU process (19) with $\bar{\alpha} = 0.015$ and $\bar{\sigma} = 0.015$. To train the model, we have used teacher forcing as described above, with $\rho = 10^{-4}$ and minimized the cost function with the ADAM gradient descent routine with step size 0.003, $\beta_1 = \beta_2 = 0.9$, and a mini-batch size of 484000 datapoints. In Figure 3 we have used multiple shooting with $\rho = 10^{-4}$ and the ADAM gradient descent routine with step size 0.001 and momentum parameters $\beta_1 = \beta_2 = 0.9$. A shot size of $T_s = 30$ was used to improve on teacher forcing results while keeping training times in the order of minutes.

PD neuron RMM

To obtain the results of Figure 4, the state space system of the total intrinsic current (there are no synapses entering the PD model) was defined based on four subsequences of continuous time constants roughly matching four distinct time constant function ranges observed in the dynamics of STG neurons. Those ranges are represented in the four panels of Figure S1, where we have plotted the time constant functions of [44]. Using *slice* notation, the four $\{\tau_j\}_{j=1,\ldots,n}$ subsequences were chosen as $S_{\text{fast}} = \{0.2 : 0.2 : 1.6\}$, $S_{\text{slow1}} = \{2 : 2 : 16\}$, $S_{\text{slow2}} = \{20 : 10 : 90\}$, and $S_{\text{ultraslow}} = \{100 : 75 : 625\}$. The matrices A and B are constructed according to the procedure described above; the A matrix is block-diagonal with each block constructed according to Eq. (10) in *constructing linear systems*, using the discrete-time version of the time constants from each subsequence.

The PD neuron RMM contained four currents: a lumped current I_{axon}, given by (4), and three mechanistic data-driven currents I_{Ca}, I_{K}, and I_{H}, given by (8). The lumped current's MLP, which has $L = 3$ layers with 10 units in each layer, receives as inputs v_t as well as $(x_{1,t}, \ldots, x_{16,t})$ which correspond to the states evolving in the timescales of the S_{slow1} and S_{slow1} time constant subsequences; no biophysical biases are introduced during parameter initialization. Each of the activation and inactivation MLPs used in the mechanistic currents contained three layers with 15, 10 and 5 units in each layer, respectively. Inputs to mlp_+ and

mlp$_-$ of I_{K} were given by $(x_{9,t}, \ldots, x_{28,t})$ and $(x_{17,t}, \ldots, x_{28,t})$, respectively; inputs to mlp$_+$ and mlp$_-$ of I_{K} were given by $(x_{9,t}, \ldots, x_{24,t})$ and $(x_{17,t}, \ldots, x_{28,t})$, respectively; inputs to mlp$_-$ of I_{H} were given by $(x_{25,t}, \ldots, x_{32,t})$; the H current had no activation MLP. Maximal conductances were initialized at $\bar{g}_{\mathrm{K}} = 10.0$, $\bar{g}_{\mathrm{Ca}} = 1.0$, $\bar{g}_{\mathrm{H}} = 1.0$ and left fixed to let the learnable output weights of mlp$_+$ and mlp$_-$ determine the magnitude of the time-varying conductances. Nernst potentials of the mechanistic currents were initialized to $E_{\mathrm{K}} = -80$, $E_{\mathrm{K}} = 130 + \epsilon$, and $E_{\mathrm{H}} = -10 + \epsilon$, with ϵ a vector of zero-mean normally distributed random variables with standard deviation of 10 mV. Nernst potential vectors of the I_{H} and I_{Ca} currents were trained along with the capacitance and MLP layer parameters, while the Nernst potential of I_{K} was fixed. To bias the model so that lumped currents activated mostly around the intra-burst spike range of the bursting wave, and mechanistic currents activated mostly at the slow-wave range of that wave, a biased parameter initialization procedure was employed according to the *constructing ANNs* section, using $(\upsilon_{\min}, \upsilon_{\max}) = (-40, 0)$ mV for the lumped current and $(\upsilon_{\min}, \upsilon_{\max}) = (-80, -40)$ mV for the mechanistic currents.

In Figure Figure 4, training and validation data for the AB-PD subsystem consisted in voltage $\bar{v}_t$ and applied current $I_{\mathrm{app},t}$ obtained experimentally from a single PD neuron following the experimental procotol described in the *electrophysiology* section. The dataset used to train and validate the RMM consisted of a combination of seven different 30-second long trials where the PD neuron was excited with the discretized OU process (19); in four of the trials $\bar{\sigma} = 0.015$ was fixed and $\bar{\alpha} = 0.001, 0.0015, 0.001, 0.015$ was varied, and in three of the trials $\bar{\sigma} = 0.03$ was fixed and $\bar{\alpha} = 0.0015, 0.001, 0.015$ was varied. To train the model, we have posed the teacher forcing problem as described above with $\rho = 10^{-4}$, and minimized the cost function with the ADAM gradient descent routine with step size 0.003, momentum parameters of $\beta_1 = \beta_2 = 0.9$, and a mini-batch size of 484000 datapoints.

Electrophysiology

Animals

Adult male Jonah Crabs, Cancer borealis, were acquired from Commercial Lobster (Boston, MA) between August 2022 and July 2023. Animals were held in groups up to 5 in 75 Gallon tanks filled with artificial seawater at 11-13°C with a 12 hr light/dark cycle. The crabs were put on ice for at least 30 min before the dissection. Dissections were performed as described in Gutierrez, G. J., & Grashow, R. G. (2009). In brief, the crab stomach was removed, the stomatogastric nervous system (STNS) was dissected from the stomach and pinned down in a petri dish (10 ml) lined with Sylgard (The Dow Chemical Company). The STNS includes the stomatogastric ganglion (STG) containing the cells of interest, modulatory inputs descending into the STG from the esophageal ganglion and the two bilateral commissural ganglia, and the motor nerves coming from STG. The STNS was held at 4°C overnight in solution (saline) containing 440 mM NaCl, 11 mM KCl, 26 mM MgCl2, 13 mM CaCl2, 11.2 mM Trizma base, 5.1 mM maleic acid, pH 7.45, and used for the experiment on the next day.

Recordings

The STG was desheathed and intracellular recordings from somata of the lateral pyloric (LP) and pyloric dilator (PD) neurons were obtained. Sharp glass microelectrodes (10-20 MΩ) filled with 0.6 M K2SO4 and 20 mM KCl were used to record the intracellular activity. During the experiment the STNS-containing dish was pump-perfused with 11°C saline at a rate of 5ml/min. The temperature was controlled by Peltier device connected to the model CL-100 temperature controller (Warner Instruments). The cells were identified by their activity (monitored by extracellular electrodes) on the motor nerves: PD activity on the pyloric dilator nerve and LP activity on the lateral ventricular nerve. Extracellular recordings were done with

a stainless-steel electrode in a well (Vaseline and 10% mineral oil mixture) around the motor nerve of interest. An Axoclamp 900A amplifier (Molecular Devices, San Jose) was used to amplify intracellular signals. Extracellular signals were amplified by an A-M Systems 1700 extracellular amplifiers. Data were obtained with pClamp data acquisition software (Molecular Devices, San Jose, version 10.5), Digidata 1440A digitizer (Molecular Devices, San Jose) and Real-Time eXperiment Interface (RTXI) software version 2.2. Recordings were acquired at 10 kHz sampling frequency. After the cells were identified, the solution was switched to saline containing 10^{-5}M picrotoxin (PTX), Sigma-Aldrich, to block the inhibitory synaptic connections between cells. Recording started 30 min after PTX was applied. The membrane voltage was recorded (in PD and LP) and current was injected (in LP) with a single electrode in Discontinuous Current Clamp mode. A custom RTXI module was used for the Ornstein-Uhlenbeck noise current injection.

Hardware

Training times stated in this paper were obtained with a desktop computer set up with an AMD Ryzen 7 7800X3D 8-Core Processor, 64 Gigabites of RAM, and a NVIDIA GeForce RTX 4090 graphics card with 24 GB of VRAM.

Acknowledgements

The research leading to these results has been funded by the Kavli Foundation, whose support is greatly appreciated. The authors would like to thank Rodolphe Sepulchre, Fulvio Forni, Dhruva Raman, Alessio Franci and Kyra Schapiro for their insightful comments on this work.

References

[1] L. Paninski, J. Pillow, and E. Simoncelli, "Comparing integrate-and-fire models estimated using intracellular and extracellular data," *Neurocomputing*, vol. 65, pp. 379–385, June 2005.

[2] H. Abarbanel, D. Creveling, R. Farsian, and M. Kostuk, "Dynamical State and Parameter Estimation," *SIAM Journal on Applied Dynamical Systems*, vol. 8, pp. 1341–1381, Jan. 2009.

[3] C. D. Meliza, M. Kostuk, H. Huang, A. Nogaret, D. Margoliash, and H. D. I. Abarbanel, "Estimating parameters and predicting membrane voltages with conductance-based neuron models," *Biological Cybernetics*, vol. 108, pp. 495–516, Aug. 2014.

[4] J.-M. Lueckmann, P. J. Goncalves, G. Bassetto, K. Öcal, M. Nonnenmacher, and J. H. Macke, "Flexible statistical inference for mechanistic models of neural dynamics," in *31st Conference on Neural Information Processing Systems (NIPS 2017)*, (Long Beach, CA, USA), p. 18, 2017.

[5] K. Abu-Hassan, J. D. Taylor, P. G. Morris, E. Donati, Z. A. Bortolotto, G. Indiveri, J. F. R. Paton, and A. Nogaret, "Optimal solid state neurons," *Nature Communications*, vol. 10, p. 5309, Dec. 2019.

[6] L. M. Alonso and E. Marder, "Visualization of currents in neural models with similar behavior and different conductance densities," *eLife*, vol. 8, p. e42722, Jan. 2019.

[7] J. D. Taylor, S. Winnall, and A. Nogaret, "Estimation of neuron parameters from imperfect observations," *PLOS Computational Biology*, vol. 16, p. e1008053, July 2020.

[8] P. J. Gonçalves, J.-M. Lueckmann, M. Deistler, M. Nonnenmacher, K. Öcal, G. Bassetto, C. Chintaluri, W. F. Podlaski, S. A. Haddad, T. P. Vogels, D. S. Greenberg, and J. H. Macke, "Training deep neural density estimators to identify mechanistic models of neural dynamics," *eLife*, vol. 9, p. e56261, Sept. 2020.

[9] T. B. Burghi and R. Sepulchre, "Adaptive observers for biophysical neuronal circuits," *IEEE Transactions on Automatic Control*, vol. 69, no. 8, pp. 5020–5033, 2024.

[10] S. A. Wells, J. D. Taylor, P. G. Morris, and A. Nogaret, "Inferring the Dynamics of Ionic Currents from Recursive Piecewise Data Assimilation of Approximate Neuron Models," *PRX Life*, vol. 2, p. 023007, Apr. 2024.

[11] J. Pillow and M. Park, "Adaptive Bayesian Methods for Closed-Loop Neurophysiology," in *Closed Loop Neuroscience*, pp. 3–18, Cambridge, MA: Academic Press, 2016.

[12] W. Gerstner, W. M. Kistler, R. Naud, and L. Paninski, *Neuronal Dynamics: From Single Neurons to Networks and Models of Cognition*. Cambridge, UK: Cambridge University Press, 2014.

[13] T. B. Burghi, M. Schoukens, and R. Sepulchre, "System identification of biophysical neuronal models," in *59th IEEE Conference on Decision and Control*, (Jeju Island, South Korea), pp. 6180–6185, Dec. 2020.

[14] D. Beniaguev, I. Segev, and M. London, "Single cortical neurons as deep artificial neural networks," *Neuron*, vol. 109, pp. 2727–2739.e3, Sept. 2021.

[15] K. W. Latimer, F. Rieke, and J. W. Pillow, "Inferring synaptic inputs from spikes with a conductance-based neural encoding model," *eLife*, vol. 8, p. e47012, Dec. 2019.

[16] D. Durstewitz, G. Koppe, and M. I. Thurm, "Reconstructing computational system dynamics from neural data with recurrent neural networks," *Nature Reviews Neuroscience*, vol. 24, pp. 693–710, Nov. 2023.

[17] M. Aguiar, A. Das, and K. H. Johansson, "Learning flow functions of spiking systems," in *Proceedings of the 6th Annual Learning for Dynamics & Control Conference*, pp. 591–602, PMLR, June 2024.

[18] M. Brenner, F. Hess, J. M. Mikhaeil, L. F. Bereska, Z. Monfared, P.-C. Kuo, and D. Durstewitz, "Tractable Dendritic RNNs for Reconstructing Nonlinear Dynamical Systems," in *Proceedings of the 39th International Conference on Machine Learning*, pp. 2292–2320, PMLR, June 2022.

[19] Q. J. M. Huys, M. B. Ahrens, and L. Paninski, "Efficient Estimation of Detailed Single-Neuron Models," *Journal of Neurophysiology*, vol. 96, pp. 872–890, Aug. 2006.

[20] A. A. Prinz, D. Bucher, and E. Marder, "Similar network activity from disparate circuit parameters," *Nature Neuroscience*, vol. 7, pp. 1345–1352, Dec. 2004.

[21] K. Doya, "Bifurcations in the learning of recurrent neural networks," in *[Proceedings] 1992 IEEE International Symposium on Circuits and Systems*, vol. 6, pp. 2777–2780 vol.6, May 1992.

[22] A. H. Ribeiro, K. Tiels, J. Umenberger, T. B. Schön, and L. A. Aguirre, "On the smoothness of nonlinear system identification," *Automatica*, vol. 121, p. 109158, Nov. 2020.

[23] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *Proceedings of the 30th International Conference on Machine Learning*, vol. 28 of *ICML'13*, (Atlanta, GA, USA), pp. 1310–1318, June 2013.

[24] J. Keener and J. Sneyd, *Mathematical Physiology*, vol. 8/1. New York, NY: Springer, 2 ed., 2009.

[25] S. Boyd and L. Chua, "Fading memory and the problem of approximating nonlinear operators with Volterra series," *IEEE Transactions on Circuits and Systems*, vol. 32, pp. 1150–1161, Nov. 1985.

[26] I. W. Sandberg, "Uniform approximation and the circle criterion," *IEEE Transactions on Automatic Control*, vol. 38, pp. 1450–1458, Oct. 1993.

[27] A. L. Hodgkin and A. F. Huxley, "A quantitative description of membrane current and its application to conduction and excitation in nerve," *The Journal of Physiology*, vol. 117, no. 4, pp. 500–544, 1952.

[28] J. Golowasch and E. Marder, "Ionic currents of the lateral pyloric neuron of the stomatogastric ganglion of the crab," *Journal of Neurophysiology*, vol. 67, pp. 318–331, Feb. 1992.

[29] A. L. Taylor, J.-M. Goaillard, and E. Marder, "How Multiple Conductances Determine Electrophysiological Properties in a Multicompartment Model," *Journal of Neuroscience*, vol. 29, pp. 5573–5586, Apr. 2009.

[30] E. M. Izhikevich, *Dynamical Systems in Neuroscience*. Cambridge, MA: MIT Press, 2007.

[31] G. B. Ermentrout and D. H. Terman, *Mathematical Foundations of Neuroscience*. New York: Springer, 2010.

[32] G. Drion, A. Franci, J. Dethier, and R. Sepulchre, "Dynamic Input Conductances Shape Neuronal Spiking," *eneuro*, vol. 2, pp. ENEURO.0031–14.2015, Feb. 2015.

[33] W. V. Geit, E. D. Schutter, and P. Achard, "Automated neuron model optimization techniques: A review," *Biological Cybernetics*, vol. 99, pp. 241–251, Nov. 2008.

[34] M. Almog and A. Korngreen, "Is realistic neuronal modeling realistic?," *Journal of Neurophysiology*, vol. 116, pp. 2180–2209, Nov. 2016.

[35] D. A. McCormick, Y. Shu, and Y. Yu, "Hodgkin and Huxley model — still standing?," *Nature*, vol. 445, pp. E1–E2, Jan. 2007.

[36] E. Marder and D. Bucher, "Understanding Circuit Dynamics Using the Stomatogastric Nervous System of Lobsters and Crabs," *Annual Review of Physiology*, vol. 69, no. 1, pp. 291–316, 2007.

[37] D. M. Blitz and M. P. Nusbaum, "State-Dependent Presynaptic Inhibition Regulates Central Pattern Generator Feedback to Descending Inputs," *Journal of Neuroscience*, vol. 28, pp. 9564–9574, Sept. 2008.

[38] E. Marder, G. J. Gutierrez, and M. P. Nusbaum, "Complicating connectomes: Electrical coupling creates parallel pathways and degenerate circuit mechanisms," *Developmental Neurobiology*, vol. 77, no. 5, pp. 597–609, 2017.

[39] B. Johnson, J. Peck, and R. Harris-Warrick, "Differential modulation of chemical and electrical components of mixed synapses in the lobster stomatogastric ganglion," *Journal of Comparative Physiology A*, vol. 175, pp. 233–249, Aug. 1994.

[40] M. Innes, "Flux: Elegant machine learning with julia," *Journal of Open Source Software*, 2018.

[41] D. Nguyen and B. Widrow, "Improving the learning speed of 2-layer neural networks by choosing initial values of the adaptive weights," in *1990 IJCNN International Joint Conference on Neural Networks*, pp. 21–26 vol.3, June 1990.

[42] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pp. 249–256, JMLR Workshop and Conference Proceedings, Mar. 2010.

[43] P. S. C. Heuberger, P. M. J. van den Hof, and B. Wahlberg, *Modelling and Identification with Rational Orthogonal Basis Functions*. London: Springer, Dec. 2005.

[44] Z. Liu, J. Golowasch, E. Marder, and L. F. Abbott, "A Model Neuron with Activity-Dependent Conductances Regulated by Multiple Calcium Sensors," *Journal of Neuroscience*, vol. 18, pp. 2309–2320, Apr. 1998.

[45] K. J. Aström and R. M. Murray, *Feedback Systems: An Introduction for Scientists and Engineers*. Princeton: Princeton University Press, Apr. 2008.

[46] H. D. I. Abarbanel, D. R. Creveling, and J. M. Jeanne, "Estimation of parameters in nonlinear systems using balanced synchronization," *Physical Review E*, vol. 77, Jan. 2008.

[47] A. H. Ribeiro, K. Tiels, L. A. Aguirre, and T. Schön, "Beyond exploding and vanishing gradients: Analysing RNN training using attractors and smoothness," in *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, pp. 2370–2380, PMLR, June 2020.

[48] T. B. Burghi, M. Schoukens, and R. Sepulchre, "Feedback identification of conductance-based models," *Automatica*, vol. 123, p. 109297, Jan. 2021.

[49] L. Ljung, *System Identification: Theory for the User*. Upper Saddle River, NJ: Prentice Hall PTR, 1999.

[50] A. A. Prinz, C. P. Billimoria, and E. Marder, "Alternative to Hand-Tuning Conductance-Based Models: Construction and Analysis of Databases of Model Neurons," *Journal of Neurophysiology*, vol. 90, pp. 3998–4015, Dec. 2003.

Supporting Information Appendix (SI)

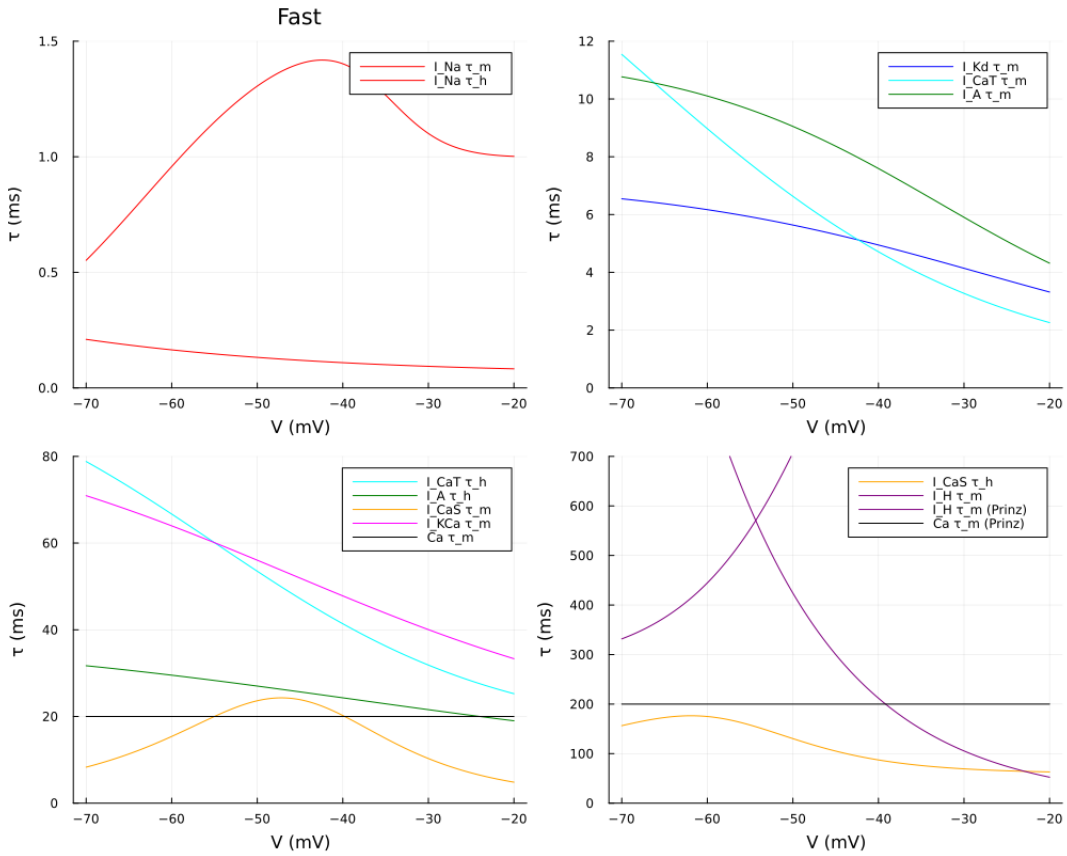

Figure S1: Time constant functions of the conductance-based model of [44]. These functions are also used in [50], who considered different time constants for calcium concentration and the H current, and doubled the values of all time constant functions plotted above.

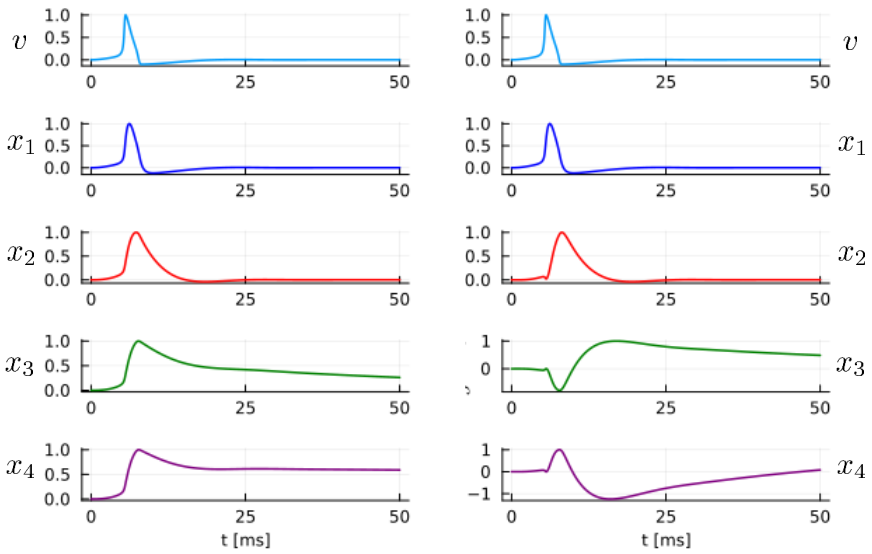

Figure S2: Spike responses of two different choices of state space systems. **Left:** state component responses $x_{i,t}$ of a four-dimensional diagonal state space system given by $A = \mathrm{diag}(\lambda_i)$, $B = ((1-\lambda_1), \ldots, (1-\lambda_n))^\intercal$). Here, $\lambda_i = \exp(-\delta/\tau_i)$ is computed using a sampling period of $\delta = 0.01$ and continuous time constants $\tau_1 = 0.5$, $\tau_2 = 5$, $\tau_3 = 50$, and $\tau_4 = 500$. Spike responses, which are shown normalized by maximum value, are the output of the system to the spike shown on top. **Right:** spike responses of orthogonal filters, constructed with the same sequence of λ_i according to the procedure described in Methods. In the immediate aftermath of a spike, orthogonal filters provide richer signal features for artificial neural networks to be trained and recombine into ionic currents.