Simultaneous Identification of Models and Parameters of Scientific Simulators

Cornelius Schröder [1] **Jakob H. Macke** [1 2]

Abstract

Many scientific models are composed of multiple discrete components, and scientists often make heuristic decisions about which components to include. Bayesian inference provides a mathematical framework for systematically selecting model components, but defining prior distributions over model components and developing associated inference schemes has been challenging. We approach this problem in a simulation-based inference framework: We define model priors over candidate components and, from model simulations, train neural networks to infer joint probability distributions over both model components and associated parameters. Our method, simulation-based model inference (SBMI), represents distributions over model components as a conditional mixture of multivariate binary distributions in the Grassmann formalism. SBMI can be applied to any compositional stochastic simulator without requiring likelihood evaluations. We evaluate SBMI on a simple time series model and on two scientific models from neuroscience, and show that it can discover multiple data-consistent model configurations, and that it reveals non-identifiable model components and parameters. SBMI provides a powerful tool for data-driven scientific inquiry which will allow scientists to identify essential model components and make uncertainty-informed modelling decisions.

1. Introduction

Computational models are a powerful tool to condense scientific knowledge into mathematical equations. These models can be used for interpreting and explaining empirically observed phenomena and predicting future observations. Scientific progress has always been driven by competing models, dating back to disputes about the heliocentric system (Copernicus, 1543). However, newly developed models are rarely that disruptive; instead, they are often created by combining existing components into larger models. For example, the original SIR model (Kermack & McKendrick, 1927) describes the dynamics of infectious diseases by three population classes (**s**usceptible, **i**nfective, **r**ecovered), but was later expanded to include further epidemiological classes (e.g., temporary immune groups, Hethcote, 2000). Similar modularity can be found, for example, in computational neuroscience models: The original Hodgkin-Huxley model (Hodgkin & Huxley, 1952) for the dynamics of action potentials consisted of only two voltage-gated ion channels (K^+, Na^+), but more recent models (McCormick & Huguenard, 1992; Pospischil et al., 2008) are based on compositions of a myriad of different channels (Podlaski et al., 2017). Similarly, there exist many variants of drift-diffusion models (DDM) (Ratcliff, 1978) in cognitive neuroscience: All of them follow the basic concept of modeling the decision process by a particle following a stochastic differential equation and eventually hitting a decision-boundary. There are many possible choices of noise models, drift dependencies, and boundary conditions. This rich model class and many of the different components have been extensively studied on a wide range of experimental measurements (Ratcliff & McKoon, 2008; Latimer et al., 2015; Turner et al., 2015).

How can one automatically infer such models from data, including *both* the compositions of components and the associated parameters? One challenge is posed by the fact that, for many such models, evaluating the likelihood function is not tractable, rendering standard likelihood-based approaches inapplicable. Approximate Bayesian computation (ABC) (Sisson et al., 2018), offers a framework to deal with this challenge in a systematic way, and in the last years, the development of new methods has been fueled by advances in neural network-based density estimation (Papamakarios et al., 2021) leading to new simulation-based inference (SBI) methods (Papamakarios & Murray, 2016; Lueckmann et al., 2017; Cranmer et al., 2020a). SBI has been successfully applied to various fields like astronomy (Dax et al., 2022), robotics (Marlier et al., 2021), neuroscience (Gonçalves et al., 2020; Deistler et al., 2022; Groschner et al., 2022) and cognitive science (Radev et al., 2020; Boelts et al., 2022).

However, in addition to inferring parameters, we also need

[1]Machine Learning in Science, University of Tübingen and Tübingen AI Center, Germany [2]Max Planck Institute for Intelligent Systems, Department Empirical Inference, Tübingen, Germany. Correspondence to: Cornelius Schröder, Jakob Macke <firstname.lastname@uni-tuebingen.de>.

Proceedings of the 41^{st} International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

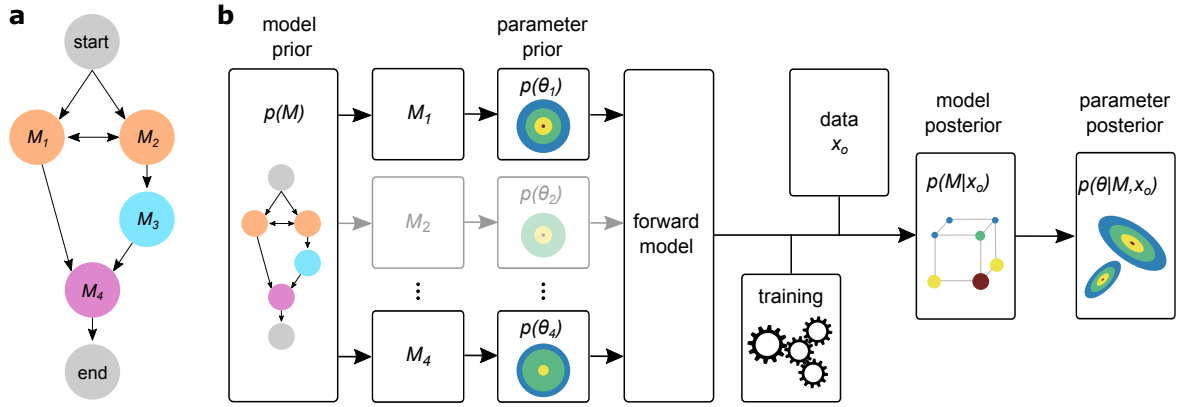

Figure 1. **Simulation-based model inference (SBMI) scheme. (a)** The model prior $p(M)$ is given implicitly by a graph. A random walk from the *start* to the *end* node corresponds to a draw from this prior. **(b)** We first sample from the model prior and the corresponding parameter priors $p(\theta_i)$ to compile a forward model. Following this sampling procedure, we generate training data with which we can learn a approximation of the joint posterior $p(M, \theta|x)$ by factorizing the posterior into $p(M|x)p(\theta|M, x)$. Finally we can evaluate this posterior for some observed data x_o.

to be able to compare and select models comprised of different components to select between competing theories. Standard methods for Bayesian model comparison (or selection) rely on the *Bayes factor* (Kass & Raftery, 1995), i.e., the ratio of model evidence for two different models M_1 and M_2: $B_{12} := p(x_o|M_1)/p(x_o|M_2)$. Multiple approaches have been developed for estimating Bayes factors, most of which are based on (rejection) sampling (Trotta, 2008) and are computationally expensive. Alternative approaches include approximating the model evidence by applying harmonic mean estimators to likelihood emulators (Mancini et al., 2022), or by directly targeting the model posteriors in an amortized manner (Boelts et al., 2019; Radev et al., 2021). While these methods infer the model evidence separately for each model or assume a fixed set of models to compare, our approach allows for a comparison of flexible combinations of model components in a fully amortized manner.

Symbolic regression approaches aim to learn interpretable mathematical equations from observations— while this might seem like a conceptually very different problem, it is methodologically related, as one can also interpret mathematical equations as being composed of different model components. Inferring symbolic equations from data can be tackled by genetic programming (Schmidt & Lipson, 2009; Dubčáková, 2011), by performing sparse regression over a large set of base expressions (Brunton et al., 2016; Bakarji et al., 2022) or by using graph neural networks (Cranmer et al., 2020b). Alternatively, symbolic regression has been approached by designing neural networks with specific activation functions (Martius & Lampert, 2016; Sahoo et al., 2018), optimizing these networks with sparsity priors (Werner et al., 2021) and using Laplace approximations to infer uncertainties over their weights (Werner et al., 2022). Building on the success of transformers, Biggio et al. (2021) introduced a transformer-based approach for symbolic regression, which was recently extended to capture differential equations (Becker et al., 2022).

Our work builds on these advances in both SBI and symbolic regression. However, our goal is to infer *joint* posterior distributions over a set of different model components, *as well as* over their associated parameters. One can interpret our approach as performing fully probabilistic symbolic regression not on 'atomic' symbols, but rather on expression 'molecules' which are provided by domain experts and represent different mechanisms that might explain the observed data. As we will show, accurate inference of joint posteriors is crucial for obtaining interpretable results in the presence of redundant model components: A common situation in scientific applications is that different components are functionally similar (e.g., ion channels with similar dynamics, Podlaski et al., 2017), resulting in explaining-away effects and strongly correlated posterior distributions. Hence, inference methods need to be able to accurately handle such settings to obtain scientifically interpretable results.

We address this challenge by providing a network architecture for joint inference, which includes a flexible representation over model components using mixtures of multivariate binary distributions in the Grassmann formalism (Arai, 2021). Second, for such a procedure to be able to provide parsimonious results, the ability to flexibly specify priors over models is crucially important. Our procedure only requires the ability to generate samples from the prior (like Biggio et al., 2021), without requiring access to evaluations of prior probabilities. Third, our approach is fully *amortized*: Once the inference network has been trained, approximate posteriors over both model components and parameters can be inferred almost instantly, without any computationally expensive MCMC sampling or post-hoc optimizations at inference-time.

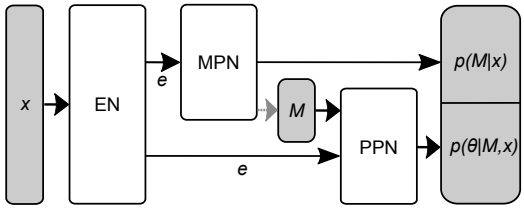

Figure 2. **SBMI network architecture.** Data x is passed through an embedding net (EN). The embedded data e is forwarded to the model posterior network (MPN), which learns posteriors over different model components, and the parameter posterior network (PPN) which learns the posterior distributions over parameters given specific models M. Gray boxes correspond to network inputs / outputs.

In the following, we first describe our inference method (Sec. 2) and showcase it on an additive model related to symbolic regression (Sec. 3.1). We then apply it to DDMs and experimental decision-making data (Sec. 3.2) as well as to Hodgkin-Huxley models and voltage recordings from the Allen Cell Types database (Allen Institute for Brain Science, 2016) (Sec. 3.3) and show that our method can successfully retrieve interpretable posteriors over models.

2. Method

Our proposed method, *simulation-based model-inference* (SBMI), performs inference over a model M consisting of different model components M_i and their associated parameters θ_i. More specifically, we use a neural posterior estimation (NPE) method to approximate a joint posterior distribution $p(M, \theta|x_o) = p(M|x_o)p(\theta|M, x_o)$ given some observed data x_o end-to-end (Fig. 1). While we take a related approach to previous NPE methods (Papamakarios & Murray, 2016; Lueckmann et al., 2017; Greenberg et al., 2019), we lift the assumption of a fixed simulator and include the inference over model components into our framework. We therefore assume that we have a 'black-box' model from which we can draw samples $x_j \sim p(x|M, \theta)$, but don't necessarily have access to the likelihood, any other internal states, or gradients of the model. Approximate Bayesian inference is performed by first generating simulations which are then used to learn posterior distributions. These can be evaluated in an amortized manner for new observations x_o to get the full joint posterior $p(M, \theta|x_o)$.

2.1. Priors and Data Generation

To allow maximal flexibility in designing appropriate priors, SBMI only requires access to an implicit prior distribution from which we can sample models. We here define the model prior by a directed graph with dynamically changing weights, defined as a triplet $(\mathcal{M}, \mathcal{E}, \mathcal{R})$. The set of vertices $\mathcal{M} = \{M_i\}_{i\in 0,...,N+1}$ corresponds to the model components and additional start/end node M_0 and M_{N+1} (Fig. 1a). The set of edges $\mathcal{E} = \{e_{ij} = (M_i, M_j, w_{ij})|M_i, M_j \in \mathcal{M}, w_{ij} \in \mathbb{R}_{\geq 0}\}$ are directed, with weights w_{ij}. To sample from the prior, we perform a random walk on the graph with $p(M_i|M_j) = \frac{w_{ij}}{\sum_l w_{il}}$ and represent each sampled model $M = (M_1, ..., M_N)$ as an ordered binary vector of length N. The set of rules $\mathcal{R} = \{R_k|R_k : (\mathcal{E}, S) \to \mathcal{E},\ S \subset \mathcal{M}\}_{k\in K}$, with index set K, defines how the weights w_{ij} are updated during a random walk. We assume the graph to be *conditionally acyclic*: While the initial edges $\mathcal{E}$ can include cycles (e.g. Fig. 3a), the updating rules $\mathcal{R}$ ensure that no cycle occurs in prior samples. By changing the edge weights we can encode prior knowledge, for example, to favour simple models over complex models, or to encourage (or discourage) the co-occurrence of specific model components. Pseudocode for the sampling procedure and example updating rules can be found in Appendix A3. This graph representation gives us the possibility to flexibly encode prior knowledge of the model by carefully defining its structure and weights with the help of domain expertise. Once we have sampled a model M, we define the prior of the corresponding model parameters as the product of the component-specific priors: $p(\theta|M) = \prod_{i|M_i=1} p(\theta_i)$, i.e. the parameter vector θ is of variable size and matches a specific model M. The component-specific priors $p(\theta_i)$ can correspond to any continuous, potentially multivariate, distribution.

To generate training data for learning an approximation of $p(M, \theta|x_o)$ we need a 'compiler' that turns the model representation (M, θ) into a simulator which then generates synthetic data x. These compilers and simulators will generally be specific to the model type and based on domain-specific toolboxes. In our numerical experiments, we built a flexible interface to symbolic calculations based on *SymPy* (Meurer et al., 2017) for the additive model (Sec. 3.1), the *PyDDM* toolbox (Shinn et al., 2020) for the drift-diffusion model (Sec. 3.2) and implemented a modularised version of the Hodgkin-Huxley model in *JAX* (Bradbury et al., 2018) (Sec. 3.3).

2.2. Inference

We want to perform inference over the joint posterior distribution $p(M, \theta|x)$ of the model and its parameters, given some data x. We can factorize this distribution $p(M, \theta|x) = p(M|x)p(\theta|M, x)$ and approximate it by jointly learning two coupled network modules (Fig. 2): The first module learns an approximation to the model posterior $q_\psi(M|x) \approx p(M|x)$ and the second one an approximation to the parameter posterior $q_\phi(\theta|M, x) \approx p(\theta|M, x)$ conditioned on the data and the model. As the data x might be high-dimensional (or, in principle, of variable length) we use an additional embedding net to project it to a low-dimensional representation before passing it to the posterior networks. This network can be replaced by 'summary statis-

tics' which capture the main features of the data (see Sec. 3.3).

Model Posterior Network To approximate the multivariate binary model posterior $p(M|x)$ we introduce a new family of mixture distributions: mixture of multivariate binary Grassmann distributions (MoGr). Multivariate binary Grassmann distributions were recently defined by Arai (2021), and allow for analytical probability evaluations. A Grassmann distribution is defined by its probability mass function on a n-dimensional binary space, for which closed-form expressions for marginal and conditional distributions are available. This in turn can be directly used for efficient sampling. An n-dimensional binary Grassmann distribution $\mathcal{G}$ on $Y = (Y_1, ..., Y_n)$ is parameterized by a $n \times n$ matrix Σ which is analogous to the covariance of a normal distribution, but not necessarily symmetric. The mean of the marginal distribution is represented on the diagonal and the covariance is the product of the off-diagonal elements:

$$\mathbb{E}[Y_i] = \Sigma_{ii}, \qquad \text{Cov}[Y_i, Y_j] = -\Sigma_{ij}\Sigma_{ji}.$$

For a valid distribution $(\Sigma^{-1} - I)$ must be a P_0-matrix, but has otherwise no further constraints (Arai, 2021). We thus define a mixture of Grassmann distribution as $\text{MoGr}(Y) = \sum_i \alpha_i \mathcal{G}_i(Y)$ for a finite partition $\sum_i \alpha_i = 1$ and Grassmann distributions $\mathcal{G}_i$. We denote the corresponding conditional distribution by $\text{MoGr}(Y|e) = \sum_i \alpha_i \mathcal{G}_i(Y|e)$, for some real-valued context vector e (which will be the embedded data in our case). Further details (including some key properties, and implementation details) in Appendix A2. We trained the model posterior $p(M|x)$ represented as conditional MoGr distribution $\text{MoGr}(M|x) \approx q_\psi(M|x)$ by minimizing the negative log-likelihood. The model loss $\mathcal{L}_M$ is therefore defined by $\mathcal{L}_M(\psi) = -\log q_\psi(M|x)$.

Parameter Posterior Network The parameter posterior network $q_\phi(\theta|M, x)$ needs the flexibility to deal with different dimensionalities, as θ is only defined when the respective model component ($M_i = 1$) that uses θ_i is included. While recent SBI approaches typically used normalizing flows (Papamakarios et al., 2021) for parameter inference, we use a mixture density network (MDN) of Gaussian distributions on the full-dimensional parameter space (with dimension $n = \sum_i \dim(\theta_i)$) and marginalize out the non-enclosed model components. This allows the network to learn dependencies across model components (which is critical, e.g., to account for compensation effects between redundant components). We construct this flexible MDN by defining for every θ its complement θ^C as the parameter dimensions not present in θ and $\bar{\theta} = (\theta, \theta^C)$. We further define $\bar{p}$ as the n-dimensional distribution acting on $\bar{\theta}$. We can now define the parameter posterior network $q_\phi(\theta|M, x)$ by marginalizing out θ^C,

$$q_\phi(\theta|M, x) = \int \bar{p}(\bar{\theta}|M, x) d\theta^C.$$

We use the standard NPE loss (Papamakarios & Murray, 2016) for the parameter posterior network $\mathcal{L}_\theta : \mathcal{L}_\theta(\phi) = -\log\ q_\phi(\theta|M, x)$. The final loss function for the training of the three different network modules (embedding net, model, and parameter posterior network) is then defined as the expected sum of the two posterior losses: $\mathcal{L}(\psi, \phi) = \frac{1}{L}\sum_l \mathcal{L}_{M_l}(\psi) + \mathcal{L}_{\theta_l}(\phi)$, for a batch of training samples $\{(\theta_l, M_l, x_l)\}_l$ of size L. In Proposition A6.1 we show that optimizing this loss functions minimizes the expected Kullback-Leibler divergence between true joint posterior $p(M, \theta|x)$ and the approximated posterior

$$\mathbb{E}_{p(x)}\big[D_{KL}(p(M, \theta|x)||q_\phi(M|x)q_\psi(\theta|M, x)\big],$$

and is therefore retrieving the object of interest. See Algorithm 2 for pseudocode. Our implementation is based on the *sbi* toolbox (Tejero-Cantero et al., 2020) (see Appendix A4 for details).

Local and Global Uncertainties: SBMI allows us to calculate two different uncertainties for the posterior predictives, depending on whether uncertainty about model-choice is taken into account or not: *Local* uncertainties (Werner et al., 2022) are defined as the uncertainty of parameter posteriors conditioned on a specific model M_i: $x \sim p(x|M_i, \theta)$ with $\theta \sim p(\theta|M_i, x_o)$. In contrast, for the *global* uncertainty, the joint posterior is taken into account: $x \sim p(x|M, \theta)$ with $M, \theta \sim p(M, \theta|x_o)$.

Simulation-based Calibration To validate the inferred parameter posteriors we perform simulation-based calibration (SBC) on the marginal statistics (Talts et al., 2018). In SBC each marginal of the true parameter θ_o is ranked according to the marginals of samples from the parameter posterior $p(\theta|x_o)$ for a simulated data sample $x_o \sim p(x|\theta_o)$. For a well calibrated posterior, the ranks follow a uniform distribution, which we evaluated by a classifier-two-sample-test (c2st, Friedman, 2003).

3. Experiments

We demonstrate SBMI on three model classes: An illustration on an additive model of a one-dimensional function $f(t)$, variants of the *drift diffusion model* (DDM) from cognitive science and variants of the *Hodgkin-Huxley model*.

3.1. Additive Model

For the additive model, we used two linear, a quadratic, a sinusoidal, and two different noise terms (details in Table S1), all evaluated on an equidistant grid on the interval $[0, 10]$. These could be seen as the 'base functions' in a symbolic

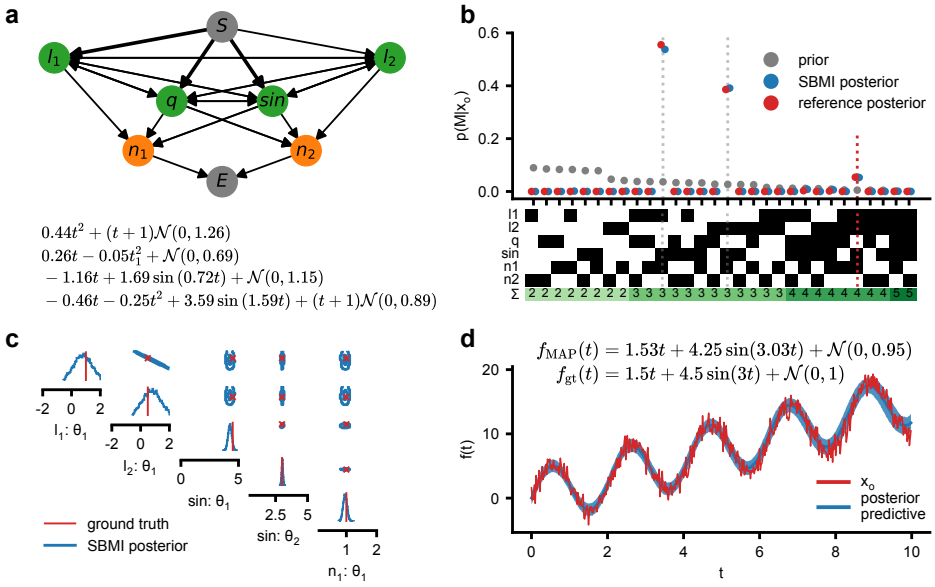

Figure 3. **Additive model. (a)** Model prior represented as a graph, the width of the edges corresponds to their initial weights, which change dynamically. A random walk from *start* (S) to *end* node (E) corresponds to one draw from the prior. Four prior samples are shown. **(b)** Empirical prior distribution, reference and SBMI posterior distribution for one example observation, generated by the model highlighted by the red dashed line. The model vectors are shown as binary image, black indicating the presence of the specific model component. SBMI accurately recovers the posterior over model components. Marginal distributions in Fig. S2. **(c)** One- and two-dimensional marginals of the parameter posterior inferred with SBMI, conditioned on the 'true model' (red dotted line in (b)). Note the strongly negatively correlated (degenerate) posterior between the redundant model components l_1 and l_2. Parameter posteriors for additional models in Fig. S2. **(d)** Predictive samples for an observation x_o from f_{gt}. Blue: Mean $\pm$ std. as local uncertainties of the posterior predictives $x \sim p(x|\theta, M)$ with $\theta \sim p(\theta|M, x_o)$.

regression task. To investigate how SBMI fares in the presence of non-identifiability, we included two identical linear components which only differ in their prior probability. We defined the model prior as a dynamically changing graph (Fig. 3a) which favors simpler models (Fig. 3b, Appendix A7.1). We used a CNN followed by fully connected layers for the embedding net (Appendix A7). We generated a dataset of 500k prior samples, of which 10% were used as validation data.

In the presented model, we have access to the likelihood function $p(x_o|M, \theta)$, and can approximate the model evidence via (importance) MC sampling. We approximate the model evidence $p(M|x_o) \approx \hat{p}_{\text{reference}}(M|x_o) \sim p(x_o|M)p(M)$ by sampling for each model M corresponding parameters θ_i^j and evaluating the likelihood $p(x_o|M)$ (Appendix A5). We call the resulting approximation *reference posterior*, and will use it to evaluate the accuracy of the posterior inferred by SBMI. As the parameter space for θ can be high-dimensional, and the corresponding posterior distribution $p(\theta|M, x_o)$ can be narrow, a reliable numerical approximation needs an extensive amount of samples and model evaluations for each of the model combination M. It is therefore not feasible for larger model spaces.

Across 100 observations x_o for which we computed reference posteriors, the Kullback–Leibler divergence (KL) between the reference posterior and the SBMI posterior $\text{KL}(\hat{p}_{\text{reference}}(M|x_o)||q_\psi(M|x_o))$ decreased substantially with the number of training samples. When we replaced the MoGr distribution by a 'flat' categorical distribution, and left all other components unchanged, the inference is much less data efficient (Fig. 4a). Additionally, we compared the marginal distribution of the model posterior to the ground truth model. The performance of the marginal model posterior distributions inferred by SBMI is very similar to that of the reference posteriors (Table S3) with a correlation of 0.85 (Fig. S3). We note that, in initial experiments in which we used masked autoregressive density estimators (MADE) (Germain et al., 2015) (instead of the Grasmann mixtures) exhibited worse performances in comparison (Fig. S1), indicating the power and flexibility of MoGr distributions.

For the evaluation of the joint posterior $p(M, \theta|x_o)$, we focused on the evaluation in the predictive (data) space. To this end, we sampled models $M_l \sim q_\psi(M|x_o)$ and associated parameters $\theta_l \sim q_\phi(\theta|M_l, x_o)$ from the inferred posteriors for 1k test observations x_o and ran the forward model $x_l \sim p(x|M_l, \theta_l)$. Based on these simulations, we calculated the root-mean-squared-error (RMSE) of the simulations x_l to the observed data x_o. The average RMSE

between posterior predictive samples follow the same trend as the KL divergence and approach the RMSE between the observations x_o and samples with the same ground truth parameters at around 200k training samples (Fig. 4b).

Next, we showcase SBMI for a specific example observation x_o in which the ground truth model has two linear, a sinusoidal, and a stationary noise component (Fig. 3b-d): The SBMI model posterior matches nearly perfectly the reference posterior and predicts the linear components as expected, ordered by the prior probabilities (Fig. 3b). The parameter posterior obtained with SBMI and conditioned on the ground truth model accurately recovers the ground truth parameters (Fig. 3c). Accessing the joint posterior distribution enables us to first see the perfectly correlated parameter distribution for the slope parameter of the linear components. Second, we detect compensations mechanisms for a model which contains only one linear component: In this case, the predicted parameter for l_1 is the sum of the ground truth parameters of l_1 and l_2, resulting in the same functional expression (Fig. 3d). For the posterior predictives (Fig. 3d) we see that most of the observed data x_o lies within an uncertainty bound of one standard deviation around the mean prediction. The local uncertainties overlap almost perfectly in this case, as all models with non-negligible posterior mass have the same expressional form. With the inferred model posterior we can easily compute the Bayes factors via $\frac{p(M_1|x_o)}{p(M_2|x_o)}\frac{p(M_2)}{p(M_1)}$ to compare different models on an observation x_o, and an example for Fig. 3 is shown in Appendix A7.2.

In two additional sets of experiments we increased the space of model components to eleven and twenty (see Tab. S2 and Tab. S4 for details). As calculating reference posteriors is not computationally feasible any more, we focused on the evaluation in the predective space and calculated in the first set the RMSE for different numbers of training samples for MoGr as well as for categorical model posterior distributions. While the MoGr did almost reach the lower bound with 1M training samples, the categorical distribution had a much worse performance and did not reach this lower bound. When we increased the number of mixture components for the MoGr as well as for the Gaussian MDN from three up to twelve, the RMSE did not change substantially, except that less mixture components were preferable in a low data regime (Fig. S4a). In the second set of experiments with twenty model components, the categorical distribution would need to learn all $2^{20} \approx 1M$ possible combinations, which is not feasible any more. However, the MoGr was still able to learn the posterior distributions and the predictive performance reached almost the lower bound with 1M training sample (Fig. S4b).

When we applied SBC to the parameter posteriors of the additive model, which were conditioned on the "ground truth model", we found well calibrated posterior distributions for all 30 possible models and parameters (Fig. 4c). This still holds true for the large additive model, for which we included all models with at least 50 samples in the test dataset of 100k samples (Fig. 4d). When we only looked at the 100 most likely models the c2st values were even more tightly centered around one half. We found no systematic bias for individual parameters in our inference method and most posterior ranks fall into the 99% confidence interval of a uniform distribution (Fig. S5). These results show that the parameter posteriors are well calibrated for the additive model, even for models which are less likely under the prior distribution and the posteriors reflect the underlying uncertainty well.

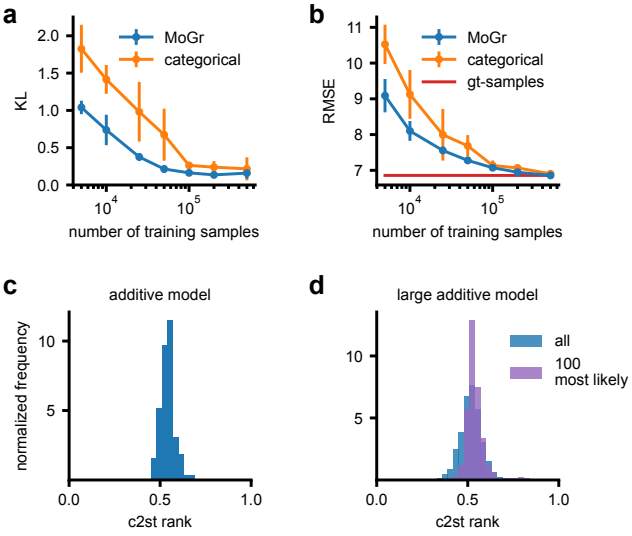

Figure 4. **SBMI performance for the additive model. (a)** KL divergence of the SBMI model posterior $q_\psi(M|x_o)$ to the reference posterior $\hat{p}_{\text{ref}}(M|x_o)$ for 100 observations x_o. **(b)** Posterior predictive performance in terms of RMSE between 1k observations x_o and posterior samples x_l. Red line indicates RMSE between x_o and samples from the ground truth (gt) model as lower bound. (a) and (b) show mean and std. for 5 training runs and different numbers of training samples (from 5k to 500k). **(c)** Histogram of the c2st ranks for the additive models with six components with a c2st mode of 0.54 (0.48/0.61 as .05/.95 percentiles). A value of 0.5 indicates a well calibrated posterior for which the rank statistics are indistinguishable from a uniform distribution. **(d)** Same as (c) for the additive model with eleven components with a c2st mode of 0.52 (0.43/0.62, 'all') and 0.53 (0.48/0.61, '100 most likely'). See also Fig. S5 for individual SBC plots.

3.2. Drift-Diffusion Model

After this illustrative example, we turn to DMMs, a scientific model class that we will apply to experimental data. DDMs can be described by a stochastic differential equation for a decision variable z: $\mathrm{d}z = d(z,t)\mathrm{d}t + \mathrm{d}W$, with initial condition z_0, drift term d, and a Wiener noise process

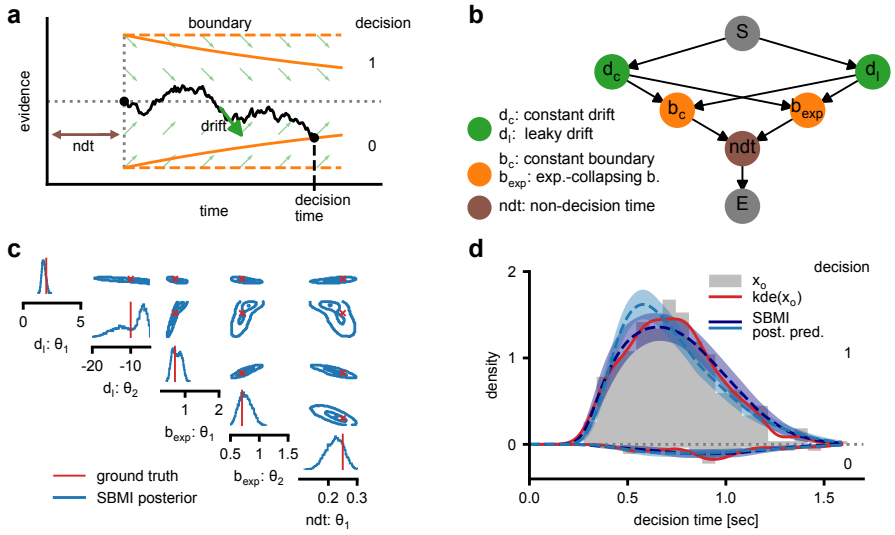

Figure 5. **SBMI on Drift-Diffusion Models. (a)** A decision process is modelled by a one-dimensional stochastic process. A binary decision is taken once the process hits the upper or lower boundary, resulting in a two-dimensional output (a continuous decision time and a binary decision). **(b)** The model prior is a graph consisting of two drift (d_c, d_l) and two boundary (b_c, b_{exp}) components, as well as a non-decision time (ndt). **(c)** Example parameter posterior inferred with SBMI for which both, the ground truth model and the predicted model, have leaky drift and exponentially collapsing boundary conditions. **(d)** Posterior predictives with local uncertainties as mean $\pm$ std. for the two most likely models (dark blue with $q_\psi(M|x_o) = 0.75$ and light blue with $q_\psi(M|x_o) = 0.25$).

W. A decision is taken when the decision variable hits a boundary $|d(z,t)| \geq b(t)$ (Fig. 5a). An additional parameter delays the starting time of the process ('non-decision time'). We included two different drift terms (constant and leaky), two boundary conditions (constant and exponentially collapsing), and the non-decision time to our prior (Fig. 5b, Appendix A7.3), resulting in a highly flexible model class. Similar models have previously been applied to experimental data (Shinn et al., 2020).

Training data was generated with the *pyDDM* toolbox (Shinn et al., 2020) and for each θ we sampled 400 identically distributed (iid) trials trial. These were embedded by a permutation invariant embedding network (Chan et al., 2018; Radev et al., 2020) (Appendix A7).

For the DDM, we don't have efficient access to the likelihood and therefore cannot compute reference posteriors. To still evaluate the performance of SBMI, we focus on the evaluation of model posteriors and predictive performances for a test set of 1k data points. The average marginal performance of the model posterior for the drift and boundary components is 0.87$\pm$0.21 (std.) (see Table S7 for individual performances). For about 40% of the test data we get highly certain model posteriors with $p(M|x_o) > 0.99$, indicating that model identifiability is dependent on the observed data x_o. To measure the performance of the posterior predictives, we compared the mean decision times, the standard deviation of the decision times, and the number of correct trials to the observed data x_o. Additionally, we used the mean-squared error (MSE) on the weighted density functions of the two different decisions, similar to (Shinn et al., 2020). The different measures on the posterior predictives for the test data are close to their lower bounds (Table 1), calculated on trials resampled from a model with the ground truth parameter. This suggests that, even for non-identifiable models, the SBMI inferred posterior predictives are close to data from the ground truth model.

Table 1. **DDM posterior predictive performance.** Comparison of decision times (mean μ and std. σ) of the ground truth data ($\hat{\cdot}$) to posterior predictive samples. The lower bound is based on resampling 400 trials with the ground truth parameters. Mean and standard deviation for 1k test points.

Measure	Lower bound	Posterior
decision time: $\lvert\mu - \hat{\mu}\rvert$	0.03 (0.04)	0.08 (0.21)
decision time: $\lvert\sigma - \hat{\sigma}\rvert$	0.21 (0.27)	0.26 (0.35)
deviation correct trials in %	1.10 (1.37)	1.56 (2.64)
MSE on densities ($\cdot 10^{-2}$)	3.14 (5.83)	3.23 (6.57)

For an example observation x_o from a model with a leaky drift component and exponential collapsing boundaries, the 'true' model has a posterior probability of 0.75 and a model with a constant drift instead has a posterior probability of 0.25, resulting in a Bayes factor of $B = 2.32$, or a 'barely worth mentioning' difference (Jeffreys, 1998). For the 'true' model the ground truth parameters lie in regions of high parameter posterior mass, with some uncertainty, especially in the leak parameter θ_2 of the drift component (Fig. 5c). The posterior predictives match the data well if conditioned on the 'true' model. For the model with the constant drift term,

we see a slight skew to earlier decision times, compared to the model with leaky drift (Fig. 5d). If we inspect the global uncertainties (Fig. S6d) we see a good correspondence for the global uncertainties, also reflected in the MSE losses (scaled by 10^2): For trials resampled with the ground truth parameters we find an MSE of 0.57±0.13 which matches the MSE of the first model (0.58±0.14) and the second model is only slightly worse (0.61±0.14). Further inspecting the posterior distributions shows that the model with the constant drift term exhibits shorter non-decision times, larger initial boundaries and faster collapsing boundaries (Table S9). Interpreting the inferred values model-independent as behavioral variables can therefore be difficult, as different models might lead to different inferred values.

DDM on Experimental Data To demonstrate SBMI on empirical data, we used a published dataset of perceptual decision-making data from monkeys (Roitman & Shadlen, 2002) performing a random dot motion discrimination task. Moving dots with different coherence rates (0, 3.2, 6.4, and 12.8%) were visually presented and animals had to identify the direction of movement (Appendix A7.3).

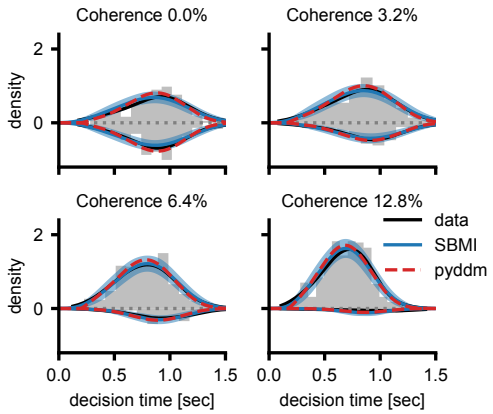

Figure 6. **SBMI on experimental data.** Experimental data with histograms (grey), mean posterior predictives ± 2std. and *pyDDM* fits for different coherence rates.

When we used the trained posterior network to perform amortized inference on the different experimental conditions of the experimental data, the model posterior is certain about the leaky drift and exponentially collapsing boundary component with $p(M|x_o) \approx 1$ for all coherence rates. For all measures on the posterior predictives we found similar mean performances for the SBMI inferred models compared to point estimates (Table S10). But, as expected, the MSE had higher variances in the different experimental conditions compared to the variance of multiple point estimates (Table S11). This can also be seen in the decision time densities of the posterior predictives for which the experimental data lies within the uncertainty bounds (Fig. 6), whereas the predictives of the point estimates from *pyDDM* are not distinguishable. However, in the parameter space we see that different point estimates are spread out for some of the parameters, and all lie in regions of high SBMI parameter posterior mass. An example of the two-dimensional marginals for the coherence of 6.4% is shown in Figure S7.

3.3. Hodgkin-Huxley Model

Finally, we apply SBMI to the Hodgkin-Huxley model, a biophysical model for spike generation in neurons. We include a leakage current, four different voltage-gated ion channel types (Na, K, K_m, Ca_L) and a noise term (Appendix A7.4). We encode the domain knowledge that Na and K channels are necessary for spike generation, resulting in a simple prior (Fig. S8), which could be easily extended to further channel types. We replace the embedding net by commonly used summary statistics (Gonçalves et al., 2020; Scala et al., 2021), but leave the inference networks unchanged.

When we evaluated SBMI on 1k synthetic test traces, we found a mean marginal performance of the model posterior of 0.85±0.16 (std.) for the two essential model components K_m and Ca_L. This performance rises drastically to 0.96±0.03 if we only include voltage traces with spikes in the test data ($n = 439$), indicating that the model components are well identifiable if spikes are present. When inspecting the parameter posteriors for the presented examples, the ground truth parameters lie in regions of high posterior mass (Fig. S11a). For the MSE on the posterior predictives' normalized summary statistics we get 0.06±0.09 for the example traces shown in Fig. 7 and S9a, indicating a good performance in the predictive space.

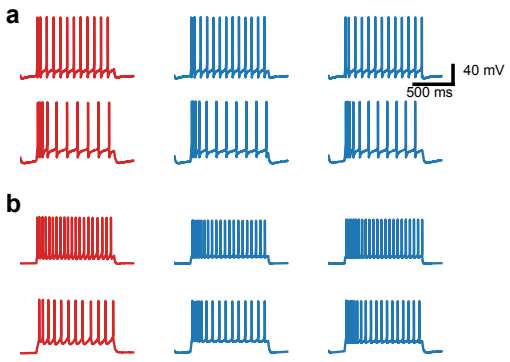

Figure 7. **Posterior predictives of the Hodgkin-Huxley model.** **(a)** Two synthetic samples (red) with two posterior predictive samples each (blue). **(b)** Two voltage recordings from the Allen Cell database with two posterior predictive samples each. See Fig. S9 for more samples, Fig. S10c,d for example summary statistics and Fig. S11 for SBMI posteriors.

To apply SBMI to experimental voltage recordings we took ten voltage traces from the Allen Cell Types database

(Allen Institute for Brain Science, 2016) previously used in (Gonçalves et al., 2020). The model posterior identified Ca_L channel in some (4 out of 10) of the recordings, which were previously not used to fit these traces. The posterior predictives look slightly worse than for the synthetic data, but still capturing the main features (Fig. 7b), resulting in a MSE of 0.24 $\pm$0.11 (std.) on normalized summary statistics.

To showcase the influence of the model prior, we run the same SBMI inference scheme with a fully connected model prior, and therefore omitted any domain knowledge for the model structure which we had included before. First of all, for a training dataset of 100k samples this resulted in only 31k spiking models (compared to 41k in the initial experiment). After training, SBMI shows still reasonable performance for most synthetic samples (MSE of 0.10 ± 0.14 on the summary statistics). Additionally, for all observations from the Allen dataset the model posterior still had the same ion channel composition as in our initial experiments, except for one trace. However, the posterior predictive performance on the Allen dataset was substantially worse (MSE of 0.48 ± 0.55 on the summary statistics, Fig. S10).

When we applied SBC to the Hodgkin-Huxley model we found a c2st mode of 0.53 (with 0.49/0.64 as .05/.95 percentiles, Fig. S12a). While this indicates overall a good calibration, we found parameter specific differences. For example the posteriors for g_L were slightly under confident and for E_L the true parameter was systematically overestimated (Fig. S12b). However, given the relatively small training dataset of 100k prior samples, and the complexity of the Hodgkin-Huxley model, the posteriors are relatively well calibrated.

4. Discussion

We presented SBMI, a method for inferring posterior distributions over both model components of scientific simulators and their associated parameters end-to-end. For the model inference network, we used a mixture of conditional multivariate binary Grassmann distributions to flexibly and efficiently approximate posterior distributions over models. To deal with the variable dimensionality of the parameter posterior, we used a Gaussian Mixture Density Network which allows efficient marginalization over absent model components during training time. By inferring the joint posterior distribution over models and parameters, SBMI allows us to learn parameter dependencies between model components and compensatory mechanisms, in a fully amortized way.

We first showcased SBMI on additive models and showed that posteriors retrieved by SBMI are in very close agreement with reference posteriors. Our application of DMMs yielded posteriors with highly accurate posterior predictives, and allowed identification of compensatory mechanisms for some parameters. This demonstrates the importance of a 'model-aware' interpretation of parameter posteriors, enabled by SBMI. On experimental data, SBMI automatically retrieved a model which was previously suggested by domain-scientists (Shinn et al., 2020), and which outperformed simpler alternatives. Finally, we ran SBMI on the Hodgkin-Huxley model, both on synthetic data and voltage recordings from the Allen Celltype database. For most of the traces we recover the same model structure as previously used to fit these traces but for some traces additional Ca_L channels might be advantageous. Further in-depth experiments with more channel types could be run to identify the cell mechanisms more exactly.

SBMI gives us not only access to full parameter posteriors, but also infers the uncertainty related to the model *choice* itself and potential interactions between the parameters of different model components. For symbolic regression, a similar perspective was presented (Werner et al., 2022) who estimated local uncertainties by Laplace approximations, and used a fixed number of equations for the global uncertainty. While this gives some measure of uncertainty, SBMI is able to recover the *full* posterior and its associated uncertainty.

SBMI enables us to compare different model compositions in a fully amortized manner, allowing scientists to test and compare a large set of competing theories without the need to exhaustively infer each possible combination individually for separate comparisons based on Bayes-factors. Additionally, the amortized nature of SBMI makes it easy to check how robust posteriors over models are when observations change. Similarly, amortization also makes it straightforward to perform additional coverage and calibration tests (Zhao et al., 2021; Hermans et al., 2021).

For real-world applications the success of SBI relies on appropriate prior choices (Oesterle et al., 2020; Deistler et al., 2022) and a well chosen model prior for SBMI can not only increase the data efficiency by simulating more "meaningful" models, but can also enhance the model inference by decreasing the space of possible combinations. Although the presented framework already covers many scientific scenarios, the representation of model prior could be further enhanced by lifting the restriction of an ordered model vector of fixed length. Using more flexible embedding networks like transformers (Vaswani et al., 2017; Lee et al., 2019) could be used to generalize SBMI to simulator outputs x of varying size (Biggio et al., 2021).

In summary, SBMI provides a powerful tool for data-driven scientific inquiry. It will allow scientists to systematically identify essential model components which are consistent with observed data. Incorporating the uncertainty into their model choices will help to resolve competing models and theories.

Impact Statement

We used data recorded from animal experiments in monkey and mouse. The data we used were recorded in independent experiments and are publicly available (Roitman & Shadlen, 2002; Allen Institute for Brain Science, 2016).

While our paper presents work whose goal is to advance the field of Machine Learning and scientific discovery, there are many potential societal consequences of our work, none which have unique implications that warrent specifically being highlighted here.

Acknowledgements

We thank all group members of the Mackelab for their insightful discussions and valuable feedback on the manuscript. This work was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – EXC number 2064/1 – 390727645, and under SFB 1233, Robust Vision: Inference Principles and Neural Mechanisms, project 6, number: 276693517 and by the German Federal Ministry of Education and Research (BMBF): Tübingen AI Center, FKZ: 01IS18039A. Co-funded by the European Union (ERC, DeepCoMechTome, 101089288).

References

Akiba, T., Sano, S., Yanase, T., Ohta, T., and Koyama, M. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019.

Allen Institute for Brain Science. Allen cell type database, 2016. URL http://celltypes.brain-map.org/.

Arai, T. Multivariate binary probability distribution in the grassmann formalism. *Physical Review E*, 103(6):062104, 2021.

Bakarji, J., Champion, K., Kutz, J. N., and Brunton, S. L. Discovering governing equations from partial measurements with deep delay autoencoders. *arXiv preprint arXiv:2201.05136*, 2022.

Becker, S., Klein, M., Neitz, A., Parascandolo, G., and Kilbertus, N. Discovering ordinary differential equations that govern time-series. *arXiv preprint arXiv:2211.02830*, 2022.

Biggio, L., Bendinelli, T., Neitz, A., Lucchi, A., and Parascandolo, G. Neural symbolic regression that scales. In *International Conference on Machine Learning*, pp. 936–945. PMLR, 2021.

Boelts, J., Lueckmann, J.-M., Goncalves, P. J., Sprekeler, H., and Macke, J. H. Comparing neural simulations by neural density estimation. In *Conference on Cognitive Computational Neuroscience*, pp. 1289–1299, 2019.

Boelts, J., Lueckmann, J.-M., Gao, R., and Macke, J. H. Flexible and efficient simulation-based inference for models of decision-making. *Elife*, 11:e77220, 2022.

Bradbury, J., Frostig, R., Hawkins, P., Johnson, M. J., Leary, C., Maclaurin, D., Necula, G., Paszke, A., VanderPlas, J., Wanderman-Milne, S., and Zhang, Q. JAX: composable transformations of Python+NumPy programs, 2018.

Brunton, S. L., Proctor, J. L., and Kutz, J. N. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the national academy of sciences*, 113(15):3932–3937, 2016.

Chan, J., Perrone, V., Spence, J., Jenkins, P., Mathieson, S., and Song, Y. A likelihood-free inference framework for population genetic data using exchangeable neural networks. *Advances in neural information processing systems*, 31, 2018.

Copernicus, N. De revolutionibus orbium coelestium, 1543.

Cranmer, K., Brehmer, J., and Louppe, G. The frontier of simulation-based inference. *Proceedings of the National Academy of Sciences*, 117(48):30055–30062, 2020a.

Cranmer, M., Sanchez Gonzalez, A., Battaglia, P., Xu, R., Cranmer, K., Spergel, D., and Ho, S. Discovering symbolic models from deep learning with inductive biases. *Advances in Neural Information Processing Systems*, 33: 17429–17442, 2020b.

Dax, M., Green, S. R., Gair, J., Deistler, M., Schölkopf, B., and Macke, J. H. Group equivariant neural posterior estimation. In *International Conference on Learning Representations*, 2022.

Deistler, M., Macke, J. H., and Gonçalves, P. J. Energy-efficient network activity from disparate circuit parameters. *Proceedings of the National Academy of Sciences*, 119(44):e2207632119, 2022.

Dubčáková, R. Eureqa: software review, 2011.

Friedman, J. H. On multivariate goodness-of-fit and two-sample testing. *Statistical Problems in Particle Physics, Astrophysics, and Cosmology*, 1:311, 2003.

Germain, M., Gregor, K., Murray, I., and Larochelle, H. Made: Masked autoencoder for distribution estimation. In *International conference on machine learning*, pp. 881–889. PMLR, 2015.

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., et al. Training deep neural density estimators to identify mechanistic models of neural dynamics. *Elife*, 9:e56261, 2020.

Greenberg, D., Nonnenmacher, M., and Macke, J. Automatic posterior transformation for likelihood-free inference. In *International Conference on Machine Learning*, pp. 2404–2414. PMLR, 2019.

Groschner, L. N., Malis, J. G., Zuidinga, B., and Borst, A. A biophysical account of multiplication by a single neuron. *Nature*, 603(7899):119–123, 2022.

Hagberg, A. A., Schult, D. A., and Swart, P. J. Exploring network structure, dynamics, and function using networkx. In Varoquaux, G., Vaught, T., and Millman, J. (eds.), *Proceedings of the 7th Python in Science Conference*, pp. 11 – 15, Pasadena, CA USA, 2008.

Hermans, J., Delaunoy, A., Rozet, F., Wehenkel, A., and Louppe, G. Averting a crisis in simulation-based inference. *arXiv preprint arXiv:2110.06581*, 2021.

Hethcote, H. W. The mathematics of infectious diseases. *SIAM review*, 42(4):599–653, 2000.

Hodgkin, A. L. and Huxley, A. F. A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of physiology*, 117 (4):500, 1952.

Jeffreys, H. *The theory of probability*. OuP Oxford, 1998.

Kass, R. E. and Raftery, A. E. Bayes factors. *Journal of the american statistical association*, 90(430):773–795, 1995.

Kermack, W. O. and McKendrick, A. G. A contribution to the mathematical theory of epidemics. *Proceedings of the royal society of london. Series A, Containing papers of a mathematical and physical character*, 115(772):700–721, 1927.

Latimer, K. W., Yates, J. L., Meister, M. L., Huk, A. C., and Pillow, J. W. Single-trial spike trains in parietal cortex reveal discrete steps during decision-making. *Science*, 349(6244):184–187, 2015.

Lee, J., Lee, Y., Kim, J., Kosiorek, A., Choi, S., and Teh, Y. W. Set transformer: A framework for attention-based permutation-invariant neural networks. In *International conference on machine learning*, pp. 3744–3753. PMLR, 2019.

Lueckmann, J.-M., Goncalves, P. J., Bassetto, G., Öcal, K., Nonnenmacher, M., and Macke, J. H. Flexible statistical inference for mechanistic models of neural dynamics. *Advances in neural information processing systems*, 30, 2017.

Mancini, A. S., Docherty, M., Price, M., and McEwen, J. Bayesian model comparison for simulation-based inference. *arXiv preprint arXiv:2207.04037*, 2022.

Marlier, N., Brüls, O., and Louppe, G. Simulation-based bayesian inference for multi-fingered robotic grasping. *arXiv preprint arXiv:2109.14275*, 2021.

Martius, G. and Lampert, C. H. Extrapolation and learning equations. *arXiv preprint arXiv:1610.02995*, 2016.

McCormick, D. A. and Huguenard, J. R. A model of the electrophysiological properties of thalamocortical relay neurons. *Journal of neurophysiology*, 68(4):1384–1400, 1992.

Meurer, A., Smith, C. P., Paprocki, M., Čertík, O., Kirpichev, S. B., Rocklin, M., Kumar, A., Ivanov, S., Moore, J. K., Singh, S., et al. Sympy: symbolic computing in python. *PeerJ Computer Science*, 3:e103, 2017.

Oesterle, J., Behrens, C., Schröder, C., Hermann, T., Euler, T., Franke, K., Smith, R. G., Zeck, G., and Berens, P. Bayesian inference for biophysical neuron models enables stimulus optimization for retinal neuroprosthetics. *Elife*, 9:e54997, 2020.

Papamakarios, G. and Murray, I. Fast ε-free inference of simulation models with bayesian conditional density estimation. *Advances in neural information processing systems*, 29, 2016.

Papamakarios, G., Nalisnick, E., Rezende, D. J., Mohamed, S., and Lakshminarayanan, B. Normalizing flows for probabilistic modeling and inference. *Journal of Machine Learning Research*, 22(57):1–64, 2021.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

Podlaski, W. F., Seeholzer, A., Groschner, L. N., Miesenböck, G., Ranjan, R., and Vogels, T. P. Mapping the function of neuronal ion channels in model and experiment. *Elife*, 6:e22152, 2017.

Pospischil, M., Toledo-Rodriguez, M., Monier, C., Piwkowska, Z., Bal, T., Frégnac, Y., Markram, H., and Destexhe, A. Minimal hodgkin–huxley type models for different classes of cortical and thalamic neurons. *Biological cybernetics*, 99:427–441, 2008.

Radev, S. T., Mertens, U. K., Voss, A., Ardizzone, L., and Köthe, U. Bayesflow: Learning complex stochastic models with invertible neural networks. *IEEE transactions on neural networks and learning systems*, 33(4):1452–1466, 2020.

Radev, S. T., D'Alessandro, M., Mertens, U. K., Voss, A., Köthe, U., and Bürkner, P.-C. Amortized bayesian model comparison with evidential deep learning. *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

Ratcliff, R. A theory of memory retrieval. *Psychological review*, 85(2):59, 1978.

Ratcliff, R. and McKoon, G. The diffusion decision model: theory and data for two-choice decision tasks. *Neural computation*, 20(4):873–922, 2008.

Roitman, J. D. and Shadlen, M. N. Response of neurons in the lateral intraparietal area during a combined visual discrimination reaction time task. *Journal of neuroscience*, 22(21):9475–9489, 2002.

Sahoo, S., Lampert, C., and Martius, G. Learning equations for extrapolation and control. In Dy, J. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 4442–4450. PMLR, 10–15 Jul 2018.

Scala, F., Kobak, D., Bernabucci, M., Bernaerts, Y., Cadwell, C. R., Castro, J. R., Hartmanis, L., Jiang, X., Laturnus, S., Miranda, E., et al. Phenotypic variation of transcriptomic cell types in mouse motor cortex. *Nature*, 598(7879): 144–150, 2021.

Schmidt, M. and Lipson, H. Distilling free-form natural laws from experimental data. *Science*, 324(5923):81–85, 2009.

Shinn, M., Lam, N. H., and Murray, J. D. A flexible framework for simulating and fitting generalized drift-diffusion models. *ELife*, 9:e56938, 2020.

Sisson, S. A., Fan, Y., and Beaumont, M. *Handbook of approximate Bayesian computation*. CRC Press, 2018.

Talts, S., Betancourt, M., Simpson, D., Vehtari, A., and Gelman, A. Validating bayesian inference algorithms with simulation-based calibration. *arXiv preprint arXiv:1804.06788*, 2018.

Tejero-Cantero, A., Boelts, J., Deistler, M., Lueckmann, J.-M., Durkan, C., Gonçalves, P. J., Greenberg, D. S., and Macke, J. H. sbi: A toolkit for simulation-based inference. *Journal of Open Source Software*, 5(52):2505, 2020.

Trotta, R. Bayes in the sky: Bayesian inference and model selection in cosmology. *Contemporary Physics*, 49(2): 71–104, 2008.

Turner, B. M., Van Maanen, L., and Forstmann, B. U. Informing cognitive abstractions through neuroimaging: the neural drift diffusion model. *Psychological review*, 122 (2):312, 2015.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Werner, M., Junginger, A., Hennig, P., and Martius, G. Informed equation learning. *arXiv preprint arXiv:2105.06331*, 2021.

Werner, M., Junginger, A., Hennig, P., and Martius, G. Uncertainty in equation learning. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pp. 2298–2305, 2022.

Yadan, O. Hydra - a framework for elegantly configuring complex applications. Github, 2019. URL https://github.com/facebookresearch/hydra.

Zhao, D., Dalmasso, N., Izbicki, R., and Lee, A. B. Diagnostics for conditional density models and bayesian inference algorithms. In *Uncertainty in Artificial Intelligence*, pp. 1830–1840. PMLR, 2021.

Appendix

A1. Software and Computational Ressources

Code is available at `https://github.com/mackelab/simulation_based_model_inference`.

All networks were implemented in *pytorch* (Paszke et al., 2019). Additionally, we used the following software and toolboxes in this work: *sbi* (Tejero-Cantero et al., 2020) for the implementation of SBMI, *NetworkX* (Hagberg et al., 2008) for the construction of prior graphs, *SymPy* (Meurer et al., 2017) for symbolic calculations, *pyDDM* (Shinn et al., 2020) as the backend for the DDM experiments. To manage the configuration settings we used *Hydra* (Yadan, 2019) and the *Optuna Sweeper* (Akiba et al., 2019) plugin for a coarse hyperparameter search in the DDM setting.

All models were trained on an Nvidia RTX 2080ti GPU accessed via a slurm cluster.

A2. Mixture of Grassmann Distribution

Previously, Arai introduced the Grassmann formalism for multivariate binary distributions (Arai, 2021) by using anticommuting numbers, called Grassmann numbers. A Grassmann distribution $\mathcal{G}$ is an n-dimensional binary distribution parameterized by an $n \times n$ matrix Σ. The probability mass function of $\mathcal{G}$ with parameter Σ on $Y = (Y_1, ..., Y_n)$ is defined as

$$\mathcal{G}(y|\Sigma) = \det \begin{pmatrix} \Sigma_{11}^{y_1}(1-\Sigma_{11})^{1-y_1} & \Sigma_{12}(-1)^{1-y_2} & \cdots \\ \Sigma_{21}(-1)^{1-y_1} & \Sigma_{22}^{y_2}(1-\Sigma_{22})^{1-y_2} & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix}.$$

For a valid distribution $\Sigma^{-1} - I$ must be a P_0 matrix, but has otherwise no further constraints (Arai, 2021).

This definition gives access to the analytical derivations of properties such as the mean, covariance, and marginal and conditional distributions. Here, we only recapitulate the analytical formula for conditional distribution, which is used for sampling. Their derivation and further details can be found in (Arai, 2021). In the following paragraph we follow the notation of Arai (Arai, 2021).

For a conditional distribution on $Y = (Y_1, ..., Y_n)$, we denote by C the indices of the observed variables $y_j \in \{0, 1\}$ and R the remaining indices $R = \{1, ..., n\} \setminus C$. Without loss of generality, the parameter matrix can be written as

$$\Sigma = \begin{pmatrix} \Sigma_{RR} & \Sigma_{RC} \\ \Sigma_{CR} & \Sigma_{CC} \end{pmatrix}.$$

The conditional distribution is then given by the Grassmann distribution

$$p(y_R|y_C) = \mathcal{G}(y_R|\Sigma_{R|y_C})$$

with

$$\Sigma_{R|y_C} = \Sigma_{RR} - \Sigma_{RC}\big(\Sigma_{CC} - \text{diag}(1 - y_C)\big)^{-1}\Sigma_{CR},$$

where $\text{diag}(1 - y_C)$ is the diagonal matrix with $(1 - y_C)$ on its diagonal. An analogous formula can be derived by using the notation $\Lambda^{-1} = \Sigma$ (Arai, 2021).

Mixture of Grassmann Distribution We define a mixture of Grassmann distribution (MoGr) on $\{0, 1\}^n$ in the same formalism as $p(y) = \sum_i \alpha_i \mathcal{G}_i(y|\Sigma_i)$ for a finite partition $\sum_i \alpha_i = 1$ and Grassmann distributions $\mathcal{G}_i$. Using the means μ_i and covariances C_i for each component $\mathcal{G}_i$ we can calculate the mean and covariance for the mixture distribution by introducing a discrete latent variable Z and reformulate the mixture distribution as

$$p(y|Z = i) = \mathcal{G}_i(y|\Sigma_i),$$
$$p(Z = i) = \alpha_i.$$

Using the law of total expectation and variance we get analytical expressions for the mean and covariance of a MoGr:

$$\mathbb{E}[Y] = \mathbb{E}[\mathbb{E}[Y|Z = i]] = \sum_i \alpha_i \mu_i$$

and

$$\begin{aligned}\text{Cov}(Y) &= \mathbb{E}[\text{Cov}(Y|Z=i)] + \text{Cov}(\mathbb{E}[Y|Z=i]) \\ &= \sum_i \alpha_i C_i + \sum_i \alpha_i (\mu_i - \bar{\mu})(\mu_i - \bar{\mu})^T,\end{aligned}$$

where $\bar{\mu} = \mathbb{E}[Y]$.

To sample from a MoGr we use the standard procedure of first sampling one component $z_i \sim p(Z)$, and then using the conditional expression of a Grassmann distribution to sample $y \sim \mathcal{G}_{z_i}$.

Implementation Arai (Arai, 2021) proposed the following parametrization for Σ that ensures the P_0 criterion for Σ:

$$\Sigma^{-1} = BC^{-1} + I,$$

where B and C are strictly row diagonal dominant matrices, namely

$$b_{ii} > \sum_{j \neq i} |b_{ij}|, \quad \text{and} \quad c_{ii} > \sum_{j \neq i} |c_{ij}|.$$

We make use of this parametrization by optimizing unconstrained matrices $\tilde{B}$ and $\tilde{C}$ and defining B by replacing the diagonal elements of $\tilde{B}$ by

$$b_{ii} = \exp(\tilde{b}_{ii}) + \sum_{j \neq i} |\tilde{b}_{ij}|,$$

and analogously for C. Instead of $\exp$ any other positive function could be chosen and even the non-negative ReLU function showed good training behaviour in initial experiments.

We used a similar parameterization for a mixture of Grassmann distribution for each component and a softmax layer to learn the partition $\sum_i \alpha_i = 1$.

A3. Model Prior

Sampling from the prior: A draw from the prior corresponds to a random walk on the graph $(\mathcal{M}, E, \mathcal{R})$, starting at node M_0 and walking through the graph in the following way (see Algorithm 1 for pseudocode): For each step, we first normalize the outgoing edges for the current node M_c. We use these weights to sample the next node M_{c+1} from a categorical distribution on all connected nodes and append it to the set of sampled nodes S. Next, we update all weights following the updating rules $\mathcal{R}$. We then start the next step by normalizing the outgoing edges of the updated current node and repeat this procedure until the end node M_{N+1} is reached.

Algorithm 1: Sampling procedure for model prior

Inputs: Directed graph with dynamically changing weights: $(\mathcal{M}, \mathcal{E}, \mathcal{R})$,
with nodes $\mathcal{M} = \{M_i\}_{i \in 0,...,N+1}$, edges $\mathcal{E} = \{e_{ij} = (M_i, M_j, w_{ij}) | M_i, M_j \in \mathcal{M}, w_{ij} \in \mathbb{R}_{\geq 0}\}$, and updating rules $\mathcal{R} = \{R_k | R_k : (\mathcal{E}, S) \to \mathcal{E},\ S \subset \mathcal{M}\}_{k \in K}$
Outputs: Prior sample $\tilde{M} \sim p(M)$.

$S \leftarrow \{M_0\}$; `# initialize set of sampled nodes`
$M_c \leftarrow M_0$; `# initialize current node`
while $M_{N+1} \notin S$ **do**
  $\tilde{w}_{ci} \leftarrow w_{ci} / \sum_i w_{ci}$; `# normalize outgoing edges`
  $q_c \leftarrow \text{Cat}(\tilde{w}_{ci}, \{M_i | \tilde{w}_{ci} > 0\})$; `# define categorical distribution`
  $M_{c+1} \sim q_c$; `# sample next node`
  $S \leftarrow S \cup \{M_{c+1}\}$;
  for $R_k \in \mathcal{R}$ **do**
    $\mathcal{E} \leftarrow R_k(\mathcal{E}, S)$; `# update weights`
  $M_c \leftarrow M_{c+1}$;
$\tilde{M} = [1 \text{ if } M_i \in S \text{ else } 0, \text{ for } i \in 1, ..., N]$; `# convert to binary vector of dimension N`
return $\tilde{M}$

Example rules $\mathcal{R}$:

1. Example rule R_x^1 to ensure a conditionally acyclic graph: For a sampled M_x we set all ingoing edges of M_x to zero:
$$R_x^1 = \{\text{if } M_x \in S : w_{ix} = 0 \; \forall \; i = 0, ..., N+1\}$$

2. Example rules R_{xy}^2 and R_{yx}^2 to avoid the co-occurrence of specific components M_x and M_y: We set all ingoing edges to M_y to zero if M_x was already sampled and the other way round:
$$R_{xy}^2 = \{\text{if } M_x \in S : w_{iy} = 0 \; \forall \; i = 0, ..., N+1\}, \text{ and}$$
$$R_{yx}^2 = \{\text{if } M_y \in S : w_{ix} = 0 \; \forall \; i = 0, ..., N+1\}.$$

3. Example rules R_{xy}^3 and R_{yx}^3 to decrease the co-occurrence of specific components M_x and M_y by decreasing the weight of all ingoing edges to M_y by a constant factor c if M_x was already sampled and the other way round:
$$R_{xy}^3 = \{\text{if } M_x \in S : w_{iy} = cw_{iy} \; \forall \; i = 0, ..., N+1 \text{ and } 0 < c < 1\}, \text{ and}$$
$$R_{yx}^3 = \{\text{if } M_y \in S : w_{ix} = cw_{ix} \; \forall \; i = 0, ..., N+1 \text{ and } 0 < c < 1\}.$$

4. Example rule $R_{x\text{end}}^4$ to favor simpler models. For a sampled M_x with no direct vertex to the end node, we increase the weights vertices which are directly connected to the end node by a constant factor c:
$$R_{x\text{end}}^4 = \{\text{if } M_x \in S \text{ and } w_{x\text{end}} = 0 : w_{i\text{end}} = cw_{i\text{end}} \; \forall \; i = 0, ..., N+1 \text{ and } c > 1\}.$$

A4. Inference

Algorithm 2: Simulation-base model inference: SBMI

Inputs: Model prior $p(M)$, parameter priors $p(\theta|M)$, compiler C, number of simulations L, embedding net $e_\zeta(x)$, model posterior network $q_\psi(M|e)$, parameter posterior network $q_\phi(\theta|M, e)$.
Outputs: Trained embedding network $e_\zeta(x)$, model posterior network $q_\psi(M|x)$, parameter posterior network $q_\phi(\theta|M, x)$.
Generate dataset:
for $l = 1, ..., L$ **do**
  $M_l \sim p(M)$; `# sample model`
  $\theta_l \sim p(\theta|M_l)$; `# sample parameters`
  $S_l \leftarrow C(M_l, \theta_l)$; `# compile simulator`
  $x_l \sim S_l$; `# simulate data`
return $\{(M_l, \theta_l, x_l)\}_{l=1,...,L}$
Training: ; `# We omit the use of training batches here.`
while *not converged* **do**
  $\mathcal{L}^M \leftarrow -\frac{1}{L}\sum_l \log\, q_\psi(M_l|e_\zeta(x_l))$; `# compute model loss`
  $\mathcal{L}^\theta \leftarrow -\frac{1}{L}\sum_l \log\, q_\phi(\theta_l|M_l, e_\zeta(x_l))$; `# compute parameter loss`
  $(\zeta, \psi, \phi) \leftarrow (\zeta, \psi, \phi) - \text{Adam}(\nabla_{(\zeta,\psi,\phi)}(\mathcal{L}^M + \mathcal{L}^\theta))$; `# take gradient step`
return $e_\zeta(x)$, $q_\psi(M|x)$, $q_\phi(\theta|M, x)$

A4.1. Model Posterior Network

We used a conditional MoGr distribution as model posterior network. The conditional parameters $\Sigma_i|x$ are parameterized by two matrices B_i and C_i (Section A2). We used a fully connected neural network with ReLU activation to parametrize the unconstrained matrices $\tilde{B}_i$, $\tilde{C}_i$ and a softmax layer for the partition α with $\sum_i \alpha_i = 1$. The input to the MoGr network is the output $e(x)$ of the embedding net and the used hyperparameters can be found in Table S5 and S8.

A4.2. Parameter Posterior Network

For the paremeter posterior network, we used a conditioned mixture of (Gaussian) density network, which allowed us to marginalize analytically over the parameters of the absent model components during training time. For efficient training, we divided each batch into sub-batches with the same number of parameters and processed each sub-batch in parallel.

The conditioning network was implemented as fully connected network with ReLU activation. The specifics for the different settings can be found in Table S5, S6, S8, and S13.

A4.3. Computational Efficiency

The parameter posterior network shares the computational complexity with standard MDNs, as marginalization of the MDN is performed analytically. The evaluation of the MoGr distribution involves computing a determinant of a $n \times n$ matrix, which in general has the complexity of $O(n^3)$. However, n is the number of model components, which is relatively small in the presented work (up to 20 for our experiments).

A4.4. Training

We used the standard training loop of the *sbi* toolbox (Tejero-Cantero et al., 2020): as validation set, we used 10% of the training samples and as stopping criterion we defined 25 consecutive epochs of no improvement of the loss function on the validation set.

For the additive model we used a batch size of 3000 samples, for the DDM a batchsize of 2000 samples, and for the Hodgkin-Huxley model of 4000 samples.

A5. Performance Measures

A5.1. MAP Estimate

Once we trained the full network, we can easily get a *maximum a posteriori* estimate (MAP) by searching the discrete model space:

$$\begin{aligned}\max_{M,\theta} p(M,\theta|x_o) &= \max_{M,\theta} p(M|x_o)\cdot p(\theta|M,x_o)\\ &= \max_{i\in I}\{p(M_i|x_o)\cdot \max_{\theta} p(\theta|M_i,x_o)\}.\end{aligned}$$

While mathematically correct, this MAP is often dominated by the density function of the parameter posterior, which can take arbitrarily large values for small variances and can be susceptible to noise in the training process. The discrete distribution $p(M|x_o)$ is, however, bounded in $[0,1]$. Therefore, we are often interested in the more stable MAP parameter estimates of the most likely model:

$$\theta_{map} = \operatorname*{argmax}_{\theta} p(\theta|M_{map},x_o),$$

where $M_{map} = \operatorname*{argmax}_{M_i,i\in I} p(M_i|x_o)$.

This MAP of the most likely model is shown as f_{MAP} in Figure 3d.

A5.2. Additive Model

For the additive model we can approximate the ground truth model posterior $p(M|x_o)$ by calculating the model evidence by

$$p(M|x_o) = \frac{p(x_o|M)p(M)}{p(x_o)} \sim p(x_o|M)p(M).$$

The prior $p(M)$ is only given implicitly, but as the model space is low-dimensionial, we can approximate the prior by the empirical sampling distribution $\hat{p}(M)$ (shown in Fig. 3). We therefore get the approximation

$$\begin{aligned}p(M|x_o) &\sim p(M)\int p(x_o|M,\theta)p(\theta|M)d\theta\\ &\approx \hat{p}(M)\frac{1}{N}\sum_{j=1}^{N} p(x_o|M,\theta_j),\end{aligned}$$

where θ_j are samples from the parameter prior $p(\theta|M)$. Since we used a Gaussian noise model, we can calculate the expression $p(x_o|M,\theta_j)$ by evaluating $\mathcal{N}(f_{\theta_j}(t),\Sigma_{\theta_j}(t))$.

In practice, we apply importance sampling to avoid regions with a low probability, such that we get

$$p(M|x_o) \sim \hat{p}(M) \frac{1}{N} \sum_{j=1}^{N} p(x_o|M, \theta_j) \frac{p(\theta_j|M)}{q_\phi(\theta_j|M, x_o)},$$

where $\theta_j \sim q_\phi(\theta|M, x_o)$ are samples from the approximated parameter posterior.

Even with importance sampling, a lot of samples were necessary to get reliable estimates. We used 100k samples $\theta_j \sim q_\phi(\theta|M, x_o)$ per observation and were therefore restricted to few observations x_o (100 for the presented results in Section 3.1).

A5.3. DDM

We used the mean-squared error (MSE) on the weighted density functions of the two different decisions, similar to (Shinn et al., 2020). In the same work, they showed that the relative MSE is in good correspondence with other performance metrics on the used experimental data. We therefore used the loss function implemented as `LossSquaredError` in the *pyDDM* package (Shinn et al., 2020).

A6. SBMI Loss and Kullback-Leibler Divergence

Proposition A6.1. *Optimizing the SBMI loss function* $\mathcal{L}(\psi, \phi) = -\frac{1}{L} \sum_l \mathcal{L}_{M_l}(\psi) + \mathcal{L}_{\theta_l}(\phi)$ *minimizes the expected Kullback-Leibler divergence between the true joint posterior* $p(M, \theta|x)$ *and the approximation* $q_\phi(M|x)q_\psi(\theta|M, x)$*:*

$$\mathbb{E}_{p(x)}\big[D_{KL}(p(M, \theta|x)||q_\phi(M|x)q_\psi(\theta|M, x)\big].$$

Proof.

$$\begin{aligned}
&\mathbb{E}_{p(x)}[D_{KL}\left(p(M, \theta|x)||q_\phi(M|x)q_\psi(\theta|M, x)\right)] \\
&= \mathbb{E}_{p(x)}\left[\mathbb{E}_{p(M,\theta|x)}\left[\log \frac{p(M, \theta|x)}{q_\phi(M|x)q_\psi(\theta|M, x)}\right]\right] \\
&= \mathbb{E}_{p(x,M,\theta)}\left[-\log q_\phi(M|x) - q_\psi(\theta|M, x)\right] + C \\
&\approx \frac{1}{L} \sum_l \big(-\log q_\phi(M|x) - q_\psi(\theta|M, x)\big) + C \\
&= -\frac{1}{L} \sum_l \mathcal{L}_{M_l}(\psi) + \mathcal{L}_{\theta_l}(\phi) + C \\
&= \mathcal{L}(\psi, \phi) + C
\end{aligned}$$

where C is a constant independent of ϕ and ψ. □

A7. Model Details

A7.1. Additive Model

Prior To show the flexibility of the presented prior over model components, we defined a dynamically changing graph for the additive model. During a random walk, we increased the edge weights of the direct model paths to the end node with every sampled component and additionally decreased the weight between the linear components if one component is sampled by a factor of two. This favors simple models and disadvantages the co-occurrence of both linear components. This corresponds to rules R^3_{xy}, R^3_{yx} and $R^4_{x\text{end}}$ in Appendix A3 with $c = 0.5$ and $c = 2$ respectively. The resulting empirical prior distribution is shown in grey in Figure 3b.

The parameter priors for the model components are shown in Table S1.

Network Details We used a one-dimensional convolutional network followed by fully connected layers as an embedding net for the additive model. The convolutional layers used a kernel size of five and stride one. The output of the last convolutional layer was flattened before passing it on to the fully connected network. All further parameters can be found in Table S5 and S6.

Table S1. **Details for the additive model with six components.** The parameter θ_1 in the noise terms n_1 and n_2 defines the standard deviation of a normal distribution $\mathcal{N}$, and $\mathcal{U}(a, b)$ defines a uniform distribution on the interval $[a, b]$. Overall the model has seven parameters. For the performance we report the mean and standard deviation.

Model Component	Token	Parameter Prior	Performance $\hat{p}_{\text{reference}}(M_i\|x_o)$	Performance $q_\psi(M_i\|x_o)$
$\theta_1 \cdot t$	l_1	$\theta_1 \sim \mathcal{U}(-2, 2)$	0.70 (0.27)	0.65 (0.24)
$\theta_1 \cdot t$	l_2	$\theta_1 \sim \mathcal{U}(-2, 2)$	0.70 (0.26)	0.67 (0.24)
$\theta_1 \cdot t^2$	q	$\theta_1 \sim \mathcal{U}(-0.5, 0.5)$	0.97 (0.09)	0.93 (0.15)
$\theta_1 \cdot \sin(\theta_2 t)$	sin	$\theta_1 \sim \mathcal{U}(0, 5)$ $\theta_2 \sim \mathcal{U}(0.5, 5)$	0.95 (0.15)	0.91 (0.18)
noise$_1$: $n_{t_i} \sim \mathcal{N}(0, \theta_1)$	n_1	$\theta_1 \sim \mathcal{U}(0.1, 2)$	1.00 (0.00)	1.00 (0.00)
noise$_2$: $n_{t_i} \sim (t_i + 1)\mathcal{N}(0, \theta_1)$	n_2	$\theta_1 \sim \mathcal{U}(0.5, 2)$	1.00 (0.00)	1.00 (0.00)

Table S2. **Details for the additive model with eleven components.** The noise terms are the same as in Table S1. Overall the model has 13 paramters. For the performance we report the mean and mean of the standard deviation on training with 1M datapoints across 5 optimization runs.

Model Component	Token	Parameter Prior	Performance $q_\psi(M_i\|x_o)$
$\theta_1 \cdot t$	l_1	$\theta_1 \sim \mathcal{U}(-2, 2)$	0.68 (0.24)
$\theta_1 \cdot t$	l_2	$\theta_1 \sim \mathcal{U}(-2, 2)$	0.69 (0.24)
$\theta_1 \cdot t^2$	q_1	$\theta_1 \sim \mathcal{U}(-0.5, 0.5)$	0.80 (0.24)
$(\theta_1 + t)^2$	q_2	$\theta_1 \sim \mathcal{U}(-5, 0)$	0.98 (0.10)
$\theta_1 \cdot t^3$	cub	$\theta_1 \sim \mathcal{U}(-0.1, 0.1)$	0.88 (0.21)
$\theta_1 \cdot \sin(\theta_2 t)$	sin	$\theta_1 \sim \mathcal{U}(0, 5)$ $\theta_2 \sim \mathcal{U}(0.5, 5)$	0.86 (0.23)
$\theta_1 \cdot \cos(\theta_2 t)$	cos	$\theta_1 \sim \mathcal{U}(0, 5)$ $\theta_2 \sim \mathcal{U}(0.5, 5)$	0.89 (0.21)
θ_1	$const_1$	$\theta_1 \sim \mathcal{U}(-5, 5)$	0.72 (0.25)
θ_1	$const_2$	$\theta_1 \sim \mathcal{U}(0, 10)$	0.80 (0.25)
noise$_1$: $n_{t_i} \sim \mathcal{N}(0, \theta_1)$	n_1	$\theta_1 \sim \mathcal{U}(0.1, 2)$	1.00 (0.02)
noise$_2$: $n_{t_i} \sim (t_i + 1)\mathcal{N}(0, \theta_1)$	n_2	$\theta_1 \sim \mathcal{U}(0.5, 2)$	1.00 (0.02)

A7.2. Bayes Factors for the shown Example

For the example shown in Fig. 3 we get Bayes factors of $B_{l_1 l_2} = 1.02$ for the comparison of the two models with a single linear component (either l_1 or l_2), and $B_{l_1 l_{12}} = 1.45$, if we compare the model with component l_1 with the one in which both model components are present (l_{12}). Following the scale by (Jeffreys, 1998) this would be 'inconclusive' about the preference of the models.

Table S3. **SBMI performance for the additive model for 500k training samples.** Comparison of SBMI and reference model posteriors in terms of Kullback-Leibler divergence (KL) and marginal performances. We calculated reference posteriors for 100 observations x_o (see Table S1 for performances of individual components). For the RMSE we used 1k observations x_o and 'Reference' corresponds to the RMSE between the observations x_o and samples x under the ground truth model and parameters. We report mean and standard deviation.

Measure	Reference (Posterior)	SBMI Posterior	Prior
KL	-	0.28 (0.71)	11.26 (1.88)
Marginal Performance	0.88 (0.15)	0.86 (0.09)	0.53 (0.12)
RMSE	6.87 (6.05)	7.05 (6.19)	15.24 (7.95)

Table S4. **Details for the additive model with 20 components.** The model prior is a fully connected graph for the additive components. The noise components are mutually exclusive as in the smaller models. Overall the model has 28 parameters.

Model Component	Token	Parameter Prior
$\theta_1 \cdot t$	l_1	$\theta_1 \sim \mathcal{U}(-2, 2)$
$\theta_1 \cdot t$	l_2	$\theta_1 \sim \mathcal{U}(-2, 2)$
$\theta_1 \cdot t^2$	q_1	$\theta_1 \sim \mathcal{U}(-0.5, 0.5)$
$(\theta_1 + t)^2$	q_2	$\theta_1 \sim \mathcal{U}(-5, 0)$
$\theta_1 \cdot t^3$	cub	$\theta_1 \sim \mathcal{U}(-0.1, 0.1)$
$\theta_1 \cdot \sin(\theta_2 t)$	sin	$\theta_1 \sim \mathcal{U}(0, 5)$ $\theta_2 \sim \mathcal{U}(0.5, 5)$
$\theta_1 \cdot \cos(\theta_2 t)$	cos	$\theta_1 \sim \mathcal{U}(0, 5)$ $\theta_2 \sim \mathcal{U}(0.5, 5)$
θ_1	$const_1$	$\theta_1 \sim \mathcal{U}(-5, 5)$
θ_1	$const_2$	$\theta_1 \sim \mathcal{U}(0, 10)$
$\theta_1 \cdot \tanh(t - \theta_2)$	$tanh_1$	$\theta_1 \sim \mathcal{U}(1, 10)$ $\theta_2 \sim \mathcal{U}(2, 8)$
$\theta_1 \cdot \tanh(\theta_2 - t)$	$tanh_2$	$\theta_1 \sim \mathcal{U}(1, 10)$ $\theta_2 \sim \mathcal{U}(2, 8)$
$\theta_1 \cdot \exp(-(t - \theta_2)^2)$	g_1	$\theta_1 \sim \mathcal{U}(1, 10)$ $\theta_2 \sim \mathcal{U}(2, 8)$
$\theta_1 \cdot \exp(-(t - \theta_2)^2/8$	g_2	$\theta_1 \sim \mathcal{U}(1, 10)$ $\theta_2 \sim \mathcal{U}(2, 8)$
$\theta_1 \cdot \text{ReLU}(t - \theta_2)$	$relu_1$	$\theta_1 \sim \mathcal{U}(1, 5)$ $\theta_2 \sim \mathcal{U}(2, 8)$
$\theta_1 \cdot \text{ReLU}(\theta_2 - t)$	$relu_2$	$\theta_1 \sim \mathcal{U}(1, 5)$ $\theta_2 \sim \mathcal{U}(2, 8)$
noise$_1$: $n_{t_i} \sim \mathcal{N}(0, \theta_1)$	n_1	$\theta_1 \sim \mathcal{U}(0.1, 2)$
noise$_2$: $n_{t_i} \sim (t_i + 1)\mathcal{N}(0, \theta_1)$	n_2	$\theta_1 \sim \mathcal{U}(0.5, 2)$
noise$_3$: $n_{t_i} \sim (11 - t_i)\mathcal{N}(0, \theta_1)$	n_3	$\theta_1 \sim \mathcal{U}(0.5, 2)$
noise$_4$: $n_{t_i} \sim (t_i^2 + 1)\mathcal{N}(0, \theta_1)$	n_4	$\theta_1 \sim \mathcal{U}(0.2, 1)$
noise$_5$: $n_{t_i} \sim (11 - t_i^2)\mathcal{N}(0, \theta_1)$	n_5	$\theta_1 \sim \mathcal{U}(0.2, 1)$

Table S5. **Network details for the additive model.** Square brackets indicate the layer-wise parameters, otherwise the same parameters were used for all layers.

	Number of Layers	Dimensions / #Channels	Components
Convolutional layers	2	[10, 16]	-
Fully connected layers	3	[200, 200, 50]	-
MoGr net	3	80	3
MDN net	3	120	3

Table S6. **Network details for the large additive model.** Square brackets indicate the layer-wise parameters, otherwise the same parameters were used for all layers.

	Number of Layers	Dimensions / #Channels	Components
Convolutional layers	2	[10, 16]	-
Fully connected layers	3	[200, 200, 50]	-
MoGr net	3	120	3
MDN net	3	200	3

A7.3. DDM

Drift diffusion models can be described as a stochastic differential equation for a decision variable z:

$$\mathrm{d}z = d(z,t)\mathrm{d}t + \mathrm{d}W,$$

with initial condition z_0, drift term d, and a Wiener noise process W. A decision is taken when the decision variable hits the boundary $|d(z,t)| \geq b(t)$ (Figure 5a). An additional parameter delays the starting time of the process ('non-decision time').

We included two different drift terms d:

1. constant drift: $d(z,t) = \theta_1$, and
2. leaky drift: $d(z,t) = \theta_1 + \theta_2 \cdot z$ (with $\theta_2 < 0$),

and two boundary conditions b:

1. constant boundary: $b(t) = \theta_1$, and
2. exponentially collapsing boundary: $b(t) = \theta_1 - \exp(-t/\theta_2)$.

The initial condition z_0 was fixed to be zero and the noise term had a constant standard deviation of one. The non-decision time was a free parameter but was present in all models (see Figure 5b).

Table S7. **Details for the DDM.** We used independent uniform distributions $\mathcal{U}$ for all parameter priors. The performances were calculated on 1k samples from the prior distribution, and we report mean and standard deviation.

Model Component	Token	Parameter Prior	Performance Model Posterior	Performance Model MAP
constant drift	d_c	$\theta_1 \sim \mathcal{U}(0,5)$	0.85 (0.23)	0.90 (0.31)
leaky drift	d_l	$\theta_1 \sim \mathcal{U}(0,5)$ $\theta_2 \sim \mathcal{U}(-20,-5)$	0.85 (0.23)	0.90 (0.31)
constant boundary	b_c	$\theta_1 \sim \mathcal{U}(0.3,2)$	0.90 (0.20)	0.92 (0.27)
exp. collapsing boundary	b_{exp}	$\theta_1 \sim \mathcal{U}(0.3,2)$ $\theta_2 \sim \mathcal{U}(0.5,1.5)$	0.90 (0.20)	0.92 (0.27)
non-decision time	ndt	$\theta_1 \sim \mathcal{U}(0.1,0.3)$	1.00 (0.00)	1.00 (0.00)

Training Data We used the *pyDDM* toolbox (Shinn et al., 2020) to solve the DDM numerically for every θ using the Fokker-Planck equation. From the approximated decision time and choice distribution we then sampled 400 iid trials for each θ. This results in a 400×2 data matrix with the recorded continuous decision times and binary decisions.

As training data, we sampled 200k models from the prior, solved these DDMs and drew 400 trials. From this data, we excluded datapoints with more than 300 undecided trials (defined as trials with a decision time larger than 10 seconds). From the remaining ≈180k datapoints we hold back 1k test datapoints and divided the other part into 10% validation and 90% training data.

Prior Initial experiments showed that models with leaky drift and constant boundary conditions often resulted in unrealistically long decision times (>10sec), and we therefore discouraged their co-occurrence by including a negative coupling between these two terms in the model prior.

All edges of the model prior have initially the same weight in the shown prior graph (Figure 5b). If the leaky drift component is visited in a random walk, the edge weight of the constant boundary condition is decreased by a factor of two (corresponding to R^2_{xy}).

The parameter priors for the different model components are shown in Table S7

Network Details To account for the iid trial structure of the DDM data, we used a permutation invariant embedding net (Chan et al., 2018; Radev et al., 2020). In this setup, the single-trial data is first processed by a fully connected network, mean pooled, and then passed through additional fully connected layers.

Each trial (represented as a vector (decision time, decision)) is first processed by the 'single trial net', which we implemented as a fully connected neural network. The output is then averaged (making it permutation invariant) and passed on to a second fully connected network. The used hyperparameters (Table S8) were the best hyperparameters in a coarse hyperparameter sweep over eight models, varying three hyperparameters. To this end, we used *Optuna* (Akiba et al., 2019) and varied the embedding dimensions (last layer of the single trial net and the last layer of the fully connected embedding net) and the dimension of the MoGr net.

Table S8. **Network details for the DDM.** Square brackets indicate the layer-wise parameters, otherwise the same parameters were used for all layers.

	Number of Layers	Dimensions	Components
Single trial net	3	[120, 120, 100]	-
Fully connected embedding net	3	[120, 120, 30]	-
MoGr net	3	80	3
MDN net	3	120	3

Dataset The used data (Roitman & Shadlen, 2002) was collected from two monkeys performing a random dot motion discrimination task. Visual stimuli of moving dots with different coherence rates (0, 3.2, 6.4 and 12.8%) were presented and the monkeys had to decide on the moving direction. We randomly subsampled 400 trials for each stimulus condition to match the dimension of our training data and show the results for 'monkey N' throughout the manuscript. The dataset can be found here: `https://shadlenlab.columbia.edu/resources/RoitmanDataCode.html`.

Table S9. **DDM parameter comparison for example observation.** Sample mean and standard deviation for 10k samples from the SBMI parameter posterior for the example observation from Figure 5. The model posteriors are $q_\psi(\text{gt-model}|x_o) = 0.75$ and $q_\psi(\text{c.-drift-model}|x_o) = 0.25$.

Model Component	Parameter	Ground Truth	SBMI posterior | gt-model	SBMI posterior | c.-drift-model
constant drift	θ_1	-	-	1.37 (0.08)
leaky drift	θ_1	2.00	1.79 (0.17)	-
	θ_2	-10.00	-9.71 (3.60)	-
constant boundary	θ_1	-	-	-
exp. collapsing boundary	θ_1	0.70	0.75 (0.13)	1.73 (0.15)
	θ_2	0.70	0.76 (0.11)	1.07 (0.11)
non-decision time	θ_1	0.25	0.22 (0.03)	0.14 (0.02)

Table S10. **DDM predictive performance for experimental data.** Comparison of mean decision times μ and standard deviation of decision times σ of the experimental data ($\hat{\cdot}$) to posterior predictive samples. We report the mean and standard deviation for the different measures based on 10k SBMI posterior samples and for ten *pyDDM* fits with different random seeds. The statistics are pooled over the different coherence rates.

Measure	*pyDDM*	SBMI
decision time: $\lvert\mu - \hat{\mu}\rvert$	0.06 (0.06)	0.06 (0.06)
decision time: $\lvert\sigma - \hat{\sigma}\rvert$	0.17 (0.15)	0.13 (0.15)
deviation correct trials in %	2.08 (1.75)	2.22 (1.83)
MSE on densities ($\cdot 10^{-2}$)	9.66 (9.18)	9.66 (8.94)

Table S11. **DDM predictive performance for experimental data for individual coherence rates.** We report the mean and standard deviation for the MSE based on 10k SBMI posterior samples and for ten *pyDDM* fits with different random seeds. See Table S1 (main paper) for the pooled statistics.

Measure	Coherence %	*pyDDM*	SBMI
MSE on densities ($\cdot 10^{-2}$)	0.0	23.80 (0.004)	23.79 (0.010)
	3.2	10.66 (0.002)	10.67 (0.009)
	6.4	3.64 (0.001)	3.64 (0.008)
	12.8	0.55 (0.001)	0.56 (0.011)

A7.4. Hodgkin-Huxley model

We implemented a version of the Hodgkin-Huxley model based on (Pospischil et al., 2008) in *JAX* (Bradbury et al., 2018).

The differential equations are given by

$$\begin{aligned}\frac{dV}{dt} =& g_L(E_L - V) + g_{Na}m^3h(E_{Na} - V) + g_K n^4(E_K - V) + g_M p(E_K - V) + g_{Ca}q^2r(E_{Ca} - V) \\ &+ I_{inj} + \sigma\eta(t),\end{aligned}$$

and

$$\frac{ds}{dt} = \frac{s_\infty(V) - s}{\tau_s(V)}; s \in \{m, h, n, p, q, r\},$$

where I_{inj} corresponds to the injected current to stimulate the cell. The parameters V_t for K and τ for K_m define the steady state s_∞ and the time constant τ_s for the corresponding gating parameters. All details can be found in (Pospischil et al., 2008).

For the noise term $\sigma\eta(t)$ we sampled for each time step independent noise and scaled it corresponding to the standard deviation $\theta_1 = \sigma$. We used a step current I_{inj} of $2\mu A/cm^2$ for $1000ms$ and run the simulation for $1450ms$. This stimulus and recording protocol corresponds to the voltage recordings from the Allen Cell database.

Prior The graph for the model prior is shown in Fig. S8, and all edge weights were set to one, except the edge from K to Na which was set to $1/3$. The parameter priors for the different model components are shown in Table S12 and adapted from (Gonçalves et al., 2020).

Training data We sampled and simulated 100k models. We excluded simulations with *Nan* and *inf* values, resulting in a training dataset of 99,895 simulations. We used these simulations to calculate 24 summary statistics and imputed *Nan* values either by the appropriate *max*, *min* or *mean* value. The used summary statistics were adapted from previously used summary statistics in (Scala et al., 2021).

Network details The network details can be found in Table S13.

Table S12. **Details for the Hodgkin-Huxley model.** The parameter θ_1 in the noise terms defines the standard deviation of a normal distribution $\mathcal{N}$, and $\mathcal{U}(a, b)$ defines a uniform distribution on the interval $[a, b]$. For the performance we report the mean and standard deviation over all test samples x_o as well as over test samples which have at least one spike x_o^s.

Model Component	Token	Parameter Prior	Performance (all) $q_\psi(M_i\|x_o)$	Performance (spikes) $q_\psi(M_i\|x_o^s)$
Leak current	I_L	$g_L \sim \mathcal{U}(10^{-6}, 3 \cdot 10^{-4})$ $E_L \sim \mathcal{U}(-80, -60)$	1.00 (0.00)	1.00 (0.00)
Potassium channel	K	$g_K \sim \mathcal{U}(1.5 \cdot 10^{-3}, 1.5 \cdot 10^{-2})$ $V_t \sim \mathcal{U}(-70, -50)$	1.00 (0.00)	1.00 (0.00)
M-type potassium channel	K_m	$g_M \sim \mathcal{U}(10^{-5}, 6 \cdot 10^{-4})$ $\tau \sim \mathcal{U}(200, 2000)$	0.96 (0.12)	0.98 (0.01)
Sodium channel	Na	$g_{Na} \sim \mathcal{U}(8 \cdot 10^{-3}, 8 \cdot 10^{-2})$	1.00 (0.00)	1.00 (0.00)
Calcium channel	Ca	$g_{Ca} \sim \mathcal{U}(5 \cdot 10^{-5}, 10^{-3})$	0.74 (0.25)	0.94 (0.18)
Noise	$Noise$	$\theta_1 \sim \mathcal{U}(10^{-4}, 1.5 \cdot 10^{-1})$	1.00 (0.00)	1.00 (0.00)

Table S13. **Network details for the Hodgkin-Huxley model.** Note that we replaced the embedding net by summary statistics in this case.

	Number of Layers	Dimensions	Components
MoGr net	3	80	3
MDN net	3	150	3

A8. Supplementary Figures

a

Model Component	Performance $\hat{p}_{\text{reference}}(M_i\|x_o)$	Performance $q_\psi(M_i\|x_o)$ (MADE)
$\theta_1 \cdot t$	0.70 (0.27)	0.65 (0.20)
$\theta_1 \cdot t$	0.70 (0.26)	0.67 (0.23)
$\theta_1 \cdot t^2$	0.97 (0.09)	0.82 (0.11)
$\theta_1 \cdot \sin(\theta_2 t)$	0.95 (0.15)	0.83 (0.14)
noise_1	1.00 (0.00)	0.93 (0.08)
noise_2	1.00 (0.00)	0.93 (0.08)

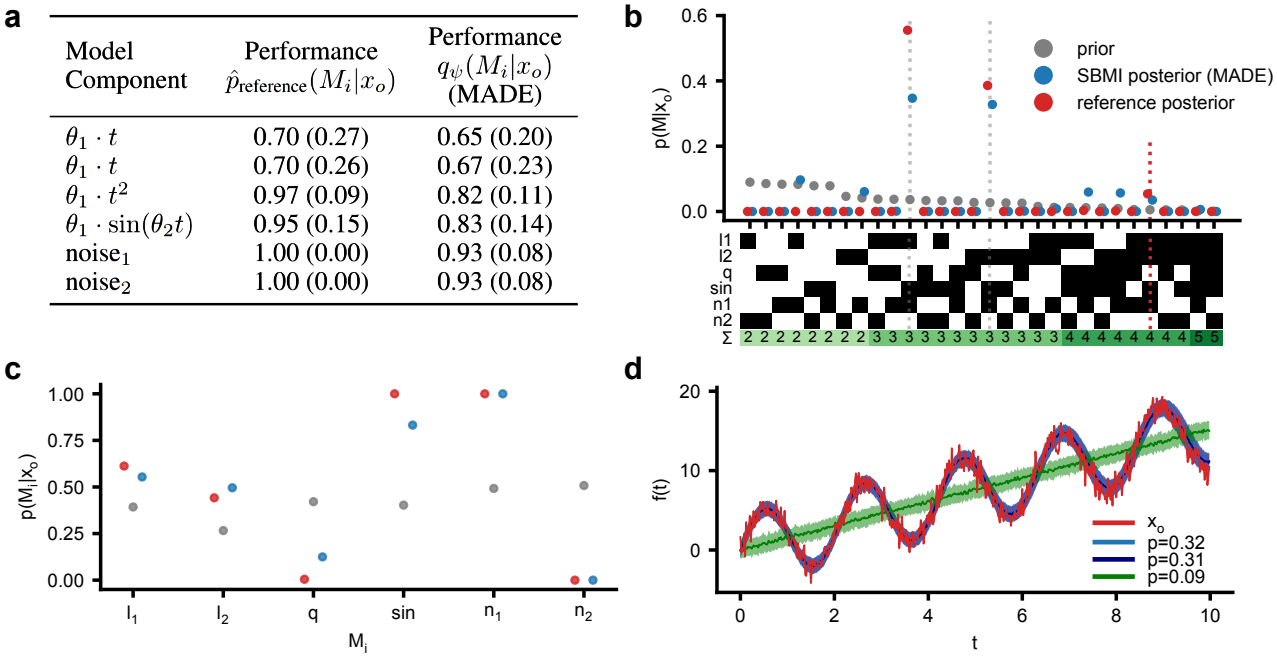

Figure S1. **SBMI on the Additive Model** (using MADE instead of MoGr). **(a)** Table with marginal posterior performance for the different model components of the additive model. Initial experiments showed a worse performance of the posterior implemented as MADE compared to a MoGr (Table S1). **(b)** Model posterior implemented as MADE conditioned on x_o shown in (d) (similar to Fig. 3b). **(c)** Marginal model posterior implemented as MADE for the same observation. **(d)** Posterior predictives (and local uncertainties as mean $\pm$ std.) of the three most likely models, covering 72% of the model posterior mass. Compared to Fig. 3d models without the sinusoidal get non-negligible posterior mass.

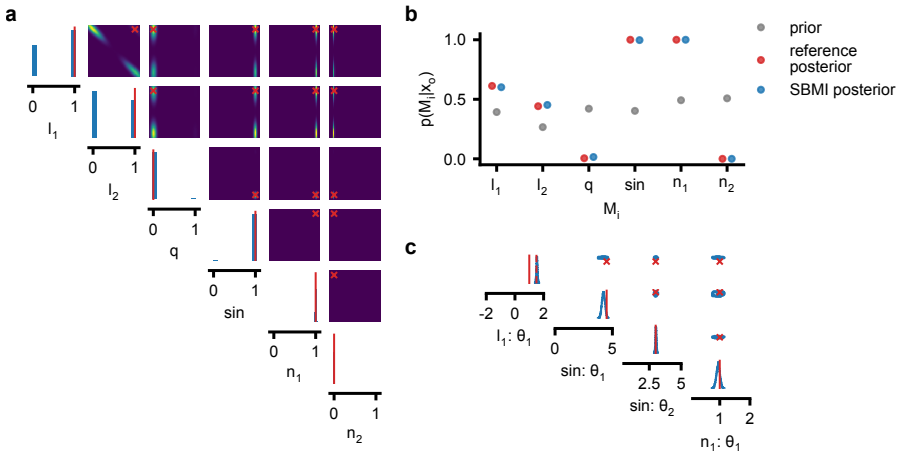

Figure S2. **SBMI posterior for the additive model. (a)** Smoothed one- and two-dimensional marginal distribution for the (binary) model posterior. The ground truth model is indicated in red. **(b)** Marginal model posterior distribution for the observation x_o shown in Figure 3. The ground truth model consists of the components l_1, l_2, sin, and n_1. **(c)** One- and two-dimensional marginal distribution of the SBMI parameter posterior given the MAP model. The ground truth parameters are indicated in red. The sum of the coefficients θ_1 for the two linear components l_1 and l_2 of the ground truth model is indicated as a dashed line in the most left plot. It matches the mean of the posterior marginal for l_1.

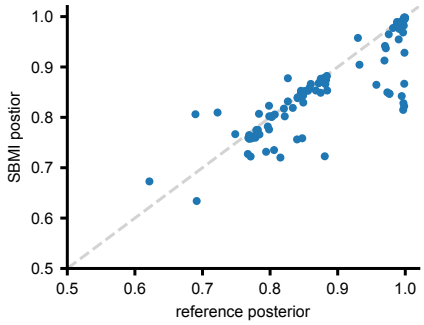

Figure S3. **Model posterior for the additive model.** Mean marginal performance of the reference posterior vs the SBMI posterior for 100 examples of the additive model with a correlation of 0.85. Note that uncertainty in the model posterior is reflected by lower 'performance' values which is not necessarily a bad sign as it might reflect the true underlying uncertainty. The high correlation indicates that the uncertainty of both posteriors is similar for most examples, which validates our SBMI approach.

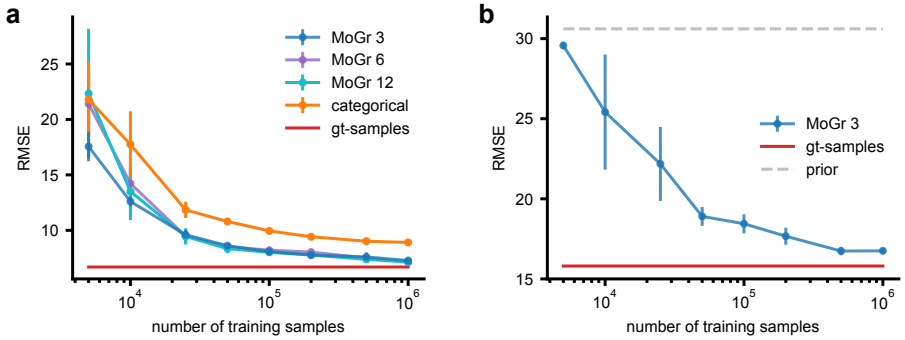

Figure S4. **SBMI performance is robust to the number of mixture components and scales to larger models. (a)** Posterior predictive performance for the large additive model with 11 components, and $2^{11} = 2048$ possible combinations of model components in terms of RMSE. We compare different choices for the distribution-family of the model posterior (either a mixture of Grassmann distributions with varying number of components (3, 6 and 12) (MoGr) or a categorical distribution). At the same time we increase also the number of components for the mixture of Gaussian parametere posterior network. **(b)** Posterior predictive performance for an additive model with 20 components (specified in Table S4) and $2^{20} \approx 1M$ possible combinations of model components. A categorical distribution can not be fitted anymore as we have very few to no samples for each combination of model components in a dataset of size $1M$.

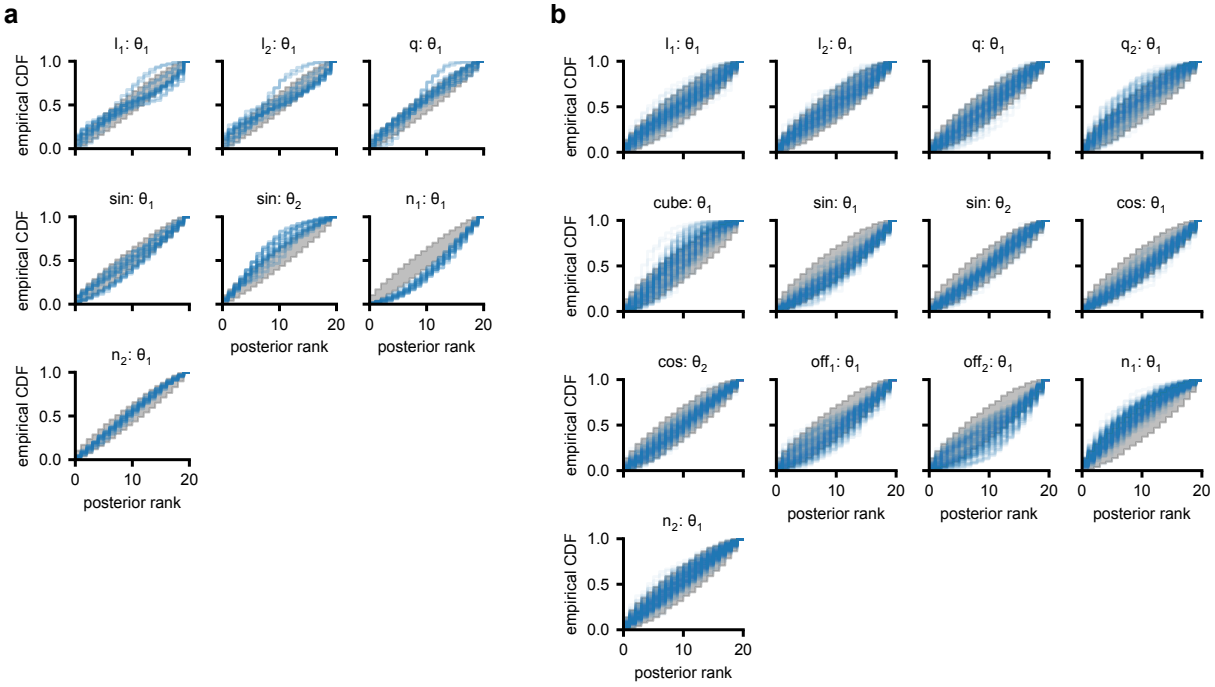

Figure S5. **Simulation-based calibration of the parameter posterior for the additive models. (a)** Posterior calibration of the *small* additive model, by individual model parameters for all possible model component combinations. Grey regions indicate the 99% confidence intervals of a uniform distribution, given the provided number of samples. **(b)** Same as (a) for the *large* additive model and for all models with at least 50 samples in the test dataset of 100k samples. For all plots we ranked the true parameter θ_o against 1k posterior samples.

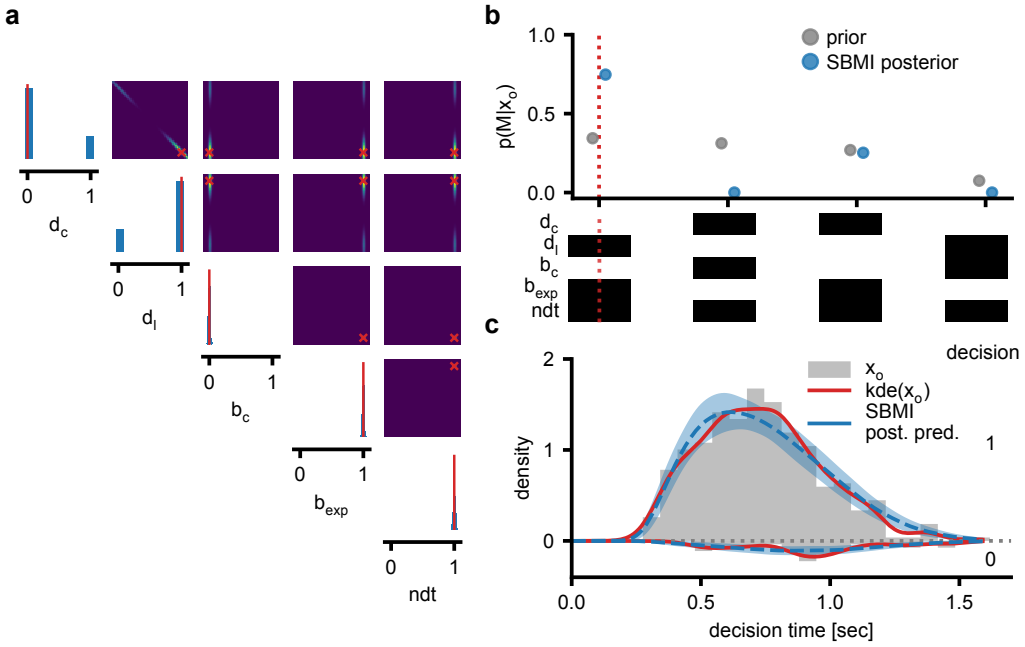

Figure S6. **DDM posterior and posterior predictives. (a)** Smoothed one- and two-dimensional marginal distribution for the (binary) model posterior. The ground truth model is indicated in red. **(b)** Model prior and SBMI model posterior with the ground truth model indicated as a red dotted line. **(c)** Posterior predictives for the example observation of Figure 5. The global uncertainty is shown as mean ± std. over predictions from different model posterior samples.

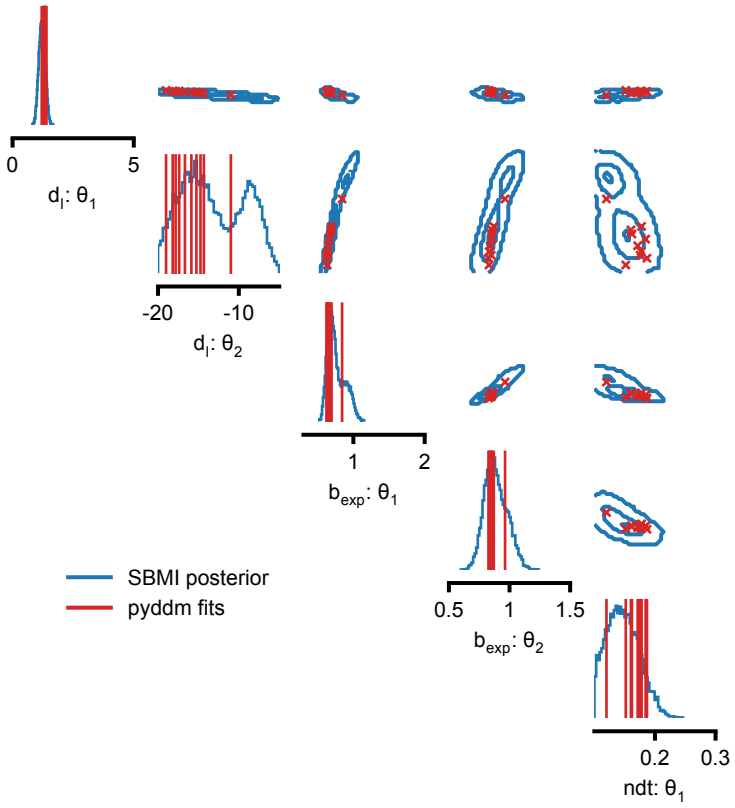

Figure S7. **Parameter posterior distribution for the DDM on experimental data.** One- and two-dimensional marginals of the SBMI parameter posterior for a coherence rate of 6.4%. Red markers indicate ten *pyDDM* fits for a fixed model with different random seeds, all resulting in similar loss values.

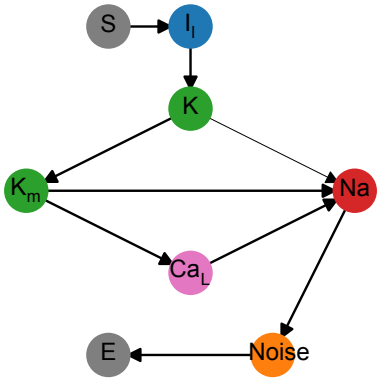

Figure S8. **Model prior for the Hodgkin-Huxley model.** Model prior graph. While potassium and sodium channels are always present in this definition, the presence of m-type potassium and calcium channels is inferred from the observation. This reflects our prior knowledge for spiking neurons in which sodium and potassium channels are essential for the spiking mechanism. The sampling distribution is shown in Fig. S11.

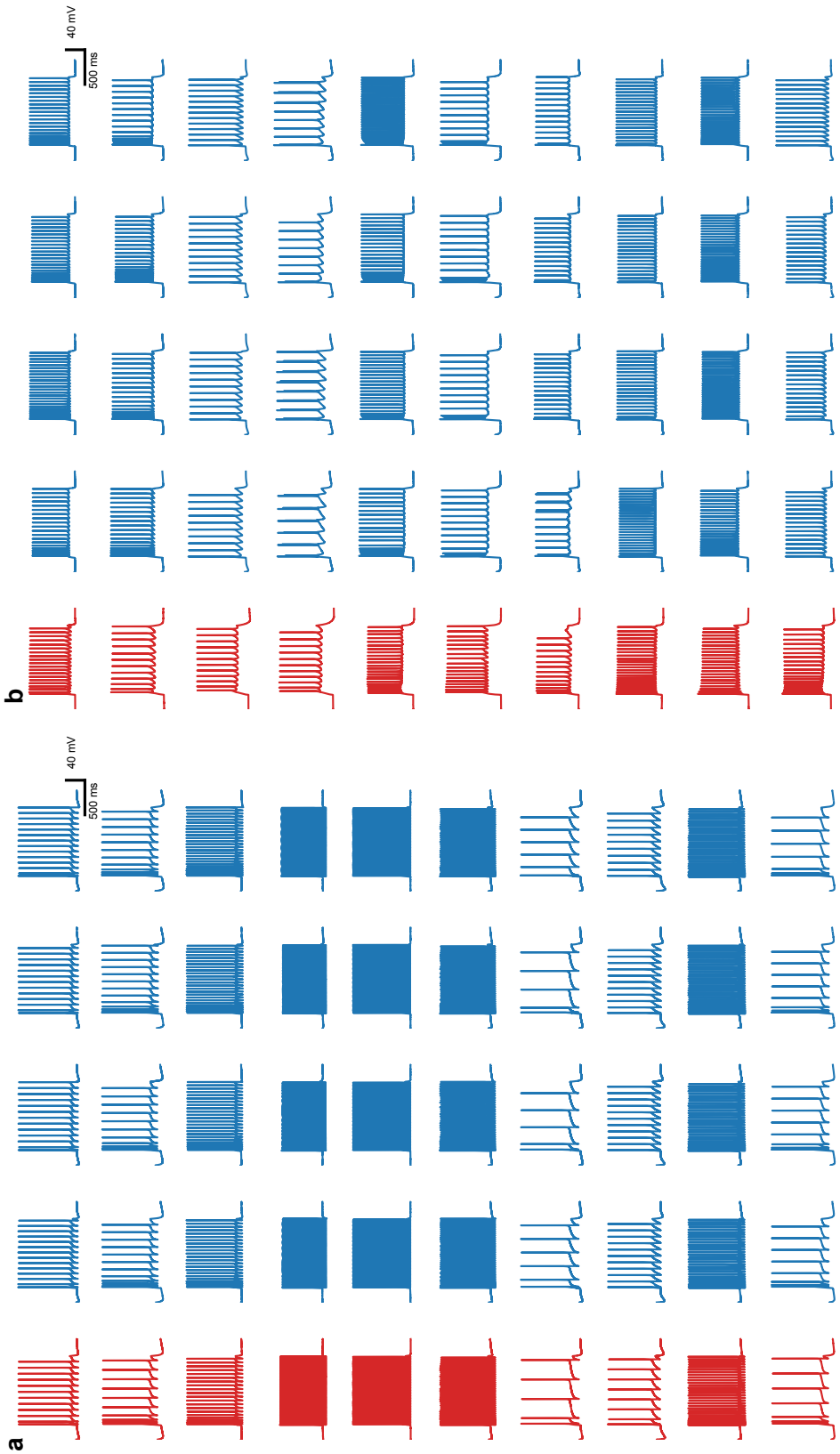

Figure S9. **Posterior predictives of the Hodgkin-Huxley model. (a)** Ten synthetic samples (red) with four posterior predictive samples each (blue). **(b)** Ten voltage recordings from the Allen Celltype database with four posterior predictive samples each.

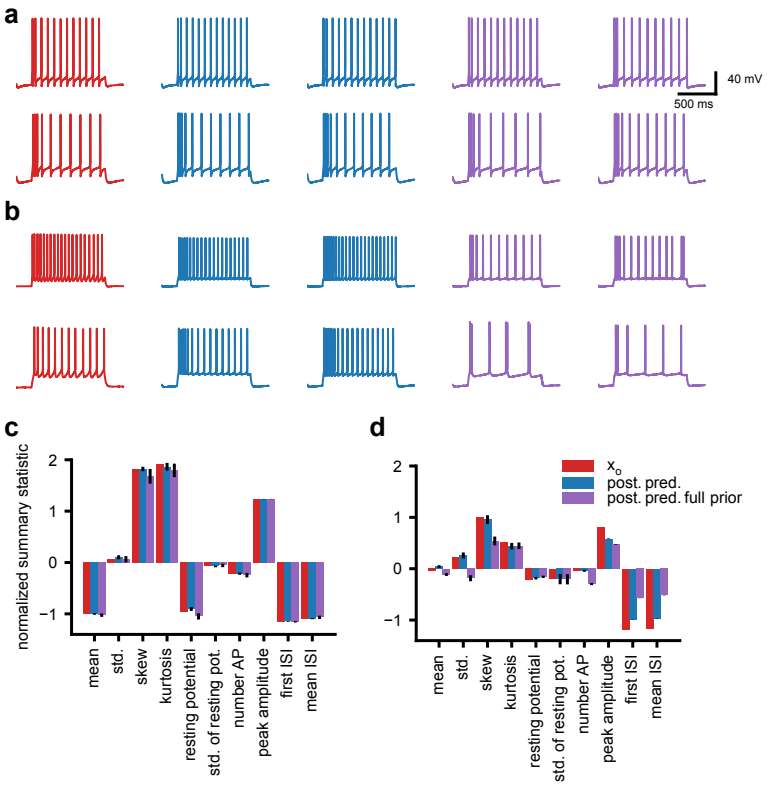

Figure S10. **Leveraging domain knowledge enhances posterior performance for the Hodgkin-Huxley model. (a)** Two synthetic samples (red) with two posterior predictive samples each (blue) (same as Fig. 7 a) with additional two posterior predictives from a model trained on a fully connected model prior (violet). **(b)** Two voltage recordings from the Allen Cell database (red) with two posterior predictive samples each for the standard model (blue) (same as Fig. 7 b) with additional two posterior predictives from a model trained on a fully connected model prior (violet). **(c)** Ten example summary statistics (out of 24) for the upper trace in (a) and summary statistics for ten posterior predictive samples from the respective model (mean±std.). **(d)** Ten example summary statistics for the voltage recording from the Allen dataset shown in (b), upper trace, and summary statistics for ten posterior predictive samples from the respective model (mean±std.).

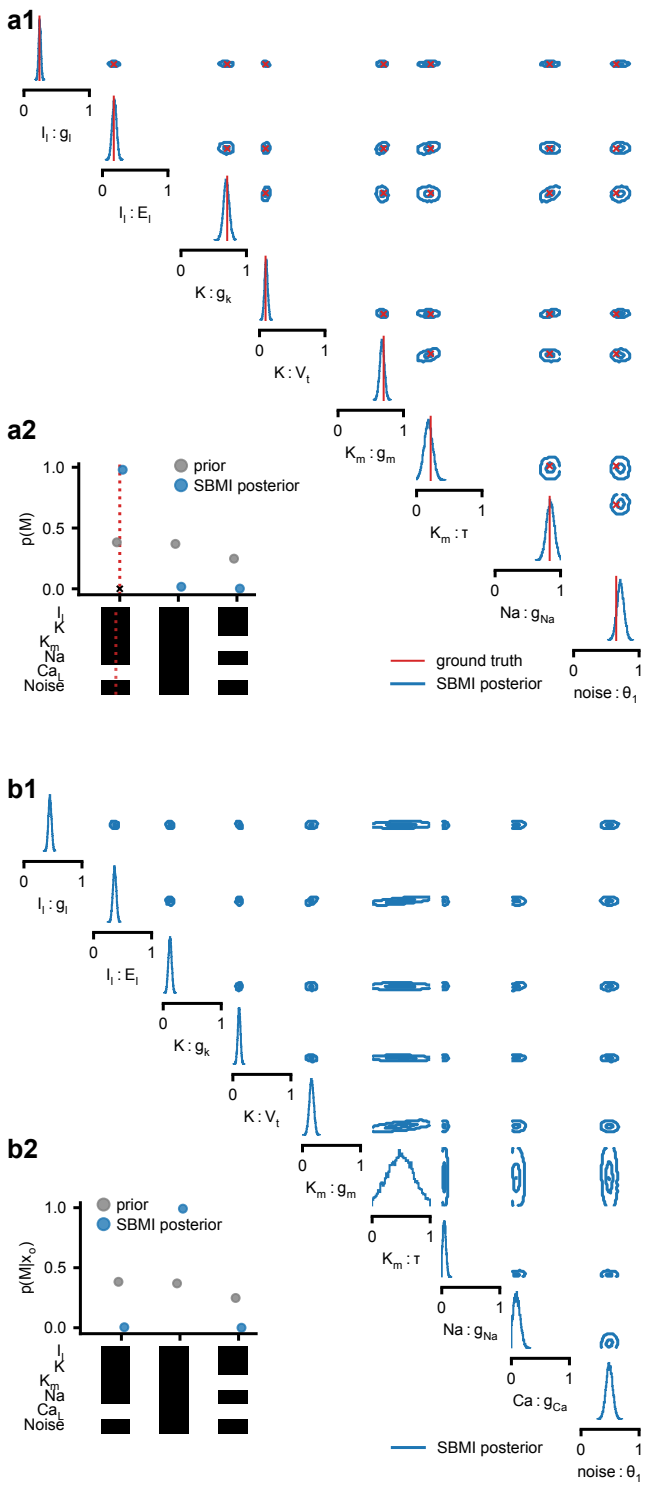

Figure S11. **Model and parameter posteriors for the Hodgkin-Huxley model. (a1)** Normalised one and two dimensional marginal distributions for the parameter posterior for the synthetic example shown in Fig. 7 a (upper trace). Red dots/lines are indicating the ground truth parameter θ_o. **(a2)** The corresponding model posterior for the example in (a1). Red dotted line indicates the ground truth model. **(b1)** Normalised one and two dimensional marginal distributions for the parameter posterior for the voltage recording from the Allen database shown in Fig. 7 b (upper trace). **(b2)** The corresponding model posterior for the example in (b1).

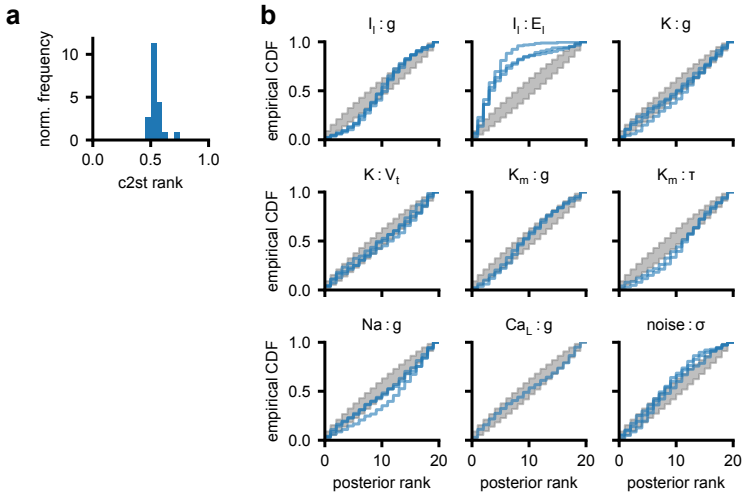

Figure S12. **Simulation-based calibration of the parameter posterior for the Hodgkin-Huxley model. (a)** Histogram of the c2st ranks. A value of 0.5 indicates a well calibrated posterior for which the rank statistics are indistinguishable from a uniform distribution. **(b)** Posterior calibration by individual model parameters for all possible model component combinations. Grey regions indicate the 99% confidence intervals of a uniform distribution, given the provided number of samples. For all plots we ranked the true parameter θ_o against 1k posterior samples.