

Efficient Inference on a Network of Spiking Neurons using Deep Learning

Nina Baldy^a, Marmaduke M. Woodman¹, Martin Breyton¹, Viktor K. Jirsa ^{†1,*}, Meysam Hashemi ^{†1,*}

^a*Aix Marseille Univ, INSERM, INS, Inst Neurosci Syst, Marseille, France*

Abstract

The process of making inference on networks of spiking neurons is crucial to decipher the underlying mechanisms of neural computation. Mean-field theory simplifies the interactions between neurons to produce macroscopic network behavior, facilitating the study of information processing and computation within the brain. In this study, we perform inference on a mean-field model of spiking neurons to gain insight into likely parameter values, uniqueness and degeneracies, and also to explore how well the statistical relationship between parameters is maintained by traversing across scales. We benchmark against state-of-the-art optimization and Bayesian estimation algorithms to identify their strengths and weaknesses in our analysis. We show that when confronted with dynamical noise or in the case of missing data in the presence of bistability, generating probability distributions using deep neural networks outperforms other algorithms, such as adaptive Monte Carlo sampling. However, this class of deep generative models may result in an overestimation of uncertainty and correlation between parameters. Nevertheless, this issue can be improved by incorporating time-delay embedding. Moreover, we show that training deep Neural ODEs on spiking neurons enables the inference of system dynamics from microscopic states. In summary, this work demonstrates the enhanced accuracy and efficiency of inference on networks of spiking neurons when deep learning is harnessed to solve inverse problems in neural computation.

Keywords: Mean-field models, Bayesian inference, Deep neural networks, Simulation-based-inference, Hamiltonian Monte Carlo, Optimization, Bistability

Abbreviations:

MF, Mean-Field; QIF, Quadratic Integrate-and-Fire; SBI, Simulation-Based Inference; NPE, Neural Posterior Estimation; MCMC, Markov chain Monte Carlo; HMC, Hamiltonian Monte Carlo; MAP, Maximum a Posteriori; DE, Differential Evolution; PSO, Particle Swarm Optimization; BO, Bayesian Optimization; ODEs, Ordinary Differential Equations; SDEs, Stochastic Differential Equations; Neural ODEs, Neural Ordinary Differential Equations

*Corresponding author

Email addresses: viktor.jirsa@univ-amu.fr (Viktor K. Jirsa [†]), meysam.hashemi@univ-amu.fr (Meysam Hashemi [†])

[†]These authors contributed equally.

1. Introduction

Neural computation involves the complex information processing performed by spiking neurons, where their interactions within neural circuits contribute to higher-level computations and cognitive functions. Consequently, the cornerstone of theoretical neuroscience lies in the use of mathematical models to understand the intricacies of neural computation (Dayan and Abbott, 2005; Hertz, 2018). This approach enables researchers to explore the complexities of neural processes, thereby revealing insights into the mechanisms that drive brain function and behavior. These mathematical models can range from simple idealized representations of single neurons (Hopfield, 1982; Izhikevich, 2003) to complex network models that simulate the interactions within neural circuits (Marder, 1998; Sussillo, 2014; O’Leary et al., 2015; Bittner et al., 2021). The biological neural computation forms the basis for the brain’s ability to execute various inference tasks, such as recognizing patterns, making decisions, and generating responses to stimuli. This capability relies on the collective behavior of spiking neurons, where the interactions and coordination among these neurons within neural circuits enable the brain to process, integrate, and interpret sensory information, ultimately leading to higher-level cognitive functions and adaptive behavior (Kandel et al., 2000; Gerstner et al., 2014; Friston and Kiebel, 2009; Friston et al., 2017).

Mean-Field (MF) models serve as effective computational abstractions that represent the collective behavior of large populations of neurons, while maintaining a degree of mathematical tractability (Amari, 1977; Wilson and Cowan, 1973; Jirsa and Haken, 1996; David and Friston, 2003; Deco et al., 2008; Hutt et al., 2015; Coombes and Byrne, 2018; Bandyopadhyay et al., 2021; Cook et al., 2022). Hence, MF models facilitate the study of information processing and computation within the brain, which underlie cognitive processes such as perception, memory, and learning. Nevertheless, deriving and validating MF models from spiking neural networks present many challenges, particularly when assessing how the interaction between individual neurons and parameters leads to macroscopic behavior that aligns with the averaged activity of neural populations.

Recently, an analytically-driven MF model of spiking neurons has been formulated, which can effectively describe all potential macroscopic dynamical states of the network, including states of synchronous spiking activity (Montbrió et al., 2015). However, the operation of such complex systems (governed by coupled non-linear differential equations) is determined by the selection of (biological or phenomenological) parameters, which when set in a specific configuration, give rise to a measurable signature of a computation (Achard and De Schutter, 2006; Sussillo, 2014). Analyzing MF models and comparing their emergent dynamics against a network of neurons involves solving inverse problems to ascertain the optimal parameter setting. This process requires swift outcomes to inform real-time decisions and also to deal with observation and dynamical noise. Yet, even in the simplest models, there can be a degenerate relationship between the model parameters and its overall emergent function (Edelman and Gally, 2001; Prinz et al., 2004; Alonso and Marder, 2019).

Maintaining the inter-dependency between parameters by traversing across scales adds an additional layer of complexity to the validation process. It is crucial to distinguish between genuine, biologically relevant correlations, and artificial correlations that may arise from the inference process or modeling assumptions. Furthermore, due to the intricate nature of computation within neural circuits, it becomes intractable to analytically derive MF models that include more biological realism (such as adaption, neuromodulation, extra-synaptic transmission, and E/I ratios). Statistical inference offers an efficient and adaptable approach to solving the inverse problem by identifying approximate parameter distributions that are responsible for generating computations in a biologically realistic model (Achard and De Schutter, 2006; Liepe et al., 2014; Lueckmann et al., 2017; Gonçalves et al., 2020; Bittner et al., 2021; Mlynarski et al., 2021).

The performance of statistical inference algorithms depends on the task, and there is no universally best algorithm for different inverse problems. Therefore, we conducted a benchmarking analysis against state-of-the-art optimization and Bayesian estimation algorithms to discern their respective advantages and limitations. In practice, optimization

methods are commonly used to quickly determine unknown quantities through a single point estimate (Mendes and Kell, 1998; Nocedal and Wright, 1999; Kelley, 1999; Floudas and Gounaris, 2009). These methods involve iteratively adjusting parameters to minimize or maximize an objective function, scoring the model’s performance against observed data (e.g. through minimizing distance errors or maximizing correlation; Banga and Balsa-Canto (2008); Tashkova et al. (2011); Svensson et al. (2012); Hashemi et al. (2018)).

The Bayesian approach offers a principled method for making inferences, predictions, establishing relationships between parameters, and quantifying uncertainty in the decision-making process (Gelman et al., 1995; Bishop, 2006; Gelman et al., 2020; van de Schoot et al., 2021). In Bayesian modeling, all model parameters are treated as random variables and their values are subject to variation based on their underlying probability distributions. Such probabilistic techniques provide the full posterior distribution of unknown quantities hidden in the underlying data generating process. The uncertainty and inter-dependency in Bayesian estimation are naturally quantified by assigning a probability distribution to the parameters (known as the prior distribution), which is then updated based on the information provided by the data (referred to as the likelihood function). To conduct a full Bayesian procedure, the state-of-the-art MCMC method is adaptive Hamiltonian Monte Carlo (HMC; Duane et al. (1987); Neal (2010); Hoffman and Gelman (2014)), which utilizes gradient information to avoid random walk behavior, enabling efficient sampling from high-dimensional distributions that may exhibit strong correlations (Betancourt, 2017).

Simulation-Based Inference (SBI; Cranmer et al. (2020); Brehmer (2021)) or likelihood-free inference (Papamakarios and Murray, 2016; Brehmer et al., 2020) leverages deep generative models to conduct approximate Bayesian estimation, using low-dimensional data features that are generated by random simulations (Gonçalves et al., 2020; Lueckmann et al., 2021; Boelts et al., 2022). This method, transforms a simple base probability distribution (prior) into a more complex distribution (posterior), through a sequence of invertible transformations (i.e., Normalizing Flows; Rezende and Mohamed (2015); Papamakarios et al. (2019a)). Notably, it allows for direct estimation of joint posterior distributions, bypassing the need for MCMC sampling (Greenberg et al., 2019; Papamakarios et al., 2019b). Moreover, SBI has the potential to capture parameter degeneracy or multi-modalities in the distributions (Hashemi et al., 2023).

Data-driven methods for learning dynamical models from time-series data have been extensively researched for several decades (Juang, 1994; Ljung, 1998; Brunton et al., 2016; Linderman et al., 2017; Duncker et al., 2019; Koppe et al., 2019; Sip et al., 2023). Instead of relying on discretized maps, Neural Ordinary Differential Equations (Neural ODEs; Chen et al. (2018)) form a new family of deep neural network models for modeling continuous-time dynamics. Neural ODEs define the vector fields and ODE solution as a black-box differential equation solver, allowing for uncovering the dynamics of a system even when the governing equations are unknown (Dupont et al., 2019; Biloš et al., 2021). This data-driven approach involves parameterizing system dynamics as continuous functions, enabling smooth and uninterrupted modeling of temporal evolution (Yan et al., 2019; Kim et al., 2021). Neural ODEs naturally adapt to varying time intervals and can accommodate fluctuations in the frequency of data observations (Zhu et al., 2022; Goyal and Benner, 2023).

Through an exploration of the aforementioned methods, we demonstrate that global optimization algorithms, such as Differential Evolution algorithm, offer fast and accurate point estimation of the true generative parameters when the dynamical noise is absent. However, when dealing with dynamic evolution that are subject to noise, SBI using deep neural density estimators emerges as the superior approach, outperforming other algorithms, such as HMC sampling. Additionally, when dealing with missing data (such as population firing rate) in state-space modeling, HMC fails to capture the dynamics of bistable switching behaviour. Instead, SBI is able to accurately recover the diverse dynamics in the phase-space representation. Nevertheless, this approach may lead to an overestimation of uncertainty and correlation between parameters, which can be mitigated by using time-delay embedding technique to improve the results. We validate this by employing a MF model of quadratic

integrate-and-fire (QIF) neurons, which demonstrates that the inter-dependencies between parameters are maintained when traversing across scales. Finally, we demonstrate that training deep Neural ODEs on spiking neurons enables the inference of vector fields at macroscopic level. This allows for the prediction of emergent behaviors and system dynamics based on the microscopic state of the spiking neurons.

2. Materials and methods

2.1. A mean-field model of spiking neurons

The quadratic integrate-and-fire (QIF) neurons are a class of simplified computational models that are extensively used to study the dynamics of spiking neurons (Gerstner and Kistler, 2002; Izhikevich, 2007). In the QIF model, the membrane potential of neuron evolves according to a quadratic differential equation until it reaches a threshold, at which point the neuron emits a spike and the potential is reset.

Montbri   et al. (2015) have proposed a MF model that accurately describes macroscopic states of populations of firing neurons. This mechanistic model derives the firing rate equations for networks of heterogeneous, all-to-all coupled QIF neurons, which is exact in the thermodynamic limit, i.e. for large numbers of neurons. Specifically, when considering specific distributions of heterogeneity, the Lorentzian ansatz yields a non-linear system of two ordinary differential equations for the firing rate r and mean membrane potential v of the neuronal population:

$$\dot{r} = 2rv + \Delta/\pi \quad (1a)$$

$$\dot{v} = v^2 - \pi^2 r^2 + Jr + \eta + I(t) \quad (1b)$$

where η is the average excitability, J denotes the synaptic weight, and Δ indicates the spread of the neuronal excitability distribution in the neural population. Depending on the parameter settings, the phase diagram exhibits three qualitatively distinct regions: a single stable node, which represents a low-activity state; a single stable focus (spiral), which generally corresponds to a high-activity state; and a region of bistability, where both low and high firing rates can coexist (see Figure S1). This model has succeeded in establishing an exact correspondence between the time evolution of firing rate of the network and the underlying microscopic state of the spiking neurons (Montbri   et al., 2015).

2.2. Inference methods

We validate the results by comparing them with various parameter estimation inference methods. To evaluate these methods, we use synthetic data generated from the MF model and a network of QIF neurons. Inference methods investigated in this work can be broadly divided into two classes:

(i) Optimization methods that return a point estimate of best fit based on the minimizing of a cost function, such as chi-squared error criterion defined by

$$\chi^2(\theta) = \sum_{i=1}^{N_t} (\hat{x}(t_i, \theta) - x(t_i))^2, \quad (2)$$

where $x(t_i)$ denotes the observed data at time points t_i with $i \in \{1, 2, \dots, N_t\}$, and $\hat{x}(t_i, \theta)$ represents the corresponding model prediction. Here $\theta \in \{\eta, J, \Delta\}$ is the set of unknown parameters, and the set of observation combines activity of both r, v (unless it is missing). Assuming no prior information and a Gaussian likelihood function with uncorrelated noise, this casts as a Maximum Likelihood Estimation (MLE) problem (Hashemi et al., 2018).

(ii) Bayesian methods return a posterior distribution of parameters, $p(\theta | x)$, which represents an ensemble of parameter sets that are plausible given the observed data. Given the data x and model parameters θ , Bayes rule defines the posterior distribution as

$$p(\theta | x) = \frac{p(\theta)p(x | \theta)}{p(x)}. \quad (3)$$

The prior information $p(\theta)$ is typically determined before seeing the data (through beliefs and previous evidence). The likelihood function $p(x | \theta)$ represents the probability of some observed outcomes given a certain set of parameters (the information provided by the observed data). The denominator $p(x) = \int p(x | \theta)p(\theta)d\theta$ represents the model evidence or marginal likelihood, which amounts to simply a normalization factor.

From the optimization methods, we use the global search algorithms that incorporate a bio-inspired random search principle: Differential Evolution (DE; [Storn and Price \(1997\)](#); [Price \(1999\)](#)), and Particle Swarm Optimization (PSO; [Kennedy and Eberhart \(1995\)](#); [Eberhart and Kennedy \(1995\)](#)). These algorithms do not require an initial guess for the parameters or the gradient information of the objective function ([Hashemi et al., 2018](#)). We also consider Bayesian Optimization (BO; [Snoek et al. \(2012\)](#); [Shahriari et al. \(2015\)](#)), an algorithm that constructs a probabilistic model for the objective function using Gaussian processes. This approach allows for the integration of uncertainty in the optimization process.

From the Bayesian methods, we compare the results of two state-of-the-art Bayesian computation algorithms: Hamiltonian Monte Carlo (HMC; [Duane et al. \(1987\)](#); [Neal \(2010\)](#)) which is unbiased and exact in infinite runs, and a Simulation-Based Inference (SBI; [Cranmer et al. \(2020\)](#); [Brehmer \(2021\)](#)) which approximates the posterior using deep generative models. Here, generative modeling is an unsupervised machine learning method to model a probability distribution based on the samples drawn from that distribution. From the Bayesian methods, we also report the results of Maximum a Posteriori (MAP) estimation.

2.3. Hamiltonian Monte Carlo

Markov chain Monte Carlo (MCMC) is a powerful class of computational algorithms used for sampling from a distribution, in which the sampling process does not require knowledge of the entire distribution, making it a versatile tool. ([Andrieu et al., 2003](#); [Murphy, 2022](#); [McElreath, 2020](#)). MCMC is unbiased and asymptotically exact, in the limit of infinite runs. Hamiltonian Monte Carlo (HMC; [Duane et al. \(1987\)](#); [Neal \(2010\)](#)) is a gradient-based MCMC designed to avoid random walk behaviour, and it can efficiently sample from high-dimensional distributions that may exhibit strong correlations ([Betancourt, 2017](#)). However, the efficiency of HMC is sensitive to the algorithm parameters.

In this study we use a self-tuning variant of HMC (known as the No-U-Turn Sampler; ([Hoffman and Gelman, 2014](#))) from a high-level statistical modeling tool called Stan ([Carpenter et al., 2017](#)). In particular, the NUTS calibrates the number of steps and step-size of the leapfrog integrator (in solving the Hamiltonian equations of motion) during a warm-up phase to achieve a target Metropolis acceptance rate. For more details see [Betancourt \(2013\)](#); [Baldy et al. \(2023\)](#). Moreover, Stan offers alternative methods such as MAP estimation using L-BFGS optimization, automatic differentiation for efficient gradient computation, and various diagnostics to assess the convergence of the inference process (see <https://mc-stan.org>).

2.4. Simulation-Based Inference

Simulation-Based Inference (SBI) conducts efficient Bayesian inference for complex models when the calculation of the likelihood function is either analytically or computationally intractable ([Cranmer et al., 2020](#); [Brehmer, 2021](#)). In simulator models, where the data can be generated through stochastic computer programming, SBI leverages repeated simulations from the generative model and employs probabilistic machine learning to estimate a target probability distribution. Instead of directly sampling from distributions using MCMC or explicitly evaluating the likelihood function, SBI overcomes these challenges by using deep neural density estimators, such as Masked Autoregressive Flows ([Papamakarios et al., 2017](#))). These density estimators learn an invertible transformation between the distributions of parameters and low-dimensional data features at a very low computational cost, to efficiently sample from distributions.

Taking prior distribution $p(\theta)$ over the parameters θ , a limited number of N simulations are generated for training step from the generative model i.e., $\{(\theta_i, x_i)\}_{i=1}^N \sim p(\theta, x)$. After the training step, we are able to quickly estimate the approximated posterior $q_\phi(\theta | x)$ with learnable parameters ϕ , so that for the observed data x_{obs} : $q_\phi(\theta | x_{obs}) \simeq p(\theta | x_{obs})$. For more details see [Gonçalves et al. \(2020\)](#); [Hashemi et al. \(2023\)](#).

The methods for SBI often include a sequential training procedure, which adaptively guides simulations to yield more informative estimates ([Papamakarios et al., 2019b](#); [Lueckmann et al., 2019](#); [Durkan et al., 2020](#); [Wiqvist et al., 2021](#); [Deistler et al., 2022](#)). In particular, Sequential Neural Posterior Estimation (SNPE; [Greenberg et al. \(2019\)](#); [Gonçalves et al. \(2020\)](#)) dynamically refines the proposals, network weights, and posterior estimates to learn the relationship between model parameters and the observed summary statistics of the data. In this study, we used SNPE with a single round to take advantage of an amortized strategy; After incurring an initial computational cost for the simulation and training steps to learn all the joint posterior distributions, then the posterior can be quickly estimated from any new observations (by a forward pass through neural networks) without any additional computational overhead or further simulations.

2.5. Time-delay embedding

Time-delay embedding is a commonly used technique for characterizing dynamical systems based on limited measurements, time series analysis, and prediction ([Takens, 1981](#)). In time-delay embedding, the reconstruction of a latent high-dimensional system relies on incorporating incomplete measurements along with a temporal history of preceding measurements to create a comprehensive representation ([Kennel et al., 1992](#); [Hirsh et al., 2021](#)). In a subsequent analysis, we challenge the inference process by assuming that the firing rate activity r is not directly observed (missing data problem). Instead, to improve the inference, we recovered the latent time-series r_{rec} from the observed activity v , which is coupled to r according to Eq. 1. By expanding on a method introduced by [Abarbanel et al. \(1994\)](#), which primarily focuses on predicting physical variables in time-delay embedding, we removed the assumption that we have the access to training data points from the hidden time series. Instead, we leverage our understanding of the generator to simulate data pairs (r, v) that can be used for training purposes.

Time-delay embedding requires the setting of hyperparameters, such as the delay (time lag), T , and the dimension of the embedding space, d . These hyperparameters are typically set prior to training, often based on the minimization of mutual information to determine the appropriate delay and the false nearest neighbors method to determine the number of embedding dimensions ([Kennel et al., 1992](#); [Tan et al., 2023](#)). However, in the present application, we have found that the hyperparameters suggested by these methods were not the most effective in achieving an accurate fit. Instead, we select hyperparameters that minimize the mean square error of the fit to simulated data, with $T = 160$ points (i.e., 0.16 sec) and $d = 12$. To do this, a set of 100 pairs of coupled time series (r, v) was simulated, with an observation noise intensity of 0.1 and varying parameters Δ, η, J . These pairs were then used to infer regression coefficients that closely match r to the delay embedding space representation of v .

2.6. Neural ODEs

Neural ODEs are a set of machine learning techniques that allows to reconstruct the phase-space of a dynamical system from a training set of observations ([Chen et al., 2018](#)). In a first analysis, we trained a Neural ODE on time-series from the MF model, and then applied the same method to the data generated by a network of QIF spiking neurons. If $\mathbf{x}(t)$ is the vector of state variables governed by the dynamical system $\dot{\mathbf{x}} = f(\mathbf{x}, \theta, I_{ext})$, one can use a Neural ODE to approximate the function f with a artificial neural network F_ϕ yielding the corresponding dynamical equation $\dot{\mathbf{x}} = F_\phi(\hat{\mathbf{x}}, \theta, I_{ext})$. F_ϕ is learned through

backpropagation, minimizing the loss function for each training example of length T:

$$L_\phi(\hat{x}, x) = \sum_{t=0}^T (\hat{x}(t, \theta) - x(t))^2$$

where \hat{x} and x are times-series generated by F_ϕ and f , respectively, with the same integration scheme (e.g., Heun's method). In this study, a Neural ODE was implemented in JAX ([Bradbury et al., 2018](#)) by constructing a multi-layer-perceptron (MLP) with one hidden layer of 16 units with hyperbolic tangent activation functions. The cost function was the average loss functions across training examples $\mathcal{L}_\phi = \frac{1}{N*T} \sum_{n=1}^N L_{\phi,n}$. At each training iteration, the initial (r, v) values of each segment is fed to the Neural ODEs and the forecast trajectory according to current state of the F_ϕ enters the cost function, then minimized through backpropagation using the Adam optimizer.

2.7. Software and algorithmic setup

We use Python implementations of the optimization and Bayesian algorithms: DE from SciPy ([Virtanen et al., 2020](#)), PSO from the toolkit PySwarms ([Miranda, 2018](#)), and BO from the package Bayesian Optimization ([Nogueira et al., 2014](#)). DE is set with population size of 10, the maximum iterations of 500. PSO is set with 10 particles, and 500 iterations. BO is set with 50 random initialisation, 500 iterations, and kappa=100 (higher values increase exploration).

The MAP estimation is obtained using a quasi-Newton L-BFGS optimizer from Python interface to Stan ([Carpenter et al., 2017](#)), an open-source probabilistic programming language for Bayesian modeling and statistical inference. We also use Stan's implementation of the NUTS for fully Bayesian estimation by HMC, with target Metropolis acceptance rate of 0.8, maximum tree depth of 10, warm-up iterations of 1000, and 500 sampling phase iterations.

Finally, for SBI we used Sequential Neural Posterior Estimation (SNPE; [Greenberg et al. \(2019\)](#); [Gonçalves et al. \(2020\)](#)) from the python SBI toolkit ([Tejero-Cantero et al., 2020](#)). SNPE was run for a single round with 100,000 simulations, unless specified otherwise. For training step, we used a MAF, with 5 autoregressive layers, each with two hidden layers of 50 units. The set of data features includes the statistical moments of time-series up to the fourth order (mean, standard deviation, skewness, and kurtosis), and the peak properties (number and location of first peak, as shown in [Figure S2](#)).

Parameter bounds for optimization, and uniform prior for Bayesian inference, were set as: $\Delta \in [0.1, 5]$, $\eta \in [-10, -3]$, $J \in [5, 20]$. For the simulation of Eq. 1, we used Euler-Maruyama integration for $t = 100 \text{ sec}$ with a time step of $dt = 1 \text{ msec}$. The ground truth parameters were set as $\eta = -4.6$, $J = 14.5$, and $\Delta = 0.7$, ensuring that the system is in a bistable regime, unless specified otherwise. Moreover, a step current with amplitude of 3 v from time 30 sec to 60 sec was added to the v variable. Each simulation took around 0.01 sec to run using a Just-In-Time (JIT) compiler. To generate stochastic dynamics, a zero-mean Gaussian noise with $\sigma = 0.1$ was added to the both r , and v variables in Eq. 1. The prior on dynamical noise was set as $\sigma \in [0, 1]$. For comparison with the MF model, we ran a network of 10^4 all-to-all connected QIF neurons using the Brian simulator ([Stimberg et al., 2019](#)).

The model simulation and parameter estimation were performed on a Linux machine with 3.60 GHz Intel Core i7-7700 and 8 GB of memory.

3. Results

We present the results of the optimization and Bayesian algorithms for three cases: (i) inferring from exact deterministic synthetic data, (ii) inferring from stochastic synthetic data with dynamic evolution affected by external noise, and (iii) inferring with stochastic synthetic data when the activity for r is unknown (missing data). We report and compare the

goodness-of-fit using the root mean square error (RMSE) to the true (observed or hidden) time-series and parameters, the variance in the estimation process and the computational cost.

Before running inference process, we conduct a sensitivity analysis to assess the structural identifiability of the parameters. This analysis involves calculating local sensitivity coefficients, which measure the effect of small adjustments in a parameter on the model’s output while keeping all other parameters constant. A small change in a sensitive parameter leads to a significant alteration in the model output, indicating its identifiability. Conversely, when there are no alterations in the model output despite adjustments in a parameter, it suggests the parameter’s non-identifiability. To assess the identifiability of the parameters, we used the profile likelihood (Raue et al., 2009; Wieland et al., 2021) and Hessian matrix, a metric describing the local curvature of a function based on its second partial derivatives (Hashemi et al., 2018, 2023). Our results indicate that all three parameters Δ , η , and J can be identifiable, however, the profile likelihood obtained using Eq. 2 reveals the presence of numerous local minima (see Figure S3). This highlights the difficulty in attaining the global minimum during the inference process.

3.1. Inference on deterministic synthetic data

Here, we compare the inference results of different algorithms when we have complete observations of both state variables (r, v), and the system operates without any external noise (see Figure 1). The observed time-series and trajectories in phase-plane, which exhibit a bistability between a stable node and a stable focus, are illustrated in Figure 1A. From the results demonstrated in Figure 1B, it can be seen that all the algorithms qualitatively reproduce the bistability behaviour in the phase-plane. However, when evaluating their performance by estimating the true generative parameters, all optimization methods, except DE, get stuck in a local minima (Figure 1C). The results indicate that DE, SBI and HMC algorithms correctly recover the ground-truth parameters, the former as an almost exact point estimate (Figure 1C, top panels), while the latter two, SBI and HMC, yield full posterior distributions (Figure 1C, bottom panels). In terms of uncertainty quantification, HMC offers a more informative posterior distribution compared to SBI (see Figure S4).

When it comes to matching the observed time-series, the optimization algorithms such as PSO and BO deviate significantly from true values, while other approaches equally retrieve an almost perfect fit to the observed time-series (see Figure 1D, top panel). Nevertheless, MAP largely fails to accurately estimate model parameters due to overfitting (Figure 1D, bottom panel). This type of overfitting emphasizes the importance of quantifying uncertainty to verify the reliability of inference, going beyond a point estimation. Note that for HMC and SBI, we report the results using posterior predictive check i.e., re-generating data using the parameters drawn from the estimated posterior and then comparing them with the observed data.

In terms of computational cost for inference, DE has a clear advantage with its rapid performance, typically taking less than a minute to complete (Figure 1E). Despite a high precision, the computational cost of HMC in this example is prohibitively expensive, taking almost a hundred hours to complete the inference process. On the other hand, SBI was terminated in nearly one and a half hours (including 100k random simulations, training, and sampling), making it approximately 60 orders of magnitude faster than HMC. Additionally, due to the amortized approach adopted by SBI, each sampling process takes less than one minute to estimate the joint posterior distributions from new data.

In summary, DE demonstrates superior speed and accuracy, but it only provides a point estimate. Among Bayesian methods, HMC is exact and provides certain estimates, but it is computationally prohibitive. On the other hand, SBI effectively provides accurate estimates along with associated uncertainties at a reasonable computational cost (see Table 1, Movie 1, and Movie 2).

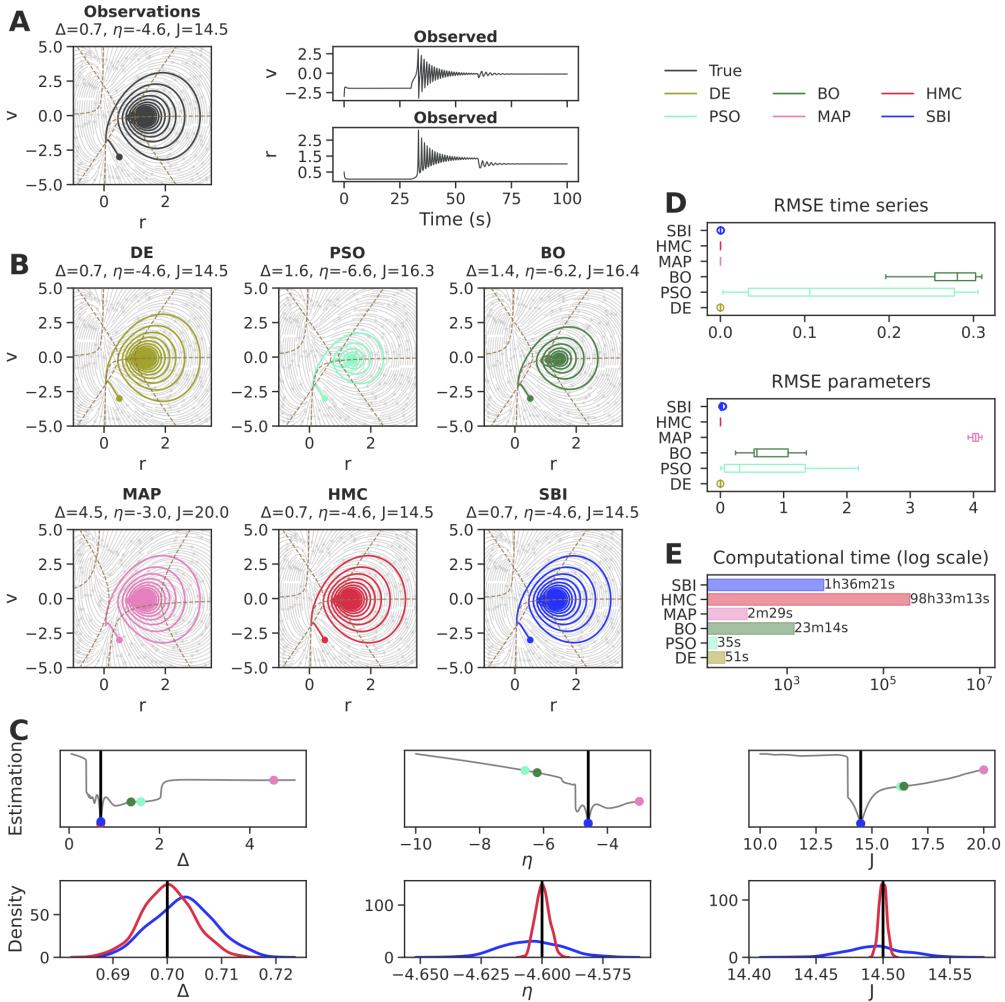


Figure 1. Inference on deterministic data. **(A)** The trajectories in phase-plane exhibiting bistable dynamics, and the corresponding time-series of firing rate (r) and membrane potential (v), used as the observations for inference (ground truth: $\Delta = 0.7$, $\eta = -4.6$, $J = 14.5$). **(B)** The estimated trajectories in phase-plane, along with the corresponding parameters displayed in the top panels. All the algorithms capture the bistable dynamics in a qualitative manner. **(C)** For parameters Δ , η , and J , the point estimations are displayed in the top panels along with the profile likelihood (in grey), while the full posterior distributions are shown in the bottom panels. HMC leads to more precise parameter estimates with lower uncertainty compared to SBI. **(D)** Accuracy in estimation based on the sum over RMSE values. The bootstrap uncertainty is calculated for time-series (top panel) and parameters (bottom panel) through multiple runs. **(E)** Computational cost for each inference algorithm. Overall, DE excels in both speed and accuracy, but it offers only a point estimate. HMC is exact and provides informative posterior estimates, but it is very computationally prohibitive. Rather, SBI effectively provides accurate estimates along with associated uncertainties at a reasonable computational cost.

3.2. Inference on stochastic synthetic data

The inherent randomness in stochastic systems introduces uncertainty into the behavior of the system under study, which makes it challenging to accurately identify its underlying dynamics. In particular, the presence of dynamical noise significantly increases the complexity of the inference process and necessitates the use of robust probabilistic methodologies that can effectively account for and handle the stochastic nature of the data.

Here, we report the results of inference on synthetic data with zero-centered Gaussian dynamical noise, where the intensity is $\sigma = 0.1$ (see [Figure 2](#)). The observed noisy time-series and trajectories in phase-plane, exhibiting a bistable behavior between a stable node and a stable focus, are illustrated in [Figure 2A](#). From the results demonstrated in [Figure 2B](#), we can see that all the algorithms qualitatively replicate the bistability behavior in the phase-plane. Yet, when assessing how well they recover the true generative parameters, only DE yields an accurate point estimate among the optimization algorithms ([Figure 2C](#), top panels).

In contrast to the deterministic data, SBI generates posteriors that are tightly centered on the ground-truth parameters, in comparison to the HMC sampling (see [Figure 2C](#), bottom panels, and [Figure S5](#)). A detailed comparison of convergence diagnostics indicates that both SBI and HMC methods provide ideal Bayesian estimation (see [Figure S6](#)). Regarding the proximity to the observed time-series and true parameters, both HMC and SBI offer a closer match compared to the other algorithms ([Figure 2D](#)). This highlights the challenges inherent in inferring stochastic systems via optimization algorithms, as the error metrics such as RMSE used to define the objective function may not reliably gauge accuracy.

In terms of computational efficiency for inference, the running time of all algorithms increases when the noise is present, except for HMC ([Figure 2E](#)). Nevertheless, when considering the entire process (including random simulations, training, and sampling), SBI remains approximately 8 orders of magnitude faster than HMC. Interestingly, SBI is also able to accurately and efficiently estimate the dynamical noise in the system (see [Figure S7](#)).

In summary, these results demonstrate that when considering overall accuracy in the presence of noise and bistability, uncertainty quantification, and computational cost, the SBI outperforms all other algorithms, including HMC (see [Table 2](#), [Movie 3](#), and [Movie 4](#)).

Next, we investigate how varying the intensity of dynamical noise impacts the inference on stochastic data. Our results indicate that the overall quality of fit to the noisy data ([Figure 3A](#)) and the accuracy of recovered parameters ([Figure 3B](#)) remain stable up to $\sigma \leq 10^{-1}$ for all algorithms. However, both metrics exhibit a significant increase beyond this threshold, particularly for optimization methods.

The MAP estimation is robust with regard to the amount of dynamical noise, but it tends to overfit. This is because the MAP estimation consistently provides the least accurate inference on parameters, even though its fit to the time-series data is almost perfect. In contrast, Bayesian inference methods such as HMC and SBI are overall more accurate and significantly more robust compared to optimization methods. Both HMC and SBI exhibit significantly lower error than others, even at high noise levels (e.g., $\sigma = 1$).

Interestingly, SBI appears to be more resilient to high levels of noise compared to HMC. Given a sufficient number of simulations for training (e.g., 100k), SBI demonstrates the most accurate and robust fit to the data, consistently staying close to a perfect fit across various noise values. See [Figure S8](#) for a systematic investigation on the impact of the number of simulations on the performance of SBI. Overall, these results indicate that SBI is the most accurate and robust algorithm in our benchmark.

3.3. Inference on missing data in the state-space

Here, we explore the performance of inference methods in situations where data is available for only one of the variables in the state-space modeling. We consider the mean membrane potential v as the observed data, which is simulated with dynamic noise of intensity $\sigma = 0.1$, while the firing rate r remains latent ([Figure 4A](#)). The noise intensity is fixed for inference process.

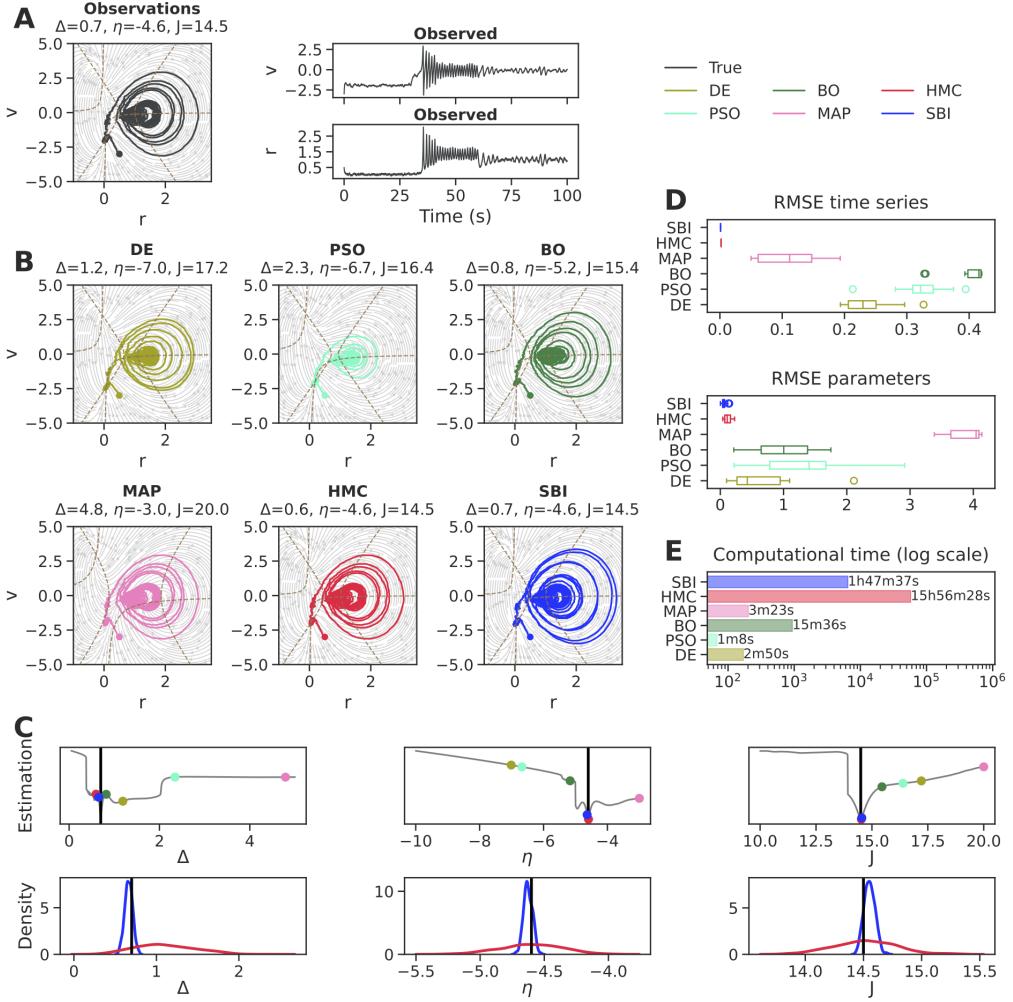


Figure 2. Inference on noisy data. **(A)** The generated bistable dynamics in the phase-plane and the corresponding time-series of firing rate (r) and membrane potential (v) used as the observations for inference (ground truth: $\Delta = 0.7$, $\eta = -4.6$, $J = 14.5$). **(B)** The estimated trajectories in the phase-planes, along with the corresponding parameters displayed in the top panels. All the algorithms qualitatively capture the bistable behavior. **(C)** The point estimation along with the profile likelihood (top panels) and the full posterior (bottom panels) for parameters Δ , η , and J . SBI leads to more precise parameter estimates with lower uncertainty compared to HMC. **(D)** Accuracy in estimation based on the sum over RMSE values for the time-series (top panel) and parameters (bottom panel). The bootstrap uncertainty is calculated through multiple runs. **(E)** Computational cost for each inference algorithm. When evaluating overall accuracy, uncertainty quantification, and computational cost, then SBI outperforms all other algorithms.

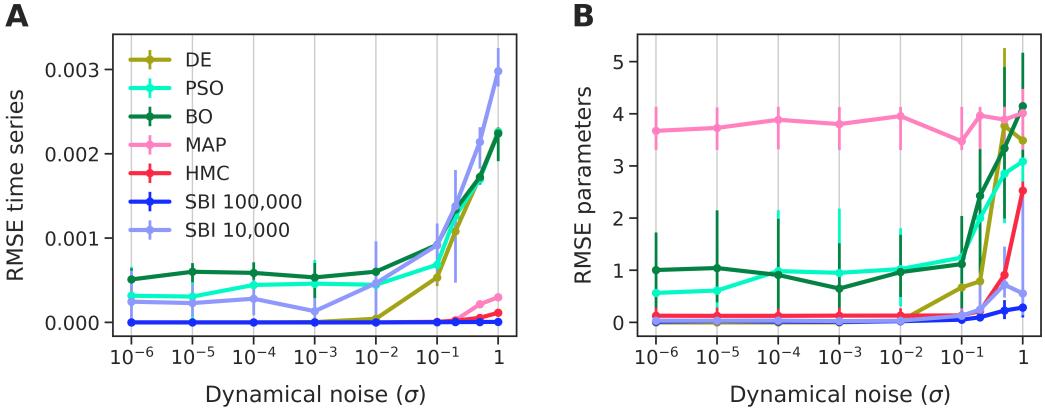


Figure 3. The performance of inference algorithms with increasing the intensity of dynamical noise using the sum over: (A) RMSE of the time-series, and (B) RMSE of the model parameters. Overall, with a sufficient number of simulations for training, the SBI emerges as the most accurate and robust method in our algorithmic benchmark.

From Figure 4B, we observe that only DE and SBI were capable of retrieving the true dynamics in the phase-plane. As shown in Figure 4C (top panel), among optimization algorithms, only DE correctly estimates the true parameters. Notably, HMC fails considerably in this problem by proposing over-confident posterior distributions that are far from the ground truth (Figure 4C, bottom panel). In contrast, the estimated posteriors using SBI are centered on the ground-truth parameters, but they exhibit a more diffuse uncertainty compared to the previous results (see Figure S9). Note that HMC also fails to accurately reconstruct the latent variable r from observed v (see Figure S10), even though its error on the fitted trajectories is comparable to that of SBI (Figure 4D, top panel). Nevertheless, the unreliability in the results produced by HMC becomes evident when observing the RMSE values for model parameters (Figure 4D, bottom panel).

In terms of computational cost for inference (Figure 1E), optimization by DE still has a clear advantage with its rapid performance. Interestingly, SBI remains efficient, approximately 68 orders of magnitude faster than HMC (see Table 3, Movie 5, and Movie 6)). Overall, SBI outperforms HMC in the recovery of the bistable dynamics, including the hidden firing rate. However, this approach can lead to an overestimation in the associated uncertainty, when compared to the full observed state-space dynamics (see Figure 2 versus Figure 4).

3.4. Phase-space reconstruction

As demonstrated in the previous section, the inference process posed a challenge when only the membrane potential v was observed and the firing rate r was missing. To improve the inference in such cases, we approximately reconstruct the hidden r from the observed counterpart v , using the time-delay embedding technique (Figure 5).

As it can be seen from Figure 5A, the reconstructed firing rates closely follow the original time-series (RMSE=0.151). This leads to accurately capturing the bistable switching behavior in the phase-plane, as shown in Figure 5B. We subsequently explored how the access to the trajectories of state variables influences the statistical relationship between parameters. When both the firing rate (r) and membrane potential (v) are observed, both HMC and SBI algorithms reveal a strong correlation between the average excitability η , and the synaptic weight J ($|\rho_{\eta,J}| = 0.78$), while the other parameters exhibit no codependency (see Figure 5C).

When the firing rate is latent and only the membrane potential is observed, the correlation between parameters is more pronounced, as estimated by HMC, while SBI tends to overestimate this co-dependency as fully degenerate (see Figure 5D). This highlights the

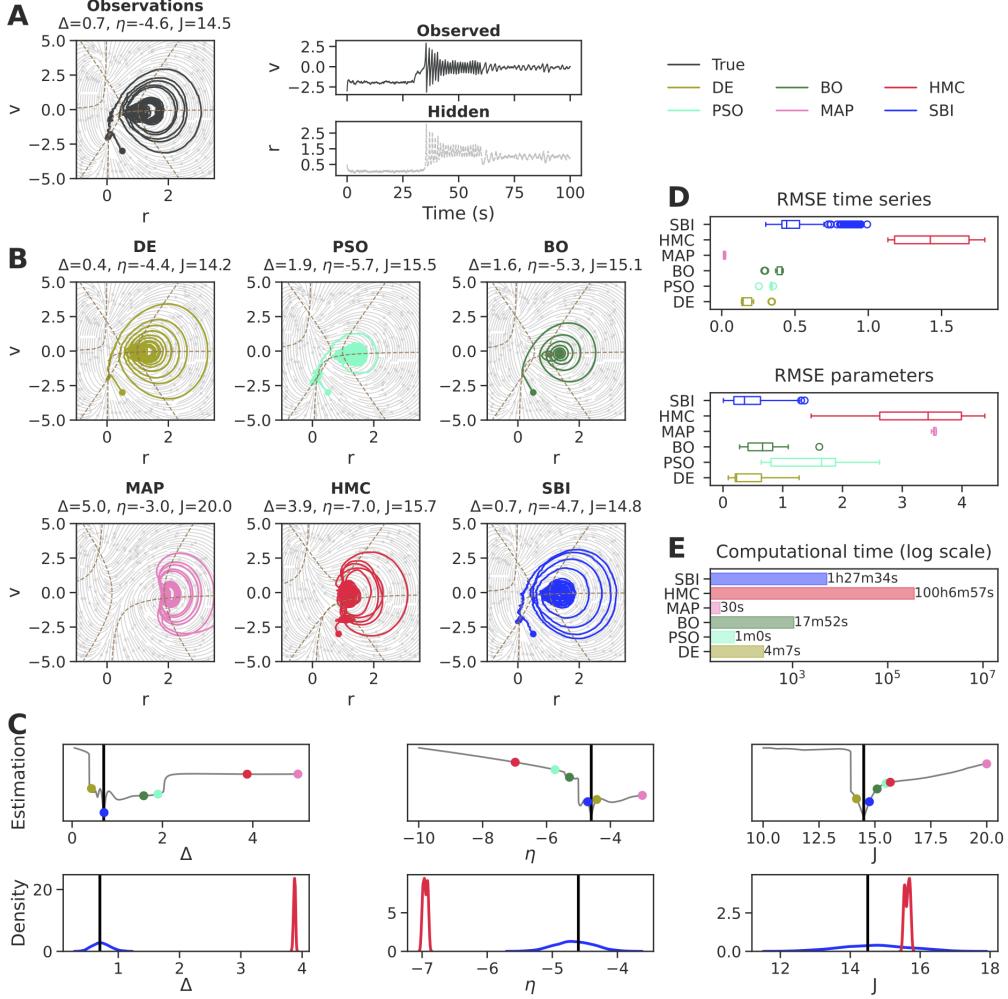


Figure 4. Inference on noisy observation with missing data. The generated bistable dynamics in the phase-plane and the corresponding time-series of observed membrane potential (v) and hidden firing rate (r), with ground truth: $\Delta = 0.7, \eta = -4.6, J = 14.5$. (B) The estimated trajectories in the phase-planes, along with the corresponding parameters displayed in the top panels. Only DE and SBI provide a close agreement with the observed bistable trajectories. (C) The point estimation along with the profile likelihood (top panels), and the full posterior (bottom panels) for parameters Δ , η , and J . (D) The accuracy of estimation is evaluated by summing the RMSE values for both the true time-series (top panel) and parameters (bottom panel), while considering the bootstrap uncertainty through multiple iterations. (E) The computational cost for each inference algorithm. Overall, DE and SBI outperform other algorithms in inferring the bistable dynamics when dealing with missing data.

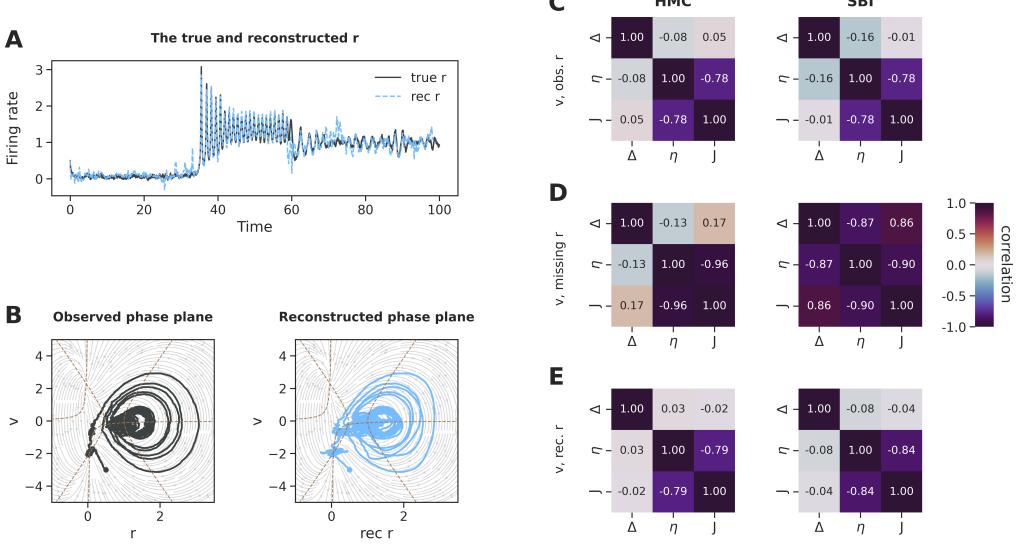


Figure 5. Reconstruction of firing rate r from mean membrane potential v , and comparison of inter-dependency between parameters. **(A)** Original and reconstructed mean firing rate. **(B)** Observed (left) and reconstructed (right) system dynamics in phase-plane. Correlations between estimated joint posterior distributions, using HMC (left) and SBI (right), when: **(C)** Both r and v are observed, **(D)** Only v is observed and r is hidden, **(E)** Reconstructed r from the observed v using time-delay embedding.

challenges in the inference or parameter estimation process when lacking access to complete knowledge of the system dynamics. This issue can be improved by using time-delay embedding to reconstruct the full phase-space dynamics and inform the inference process, as shown by the reduced correlation between parameters in Figure 5E.

3.5. Inter-dependency between generative parameters

In the previous sections, we performed inference against the system dynamics derived from the MF model given by Eq. (1). We now aim to investigate how the inference of the posterior distribution and the relationships between parameters remains consistent by traversing across scales. This can be achieved by conducting inference using observed data generated by QIF neurons (at the microscopic level) versus the data generated by MF model (at the macroscopic level).

First, we compared the simulated membrane potential v and firing rate r using a MF model given by Eq. (1) to the averaged activities of a network of 10^4 all-to-all connected QIF neurons as shown in Figure 6(A). Figure 6(B) demonstrates that the MF model can accurately generate the (smoothed) transient dynamics that emerge from an ensemble of spiking neurons. However, the first spike emitted by the MF model after stimulation exhibits a short delay compared to the averaged QIF neurons. Then, we used the data generated by MF and QIF models as the observations, and conducted inference using the MF model through Bayesian estimation algorithms, such as HMC and SBI. Specifically, we compared the correlations between the estimated joint posteriors of model parameters in each case, as shown in Figure 6(C).

We observed consistent correlations between parameters across the two types of datasets (macroscopic MF and microscopic QIF) when fitted using the MF model. Our results indicate that the strong negative linear correlation between excitability and synaptic weight ($\rho_{\eta,J} \approx -0.78$) persists when using HMC and SBI against both datasets. Considering the agreement between HMC and SBI across two datasets, we can conclude that the consistent high correlation between parameters η and J is intrinsic and not induced by the inference process or model assumptions.

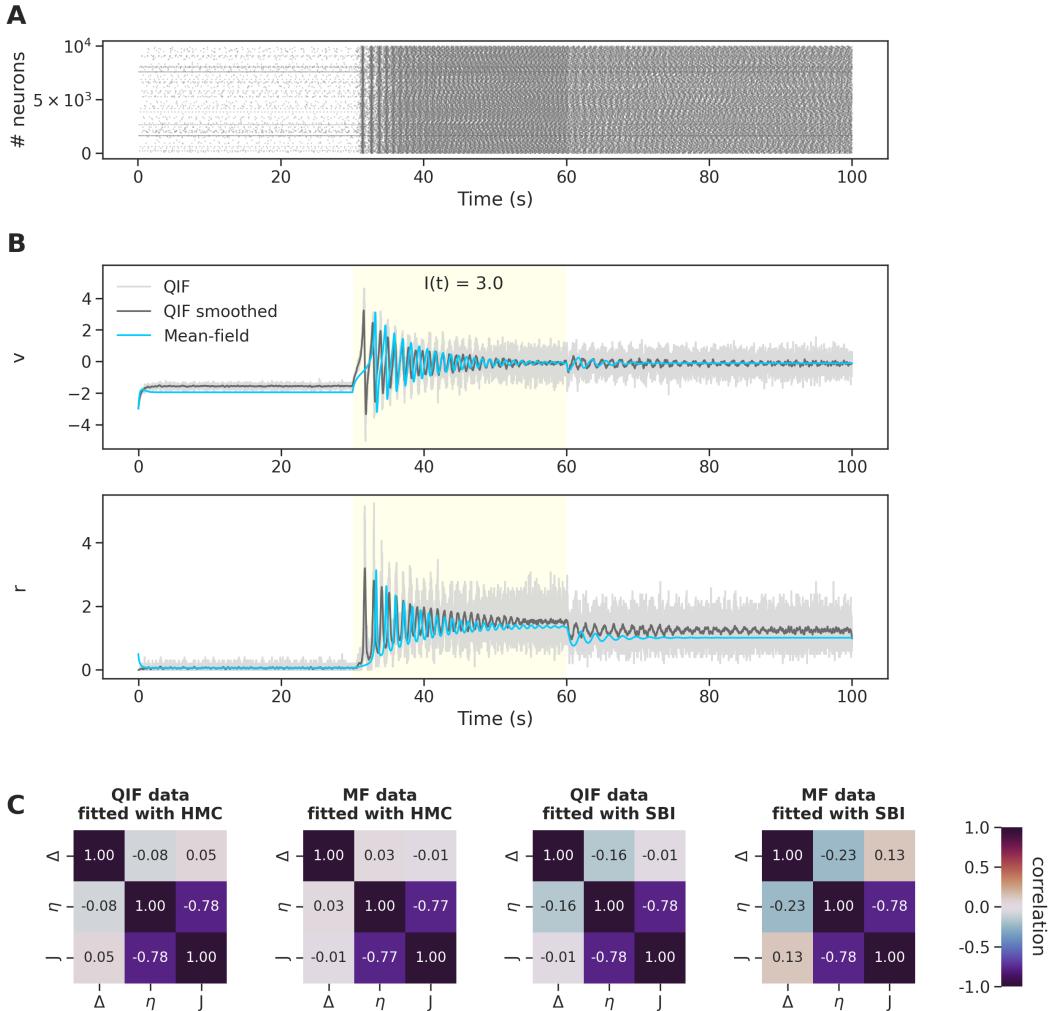


Figure 6. Comparing the transient dynamics of a MF model with the emergent dynamics of a network of QIF neurons, and exploring the interdependency between parameters. **(A)** Raster plot of QIF neurons. **(B)** The membrane potential v (top panel) and firing rate r (bottom panel) generated by MF model (in cyan) versus averaged activities of QIF neurons, as the raw simulations (in light gray) and smoothed (dark gray). At time $t = 30$ sec, a current $I_0 = 3$ mv is applied to all neurons, and set to zero again at $t = 60$ sec. **(C)** The Pearson correlation coefficients between parameters in the MF model, estimated using Bayesian algorithms (HMC and SBI), against data generated by different models (MF and QIF).

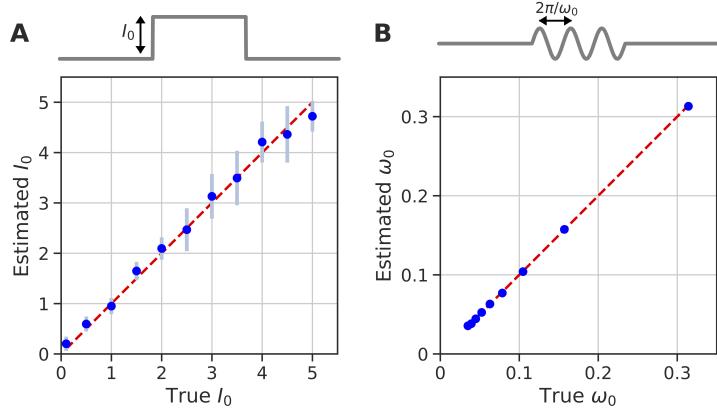


Figure 7. SBI on the stimulus current. **(A)** Step current as $I(t) = I_0 \cdot \mathbf{1}_{\text{stim}}(t)$. **(B)** Sinusoidal current as $I(t) = \sin(\omega_0 t) \cdot \mathbf{1}_{\text{stim}}(t)$. Dashed red line represents a perfect fit.

3.6. SBI on the stimulus current

Here, we challenge SBI approach in inferring system dynamics while also accounting for an unknown input current, which plays a crucial role in emerging the bistable behavior. Given only the position or waveform of input currents, we estimate the intensity or angular velocity across various ground-truth values: I_0 in a step current as $I(t) = I_0 \cdot \mathbf{1}_{\text{stim}}(t)$ and ω_0 in a sinusoidal current as $I(t) = \sin(\omega_0 t) \cdot \mathbf{1}_{\text{stim}}(t)$, as shown in Figure 7(A), and (B), respectively.

Our results demonstrate that in both cases, the SBI approach accurately recovers the unknown parameters in the input currents. The posterior credibility intervals, visualized as error bars, indicate certain estimates that are close to a perfect fit ($y = x$, in red) across all different values. See Figure S11 for the observed and predicted time-series. This validates the capability of SBI in accurately estimating the system dynamics, even when the characteristics of the input current are unknown.

3.7. SBI on stability of system dynamics

Here, we show that SBI can be used to investigate the stability of system dynamics from low-dimensional summary statics of observed time-series. Figure 8 shows a phase diagram of the system as a function of the mean η and synaptic weight J , both normalized by the width of the input distribution Δ . There are three qualitatively distinct regions of the phase diagram: (i) A single stable node corresponding to a low-activity state (shown in blue), (ii) A single stable focus generally corresponding to a high-activity state (shown in red), and (iii) A region of bistability between low and high firing rate (shown in cyan).

Interestingly, a similar basin of bi-stability in phase diagram can be readily reproduced using deep neural density estimators (such as MAF model) in SBI approach. By training on the low-dimensional data features extracted from the time-series (presence or absence of oscillations before, during, and after stimulation), the generated posterior samples display a very close agreement with the results obtained from linear stability analysis. This demonstrate the capability of SBI in accurately estimating system dynamics from summary statics, including the presence of bi-stability in the phase diagram.

3.8. Neural ODEs on system dynamics

In this section, our aim is to infer the collective dynamics of QIF neurons without making any assumptions about the underlying generating dynamics. To achieve this, we used Neural ODEs as a powerful tool for modeling continuous-time dynamics without assuming any prior knowledge of the underlying equations governing the system. Unlike traditional discrete-time models, Neural ODEs parameterize the continuous-depth formulation, allowing for seamless interpolation between observed data points.

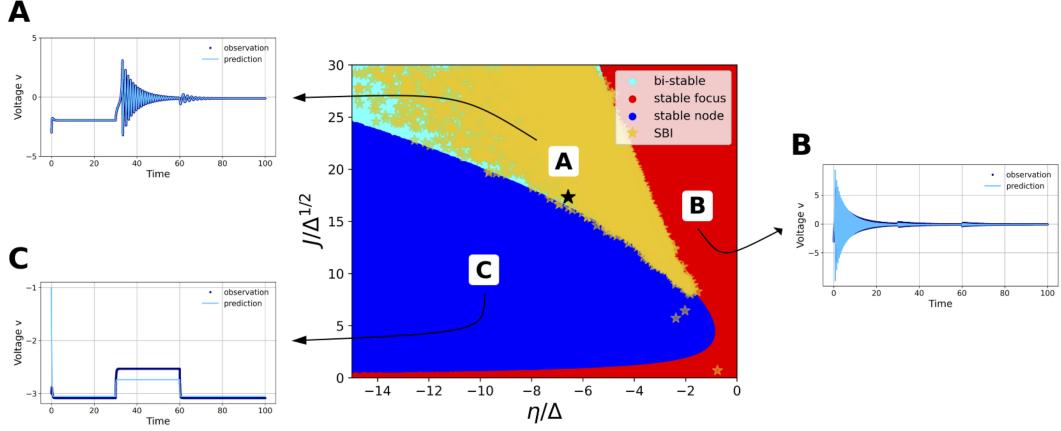


Figure 8. SBI over the stability of system dynamics in phase diagram. In the wedge-shaped region (shaded in cyan), bistability exists between a high and a low activity state. On the right side, a stable focus is indicated (shaded in red), while on the left side, a stable node is depicted (shown in blue). By training a deep neural density estimator on data features such as damped oscillations due to the input, the posterior samples generated using SBI accurately capture the bistable dynamics (shown in yellow), aligning closely with results from linear stability analysis. The black asterisk denotes the observation point used for inference. The insets display observed (dark blue) and predicted (light blue) time-series in different regimes: (A) bistability, (B) stable focus, and (C) stable node.

We first validate Neural ODEs using the data garnered by MF model described by Eq. (1). We generated 3 different datasets using the MF model given by Eq. (1), with a set of parameters corresponding to a bistable regime: $\{\Delta = 1, J = 15, \eta = -5\}$. For each datasets, we sampled the phase-space by varying initial conditions according to a regular grid so that $r_0 \in [0.1, 3]$ and $v_0 \in [-2, 2]$. We then solved the system for 1000 time points (with an integration time step of $dt = 0.01$). To speed up training, each trajectory was downsampled by a factor of 10 and divided into 10 segments, each consisting of 10 time points. The training and test data were randomly split, with 75% of the data used for training and 25% used for testing.

The first training set, comprising of $N = 100$ deterministic trajectories, was generated (see Figure 9A). The results indicate that the Neural ODE almost perfectly reconstructed the phase-space (see Figure 9B). In the second training set, which had the same size, a dynamical noise with a standard deviation of $\sigma = 0.1$ was added during the integration process (Figure 9C). In this case, the estimated nullclines suffered from overfitting although without affecting the overall reconstructed dynamics, as the predicted trajectories were still very similar to the original data (as shown Figure 9D). Increasing the size of the training dataset (see Figure 9E) significantly reduces overfitting. As a result, the reconstructed nullclines show a very close agreement with those obtained from the deterministic data (Figure 9F). To illustrate overfitting, Figure S12 shows the loss functions for the different scenarios, as well as snapshots of phase-space reconstruction during training. Overall, the Neural ODE successfully reconstructs the underlying deterministic system, even in the presence of noise (see Movie 7).

We then trained Neural ODEs using data generated by 10^4 QIF neurons with a uniform stimulus (see Figure 10). The data was partitioned using the first 400 points for training and predicting the remaining 1600 points. The results indicate that using derivatives dynamics, we can achieve a reliable understanding and prediction of the complex behavior of a network of spiking neurons, as illustrated in Figure 10. The emergent dynamics vary based on different parameter settings, resulting the stable node, stable focus, and bistable regime. See Figure S13 for the loss function in the training and test sets.

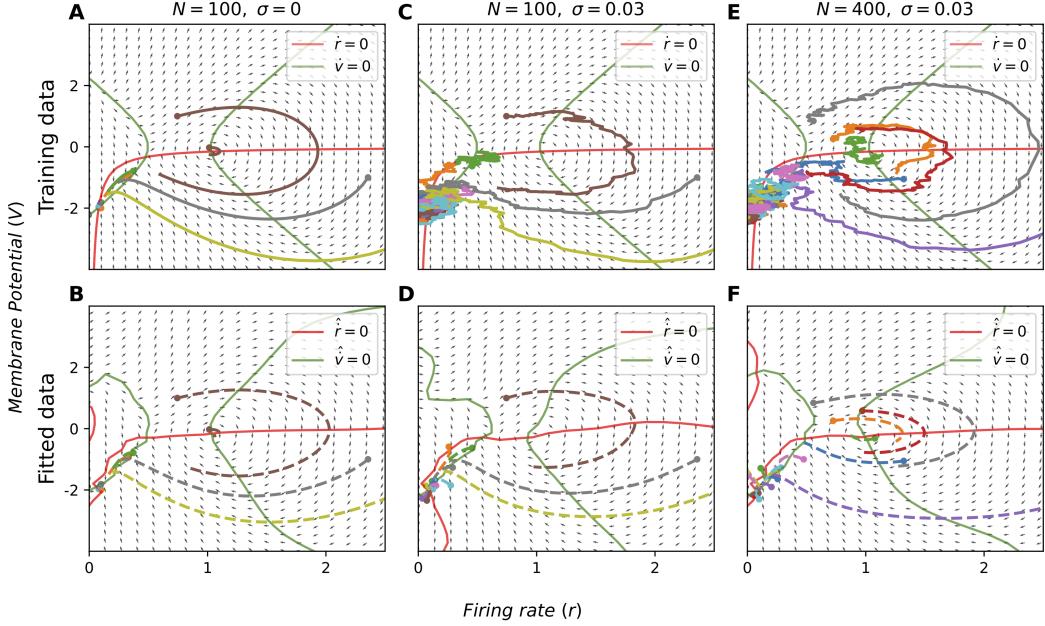


Figure 9. Example of training data generated from the MF model set in a bistable regime (top row). The trajectories are displayed after the 10-fold split, without the downsampling step for better visualization. Corresponding fit obtained with the Neural ODE for the same initial conditions and estimated phase-plane (bottom row) after 45000 training iterations. Either (**A, B**) using a training set of 100 deterministic trajectories, (**C, D**) using a training set with dynamical noise, (**E, F**) or a larger dataset. The red and green curves represent the nullclines of r and v , respectively. The dots represent the initial values for each trace, while the dashed lines correspond to the Neural ODE trajectories. The neural ODE is prone to overfitting when noise is introduced in the training data, although it still preserves the overall dynamics. A larger dataset helps in recovering smoother nullclines.

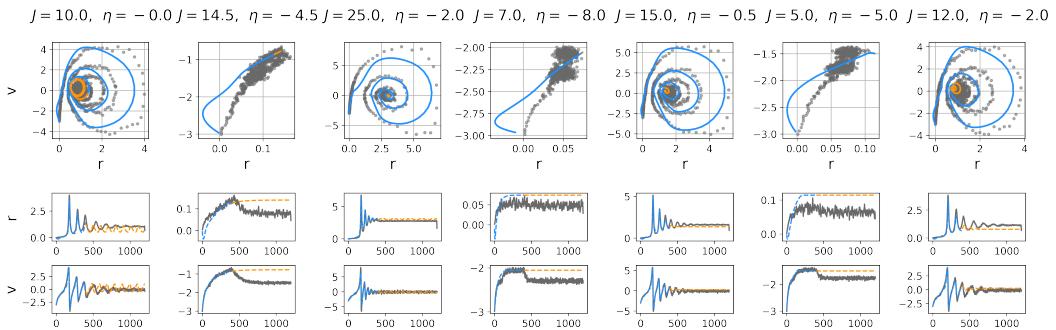


Figure 10. Reconstruction and extrapolation on QIF data with various parameters and dynamic regimes, using Neural ODEs. (**A**) Stable nodes, (**B, C**) stable foci, (**D, E**) bistability. The grey dots represent sparse (down-sampled) observations, blue lines represent predictions (400 points), and orange lines represent extrapolations (1600 points). We can see that our Neural ODE has learned the system dynamics and provides accurate predictions based on data generated at the microscopic level.

4. Discussion

In the ever-evolving field of computational neuroscience, accurately estimating parameters that consistently govern the collective behavior of neural networks is a crucial endeavor, especially within the framework of recurrently coupled spiking neurons. In this study, we emphasize the use of mean-field (MF) theory to streamline the inference process for networks of spiking neurons. This choice is driven by the computational challenges in the calculation of likelihood function that arise when attempting inference using these networks in forward modeling, which the posterior estimation becomes computationally prohibitive.

The needs and objectives of researchers in the field of computational neuroscience are as diverse as the neural systems they aim to understand. Some researchers may prioritize rapid point estimation by optimization, seeking quick outcomes to inform real-time decisions. On the other hand, some researchers find value in exploring the full distribution of parameters, which provides a nuanced understanding of parameter uncertainty for reliable decision making. This paper offers a comprehensive, yet non-exhaustive, benchmarking of the state-of-the-art inference methods applied to a MF model of spiking neurons (see [Figure 1](#), [Figure 2](#), and [Figure 4](#)). This provides researchers with the automatic tools to evaluate the reliability, efficiency, and applicability of these methods in various inverse problems. Our comparative analyses offer practical guidance, assisting researchers in selecting the most suitable method for their specific datasets and research inquiries.

While the optimization method can construct a confidence interval for the estimation based on a threshold for accepting or rejecting the estimates, the results are highly dependent on the chosen threshold value ([Beaumont et al., 2002](#); [Cranmer et al., 2020](#)). In contrast, Bayesian inference naturally provides uncertainty quantification by placing a distribution over parameters and treating them as random variable. Following Bayes' rule, this distribution is updated with evidence from observed data to form the posterior distribution, which furnishes comprehensive information for inference and prediction.

Our results indicated that evolutionary algorithm for solving global optimization problems, such as DE, provide rapid and accurate point estimation of the true generative parameters when there is no dynamical noise present. Challenges arise when using optimization methods in the presence of noisy data, mainly due to the lack of an efficient form of the objective function. The selection of the objective function plays a crucial role in determining the estimation through optimization methods. The error explanation with distance metrics such as RMSE is limited in accurately capturing the underlying data generation process. This is because when generative parameters remain unchanged but when dynamic noise is introduced, the time series can show large fluctuations, resulting in deviations from the observed data. In particular, the presence of noise can easily lead to unreliable estimations, increasing the risk of overfitting, where the model fits to the noise rather than capturing the true underlying relationship. Consequently, it becomes more conspicuous to conduct inference using distributions in the Bayesian framework, particularly when dealing with dynamical noise. Diagnosing of overfitting, as shown using MAP estimation (see [Figure 2](#) and [Figure 4](#)), can be better understood through uncertainty quantification.

Furthermore, Bayesian inference reveals the relationships between parameters, capturing degeneracy in the data ([Edelman and Gally, 2001](#); [Hashemi et al., 2023](#)). For example, when we assess the agreement between HMC and SBI on different datasets (as shown in [Figure 5](#), [Figure S4](#), [Figure S5](#), and [Figure S9](#)), it can be concluded that the persistent strong correlation between parameters η and J is inherent and not influenced by the inference procedure or model assumptions. This is in line with previous findings that have reported a strong and robust correlation between firing rates and synaptic weights across different brain states, environments and situations ([Buzsáki and Mizuseki, 2014](#)).

Using high-performance computing, model simulations can be run independently, creating a large training dataset for training deep neural density estimators in SBI approach ([Hashemi et al., 2023](#)). In contrast, HMC is limited to embarrassingly parallel execution with only independent chains on computational nodes ([Hashemi et al., 2021](#)). Moreover, when dealing with bistability in the state-space representation, HMC methods require sig-

nificant computational time to detect state transitions in the latent space (see [Table 1](#), [Table 2](#), and [Table 3](#)) or need to be augmented with generative models such as normalizing flows ([Hoffman et al., 2019](#); [Gabrié et al., 2022](#)). On the other hand, SBI offers efficient Bayesian estimation, even without detailed knowledge of the system’s state-space representation. By addressing questions such as computational cost, uncertainty quantification, inter-dependency exploration, and data availability, we conclude that SBI is more efficient than alternatives in making informed choices from microscopic states to emergent dynamics at the macro scale.

One of the main findings of this study is the effectiveness of deep neural networks in generating probability distributions for parameters of networks of spiking neurons. This approach outperforms other computational algorithms, such as MCMC, particularly in real-world applications involving missing data in bistable systems (see [Figure 4](#)). The effectiveness of deep generative models for inference from the mechanistic model of networks of spiking neurons is confirmed by the robustness of the estimation under significant dynamic noise ([Figure 3](#), and [Figure S7](#)), as well as the precise estimation of input current ([Figure 7](#)) and the consistency with linear stability analysis ([Figure 8](#)). However, it is important to acknowledge that the use of deep generative models can result in an overestimation of uncertainty and correlations between parameters. To address this challenge, incorporating time-delay embedding is an effective remedy ([Figure 5](#)). Furthermore, this study highlights the use of deep Neural ODEs in inferring vector fields at a macroscopic level, which enables the prediction of system dynamics from microscopic states. This emphasizes the potential applications of this approach to make interpretable predictions at the whole-brain scale from simulations at a detailed level, aiding in the prognosis and diagnosis of brain diseases. Symbolic regression applied to the outcomes from Neural ODEs has the potential to uncover closed-form equations, offering promising prospects for future work. This approach may lead to the discovery of concise data-driven mean-field representations of complex neural dynamics, contributing to our understanding of brain function.

In conclusion, this work highlights the improved accuracy and efficiency that deep learning techniques bring to the inference from networks of spiking neurons. It opens up exciting possibilities for future research in neural computation, where the trade-off between accuracy and uncertainty needs to be carefully considered. As we continue to explore the capabilities of Neural ODEs, this study serves as a valuable step forward in our quest to unravel the complexities of neural networks and their computational mechanisms.

Information Sharing Statement

All code is available at GitHub (https://github.com/ins-amu/Inference_MFM).

Acknowledgements

This work was funded by Human Brain Project SGA3, and the Fondation pour la Recherche Médicale (DIC20161236442). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Author contributions

Conceptualization: M.W., V.J., and M.H. Methodology: M.W., and M.H. Software: M.W., and M.H. Investigation: N.B., M.B., and M.H. Visualization: N.B., M.B., and M.H. Supervision: M.H., ad V.J. Funding acquisition: V.J. Writing - original draft: N.B., and M.H. Writing - review & editing: N.B, M.B, M.W., V.J, and M.H.

5. Supplementary

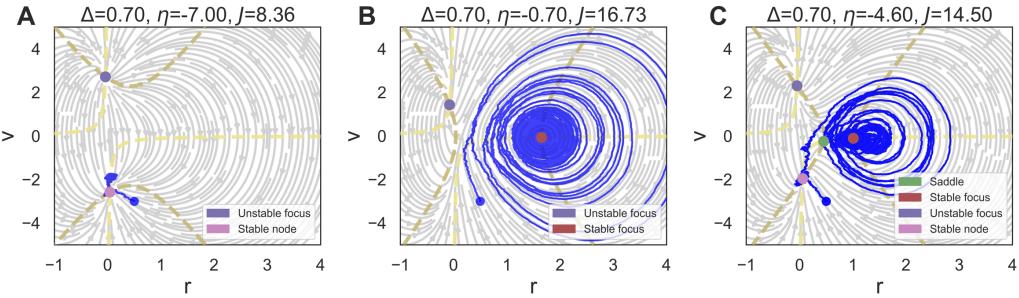


Figure S1. The phase-plane analysis of a mechanistic model of a network of all-to-all connected QIF neurons, using linear stability analysis. The stability of the system depends on the mean excitability (η) and synaptic weight (J), both normalized by the width of the input distribution (Δ). **(A)** A single stable node corresponding to a low-activity state. **(B)** A single stable focus (spiral) corresponding to a high-activity state. **(C)** A bistability between low and high firing rates (stable node and stable focus, respectively). The upper section between v - and r -nulclines (in dark and light yellow, respectively) corresponds to an unstable focus.

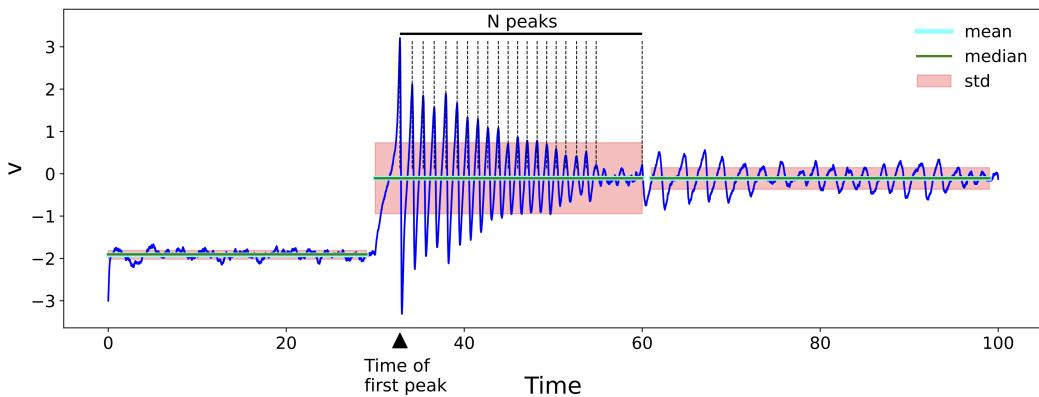


Figure S2. Data features extracted from mean membrane potential data. Additional data features that are not represented on this diagram include skewness and kurtosis.

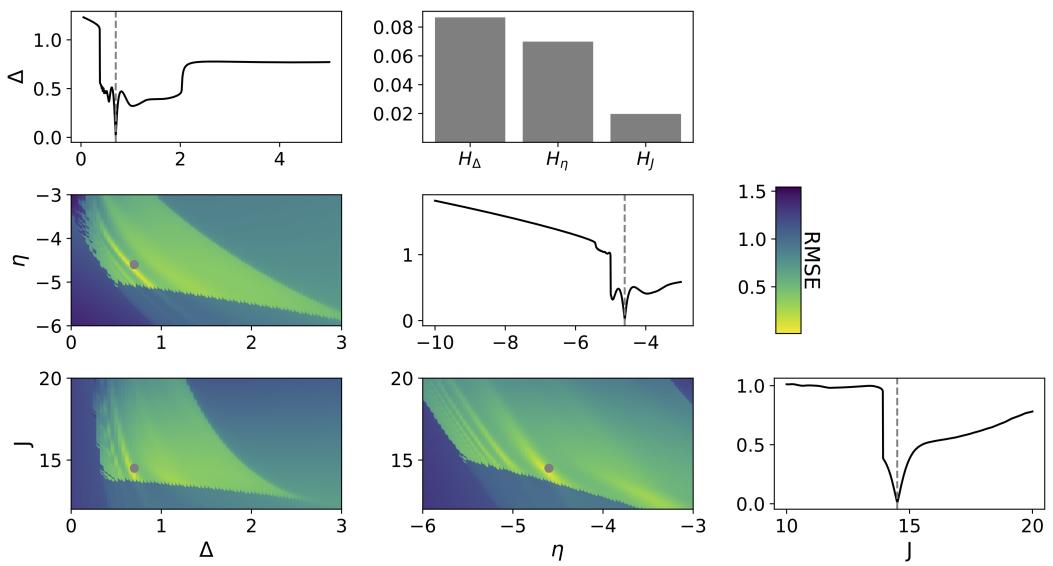


Figure S3. Sensitivity analysis on model parameters. The profile likelihood is calculated using the root-mean-squared error (RMSE) between the observed and generated data, considering either a single parameter (represented by black curves) or multiple parameters to vary (represented by colored surfaces), while keeping the other parameters fixed. The true parameters are shown in gray, represented by a dashed vertical line (for a single parameter) or 2D coordinates (for multiple parameters). The value of the Hessian matrix at the global minimum is displayed in the bar plot.

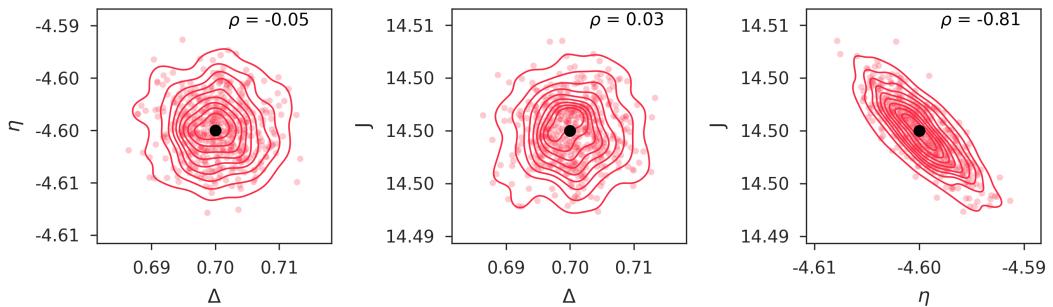
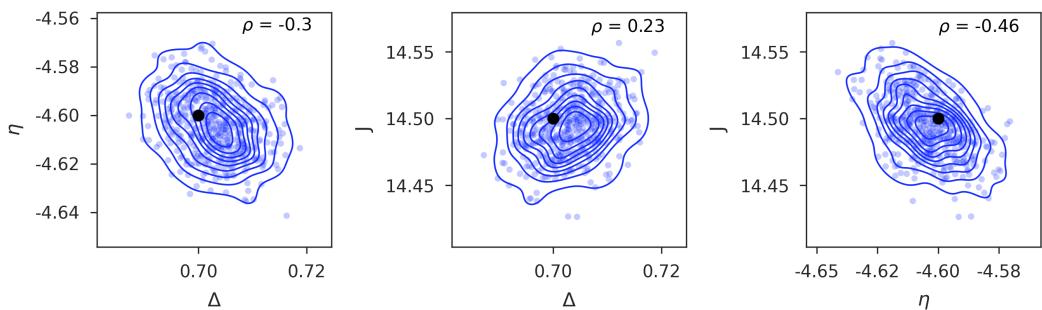
A - HMC**B - SBI**

Figure S4. Paired posterior samples of model parameters inferred from deterministic data, using (A) HMC and (B) SBI. Samples do not exhibit significant pair-wise correlation in parameter couples (Δ, η) and (Δ, J) . HMC manifests high linear correlation in sampling from joint parameters (η, J) . In terms of uncertainty quantification, HMC offers a more informative posterior distribution compared to SBI. In terms of computational cost, SBI is approximately 60 orders of magnitude faster than HMC.

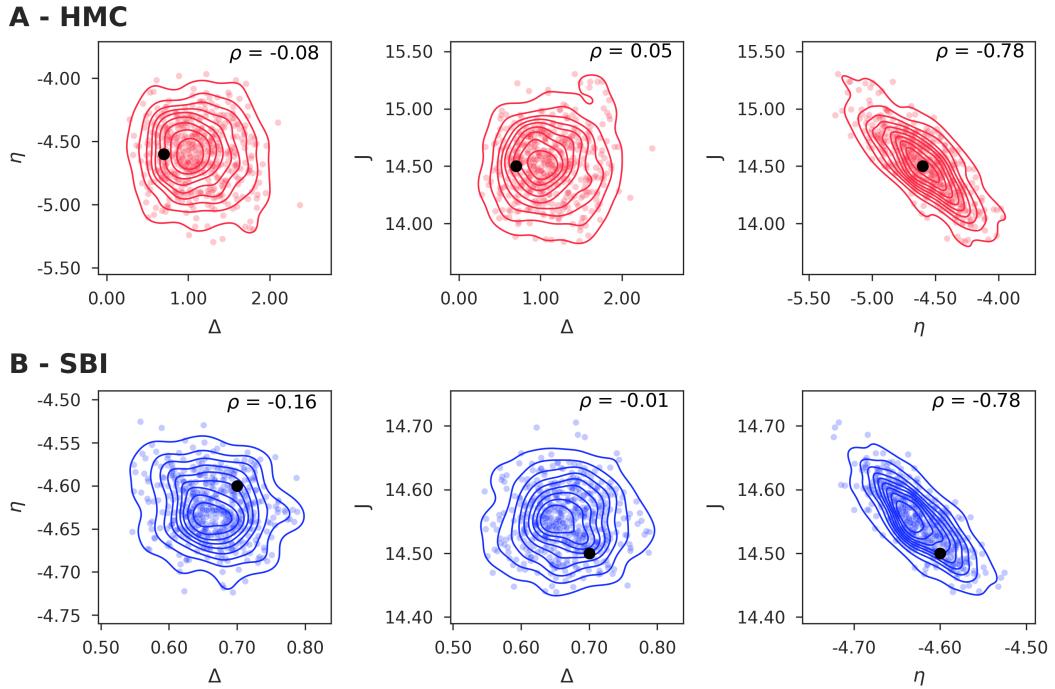


Figure S5. Paired posterior samples of model parameters inferred from noisy data, using (A) HMC and (B) SBI. Both algorithms provide uncorrelated samples for parameter couples (Δ, η) and (Δ, J) but exhibit the same level of strong linear correlation in sampling from joint parameters (η, J) . Compared to HMC, SBI generates posteriors that more tightly center around the ground-truth parameters. Moreover, SBI remains approximately 8 orders of magnitude faster than HMC.

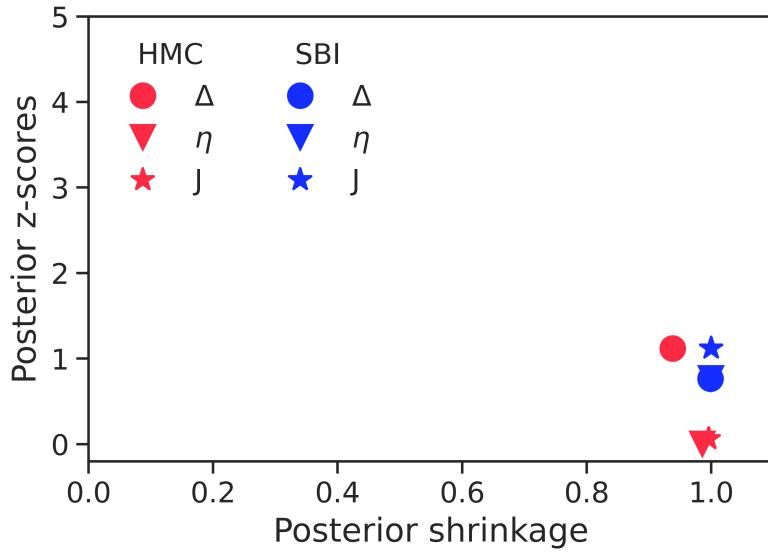


Figure S6. Posterior z-scores versus shrinkage, as a diagnostic for the reliability of Bayesian inference. An ideal Bayesian inference yields small z-scores (indicating less error) and high posterior shrinkage (indicating more contraction with respect to the prior distribution after learning from data). Therefore, their concentration lies in the right-bottom corner of the plot. When no data is missing, both HMC (red) and SBI (blue) perform near-optimally. However, SBI is significantly faster than HMC (at least 8 orders of magnitude). The posterior z-scores is defined as $z = \left| \frac{\bar{\theta} - \theta^*}{\sigma_{post}} \right|$, where $\bar{\theta}$ and θ^* are the posterior mean and the true values, respectively, whereas σ_{prior} , and σ_{post} indicate the standard deviations of the prior and the posterior, respectively. The posterior shrinkage defined as $s = 1 - \frac{\sigma_{post}^2}{\sigma_{prior}^2}$, where σ_{prior} , and σ_{post} indicate the standard deviations of the prior and the posterior, respectively.

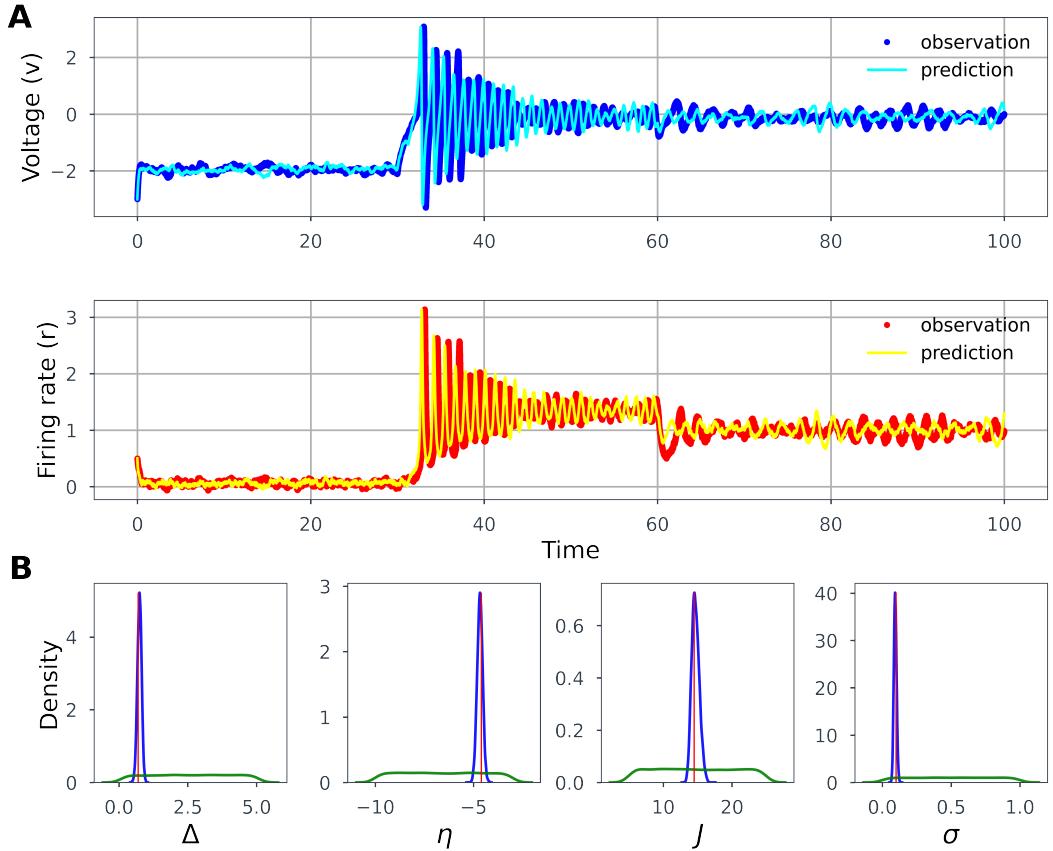


Figure S7. SBI from noisy data. (A) The time-series prediction of membrane potential (v) and firing rate (r). (B) Estimated posteriors of parameters Δ , η , J , and the intensity of dynamical noise σ . The true values are shown by vertical red lines. The priors and estimated posteriors using SBI are shown in green and blue colors, respectively. Given the low-dimensional data features of time-series, SBI is able to accurately and efficiently estimate the MF parameters including the noise in the system.

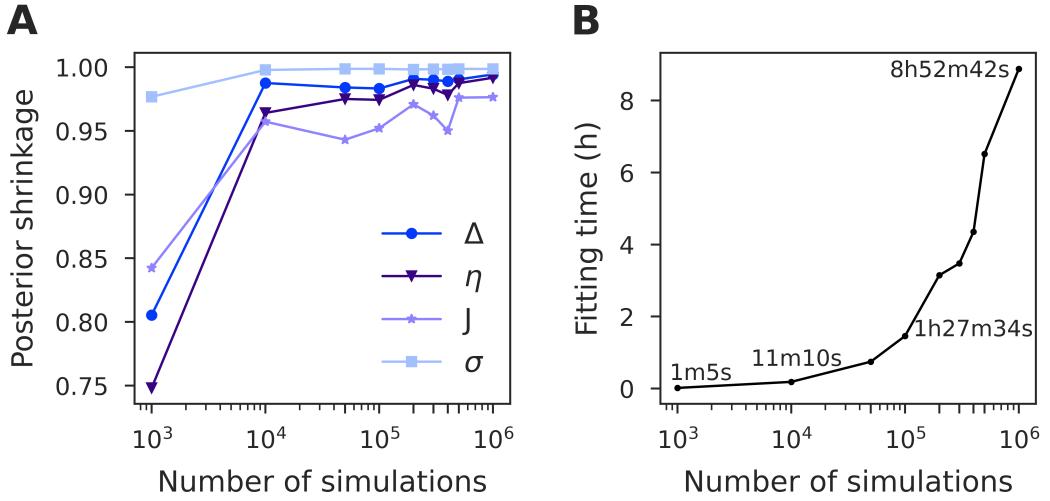


Figure S8. Performance of SBI as a function of the number of simulations for the training step. **(A)** The tendency of shrinkage towards one indicates that all the posteriors are well-identified. While the shrinkage of the posterior is significantly improved when increasing the number of simulations from 1k to 10k, further increasing it beyond 100k only results in a marginal improvement. **(B)** The computational cost for SBI, which includes the simulation, training, and sampling steps, increases exponentially with respect to the number of simulations.

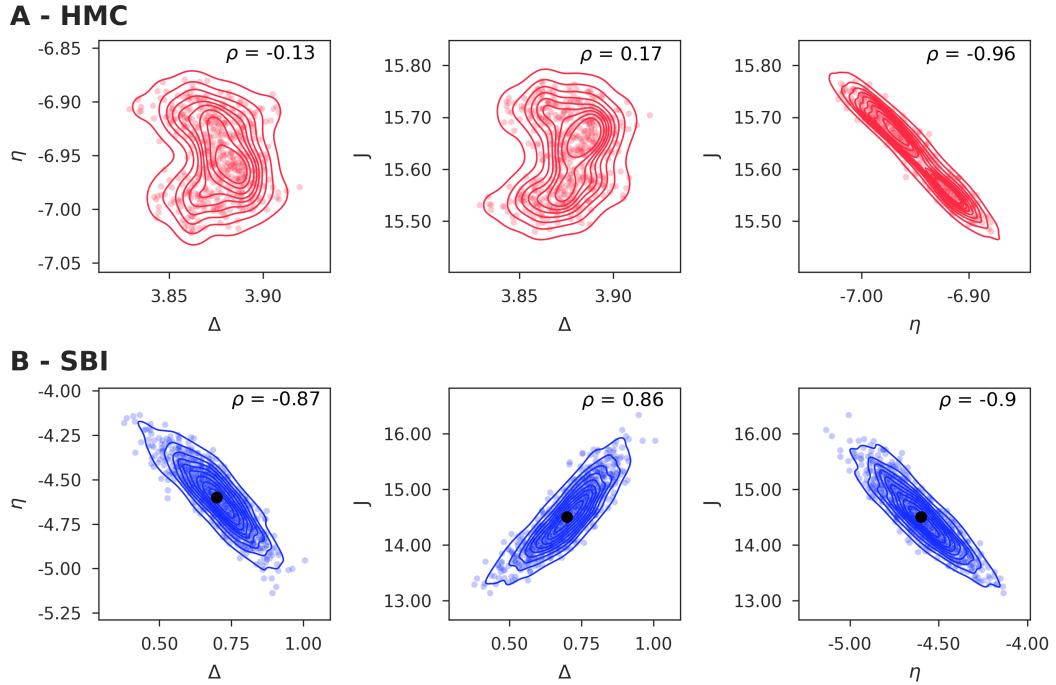


Figure S9. Paired posterior samples of model parameters inferred from noisy observation with missing data, using **(A)** HMC and **(B)** SBI, when only one variable (mean membrane potential v) is available. HMC samples deviate considerably from the true values, while SBI samples prove still reliable under such conditions, although accompanied with high correlation. Interestingly, SBI is approximately 68 orders of magnitude faster than HMC.

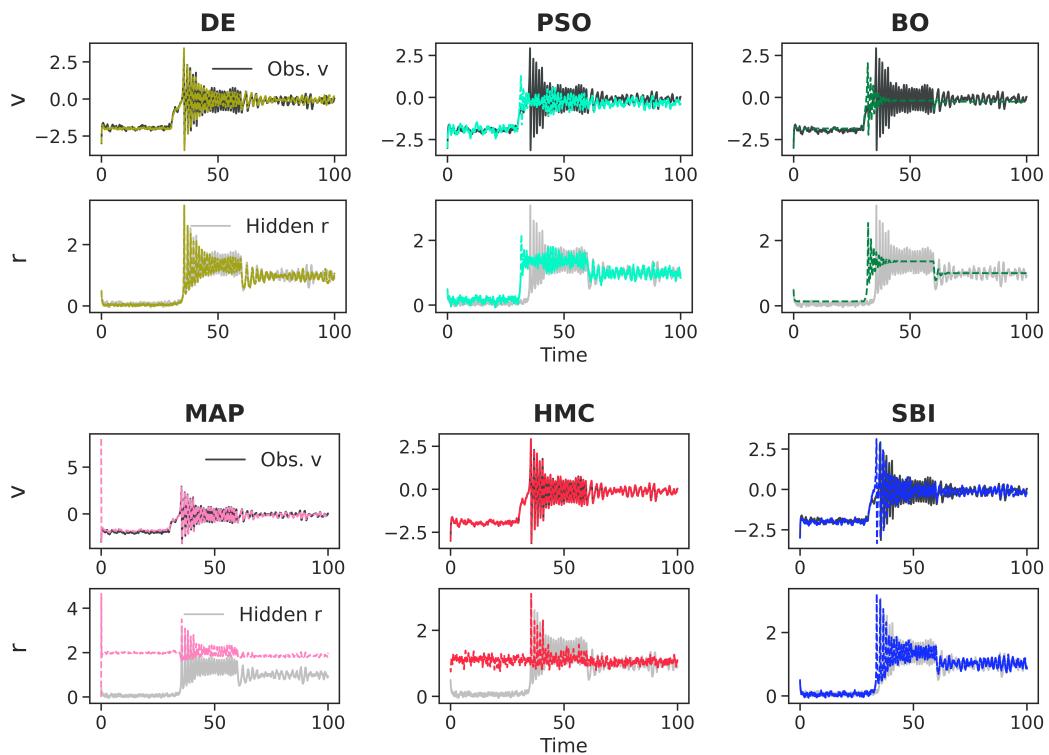


Figure S10. Fit to the noisy observation with missing data (i.e., when only one variable (mean membrane potential v) is available): observation v (top, black) and hidden r (grey, bottom). The time-series fit provided by the different inference algorithms is plotted in colors.

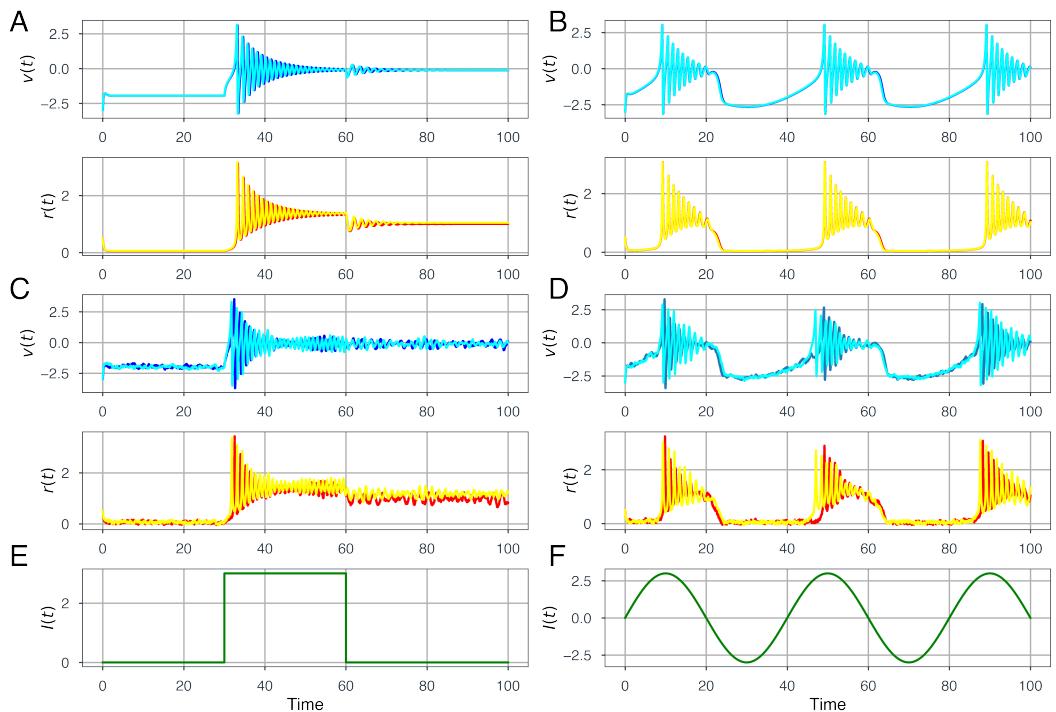


Figure S11. Exemplifying observed (v in blue and r in red) and predicted (v in cyan and r in yellow) time-series using SBI on (A, B) Deterministic data, (C, D) Stochastic data, given the input as (E, F) Step current with $I(t) = I_0 \cdot \mathbf{1}_{\text{stim}}(t)$, and sinusoidal current with $I(t) = \sin(\omega_0 t) \cdot \mathbf{1}_{\text{stim}}(t)$, respectively.

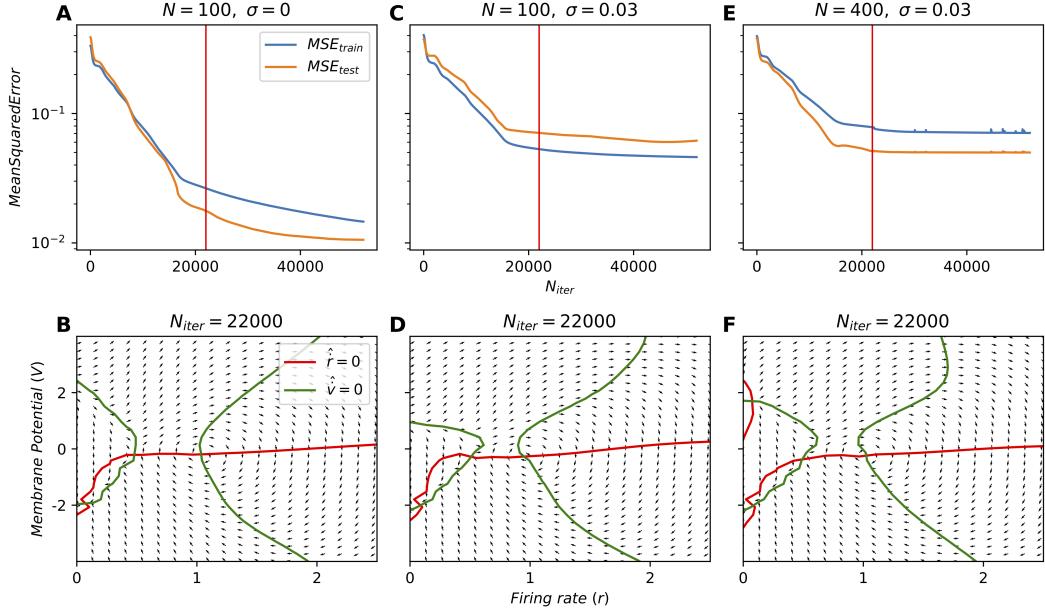


Figure S12. Loss functions for the training and test data using Neural ODEs on MF model. (A, B) For the deterministic training set of size 100, the phase-space estimation is performed at the 22000th iteration. (C, D) Training set of size 100 with dynamical noise. (E, F) Training set of size 400 with dynamical noise. The blue line represents the training error, while the orange line represents the test error. The vertical red line indicates the iteration at which a snapshot of the training is provided in the plot below.

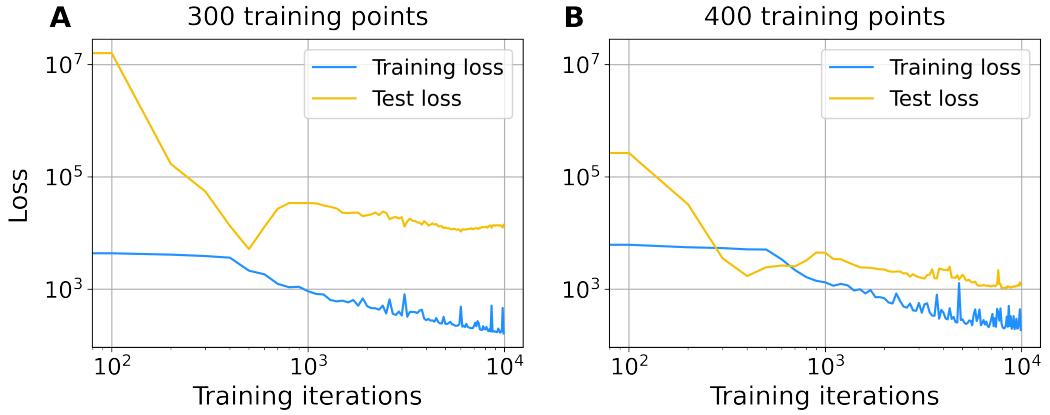


Figure S13. The loss function by training Neural ODEs on QIF data. The train and test quadratic loss function is calculated, when the train set is made of (A) 300 first points and (B) 400 first points. In each case, 10,000 iterations were used for training the model.

Method	RMSE parameters		RMSE time series		Running time
	mean	std	mean	std	
DE	5.256098e-16	1.662124e-15	6.757684e-19	1.114034e-18	0:0:51
PSO	7.181840e-01	8.508900e-01	3.221500e-04	2.887217e-04	0:0:35
BO	6.516364e-01	2.919346e-01	5.639526e-04	1.182884e-04	0:23:14
MAP	4.033716e+00	6.842649e-02	4.545021e-09	4.865735e-10	0:2:29
HMC	1.114323e-03	3.530420e-04	3.989852e-07	1.493006e-08	98:33:13
SBI	1.332644e-02	7.094355e-03	2.866951e-07	2.034953e-07	1:36:21

Table 1. Benchmark on deterministic synthetic data.

Method	RMSE parameters		RMSE time series		Running time
	mean	std	mean	std	
DE	0.672351	0.617302	0.000533	9.850079e-05	0:2:50
PSO	1.362540	0.852908	0.000717	1.116449e-04	0:1:8
BO	1.021078	0.519784	0.000888	8.232933e-05	0:15:36
MAP	3.883882	0.302616	0.000250	1.228436e-04	0:3:23
HMC	0.121416	0.081035	0.000004	1.427944e-08	15:56:28
SBI	0.052948	0.024310	0.000002	2.178518e-07	1:47:37

Table 2. Benchmark on stochastic synthetic data.

Method	RMSE parameters		RMSE time series		Running time
	mean	std	mean	std	
DE	0.426	0.388	0.218	0.102	0:4:7
PSO	1.495	0.720	0.406	0.054	0:1:0
BO	0.716	0.407	0.384	0.050	0:17:52
MAP	3.539	0.029	1.838	0.185	0:0:30
HMC	3.181	1.271	0.545	0.179	100:6:57
SBI	0.427	0.296	0.609	0.213	1:27:34

Table 3. Benchmark on missing data.

References

- Abarbanel, H.D., Carroll, T., Pecora, L., Sidorowich, J., Tsimring, L.S., 1994. Predicting physical variables in time-delay embedding. *Physical Review E* 49, 1840.
- Achard, P., De Schutter, E., 2006. Complex parameter landscape for a complex neuron model. *PLoS computational biology* 2, e94.
- Alonso, L.M., Marder, E., 2019. Visualization of currents in neural models with similar behavior and different conductance densities. *Elife* 8, e42722.
- Amari, S.i., 1977. Dynamics of pattern formation in lateral-inhibition type neural fields. *Biological cybernetics* 27, 77–87.
- Andrieu, C., De Freitas, N., Doucet, A., Jordan, M.I., 2003. An introduction to mcmc for machine learning. *Machine learning* 50, 5–43.
- Baldy, N., Simon, N., Jirsa, V., Hashemi, M., 2023. Hierarchical bayesian pharmacometrics analysis of baclofen for alcohol use disorder. *Machine Learning: Science and Technology* .
- Bandyopadhyay, A., Rabuffo, G., Calabrese, C., Gudibanda, K., Depannemaeker, D., Ivanov, A., Bernard, C., Jirsa, V.K., Petkoski, S., 2021. Mean-field approximation of network of biophysical neurons driven by conductance-based ion exchange. *bioRxiv* , 2021–10.
- Banga, J., Balsa-Canto, E., 2008. Parameter estimation and optimal experimental design. *Essays Biochem* 45, 195–210.
- Beaumont, M.A., Zhang, W., Balding, D.J., 2002. Approximate bayesian computation in population genetics. *Genetics* 162, 2025–2035.
- Betancourt, M., 2017. A conceptual introduction to hamiltonian monte carlo. *arXiv preprint arXiv:1701.02434* .
- Betancourt, M.J., 2013. Generalizing the no-u-turn sampler to riemannian manifolds. *arXiv preprint arXiv:1304.1920* .
- Biloš, M., Sommer, J., Rangapuram, S.S., Januschowski, T., Günnemann, S., 2021. Neural flows: Efficient alternative to neural odes. *Advances in neural information processing systems* 34, 21325–21337.
- Bishop, C.M., 2006. *Pattern Recognition and Machine Learning*. Springer.
- Bittner, S.R., Palmigiano, A., Piet, A.T., Duan, C.A., Brody, C.D., Miller, K.D., Cunningham, J., 2021. Interrogating theoretical models of neural computation with emergent property inference. *Elife* 10, e56265.
- Boelts, J., Lueckmann, J.M., Gao, R., Macke, J.H., 2022. Flexible and efficient simulation-based inference for models of decision-making. *Elife* 11, e77220.
- Bradbury, J., Frostig, R., Hawkins, P., Johnson, M.J., Leary, C., Maclaurin, D., Necula, G., Paszke, A., VanderPlas, J., Wanderman-Milne, S., Zhang, Q., 2018. JAX: composable transformations of Python+NumPy programs. URL: <http://github.com/google/jax>.
- Brehmer, J., 2021. Simulation-based inference in particle physics. *Nature Reviews Physics* 3, 305–305.

- Brehmer, J., Louppe, G., Pavez, J., Cranmer, K., 2020. Mining gold from implicit models to improve likelihood-free inference. *Proceedings of the National Academy of Sciences* 117, 5242–5249.
- Brunton, S.L., Proctor, J.L., Kutz, J.N., 2016. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the national academy of sciences* 113, 3932–3937.
- Buzsáki, G., Mizuseki, K., 2014. The log-dynamic brain: how skewed distributions affect network operations. *Nature Reviews Neuroscience* 15, 264–278.
- Carpenter, B., Gelman, A., Hoffman, M., Lee, D., Goodrich, B., Betancourt, M., Brubaker, M., Guo, J., Li, P., Riddell, A., 2017. Stan : A probabilistic programming language. *Journal of Statistical Software* 76. doi:[10.18637/jss.v076.i01](https://doi.org/10.18637/jss.v076.i01).
- Chen, R.T., Rubanova, Y., Bettencourt, J., Duvenaud, D.K., 2018. Neural ordinary differential equations. *Advances in neural information processing systems* 31.
- Cook, B.J., Peterson, A.D., Woldman, W., Terry, J.R., 2022. Neural field models: A mathematical overview and unifying framework. *Mathematical Neuroscience and Applications* 2.
- Coombes, S., Byrne, A., 2018. Next generation neural mass models, in: *Nonlinear dynamics in computational neuroscience*. Springer, pp. 1–16.
- Cranmer, K., Brehmer, J., Louppe, G., 2020. The frontier of simulation-based inference. *Proceedings of the National Academy of Sciences* 117, 30055–30062. URL: <https://doi.org/10.1073%2Fpnas.1912789117>, doi:[10.1073/pnas.1912789117](https://doi.org/10.1073/pnas.1912789117).
- David, O., Friston, K.J., 2003. A neural mass model for meg/eeg:: coupling and neuronal dynamics. *NeuroImage* 20, 1743–1755.
- Dayan, P., Abbott, L.F., 2005. *Theoretical neuroscience: computational and mathematical modeling of neural systems*. MIT press.
- Deco, G., Jirsa, V.K., Robinson, P.A., Breakspear, M., Friston, K., 2008. The dynamic brain: from spiking neurons to neural masses and cortical fields. *PLoS computational biology* 4, e1000092.
- Deistler, M., Goncalves, P.J., Macke, J.H., 2022. Truncated proposals for scalable and hassle-free simulation-based inference. *Advances in Neural Information Processing Systems* 35, 23135–23149.
- Duane, S., Kennedy, A.D., Pendleton, B.J., Roweth, D., 1987. Hybrid monte carlo. *Physics letters B* 195, 216–222.
- Duncker, L., Bohner, G., Boussard, J., Sahani, M., 2019. Learning interpretable continuous-time models of latent stochastic dynamical systems, in: *International Conference on Machine Learning*, PMLR. pp. 1726–1734.
- Dupont, E., Doucet, A., Teh, Y.W., 2019. Augmented neural odes. *Advances in neural information processing systems* 32.
- Durkan, C., Murray, I., Papamakarios, G., 2020. On contrastive learning for likelihood-free inference, in: *International conference on machine learning*, PMLR. pp. 2771–2781.
- Eberhart, R.C., Kennedy, J., 1995. New optimizer using particle swarm theory. *Proceedings of the 6th International Symposium on Micro Machine and Human Science* , 39–43.

- Edelman, G.M., Gally, J.A., 2001. Degeneracy and complexity in biological systems. *Proceedings of the National Academy of Sciences* 98, 13763–13768.
- Floudas, C., Gounaris, C.E., 2009. A review of recent advances in global optimization. *J Glob Optim* , 3–38.
- Friston, K., FitzGerald, T., Rigoli, F., Schwartenbeck, P., Pezzulo, G., 2017. Active inference: a process theory. *Neural computation* 29, 1–49.
- Friston, K., Kiebel, S., 2009. Predictive coding under the free-energy principle. *Philosophical transactions of the Royal Society B: Biological sciences* 364, 1211–1221.
- Gabrié, M., Rotkoff, G.M., Vanden-Eijnden, E., 2022. Adaptive monte carlo augmented with normalizing flows. *Proceedings of the National Academy of Sciences* 119, e2109420119.
- Gelman, A., Robert, C., Chopin, N., Rousseau, J., 1995. Bayesian Data Analysis. CRC Press.
- Gelman, A., Vehtari, A., Simpson, D., Margossian, C.C., Carpenter, B., Yao, Y., Kennedy, L., Gabry, J., Bräuer, P., Modrák, M., 2020. Bayesian workflow URL: <https://arxiv.org/abs/2011.01808>, doi:[10.48550/ARXIV.2011.01808](https://doi.org/10.48550/ARXIV.2011.01808).
- Gerstner, W., Kistler, W.M., 2002. Spiking neuron models: Single neurons, populations, plasticity. Cambridge university press.
- Gerstner, W., Kistler, W.M., Naud, R., Paninski, L., 2014. Neuronal dynamics: From single neurons to networks and models of cognition. Cambridge University Press.
- 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., Macke, J.H., 2020. Training deep neural density estimators to identify mechanistic models of neural dynamics. *eLife* 9, e56261. URL: <https://doi.org/10.7554/eLife.56261>, doi:[10.7554/eLife.56261](https://doi.org/10.7554/eLife.56261).
- Goyal, P., Benner, P., 2023. Neural ordinary differential equations with irregular and noisy data. *Royal Society Open Science* 10, 221475.
- Greenberg, D., Nonnenmacher, M., Macke, J., 2019. Automatic posterior transformation for likelihood-free inference, in: International Conference on Machine Learning, PMLR. pp. 2404–2414.
- Hashemi, M., Hutt, A., Buhry, L., Sleigh, J., 2018. Optimal model parameter estimation from eeg power spectrum features observed during general anesthesia. *Neuroinformatics* 16, 231–251. doi:[10.1007/s12021-018-9369-x](https://doi.org/10.1007/s12021-018-9369-x).
- Hashemi, M., Vattikonda, A.N., Jha, J., Sip, V., Woodman, M.M., Bartolomei, F., Jirsa, V.K., 2023. Amortized bayesian inference on generative dynamical network models of epilepsy using deep neural density estimators. *Neural Networks* 163, 178–194.
- Hashemi, M., Vattikonda, A.N., Sip, V., Diaz-Pier, S., Peyser, A., Wang, H., Guye, M., Bartolomei, F., Woodman, M.M., Jirsa, V.K., 2021. On the influence of prior information evaluated by fully bayesian criteria in a personalized whole-brain model of epilepsy spread. *PLoS computational biology* 17, e1009129.
- Hertz, J.A., 2018. Introduction to the theory of neural computation. Crc Press.
- Hirsh, S.M., Ichinaga, S.M., Brunton, S.L., Nathan Kutz, J., Brunton, B.W., 2021. Structured time-delay models for dynamical systems with connections to frenet–serret frame. *Proceedings of the Royal Society A* 477, 20210097.

- Hoffman, M., Sountsov, P., Dillon, J.V., Langmore, I., Tran, D., Vasudevan, S., 2019. Neutra-lizing bad geometry in hamiltonian monte carlo using neural transport. arXiv preprint arXiv:1903.03704 .
- Hoffman, M.D., Gelman, A., 2014. The no-u-turn sampler: Adaptively setting path lengths in hamiltonian monte carlo. *Journal of Machine Learning Research* 15, 1593–1623. URL: <http://jmlr.org/papers/v15/hoffman14a.html>.
- Hopfield, J.J., 1982. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences* 79, 2554–2558.
- Hutt, A., Hashemi, M., beim Graben, P., 2015. How to render neural fields more realistic. *Validating Neuro-Computational Models of Neurological and Psychiatric Disorders* , 141–159.
- Izhikevich, E.M., 2003. Simple model of spiking neurons. *IEEE Transactions on neural networks* 14, 1569–1572.
- Izhikevich, E.M., 2007. *Dynamical systems in neuroscience*. MIT press.
- Jirsa, V.K., Haken, H., 1996. Field theory of electromagnetic brain activity. *Physical review letters* 77, 960.
- Juang, J.N., 1994. *Applied system identification*. Prentice-Hall, Inc.
- Kandel, E.R., Schwartz, J.H., Jessell, T.M., Siegelbaum, S., Hudspeth, A.J., Mack, S., et al., 2000. *Principles of neural science*. volume 4. McGraw-hill New York.
- Kelley, C.T., 1999. *Iterative Methods for Optimization*. North Carolina State University, Raleigh, North Carolina.
- Kennedy, J., Eberhart, R., 1995. Particle swarm optimization 4, 1942–1948 vol.4. doi:[10.1109/ICNN.1995.488968](https://doi.org/10.1109/ICNN.1995.488968).
- Kennel, M.B., Brown, R., Abarbanel, H.D., 1992. Determining embedding dimension for phase-space reconstruction using a geometrical construction. *Physical review A* 45, 3403.
- Kim, S., Ji, W., Deng, S., Ma, Y., Rackauckas, C., 2021. Stiff neural ordinary differential equations. *Chaos: An Interdisciplinary Journal of Nonlinear Science* 31.
- Koppe, G., Toutounji, H., Kirsch, P., Lis, S., Durstewitz, D., 2019. Identifying nonlinear dynamical systems via generative recurrent neural networks with applications to fmri. *PLoS computational biology* 15, e1007263.
- Liepe, J., Kirk, P., Filippi, S., Toni, T., Barnes, C.P., Stumpf, M.P., 2014. A framework for parameter estimation and model selection from experimental data in systems biology using approximate bayesian computation. *Nature protocols* 9, 439–456.
- Linderman, S., Johnson, M., Miller, A., Adams, R., Blei, D., Paninski, L., 2017. Bayesian learning and inference in recurrent switching linear dynamical systems, in: *Artificial Intelligence and Statistics*, PMLR. pp. 914–922.
- Ljung, L., 1998. System identification, in: *Signal analysis and prediction*. Springer, pp. 163–173.
- Lueckmann, J.M., Bassetto, G., Karaletsos, T., Macke, J.H., 2019. Likelihood-free inference with emulator networks, in: *Symposium on Advances in Approximate Bayesian Inference*, PMLR. pp. 32–53.

- Lueckmann, J.M., Boelts, J., Greenberg, D., Gonçalves, P., Macke, J., 2021. Benchmarking simulation-based inference, in: International Conference on Artificial Intelligence and Statistics, PMLR. pp. 343–351.
- Lueckmann, J.M., Gonçalves, P.J., Bassetto, G., Öcal, K., Nonnenmacher, M., Macke, J.H., 2017. Flexible statistical inference for mechanistic models of neural dynamics. Advances in neural information processing systems 30.
- Marder, E., 1998. From biophysics to models of network function. Annual review of neuroscience 21, 25–45.
- McElreath, R., 2020. Statistical rethinking: A Bayesian course with examples in R and Stan. Chapman and Hall/CRC.
- Mendes, P., Kell, D., 1998. Non-linear optimization of biochemical pathways: applications to metabolic engineering and parameter estimation. Bioinformatics (Oxford, England) 14, 869–883.
- Miranda, L.J., 2018. Pyswarms: a research toolkit for particle swarm optimization in python. Journal of Open Source Software 3, 433.
- Młynarski, W., Hledík, M., Sokolowski, T.R., Tkačik, G., 2021. Statistical analysis and optimality of neural systems. Neuron 109, 1227–1241.
- Montbrió, E., Pazó, D., Roxin, A., 2015. Macroscopic description for networks of spiking neurons. Phys. Rev. X 5, 021028. URL: <https://link.aps.org/doi/10.1103/PhysRevX.5.021028>, doi:[10.1103/PhysRevX.5.021028](https://doi.org/10.1103/PhysRevX.5.021028).
- Murphy, K.P., 2022. Probabilistic machine learning: an introduction. MIT press.
- Neal, R.M., 2010. MCMC using Hamiltonian dynamics. Handbook of Markov Chain Monte Carlo 54, 113–162.
- Nocedal, J., Wright, S.J., 1999. Numerical optimization. Springer.
- Nogueira, F., et al., 2014. Bayesian optimization: Open source constrained global optimization tool for python .
- O’Leary, T., Sutton, A.C., Marder, E., 2015. Computational models in the age of large datasets. Current opinion in neurobiology 32, 87–94.
- Papamakarios, G., Murray, I., 2016. Fast ε -free inference of simulation models with bayesian conditional density estimation, in: Advances in Neural Information Processing Systems, pp. 1028–1036.
- Papamakarios, G., Nalisnick, E., Rezende, D.J., Mohamed, S., Lakshminarayanan, B., 2019a. Normalizing flows for probabilistic modeling and inference. arXiv preprint arXiv:1912.02762 .
- Papamakarios, G., Pavlakou, T., Murray, I., 2017. Masked autoregressive flow for density estimation. Advances in Neural Information Processing Systems .
- Papamakarios, G., Sterratt, D., Murray, I., 2019b. Sequential neural likelihood: Fast likelihood-free inference with autoregressive flows, in: The 22nd International Conference on Artificial Intelligence and Statistics, PMLR. pp. 837–848.
- Price, K., 1999. An Introduction to Differential Evolution in Corne, D., Dorigo, M. and Glover, F. (eds), New Ideas in Optimization. McGraw-Hill, London.

- Prinz, A.A., Bucher, D., Marder, E., 2004. Similar network activity from disparate circuit parameters. *Nature neuroscience* 7, 1345–1352.
- Raue, A., Kreutz, C., Maiwald, T., Bachmann, J., Schilling, M., Timmer, U.K.J., 2009. Structural and practical identifiability analysis of partially observable dynamical models by exploiting the profile likelihood. *Bioinformatics* 25, 1923–1929.
- Rezende, D., Mohamed, S., 2015. Variational inference with normalizing flows, in: International conference on machine learning (ICML), PMLR. pp. 1530–1538.
- van de Schoot, R., Depaoli, S., King, R., Kramer, B., Märtens, K., Tadesse, M.G., Vannucci, M., Gelman, A., Veen, D., Willemse, J., et al., 2021. Bayesian statistics and modelling. *Nature Reviews Methods Primers* 1, 1–26.
- Shahriari, B., Swersky, K., Wang, Z., Adams, R.P., De Freitas, N., 2015. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE* 104, 148–175.
- Sip, V., Hashemi, M., Dickscheid, T., Amunts, K., Petkofi, S., Jirsa, V., 2023. Characterization of regional differences in resting-state fmri with a data-driven network model of brain dynamics. *Science Advances* 9, eabq7547.
- Snoek, J., Larochelle, H., Adams, R.P., 2012. Practical bayesian optimization of machine learning algorithms 25. URL: https://proceedings.neurips.cc/paper_files/paper/2012/file/05311655a15b75fab86956663e1819cd-Paper.pdf.
- Stimberg, M., Brette, R., Goodman, D.F., 2019. Brian 2, an intuitive and efficient neural simulator. *eLife* 8, e47314. doi:[10.7554/eLife.47314](https://doi.org/10.7554/eLife.47314).
- Storn, R., Price, K., 1997. Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization* 11, 341–359. URL: <https://doi.org/10.1023/A:1008202821328>, doi:[10.1023/A:1008202821328](https://doi.org/10.1023/A:1008202821328).
- Sussillo, D., 2014. Neural circuits as computational dynamical systems. *Current opinion in neurobiology* 25, 156–163.
- Svensson, C.M., Coombes, S., Peirce, J.W., 2012. Using Evolutionary Algorithms for Fitting High-Dimensional Models to Neuronal Data. *Neuroinformatics* 10, 199–218.
- Takens, F., 1981. Detecting strange attractors in turbulence, in: *Dynamical Systems and Turbulence, Warwick 1980: proceedings of a symposium held at the University of Warwick 1979/80*, Springer. pp. 366–381.
- Tan, E., Algar, S., Corrêa, D., Small, M., Stemler, T., Walker, D., 2023. Selecting embedding delays: An overview of embedding techniques and a new method using persistent homology. *Chaos: An Interdisciplinary Journal of Nonlinear Science* 33.
- Tashkova, K., Korosec, P., Silc, J., Todorovski, L., Dzeroski, S., 2011. Parameter estimation with bio-inspired meta-heuristic optimization: modeling the dynamics of endocytosis. *BMC systems biology* 5, 159.
- Tejero-Cantero, A., Boelts, J., Deistler, M., Lueckmann, J.M., Durkan, C., Gonçalves, P.J., Greenberg, D.S., Macke, J.H., 2020. Sbi—a toolkit for simulation-based inference. arXiv preprint arXiv:2007.09114 .
- Virtanen, P., Gommers, R., Oliphant, T.E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., et al., 2020. Scipy 1.0: fundamental algorithms for scientific computing in python. *Nature methods* 17, 261–272.

- Wieland, F.G., Hauber, A.L., Rosenblatt, M., Tönsing, C., Timmer, J., 2021. On structural and practical identifiability. *Current Opinion in Systems Biology* .
- Wilson, H.R., Cowan, J.D., 1973. A mathematical theory of the functional dynamics of cortical and thalamic nervous tissue. *Kybernetik* 13, 55–80.
- Wiqvist, S., Frellsen, J., Picchini, U., 2021. Sequential neural posterior and likelihood approximation. arXiv preprint arXiv:2102.06522 .
- Yan, H., Du, J., Tan, V.Y., Feng, J., 2019. On robustness of neural ordinary differential equations. arXiv preprint arXiv:1910.05513 .
- Zhu, A., Jin, P., Zhu, B., Tang, Y., 2022. On numerical integration in neural ordinary differential equations, in: International Conference on Machine Learning, PMLR. pp. 27527–27547.