Appendix

A1 Code and Reproducibility

Code to reproduce all experiments, including the full git commit history, will be made available upon acceptance. All neural networks and optimization loops are written in pytorch (Paszke et al., 2019). We tracked all experiments with hydra (Yadan, 2019). For NPE, we used the implementation in the sbi toolbox (Tejero-Cantero et al., 2020).

A2 Convergence proof for proposition 1

The proof closely follows standard proofs that regression converges to the conditional expectation.

Proof. We aim to prove that

$$\mathbb{E}_{\theta,\mathbf{x}\sim p(\boldsymbol{\theta},\mathbf{x}),\mathbf{a}\sim p(\mathbf{a})}[(c(\boldsymbol{\theta},\mathbf{a})-g(\mathbf{x},\mathbf{a}))^2] \geq \\ \mathbb{E}_{\theta,\mathbf{x}\sim p(\boldsymbol{\theta},\mathbf{x}),\mathbf{a}\sim p(\mathbf{a})}[(c(\boldsymbol{\theta},\mathbf{a})-\mathbb{E}_{\boldsymbol{\theta}'\sim p(\boldsymbol{\theta}|\mathbf{x})}[c(\boldsymbol{\theta}',\mathbf{a})])^2]$$

for every function $g(\mathbf{x},\mathbf{a})$. We begin by rearranging expectations:

$$\mathbb{E}_{\theta,\mathbf{x}\sim p(\boldsymbol{\theta},\mathbf{x}),\mathbf{a}\sim p(\mathbf{a})}[(c(\boldsymbol{\theta},\mathbf{a})-g(\mathbf{x},\mathbf{a}))^2] = \\ \mathbb{E}_{p(\mathbf{a})}[\mathbb{E}_{\theta,\mathbf{x}\sim p(\boldsymbol{\theta},\mathbf{x})}[(c(\boldsymbol{\theta},\mathbf{a})-g(\mathbf{x},\mathbf{a}))^2]]$$

Below, we prove that, for a given $\mathbf{a}$ and for *any* $\mathbf{x}$, the optimal $g(\mathbf{x},\mathbf{a})$ is the conditional expectation $\mathbb{E}_{\boldsymbol{\theta}'\sim p(\boldsymbol{\theta}|\mathbf{x})}[c(\boldsymbol{\theta}',\mathbf{a})]$:

$$\mathbb{E}_{\theta,\mathbf{x}\sim p(\boldsymbol{\theta},\mathbf{x})}[(c(\boldsymbol{\theta},\mathbf{a})-g(\mathbf{x},\mathbf{a}))^2] = \\ \mathbb{E}_{\theta,\mathbf{x}\sim p(\boldsymbol{\theta},\mathbf{x})}[(c(\boldsymbol{\theta},\mathbf{a})-\mathbb{E}_{\boldsymbol{\theta}'\sim p(\boldsymbol{\theta}|\mathbf{x})}[c(\boldsymbol{\theta}',\mathbf{a})]+\mathbb{E}_{\boldsymbol{\theta}'\sim p(\boldsymbol{\theta}|\mathbf{x})}[c(\boldsymbol{\theta}',\mathbf{a})]-g(\mathbf{x},\mathbf{a}))^2] = \\ \mathbb{E}_{\theta,\mathbf{x}\sim p(\boldsymbol{\theta},\mathbf{x})}[(c(\boldsymbol{\theta},\mathbf{a})-\mathbb{E}_{\boldsymbol{\theta}'\sim p(\boldsymbol{\theta}|\mathbf{x})}[c(\boldsymbol{\theta}',\mathbf{a})])^2+(\mathbb{E}_{\boldsymbol{\theta}'\sim p(\boldsymbol{\theta}|\mathbf{x})}[c(\boldsymbol{\theta}',\mathbf{a})]-g(\mathbf{x},\mathbf{a}))^2]+X$$

with

$$X=\mathbb{E}_{\theta,\mathbf{x}\sim p(\boldsymbol{\theta},\mathbf{x})}[(c(\boldsymbol{\theta},\mathbf{a})-\mathbb{E}_{\boldsymbol{\theta}'\sim p(\boldsymbol{\theta}|\mathbf{x})}[c(\boldsymbol{\theta}',\mathbf{a})])(\mathbb{E}_{\boldsymbol{\theta}'\sim p(\boldsymbol{\theta}|\mathbf{x})}[c(\boldsymbol{\theta}',\mathbf{a})]-g(\mathbf{x},\mathbf{a}))],$$

By the law of iterated expectations, one can show that $X=0$:

$$X=\mathbb{E}_{\mathbf{x}'\sim p(\mathbf{x})}\mathbb{E}_{\mathbf{x}\sim p(\mathbf{x})}\mathbb{E}_{\boldsymbol{\theta}\sim p(\boldsymbol{\theta}|\mathbf{x})}[(c(\boldsymbol{\theta},\mathbf{a})-\mathbb{E}_{\boldsymbol{\theta}'\sim p(\boldsymbol{\theta}|\mathbf{x})}[c(\boldsymbol{\theta}',\mathbf{a})])(\mathbb{E}_{\boldsymbol{\theta}'\sim p(\boldsymbol{\theta}|\mathbf{x})}[c(\boldsymbol{\theta}',\mathbf{a})]-g(\mathbf{x},\mathbf{a}))].$$

The first term in the product reads a difference of the same term, and, thus, is zero. Thus, since $(\mathbb{E}_{\boldsymbol{\theta}'\sim p(\boldsymbol{\theta}|\mathbf{x})}[c(\boldsymbol{\theta}',\mathbf{a})]-g(\mathbf{x},\mathbf{a}))^2\geq 0$, we have:

$$\mathbb{E}_{\theta,\mathbf{x}\sim p(\boldsymbol{\theta},\mathbf{x})}[(c(\boldsymbol{\theta},\mathbf{a})-g(\mathbf{x},\mathbf{a}))^2]\geq\mathbb{E}_{\theta,\mathbf{x}\sim p(\boldsymbol{\theta},\mathbf{x})}[(c(\boldsymbol{\theta},\mathbf{a})-\mathbb{E}_{\boldsymbol{\theta}'\sim p(\boldsymbol{\theta}|\mathbf{x})}[c(\boldsymbol{\theta}',\mathbf{a})])^2]$$

Because this inequality holds for *any* $\mathbf{a}$, the average over $p(\mathbf{a})$ will also be minimized if and only if $g(\mathbf{x},\mathbf{a})$ matches the conditional expectation $\mathbb{E}_{\boldsymbol{\theta}'\sim p(\boldsymbol{\theta}|\mathbf{x})}[c(\boldsymbol{\theta}',\mathbf{a})]$ for any $\mathbf{a}$ within the support of $p(\mathbf{a})$. □

A3 Experimental setup

We used mainly the same hyperparameters for all benchmark tasks and for the Bayesian Virtual Epileptic Patient task. Only the learning rate was adjusted to 0.005 for Lotka Volterra, while it was set to 0.001 for all other tasks.

For NPE, we used the default parameters of the sbi package (Tejero-Cantero et al., 2020), apart from the density estimator, for which we used a neural spline (Durkan et al., 2019). Following Hashemi et al. (2023), we used a masked autoregressive flow (Papamakarios et al., 2017) for the Bayesian Virtual Epileptic Patient with default parameters as set in the sbi package.

For BAM, we used a feedforward residual network (He et al., 2016) with 3 hidden layers and 50 hidden units each. We use ReLU activation functions and, for cases where we know that the expected cost is going to be positive and bounded by 1, squash the output through a sigmoid. We use the Adam optimizer (Kingma and Ba, 2015). As described in Sec. 4, the size of the training dataset varied between 500 and 100k. In all cases, the training dataset was split 90:10 into training and validation and with a batchsize of 500.

As distribution over all permissible actions, $p(\mathbf{a})$, we use a uniform distribution over continuous actions between 0 and 100 for all benchmark tasks, $\mathcal{U}(0, 100)$, and a discrete uniform distribution over three classes for the Bayesian Virtual Epileptic Patient.

A4 Benchmark tasks

Below, we provide details on the simulators and cost functions used for benchmarking NPE-MC and BAM.

Toy example: The prior follows a uniform distribution

$$p(\boldsymbol{\theta}) = \mathcal{U}[0, 5]$$

and the likelihood is given by:

$$p(\mathbf{x}|\boldsymbol{\theta}) = 50 + 0.5\boldsymbol{\theta}(5 - \boldsymbol{\theta})^4 + \nu,$$

where $\nu \sim \mathcal{N}(\theta, 10)$. We obtained (an approximation to) the ground truth posterior via quadrature.

As cost function, we used a flipped (along the y-axis) bell-shaped curve whose width decreases with increasing parameter $\boldsymbol{\theta}$:

$$c(\boldsymbol{\theta}, \mathbf{a}) = 1 - \exp\left(\frac{(\boldsymbol{\theta} - \mathbf{a})^2}{(\frac{2}{|\boldsymbol{\theta}| + \epsilon})^2}\right)$$

with $\epsilon = 0.1$. As the parameter space and the action space differ, $\boldsymbol{\theta}$ and $\mathbf{a}$ were rescaled to both lie within the range of $[0, 10]$.

Linear Gaussian: We used exactly the same simulator as in Lueckmann et al. (2021). As cost function, we used a flipped (along the y-axis) bell-shaped curve whose width decreases towards the extremes of the parameter range:

$$c(\boldsymbol{\theta}, \mathbf{a}) = 1 - \exp\left(\frac{(\boldsymbol{\theta} - \mathbf{a})^2}{(\frac{0.5}{|\boldsymbol{\theta} - 0.5 \cdot (\boldsymbol{\theta}_{max} - \boldsymbol{\theta}_{min})| + \epsilon})^2}\right)$$

with $\epsilon = 0.1$. As the parameter space and the action space differ, $\boldsymbol{\theta}$ and $\mathbf{a}$ were rescaled to both lie within the range of $[0, 10]$. Additionally, a shift was introduced such that the variance is widest at 5 and decreases towards both ends of the value range.

Lotka Volterra: We used exactly the same simulator as in Lueckmann et al. (2021). We evaluated four different cost functions, where each individual cost function assesses only the accuracy of the estimated marginal distribution (i.e., the cost function ignores three out of four parameters). The cost function for individual parameters θ_i with $i = 0, 1, 2, 3$ is given by:

$$c(\boldsymbol{\theta}_i, \mathbf{a}) = 1 - \exp\left(\frac{(\boldsymbol{\theta}_i - \mathbf{a})^2}{(\frac{3}{|\boldsymbol{\theta}| + \epsilon})^2}\right)$$

with $\epsilon = 0.1$. As the parameter space and the action space differ, $\boldsymbol{\theta}$ and $\mathbf{a}$ were rescaled to both lie within the range of $[0, 10]$.

SIR: We used exactly the same simulator as in Lueckmann et al. (2021). The cost function is based on the ration between parameters, $\mathbf{r} = \frac{\boldsymbol{\beta}}{\boldsymbol{\gamma}}$, and is given by:

$$c(\mathbf{r}, \mathbf{a}) = 1 - \exp\left(\frac{(\mathbf{r} - \mathbf{a})^2}{(\frac{2}{|10 - |\mathbf{r} - 1|| + \epsilon})^2}\right)$$

with $\epsilon = 0.1$. As the parameter space and the action space differ, $\mathbf{r}$ and $\mathbf{a}$ were rescaled to both lie within the range of $[0, 10]$. Additionally a shift by 1 in parameter space was introduced such that the cost function is very sensitive around $\mathbf{r} = 1$ and variance increases with increasing distance to 1.

A5 Details on Bayesian Virtual Epileptic Patient (BVEP) simulator

The Bayesian Virtual Epileptic Patient provides a framework to simulate neural activity in a connected network of brain regions to model epilepsy spread (Hashemi et al., 2020; Jirsa et al., 2017). The dynamics of individual brain regions are controlled by a simulator called 'Epileptor' that allows to reproduce dynamics of electrical activity during seizure-like events. Hashemi et al. (2020) introduce two versions of the Epileptor, the full 6D model and a reduced variant, the 2D Epileptor. The full Epileptor model (Jirsa et al., 2014; Hashemi et al., 2020) is composed of five state variables that couple two oscillatory dynamical systems on three different time scales. Variables x_1 and y_1 are associated with fast electrical discharges during seizures at the fastest time scale. Variables x_2 and y_2 act on an intermediate timescale and the permittivity state variable z that governs the transition between ictal, i.e. during a seizure, and interictal, i.e. in between seizures, states on a slow time scale. See Jirsa et al. (2014) and Hashemi et al. (2020) for a detailed description of the Epileptor equations. For this work, we focus on a single isolated brain region and used the reduced '2D Epileptor' model which results from applying averaging methods to make the effects of x_2 and y_2 negligible and further assuming time scale separation so that x_1 and y_1 collapse on the slow manifold (Hashemi et al., 2020).

The 2D Epileptor takes four parameters to generate a time series of neural activity: η describes the excitability of the tissue, τ controls the time scale separation and x_{init} and z_{init} are initial values for the state variables. See Hashemi et al. (2020) for a full description. Based on the level of excitability η, brain regions can be categorized into three types (Hashemi et al., 2020):

- Epileptogenic Zone (EZ): if $\eta > \eta_{EZ}$, the Epileptor can trigger seizures autonomously
- Propagation Zone (PZ): if $\eta_{PZ} = \eta_{EZ} - \Delta\eta < \eta < \eta_{EZ}$, the Epileptor cannot trigger seizures autonomously, but (in a connected network of brain regions) can be recruited during the evolution of an epileptic seizure
- Healthy Zone (HZ): if $\eta < \eta_{PZ}$, the Epileptor does not trigger seizures

Following Hashemi et al. (2023), we set the critical value $\eta_{EZ} = -2.05$ and $\Delta\eta = 1.0$, i.e. $\eta_{PZ} = \eta_{EZ} - \Delta\eta = -3.05$. These types of brain regions form the basis for the decision task described in Sec. 4.3.

When observations are high-dimensional as it is the case for the time series generated by the Epileptor, low-dimensional summary statistics are commonly used for inference. For both, NPE-MC and BAM, we used 10 time-independent summary statistics to describe the observed neural activity. These are the mean, median, standard deviation, skew, and kurtosis of the time series as well as higher moments (up to the 10th moment of the observation), the power envelope, seizure onset, the amplitude phase, and spectral power.

A6 Supplementary figures

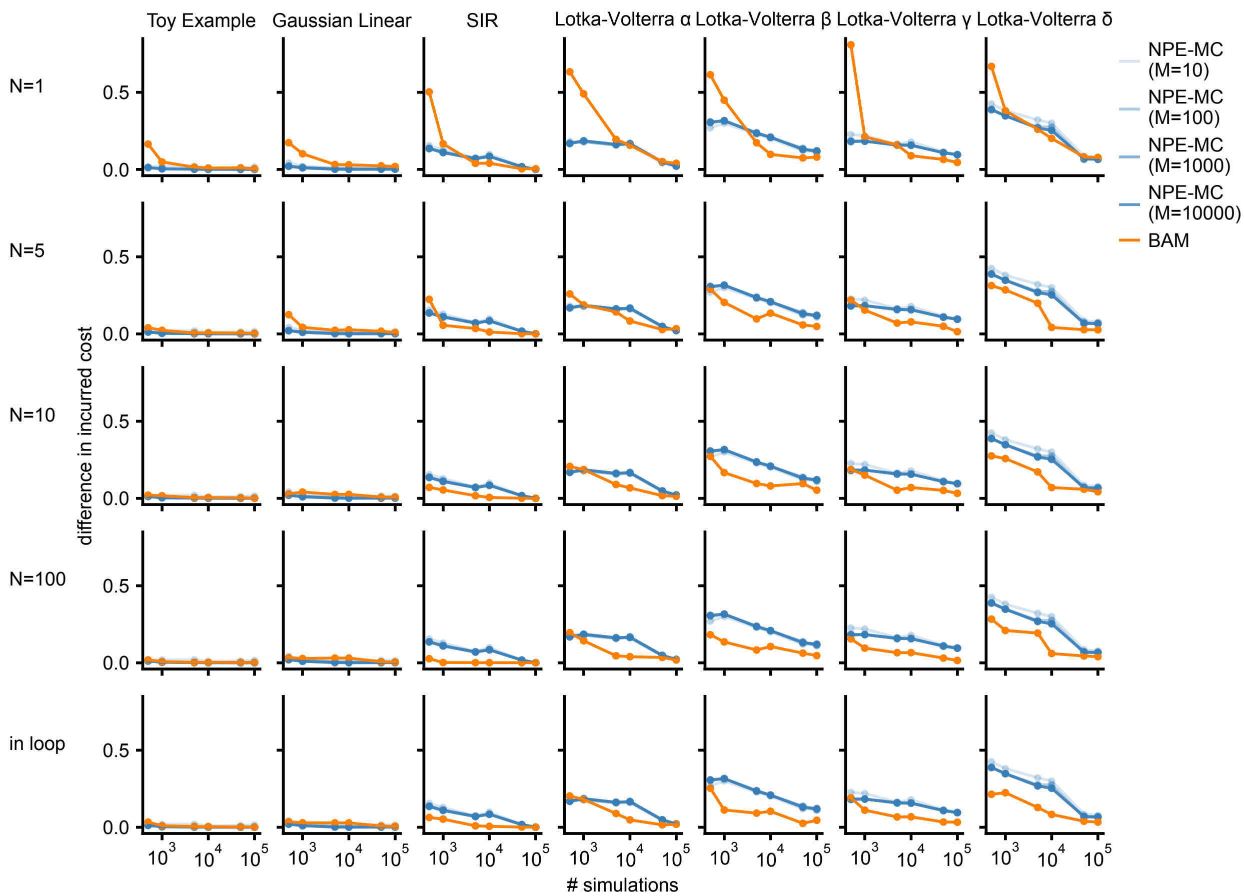

Figure A1: **Expected costs for sampling diverse numbers of actions per data point and for a different number of Monte-Carlo (MC) samples.** NPE-MC is the same for every row (blue). Different shades of blue indicate NPE-MC with different number of MC-samples. For BAM, we sampled $N = \{1, 5, 10, 100\}$ actions per $(\boldsymbol{\theta}, \mathbf{x})$ pair in the training dataset (rows). In the last row we sampled one new action at every epoch. This is shown in the main paper in Fig. 3.