Inferring parameters of pyramidal neuron excitability in mouse models of Alzheimer’s disease using biophysical modeling and deep learning

Saghafi, S., Rumbell, T., Gurev, V., Kozloski, J., Tamagnini, F. ORCID: https://orcid.org/0000-0002-8741-5094, Wedgwood, K. C. A. and Diekman, C. O. (2024) Inferring parameters of pyramidal neuron excitability in mouse models of Alzheimer’s disease using biophysical modeling and deep learning. Bulletin of Mathematical Biology, 86 (5). 46. ISSN 1522-9602 doi: 10.1007/s11538-024-01273-5 Available at https://centaur.reading.ac.uk/119119/

To link to this article DOI: http://dx.doi.org/10.1007/s11538-024-01273-5

Publisher: Springer

Bulletin of Mathematical Biology (2024) 86:46
https://doi.org/10.1007/s11538-024-01273-5

Society for Mathematical Biology

ORIGINAL ARTICLE

Inferring Parameters of Pyramidal Neuron Excitability in Mouse Models of Alzheimer's Disease Using Biophysical Modeling and Deep Learning

Soheil Saghafi[1,5] · **Timothy Rumbell**[2] · **Viatcheslav Gurev**[2] · **James Kozloski**[2] · **Francesco Tamagnini**[3] · **Kyle C. A. Wedgwood**[4] · **Casey O. Diekman**[1]

Received: 28 February 2023 / Accepted: 19 February 2024

Abstract

Alzheimer's disease (AD) is believed to occur when abnormal amounts of the proteins amyloid beta and tau aggregate in the brain, resulting in a progressive loss of neuronal function. Hippocampal neurons in transgenic mice with amyloidopathy or tauopathy exhibit altered intrinsic excitability properties. We used deep hybrid modeling (DeepHM), a recently developed parameter inference technique that combines deep learning with biophysical modeling, to map experimental data recorded from hippocampal CA1 neurons in transgenic AD mice and age-matched wildtype littermate controls to the parameter space of a conductance-based CA1 model. Although mechanistic modeling and machine learning methods are by themselves powerful tools for approximating biological systems and making accurate predictions from data, when used in isolation these approaches suffer from distinct shortcomings: model and parameter uncertainty limit mechanistic modeling, whereas machine learning methods disregard the underlying biophysical mechanisms. DeepHM addresses these shortcomings by using conditional generative adversarial networks to provide an inverse mapping of data to mechanistic models that identifies the distributions of mechanistic modeling parameters coherent to the data. Here, we demonstrated that DeepHM accurately infers parameter distributions of the conductance-based model on several test cases using synthetic data generated with complex underlying parameter structures.

✉ Casey O. Diekman
diekman@njit.edu

[1] Department of Mathematical Sciences, New Jersey Institute of Technology, University Heights, Newark, NJ 07102, USA

[2] IBM T.J. Watson Research Center, Yorktown Heights, NY 10598, USA

[3] Pharmacology, University of Reading, Reading, UK

[4] Mathematics and Statistics, University of Exeter, Exeter, UK

[5] Department of Biomedical Informatics, School of Medicine, Emory University, Atlanta, GA 30322, USA

Published online: 25 March 2024

We then used DeepHM to estimate parameter distributions corresponding to the experimental data and infer which ion channels are altered in the Alzheimer's mouse models compared to their wildtype controls at 12 and 24 months. We found that the conductances most disrupted by tauopathy, amyloidopathy, and aging are delayed rectifier potassium, transient sodium, and hyperpolarization-activated potassium, respectively.

Keywords Pyramidal neuron excitability · Parameter inference · Generative adversarial network · Population of models

1 Introduction

Although the underlying cause of Alzheimer's disease (AD) remains poorly understood, it is believed to occur when abnormal amounts of the proteins amyloid beta and tau aggregate in the brain, forming extracellular plaques (amyloidopathy) and neurofibrillary tangles (tauopathy) that result in a progressive loss of neuronal function and dementia (Hardy 2009; Spillantini and Goedert 2013). In transgenic mice with amyloidopathy, neurons in the hippocampus—a brain structure critical for memory—exhibit altered intrinsic excitability properties, such as action potentials with reduced peaks and widths (Tamagnini et al. 2015a, b; Vitale et al. 2021). Hippocampal neurons in transgenic mice with tauopathy also show altered excitability, but in different properties such as hyperpolarization-activated membrane potential sag and action potential threshold (Booth et al. 2016; Tamagnini et al. 2017).

Ideally, biophysical modeling could be used to gain insights into the mechanisms underlying the disrupted electrophysiological properties of these Alzheimer's mutant mice. However, determining whether or not such a biophysical model and its outputs are coherent with a set of experimental observations is a major challenge since such models contain many unknown parameters and are not amenable to statistical inference due to their non-invertibility. The main difficulty in solving the inverse problem for mechanistic models arises from intractability of the likelihood function (Cranmer et al. 2020). On the other hand, neither purely statistical models with tractable likelihoods nor purely data-driven machine learning algorithms offer much insight into underlying biological mechanisms (Gonçalves et al. 2020; Panahi et al. 2021). Here, we use deep learning to perform inversion of complex biophysical models and enable the mapping of experimental data into the space of biophysical model parameters. Since this approach combines deep learning with mechanistic modeling, we refer to it as deep hybrid modeling (DeepHM).

In biological systems, the tremendous amount of inherent cell-to-cell variability presents a significant challenge to mapping experimental data to underlying cellular mechanisms. It is common to handle this variability by simply averaging over the data and finding a single set of model parameters that best fits the averaged data. The "populations of models" approach allows deterministic models to reflect the inherent variability in biological data through identification of not just the single best parameter set but a population of parameter sets such that the output of the group of models displays the same heterogeneity as the population being modeled (Gonçalves et al. 2020; Lawson et al. 2018; Britton et al. 2013; Marder and Taylor 2011; Prinz et al. 2004;

Sobie 2009; Allen et al. 2016). The problem of constructing populations of deterministic models and identifying distributions of model input parameters from observations from multiple individuals in a population is known as the stochastic inverse problem (SIP). See Butler et al. (2018) and Pilosov et al. (2023) for formal discussions of SIP terminology. State-of-the-art methods for solving SIPs apply Bayesian inference techniques, including Markov chain Monte Carlo (MCMC) sampling, and are limited to finding a distribution for a single set of observations (Lawson et al. 2018; Parikh et al. 2022; Cheng et al. 2017; Rieger et al. 2018). To draw inferences about a new target dataset, the SIP would have to be solved again. We and others have recently proposed alternative approaches to solving SIPs, using generative adversarial networks (GANs), that enable *amortized inference*—i.e, the trained GAN can be reused on many target datasets without retraining (Rumbell et al. 2023; Ramesh et al. 2022).

GANs are a deep learning paradigm involving two artificial neural networks that compete with each other. The *Generator* network attempts to produce fake samples that are as similar as possible to a distribution of real samples, and the *Discriminator* network tries to distinguish fake samples from real samples. Since being introduced in 2014, GANs have garnered significant interest across a wide range of fields, including applications in image processing, cybersecurity, cryptography, and neuroscience (Goodfellow et al. 2020; Hitawala 2018; Molano-Mazon et al. 2018). Several extensions of GANs have been developed to address particular tasks (Rosenfeld et al. 2022; Butte et al. 2022; Gulrajani et al. 2017). To solve SIPs, we use a conditional GAN (cGAN) structure (Mirza and Osindero 2014) where the Generator is trained with parameter sets X conditioned on the output features Y of a mechanistic model.

In this paper, we wish to solve an SIP to identify which ion channels are responsible for the altered excitability properties of hippocampal neurons in mouse models of amyloidopathy and tauopathy. The data for the SIP of interest in this paper are voltage traces recorded from hippocampal CA1 neurons in 12-month-old rTg4510 mice expressing pathogenic tau (Tamagnini et al., unpublished data), 24-month-old PDAPP mice overexpressing amyloid beta (Tamagnini et al. 2015a), and age-matched wildtype littermate controls for each transgenic phenotype. From these traces, we extract several electrophysiological features (such as action potential peak, width, and threshold) that capture the excitability properties of the cells. Neuronal excitability can be simulated using the conductance-based modeling formalism originally developed by Hodgkin and Huxley (Kass et al. 2018). The mechanistic model for the SIP of interest in this paper is a conductance-based model of CA1 neurons that has been shown to be capable of reproducing key electrophysiological features of recorded voltage traces (Booth et al. 2016; Nowacki et al. 2011). By solving this SIP, we will map the features of the recorded voltage traces in the AD mutant and wildtype mice to the parameter space of the CA1 model. Our goal is to use the resulting parameter distributions to infer which ion channel conductances are disrupted in the amyloidopathy and tauopathy mice compared to their age-matched wildtype controls, and which conductances change with age in the wildtype mice.

The remainder of the paper is organized as follows. In Sect. 2, we describe the experimental data and the features we extract from the recorded action potentials and hyperpolarization traces. In Sect. 3, we introduce a biophysical model of CA1 pyramidal neurons and our initial optimizations of the model parameters using differential evolution. In Sect. 4, we give a brief description of GANs and cGANs and illustrate our parameter inference methodology using the Rosenbrock function as a toy model. In Sect. 5, we train the cGAN on output of the CA1 model and then present it with synthetic target data. We show that cGAN outperforms a benchmark MCMC method on a relatively simple parameter inference task. We then validate its ability to accurately infer complex parameter distributions through a series of tests with synthetic target data. In Sect. 6, we present the trained cGAN with real target data and use the inferred parameter distributions to identify which ionic conductances are affected by age, amyloidopathy, and tauopathy. We conclude with a discussion of alternative methods and future work in Sect. 7.

2 Experimental Data and Feature Extraction

The experimental data we use consists of patch-clamp recordings made from hippocampal CA1 neurons associated with two previous studies involving mouse models of Alzheimer's disease. In Tamagnini et al. (2015a), CA1 current-clamp recordings were obtained from transgenic PDAPP mice exhibiting amyloidopathy. In unpublished data, Tamagnini et al. obtained CA1 current-clamp recordings from transgenic rTg4510 mice exhibiting tauopathy. In this paper, we use voltage traces from $n = 30$ cells of 24-month-old PDAPP mice (and $n = 19$ cells from their age-matched WT littermate controls) and $n = 26$ cells of 12-month-old rTg4510 mice (and $n = 26$ cells from their age-matched WT littermate controls).

To characterize the excitability of these cells, we focused on the properties of the first action potential (AP) elicited in response to a square depolarizing current pulse (300 pA, 500 ms; Fig. 1a) and on the electrotonic properties of the plasma membrane measured upon the membrane potential exponential decay in response to a square hyperpolarizing current pulse (-100 pA, 500 ms; Fig. 1b). To account for the biasing effect of cell-to-cell variability of the membrane potential over the excitability properties, the cells were held at a membrane potential of $V \approx -80$ mV until the depolarizing or hyperpolarizing current pulse began. This V value was obtained via the constant injection of a biasing current. To summarize the behavior of these voltage traces, we defined 9 features associated with the APs and 4 features associated with the membrane hyperpolarization.

The AP features are illustrated in Fig. 2a–b and are defined as follows:

1. *AP threshold:* voltage at 10 percent of the AP max positive rate of rise (feature 6)
2. *AP peak:* maximum value of the voltage trace
3. *AP trough:* minimum value of the voltage in the 2 ms time interval after the AP peak
4. *AP width:* duration of time that the voltage is above the AP voltage at max positive rate of rise (feature 7)
5. *AP min voltage before the pulse:* minimum voltage in the 1 ms interval before the AP peak (note there is a delay of several ms between the beginning of the depolarizing pulse and the AP peak, thus this feature will take different values for different cells despite all cells being held near -80 mV until the pulse begins)

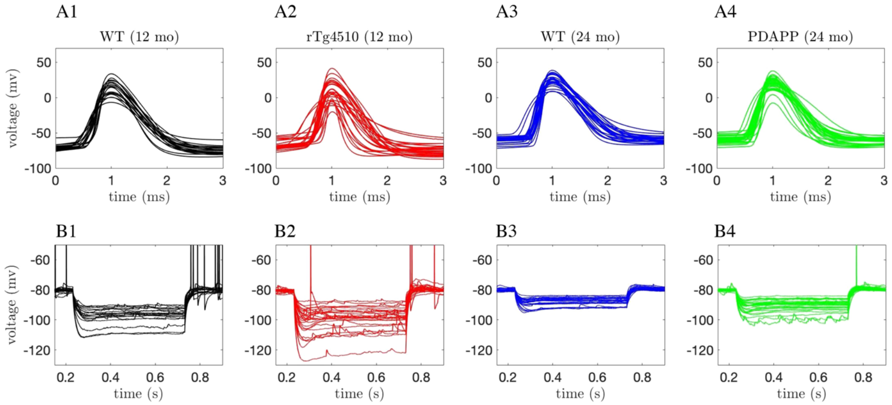

Fig. 1 Experimental recordings. Waveforms of the first action potential in response to depolarizing current pulses (**a1**–**a4**) and voltage traces in response to hyperpolarizing current pulses (**b1**–**b4**) injected into CA1 pyramidal neurons for 4 different categories of mice: wildtype (WT) 12-month-old mice (black traces, **a1**–**b1**), tau mutant (rTg4510) 12-month old mice (red traces, **a2**–**b2**), WT 24-month old mice (blue traces, **a3**–**b3**), and amyloid beta mutant (PDAPP) 24-month-old mice (green traces, **a4**–**b4**) (Color figure online)

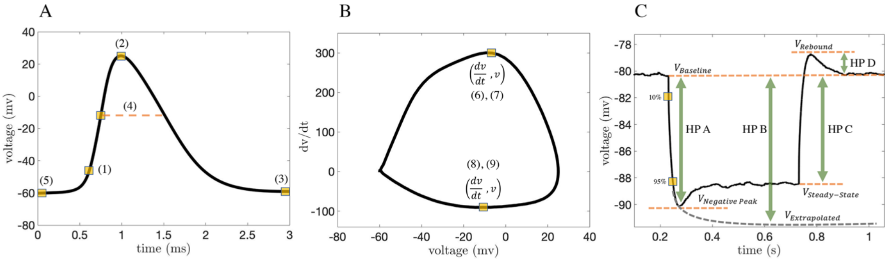

Fig. 2 Schematic of feature extraction. **a**–**b** Action potential features: (1) AP threshold, (2) AP peak, (3) AP trough, (4) AP width, (5) AP min voltage before the pulse, (6) AP max positive rate of rise, (7) AP voltage at max positive rate of rise, (8) AP max negative rate of rise, (9) AP voltage at max negative rate of rise. **c** Hyperpolarization features: (10) HP A—voltage at negative peak and baseline differences, (11) HP B—voltage at exponential fit and baseline differences, (12) HP C—voltage at steady state and baseline differences and (13) HP D—voltage at rebound and baseline differences (Color figure online)

6. *AP max positive rate of rise:* maximum value of dV/dt in the 3 ms time interval around the AP peak (i.e. 1 ms before and 2 ms after the peak)
7. *AP voltage at max positive rate of rise:* voltage value at the AP max positive rate of rise
8. *AP max negative rate of rise:* minimum value of dV/dt in the 3 ms time interval around the AP peak (i.e. 1 ms before and 2 ms after the peak)
9. *AP voltage at max negative rate of rise:* voltage value at the AP max negative rate of rise.

The membrane hyperpolarization features are illustrated in Fig. 2c and are defined as follows:

10. *HP A* voltage at negative peak and baseline differences
11. *HP B* voltage at exponential fit and baseline differences

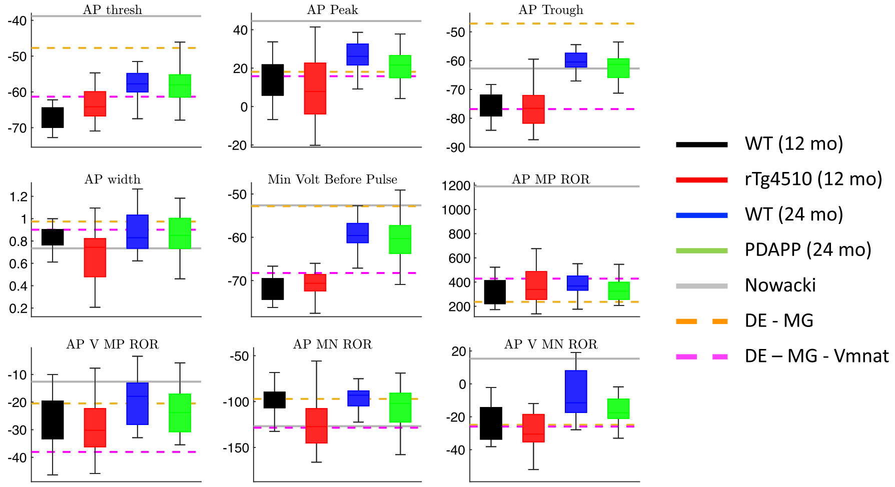

Fig. 3 Action potential features in experimental data and initial models. Box and whisker plots of the action potential (AP) features extracted from the experimental data and the biophysical CA1 model with three different parameter sets: (1) the parameters in the original Nowacki et al. paper ((2011), solid gray lines), (2) parameters obtained using differential evolution, optimizing only the maximal conductance parameters (DE-MG, dashed orange lines), and (3) parameters obtained using differential evolution, optimizing the maximal conductances and the half-activation voltage of the transient sodium current (DE-MG-Vmnat, dashed magenta lines) (Color figure online)

12. *HP C* voltage at steady state and baseline differences
13. *HP D* voltage at rebound and baseline differences.

We note that these features were chosen to try to capture as much of the behavior of the voltage traces in as few features as possible. Increasing the dimensionality of the feature space can reduce the accuracy of cGAN training if the additional features are not sufficiently informative.

We then calculated these features for the voltage traces from PDAPP, rTg4510, and WT mice (see Fig. 3 for the AP features, and Fig. 4 for the hyperpolarization features). Despite the large amount of variability within each category, for some features clear differences are observed across categories. For example, AP peak appears to be reduced in PDAPP mice compared to their WT controls (Fig. 3 top middle panel) and AP width appears to be reduced in rTg4510 mice compared to their WT controls (Fig. 3 middle left panel).

3 Biophysical Model

CA1 Pyramidal Neuron Model

Conductance-based modeling to describe the electrical activity of neurons was introduced by Hodgkin and Huxley in 1952 to explain the ionic mechanisms underlying the initiation and propagation of action potentials (APs) in the squid giant axon (Hodgkin and Huxley 1952). Nowacki et al. (2011) developed a conductance-based model of

CA1 pyramidal neurons that includes the following ionic currents: two Na^+-currents, one transient (I_{Na_T}) and one persistent (I_{Na_P}); two Ca^{2+}-currents, one T-type (I_{Ca_T}) and one high-voltage activated (I_{Ca_H}); and three K^+-currents, delayed rectifier ($I_{K_{DR}}$), M-type (I_{K_M}), and leak (I_L). The dynamics of the membrane potential V and ionic gating variables x are governed by the following system of ordinary differential equations:

$$C\frac{dV}{dt} = I_{app} - I_{Na_T} - I_{Na_P} - I_{Ca_T} - I_{Ca_H} - I_{K_{DR}} - I_{K_M} - I_L - I_{K_H} \quad (1)$$

$$\frac{dx}{dt} = \frac{x_\infty - x}{\tau_x} \quad (2)$$

where:

$$\begin{aligned} I_{Na_T} &= g_{Na_T} m^3_{Na_{T_\infty}} h_{Na_T}(V - E_{Na}), \quad I_{Na_P} = g_{Na_P} m_{Na_{P_\infty}}(V - E_{Na}) \\ I_{Ca_T} &= g_{Ca_T} m^2_{Ca_T} h_{Ca_T}(V - E_{Ca}), \quad I_{Ca_H} = g_{Ca_H} m^2_{Ca_H} h_{Ca_H}(V - E_{Ca}) \\ I_{K_{DR}} &= g_{K_{DR}} m_{K_{DR}} h_{K_{DR}}(V - E_K), \quad I_{K_M} = g_{K_M} m_{K_M}(V - E_K) \\ I_L &= g_L(V - E_L), \quad I_H = g_H(p\,m_H + (1-p)\,n_H)(V - E_H) \end{aligned}$$

and $x \in \{h_{Na_T}, m_{Ca_T}, h_{Ca_T}, m_{Ca_H}, h_{Ca_H}, m_{K_{DR}}, h_{K_{DR}}, m_{K_M}, m_H, n_H\}$.

The ionic currents I are described by Ohm's Law with maximal conductance parameters g and reversal potentials E. The steady-state activation and inactivation functions x_∞ for all gating variables, including m_{NaT} and m_{NaP}, are given in Boltzmann form:

$$x_\infty(V) = \frac{1}{1 + \exp\left(-\frac{V - V_x}{k_x}\right)}.$$

The time constants τ_x for all gating variables are fixed parameters, except for h_{NaT}, for which the time constant is a voltage-dependent function:

$$\tau_{h_{Na_T}}(V) = 0.2 + 0.007\exp\left(\exp\left(-(V - 40.6)/51.4\right)\right).$$

First, we simulated the pyramidal neuron model with the parameter values provided in Nowacki et al. (2011) (Supplementary Table 1) and calculated feature values based on the model output (i.e. the simulated voltage trace). For certain features, the model's feature value is outside the range of the feature values observed in the experimental data (solid gray lines in Figs. 3 and 4). For example, the AP threshold and AP peak in the model are significantly more depolarized than the AP thresholds and peaks seen in these CA1 neurons (Fig. 3 top left and top middle panels).

Thus, we used stochastic global optimization to find model parameters that produce model output with feature values consistent with the experimentally observed feature values. Specifically, we used differential evolution (DE), a population-based search technique first introduced by Storn and Price (1997) and Price et al. (2006). The objective function for the DE algorithm was to minimize the sum of squares

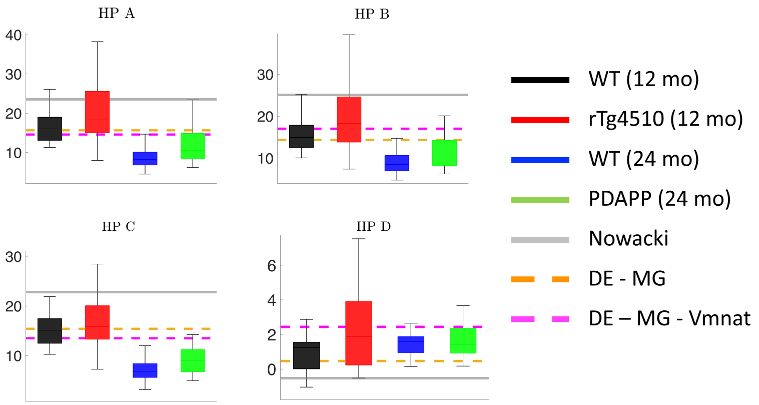

Fig. 4 Membrane hyperpolarization features in experimental data and initial models. Box and whisker plots of the membrane hyperpolarization features extracted from the experimental data and the biophysical CA1 model with three different parameter sets: (1) the parameters in the original Nowacki et al. paper ((Nowacki et al. 2011), solid gray lines), (2) parameters obtained using differential evolution, optimizing only the maximal conductance parameters (DE-MG, dashed orange lines), and (3) parameters obtained using differential evolution, optimizing the maximal conductances and the half-activation voltage of the transient sodium current (DE-MG-Vmnat, dashed magenta lines) (Color figure online)

error between the simulated voltage trace and the average voltage trace for each category (PDAPP, rTg4510, WT 12 and 24 month) across all four categories. More details on our implementation of the DE algorithm are provided in the Supplementary Methods.

Initially, we chose to hold all the reversal potentials and gating variable parameters at their original Nowacki et al. values, so the only free parameters for DE to optimize were the 8 maximal conductances. The model with optimized maximal conductances produced model output with feature values more consistent with the range of the feature values in the experimental data (dashed orange lines in Figs. 3 and 4). However, this model's AP threshold was still significantly more depolarized than in the data (Fig. 3 top left panel).

We used a variance-based sensitivity analysis (Sobol's method) to determine which model parameters affect the model's AP threshold, and found that the half-activation of the transient sodium current (V_{mNaT}) has the largest effect (see Supplementary Methods for a description of our sensitivity analysis procedure). We then ran DE again, this time with V_{mNaT} as a free parameter in addition to the maximal conductances. The model with optimized V_{mNaT} produced model output with feature values within the range of the feature values in the experimental data, including the AP threshold (dashed magenta lines in Figs. 3 and 4). Furthermore, the action potential and membrane hyperpolarization voltage traces produced by this model agree well with the experimental voltage traces themselves, as the model output appears to lie in the middle of the four voltage traces obtained by averaging the traces within each of the four categories (Fig. 5).

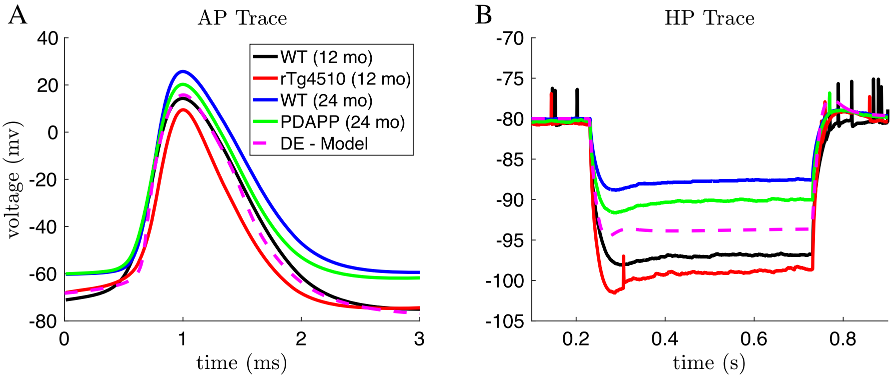

Fig. 5 Average AP and hyperpolarization voltage traces from experimental data and optimized model. The DE-model shown here is the same model that was labeled DE-MG-Vmnat in Fig. 4, and was obtained by minimizing the least square error between the DE-Model output and the average AP and hyperpolarization traces across all 4 categories. **a** Mean of the AP waveforms in the experimental data for each category, and the AP waveform simulated using the optimized DE model (magenta). **b** Mean of the membrane hyperpolarization traces in the experimental data for each category, and the hyperpolarization trace simulated using the optimized DE model (magenta) (Color figure online)

We refer to the parameter set obtained through DE with V_{mNaT} and maximal conductances as free parameters as the "default" model parameters. We then used these parameters to set parameter bounds when creating the training dataset for cGAN. The default parameters are provided in Supplementary Table 1.

4 Parameter Inference Methodology: Conditional Generative Adversarial Networks

Generative adversarial networks (GANs) are an example of generative models in machine learning. Since GANs have a deep neural network architecture we can classify them as deep learning models. In Sect. 4.1, we briefly describe conditional GANs (cGANs), which are an extension of GANs that enable conditioning of the generative model on additional information. In Sect. 4.2, we illustrate the process of training a cGAN on a simple mechanistic model. For an introduction to standard GANs, see Supplementary Methods.

The application we are interested in here is to build an inverse surrogate model for mapping the output of a mechanistic model into its corresponding region in parameter space. More precisely, the goal is to map the density of observed data ($\mathcal{P}_Y$) to a *coherent* density α_X of the model input parameter space. A distribution α_X is coherent if upon sampling from α_X and applying the mechanistic model, the estimated density in output space ($\hat{\mathcal{P}}_Y$) satisfies $\hat{\mathcal{P}}_Y \sim \mathcal{P}_Y$. Since there are an infinite number of possible α_x that are coherent with P_Y, the prior distribution specified and sampled for input space ($\mathcal{P}_X$) helps to obtain a unique solution α_x that is also consistent with the prior.

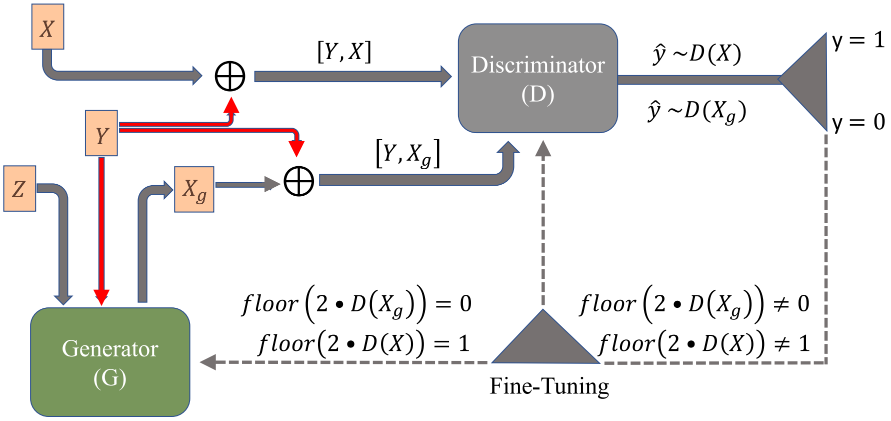

Fig. 6 Schematic of a conditional GAN (cGAN). Features of the training dataset (Y) are passed as a condition into the Generator (G)—already initialized with a Gaussian distribution (Z)—in order to produce some samples X_g given that condition $[Y, X_g]$. This output, along with real samples X augmented with their output features $[Y, X]$, are passed into the Discriminator (D). If the Discriminator's output is close to zero (one), it means the Discriminator has assigned a low (high) probability of that sample being real (Color figure online)

4.1 Conditional Generative Adversarial Networks

A standard GAN could be trained to produce samples of parameter sets, samples of feature sets, or even samples of combined parameter-feature sets. However, it is not able to produce samples of parameter sets corresponding to a set of given feature values. To accomplish this, we employ conditional GANs (cGANs) (Mirza and Osindero 2014), where features extracted from the output of the mechanistic model are passed as a condition to the Generator (G). The parameter samples produced by G, augmented with the features it was provided as a condition, are then passed to the Discriminator (D). Ground truth parameters, with their corresponding features, as a condition, are also passed to D. During the training process, G learns how to produce samples in parameter space that are similar to the ground truth parameters for a given set of features.

The overall structure of the cGAN is similar to the basic GAN model (see Section 2.6 of Supplementary Material and Fig. S1). However, the main difference is that the inputs into both G and D are augmented by the conditional variable Y, as shown in Fig. 6. The adversarial nature of the interaction between G and D is succinctly expressed by the min-max formulation of the cGAN objective function:

$$\min_{G} \max_{D} \{\mathbb{E}_{x \sim P_{data}(x)}[\log D(x \| y)] + \mathbb{E}_{z \sim P_z(z)}[\log(1 - D(G(z \| y)))]\} \tag{3}$$

where P_{data} represents the distribution of parameters in the training dataset and P_z represents the Gaussian distribution that the Generator is initialized with.

In practice, at the beginning of the training process, the Generator is not strong enough and the output of the Generator is very different from the training dataset.

Thus, the Discriminator can easily distinguish the real and the fake samples. This causes $\log(1-D(G(z\|y)))$ to saturate (see Fig. S2B), and the gradient does not provide any information as it is almost zero. Therefore, for our implementation in TensorFlow, we minimize $\log(1-D(G(z\|y)))-\log(D(G(z\|y)))$ instead of $\log(1-D(G(z\|y)))$, as suggested in Goodfellow et al. (2020). This will help ensure we have a more stable Generator loss function during the training process. Furthermore, for implementation purposes we convert the maximization of the Discriminator loss function into the corresponding minimization problem. Thus, the loss functions are implemented using the following forms:

$$\begin{cases} \underset{D}{min} & \{-\mathbb{E}_{x\sim P_{data}(x)}[\log D(x\|y)] - \mathbb{E}_{z\sim P_z(z)}[\log(1-D(G(z\|y)))]\} \\ \underset{G}{min} & \{\mathbb{E}_{z\sim P_z(z)}[\log(1-D(G(z\|y)))] - \mathbb{E}_{z\sim P_z(z)}[\log(D(G(z\|y)))]\} \end{cases} \tag{4}$$

Here we provide additional details about the cGAN model structure used in this paper:

1. G and D are both feedforward neural networks with 8 hidden layers and 180 and 130 nodes per layer, respectively.
2. All nodes in G and D use ReLU activation functions.
3. In total, 400 epochs were used in the training process and all weights of the Generator network were saved every 10 iterations.
4. The GAN model is produced via the unrolled GAN (Metz et al. 2016) numerical scheme, which uses the unrolled Adam method with a step size of 0.0005 and 4 to 8 iterations.
5. The step size of the Adam optimizer was 0.0001 for G and 0.00002 for D.
6. The β_1 and β_2 parameters of the Adam optimizer were set to their default values of 0.9 and 0.999, respectively (Kingma and Ba 2014).
7. The training dataset comprised 3,000,000 samples, with a mini-batch size set at 10,000.

Model hyperparameters, such as the number of layers and nodes per layer in the neural networks, were selected heuristically without extensive tuning, and the cGAN results should remain comparable with other choices. The step size of the Adam optimizer and the mini-batch size were chosen based on both computation time and performance. Code for running cGAN on an illustrative example involving a nonlinear function with two parameters is available at https://github.com/IBM/rgan-demo-pytorch/.

4.2 Illustration of cGAN Training Process

Over the course of the cGAN training process, both the Discriminator and the Generator improve. The Discriminator gets better at distinguishing between real (coming from P_{data}) and fake (coming from P_g) samples, and for a fixed Discriminator, the Generator gets better at fooling the Discriminator. To illustrate the process of training a cGAN, we used the Rosenbrock function (Eqn. 5) as a toy mechanistic model:

$$Y = (1-X_1)^2 + 100(X_2 - X_1^2)^2. \tag{5}$$

In this example, we have two input parameters (X_1 and X_2) and one feature (Y, the output of the Rosenbrock function). The training dataset consisted of one million samples of X_1 and X_2, taken from a uniform distribution with a range of [-5, 5], along with the Y values obtained by evaluating Eqn. 5 at these X values. During training, the cGAN is passed these Y values as a condition and asked to produce the corresponding X values. Figure 7 provides a visual summary of the training process. At the beginning of training (epoch 0), Kernel Density Estimate (KDE) plots of the X_1 and X_2 samples produced by the cGAN, as well as the Y values obtained by passing them through the Rosenbrock function, show that they are distributed differently than their counterparts in the training data (see Fig. 7a1). By training epoch 3, the KDE plots are in better agreement (Fig. 7a3), and both the Discriminator loss (Fig. 7b1) and Generator loss (Fig. 7b2) have decreased. The Jensen Shannon divergence (JSD), a measure of the similarity between the cGAN samples and training data distributions (see Supplementary Methods for more details on our JSD computation), also indicates a substantial improvement in cGAN performance over the first few training epochs (Fig. 7b3). By training epoch 46, it appears the cGAN has converged to the ground truth region in the (X_1,X_2) parameter space (Fig. 7a4). As training continues, the generator and discriminator losses continue to decrease, but the JSD measure begins to increase and the KDE plots again show discrepancies between the cGAN and training data distributions (Fig. 7a6). This phenomenon occurs due to the vanishing gradient problem, which destabilizes the training process (Roberts et al. 2022; Chen 2022). To address this, we use the JSD as a stopping criterion and select G from the epoch with the minimum JSD (in this example, epoch 142, Fig. 7a5) as our final trained cGAN.

5 cGAN Training on Biophysical Model and Validation on Synthetic Target Data

Recall that our main goal is to use cGAN to map voltage traces recorded from WT and Alzheimer's mutant mice to the parameter space of our CA1 model. To enable cGAN to learn the mapping from electrophysiological features to the CA1 model parameter space, we will create a training dataset consisting of features calculated from CA1 model simulations with randomly chosen parameter values. In Sect. 5.1, we explain how we selected which parameters to vary in the training dataset and show initial validation tests of the trained cGAN. In Sect. 5.2, we further validate the trained cGAN on synthetic target data and compare its performance to a benchmark method. Finally, in Sect. 5.3 we perform a series of parameter inference tests on synthetic target datasets with complex underlying parameter structures.

5.1 Training Dataset and Initial Validation Tests

Since we are primarily interested in how the maximal conductances of the various ion channels present in CA1 pyramidal neurons are affected by aging and amyloidopathy/tauopathy, we will only vary the maximal conductance parameters in our training

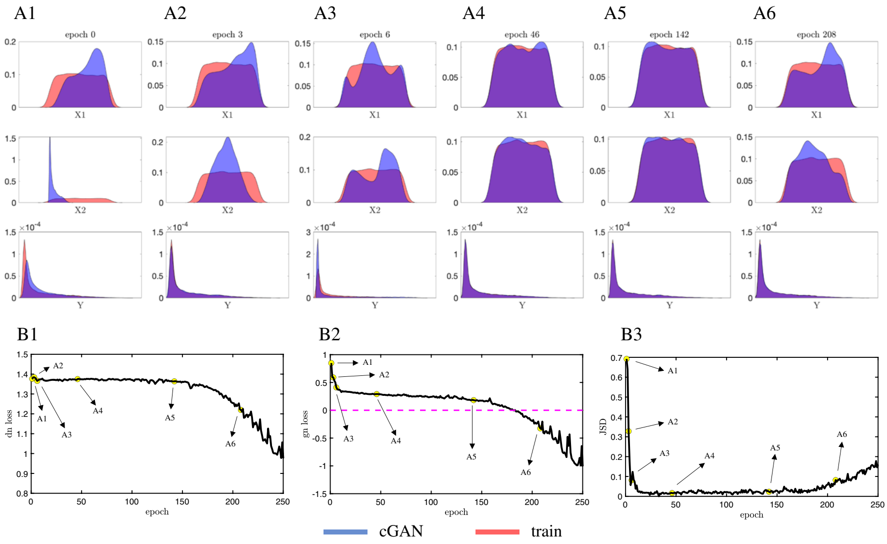

Fig. 7 Illustration of the cGAN training process on the Rosenbrock function (Eqn. 5). **(A1–A6)** KDE plots of the cGAN samples (blue) and the training data (red) for parameters $X1$, $X2$, and feature Y at epochs 0, 3, 6, 46, 142, and 208. At epoch 142 (**A5**), the cGAN distributions are good approximations of the training distributions, but by epoch 208 (**A6**) the cGAN distributions are no longer good approximations. **(B1–B3)** The Discriminator loss (**B1**), the generator loss (**B2**), and the JSD measure between the ground truth parameters versus estimated parameters (**B3**) as a function of epoch number. The point labeled A5 in panel B3 represents the JSD stopping criterion: once the JSD starts to increase we stop training the cGAN (Color figure online)

dataset and keep the gating variable kinetic parameters and the reversal potentials fixed at their default values. However, it may be that some of the maximal conductances do not have a large effect on the output features of interest. To explore this possibility, before creating a training dataset, we first conducted Sobol sensitivity analysis to see how each of the 8 maximal conductance parameters affect the features. We found that 3 of these conductances, g_{NaP}, g_{CaT}, and g_L have a smaller effect on the features than the other 5 conductances (Fig. 8). From a biological standpoint, these 3 conductances are unlikely to play a major role in determining the AP features for the following reasons: (1) persistent sodium current (I_{NaP}) is likely to be much smaller than transient sodium current (I_{NaT}), (2) T-type calcium (I_{CaT}) is likely to be much slower than I_{NaT}, and (3) the leak current (I_L) primarily affects resting membrane potential rather than AP dynamics. Therefore, when we create the training dataset, we keep those 3 conductances fixed at their default values, and only vary 5 conductances: g_{NaT}, g_{CaH}, g_{KDR}, g_{KM}, and g_H.

For the training dataset, we performed 3 million simulations of the CA1 model with these 5 conductances drawn from uniform distributions with upper and lower bounds at ±100% of their default values (see Table 1 in Supplemental Methods for these values). We then calculated the feature values for these simulations, and trained the cGAN with the parameters X conditioned on the features Y. Once the cGAN was trained, we passed the features for a subset of the training dataset (10,000 simulations)

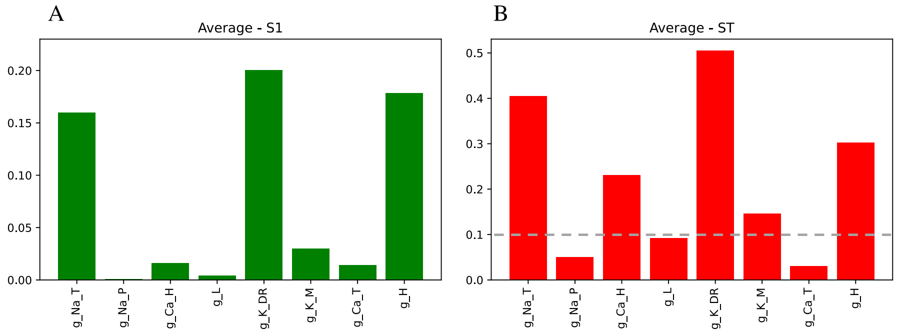

Fig. 8 Dimensionality reduction using variance-based (Sobol) sensitivity analysis. The height of the bars represent how sensitive the AP and hyperpolarization features are to each parameter. **a** S_1: average first-order index across feature space. **b** S_T: average total-effect index across feature space. Following Okoyo et al. (2022) and Marino et al. (2008), parameters with S_T values over 0.1 (above the gray dotted line) were determined to be highly influential on the model output (Color figure online)

to the trained cGAN and asked it to produce samples (i.e. parameter sets) for those features. We then simulated the CA1 model with the parameter sets from the cGAN and calculated the features from these simulations. To compare the distributions of features from the training dataset and from the cGAN samples, we plotted KDEs for each feature and scatter plots for each pair of features (Fig. 9a). These plots show that the cGAN samples produced features that were very consistent with the features in the training data. We also plotted KDEs and pairwise contour plots for each parameter, which show that the distributions of parameters produced by the cGAN are similar to the parameter distributions in the training dataset (Fig. 9b). This visual conclusion was confirmed by calculating the Jensen Shannon Divergence. The JSD between the cGAN samples and training data on the joint distribution of parameters and features was approximately zero (1.11×10^{-14}).

5.2 Comparison with Markov Chain Monte Carlo Method on Synthetic Target Data

Although the results shown in Fig. 9b are encouraging, it is important to test the performance of the cGAN on data that were not part of the training dataset. Since the cGAN was trained on data with uniformly distributed parameter values, for the remaining validation tests we use target data that is not uniformly distributed. In particular, to create synthetic target data to use for validation, we constructed 100 random parameter sets with each parameter p drawn from a normal distribution with mean μ_p and standard deviation $\mu_p/8$, where μ_p is the default value of that parameter (see Table 1 in Supplemental Methods for the μ_p values). If the randomly drawn value was negative or was larger than the upper bound for that parameter in the training set, then the value was set to zero or the upper bound, respectively.

We simulated these 100 parameter sets with the CA1 model, calculated the features, and passed the features to the trained cGAN to generate 100 cGAN parameter samples. We then simulated these cGAN parameter samples with the CA1 model, calculated the features, and compared the target and cGAN feature distributions. KDE plots for each feature, as well as 2D KDE plots for each pair of features, show that the cGAN feature distributions are similar to the target feature distributions (Figs. 10 and 11a lower triangles). Furthermore, KDE plots for each parameter and each pair of parameters show that the cGAN parameter distributions are similar to the parameter distributions used to generate the target data (Fig. 11b lower triangle). To confirm these visual conclusions, we performed two-sample Kolmogorov–Smirnov (KS) tests to compare the cGAN and target distributions in feature and parameter space. In all cases but one (the voltage at the maximum positive rate of rise feature), the KS test failed to reject the null hypothesis (with a p-value threshold of 0.01) that these two sets of samples come from the same probability distribution, suggesting that the cGAN distributions not statistically different from the target distributions.

We then performed the same procedure using a Markov chain Monte Carlo (MCMC) method instead of cGAN to infer parameters from the target data. The details of our standard MCMC implementation are provided in the Supplementary Methods. We chose MCMC as the benchmark method for comparison with cGAN because most state-of-the-art methods for solving stochastic inverse problems are based on MCMC (Lawson et al. 2018; Butler et al. 2018). We passed the same 100 target data features to the MCMC algorithm as we did the cGAN, and then simulated the parameters produced by MCMC with the CA1 model. The feature distributions produced by the MCMC parameters, and the distributions of the MCMC parameters themselves, differ from their respective target distributions (Figs. 10 and 11 upper triangles). Furthermore, we performed KS tests between the MCMC parameters and features and the target parameters and features, and in all cases the null hypothesis that these samples come from the same probability distribution was rejected, suggesting that the MCMC distributions are indeed different from the target distributions. We do not claim this will be the case in general, especially if one uses more advanced MCMC sampling algorithms. We do note, however, that the cGAN approach has computational benefits compared to MCMC-based methods when it comes to performing inference on new target datasets due to the inference amortization properties of the cGAN.

5.3 Parameter Inference Tests on Synthetic Target Data

To further test the ability of cGAN to accurately infer biophysical model parameters from feature data, we generated synthetic target datasets with a variety of underlying parameter structures. These structures were chosen to reflect the possible scenarios one may encounter when working with data from 2 different categories (e.g. data from WT versus mutant mice, or data from young versus old mice). For the CA1 model, we are interested in 5 parameters. Suppose that in the mutant mice, only 1 of these parameters (e.g., g_{NaT}) is altered compared to WT, and the other 4 parameters are not. To simulate this scenario, we construct two groups of target data. For each of the 4 parameters that are not hypothesized to be altered by the mutation (i.e., g_{CaH}, g_{KDR}, g_{KM}, and g_H), we draw 100 values from the same normal distribution $\mathcal{N}(\mu_p, (\mu_p/8)^2)$ for each group. For the parameter that is altered by the mutation (g_{NaT}), we draw 100 values from $\mathcal{N}(0.5\mu_p, (\mu_p/8)^2)$ for Group 1 and 100 values from $\mathcal{N}(1.5\mu_p, (\mu_p/8)^2)$ for Group 2. For each group, we then: (1) simulate these parameter sets using the CA1

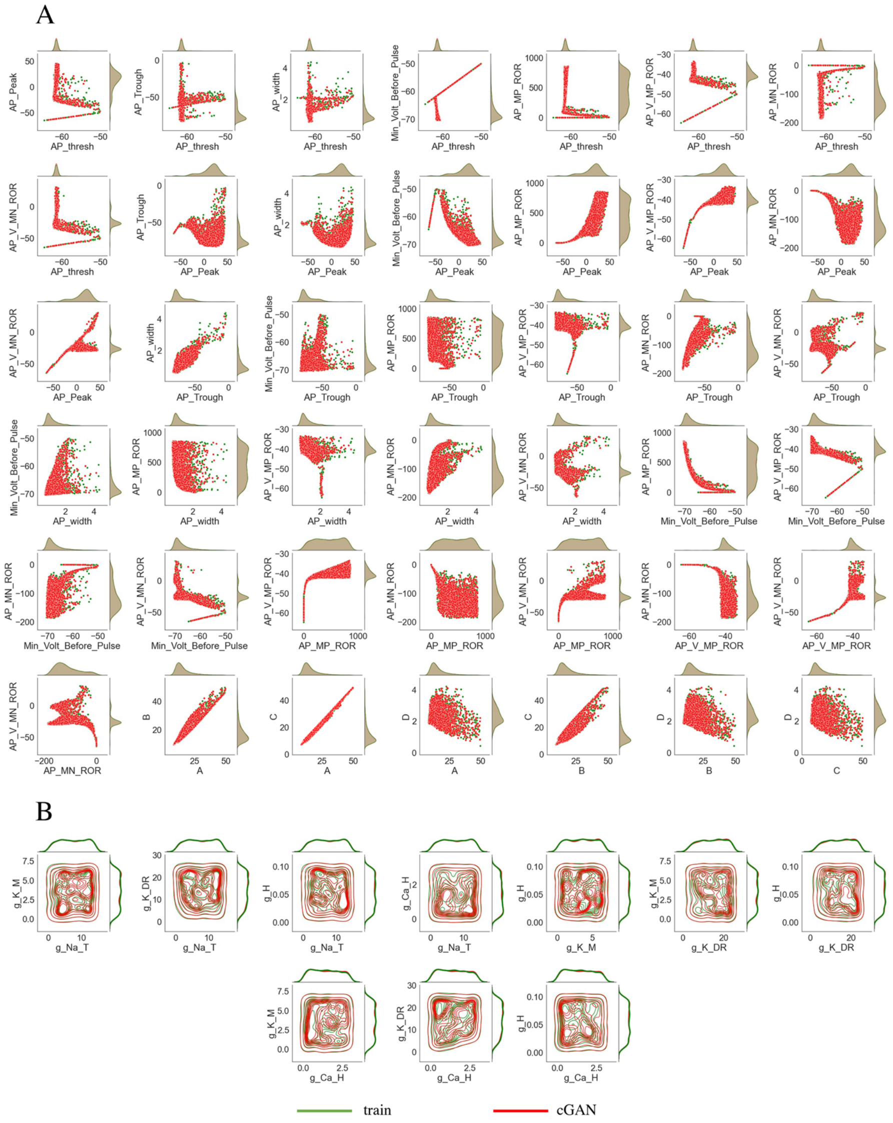

Fig. 9 Comparison of cGAN samples and training dataset. **a** Feature space: scatterplots (center of panel) and KDE plots (top and right of each panel) with cGAN samples in red and the training dataset in green. **b** Parameter space: contour plots (center of panel) and KDE plots (top and right of each panel) with cGAN samples in red and the training dataset in green. In both (**a**) and (**b**), the KDE plots for the cGAN samples are nearly identical to the KDE plots for the training dataset (and have almost zero JSD measure) (Color figure online)

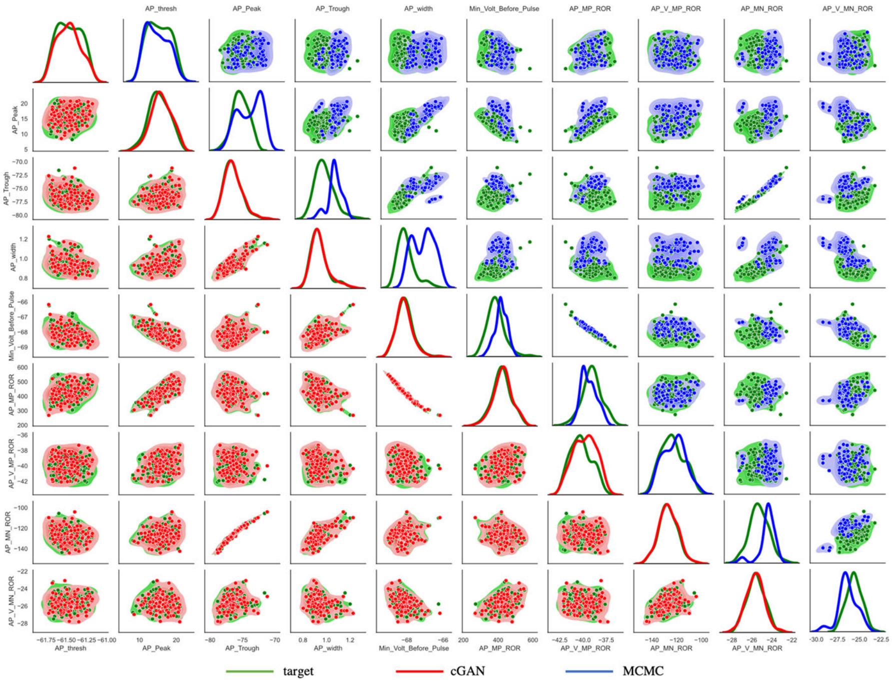

Fig. 10 Performance of cGAN and MCMC on synthetic target data—AP features. *Lower main diagonal and lower triangle*—KDE and scatter plots of the cGAN samples (red) versus target (green). *Upper main diagonal and upper triangle* - KDE and scatter plots of the MCMC samples (blue) versus target (green) (Color figure online)

model and calculate the features, (2) pass the features to the trained cGAN as target data to obtain cGAN samples (parameter sets), (3) simulate the cGAN parameter sets using the CA1 model and calculate the features, and (4) compare the cGAN feature and parameter distributions between Group 1 (G1) and Group 2 (G2) and to their respective targets. The G1 and G2 target distributions of some AP features are quite different from each other (Fig. S9 lower triangle), whereas the G1 and G2 membrane hyperpolarization feature target distributions are similar (Fig. S10 lower triangle), illustrating that the value of g_{NaT} affects some features more than others. Nonetheless, for all features the cGAN samples reproduce the target distributions well across both G1 and G2. Figure 12 (lower triangle) shows that the cGAN was able to accurately infer the distributions of all 5 parameters as well. Importantly, the cGAN-inferred distributions for g_{NaT} are distinct between Groups 1 and 2, whereas for the other 4 parameters the cGAN-inferred distributions are similar for G1 and G2.

We also used KS tests to assess the cGAN performance. First, we ran KS tests on the target data from G1 and G2. For the parameters, the null hypothesis that the G1 and G2 target samples come from the same distribution is rejected for g_{NaT}, but is not rejected for the other 4 parameters, as one would hope since this was the true structure used to create the target data. For the feature distributions, the KS tests reject the null for 10 out of the 13 features. When we ran the KS tests on the cGAN G1 and G2 distributions, we get the exact same results for both the parameters and features as we did for the target data. This consistency in KS test results suggests that cGAN is able to correctly identify the structure of parameter variations between two groups of samples based on their features. Additionally, we ran KS tests to compare the cGAN distributions for G1 to the target data for G1, and the cGAN distributions for G2 to the target data for G2. For G1, all of the KS tests (for both parameters and features) failed to reject the null, again suggesting that the cGAN distributions are not statistically different from the target distributions. For G2, all of the KS tests failed to reject the null except for one feature (the voltage at the maximum positive rate of rise). We then repeated this simulation and testing procedure 4 more times with the other possible choices for having a single parameter distinguish G1 and G2. For the G1 vs. G2 KS tests, the cGAN sample tests returned the same result as the target data tests 70 out of 72 times (Supplementary Fig. S2 top panels). For the cGAN sample vs. target data KS tests, the null was rejected 0 out of 72 times for G1 (Fig. S2 bottom left panel) and 2 out of 72 times for G2 (Fig. S2 bottom right panel). Overall, out of 270 KS tests, only 3 of them indicated a discrepancy between the cGAN distributions and the target distributions. We inspected these 3 cases further as shown in Fig. S8. We found that for the 1 test involving parameter estimation, the cGAN still correctly identified that g_{KM} took higher values in G2 than G1. For the 2 tests involving features, the cGAN correctly identified that the HP-B feature took lower values in G2 than G1 and reproduced the general shape of the AP-V-MP-ROR distribution reasonably well. Thus, in these cases, it appears the KS tests were detecting relatively minor differences between the cGAN and target distributions.

Next, we investigated scenarios with more than one parameter distinguishing G1 and G2. For example, we considered the case where g_{NaT} is not altered by the mutation, but the other 4 parameters all are altered by the mutation (i.e. $g_{NaT} \sim \mathcal{N}(\mu_p, (\mu_p/8)^2)$ in both G1 and G2, but g_{CaH}, g_{KDr}, g_{KM}, and g_H are all distributed $\mathcal{N}(0.5\mu_p, (\mu_p/8)^2)$ in G1 and $\mathcal{N}(1.5\mu_p, (\mu_p/8)^2)$ in G2). The KDE and scatter plots for the AP features (Fig. S9 upper triangle), membrane hyperpolarization features (Fig. S10 upper triangle), and parameters (Fig. 12 upper triangle) indicate that the cGAN samples are consistent with the target data distributions. We also simulated the 4 other cases where each of the other 4 parameters was the only one not altered by the mutation. For the G1 versus G2 KS tests, the cGAN sample tests returned the same result as the target data tests 89 out of 90 times (Fig. S5 top panels upper portion). For the cGAN sample vs. target data KS tests, the null was rejected 0 out of 90 times for G1 and 3 out of 90 times for G2 (Fig. S5 top panels, bottom left and bottom right portions, respectively).

There are 35 other ways that exactly 4 out of the 5 parameters could be altered by the mutation. In Fig. 12 (upper triangle), the other 4 parameters all had lower means in G1 than in G2. Instead, up to three of these parameters could have higher means in G1 than in G2 (if all 4 had higher means, it would be equivalent to the case we already simulated just with the G1 and G2 labels swapped). We simulated and tested these additional parameter structures. For the G1 versus G2 KS tests, the cGAN sample tests returned the same result as the target data tests 624 out of 630 times (Fig. S5 bottom 7 panels upper portions). For the cGAN sample vs. target data KS tests, the null was rejected 15 out of 630 times for G1 and 38 out of 630 times for G2 (Fig. S5 bottom 7 panels lower portions).

So far, we have discussed scenarios where either exactly 1 parameter was affected by the mutation or exactly 4 parameters were affected by the mutation. Here, we consider the remaining scenarios of exactly 2, 3, or 5 parameters being affected. The number of possible cases for each scenario is given by

$$\binom{5}{k} \times 2^{k-1}, \quad k = 1, \cdots, 5 \tag{6}$$

where k is the number of parameters affected by the mutation. Thus, for $k = 2$, we have 20 different cases. For the G1 vs. G2 KS tests, the cGAN sample tests returned the same result as the target data tests 353 out of 360 times (Fig. S3 upper portions of panels). For the cGAN sample vs. target data KS tests, the null was rejected 3 out of 360 times for G1 and 15 out of 360 times for G2 (Fig. S3 lower portions of panels). For $k = 3$, there are 40 different cases. For the G1 vs. G2 KS tests, the cGAN sample tests returned the same result as the target data tests 711 out of 720 times (Fig. S4 upper portions). For the cGAN sample vs. target data KS tests, the null was rejected 9 out of 720 times for G1 and 38 out of 720 times for G2 (Fig. S4 lower portions). Finally, for $k = 5$, we have 16 different cases. For the G1 vs. G2 KS tests, the cGAN sample tests returned the same result as the target data tests 288 out of 288 times (Fig. S6 upper portions). For the cGAN sample vs. target data KS tests, the null was rejected 9 out of 288 times for G1 and 13 out of 288 times for G2 (Fig. S6 lower portions). The results of the KS tests for all of the 5 choose k cases are summarized in Fig. S7.

In summary, these results on synthetic target data demonstrate that cGAN is capable of accurately identifying complex parameter variation structures from subtle differences in the features of CA1 model simulations. This gives us the confidence to apply the cGAN method to experimental data in Sect. 6.

6 Parameter Inference on Experimental Target Data

Now that we have established cGAN as a tool for mapping observed traces to unobserved/unmeasured parameter values, we turn our attention back to experimental data (Figs. 1, 3, 4) and seek to answer the following questions: Which maximal conductances are responsible for the differences observed in feature space between (1) the wild type versus the mutant mice (i.e. the disease effect), and (2) the 12-month old mice versus the 24-month old mice (i.e. the age effect)?

To answer these questions, we will pass the features for each cell to the cGAN to obtain a population of models for each individual cell. For some cells, certain feature values fall outside the range of the values for that feature used in our training dataset. This can lead to inaccurate cGAN samples; thus, if a cell has a feature value outside that range we replaced that value with the median value of that feature across the training dataset. We note that we also tested two other methods of replacement—replacing with the mean value of that feature in the training dataset, and replacing with the closest value of that feature in the training dataset to the feature value being replaced—and found that they gave similar results. We obtained 100 cGAN samples for each cell, and then pushed those parameters forward through the mechanistic model. Figure 13a

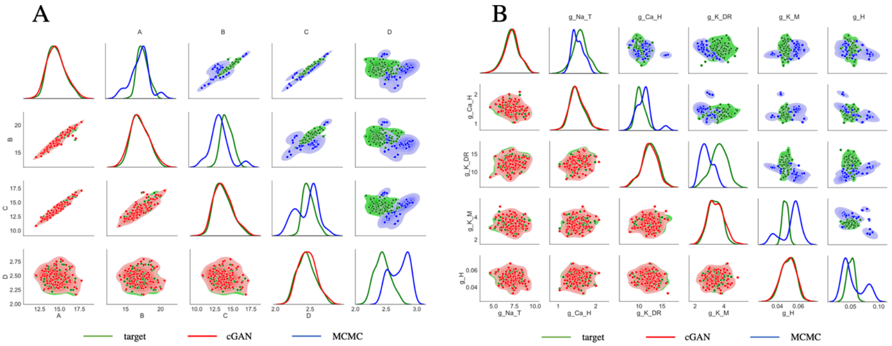

Fig. 11 Performance of cGAN and MCMC on synthetic target data—HP features and parameter space. *Lower main diagonal and lower triangle*—KDE and scatter plots of the cGAN samples (red) versus target (green). *Upper main diagonal and upper triangle*—KDE and scatter plots of the MCMC samples (blue) versus target (green). **a** HP features. **b** Parameter space. Note that the target data parameters are normally distributed, whereas the cGAN was trained on parameters drawn from uniform distributions (Color figure online)

shows that the mean AP and hyperpolarization traces produced by the cGAN samples agree well with the mean AP and hyperpolarization traces from the experimental recordings in each of the 4 categories. For example, we can see from the voltage traces that the AP peak feature exhibits the same trend in the cGAN samples and experimental data, with WT 24 month having the highest mean AP peak, followed by PDAPP, WT 12 month, and rTG4510, respectively. To give a sense of how the variability of the cGAN samples compares to the experimental recordings, Fig. 13b shows the mean ± standard deviation of the AP and hyperpolarization traces. Overall, the amount of variability in the cGAN samples appears comparable to the amount of variability in the experimental data across the 4 categories, with the exception of the hyperpolarization traces for WT 24 month where there is less variability in the cGAN samples than in the data. Furthermore, box-and-whisker plots for the output of the cGAN samples in feature space also agree well with the feature distributions in the experimental data (compare Fig. S11 to Fig. 3, and Fig. S12 to Fig. 4).

Seeing that the cGAN samples produce appropriate behavior in feature space, we move on to assessing the distributions of the cGAN samples in parameter space in order to answer the questions posed at the beginning of this section. First, we compare the cGAN samples for rTg4510 and their age-matched controls (WT 12 month). Based on KDE plots for each parameter (lower main diagonal of Fig. 14), we see that for 3 of the parameters (g_{NaT}, g_{CaH}, g_H), the WT 12 month and rTg4510 distributions are centered around the same values. However, the distribution for g_{KDR} is shifted to the right in rTg4510 relative to WT 12 month, whereas the distribution of g_{KM} is shifted to the left in rTg4510 relative to WT 12 month. These visual observations were supported by the calculation of Cohen's *d* effect size, defined as the difference between two means divided by the pooled standard deviation of the data (see Supplementary Methods). Based on Cohen (1988)) and Sawilowsky (2009), the effect sizes for g_{CaH}, g_H, and g_{NaT} (0.042, 0.044, and 0.336, respectively) were classified as "very small" or "small", whereas the effect sizes of g_{K_M} and $g_{K_{DR}}$ (0.986 and 1.017)

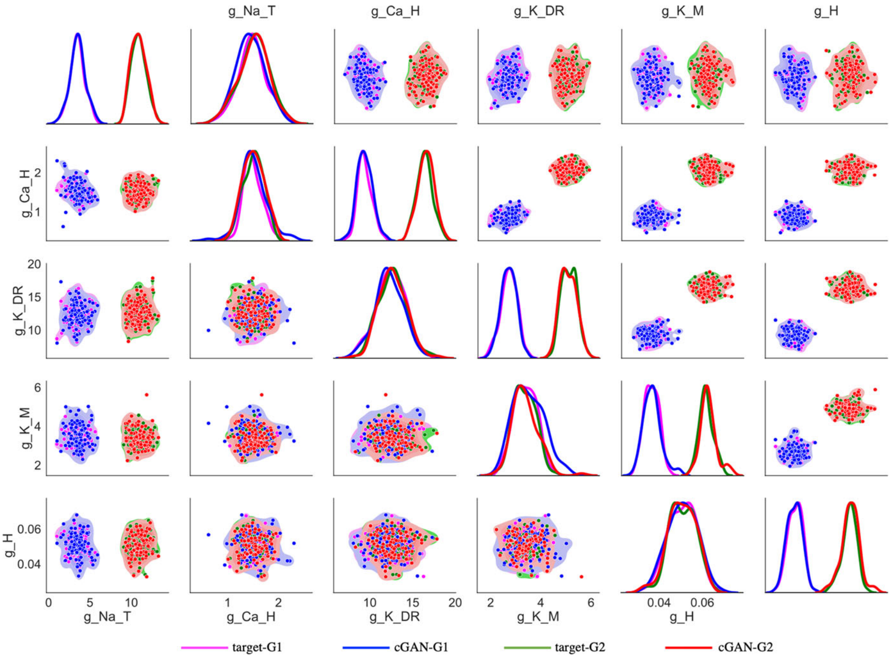

Fig. 12 Performance of cGAN in parameter space on synthetic targets from 2 groups with distinct parameter structures. KDE plots (main diagonals) and scatter plots (lower and upper triangles) for Group 1 (G1) target data (magenta), Group 2 (G2) target data (green), cGAN samples for G1 (blue) and cGAN samples for G2 (red). *Lower main diagonal and lower triangle*—only 1 parameter (g_{NaT}) is distributed differently in the G1 target data than in the G2 target data, and the other 4 parameters have the same distribution in the G1 and G2 target data. *Upper main diagonal and upper triangle*—four parameters (all parameters except g_{NaT}) are distributed differently in the G1 target data than in the G2 target data (Color figure online)

were classified as “very large” (see Supplementary Methods for Cohen’s d effect size classifications). The KDE plots comparing PDAPP and their age-matched controls (WT 24 month) show a similar trend, with the g_{KDR} and g_{KM} distributions shifted to the right and left, respectively, for PDAPP relative to WT (upper main diagonal of Fig. 14). For PDAPP, the distribution of g_{NaT} is also shifted to the left relative to WT. The Cohen’s d effect sizes were “very small" or “small" for g_{CaH} and g_H (0.051 and 0.201, respectively), and “medium” for g_{KM}, g_{NaT}, and g_{KDR} (0.422, 0.536, and 0.617, respectively). From these observations, we hypothesize that for the mouse model of tauopathy, it is the conductances g_{KDR} and g_{KM} that are responsible for the altered excitability properties. For the mouse model of amyloidopathy, we hypothesize that these conductances plus g_{NaT} play a role in the altered excitability.

Having considered the disease effect, we now move on to assessing the age effect. First, we compare the cGAN samples for WT 12 month and WT 24 month. Based on KDE plots for each parameter (lower main diagonal of Fig. 15), we see the most striking differences for g_H, with the distribution for the older mice shifted to the right relative to the distribution for the younger mice. The parameter g_{KM} also shows a rightward shift in the older mice. On the other hand, the distribution for g_{KDR} is

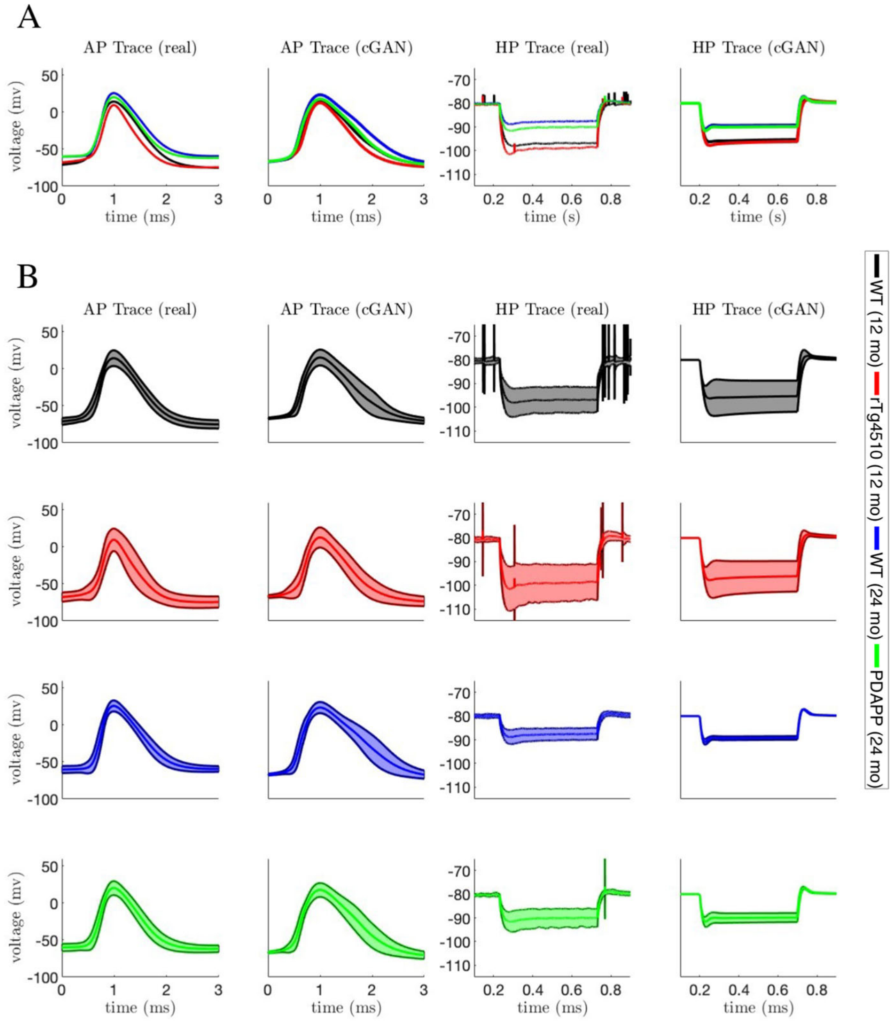

Fig. 13 AP and membrane hyperpolarization traces from cGAN samples with experimental targets. **a** Mean AP and membrane hyperpolarization traces from each experimental data category (1st and 3rd panels) and mean AP and membrane hyperpolarization traces from simulated the mechanistic model with 100 cGAN parameter samples for each cell in each category (2nd and 4th panels). **b** Same as (**a**), but shading shows the mean ± standard deviation for each category (Color figure online)

shifted to the left in the older mice. The Cohen's d effect sizes were "small" for g_{Na_T} and g_{Ca_H}, "large" for $g_{K_{DR}}$ and g_{K_M}, and "very large" for g_H (0.118, 0.373, 0.772, 0.886, and 1.416, respectively). The 12-month WT to 24-month WT comparison is the most appropriate one for assessing an age effect, since the only difference between these two groups of mice is their age. However, for the sake of completeness we also compared the 12-month mutant (rTg4510) to the 24-month mutant (PDAPP). Remarkably, the 3 shifts in the parameter distributions that we observed with age in

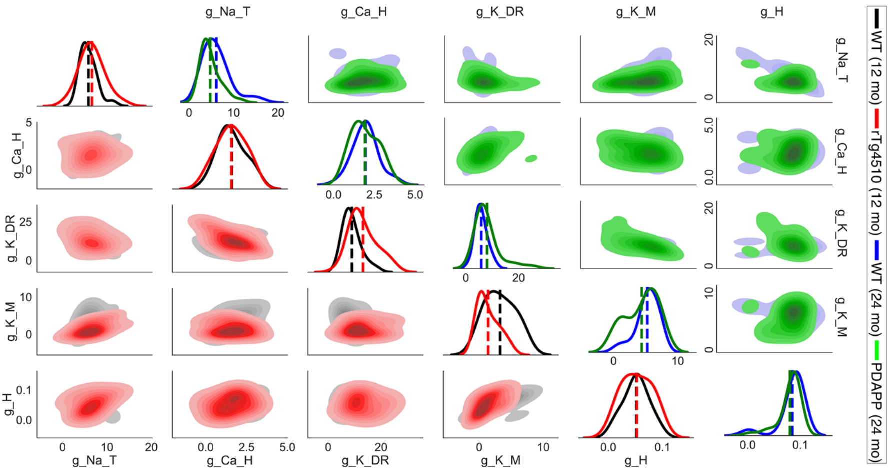

Fig. 14 Disease effect: parameter distributions from cGAN samples with experimental targets. *Lower main diagonal and lower triangle*—KDE and shaded contour plots of cGAN samples for 12-month-old WT (black) and 12-month-old tau mutant (rTg4510, red) mice. *Upper main diagonal and upper triangle*—KDE and shaded contour plots of cGAN samples for 24-month-old (blue) and 24-month-old amyloid beta mutant (PDAPP, green) mice (Color figure online)

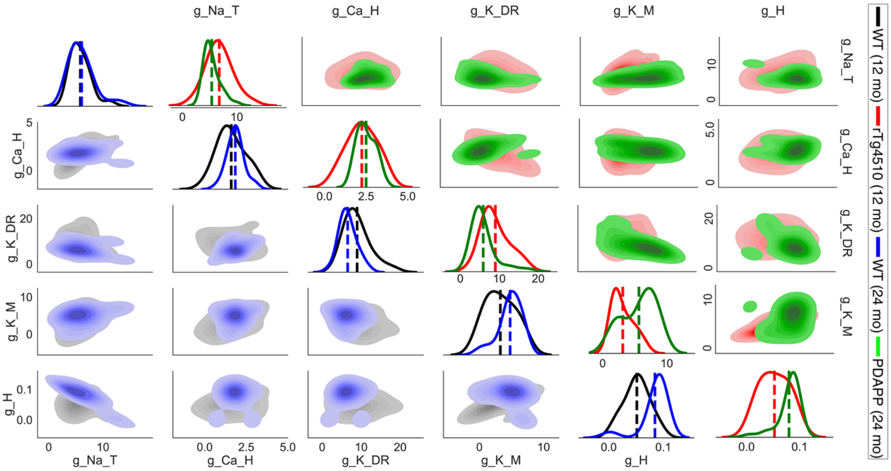

Fig. 15 Age effect: parameter distributions from cGAN samples with experimental targets. *Lower main diagonal and lower triangle*—KDE and shaded contour plots of cGAN samples for 12-month-old (black) and 24-month-old (blue) WT mice. *Upper main diagonal and upper triangle*—KDE and shaded contour plots of cGAN samples for 12-month-old (rTg4510, red) and 24-month-old (PDAPP, green) mutant mice (Color figure online)

the WT mice were all preserved in the mutant mice, despite the fact that the 12 and 24 month mutants have different mutations (tauopathy and amyloidopathy, respectively). Specifically, the g_H and g_{KM} distributions are shifted to the right in the older mutants relative to the younger mutants, while g_{KDR} is shifted to the left in the older mutants (upper main diagonal of Fig. 15). The Cohen's d effect sizes were "small" for g_{Ca_H}, "medium" for g_{Na_T}, "large" for $g_{K_{DR}}$, and "very large" for g_H and g_{K_M} (0.365, 0.715, 0.951, 1.158, and 1.229, respectively). Overall, these results lead us to hypothesize that g_H, g_{K_M}, and $g_{K_{DR}}$ are the 3 main conductances underlying the changes in excitability properties observed with aging.

7 Discussion

The last decade has seen a rise in the application of population-based modeling in the neuronal and cardiac electrophysiology domains (Marder and Taylor 2011; Prinz et al. 2004; Sobie 2009; Lancaster and Sobie 2016; Muszkiewicz et al. 2014; Passini et al. 2016; Sánchez et al. 2014). The development of methods for selecting and producing virtual patient populations that accurately reflect the statistics of clinical populations has also received a lot of attention in fields such as quantitative systems pharmacology (Allen et al. 2016; Cheng et al. 2017; Rieger et al. 2018; Gadkar et al. 2014; Jamei et al. 2009). The process of learning model parameters that capture the heterogeneity of data has been referred to as "population calibration" to indicate that it is an inverse (calibration) problem that respects the variability of a population (Drovandi et al. 2022). Drovandi et al. (2022) provide an overview of likelihood-free Bayesian inference approaches to this class of problems, and introduce a likelihood-free framework that produces uncertainty quantification on the estimated distribution. Nonlinear mixed effects (NLME) models are also a widely used approach for modeling the variability of individuals in a population (Augustin et al. 2023). In this approach, patient demographic data or other medical record information can be used as an input and regressed into the model's parameter space to generate a regression model with a variational inference component and parameterized model parameter distributions. Augustin et al. (2023) review the NLME modeling framework and introduce a computationally efficient method for estimating NLME model parameters.

Here, we have employed a deep hybrid modeling (DeepHM) framework (Parikh et al. 2022; Rumbell et al. 2023) featuring conditional generative adversarial networks (cGANs) that can be categorized as a population of models technique. It is distinct from NLME-based approaches in that cGAN "regresses" model output into distributions of model input. We compared the performance of cGAN and a standard Bayesian inference Markov chain Monte Carlo (MCMC) method (Poole and Raftery 2000) on a parameter inference task with synthetic target data where the ground truth was known. In this test, the parameter distributions inferred by cGAN matched the target distributions better than the distributions inferred by the standard MCMC. We note that our MCMC implementation uses the simplest form of the Metropolis-Hastings algorithm for sampling, and that better performance might be obtained using sequential Monte Carlo or other more complex MCMC sampling algorithms. Regarding the accuracy of the inferred parameter distributions, we are not claiming that cGAN would outperform all versions of MCMC algorithms on this problem, nor that cGAN would necessarily outperform our standard MCMC implementation on all problems. However, cGAN does have a computational advantage in that the trained cGAN can be used for parameter inference on other target datasets without retraining, whereas MCMC-based methods would need to be retrained.

In our validation tests on synthetic data, cGAN showed it is capable of producing a population of models that captures the type of variability that is often present in biological data. Since the cGAN was able to accurately detect a variety of complex differences in the distributions of parameters across 2 groups of synthetic target data, we employed the cGAN to infer the parameter distributions across 4 groups of experimental target data (WT and mutant mice at 2 different ages). From these distributions, we drew conclusions about the biophysical mechanisms (i.e. the ionic conductances) underlying the differences in the observed excitability properties of WT versus mutant and younger versus older mice. These results illustrate the value of mapping experimental data back to the parameter space of a mechanistic model. In future work, the predictions we made about the role of certain conductances in Alzheimer's disease and aging phenotypes can be tested experimentally.

The cGAN was able to recover the input distribution almost exactly in most of our tests on synthetic data. This is perhaps surprising given that stochastic inverse problems are often ill-posed in that there are an infinite number of input distributions that could match the output distribution. In our framework, it is the prior that selects from among the possible input distributions that match the output distribution. In Rumbell et al. (2023), we demonstrated that with cGANs and related techniques (regularized GANs and rejection sampling), changing the prior results in a different distribution being sampled during inference.

We have focused on starting with a uniform prior and thereby maximize the entropy of the parameter distribution given the target, thus ensuring that the cGAN will, for example, not fail to sample certain modes of the coherent parameter distribution capable of producing the target. For certain synthetic problems, we have observed that the ground truth parameter distribution used to synthesize target data may have fewer modes than those discovered by cGAN. These cases exemplify the real-world scenario where more experimentation is needed to rule out these extra modes. Deep learning models today can reach up to a trillion parameters, and therefore we have focused on avoiding the main methodological risk: introducing inductive biases into our deep learning methods that result in undersampling the parameter distribution coherent to the target. In summary, we argue that deep learning generators are the correct approach to addressing these ill-posed problems in model parameter fitting.

Since DeepHM can produce populations of cell models that accurately reflect the heterogeneous responses of real cells, this framework could prove useful in virtual drug design applications. This future direction is inspired by recent work where a population of models approach was used to identify a set of ion channel drug targets that optimally convert Huntington's disease cellular phenotypes to healthy phenotypes simultaneously across multiple measures (Allam et al. 2011). Additionally, our work here has focused on characterizing the parameter distributions of populations of cells. Future work could use cGAN to explore the relationship between parameter distributions at the level of individual cells and the aggregated population distribution. Similar approaches have been used for uncertainty quantification in cardiac electrophysiology (Groenendaal et al. 2015; Pathmanathan et al. 2015; Lei et al. 2019).

In this paper, we used two different data types for optimization: full voltage traces when fitting the model with differential evolution, and summary statistics (in the form of electrophysiological features) when training the cGAN. Lei et al. (2023)

demonstrated that using full traces rather than summary statistics may have advantages when calibrating cardiac electrophysiology models and leads to narrower parameter posterior distributions. We used a feature-based approach for training the cGAN (and when presenting target data to the cGAN) to make it less likely that the cGAN would converge to a single minimum. When defining and selecting features, we tried to avoid including redundant features, as this would increase the dimensionality of the search space unnecessarily. A more systematic process for selecting the optimal set of features, perhaps involving an assessment of whether certain features apply stronger constraints on the inferred parameters than others, would be an improvement to our *ad hoc* feature selection approach. Also we note that although the model where V_{mNaT} was optimized with differential evolution (DE-MG-Vmnat) produced a better fit to the experimental data than the model without V_{mNaT} optimized (DE-MG) for most features, there are some features (such as AP V MP ROR) where the DE-MG model fit better (see Fig. 3). This type of tradeoff, where some features worsen as others improve, is indicative of model discrepancy/error. Another limitation of our study is that we only performed inference on the maximal conductance parameters and left the kinetic parameters (τ_x and V_x, k_x in the x_∞ function) at fixed values. There can be strong and complex interactions between maximal conductances and kinetics, and it has been shown that misspecified kinetics can lead to erroneous estimates of maximal conductance parameters (Lei et al. 2020). Furthermore, we have not attempted to model the contribution of the experimental leak current (I_{leak}) that can occur due to imperfect seal, which is distinct from the physiological I_L that is included in our model. Clark et al. (2022) have shown that explicit inclusion of I_{leak} can improve the descriptive ability of electrophysiological models.

The stochastic inverse problem (SIP) considered in this paper involves experimental data from multiple individual cells across a population. Our method utilized recent advances in deep learning to generate model parameter sets that produced a population of deterministic mechanistic models with outputs that are consistent with the experimental population data. A distinct but related problem is simulation-based inference (SBI), where experimental data are acquired from a single individual and a stochastic mechanistic model is used to infer the set of model parameters most likely to have generated the data distribution observed from the individual. Deep learning methods such as neural density estimation with normalizing flows have been used in SBI problems (Gonçalves et al. 2020; Lueckmann et al. 2019). Similar to our approach, conditional GANs have recently been used to solve a physics-based inverse problem and infer a thermal conductivity field given a noisy temperature field (Ray et al. 2021). Deep neural networks and adversarial training have also been used to solve inverse problems for stochastic models of financial option pricing (Xu and Darve 2021).

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s11538-024-01273-5.

Acknowledgements This work was partially supported by NSF DMS grants 1555237 and 2152115 to COD. KCAW was graciously supported by the EPSRC New Horizons grant (EP/V048716/1) and the EPSRC Hub for Quantitative Modeling in Healthcare (EP/T017856/1). The PDAPP data were collected under the Medical Research Council grant G1100623 (FT), and the rTg4510 data under the Alzheimer's

Society Junior Fellowship grant AS-JF-14-007 (FT). KCAW and FT received support from the the Wellcome Trust ISSF2 Centre grant WT105618MA.

Data Availability Statement The datasets generated during and/or analysed during the current study are available from the corresponding author on reasonable request. Code associated with this study is available at https://github.com/IBM/rgan-demo-pytorch/.

References

Allam SL, Rumbell TH, Hoang-Trong T, Parikh J, Kozloski JR (2021) Neuronal population models reveal specific linear conductance controllers sufficient to rescue preclinical disease phenotypes. iScience 24(11):103279

Allen R, Rieger TR, Musante CJ (2016) Efficient generation and selection of virtual populations in quantitative systems pharmacology models. CPT: Pharmacomet Syst Pharmacol 5(3):140–146

Augustin D, Lambert B, Wang K, Walz A-C, Robinson M, Gavaghan D (2023) Filter inference: A scalable nonlinear mixed effects inference approach for snapshot time series data. PLoS Comput Biol 19(5):1011135

Booth CA, Witton J, Nowacki J, Tsaneva-Atanasova K, Jones MW, Randall AD, Brown JT (2016) Altered intrinsic pyramidal neuron properties and pathway-specific synaptic dysfunction underlie aberrant hippocampal network function in a mouse model of tauopathy. J Neurosci 36(2):350–363

Britton OJ, Bueno-Orovio A, Van Ammel K, Lu HR, Towart R, Gallacher DJ, Rodriguez B (2013) Experimentally calibrated population of models predicts and explains intersubject variability in cardiac cellular electrophysiology. Proc Natl Acad Sci 110(23):2098–2105

Butler T, Jakeman J, Wildey T (2018) Combining push-forward measures and Bayes' rule to construct consistent solutions to stochastic inverse problems. SIAM J Sci Comput 40(2):984–1011

Butte S, Wang H, Xian M, Vakanski A (2022) Sharp-GAN: Sharpness loss regularized GAN for histopathology image synthesis. In: 2022 IEEE 19th International symposium on biomedical imaging (ISBI), pp 1–5. IEEE

Chen M (2022) Vanishing gradient problem in training neural networks. Honours thesis, The Australian National University

Cheng Y, Thalhauser CJ, Smithline S, Pagidala J, Miladinov M, Vezina HE, Gupta M, Leil TA, Schmidt BJ (2017) QSP toolbox: computational implementation of integrated workflow components for deploying multi-scale mechanistic models. AAPS J 19(4):1002–1016

Clark AP, Clerx M, Wei S, Lei CL, de Boer T, Mirams GR, Christini DJ, Krogh-Madsen T (2022) Leak current, even with gigaohm seals, can cause misinterpretation of stem cell-derived cardiomyocyte action potential recordings. bioRxiv, 2022–10

Cohen J (1988) Statistical power analysis for the behavioral sciences. Lawrence Erlbaum Associates, New York

Cranmer K, Brehmer J, Louppe G (2020) The frontier of simulation-based inference. Proc Natl Acad Sci 117(48):30055–30062

Drovandi C, Lawson B, Jenner AL, Browning AP (2022) Population calibration using likelihood-free bayesian inference. arXiv preprint arXiv:2202.01962

Gadkar K, Budha N, Baruch A, Davis J, Fielder P, Ramanujan S (2014) A mechanistic systems pharmacology model for prediction of LDL cholesterol lowering by PCSK9 antagonism in human dyslipidemic populations. CPT: Pharmacomet Syst Pharmacol 3(11):1–9

Gonçalves PJ, Lueckmann J-M, Deistler M, Nonnenmacher M, Öcal K, Bassetto G, Chintaluri C, Podlaski WF, Haddad SA, Vogels TP et al (2020) Training deep neural density estimators to identify mechanistic models of neural dynamics. elife 9:56261

Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y (2020) Generative adversarial networks. Commun ACM 63(11):139–144

Groenendaal W, Ortega FA, Kherlopian AR, Zygmunt AC, Krogh-Madsen T, Christini DJ (2015) Cell-specific cardiac electrophysiology models. PLoS Comput Biol 11(4):1004242

Gulrajani I, Ahmed F, Arjovsky M, Dumoulin V, Courville AC (2017) Improved training of Wasserstein GANs. Adv Neural Inf Process Syst 30

Hardy J (2009) The amyloid hypothesis for Alzheimer's disease: a critical reappraisal. J Neurochem 110(4):1129–1134

Hitawala S (2018) Comparative study on generative adversarial networks. arXiv preprint arXiv:1801.04271

Hodgkin AL, Huxley AF (1952) A quantitative description of membrane current and its application to conduction and excitation in nerve. J Physiol 117(4):500

Jamei M, Dickinson GL, Rostami-Hodjegan A (2009) A framework for assessing inter-individual variability in pharmacokinetics using virtual human populations and integrating general knowledge of physical chemistry, biology, anatomy, physiology and genetics: a tale of 'bottom-up' vs 'top-down' recognition of covariates. Drug Metab Pharmacokinet 24(1):53–75

Kass RE, Amari S-I, Arai K, Brown EN, Diekman CO, Diesmann M, Doiron B, Eden UT, Fairhall AL, Fiddyment GM et al (2018) Computational neuroscience: mathematical and statistical perspectives. Annu Rev Stat Appl 5:183

Kingma DP, Ba J (2014) Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980

Lancaster MC, Sobie E (2016) Improved prediction of drug-induced torsades de pointes through simulations of dynamics and machine learning algorithms. Clin Pharmacol Ther 100(4):371–379

Lawson BA, Drovandi CC, Cusimano N, Burrage P, Rodriguez B, Burrage K (2018) Unlocking data sets by calibrating populations of models to data density: a study in atrial electrophysiology. Sci Adv 4(1):1701676

Lei CL, Clerx M, Gavaghan DJ, Polonchuk L, Mirams GR, Wang K (2019) Rapid characterization of hERG channel kinetics I: using an automated high-throughput system. Biophys J 117(12):2438–2454

Lei CL, Ghosh S, Whittaker DG, Aboelkassem Y, Beattie KA, Cantwell CD, Delhaas T, Houston C, Novaes GM, Panfilov AV et al (2020) Considering discrepancy when calibrating a mechanistic electrophysiology model. Phil Trans R Soc A 378(2173):20190349

Lei CL, Clerx M, Gavaghan DJ, Mirams GR (2023) Model-driven optimal experimental design for calibrating cardiac electrophysiology models. Comput Methods Programs Biomed 240:107690

Lueckmann J, Bassetto G, Karaletsos T, Macke J (2019) Likelihood-free inference with emulator networks. arxiv e-prints. arXiv preprint arXiv:1805.09294

Marder E, Taylor AL (2011) Multiple models to capture the variability in biological neurons and networks. Nat Neurosci 14(2):133–138

Marino S, Hogue IB, Ray CJ, Kirschner DE (2008) A methodology for performing global uncertainty and sensitivity analysis in systems biology. J Theor Biol 254(1):178–196

Metz L, Poole B, Pfau D, Sohl-Dickstein J (2016) Unrolled generative adversarial networks. arXiv preprint arXiv:1611.02163

Mirza M, Osindero S (2014) Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784

Molano-Mazon M, Onken A, Piasini E, Panzeri S (2018) Synthesizing realistic neural population activity patterns using generative adversarial networks. arXiv Preprint arXiv:1803.00338

Muszkiewicz A, Bueno-Orovio A, Liu X, Casadei B, Rodriguez B (2014) Constructing human atrial electrophysiological models mimicking a patient-specific cell group. In: Computing in cardiology 2014, pp 761–764

Nowacki J, Osinga HM, Brown JT, Randall AD, Tsaneva-Atanasova K (2011) A unified model of ca1/3 pyramidal cells: an investigation into excitability. Prog Biophys Mol Biol 105(1–2):34–48

Okoyo C, Onyango N, Orowe I, Mwandawiro C, Medley G (2022) Sensitivity analysis of a transmission interruption model for the soil-transmitted helminth infections in Kenya. Front Pub Health 10:841883

Panahi MR, Abrevaya G, Gagnon-Audet J-C, Voleti V, Rish I, Dumas G (2021) Generative models of brain dynamics–a review. arXiv Preprint arXiv:2112.12147

Parikh J, Rumbell T, Butova X, Myachina T, Acero JC, Khamzin S, Solovyova O, Kozloski J, Khokhlova A, Gurev V (2022) Generative adversarial networks for construction of virtual populations of mechanistic models: simulations to study omecamtiv mecarbil action. J Pharmacokinet Pharmacodyn 49(1):51–64

Passini E, Mincholé A, Coppini R, Cerbai E, Rodriguez B, Severi S, Bueno-Orovio A (2016) Mechanisms of pro-arrhythmic abnormalities in ventricular repolarisation and anti-arrhythmic therapies in human hypertrophic cardiomyopathy. J Mol Cell Cardiol 96:72–81

Pathmanathan P, Shotwell MS, Gavaghan DJ, Cordeiro JM, Gray RA (2015) Uncertainty quantification of fast sodium current steady-state inactivation for multi-scale models of cardiac electrophysiology. Prog Biophys Mol Biol 117(1):4–18

Pilosov M, del-Castillo-Negrete C, Yu Yen T, Butler T, Dawson C (2023) Parameter estimation with maximal updated densities. Comput Methods Appl Mech Eng. 407:115906

Poole D, Raftery AE (2000) Inference for deterministic simulation models: The Bayesian melding approach. J Am Stat Assoc 95(452):1244–1255

Price K, Storn RM, Lampinen JA (2006) Differential evolution: a practical approach to global optimization. Springer, 53851, Lappeenranta

Prinz AA, Bucher D, Marder E (2004) Similar network activity from disparate circuit parameters. Nat Neurosci 7(12):1345–1352

Ramesh P, Lueckmann J-M, Boelts J, Tejero-Cantero À, Greenberg DS, GonÀSalves PJ, Macke JH (2022) GATSBI: generative adversarial training for simulation-based inference. arXiv preprint arXiv:2203.06481

Ray, D., Patel, D.V., Ramaswamy, H., Oberai, A.: Efficient posterior inference and generalization in physics-based Bayesian inference with conditional GANs. In: NeurIPS 2021 workshop on deep learning and inverse problems

Rieger TR, Allen RJ, Bystricky L, Chen Y, Colopy GW, Cui Y, Gonzalez A, Liu Y, White R, Everett R et al (2018) Improving the generation and selection of virtual populations in quantitative systems pharmacology models. Prog Biophys Mol Biol 139:15–22

Roberts DA, Yaida S, Hanin B (2022) The principles of deep learning theory. Cambridge University Press, Cambridge

Rosenfeld B, Simeone O, Rajendran B (2022) Spiking generative adversarial networks with a neural network discriminator: local training, Bayesian models, and continual meta-learning. IEEE Trans Comput 71(11):2778–2791

Rumbell T, Parikh J, Kozloski J, Gurev V (2023) Novel and flexible parameter estimation methods for data-consistent inversion in mechanistic modelling. R Soc Open Sci 10:230668

Sánchez C, Bueno-Orovio A, Wettwer E, Loose S, Simon J, Ravens U, Pueyo E, Rodriguez B (2014) Inter-subject variability in human atrial action potential in sinus rhythm versus chronic atrial fibrillation. PLoS ONE 9(8):105897

Sawilowsky SS (2009) New effect size rules of thumb. J Mod Appl Stat Methods 8(2):26

Sobie EA (2009) Parameter sensitivity analysis in electrophysiological models using multivariable regression. Biophys J 96(4):1264–1274

Spillantini MG, Goedert M (2013) Tau pathology and neurodegeneration. Lancet Neurol 12(6):609–622

Storn R, Price K (1997) Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces. J Global Optim 11(4):341–359

Tamagnini F, Novelia J, Kerrigan TL, Brown JT, Tsaneva-Atanasova K, Randall AD (2015a) Altered intrinsic excitability of hippocampal CA1 pyramidal neurons in aged PDAPP mice. Front Cell Neurosci 9:372

Tamagnini F, Scullion S, Brown JT, Randall AD (2015b) Intrinsic excitability changes induced by acute treatment of hippocampal CA1 pyramidal neurons with exogenous amyloid β peptide. Hippocampus 25(7):786–797

Tamagnini F, Walsh DA, Brown JT, Bondulich MK, Hanger DP, Randall AD (2017) Hippocampal neurophysiology is modified by a disease-associated c-terminal fragment of tau protein. Neurobiol Aging 60:44–56

Vitale P, Salgueiro-Pereira AR, Lupascu CA, Willem M, Migliore R, Migliore M, Marie H (2021) Analysis of age-dependent alterations in excitability properties of CA1 pyramidal neurons in an appps1 model of alzheimer's disease. Front Aging Neurosci 13:668948

Xu K, Darve E (2021) Solving inverse problems in stochastic models using deep neural networks and adversarial training. Comput Methods Appl Mech Eng 384:113976

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.