Diffusion Tempering Improves Parameter Estimation with Probabilistic Integrators for Ordinary Differential Equations

Jonas Beck [1 2] **Nathanael Bosch** [2] **Michael Deistler** [2] **Kyra L. Kadhim** [1 2] **Jakob H. Macke** [2] **Philipp Hennig** [2] **Philipp Berens** [1 2]

Abstract

Ordinary differential equations (ODEs) are widely used to describe dynamical systems in science, but identifying parameters that explain experimental measurements is challenging. In particular, although ODEs are differentiable and would allow for gradient-based parameter optimization, the nonlinear dynamics of ODEs often lead to many local minima and extreme sensitivity to initial conditions. We therefore propose diffusion tempering, a novel regularization technique for probabilistic numerical methods which improves convergence of gradient-based parameter optimization in ODEs. By iteratively reducing a noise parameter of the probabilistic integrator, the proposed method converges more reliably to the true parameters. We demonstrate that our method is effective for dynamical systems of different complexity and show that it obtains reliable parameter estimates for a Hodgkin–Huxley model with a practically relevant number of parameters.

1. Introduction

Ordinary differential equations (ODEs) are ubiquitous across science, as they often provide accurate models for physical processes and mechanisms underlying dynamical systems. ODEs can, for example, be used to model the motion of a pendulum, the dynamics of predator and prey populations (Hethcote, 2000) or the action potentials of neurons (Hodgkin & Huxley, 1952). Numerical algorithms for the (approximate) solution of ODEs are well established (Hairer et al., 1987), but identifying parameters of the ODE such that it matches observations can be challenging (Prinz et al., 2003; Kravtsov, 2020). To overcome this, multiple methods have been developed, such as grid (Prinz et al., 2003) or random searches (Taylor et al., 2009), simulated annealing (Kirkpatrick et al., 1983), genetic algorithms (Ben-Shalom et al., 2012), Bayesian methods (e.g., approximate Bayesian computation (Marjoram et al., 2003), Markov-Chain Monte-Carlo (Neal, 1993), or simulation-based inference (Cranmer et al., 2020)). But, these methods can often require large simulation budgets (Cranmer et al., 2020).

By incorporating gradient information, gradient descent provides the potential to be vastly more simulation efficient and, as demonstrated by Neural ODEs (NODEs), has the potential to scale to millions of parameters (Chen et al., 2018). Unfortunately, for many real-world ODEs, gradient-based optimization is challenging: ODEs often have highly nonlinear dynamics, several local minima, and are sensitive to initial conditions (Cao et al., 2011; Dass et al., 2017). In fact, even for models with few parameters, gradient-based optimization can get stuck in local minima, leading to poor performance compared to competing gradient-free methods such as genetic algorithms (Hazelden et al., 2023).

Probabilistic numerical integrators have also been proposed for parameter inference in ODEs (Tronarp et al., 2022). Unlike 'traditional' non-probabilistic ODE solvers which usually return only a single (potentially very crude) ODE solution, probabilistic numerical methods return a posterior distribution over the ODE solution, accounting for numerical uncertainty (Hennig et al., 2022). Parameters can then be estimated by maximizing their likelihood, which thanks to automatic differentiation is amenable to first order methods. This approach has been termed "Physics-Enhanced Regression for Initial Value Problems", or Fenrir for short.

Here, we develop *diffusion tempering* for probabilistic integrators to improve gradient-based parameter inference in ODEs. The technique is based on previous observations that Fenrir can effectively 'smooth out' the loss surface of ODEs (Tronarp et al., 2022). We therefore propose to solve consecutive optimization problems using the Fenrir likelihood. We start with a very smooth loss surface, which yields poor fits to data, but lets the optimizer avoid local minima. By successively solving less and less smooth problems informed by

[1]Hertie Institute for AI in Brain Health, University of Tübingen, Tübingen, Germany [2]AI Center, University of Tübingen, Tübingen, Germany. Correspondence to: Jonas Beck <jonas.beck@uni-tuebingen.de>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

previous parameter estimates, we more reliably converge in the global optimum. This enables gradient-based maximum likelihood estimation for models with a practically relevant number of parameters such as Pospischil et al. (2008).

We first demonstrate that our method is robust to local minima for a simple pendulum. We then show that it produces more reliable parameter estimates than both classical least-squares regression and the original Fenrir method by Tronarp et al. (2022) for several models of growing complexity, even in regimes in which gradient-based parameter inference is challenging.

2. Parameter inference in ODEs

Consider an initial value problem (IVP), given by an ODE

$$\dot{y}_\theta(t) = f_\theta\big(y_\theta(t), t\big)\,, \qquad t \in [0, T], \tag{1}$$

with vector field f_θ with parameters θ, and initial value $y_\theta(0) = y_0$. In this paper, we are concerned with estimating the true ODE parameters θ from a set of noisy observations of the solution $y_\theta(t)$, of the form

$$u_i = u(t_i) = H y_\theta(t_i) + \epsilon_i, \qquad i = 1, \dots, N, \tag{2}$$

where H is a measurement matrix and $\epsilon_i \sim \mathcal{N}(0, R)$ is Gaussian noise with covariance R. We denote the set of observations by $\mathcal{D} = \{u_i\}_{i=1}^N$, and the set of observation times by $\mathbb{T}_\mathcal{D} = \{t_i\}_{i=1}^N$.

For a given parameter θ, the true ODE solution $y_\theta(t)$ is uniquely defined and thus the true marginal likelihood is

$$\mathcal{M}(\theta) = \mathcal{L}_\mathcal{D}(y_\theta) = \prod_{i=1}^N \mathcal{N}(u_i; H y_\theta(t_i), R), \tag{3}$$

where $\mathcal{L}_\mathcal{D}$ denotes the likelihood functional. Then, the parameter θ can be estimated from the data $\mathcal{D}$ by maximizing the marginal likelihood $\mathcal{M}(\theta)$, that is $\hat{\theta}_{\text{MLE}} = \arg\max_\theta \mathcal{M}(\theta)$. Unfortunately, the true solution y_θ is not generally known and cannot be computed analytically, and thus the true marginal likelihood is intractable.

In the following, we discuss two approaches to make inference tractable by approximating the marginal likelihood. First, note that we can rewrite $\mathcal{M}(\theta)$ as an integral

$$\mathcal{M}(\theta) = \int \mathcal{L}_\mathcal{D}(y) \delta(y - y_\theta)\, \mathrm{d}y. \tag{4}$$

This clarifies that the intractable part of the marginal likelihood is the Dirac measure $\delta(y - y_\theta)$. Thus, a natural approach to approximate the marginal likelihood is to replace the true solution distribution by a suitable approximation.

2.1. Classic Numerical Integration

A standard approach to parameter inference in IVPs approximates the true IVP solution $y_\theta(t)$ with a numerical solution $\hat{y}_\theta(t)$, obtained from a classic numerical IVP solver such as the well-known Runge–Kutta (RK) method (Hairer et al., 1987). The marginal likelihood then becomes

$$\mathcal{M}(\theta) \approx \widehat{\mathcal{M}}_{\text{RK}}(\theta) := \int \mathcal{L}_\mathcal{D}(y) \delta(y - \hat{y}_\theta)\, \mathrm{d}y, \tag{5a}$$

$$= \prod_{i=1}^N \mathcal{N}(u_i; H \hat{y}_\theta(t_i), R). \tag{5b}$$

Maximizing $\widehat{\mathcal{M}}_{\text{RK}}(\theta)$ is equivalent to minimizing the mean squared error

$$L_{\text{RK}}(\theta) = \frac{1}{N} \sum_{i=1}^N \left\| H \hat{y}_\theta(t_i) - u_i \right\|_2^2. \tag{6}$$

This approach is also known as non-linear least-squares regression (Bard, 1974).

2.2. Probabilistic Numerical Integration

Probabilistic numerical methods reformulate numerical problems as probabilistic inference (Hennig et al., 2022). In the context of IVPs, probabilistic numerical ODE solvers make the time-discretization of numerical methods explicitly part of their problem statement, and aim to compute a posterior distribution of the form

$$p\left(y_\theta(t) \mid y_\theta(0) = y_0, \{\dot{y}_\theta(t) = f_\theta\big(y_\theta(t), t\big)\}_{t \in \mathbb{T}_{\text{PN}}}\right), \tag{7}$$

where $\mathbb{T}_{\text{PN}}$ is the chosen time-discretization. Assuming fixed f, y_0, and $\mathbb{T}_{\text{PN}}$, we denote this posterior more compactly with $p_{\text{PN}}(y(t) \mid \theta)$. We call this object, and approximations thereof, a *probabilistic numerical ODE solution*

This object then presents itself as a natural candidate to approximate the true solution distribution $\delta(y - y_\theta)$. We obtain the PN-approximated marginal likelihood

$$\widehat{\mathcal{M}}_{\text{PN}}(\theta) = \int \mathcal{L}_\mathcal{D}(y)\, p_{\text{PN}}(y(t) \mid \theta)\, \mathrm{d}y. \tag{8}$$

The remaining question is how to compute this quantity. In the following, we consider probabilistic ODE solvers based on Bayesian filtering and smoothing, which have been shown to be a particularly efficient class of methods for probabilistic numerical simulation (Tronarp et al., 2021; 2019; Kersting et al., 2020; Schober et al., 2019), and we review the PN-approximated marginal likelihood by Tronarp et al. (2022) which builds the base of our proposed approach.

2.2.1. Probabilistic Numerical IVP Solvers

Filtering-based probabilistic numerical ODE solvers formulate the probabilistic numerical ODE solution (Equa-

tion (7)) as a Bayesian state estimation problem, described by a Gauss–Markov prior given by

$$x(0) \sim \mathcal{N}(\mu_0, \Sigma_0)\,, \tag{9a}$$

$$x(t_n) \mid x(t_{n-1}) \sim \mathcal{N}\Big(\Phi_n x(t_{n-1}), \kappa^2 Q_n\Big)\,, \tag{9b}$$

$$y^{(m)}(t_n) = E_m^\intercal x(t_n), \quad m = 0, 1, \ldots, q, \tag{9c}$$

and a likelihood and data model, or *information operator*, z (Tronarp et al., 2021; Cockayne et al., 2019), of the form

$$z(t_n) \mid x(t_n) \sim \delta\Big(E_1 x(t_n) - f_\theta\big(E_0 x(t_n), t\big)\Big)\,, \tag{9d}$$

$$z(t_n) := 0, \tag{9e}$$

which maps solutions of the initial value problem to the zero function on the chosen grid $\mathbb{T}_\mathcal{D}$.

Here, the state $x(t)$ models the solution $y(t)$ of the IVP together with its q first derivatives, and in the following we consider q-times integrated Wiener process priors for which the transition matrices Φ_n and Q_n are known in closed form (Kersting et al., 2020). E_m are selection matrices for the mth derivative. The initial mean μ_0 is chosen to match the initial condition of the given IVP exactly, and the initial covariance Σ_0 is set to zero (Krämer & Hennig, 2020). Finally, κ is a prior hyperparameter which controls the uncertainty of the prior distribution, known as the *diffusion*.

Equation (9) is a well known problem in Bayesian filtering and smoothing, and its solution can be approximated efficiently with extended Kalman filtering and smoothing (Särkkä, 2013). For a given ODE parameter θ and diffusion hyperparameter κ, we obtain a Gauss–Markov posterior distribution $\hat{p}_{\mathrm{PN}}(y(t) \mid \theta, \kappa)$.

2.2.2. PN-APPROXIMATED MARGINAL LIKELIHOOD

By inserting the PN posterior $\hat{p}_{\mathrm{PN}}(y(t) \mid \theta, \kappa)$, into the marginal likelihood, we obtain the marginal likelihood model by Tronarp et al. (2022) of the form

$$\widehat{\mathcal{M}}_{\mathrm{FN}}(\theta, \kappa) = \int \mathcal{L}_\mathcal{D}(y)\, \hat{p}_{\mathrm{PN}}(y(t) \mid \theta, \kappa)\, \mathrm{d}y. \tag{10}$$

This quantity can again be computed efficiently with Kalman filtering: Since the posterior distribution $\hat{p}_{\mathrm{PN}}$ has known backward transition densities of the form (Tronarp et al., 2022, Proposition 3.1)

$$\hat{p}_{\mathrm{PN}}\big(x(t_{n-1}) \mid x(t_n)\big) = \mathcal{N}\big(G_n^\theta x(t_n) + \zeta_n^\theta, \kappa^2 P_n^\theta\big), \tag{11}$$

$\widehat{\mathcal{M}}_{\mathrm{FN}}(\theta, \kappa)$ can be computed by running a Kalman filter backwards in time on the state-space model

$$x(t_N) \sim \mathcal{N}\big(x(t_N); \xi_\theta(t_N), \kappa^2 \Lambda_\theta(t_N)\big), \tag{12a}$$

$$x(t_{n-1}) \mid x(t_n) \sim \mathcal{N}\big(G_n^\theta x(t_n) + \zeta_n^\theta, \kappa^2 P_n^\theta\big), \tag{12b}$$

$$u(t_n) \mid x(t_n) \sim \mathcal{N}\big(H E_0 x(t_n), R_\theta\big), \quad t_n \in \mathbb{T}_\mathrm{D}, \tag{12c}$$

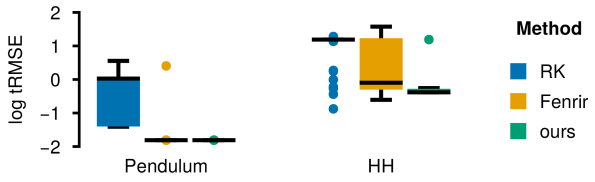

Figure 1. For a pendulum model, Fenrir produces parameter estimates with much lower trajectory mean squared errors (Definition A3.1) than RK least-squares regression. However, for the more complex HH model, the tRMSE is very similar for both RK and Fenrir. Our proposed method is able to produce much better estimates for both problems.

via the prediction error decomposition (Schweppe, 1965). The quantities $G_n^\theta, \zeta_n^\theta, P_n^\theta$ can all be computed during the forward-pass of the probabilistic ODE solver; for the full details refer to Tronarp et al. (2022).

Finally, Tronarp et al. (2022) propose to jointly optimize $\widehat{\mathcal{M}}_{\mathrm{FN}}(\theta, \kappa)$ for both the ODE parameters of interest θ and the diffusion hyperparameter κ, that is

$$\hat{\theta}, \hat{\kappa} = \arg\max_{\theta, \kappa} \widehat{\mathcal{M}}_{\mathrm{FN}}(\theta, \kappa)\,. \tag{13}$$

Then $\hat{\theta}$ is returned as the maximum likelihood estimate. We refer to this ODE parameter inference method as Fenrir.

2.3. Shortcomings

In order to ensure that an optimizer has converged at the global as opposed to a local optimum, in practice, the optimization has to be run multiple times with different initial conditions. A reliable optimizer is therefore highly desirable since a higher reliability also means less restarts to obtain a good parameter estimate. This in turn can make optimizing more complex models feasible, since we can afford many more restarts at the same cost.

Tronarp et al. (2022) show that learning θ and κ jointly leads to more reliable parameter estimates for a pendulum compared to RK least-squares regression (Figure 1) for which the optimization often converges in local minima, i.e. the constant zero function. However, a pendulum is a fairly low dimensional problem. When we evaluated Fenrir for more complex ODEs such as the HH model, we found that this gain in reliability was vastly smaller in systems with more challenging likelihood functions. In these cases, many restarts are needed to reach the global optimum, which makes Fenrir equally unfeasible as RK-based least squares for gradient descent-based parameter estimation in HH models with a practically relevant number of parameters.

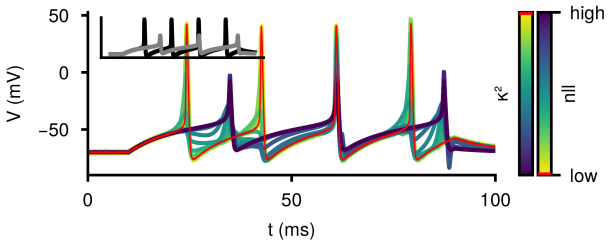

Figure 2. PN posterior means for a range of κ for a HH model. The parameters that generated the observation (inset, black) are different from the parameters of the ODE (inset, grey). For low κ the mean closely adheres to the ODE solution, while for high values, it fits the observation much better. The maximum likelihood κ is highlighted in red.

Algorithm 1 Diffusion tempering

Input: initial parameter θ_0, tempering schedule $\mathcal{T}$, number of iterations m, Optimizer OPT, Fenrir likelihood $\widehat{\mathcal{M}}(\theta, \kappa)$
for $i = 0$ **to** $m - 1$ **do**
  $\kappa_i = \mathcal{T}(i)$
  $\theta_{i+1} = \mathrm{OPT}\Big(\widehat{\mathcal{M}}_{\mathrm{PN}}, (\theta_i, \kappa_i)\Big)$
end for
$\theta_{est} = \theta_m$
Return: θ_{est}

3. Method

In the following, we attribute the observed shortcomings of Fenrir to the way the diffusion hyperparameter κ was treated and we propose an alternative method which enables effective parameter inference for more complicated loss functions and higher dimensional parameter spaces (Figure 1).

3.1. Diffusion as Regularization

To gain a better understanding of the mechanisms that lead to better optimization performance in Fenrir, we studied the effect of the diffusion hyperparameter in more detail.

We noticed that for a given set of model parameters θ, the loss was lower for high values of κ than for low values (Figure 2). Furthermore, when κ was set high, the mean of the PN posterior interpolated the observed data points, while if set low, it approximated the IVP solution much better. This implies for this particular parameter set, that the maximum likelihood estimate of the PN posterior (Equation (13)) "favors" a good interpolation of the data over a good approximation of the IVP solution. This is an issue for the success of parameter estimation, since we are trying to identify an IVP that reproduces the observed data well, as opposed to finding just a good interpolant of the data.

Why does this happen? Since κ scales the gain of the diffusion, it also determines the uncertainty of the ODE solution for a given θ. A high diffusion leads to a less concentrated prior for the subsequent Gauss–Markov regression. For a broad prior, many observations have a similar likelihood, hence a PN posterior whose mean interpolates the data scores favorably (Figure 2). For a low diffusion, the variance of the prior is very tightly concentrated around the ODE solution, meaning only data that closely matches the trajectory will score a high likelihood, in turn leading to interpolation of the ODE solution (Figure 2).

Based on these observations, we propose to re-interpret the diffusion parameter in Fenrir as a optimization hyperparameter which determines how the data should be "weighted" with respect to the solution of the ODE: Fixing κ regularizes the solution of the IVP on the observed data. In particular, we show in Figure 1 and argue in the following that tempering the diffusion leads to improved convergence.

3.2. Diffusion tempering

Above we argued why κ should not be optimized jointly with θ, but treated as a hyperparameter during optimization, since it can trade-off confidence in the data vs. the IVP solution (Fig. 2). Accordingly, we propose to put κ on a decreasing schedule. Starting with a high κ, Fenrir does not have much confidence in the accuracy of its physics-informed prior and therefore most ODE parameters yield a decent likelihood for the observed data. By steadily lowering the diffusion, we decrease the uncertainty of the prior, which means there are less ways for an ODE solution to fit the data. Finally, when the prior has minimal uncertainty presumably only the "true" ODE fits the data.

We call this procedure diffusion tempering (Algorithm 1). We start by defining the number of iterations m and a tempering schedule $\mathcal{T}$, which can be a list of κ or any function that takes a positive integer and returns a value for the diffusion hyperparameter. We set an initial parameter θ_0, i.e. by drawing it from a pre-specified distribution and set $\widehat{\mathcal{M}}_{\mathrm{PN}}(\theta_0, \kappa = \mathcal{T}(0))$. We run an optimizer of choice until convergence to obtain a new parameter estimate. Then, we obtain the next κ from $\mathcal{T}$ and repeat the optimization with the updated estimate of θ, returning the final θ as our parameter estimate. In our case, we initialized the parameters uniformly distributed and opted for a tempering schedule of $\mathcal{T}(i, \kappa_0) = 10^{(\kappa_0 - i)}$, $\kappa_0 = 20$ with a schedule length of 21. This is equivalent to linearly decreasing $\log_{10}(\kappa)$ from 20 to 0.

4. Experiments

We first investigate the effect of diffusion tempering for a pendulum with a single parameter, and show even in this

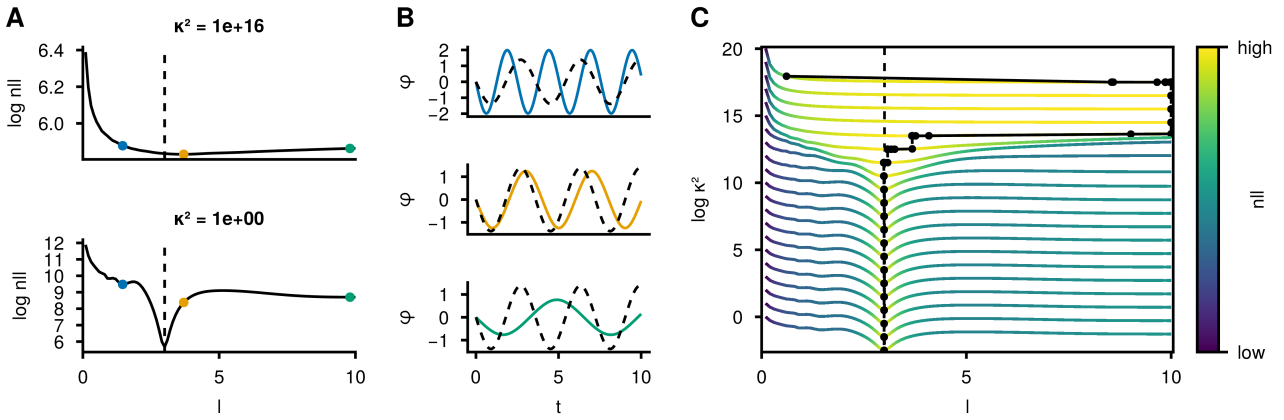

Figure 3. The effect of diffusion tempering on the marginal likelihood $\mathcal{M}(\theta, \kappa)$ of a pendulum with parameter l. **A** The negative log likelihood (nll) for a high κ is very smooth with only one shallow global optimum (yellow). The likelihood for low κ has one sharp global minimum at the true parameters (dashed), but also other local minima (blue, green). **B** IVP solutions for the different local optima marked in A and the true solution (dashed). **C** Diffusion tempering for an exemplary optimization run (black): Optimization starts with a high κ (top) on a very smooth loss landscape. κ is then progressively lowered, revealing a shallow global optimum near the true parameters ($\kappa = 10^{16}$). Further lowering κ finally reveals a sharp global optimum and the optimization converges correctly. By starting with the parameters of the previous optimization, diffusion tempering ensures that the optimization closes in on the correct global optimum.

comparatively simple case that it converges more reliably around the true parameters. We then demonstrate that diffusion tempering also improves Fenrir's convergence for complex ODE models such as the HH model (Hodgkin & Huxley, 1952). Finally, we show that diffusion tempering enables parameter estimation for HH models in parameter regimes where learned diffusions and least-squares methods stop working altogether.

4.1. Exploration of a simple case: The 1D Pendulum

We first demonstrate our algorithm for a pendulum. While this is a relatively simple problem by construction, it is already challenging for classical simulation-based methods such as Runge–Kutta least-squares approaches (Bard, 1974), which tend to fail for high frequencies and are sensitive to initialization (Benson, 1979). Furthermore, with only a single free parameter, the pendulum length l, the effect of tempered optimization can be illustrated well (Figure 3). Details on the full dynamics and the parameterisation are provided in Appendix A5.1.

We evaluated the Fenrir marginal likelihood at different pendulum lengths for a high diffusion constant. This confirmed our theory from Section 3.1 that a high κ indeed leads to similar likelihoods for most parameters (Figure 3A). Additionally, with increasing length l the loss also increased, which discourages the zero-function ($l \to \infty$) which would otherwise be a local optimum. However, while the global optimum was roughly in the right place, it was not yet located at the true l. This was in stark contrast to when we evaluated the marginal likelihood for a low diffusion. Here the global optimum correctly identified the true l, but the loss landscape had additional local optima (Figure 3B). When we repeated this process for every κ that our tempering schedule visited, we observed that solving separate optimization problems at subsequently lower values of κ exploited the smooth nature of the loss landscape to skip over later forming local optima. While earlier minima might not have been accurate, they provided a better initialization for following optimizations at lower κ (Figure 3C). Diffusion tempering first interpolated the observed data, but later put more and more weight on a good ODE solution. This avoided local minima that provided a likelihood that explained neither the ODE or the data well. For the pendulum, diffusion tempering converged as often as Fenrir but three times as often as RK least-squares regression in a ball around the true parameters (Table 1, PD).

4.2. Exploration of a complex case: The Hodgkin–Huxley Model

We next turned to a more complex system of ODEs, the Hodgkin–Huxley model for neural membrane biophysics (Hodgkin & Huxley, 1952). It models the change in voltage across a cell's membrane $V(t)$ in response to an external input $I_{\text{in}}(t)$ as different ionic currents flowing through a circuit (Figure 4A). For the detailed equations and parameterisation see Appendix A5.3.

This model is an interesting case study because there is considerable scientific interest in identifying the conductances of these ionic currents from voltage recordings to inform neuron simulations. All-or-none action potentials (spikes)

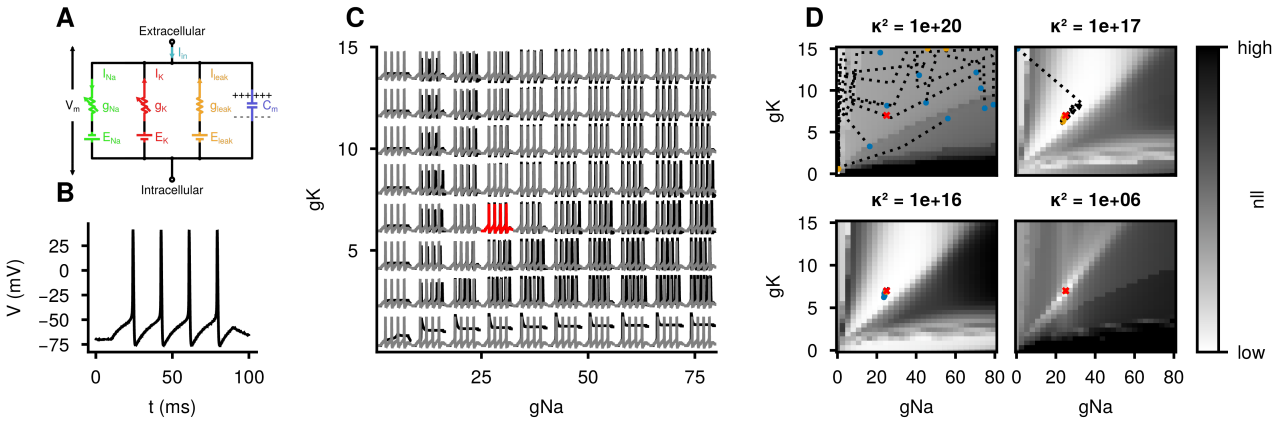

Figure 4. Parameter estimation for the sodium g_{Na} and potassium g_K conductance of a HH model. **A** Circuit modeling components of a neuronal cell as electrical elements. The lipid bilayer acts as a capacitor (C_m). Ion channels are represented by resistors. Sodium and potassium conductances are voltage dependent (g_{Na},g_K), and the leak conductance is constant (g_{leak}). The electrochemical gradients driving the flow of ions can be represented as voltage sources (E_{Na},E_K,E_{leak}). **B** Noisy observation of the membrane voltage. **C** Solutions of the ODE for different combinations of parameters (black) compared to the true parameters (grey). The true parameters are highlighted in red. **D** Parameter optimization during four different stages of diffusion tempering for a random subset of initializations. Optimization trajectories (dotted) in each likelihood landscape are shown from start (blue) to convergence (orange). Very high loss values were clipped for better visual clarity. Plots for the full schedule are provided in Figure A1, with corresponding solutions in Figure A2.

and steep voltage gradients make this challenging, which is why grid (Prinz et al., 2003) or random searches (Taylor et al., 2009), genetic algorithms (Druckmann et al., 2007; Ben-Shalom et al., 2012; Achard & De Schutter, 2006) and simulation-based Inference (Gonçalves et al., 2020; Lueckmann et al., 2017) are common tools.

For our experiments, we generated observations from the HH model with a square wave depolarizing current of 210 pA between 10ms and 90 ms as stimulus. In the initial experiments, we only considered two free parameters, g_{Na} and g_{K}, for which we generated data at 25 mS/cm^2 and 7 mS/cm^2, respectively. The initial voltage was chosen as $V(0) = -70$ mV and the membrane voltage was simulated for a time interval of 100 ms, leading to a spiking trace with four action potentials (Figure 4B). See Appendix A5.3 for details on how we initialized the gating variables.

Even with just two free parameters, the HH model displayed highly complex patterns of activity ranging from non-spiking solutions in regimes of low sodium or potassium conductance to spike trains of different amplitudes, frequencies or phase (Figure 4C). Additionally, small changes in the parameter space could sometimes lead to dramatic changes in the solution, like additional spikes appearing.

To explore this and the effect on the inference more systematically, we computed the loss landscape of the model with respect to observed data for different diffusion parameters (Figure 4D). We observed a similar behavior as in Section 4.1 for the pendulum. For a high diffusion parameter, the loss was very smooth and sloped only in one direction, leading the optimization runs to converge at the non-spiking response simply interpolating the baseline of the action potentials (Figure 4D top, left). This likely happened because the action potentials were narrow, consisting only of a few measurements each. Hence, their effect on the loss was not strong enough to make these solutions prohibitively costly, as the high diffusion parameter made the inference focus on data interpolation. For lower diffusion parameters, a large basin around the true parameters appeared, which the parameter estimates started to converge to (Figure 4D top, right). For very low diffusion values, the basin then morphed into a much sharper global optimum around the true parameters which attracted the estimates that were previously collecting in the basin (Figure 4D bottom, left).

4.3. Systematic performance benchmarking against alternative methods

To demonstrate the effect of tempering the diffusion more systematically, we compared to both learning and fixing κ in the following. More specifically we studied: (i) non-linear least-squares regression with Runge–Kutta (RK), (ii) Fenrir with learned diffusion as proposed by Tronarp et al. (2022), and (iii) Fenrir with a range of fixed diffusion parameters: We explore a low, a high and the best κ according to a grid search over all values visited during tempering (for details see Figure A3) as well as the maximum likelihood κ at the true parameters (optimal).

We considered an optimization run as correctly converged if the final parameter estimate deviated at most 5% from the

Table 1. Comparison of different methods, models and model sizes for parameter estimation. LV: Lottka-Volterra, PD: Pendulum, HH_x Hodgkin–Huxley with x compartments, D_θ: Number of parameters, N_θ: The number of correctly identified parameters, CONV: correctly converged. *In only a few cases the optimization diverged, which lead to NaNs in the loss and tRMSEs orders of magnitude above the mean. We excluded these runs from the statistics marked with a* *.

ODE	D_θ	ALG	ITER	pRMSE	CONV	tRMSE	N_θ
PD	1	FENRIR	74.12 ± 11.55	0.01 ± 0.08	0.99	0.04 ± 0.26	0.99 ± 0.10
PD	1	RK	$\mathbf{34.11 \pm 18.45}$	1.42 ± 1.09	0.32	0.98 ± 0.72	0.32 ± 0.47
PD	1	OURS	303.89 ± 12.43	$\mathbf{0.00 \pm 0.00}$	**1.00**	$\mathbf{0.02 \pm 0.00}$	$\mathbf{1.00 \pm 0.00}$
PD	1	OURS+	92.21 ± 0.95	$\mathbf{0.00 \pm 0.00}$	**1.00**	$\mathbf{0.02 \pm 0.00}$	$\mathbf{1.00 \pm 0.00}$
LV	2	FENRIR	$\mathbf{60.68 \pm 36.73}$	1.41 ± 1.01	0.20	2.84 ± 2.71	0.40 ± 0.80
LV	2	RK	97.15 ± 146.95	1.35 ± 1.11	0.24	2.35 ± 1.32	0.48 ± 0.86
LV	2	OURS	411.18 ± 151.25	0.41 ± 0.82	0.77	$^*0.54 \pm 1.14$	1.54 ± 0.85
LV	2	OURS+	132.43 ± 31.01	$\mathbf{0.35 \pm 0.77}$	**0.79**	$^*\mathbf{0.43 \pm 1.03}$	$\mathbf{1.59 \pm 0.81}$
LV	4	FENRIR	$\mathbf{112.45 \pm 47.24}$	1.31 ± 0.94	0.23	$\mathbf{4.18 \pm 13.65}$	1.07 ± 1.70
LV	4	RK	271.49 ± 151.05	1.09 ± 0.76	0.24	6.57 ± 34.45	0.97 ± 1.71
LV	4	OURS	727.15 ± 482.93	0.81 ± 0.88	0.43	$^*10.41 \pm 69.37$	1.76 ± 1.98
LV	4	OURS+	292.05 ± 139.98	$\mathbf{0.74 \pm 0.82}$	**0.44**	$^*13.77 \pm 79.08$	$\mathbf{1.79 \pm 1.98}$
HH_1	1	FENRIR	46.96 ± 19.08	0.38 ± 0.67	0.68	7.85 ± 10.70	0.68 ± 0.47
HH_1	1	RK	$\mathbf{43.30 \pm 43.45}$	0.42 ± 0.48	0.57	7.54 ± 8.26	0.57 ± 0.50
HH_1	1	OURS	382.08 ± 32.19	$\mathbf{0.00 \pm 0.00}$	**1.00**	$\mathbf{0.43 \pm 0.02}$	$\mathbf{1.00 \pm 0.00}$
HH_1	1	OURS+	116.73 ± 7.11	$\mathbf{0.00 \pm 0.00}$	**1.00**	0.43 ± 0.11	$\mathbf{1.00 \pm 0.00}$
HH_1	2	FENRIR	110.04 ± 61.70	0.20 ± 0.37	0.75	5.89 ± 10.15	1.53 ± 0.83
HH_1	2	RK	$\mathbf{54.02 \pm 62.60}$	0.28 ± 0.45	0.72	4.88 ± 7.58	1.44 ± 0.90
HH_1	2	OURS	696.22 ± 78.10	$\mathbf{0.00 \pm 0.00}$	**1.00**	$\mathbf{0.42 \pm 0.04}$	$\mathbf{2.00 \pm 0.00}$
HH_1	2	OURS+	197.81 ± 40.38	0.04 ± 0.19	0.96	1.08 ± 3.20	1.92 ± 0.39
HH_1	3	FENRIR	$\mathbf{122.15 \pm 49.74}$	0.59 ± 0.65	0.51	9.31 ± 9.63	1.53 ± 1.51
HH_1	3	RK	223.55 ± 117.31	0.90 ± 0.25	0.03	14.33 ± 4.01	0.13 ± 0.56
HH_1	3	OURS	676.33 ± 150.72	$\mathbf{0.01 \pm 0.10}$	**0.99**	$\mathbf{0.60 \pm 1.51}$	$\mathbf{2.97 \pm 0.30}$
HH_1	6	FENRIR	$\mathbf{108.06 \pm 108.49}$	13.36 ± 6.97	0.00	26.21 ± 7.38	1.05 ± 0.22
HH_1	6	RK	210.26 ± 120.86	12.27 ± 6.87	0.00	16.80 ± 3.81	1.18 ± 0.39
HH_1	6	OURS	2159.60 ± 532.55	$\mathbf{10.36 \pm 7.72}$	0.00	$^*\mathbf{15.20 \pm 5.41}$	$\mathbf{1.21 \pm 0.46}$
HH_2	4	FENRIR	286.54 ± 205.37	0.28 ± 0.44	0.68	12.00 ± 17.37	2.80 ± 1.78
HH_2	4	RK	$\mathbf{136.50 \pm 200.20}$	0.43 ± 0.56	0.50	7.98 ± 9.86	2.08 ± 1.81
HH_2	4	OURS	1492.03 ± 335.17	$\mathbf{0.00 \pm 0.00}$	**1.00**	$\mathbf{0.60 \pm 0.01}$	$\mathbf{4.00 \pm 0.00}$
HH_2	6	FENRIR	$\mathbf{221.28 \pm 144.56}$	0.62 ± 0.72	0.50	13.01 ± 13.10	3.06 ± 2.96
HH_2	6	RK	390.34 ± 195.85	0.88 ± 0.22	0.00	19.36 ± 5.54	0.31 ± 0.75
HH_2	6	OURS	1525.57 ± 448.56	$\mathbf{0.12 \pm 0.32}$	**0.88**	$\mathbf{3.01 \pm 6.70}$	$\mathbf{5.28 \pm 1.96}$

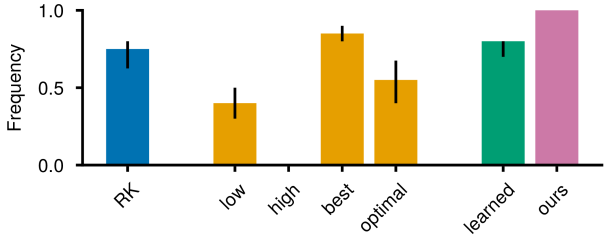

Figure 5. Convergence for a two parameter HH model: Diffusion tempering converges more reliably than than non-linear least-squares regression using a RK solver and Fenrir with a learned and the best single κ. Histogram shows the medians with black bars indicating quartiles for 100 runs split into groups of 10.

true parameters (for details see Appendix A3).

We observed that for a two parameter HH model, learning the diffusion hyperparameter performed on par with the classical RK method, suggesting Tronarp et al. (2022)'s approach did not provide any benefit in this setting (Figure 5). Moreover, the same performance could be achieved if the diffusion parameter was just fixed at a well selected value. Interestingly, the κ that maximized the likelihood at the true parameters performed worse than when the diffusion was just fixed at a low value (Figure 5). Nonetheless, our method still yielded the best convergence, implying that active adaptation of κ during optimization is imperative.

To show that diffusion tempering worked for a variety of different model sizes and complexities, we benchmarked our algorithm against RK least-squares regression and the original method of Tronarp et al. (2022) for the 1D pendulum (Appendix A5.1), the Lotka-Volterra equations (Appendix A5.2) with two and four free parameters as well as multiple HH models with one (HH_1) and two compartments (HH_2) and one to six free different parameters (Appendix A5.3). For 100 different initializations, we evaluated the algorithms based on trajectory mean squared error (tRMSE, Definition A3.1), relative parameter mean squared error (pRMSE,

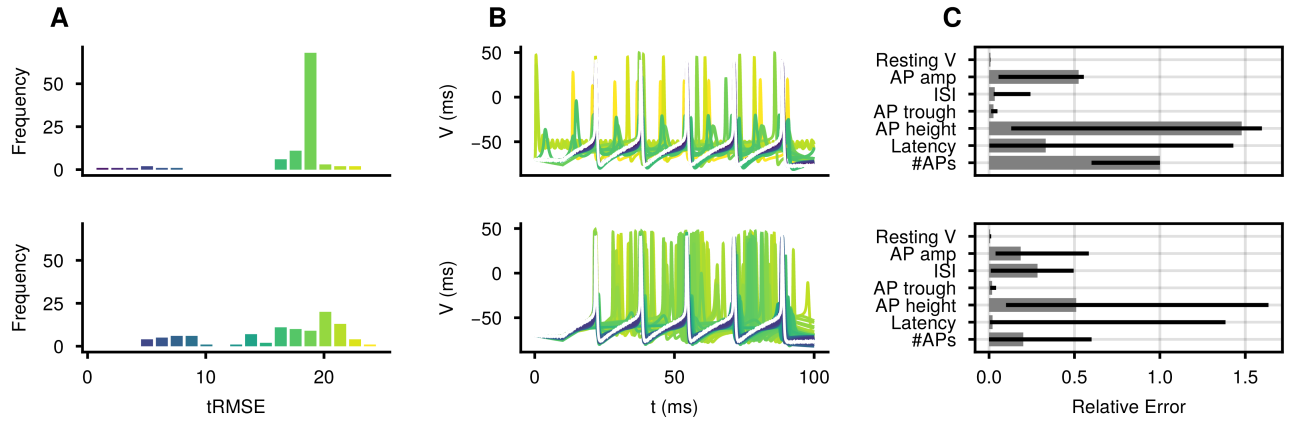

Figure 6. Parameter inference for a six parameter HH model for 100 initializations of RK (top) and tempered Fenrir (bottom). **A** Distribution of tRMSEs. The average tRMSE for Fenrir is lower than for RK. **B** Voltage traces for the inferred parameters, colored by their tRMSE. The observation is shown in white. **C** Median errors for seven commonly used electrophysiological features (as defined in Appendix A3). Quartiles are marked as black bars. Fenrir is able to reproduce qualitative aspects of the data much better than RK.

Definition A3.2), fraction of successfully converged runs (CONV), the number of iteration steps (ITER) and the number of correctly identified parameters (N_θ). While the focus of our work was on the reliability of convergence we also show that the inclusion of a simple early stopping criterion (see Appendix A2 for details) makes our algorithm competitive on cost.

We observed that diffusion tempering (OURS) yielded the best results across all of our experiments in terms of tRMSE, pRMSE and the number of successfully converged initializations (Table 1). Moreover, with our method, we recovered the true parameters from about twice to four times as often as the default Fenrir depending on the model. Even in settings where the RK baseline was not able to converge correctly at all, as was the case for the two compartment model with six parameters, our method was correct 88% of the time. Additionally, in the case of the three parameter HH_1 model, diffusion tempering doubled the performance of Fenrir. As expected, using a schedule came at the expense of additional optimization steps. For our selected schedule, this meant about an order of magnitude more steps compared to the original method. However, the inclusion of a simple early stopping criterion (OURS+) made diffusion tempering cost competitive without loss of performance for the subset of problems we tested. This can be attributed to the fact that convergence with a high precision is unnecessary during early tempering stages and that most trajectories converged far before the end of the schedule (Figure A1). Combined with steeper schedules (Appendix A8.1 this would bring the costs in line with the original Fenrir method while retaining superior performance.

While for the HH_1 model with 6 parameters all methods seemingly performed on par with respect to mean pRMSE and tRMSE, no algorithm manages to converged to the true parameters reliably. However, this was a very complicated model, which is why we looked at the parameter estimates in detail. First we looked at the distributions of tRMSEs.

We noticed that the tRMSEs obtained via diffusion tempering were much more uniformly distributed than those from RK, while being in the same range (Figure 6A). However, not only was the fraction of parameter estimates which yielded steady-state solutions for RK extremely high (Figure 6B) – for many applications these are discarded immediately (Prinz et al., 2003; Rossant et al., 2011) – but they often displayed very different spike frequencies from the observed data. This was not the case for our method for which the solutions come qualitatively much closer to that of the observed data. For practitioners interested in parameter estimation for this model, it mainly matters whether a set of commonly used eletrophysiological summary statistics is reproduced by the fitted model (for their definitions see Appendix A3.1). Here we found that our method deviated much less from the observed behavior, suggesting that the parameters capture essential aspects of the data much better than the classical baseline (Figure 6C). Both methods also performed much better than when the diffusion was learned (Figure A4).

5. Discussion

Gradient descent has the potential to efficiently fit ODEs to data, but non-linear differential equations can exhibit many local minima, leading to poor convergence and extreme sensitivity to initial conditions. Here, we used probabilistic integrators to improve the convergence of gradient descent based parameter fitting of ODEs. We developed diffusion

tempering, a method that successively solves several parameter search tasks. We demonstrated that our method dramatically improves the reliability of parameter optimization across several popular ODEs, and we showed that our method enables gradient descent based parameter estimation on a six-parameter Hodgkin–Huxley model, which had been unattainable with existing gradient-based methods.

5.1. Limitations

Of the compared methods diffusion tempering was the most expensive in terms of runtime, which we attribute to two aspects: the runtime of the ODE solver, and the sequential solving of multiple optimization problems. The probabilistic ODE solvers used by our method are semi-implicit and A-stable (Tronarp et al., 2019). While this allows them to perform well on stiff ODEs, this comes at the cost of cubic scaling with the ODE dimension, shared by traditional implicit Runge–Kutta methods (Hairer & Wanner, 1999). Using explicit probabilistic ODE solvers, which scale linearly with the ODE dimension (Krämer et al., 2022), would improve the runtime of diffusion tempering on larger systems. Additionally, the overall runtime of diffusion tempering is strongly dependent on the iterative optimization procedure that is used. This includes the tempering curve as well as the optimization itself. However, we did show that early stopping and steeper schedules, together, can bring down the computational cost to much closer to that of the other methods, and we suspect better tempering schemes can improve this even further. We also note that in our work we assumed the magnitude of the observation noise R to be known, which in practice is often not the case and which can be taken into account by Fenrir as well.

As the scope of this paper was gradient descent based parameter estimation in ODEs, we did not include other commonly used approaches such as genetic algorithms (Hazelden et al., 2023) in our experimental evaluation. Furthermore, we only looked at recovering parameters for a known form of dynamics. While optimizing parameters of NODEs (Chen et al., 2018) presents an interesting problem set, we see a few potential challenges before our method can be applied here. Firstly, computing gradients efficiently would require the use of adjoints (Chen et al., 2018), which, to our knowledge, has not yet been derived in the probabilistic numerical framework. Additionally, the $O(d^3)$ scaling of our method in the ODE dimension would limit the number of latent dimensions that can be modeled.

5.2. Future Work

While we already presented ways to reduce the cost of diffusion tempering using simple early stopping criteria or different schedules, more elaborate and potentially problem aware tempering schemes could further increase the efficiency and effectiveness of our method. Additionally, in our work we assumed the observation noise to be known, hence further work is needed to study the effect of also learning or scheduling this parameter, which could have a similarly regularizing effect on the optimization.

Authorship Contribution

Conceptualization: PB, PH, JHM, JB, NB; Methodology: JB, NB; Software: JB, NB; Investigation: JB, NB, MD; Formal Analysis: JB; Visualization: JB; Supervision: PB, PH, JHM; Writing – Original Draft: JB, NB; Writing – Review & Editing: PB, NB, MD, PH, JHM, KLK; Funding Acquisition: PB, PH, JHM

Acknowledgments

We thank the members of the Probabilistic Inference in Mechanistic Models (PIMMS) Network project of the Cluster of Excellence "Machine Learning — New Perspectives for Science" for discussion. The work was funded by the German Science Foundation (EXC 2064 "Machine Learning — New Perspectives for Science", 2064/1 – 390727645), the Hertie Foundation, the European Union (ERC, "NextMechMod", ref. 101039115, "ANUBIS", ref. 101123955, "DeepCoMechTome", ref. 101089288), as well as the German Federal Ministry of Education and Research (BMBF) through the Tübingen AI Center (01IS18039A). Views and opinions expressed are however those of the authors only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency. Neither the European Union nor the granting authority can be held responsible for them. PH and NB are supported by funds of the Ministry of Science, Research and Arts of the State of Baden-Württemberg. We also thank the International Max Planck Research School for Intelligent Systems (IMPRS-IS) for supporting JB, NB, MD and KLK.

Impact Statement

This paper contributes to the advancement of the field of machine learning. It aligns with the ongoing progress in the broader landscape of machine learning research. Compared to other methods in the field our work does not introduce unique and distinct societal challenges that warrant explicit discussion.

References

Achard, P. and De Schutter, E. Complex parameter landscape for a complex neuron model. *PLoS computational biology*, 2(7):e94, 2006. doi: 10.1371/journal.pcbi.0020094.

Armijo, L. Minimization of functions having lipschitz continuous first partial derivatives. *Pacific Journal of Mathematics*, 16:1–3, January 1966. ISSN 0030-8730. doi: 10.2140/pjm.1966.16.1.

Bard, Y. *Nonlinear parameter estimation*. Academic Press, New York, 1974. ISBN 978-0-12-078250-5.

Ben-Shalom, R., Aviv, A., Razon, B., and Korngreen, A. Optimizing ion channel models using a parallel genetic algorithm on graphical processors. *Journal of Neuroscience Methods*, 206(2):183–194, 2012. doi: 10.1016/j.jneumeth.2012.02.024.

Benson, M. Parameter fitting in dynamic models. *Ecological Modelling*, 6(2):97–115, February 1979. ISSN 03043800. doi: 10.1016/0304-3800(79)90029-2.

Bezanson, J., Edelman, A., Karpinski, S., and Shah, V. B. Julia: A fresh approach to numerical computing. *SIAM Review*, 59(1):65–98, January 2017. ISSN 0036-1445, 1095-7200. doi: 10.1137/141000671.

Cao, J., Wang, L., and Xu, J. Robust estimation for ordinary differential equation models. *Biometrics*, 67(4): 1305–1313, December 2011. ISSN 0006341X. doi: 10.1111/j.1541-0420.2011.01577.x.

Chen, T. Q., Rubanova, Y., Bettencourt, J., and Duvenaud, D. Neural ordinary differential equations. In Bengio, S., Wallach, H. M., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in neural information processing systems 31: Annual conference on neural information processing systems, NeurIPS 2018*, pp. 6572–6583, 2018.

Cockayne, J., Oates, C., Sullivan, T., and Girolami, M. Bayesian probabilistic numerical methods. *SIAM Review*, 61(3):756–789, 2019. ISSN 0036-1445, 1095-7200. doi: 10.1137/17M1139357.

Cranmer, K., Brehmer, J., and Louppe, G. The frontier of simulation-based inference. *Proceedings of the National Academy of Sciences of the United States of America*, 117 (48):30055–30062, December 2020. ISSN 1091-6490. doi: 10.1073/pnas.1912789117.

Dass, S. C., Lee, J., Lee, K., and Park, J. Laplace based approximate posterior inference for differential equation models. *Statistics and Computing*, 27(3):679–698, May 2017. ISSN 0960-3174, 1573-1375. doi: 10.1007/s11222-016-9647-0.

Dayan, P. and Abbott, L. F. *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems*. MIT Press, 2001. ISBN 978-0-262-54185-5.

Druckmann, S., Banitt, Y., Gidon, A., Schürmann, F., Markram, H., and Segev, I. A novel multiple objective optimization framework for constraining conductance-based neuron models by experimental data. *Frontiers in Neuroscience*, 1:1, 2007. ISSN 1662-453X. doi: 10.3389/neuro.01.1.1.001.2007.

Gonçalves, P. J., Lueckmann, J.-M., Deistler, M., Nonnenmacher, M., Öcal, K., Bassetto, G., Chintaluri, C., Podlaski, W. F., Haddad, S. A., Vogels, T. P., Greenberg, D. S., and Macke, J. H. Training deep neural density estimators to identify mechanistic models of neural dynamics. *eLife*, 9:e56261, 2020. ISSN 2050-084X. doi: 10.7554/eLife.56261.

Hairer, E. and Wanner, G. Stiff differential equations solved by radau methods. *Journal of Computational and Applied Mathematics*, 111(1):93–111, November 1999. ISSN 0377-0427. doi: 10.1016/S0377-0427(99)00134-X.

Hairer, E., Nørsett, S. P., and Wanner, G. *Solving Ordinary Differential Equations I*, volume 8 of *Springer Series in Computational Mathematics*. Springer Berlin Heidelberg, Berlin, Heidelberg, 1987. ISBN 978-3-662-12609-7. doi: 10.1007/978-3-662-12607-3.

Hazelden, J., Liu, Y. H., Shlizerman, E., and Shea-Brown, E. Evolutionary algorithms as an alternative to backpropagation for supervised training of biophysical neural networks and neural odes. (arXiv:2311.10869), November 2023. doi: 10.48550/arXiv.2311.10869.

Hennig, P., Osborne, M. A., and Kersting, H. P. *Probabilistic Numerics*. Cambridge University Press, June 2022.

Hethcote, H. W. The mathematics of infectious diseases. *SIAM Review*, 42(4):599–653, January 2000. ISSN 0036-1445, 1095-7200. doi: 10.1137/S0036144500371907.

Hodgkin, A. L. and Huxley, A. F. A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of Physiology*, 117 (4):500–544, August 1952. ISSN 0022-3751, 1469-7793. doi: 10.1113/jphysiol.1952.sp004764.

Kersting, H., Sullivan, T. J., and Hennig, P. Convergence rates of gaussian ode filters. *Statistics and Computing*, 30(6):1791–1816, Nov 2020. ISSN 1573-1375. doi: 10.1007/s11222-020-09972-4.

Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P. Optimization by simulated annealing. *Science*, 220(4598): 671–680, May 1983. doi: 10.1126/science.220.4598.671.

Krämer, N. and Hennig, P. Stable implementation of probabilistic ode solvers. *CoRR*, 2020.

Kravtsov, S. Dynamics and predictability of hemispheric-scale multidecadal climate variability in an observationally constrained mechanistic model. *Journal of Climate*, 33(11):4599–4620, June 2020. ISSN 0894-8755, 1520-0442. doi: 10.1175/JCLI-D-19-0778.1.

Krämer, N., Bosch, N., Schmidt, J., and Hennig, P. Probabilistic ode solutions in millions of dimensions. In *Proceedings of the 39th International Conference on Machine Learning*, pp. 11634–11649. PMLR, June 2022.

Lueckmann, J.-M., Gonçalves, P. J., Bassetto, G., Öcal, K., Nonnenmacher, M., and Macke, J. H. Flexible statistical inference for mechanistic models of neural dynamics. In Guyon, I., von Luxburg, U., Bengio, S., Wallach, H. M., Fergus, R., Vishwanathan, S. V. N., and Garnett, R. (eds.), *Advances in neural information processing systems 30: Annual conference on neural information processing systems*, pp. 1289–1299, 2017.

Marjoram, P., Molitor, J., Plagnol, V., and Tavaré, S. Markov chain monte carlo without likelihoods. 100:15324–15328, December 2003. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.0306899100.

Mogensen, P. K. and Riseth, A. N. Optim: A mathematical optimization package for julia. *Journal of Open Source Software*, 3:615, April 2018. ISSN 2475-9066. doi: 10.21105/joss.00615.

Neal, R. M. Probabilistic inference using markov chain monte carlo methods. pp. 144, 1993.

Neelakantan, A., Vilnis, L., Le, Q. V., Kaiser, L., Kurach, K., Sutskever, I., and Martens, J. Adding gradient noise improves learning for very deep networks. 2017.

Nocedal, J. and Wright, S. J. *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering. Springer New York, 2006. ISBN 978-0-387-30303-1. doi: 10.1007/978-0-387-40065-5.

Pospischil, M., Toledo-Rodriguez, M., Monier, C., Piwkowska, Z., Bal, T., Frégnac, Y., Markram, H., and Destexhe, A. Minimal hodgkin–huxley type models for different classes of cortical and thalamic neurons. *Biological Cybernetics*, 99(4–5):427–441, November 2008. ISSN 0340-1200, 1432-0770. doi: 10.1007/s00422-008-0263-8.

Prinz, A. A., Billimoria, C. P., and Marder, E. Alternative to hand-tuning conductance-based models: Construction and analysis of databases of model neurons. *Journal of Neurophysiology*, 90(6):3998–4015, December 2003. ISSN 0022-3077, 1522-1598. doi: 10.1152/jn.00641.2003.

Rackauckas, C. and Nie, Q. Differentialequations.jl – a performant and feature-rich ecosystem for solving differential equations in julia. *Journal of Open Research Software*, 5(11):15, May 2017. doi: 10.5334/jors.151.

Rossant, C., Goodman, D. F., Fontaine, B., Platkiewicz, J., Magnusson, A., and Brette, R. Fitting neuron models to spike trains. *Frontiers in Neuroscience*, 5:9, 2011. ISSN 1662-453X. doi: 10.3389/fnins.2011.00009.

Scala, F., Kobak, D., Bernabucci, M., Bernaerts, Y., Cadwell, C. R., Castro, J. R., Hartmanis, L., Jiang, X., Laturnus, S., Miranda, E., Mulherkar, S., Tan, Z. H., Yao, Z., Zeng, H., Sandberg, R., Berens, P., and Tolias, A. S. Phenotypic variation of transcriptomic cell types in mouse motor cortex. *Nature*, November 2020. ISSN 0028-0836, 1476-4687. doi: 10.1038/s41586-020-2907-3.

Schober, M., Särkkä, S., and Hennig, P. A probabilistic model for the numerical solution of initial value problems. *Statistics and Computing*, 29(1):99–122, Jan 2019. ISSN 1573-1375. doi: 10.1007/s11222-017-9798-7.

Schweppe, F. Evaluation of likelihood functions for gaussian signals. *IEEE Transactions on Information Theory*, 11(1):61–70, January 1965. ISSN 0018-9448. doi: 10.1109/TIT.1965.1053737.

Särkkä, S. *Bayesian filtering and smoothing*. Cambridge University Press, 2013.

Taylor, A. L., Goaillard, J.-M., and Marder, E. How multiple conductances determine electrophysiological properties in a multicompartment model. *Journal of Neuroscience*, 29(17):5573–5586, April 2009. ISSN 0270-6474, 1529-2401. doi: 10.1523/JNEUROSCI.4438-08.2009.

Tronarp, F., Kersting, H., Särkkä, S., and Hennig, P. Probabilistic solutions to ordinary differential equations as nonlinear Bayesian filtering: a new perspective. *Statistics and Computing*, 29(6):1297–1315, 2019. doi: 10.1007/s11222-019-09900-1.

Tronarp, F., Särkkä, S., and Hennig, P. Bayesian ode solvers: the maximum a posteriori estimate. *Statistics and Computing*, 31(3):23, March 2021. ISSN 1573-1375. doi: 10.1007/s11222-021-09993-7.

Tronarp, F., Bosch, N., and Hennig, P. Fenrir: Physics-enhanced regression for initial value problems. In *Proceedings of the 39th International Conference on Machine Learning*, pp. 21776–21794. PMLR, June 2022.

Appendix

A1. Implementation

We wrote all of our experiments in the Julia programming language (Bezanson et al., 2017). For the implementation of probabilistic ODE filters we relied on the original code for Fenrir (Tronarp et al., 2022) and ProbNumDiffEq.jl. To compute the trajectories for the observations as well as for least-squares regression with RK, the `RadauIIA5` (Hairer & Wanner, 1999) solver is used, with adaptive step-size selection. The solver was provided by DifferentialEquations.jl (Rackauckas & Nie, 2017). For the optimization we used the implementation of L-BFGS from Optim.jl (Mogensen & Riseth, 2018). This is a comparable setup to that of Tronarp et al. (2022).

All computations were done on an internal cluster running Intel(R) Xeon(R) Gold 6226R CPUs @ 2.90GHz. All code and data are publicly available on GitHub and Zenodo.

A2. Additional experimental details

To generate the data used as the observations we solved the IVP with the fully-implicit Runge–Kutta `RadauIIA5` (Hairer & Wanner, 1999) method with both absolute and relative tolerances of 10^{-14} on an equi-spaced time grid $t_i \in \mathbb{T} = \{0.0, 0.01, ..., T\}$ with $dt = 0.01$. The Ts differed for each model and can be found in Appendix A5. Uniform Gaussian noise of variance $\sigma^2 = 0.1$ was then added to the data.

To evaluate the different methods we conducted all experiments for 100 different initializations.

As the probabilistic ODE solver for the IVP we used a first order linearization of the vector field and a 3-times integrated integrated Wiener process prior.

During diffusion tempering the optimizer often converged very close to the parameter bounds. This meant that the initial values for the next optimization problem would lie right on top of the bounding box, which would throw an error. We mitigate this, by giving a tiny nudge to any initial values that end up right on top of the parameter bounds before starting the optimization.

In the experiments marked with OURS+ in Table 1 we added a simple early stopping criterion to Algorithm 1 to reduce the computational cost of our algorithm. This could prematurely stop the optimization (OPT) during all but the last scheduled value of κ, if the absolute change in the Fenrir log likelihood $\widehat{\mathcal{M}}_{PN}$ across 3 consecutive updates was below 0.1. Since last tempering step was never interrupted, this ensured the final optimization was performed to the same tolerances as the other methods.

A3. Metrics

To compare the quality of the parameter estimates we use several metrics. Firstly, we compare using the trajectory root mean squared error (RMSE)

Definition A3.1 (Trajectory RMSE)**.** Let $\hat{\theta}$ be the parameters estimated by an inference algorithm, and let $\mathbb{T}_\mathsf{D}$ be the set of measurement nodes. Then, let $\hat{y}(t)$, $t \in \mathbb{T}_\mathsf{D}$, be the estimated system trajectory, computed by numerically integrating the ODE with initial values and parameters as given by the estimated $\hat{\theta}$. The trajectory RMSE (tRMSE) is then defined as

$$\text{tRMSE} := \sqrt{\frac{1}{|\mathbb{T}_\mathsf{D}|} \sum_{t \in \mathbb{T}_\mathsf{D}} \left\| \hat{y}(t) - y(t) \right\|_2^2}. \tag{14}$$

which evaluates the quality of the results in the trajectory space and which is also used in the original work(Tronarp et al., 2022). However, while this is useful to get an idea of how closely the observed data was reproduced it does not necessarily allow to deduce how well the parameters match those that have generated the original data. This is especially a problem for oscillatory systems for which the tRMSE of the zero function is often better than for a solution which is slightly phase shifted, which arguably is often a better fit. We therefore evaluate the quality of the inferred trajectories also in parameter space.

Definition A3.2 (Relative Parameter RMSE)**.** Let $\hat{\theta}$ be the parameters estimated by an inference algorithm, and let θ^* be the known, true parameters which are to be inferred. Let D be the dimensionality of the parameter space. Then the relative

parameter RMSE (pRMSE) d_θ is defined as

$$\text{pRMSE} := \sqrt{\frac{1}{D}\sum_{i=1}^{D}\left\|(\hat{\theta}_i - \theta_i^*)/\theta_i^*\right\|_2^2}. \tag{15}$$

This metric is also used to determine whether an optimization converged successfully or not. If the final parameter estimate $\hat{\theta}$ lies within a 5% ball of the true parameters, i.e. $pRMSE < 0.05$, then we consider it to have correctly converged.

A3.1. Electrophysiological Summary Features

It is difficult to evaluate the quality of parameter estimates for higher parameterised HH models, since a good tRMSE does not necessarily mean characteristic features of the data are being reproduced. Hence to score the quality of parameter estimates, we also define seven commonly used electrophysiological summary features. These reflect measures that electrophysiologists regularly use to characterize and quantify the qualitative behavior of recorded neurons (Scala et al., 2020). This reduces the high dimensional model output to a meaningful set of numbers that emphasize specific aspects of the data and are easier to interpret.

Definition A3.3 (Summary feature)**.** Let θ parameterize an IVP. and let $y(t)$, be the system trajectory, computed. Then a summary feature is defined as a function

$$x := s(y(t)). \tag{16}$$

which reduces the system trajectory to a single number x.

The following summary statistics are computed from the trajectory.

Table 2. Summary features to quantify the behavior of different voltage traces.

Feature	Definition
AP peak	Maximum voltage of an action potential (AP).
AP trough	Minimum voltage of the trough / hyperpolarization that follows the AP.
AP amp	Difference of peak V_{peak} and trough voltage V_{trough}.
ISI	The time between two consecutive APs.
Resting V	Average of $V(9\,ms < t < 10\,ms)$, i.e. 1 ms before the stimulus.
Latency	Time between stimulus onset and reaching the first peak voltage.
#APs	Number of APs.

For AP peak, AP trough, AP amp and ISI we compute the mean. To compare to the observation we then take the absolute value of the relative differences between the observed x^* and the estimated $\hat{x}$ summary features. If no spikes are detected, spike dependent features will set to NaN.

A4. Optimization

For optimization we use L-BFGS for both Fenrir and RK (Nocedal & Wright, 2006). This is also the optimizer used in the original Fenrir paper (Tronarp et al., 2022). To speed up convergence we also use backtracking line search (Armijo, 1966). To determine convergence we used an absolute tolerance of 10^{-9} and a relative tolerance of 10^{-6}, which would give an accuracy of about 10^{-4} to 10^{-5} since the total solution accuracy is roughly 1-2 digits less than the relative tolerances (see SciML Docs).

The optimization was performed with box-constraints, such that the optimization would stay within the limits provided by the uniform prior distribution with upper and lower bounds θ_{ub}, θ_{lb}. All initial parameters were also drawn from this distribution, with initial values set to $u(t_0)$. Observation noise is always fixed to the true $\sigma^2 = 0.1$.

All optimized parameters θ were transformed to ranges of 0 to 1 according to:

$$\tilde{\theta} = \frac{\theta - \theta_{lb}}{\theta_{ub} - \theta_{lb}}, \tag{17}$$

where θ_{lb} and θ_{ub} are the lower and upper bounds on the parameter. Additionally κ was first transformed to log-space, before putting it through Equation (17). This ensures that the gradients for each parameter are roughly comparable.

A5. Models

A5.1. Pendulum

The first ODE we consider is that of a pendulum (Equation (18)). Although this is a fairly simple system, parameter estimation can be challenging due to local optima. The ODE is a 2nd order equation $\frac{d^2\phi}{dt^2} = -\frac{g}{l} sin(\phi)$, which we reformulate as a system of first order equations.

$$\frac{d\phi_1}{dt} = \phi_2 \tag{18a}$$

$$\frac{d\phi_2}{dt} = -\frac{g}{l} sin(\phi_1) \tag{18b}$$

where $g = 9.81\, m/s^2$ and $\varphi(0) = \frac{\pi}{4}$. In our experiments we chose the bounds and values for the single parameter l as in Appendix A5.1.

Table 3. Parameter bounds and values that were used for our experiments.

Parameter	Lower Bound	Upper bound	observation
l	0.1	10.0	3

Hence $\theta = \{l\}$. The system was integrated for $t \in [0, 10]$ and only partially observed with a measurement matrix (see Equation (2)) of $H = [1\ 0]$.

A5.2. Lotka-Volterra

The Lotka–Volterra or predator and prey model, can be described by a pair of first-order nonlinear differential equations (Equation (19)). It describes how the populations of two species (predator and prey) evolve when they coexist in an environment.

$$\frac{dx}{dt} = \alpha x - \beta xy \tag{19a}$$

$$\frac{dy}{dt} = \delta xy - \gamma y \tag{19b}$$

We set the initial populations to $x(0) = 1.0$, $y(0) = 1.0$ and use the bounds and values to parameterize the model as shown in Appendix A5.2. Hence $\theta = \{\alpha, \beta\}$ for the 2 parameter and $\theta = \{\alpha, \beta, \gamma, \delta\}$ for the 4 parameter case. The system was

Table 4. Parameter bounds and values that were used for our experiments.

Parameter	Lower Bound	Upper bound	observation
α	$1 \cdot 10^{-3}$	5.0	1.5
β	$1 \cdot 10^{-3}$	5.0	1.0
γ	$1 \cdot 10^{-3}$	5.0	3.0
δ	$1 \cdot 10^{-3}$	5.0	1.0

integrated for $t \in [0, 20]$ and only partially observed with a measurement matrix (Equation (2)) of $H = [0\ 1]$, such that only the prey species is observed.

A5.3. The Hodgkin Huxley Model

Here we outline the specific versions of the HH model that were used in our experiments. We consider sodium, potassium and leak currents to generate spiking, a slow non-inactivating K^+ current to enable spike-frequency adaptation, and a high-threshold Ca^{2+} current to generate bursting (Pospischil et al., 2008):

$$C\,A\frac{dV_t}{dt} = I_t + \bar{g}_{\text{Na}} m^3 h(E_{\text{Na}} - V_t) \tag{20}$$

$$+ \bar{g}_{\text{K}} n^4 (E_{\text{K}} - V_t) + \bar{g}_{\text{leak}}(E_{\text{leak}} - V_t) + \bar{g}_{\text{M}} p(E_{\text{K}} - V_t) + \bar{g}_{\text{L}} q^2 r(E_{\text{Ca}} - V_t) \tag{21}$$

$\bar{g}_i,\ i \in \{\text{Na}, \text{K}, \text{leak}, \text{M}, \text{L}\}$ are the maximum conductances of the sodium, potassium, leak, adaptive potassium and calcium ion channels, respectively. E_i are the associated reversal potentials. I_t denotes the current per unit area, C the membrane capacitance, A the compartment area and n, m, h, q, r and p represent the fraction of independent gates in the open state, based on Hodgkin & Huxley (1952).

The dynamics of the gates can be expressed in general form as Equations (22a) and (22b), where $\alpha_z(V_t)$ and $\beta_z(V_t)$ are rate constants for each of the gating variables $z \in \{m, n, h, q, r\}$ and p.

$$\frac{dz_t}{dt} = (\alpha_z(V_t)\,(1 - z_t) - \beta_z(V_t) z_t) \tag{22a}$$

$$\frac{dp_t}{dt} = ((p_\infty(V_t) - p_t)/\tau_p(V_t)) \tag{22b}$$

They model the kinetics of different channel proteins. The parameters for $\alpha_z(V_t)$, $\beta_z(V_t)$, $p_\infty(V_t)$ and $\tau_p(V_t, \tau_{max})$ are taken from Pospischil et al. (2008). The detailed equations are also provided in Equation (23).

$$\alpha_m(V) = -0.32\frac{V - V_T - 13}{e^{-(V-V_T-13)/4} - 1}, \quad \beta_m(V) = 0.28\frac{V - V_T - 40}{e^{(V-V_T-40)/5} - 1} \tag{23a}$$

$$\alpha_n(V) = -0.032\frac{(V - V_T - 15)}{e^{-(V-V_T-15)/5} - 1}, \quad \beta_n(V) = 0.5e^{-(V-V_T-10)/40} \tag{23b}$$

$$\alpha_h(V) = 0.128e^{-(V-V_T-17)/18}, \quad \beta_h(V) = \frac{4}{e^{-(V-V_T-40)/5} + 1} \tag{23c}$$

$$\alpha_q(V) = 0.055\frac{-27 - V}{e^{(-27-V)/3.8} - 1}, \quad \beta_q(V) = 0.94e^{(-75-V)/17} \tag{23d}$$

$$\alpha_r(V) = 0.000457e^{(-13-V)/50}, \quad \beta_r(V) = 0.0065/(e^{(-15-V)/28} + 1) \tag{23e}$$

$$\tau_p(V) = \frac{\tau_{\text{max}}}{3.3e^{(V+35)/20} + e^{-(V+35)/20}} \tag{23f}$$

The parameter values and bounds we used in our experiments for the HH model can be found in Table 5.

$A = \frac{\tau}{CR_{in}}$, C the membrane capacitance, R_{in} the input resistance. τ is the membrane time constant, τ_{max} the time constant of the slow K^+ current and V_T the threshold voltage.

Table 5. Parameter bounds and values that were used for our experiments.

Parameter	Lower bound	Upper bound	Observation
C ($\mu F/cm^2$)	0.4	3	1
A (cm^2)	$1.9 \cdot 10^{-5}$	$30.2 \cdot 10^{-5}$	$8.3 \cdot 10^{-5}$
g_{Na} (mS)	0.5	80	25
g_{K} (mS)	$1 \cdot 10^{-4}$	15	7
E_{Na} (mV)	50	100	53
E_{K} (mV)	110	-70	-107
g_{leak} (mS)	$1 \cdot 10^{-4}$	0.8	0.1
E_{leak} (mV)	-110	-50	-70
V_T (mV)	90	-40	-60
g_{M} (mS)	$1 \cdot 10^{-5}$	0.6	0.01
E_{Ca} (mV)	100	150	120
g_{L} (mS)	$1 \cdot 10^{-4}$	0.6	0.01
τ_{max} (s)	50	$3 \cdot 10^3$	$4 \cdot 10^3$

The default parameters are chosen according to http://help.brain-map.org/download/attachments/8323525/BiophysModelPeri.pdf. With the parameters for the cell area A being taken from in vitro recordings from the mouse cortex from the Allen Cell Type Database the cell ID (509881736). This closely follows the experiments from (Gonçalves et al., 2020).

For the 1, 2 and 3 parameter models we removed the equations for I_M, and I_L to make it easier to simulate. This is the same as considering g_M and g_L to be 0. Furthermore, for the 6 and 8 parameter model the parameters where adjusted to $gleak = 0.05\, mS$ and $\tau_{max} = 1000\, s$. The initial voltage was set to $V(0) = -70\, mV$, from which the initial gating variables where computed according to Equation (24), where $z(0) = z_\infty(V(0))$ for $z \in \{m, n, h, p, q, r\}$. The system was integrated for $t \in [0, 100]$ms and only the voltage was observed, which yields a measurement matrix of $H = [1\ 0\ 0\ 0\ 0\ 0\ 0]$. For the full system and $H = [1\ 0\ 0\ 0]$ with I_M and I_L removed.

$$m_\infty(V) = 1/(1 + \beta_m(V)/\alpha_m(V)) \tag{24a}$$
$$n_\infty(V) = 1/(1 + \beta_n(V)/\alpha_n(V)) \tag{24b}$$
$$h_\infty(V) = 1/(1 + \beta_h(V)/\alpha_h(V)) \tag{24c}$$
$$p_\infty(V) = 1/(1 + e^{-(V+35)/10}) \tag{24d}$$
$$q_\infty(V) = 1/(1 + \beta_q(V)/\alpha_q(V)) \tag{24e}$$
$$r_\infty(V) = 1/(1 + \beta_r(V)/\alpha_r(V)) \tag{24f}$$
$$\tag{24g}$$

Depending on the model, the following parameter sets were optimized.

Table 6. Parameter sets for the different single compartment models, which we denote with HH_1.

#parameters	θ
1	$\{g_{\text{Na}}\}$
2	$\{g_{\text{Na}}, g_{\text{K}}\}$
3	$\{g_{\text{Na}}, g_{\text{K}}, g_{\text{leak}}\}$
6	$\{g_{\text{Na}}, g_{\text{K}}, g_{\text{Na}}, g_{\text{M}}, V_T, g_{\text{L}}\}$

A5.3.1. MULTICOMPARTMENT MODELS

To model neurons with a complex morphological structure the neuron is split into discrete compartments that are small enough that the variation of the membrane potential across it is negligible. Then the continuous membrane potential V can be approximated as by a sum of local compartment potentials. For a non-branching cable, this can be expressed as

$$CA_i \frac{dV_i}{dt} = -I_{ionic,i} + I_i + g_{i,i+1}\left(V_{i+1} - V_i\right) + g_{i,i-1}\left(V_{i-1} - V_i\right), \tag{25}$$

with subscript i indexing the compartments, the sum of the ionic currents across the cells membrane I_{ionic}, the external current I and coupling coefficients g. Each membrane potential V_i satisfies an equation similar to Equation (20) except that each compartments couples to two neighbors (unless at the ends of the cable). For more context see (Dayan & Abbott, 2001) Chapter 6. In our experiments we used coupling coefficients of $g = 1$ and compartment areas of $A_i = \frac{A}{N}, \quad i \in \{1, ..., N\}$, for N compartments, such that the total area would sum to the same A as in Table 5. Furthermore, we stimulated only the first compartment and considered all compartments to be observed (voltage only). The parameters used for the 4 and 6 parameter models where modified from Table 5. For the 4 parameter model the 2nd compartment was adjusted to $g_{Na} = 20\,mS$ and $g_K = 10\,mS$. For the 6 parameter model, the first $g_{leak} = 0.09\,mS$ and the 2nd compartment was changed to $g_{Na} = 20\,mS$, $g_K = 10\,mS$ and $g_{leak} = 0.11\,mS$.

Depending on the number of compartments N this yields parameter sets of the form $\theta = \{\theta_1, ..., \theta_N\}$, where θ_i are analogous to the parameters of the single compartments in Table 6. The measurements are taken with

$$H = \begin{bmatrix} H_1 & 0 & \dots \\ \vdots & \ddots & \\ 0 & & H_N, \end{bmatrix} \tag{26}$$

where H_i are the measurement matrices for a single compartment. We denote the number of compartments that were used in an experiment with a subscript HH_i.

A6. Additional Details on Section 4.2

Here we show all the trajectories and likelihoods for which a subset was presented in Figure 4. Additionally, we plotted the solutions at each of the initial and final parameters for each diffusion parameter.

While the likelihood is initially very flat and the uniformly initialized parameters collect on the top left edge, a basin around the global optimum appears at around $\kappa = 10^{18}$ (Figure A1). At $\kappa = 10^{17}$ the ridge that separates the local optimum from the basin is small enough for the optimizer to cross into the basin in which the global optimum resides. With decreasing κ the basin starts to taper more and more towards the true parameters, which becomes very distinct down to about $\kappa = 10^6$, after which point hardly any changes to the likelihood landscape are visible.

We observed that first all runs converge to the steady state solution at $\kappa = 10^{18}$, before suddenly developing spikes and converging correctly (Figure A2). After reaching a diffusion parameter of $\kappa = 10^{15}$, the subset of initial values has fully converged and hardly any changes are visible.

A7. Additional Details on Section 4.3

Once the diffusion hyperparameter is set to $\log \kappa \leq 17$ most values perform similarly well in terms of recovering the true parameters about half of the time, with the exception of $\log \kappa = 16$, which does so with about 80% reliability (Figure A3). Additionally, the average distance from the true parameters starts decreasing between $\log \kappa = 9$ and $\log \kappa = 17$, before sharply jumping again. The fact that the fraction of parameters that converge correctly does not change significantly until hitting $\log \kappa = 1e15$, suggests that the same set of, favorable, initial guesses converge no matter the loss. However, since the mean pRMSE starts decreasing overall, suggests that initializations further away also end up closer to the global optimum.

Comparison of Fenrir and Diffusion tempering for a 6 parameter HH model. Diffusion tempering yields electrophysiological summary statistics much closer to those of the true observation compared to Fenrir.

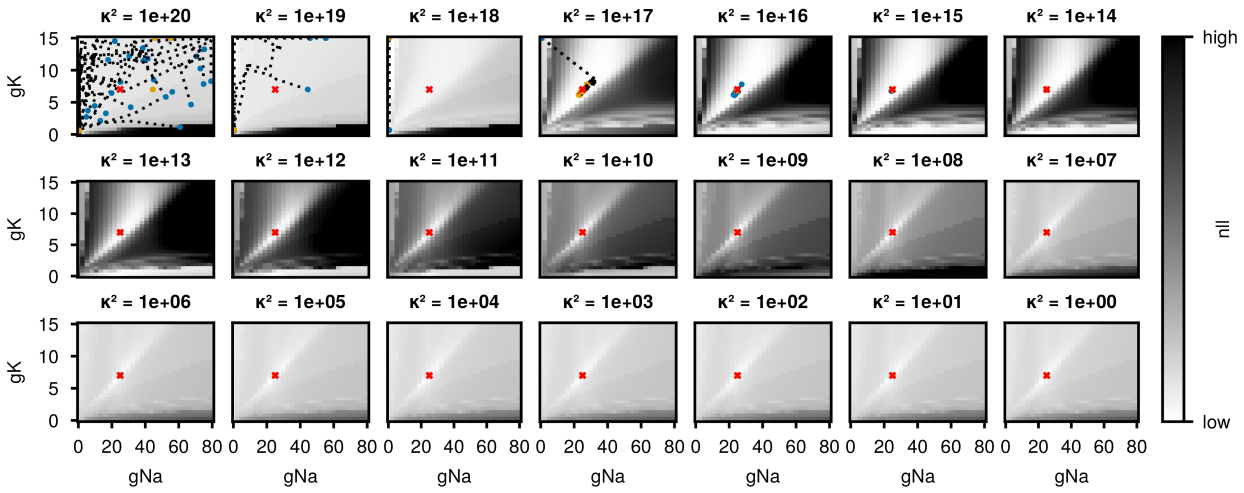

Figure A1. Parameter estimation for the sodium g_{Na} and potassium g_K conductance of a HH model. Parameter optimization during all stages of diffusion tempering for a random subset of 25 initializations. Optimization trajectories (dotted) in each likelihood landscape are shown from start (blue) to convergence (orange). Very high loss values were clipped for better visual clarity.

A8. Additional Experiments

A8.1. Effectiveness of different tempering rates

In our work we mainly focused on the convergence reliability of diffusion tempering. For this reason we picked a tempering schedule that was very simple and reliable during our testing – linearly reducing the $\log \kappa$ from 20 and 0 with a step size of 1. Despite its simplicity this already performed much better than our baselines without any further tuning. Since this schedule is very costly in the number of iterations needed (Table 1), we also investigated the effect of different schedules, i.e. with much steeper tempering rates or exponentially decreasing $\log \kappa$ on the cost of diffusion tempering for the HH_1 model with 2 parameters.

We found that much steeper, hence shorter schedules that decrease $\log \kappa$ between 20 and 0 can save about 20% iterations, while affecting the performance not at all or very little (Figure A5). In particular we found that taking more than 16 tempering steps yields no additional benefits for the HH_1 model with 2 parameters. Additionally, similar to Section 4.2 most of the trajectories started to converge to the true parameters around $\kappa = 10^{15}$, for many of the schedules.

We also tested an exponentially decaying the $\log_{10}(\kappa)$ where at each tempering step κ was set according to:

$$\mathcal{T}(i, \kappa_0) = 10^{\kappa_0 \exp(-i/\tau)}, \tag{27}$$

with initial diffusion $\kappa_0 = 20$, decay constant $\tau = 5$ and tempering steps $i \in \{0, 1, ..., 20\}$. We found that this exponential tempering schedule can save around 30% of the computation cost, while only taking a small hit to performance. Combined with the early stopping criterion we implemented (Appendix A2), this brings the cost of diffusion tempering down to that of the original Fenrir method, while retaining the superior convergence.

While we only considered a small number of possible tempering schemes, our experiments show that this can already reduce the cost of our algorithm by a third. This suggests future work on more efficient tempering schemes will reduce the cost of diffusion tempering even further.

A8.2. Injecting gradient noise during optimization

It is known that injecting noise into the gradient updates can effectively regularize the optimization procedure in challenging loss landscapes (Neelakantan et al., 2017). While Fenrir and hence diffusion tempering are are noise-free methods, we also

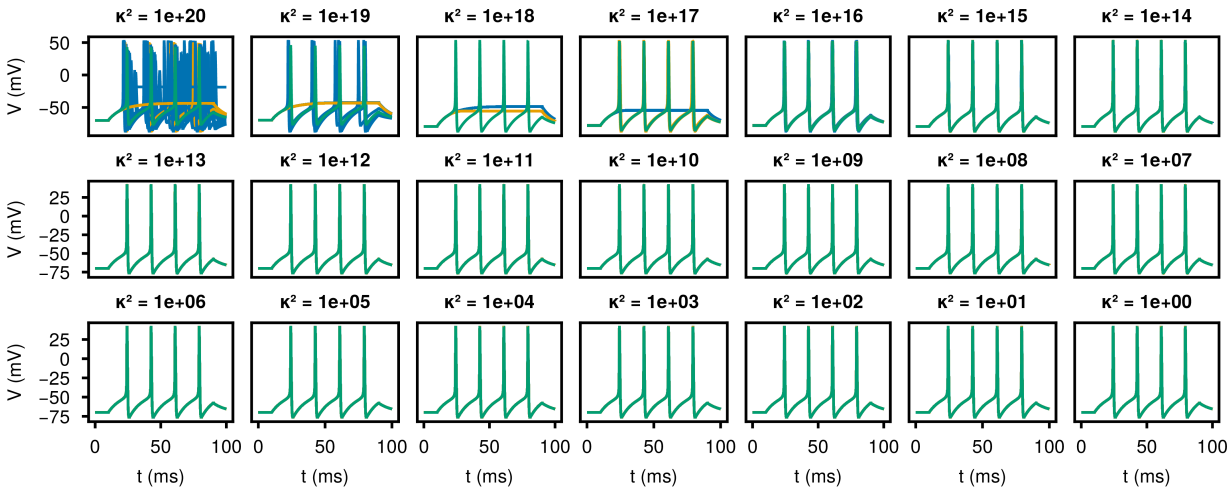

Figure A2. Parameter estimation for the sodium g_{Na} and potassium g_K conductance of a HH model. ODE solutions for the start (blue) to converged (orange) parameters at all stages of diffusion tempering for a random subset of 25 initializations.

investigated how adding gradient noise to RK compares to diffusion tempering for the pendulum.

For this purpose we optimized the pendulum as described in (Appendix A4), but Gaussian noise ϵ_n of the form $\epsilon_n \sim \mathcal{N}(0, \sigma^2 e^{-n/\tau})$, with decay rate $1/\tau$ and initial variance σ^2, was added to the gradients of the log likelihood at every parameter update step n.

The addition of gradient noise with varying scales and decay rates improved the performance of the noise-free RK variant almost across the board, even doubling the convergence rates in some instances (Table 7). Adding noise with a variance at around 5 performed the best reaching a convergence rate of 0.64 for non decaying noise levels. While this improves upon the noise-free RK baseline, this still falls short of the convergence rates that both Fenrir and diffusion tempering achieve however, with 0.75 and 1.00 respectively.

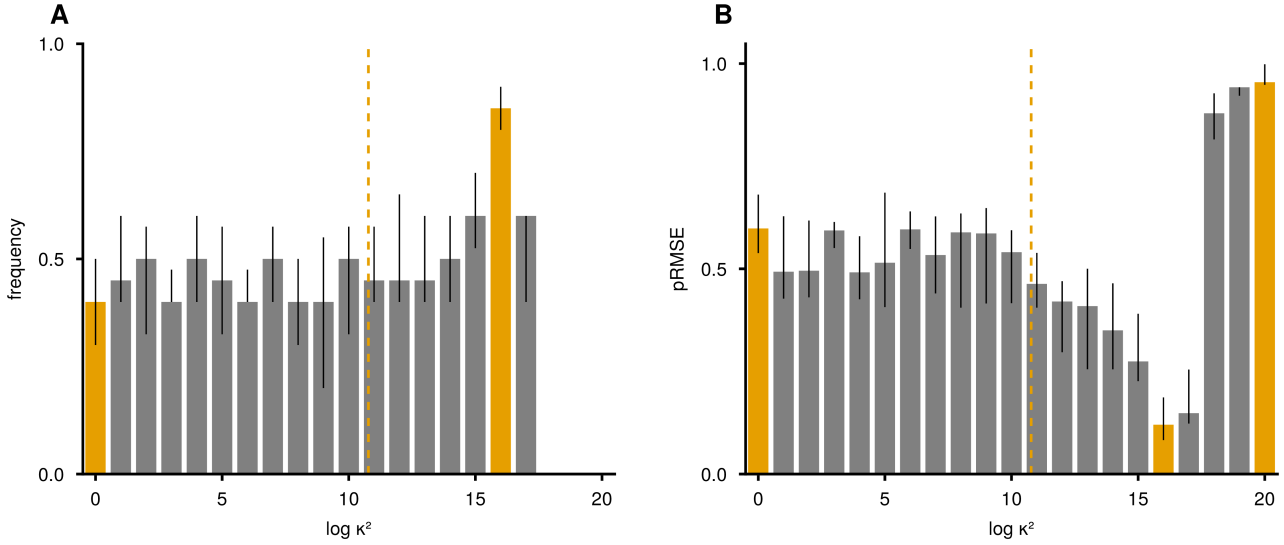

Figure A3. Convergence for a 2 parameter HH model for a grid of different fixed κs. The parameters selected for Figure 5 are highlighted in orange and the optimal diffusion parameter $\log \kappa = 11.6$ (dashed). Histogram shows the medians, with black bars indicating quartiles for 100 runs split into groups of 10. **A** Fraction of runs with $pRMSE < 5\%$. **B** pRMSEs for each diffusion.

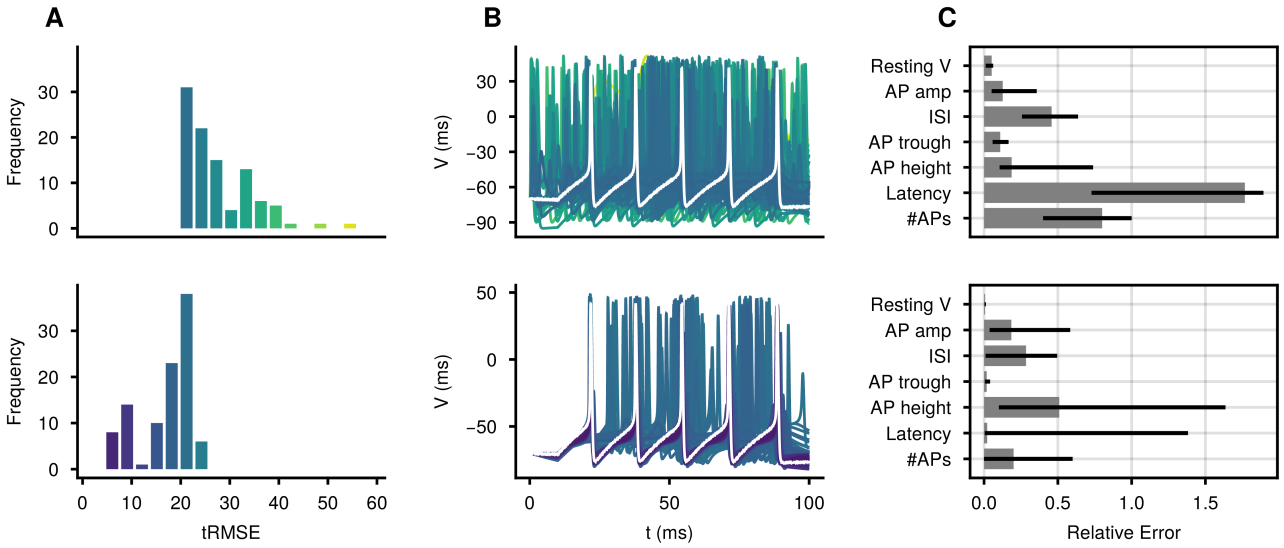

Figure A4. Parameter inference for a six parameter HH model for 100 initializations of Fenrir (top) and our method (bottom). **A** Distribution of tRMSEs. Almost all parameter estimates returned by our method have better tRMSEs than Fenrir. **B** Voltage traces for the inferred parameters, colored by their tRMSE. The observation is shown in white. The traces recovered by Fenrir seem almost random. **C** Median errors for seven commonly used electrophysiological features (as defined in Appendix A3). Quartiles are highlighted with error bars. Our method is able to reproduce qualitative aspects of the data better than Fenrir.

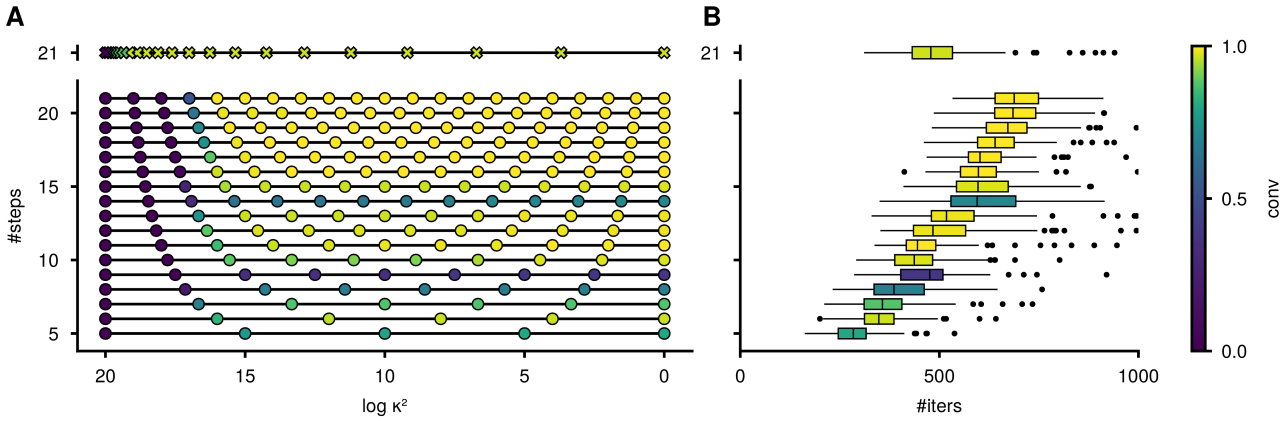

Figure A5. Comparing tempering schedules with different numbers of tempering steps between $\log \kappa = 20$ and $\log \kappa = 0$ for the HH_1 model with 2 parameters. **A** Percentage of correctly inferred parameters at each stage of tempering during optimization. Linear decrease of $\log \kappa$ (dots) vs. exponential decrease (crosses). There is a clear trade-off between the number of tempering steps and convergence, but taking more than 16 steps yields no additional benefits. **B** Cost to performance trade-off for different tempering schedules. Exponential schedules are more efficient at the same number of tempering steps.

Table 7. Effect of injecting gradient noise with different scale, σ^2, and decay rates, τ, during optimization of a pendulum on the convergence. While injecting gradient noise improves convergence compared to the noise-free RK baseline (first row), both Fenrir and diffusion tempering still converge more reliably.

σ^2	τ	CONV	PRMSE	ITER
0.00	–	0.32	1.42 ± 1.09	33.47 ± 18.11
0.01	–	0.35	1.40 ± 1.11	33.79 ± 12.84
0.05	–	0.34	1.41 ± 1.10	29.89 ± 10.77
0.10	–	0.37	1.40 ± 1.13	27.30 ± 11.84
0.50	–	0.49	1.14 ± 1.13	15.19 ± 10.72
1.00	–	0.63	0.77 ± 1.05	15.52 ± 14.12
5.00	–	**0.64**	0.51 ± 0.88	13.01 ± 8.87
10.00	–	0.39	0.64 ± 0.93	10.43 ± 7.19
50.00	–	0.45	0.49 ± 0.83	11.66 ± 8.26
100.00	–	0.28	$\mathbf{0.44 \pm 0.74}$	$\mathbf{10.26 \pm 7.42}$
0.00	5	0.32	1.42 ± 1.09	33.47 ± 18.11
0.01	5	0.31	1.45 ± 1.08	35.94 ± 18.58
0.05	5	0.33	1.45 ± 1.10	32.58 ± 13.39
0.10	5	0.35	1.40 ± 1.11	33.65 ± 12.85
0.50	5	0.43	1.24 ± 1.13	22.23 ± 14.93
1.00	5	0.53	0.98 ± 1.08	22.68 ± 18.51
5.00	5	**0.59**	0.67 ± 0.99	15.23 ± 13.33
10.00	5	0.45	0.51 ± 0.84	13.11 ± 13.49
50.00	5	0.40	$\mathbf{0.51 \pm 0.79}$	$\mathbf{11.36 \pm 11.60}$
100.00	5	0.41	0.58 ± 0.86	12.54 ± 14.20
0.00	10	0.32	1.42 ± 1.09	33.47 ± 18.11
0.01	10	0.33	1.43 ± 1.10	33.42 ± 12.90
0.05	10	0.32	1.44 ± 1.09	34.21 ± 11.00
0.10	10	0.34	1.39 ± 1.11	33.40 ± 12.70
0.50	10	0.46	1.22 ± 1.14	18.89 ± 13.20
1.00	10	0.50	1.08 ± 1.10	18.58 ± 14.95
5.00	10	**0.61**	0.58 ± 0.94	14.28 ± 12.63
10.00	10	0.58	0.55 ± 0.90	12.95 ± 10.90
50.00	10	0.32	0.41 ± 0.71	$\mathbf{9.96 \pm 8.40}$
100.00	10	0.37	$\mathbf{0.40 \pm 0.72}$	10.92 ± 8.47
FENRIR		0.75	0.20 ± 0.37	110.04 ± 61.70
OURS		**1.00**	$\mathbf{0.00 \pm 0.00}$	696.22 ± 78.10