MULTIFIDELITY SIMULATION-BASED INFERENCE FOR COMPUTATIONALLY EXPENSIVE SIMULATORS

Anonymous authors
Paper under double-blind review

ABSTRACT

Across many domains of science, stochastic models are an essential tool to understand the mechanisms underlying empirically observed data. Models can be of different levels of detail and accuracy, with models of high-fidelity (i.e., high accuracy) to the phenomena under study being often preferable. However, inferring parameters of high-fidelity models via simulation-based inference is challenging, especially when the simulator is computationally expensive. We introduce MF-(TS)NPE, a multifidelity approach to neural posterior estimation that uses transfer learning to leverage inexpensive low-fidelity simulations to efficiently infer parameters of high-fidelity simulators. MF-(TS)NPE applies the multifidelity scheme to both amortized and non-amortized neural posterior estimation. We further improve simulation efficiency by introducing MF-TSNPE-AF, a sequential variant that uses an acquisition function targeting the predictive uncertainty of the density estimator to adaptively select high-fidelity parameters. On established benchmark and neuroscience tasks, our approaches require up to two orders of magnitude fewer high-fidelity simulations than current methods, while showing comparable performance. Overall, our approaches open new opportunities to perform efficient Bayesian inference on computationally expensive simulators.

1 INTRODUCTION

Stochastic models are used across science and engineering to capture complex properties of real systems through simulations (Barbers et al., 2024; Nelson & Pei, 2021; Pillow & Scott, 2012). These simulators encode domain-specific knowledge and provide a means to generate high-fidelity synthetic data, enabling accurate forward modeling of experimental outcomes. However, inferring model parameters from observed data can be challenging, especially when simulators are stochastic, the likelihoods of the simulators are inaccessible, or when simulations are computationally expensive.

Simulation-based inference (SBI) addresses these challenges by leveraging forward simulations to infer the posterior distribution, enabling quantification of uncertainty even when the likelihood is intractable (Cranmer et al., 2020). The challenge of extending sampling-based SBI methods like Approximate Bayesian Computation (ABC) (Tavaré et al., 1997; Pritchard et al., 1999) to problems with large numbers of parameters has driven significant advancements in neural-based approaches that estimate the likelihood (Papamakarios et al., 2019), the likelihood-to-evidence ratio (Hermans et al., 2020), or directly the posterior (Greenberg et al., 2019; Lueckmann et al., 2017; Papamakarios & Murray, 2016). In particular, amortized Neural Posterior Estimation (NPE) trains a neural density estimator to directly approximate the posterior, bypassing the need to estimate the model evidence (Papamakarios & Murray, 2016). To improve inference for a fixed observation and allow stable training, truncated sequential variants have been introduced for neural posterior estimation (TSNPE) (Deistler et al., 2022), and neural ratio estimation (Miller et al., 2021). These approaches have leveraged recent progress in neural density estimation to improve the scalability and accuracy of SBI, allowing parameter inference in problems with higher dimensionality than was previously achievable (Ramesh et al., 2021; Gloeckler et al., 2024). Despite these advancements, SBI methods face computational challenges for scenarios involving expensive simulations or high-dimensional parameter spaces, as state-of-the-art methods often require extensive simulation budgets to achieve reliable posterior estimates (Lueckmann et al., 2021).

Multifidelity modeling offers a solution to this problem by balancing precision and efficiency. It combines accurate but costly high-fidelity models (Hoppe et al., 2021; Behrens & Dias, 2015) with faster, less accurate low-fidelity models. Here, low-fidelity models could be simplifications made possible through domain knowledge about the high-fidelity models, low-dimensional projection of the high-fidelity model, or surrogate modeling (Peherstorfer et al., 2018). For example, Reynolds-averaged Navier-Stokes (RANS) models simplify turbulent flow simulations in aerodynamics (Han et al., 2013), while climate models often reduce complexity by focusing on specific atmospheric effects (Held, 2005; Majda & Gershgorin, 2010). Similarly, mean-field approximations are used to capture certain features of spiking neural network dynamics (Vogels et al., 2011; Dayan & Abbott, 2001). Multifidelity methods have proven effective across domains—enhancing optimization through multifidelity Bayesian optimization (Song et al., 2019; Kandasamy et al., 2017), and improving the efficiency of inference through multifidelity Monte Carlo approaches (Peherstorfer et al., 2016; Nobile & Tesei, 2015; Giles, 2008; Zeng et al., 2023). In the context of SBI, we hypothesized that by leveraging the complementarity of high- and low-fidelity simulators, it would be possible to reduce the computational cost of inference while retaining inference accuracy.

In this work, we present MF-(TS)NPE, a multifidelity approach that improves the efficiency of amortized and non-amortized neural posterior estimation for expensive simulators. MF-(TS)NPE reduces the computational burden of posterior estimation by pre-training a neural density estimator on low-fidelity simulations and refining the inference with a smaller set of high-fidelity simulations. Additionally, we present MF-TSNPE-AF, an extension of MF-TSNPE with active learning, facilitating targeted parameter space exploration to effectively enhance high-fidelity posterior estimates given single observations. We focus on multifidelity cases where both models are simulators and where the low-fidelity model is a simplified version of the high-fidelity model, designed based on domain expertise. We demonstrate that for four benchmark tasks and two computationally expensive neuroscience simulators, our multifidelity approach can identify the posterior distributions more efficiently than NPE and TSNPE, often reducing the number of required high-fidelity simulations by orders of magnitude.

2 Background

Multifidelity methods for inference Multifidelity has been widely explored in the context of likelihood-based inference (Peherstorfer et al., 2018), from maximum likelihood estimation approaches (Maurais et al., 2023) to Bayesian inference methods (Vo et al., 2019; Catanach et al., 2020). For cases where the likelihood is not explicitly available, several sampling-based multifidelity methods have been proposed within the framework of ABC (Prescott & Baker, 2020; Warne et al., 2022; Prescott et al., 2024; Prescott & Baker, 2021). However, these methods inherit limitations of ABC approaches, particularly in high-dimensional parameter spaces, where neural density estimators offer more scalable alternatives to complex real-world problems (Lueckmann et al., 2021). Concurrently with our work, Thiele et al. (2025) developed a multifidelity SBI approach based on response distillation, Hikida et al. (2025) adapted multilevel Monte Carlo techniques to SBI, and Saoulis et al. (2025) applied transfer learning to accelerate inference on a cosmological task.

Beyond SBI, multifidelity has been explored in Bayesian optimization, where Gaussian process models integrate data of different fidelities to infer expensive functions (e.g., Song et al., 2019; Zanjani Foumani et al., 2023). These approaches focus on learning surrogate likelihood functions rather than posteriors over simulator parameters, but they highlight the broad applicability of the multifidelity concept.

Transfer learning and simulators To facilitate learning in a target domain, transfer learning borrows knowledge from a source domain (Panigrahi et al., 2021). This is often done when the target dataset is smaller than the source dataset (Larsen-Freeman, 2013). For numerical simulators, transfer learning approaches have been used to lower the simulation budget, for instance, in CO_2 forecasting (Falola et al., 2023), surrogate modeling (Wang et al., 2024) and model inversion with physics-informed neural networks (Haghighat et al., 2021). To the best of our knowledge, the potential of transfer learning for computationally efficient simulation-based inference has not been fully realized yet.

Simulation-efficient SBI Recent work reduces the cost of SBI for expensive simulators through active learning or efficient representations. Active learning methods adaptively select simulation parameters for neural likelihood or posterior estimation (Lueckmann et al., 2019; Griesemer et al., 2024), paralleling Bayesian optimization for ABC (Gutmann & Corander, 2016). Efficiency also improves through learned representations such as signature-based features (Dyer et al., 2022), compositional models (Gloeckler et al., 2025), or self-consistency objectives (Schmitt et al., 2024a;b). Unlike these single-fidelity approaches, MF-(TS)NPE leverages an expert-designed low-fidelity simulator and combines transfer learning with active learning to refine posterior estimates efficiently.

3 METHODS

MF-(TS)NPE is a multifidelity approach to Neural Posterior Estimation (NPE) for computationally expensive simulators leveraging transfer learning and, in its sequential variant, active learning. We present our approach in Sec. 3.1. In Sec. 3.1.4, we discuss the evaluation metrics used to compare our method against NPE (Greenberg et al., 2019), TSNPE (Deistler et al., 2022), and MF-ABC (Prescott & Baker, 2020). MF-(TS)NPE is summarized in Fig. 1, Algorithms 1 and 3.

3.1 MULTIFIDELITY NPE

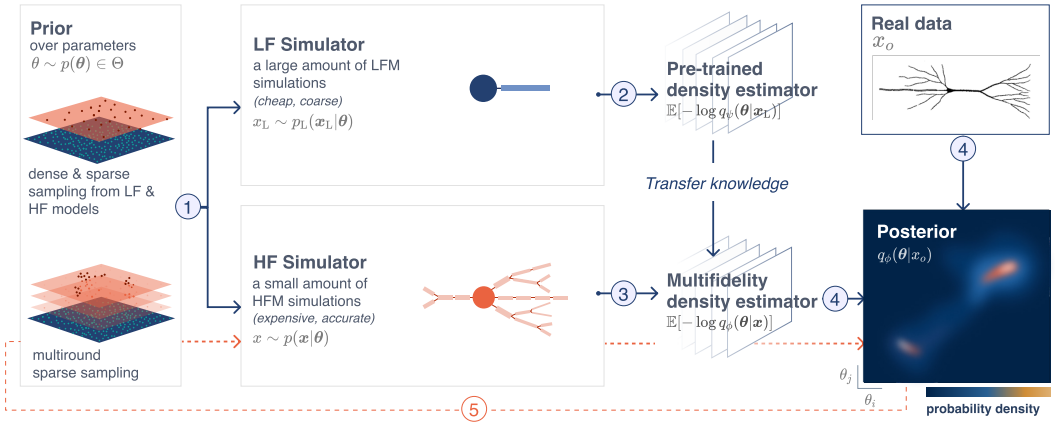


Figure 1: **Multifidelity Neural Posterior Estimation** proceeds by dense sampling from the prior distribution, running the low-fidelity simulator (*e.g., a two-compartment neuron model* (Hodgkin & Huxley, 1952)), and training a neural density estimator with a negative log-likelihood loss. MF-NPE then retrains the pre-trained network on sparse samples from the same prior distribution and respective high-fidelity simulations (*e.g., a multicompartmental neuron model* (Rall, 1995)). Given empirical observations $\boldsymbol{x_o}$, MF-NPE estimates the posterior distribution given the high-fidelity model. In the sequential case, the parameters for high-fidelity simulations are drawn from iterative refinements of the prior distribution within the support of the current posterior estimate, at some observation $\boldsymbol{x_o}$.

We aim to infer the posterior distribution over the parameters $\boldsymbol{\theta}$ of a computationally expensive high-fidelity simulator $p(\boldsymbol{x}|\boldsymbol{\theta})$, with computational cost of a single simulation c. We designate the simulator as high-fidelity if the model accurately captures the empirical phenomenon, but incurs high computational cost when generating simulations. We assume that we have access to a low-fidelity simulator $p_L(\boldsymbol{x}_\mathrm{L}|\boldsymbol{\theta})$, describing a simplification of the phenomenon of interest with cost $c_L \ll c$. We assume that both simulators operate over the same domain of observations $\boldsymbol{x}$, and the parameters of the low-fidelity model form at least a subset (and at most the entirety) of the high-fidelity parameters. Our goal is to develop an estimator that leverages low-fidelity simulations to infer the posterior distribution over parameters of the high-fidelity model with limited high-fidelity simulations, without access to a tractable likelihood for either simulator.

As with NPE (Papamakarios & Murray, 2016; Greenberg et al., 2019), to estimate the posterior density over model parameters $\boldsymbol{\theta}$ for which the likelihood function is unavailable, we consider a sufficiently

expressive neural density estimator $q_\phi(\boldsymbol{\theta}|\boldsymbol{x})$, and train it to minimize the negative log-likelihood loss:

$$\mathcal{L}(\phi) = \mathbb{E}_{\theta\sim p(\boldsymbol{\theta})}\mathbb{E}_{x\sim p(\boldsymbol{x}|\boldsymbol{\theta})}\left[-\log q_\phi(\boldsymbol{\theta}|\boldsymbol{x})\right], \tag{1}$$

where $\boldsymbol{\theta}$ is sampled from the prior distribution, $\boldsymbol{x}$ denotes the respective simulations (i.e., samples from $p(\boldsymbol{x}|\boldsymbol{\theta})$), and $\boldsymbol{\phi}$ are the network parameters. By minimizing $\mathcal{L}(\cdot)$, the neural density estimator approximates the conditional distribution $p(\boldsymbol{\theta}|\boldsymbol{x})$ directly (Papamakarios & Murray, 2016) (proof of convergence in Appendix B). Given an empirical observation $\boldsymbol{x_o}$, we can then estimate the posterior over parameters $p(\boldsymbol{\theta}|\boldsymbol{x_o})$. To ensure $q_\phi(\boldsymbol{\theta}|\boldsymbol{x_o})$ closely approximates the true posterior $p(\boldsymbol{\theta}|\boldsymbol{x_o})$, the density estimator must be sufficiently expressive. We use neural spline flows (NSFs) (Durkan et al., 2019), expressive normalizing flows that have been shown empirically to be competitive for SBI (Lueckmann et al., 2021). To avoid overfitting when training NSFs, we use the same validation-based early stopping criterion S as in the SBI package (Boelts et al., 2024) (details in Appendix.C.1).

3.1.1 Transfer learning

MF-NPE leverages representations learned from low-fidelity simulations to reduce the number of high-fidelity simulations required to approximate a high-fidelity posterior. To that end, MF-NPE adopts a *fine-tuning* strategy of transfer learning: Let ψ be the parameters of the low-fidelity neural density estimator $q_\psi(\boldsymbol{\theta}|\boldsymbol{x}_\mathrm{L})$ and let ϕ be the parameters of the high-fidelity density estimator $q_\phi(\boldsymbol{\theta}|\boldsymbol{x})$. MF-NPE minimizes the loss $\mathcal{L}(\phi) = \mathbb{E}_{\theta\sim p(\boldsymbol{\theta})}\mathbb{E}_{x\sim p(\boldsymbol{x}|\boldsymbol{\theta})}\left[-\log q_\phi(\boldsymbol{\theta}|\boldsymbol{x})\right]$ on the high-fidelity task, where the parameters ϕ are initialized on the pretrained low-fidelity network parameters ψ. We argue that by pre-training on low-fidelity simulations, the density estimator learns useful features up front (i.e., the feature spaces of the low- and high-fidelity density estimators overlap), so fewer high-fidelity simulations suffice to refine the posterior estimates. Indeed, Tahir et al. (2024) shows that once networks learn suitable features for a given predictive task, they drastically reduce the sample complexity for related tasks. Other strategies to pretraining are discussed in Appendix G.4.

MF-NPE can naturally accommodate more than two fidelity levels (Appendix L), does not require more hyperparameter tuning than NPE (Appendix C.1), and is applicable in situations where the low-fidelity model has fewer parameters than the high-fidelity model. In this setting, the parameters that are exclusive to the high-fidelity model are treated as dummy variables in the pre-trained density estimator. The pre-conditioning with these variables leads to the pre-trained neural density estimator to effectively estimate the prior distribution over the respective parameters (OU3 and OU4 in Appendix I.1). As shown below, our method is compatible with both embedding networks and hand-crafted summary statistics of the observations.

Algorithm 1 MF-NPE

Input: N pairs of $(\boldsymbol{\theta}, \boldsymbol{x}_\mathrm{L})$; M pairs of $(\boldsymbol{\theta}, \boldsymbol{x})$; conditional density estimators $q_\psi(\boldsymbol{\theta}|\boldsymbol{x}_\mathrm{L})$ and $q_\phi(\boldsymbol{\theta}|\boldsymbol{x})$ with respectively learnable parameters ψ and ϕ; early stopping criterion S.
$\mathcal{L}(\psi) = \frac{1}{N}\sum_{i=1}^{N} -\log q_\psi\left(\boldsymbol{\theta}_i|\boldsymbol{x}_i^\mathrm{L}\right)$. /* Low-fidelity model */
for epoch in epochs **do**
    train q_ψ to minimize $\mathcal{L}(\psi)$ until S is reached.
end for
Initialize q_ϕ with weights and biases of trained q_ψ. /* High-fidelity model */
$\mathcal{L}(\phi) = \frac{1}{M}\sum_{i=1}^{M} -\log q_\phi\left(\boldsymbol{\theta}_i|\boldsymbol{x}_i\right)$.
for epoch in epochs **do**
    train q_ϕ to minimize $\mathcal{L}(\phi)$ until S is reached.
end for

3.1.2 Sequential training

In addition to learning amortized posterior estimates with NPE, our approach naturally extends to sequential training schemes when estimating the non-amortized posterior $q_\phi(\boldsymbol{\theta}|\boldsymbol{x_o})$. Rather than sampling model parameters from the prior, sequential methods introduce an active learning scheme that iteratively refines the posterior estimate for a specific observation $\boldsymbol{x_o}$. These methods – known as Sequential Neural Posterior Estimation (Papamakarios & Murray, 2016; Lueckmann et al., 2017) – have shown increased simulation efficiency when compared to NPE (Lueckmann et al., 2021). However, applying these methods with flexible neural density estimators requires a modified loss

that suffers from instabilities in training and posterior leakage (Greenberg et al., 2019). Truncated Sequential Neural Posterior Estimation (TSNPE) mitigates these issues by sampling from a truncated prior distribution that covers the support of the posterior. This leads to a simplified loss function and increased training stability, while retaining performance (Deistler et al., 2022).

We apply our multifidelity approach to TSNPE. First, the high-fidelity density estimator is initialized from the learned network parameters of a low-fidelity density estimator. Then, high-fidelity simulations are generated iteratively from a truncated prior, within the support of the current posterior. We refer to this method as MF-TSNPE (complete description of the algorithm in Appendix M.1).

3.1.3 Acquisition function

To further enhance the efficiency of our sequential algorithm, we explore the use of acquisition functions to supplement our round-wise samples from the TSNPE proposal: we generate simulations for round i with a set of parameters $\boldsymbol{\theta}^{(i)} = \{\boldsymbol{\theta}^{(i)}_{\text{prop}} \cup \boldsymbol{\theta}^{(i)}_{\text{active}}\}$ where $\boldsymbol{\theta}^{(i)}_{\text{prop}}$ are samples from the proposal distribution at round i, and $\boldsymbol{\theta}^{(i)}_{\text{active}}$ are the top $\mathcal{B}$ values according to an acquisition function. We refer to this algorithm as MF-TSNPE-AF (full description in Appendix M.2). Following Järvenpää et al. (2019); Lueckmann et al. (2019), we select an acquisition function that targets the variance of the posterior estimate with respect to the epistemic uncertainty in the learned parameters $\phi|\mathcal{D}$.

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}}{\text{argmax}} \mathbb{V}_{\phi|\mathcal{D}}[q_\phi(\boldsymbol{\theta}|\boldsymbol{x_o})] \tag{2}$$

We realize this as the sample variance across an ensemble of neural density estimators trained independently on the same dataset $\mathcal{D}$, as done in Lueckmann et al. (2019). Note that we use epistemic uncertainty to guide high-fidelity simulation selection within the simulator's domain rather than out-of-distribution samples. For details on the proposal design of MF-TSNPE-AF, see Appendix M.2.

3.1.4 Evaluation metrics

We evaluate the method on observations $\boldsymbol{x_o}$ from the high-fidelity simulator, with parameter values drawn from the prior distribution. This ensured a fair evaluation of how much the low-fidelity simulator helps to infer the posterior distribution given the high-fidelity model. All methods were evaluated for a range of high-fidelity simulation budgets $(50, 10^2, 10^3, 10^4, 10^5)$, on posteriors given the same data set of observations $\boldsymbol{x_o}$.

Known true posterior We evaluate the accuracy of posterior distributions in cases where the ground-truth posterior is known with the Classifier-2-Sample Test (C2ST) and the Maximum Mean Discrepancy (MMD)(Friedman, 2004; Lopez-Paz & Oquab, 2017; Gretton et al., 2012; Lueckmann et al., 2021; Peyré & Cuturi, 2017). C2ST is commonly used in SBI, as it is easy to apply and interpret: a value close to 0.5 means that a classifier cannot effectively distinguish the two distributions, implying the posterior estimate is close to the ground-truth posterior. A value close to 1 means that the classifier can distinguish the distributions very well, indicating a poor posterior estimation. C2ST is rarely applicable in practical SBI settings, since it requires samples from the true posterior (e.g., Sec. 4.1).

Unknown true posterior The average Negative Log probability of the True Parameters (NLTP; $-\mathbb{E}[\log q(\boldsymbol{\theta_o}|\boldsymbol{x_o})]$) has been extensively used in the SBI literature for problems where the true posterior is unknown (Greenberg et al., 2019; Papamakarios & Murray, 2016; Durkan et al., 2020; Hermans et al., 2020). In the limit of a large number of pairs $(\boldsymbol{\theta_o}, \boldsymbol{x_o})$, the average over the log probability of each pair $(\boldsymbol{\theta_o}, \boldsymbol{x_o})$ approaches the expected KL divergence between the estimated and the true posterior (up to a term that is independent of the estimated posterior), as shown in (Lueckmann et al., 2021). In addition, we report the Normalized Root Mean Square Error (NRMSE), which quantifies the deviation of posterior samples from the true parameters on a scale-invariant axis. NRMSE values closer to 0 indicate better predictive performance.

4 Results

We evaluate the performance of our multifidelity approach to NPE and TSNPE on six tasks involving various types of observations (e.g., time series, images, neural spiking). We start with four bench-

marking tasks, followed by two challenging neuroscience problems with computationally expensive simulators and for which no likelihood is available: a multicompartmental neuron model and a neural network model with synaptic plasticity. We also provide a comparison to MF-ABC (Sec. E.1.1, D.3). In Sec. 4.4, we provide a discussion about the effectiveness of transfer learning in MF-NPE.

4.1 BENCHMARKING TASKS

We first evaluated MF-(TS)NPE on four benchmarking tasks: **SIR**, **SLCP**, **OUprocess**, and **Gaussian Blob**. SIR and SLCP are established SBI benchmarks (Lueckmann et al., 2021), OUprocess is a new multifidelity task with tractable likelihood (Kou et al., 2012), and Gaussian Blob is a high-dimensional image task (Lueckmann et al., 2019) (details in Appendix D). These tasks were chosen to systematically investigate various task properties that might impact the performance of transfer learning in a multifidelity setting: differing parameter dimensionality between the low- and high-fidelity models, partly observed dynamics, differing simulator types between the low- and high-fidelity models, and high-dimensional observations. Furthermore, these multifidelity tasks are not trivial in the sense that the low and high-fidelity simulators lead to different posteriors (Appendix I). Note that we do not evaluate the total cost of low- and high-fidelity simulations in these tasks, but defer this analysis to the two complex neuroscience tasks (Appendix J).

To evaluate MF-NPE, we compared the estimated densities to the respective reference posterior, estimated from the exact likelihood with Rejection Sampling (Martino et al., 2018) (OU process; closed-form of the likelihood in Sec. D.1), and using Sampling and Importance Resampling (RUBIN, 1988) to obtain a set of 10k proposal samples (SLCP, SIR), similar to Lueckmann et al. (2021). We quantified the performance with C2ST and MMD over 10 observations (30 observations for the OU process) and 10 network initializations per observation. GaussianBlob uses a CNN embedding and was evaluated with NRMSE and NLTP since no closed-form likelihood is available (Fig. 11).

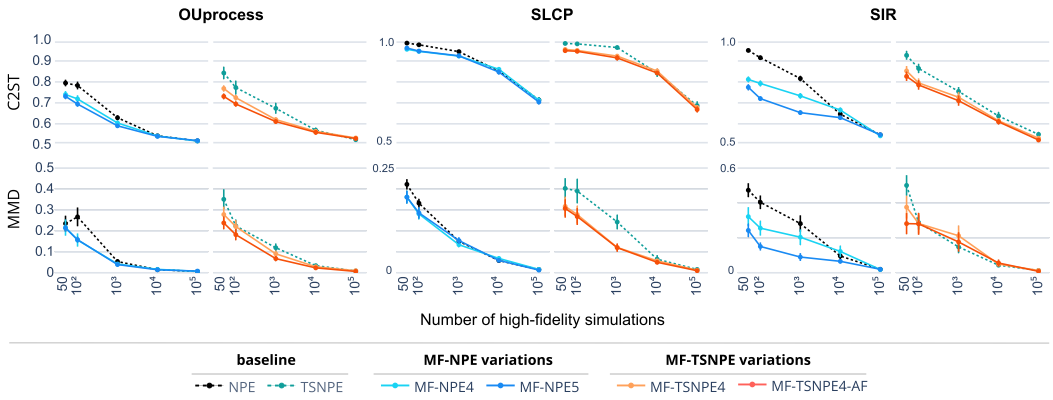


Figure 2: C2ST and MMD averaged over 10 network initializations with means and 95% confidence intervals. MF-NPE4 and MF-NPE5 are pretrained on 10^4 and 10^5 low-fidelity simulations, respectively. Results for the GaussianBlob task in Fig. 11; variations on the OU task and comparisons to MF-ABC in Fig. 8.

Across four benchmarking tasks, we observed a consistent performance increase with MF-NPE compared to NPE, and MF-TSNPE(-AF) compared to TSNPE, especially in low simulation budgets from the high-fidelity model (50-10^3 simulations) (Fig. 2; Gaussian Blob in Fig. 11). In addition, we found that having a higher number of low-fidelity samples improved performance, reinforcing that low-fidelity simulations were indeed advantageous for pre-training the neural density estimator for the downstream task. Note that for the OU and SLCP tasks, we did not observe a substantial increase in MF-NPE performance between the settings with 10^4 and 10^5 low-fidelity samples, suggesting an upper bound regarding pre-training efficacy. We also compared MF-NPE with MF-ABC, an ABC-based method for multifidelity SBI (Prescott & Baker, 2020), and observed that MF-NPE has a substantially higher performance (Appendix E.1.1). This is consistent with previous findings indicating the superior performance of NPE with respect to rejection ABC and SMC-ABC, where it is not uncommon to require orders of magnitude more simulations to obtain reliable

posterior approximations (Lueckmann et al., 2021; Frazier et al., 2024). However, a more extensive hyperparameter search could potentially lead to substantial improvements in MF-ABC performance.

As described in Sec. 3, we enhanced the sequential algorithm TSNPE (Deistler et al., 2022) with a first round of MF-NPE, and designated this approach as MF-TSNPE. We found that MF-TSNPE (details in Appendix M.1) performs better than TSNPE, especially in regimes with a low budget of high-fidelity simulations. Compared to MF-TSNPE, MF-TSNPE-AF improved inference in the OU process, but did not show significant improvements in the SLCP and SIR tasks.

Finally, we assessed the contribution of transfer learning to the overall performance in a setting where the low- and high-fidelity models have a different number of parameters, in the context of the OUprocess task (Appendix D.3). We expected that adding parameters to the high-fidelity model that are absent in the low-fidelity model would increase the inference complexity for MF-NPE, and indeed observed a performance decrease in MF-NPE, although MF-NPE still performed better than NPE and MF-ABC (see Appendix D.3). We note that MF-NPE also outperformed NPE when the low-fidelity model had more parameters than the high-fidelity model (see Appendix D.4). Overall, the results suggest that MF-NPE and MF-TSNPE can yield substantial performance gains compared to NPE, TSNPE, and MF-ABC.

4.2 Multicompartmental neuron model

The voltage response of a morphologically-detailed neuron to an input current is typically modeled with a multicompartment model wherein the voltage dynamics of each compartment are based on the Hodgkin-Huxley equations (Hodgkin & Huxley, 1952). The higher the number of compartments of the model, the more accurate the model is, but the higher the simulation cost.

In this task, we aimed to infer the densities of ion channels $\bar{g}_{Na}$ and $\bar{g}_K$ on a morphologically-detailed model of a thick-tufted layer 5 pyramidal cell (L5PC) containing 8 compartments per branch (Fig. 3A) (Van Geit et al., 2016). We injected in the first neuron compartment a noisy 100 ms step current with mean $I_m = 0.3$ nA: $I_e = I_m + \epsilon, \epsilon \sim \mathcal{N}(0, 0.01)$. The voltage response of the neuron was recorded over 120 ms, with a simulation step size of 0.025 ms and 10 ms margin before and after the current injection. We defined the high-fidelity model to have 8 compartments per branch and the low-fidelity model to have 1 compartment per branch, and both the high and low-fidelity models had the same injected current and ion channel types.

To simulate the neuron models, we used Jaxley, a Python toolbox for efficiently simulating multicompartment single neurons with biophysical detail (Deistler et al., 2024). In this setting, the simulation time for the high-fidelity model is approximately 4 times higher than that of the low-fidelity model. We characterized the neural response with four summary statistics that have been commonly used when fitting biophysical models of single neurons to empirical data: spike count, mean resting potential, standard deviation of the resting potential, and voltage mean (Gonçalves et al., 2020; Gao et al., 2023). Performances were evaluated with NLTP and NRMSE on 10^3 pairs of θ_o and respective simulation outputs x_o, averaged over 10 random network initializations (Sec. 3.1.4).

MF-(TS)NPE showed higher performance than NPE, in particular with larger low-fidelity simulation budgets (Fig. 3B; Fig. F.1), despite the right-skewed posterior distribution of the low-fidelity model (Fig. 21). Furthermore, MF-NPE posterior predictives closely matched the empirical data, in contrast with NPE, even when NPE was trained on a higher number of high-fidelity simulations (Appendix F). In addition, MF-(TS)NPE achieved comparable performance with a total computational cost 4.44 ± 0.06 times lower than standard NPE (Appendix J). Finally, TARP and simulation-based calibration tests suggest that both MF-NPE and NPE estimates were relatively well calibrated (Fig. 3C) (Talts et al., 2020; Lemos et al., 2023).

(A)-MF-TSNPE pre-trained on 10^4 low-fidelity samples outperforms MF-NPE trained on 10^5 samples. However, MF-TSNPE-AF performance comes at the cost of training time due to the use of an ensemble of density estimators (Appendix J). This additional training burden is only justified when the simulation cost is substantially higher than the training cost.

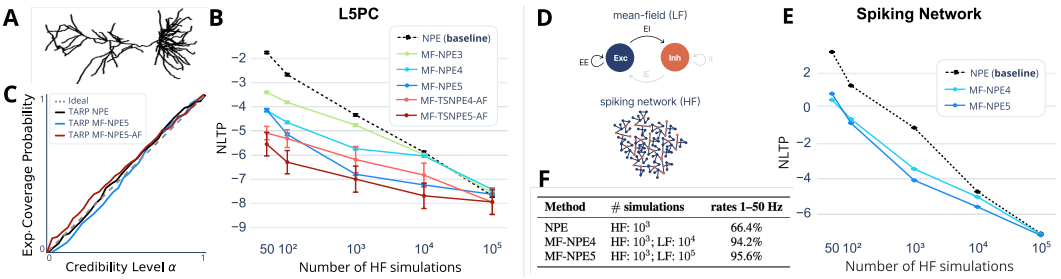


Figure 3: **(A)** Thick-tufted layer 5 pyramidal cell from the neocortex. **(B)** Performance evaluation with NLTP (same naming convention as in Fig. 2). Amortized methods are averaged over 10 network initializations; non-amortized trained once per 100 observations. Similar results were obtained with NRMSE (Appendix F.1). MF-NPE, and especially its sequential variants, are orders of magnitude more simulation-efficient than NPE. **(C)** TARP posterior calibration check shows that NPE and MF-NPE trained on 10^3 high-fidelity samples are well-calibrated (Lemos et al., 2023). Simulation-based calibration, posterior samples, and predictives are in Appendix F. **(D)** Schematic of the low and high-fidelity models of a spiking network. **(E)** Performance of NPE and MF-NPE evaluated on 10000 true observations with NLTP: averages over 10 network initializations, and 95% confidence intervals. **(F)** Proportion of posterior samples within the target firing rate bounds. MF-NPE produces a higher fraction of parameter sets within the bounds than NPE.

4.3 Recurrent spiking network

Finally, we applied MF-NPE to a challenging and timely problem in neuroscience: the inference of synaptic plasticity rules that endow large spiking neural networks with dynamics reminiscent of experimental data. This problem has been recently tackled with an SBI method (filter simulation-based inference, fSBI) that progressively narrows down the search space of parameters given different sets of summary statistics (Confavreux et al., 2023). fSBI was successful in obtaining manifolds of plasticity rules that ensure plausible network activity, but the compute requirements were reported to be very large. Here, we aim to test whether this problem can be efficiently tackled with MF-NPE.

The high-fidelity simulator consisted of a recurrent network of 4096 excitatory (E) and 1024 inhibitory (I) leaky integrate-and-fire neurons connected with conductance-based synapses (Fig. 3D). Each synapse type in this network (E-to-E, E-to-I, I-to-E, I-to-I) was plastic with an unsupervised local learning rule. For each synapse type, 6 parameters governed how the recent pre- and post-synaptic activity were used to update the synapse, for a total of 24 free parameters across all 4 synapse types (Confavreux et al., 2023). The networks were simulated using Auryn, a C++ simulator (Zenke & Gerstner, 2014) (details in Appendix G).

Mean-field theory can be applied to the dynamical system above to obtain the steady-state activities of the excitatory and inhibitory populations as a function of the parameters of the plasticity rules embedded in the network. Though such analysis is widely performed in the field (Vogels et al., 2011; Confavreux et al., 2023; Gerstner et al., 2014), it has never been used as a low-fidelity model to help with the inference of the high-fidelity model parameters. Since there are no dynamics to simulate with the mean-field model, the simulation was almost instantaneous, while the high-fidelity model took approximately 5 minutes to generate a single 2-minute long simulation on a single CPU.

Summary statistics of the low- and high-fidelity models were the average firing rates of the excitatory and inhibitory neurons at steady state (after 2 minutes of simulation in the high-fidelity model). Plastic networks were considered plausible if the firing rates were between 1 and 50Hz (Dayan & Abbott, 2001; Confavreux et al., 2023).

In this task, the low-fidelity model focuses solely on the E-to-E and E-to-I rules from the high-fidelity model, thereby having 12 out of the 24 parameters of the high-fidelity model. This setup allows us to demonstrate the performance of MF-NPE on problems with different parameter spaces, highlighting MF-NPE's flexibility and advantages. We found that MF-NPE has better performance than NPE in terms of NLTP (Fig. 3E), although we observed a diminishing performance gain with increasing discrepancy between the number of parameters of the low- and high-fidelity models (see Appendix G.3). Furthermore, MF-NPE leads to an increase of almost 30% in the proportion of

posterior samples within the target firing rate bounds (Fig. 3F), reinforcing that MF-NPE is a practical and effective method for SBI of costly real-world simulators.

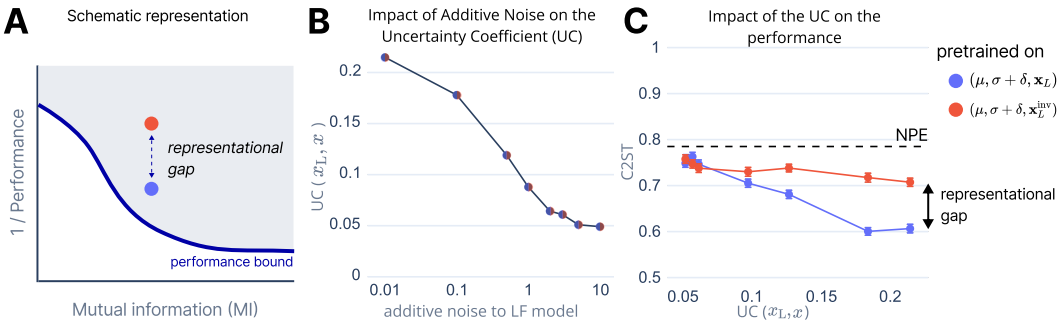


Figure 4: **(A)** Schematic figure representing lower bound on transfer error (1/MF-NPE performance) as a function of mutual information between the low- and high-fidelity models, given a fixed simulation budget. **(B)** Uncertainty coefficient monotonically decreases with noise parameter δ and is invariant to data inversion. **(C)** Empirical results with MF-NPE support the hypothesis that transfer performance is dependent on both mutual information and representational coherence. Note that NPE (with the same high-fidelity simulation budget of 10^2) has similar performance as MF-NPE in the case where the low- and high-fidelity models have low mutual information.

4.4 When does pre-training help?

In previous sections, we demonstrated that MF-NPE can significantly reduce the number of high-fidelity simulations required to accurately approximate the high-fidelity posterior by leveraging pre-training on low-fidelity simulations. This naturally leads to several key questions: Which characteristics of low-fidelity simulators enable effective transfer learning? Under what conditions can pre-training reliably enhance simulation efficiency?

Providing theoretical guarantees for these questions necessitates a formal characterization of convergence rates in NPE with transfer learning. Although recent works have begun addressing these challenges in NPE (Frazier et al., 2024), current theoretical frameworks of transfer learning (Tahir et al., 2024; Yun et al., 2020; Tripuraneni et al., 2020; Lampinen & Ganguli, 2018), rely on simplifying assumptions (e.g., linear networks) that do not fully capture the complexities of MF-NPE. Given this limitation, we instead empirically explored the conditions in which low-fidelity pre-training facilitates effective transfer learning. To do this, we evaluate MF-NPE where the low- and high-fidelity simulators are related by systematic perturbations (Fig. 4).

We hypothesize that the effectiveness of pre-training is associated with two primary factors:

1. **Mutual information** between the low- and high-fidelity simulators.
2. **Representational coherence**, i.e., similarity in how task-relevant information is encoded.

To isolate the effects of these factors, we construct controlled variants of the OU2 process in which the low-fidelity simulator differs from the high-fidelity one through two distinct transformations. In the baseline setup, the simulators generate observations according to

$$x \sim p(x \mid \mu, \sigma), \qquad x_{\mathrm{L}} \sim p(x \mid \mu, \sigma + \delta),$$

where the perturbation δ increases the noise of the low-fidelity simulator and therefore reduces $\mathbb{I}[x; x_{\mathrm{L}}]$ monotonically as δ grows.

Second, to independently manipulate representational coherence, we apply an invertible coordinate-reversal transformation $x_{\mathrm{L}}^{\mathrm{inv}} = T(x_{\mathrm{L}})$, implemented via an anti-diagonal permutation matrix that reverses the ordering of the output dimensions. Because T is invertible, the mutual information between the two simulators is unchanged:

$$\mathbb{I}[\boldsymbol{x}; \boldsymbol{x}_{\mathrm{L}}^{\mathrm{inv}}] \;=\; \mathbb{I}[\boldsymbol{x}; \boldsymbol{x}_{\mathrm{L}}] \;=\; \mathbb{H}[\boldsymbol{x}] + \mathbb{H}[\boldsymbol{x}_{\mathrm{L}}] - \mathbb{H}[\boldsymbol{x}, \boldsymbol{x}_{\mathrm{L}}].$$

Thus, while $\mathbb{I}[x; x_{\mathrm{L}}]$ decreases monotonically with the noise scale δ, the inversion leaves the information content unchanged while disrupting representational coherence. Figure 4 illustrates how

each manipulation affects the uncertainty coefficient (Figure 4B), which we estimate empirically using MINE (Belghazi et al., 2018), and MF-NPE performance under a fixed simulation budget of 10^4 low-fidelity and 10^2 high-fidelity simulations (Figure 4C).

In agreement with our hypothesis, our results suggest that the effectiveness of MF-NPE depends on both the mutual information and the representational coherence between low- and high-fidelity simulators (Fig. 4C). Specifically, mutual information is necessary for effective transfer learning but not sufficient: perturbations that preserve information (e.g., invertible transformations) can still substantially impair transfer performance. Effective pre-training strategies should therefore prioritize low-fidelity simulators that are both highly informative and representationally aligned with the high-fidelity model.

5 Discussion

We proposed a new method for simulation-based inference that leverages low-fidelity models to efficiently infer the parameters of costly high-fidelity models. By incorporating transfer learning and multifidelity approaches, MF-NPE substantially reduces the simulation budget required for accurate posterior inference. This addresses a pervasive challenge across scientific domains: the high computational cost of simulating complex high-fidelity models and linking them to empirical data. Our empirical results demonstrate MF-NPE's competitive performance in SBI across statistical benchmarks and real-world applications, as compared to a standard method such as NPE.

Limitations Despite MF-NPE's advantages, the method comes with some challenges. First, the effectiveness of MF-NPE relies on the similarity between the low-fidelity and high-fidelity models. Fortunately, in many situations, domain experts will know beforehand whether low-fidelity models are poor approximations of high-fidelity models. Second, MF-NPE and MF-TSNPE inherit the limitations of NPE and TSNPE, respectively, in particular regarding the scalability of simulation-based inference to high-dimensional parameter spaces. How to balance exploration of high-dimensional parameter spaces and computational cost in a simulation-based inference setting remains a topic of active research. Third, MF-TSNPE-AF requires the training of an ensemble of density estimators, which leads to substantial computational costs in training and hyperparameter tuning. This method should therefore only be preferred in cases where the cost incurred in simulations outweighs the training cost. We estimate this to be the case for the tasks with the multicompartment neuron model and the spiking network model, for which the cost of one simulation and the training of one density estimator are comparable in certain settings (e.g., on the order of minutes, for a network trained on 10^3 samples).

Future work We identify three promising research directions for multifidelity simulation-based inference. First, we expect the scalability and expressivity of MF-NPE could be improved by utilizing the same approaches of multifidelity and transfer learning presented here with neural density estimators other than normalizing flows, such as diffusion models (Gloeckler et al., 2024). Second, we assumed a negligible cost for low-fidelity simulations, and future work should address how to optimally allocate low- and high-fidelity simulations under a fixed computational budget. Third, similar to past efforts in developing a benchmark for simulation-based inference, it will be beneficial for the SBI community to develop a benchmark for multifidelity problems, with new tasks, algorithms and evaluation metrics. This will promote rigorous and reproducible research and catalyze new developments in multifidelity SBI, and in SBI more generally. Our work and codebase are a step in this direction.

Conclusion Overall, MF-(TS)NPE is a method for simulation-based inference that leverages low-fidelity models and transfer learning to infer the parameters of costly high-fidelity models, thus providing an effective balance between computational cost and inference accuracy.

6 Reproducibility Statement

The training and simulation costs for all tasks and SBI methods, as well as a detailed description of the experimental setup, are described in Appendices C.1 and J. The corresponding code and data are available in an **anonymous GitHub repository** (link) and will be publicly released upon publication.

REFERENCES

Elias Barbers, Friedrich Emanuel Hust, Felix Emil Arthur Hildenbrand, Fabian Frie, Katharina Lilith Quade, Stephan Bihn, Dirk Uwe Sauer, and Philipp Dechent. Exploring the effects of cell-to-cell variability on battery aging through stochastic simulation techniques. *Journal of Energy Storage*, 84:110851, April 2024. ISSN 2352-152X. doi: 10.1016/j.est.2024.110851. URL https://www.sciencedirect.com/science/article/pii/S2352152X24004353.

J. Behrens and F. Dias. New computational methods in tsunami science. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 373(2053):20140382, October 2015. doi: 10.1098/rsta.2014.0382. URL https://royalsocietypublishing.org/doi/full/10.1098/rsta.2014.0382.

Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeswar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and R Devon Hjelm. Mine: mutual information neural estimation. *arXiv preprint arXiv:1801.04062*, 2018.

Jan Boelts, Michael Deistler, Manuel Gloeckler, Alvaro Tejero-Cantero, Jan-Matthis Lueckmann, Guy Moss, Peter Steinbach, Thomas Moreau, Fabio Muratore, Julia Linhart, Conor Durkan, Julius Vetter, Benjamin Kurt Miller, Maternus Herold, Abolfazl Ziaeemehr, Matthijs Pals, Theo Gruner, Sebastian Bischoff, Nastya Krouglova, Richard Gao, Janne K. Lappalainen, Bálint Mucsányi, Felix Pei, Auguste Schulz, Zinovia Stefanidi, Pedro Rodrigues, Cornelius Schröder, Faried Abu Zaid, Jonas Beck, Jaivardhan Kapoor, David S. Greenberg, Pedro J. Gonçalves, and Jakob H. Macke. sbi reloaded: a toolkit for simulation-based inference workflows, November 2024. URL http://arxiv.org/abs/2411.17337.

Simon Carter and Helmut H. Strey. Parameter estimation from an Ornstein-Uhlenbeck process with measurement noise, August 2023. URL http://arxiv.org/abs/2305.13498.

Thomas A. Catanach, Huy D. Vo, and Brian Munsky. Bayesian inference of stochastic reaction networks using multifidelity sequential tempered Markov Chain Monte Carlo. *International Journal for Uncertainty Quantification*, 10(6):515–542, 2020. ISSN 2152-5080. doi: 10.1615/int.j.uncertaintyquantification.2020033241.

Basile Confavreux, Poornima Ramesh, Pedro J. Goncalves, Jakob H. Macke, and Tim Vogels. Meta-learning families of plasticity rules in recurrent spiking networks using simulation-based inference. *Advances in Neural Information Processing Systems*, 36:13545–13558, December 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/2bdc2267c3d7d01523e2e17ac0a754f3-Abstract-Conference.html.

Kyle Cranmer, Johann Brehmer, and Gilles Louppe. The frontier of simulation-based inference. *Proceedings of the National Academy of Sciences*, 117(48):30055–30062, December 2020. doi: 10.1073/pnas.1912789117. URL https://www.pnas.org/doi/10.1073/pnas.1912789117.

Peter Dayan and L. F. Abbott. *Theoretical neuroscience: computational and mathematical modeling of neural systems*. Computational neuroscience. Massachusetts Institute of Technology Press, Cambridge, Mass, 2001. ISBN 978-0-262-04199-7.

Michael Deistler, Pedro J. Goncalves, and Jakob H. Macke. Truncated proposals for scalable and hassle-free simulation-based inference. *Advances in Neural Information Processing Systems*, 35:23135–23149, December 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/9278abf072b58caf21d48dd670b4c721-Abstract-Conference.html.

Michael Deistler, Kyra L. Kadhim, Matthijs Pals, Jonas Beck, Ziwei Huang, Manuel Gloeckler, Janne K. Lappalainen, Cornelius Schröder, Philipp Berens, Pedro J. Gonçalves, and Jakob H. Macke. Differentiable simulation enables large-scale training of detailed biophysical models of neural dynamics, August 2024.

Conor Durkan, Artur Bekasov, Iain Murray, and George Papamakarios. Neural Spline Flows. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/hash/7ac71d433f282034e088473244df8c02-Abstract.html.

Conor Durkan, Iain Murray, and George Papamakarios. On Contrastive Learning for Likelihood-free Inference. In *Proceedings of the 37th International Conference on Machine Learning*, pp. 2771–2781. PMLR, November 2020. URL https://proceedings.mlr.press/v119/durkan20a.html.

Joel Dyer, Patrick W Cannon, and Sebastian M Schmon. Amortised likelihood-free inference for expensive time-series simulators with signatured ratio estimation. In *International Conference on Artificial Intelligence and Statistics*, pp. 11131–11144. PMLR, 2022.

Lasse Elsemüller, Valentin Pratz, Mischa von Krause, Andreas Voss, Paul-Christian Bürkner, and Stefan T. Radev. Does Unsupervised Domain Adaptation Improve the Robustness of Amortized Bayesian Inference? A Systematic Evaluation, May 2025. arXiv:2502.04949 [stat].

Yusuf Falola, Siddharth Misra, and Andres Calvo Nunez. Rapid High-Fidelity Forecasting for Geological Carbon Storage Using Neural Operator and Transfer Learning. In *ADIPEC*, Abu Dhabi, UAE, October 2023. OnePetro. doi: 10.2118/216135-MS.

David T. Frazier, Ryan Kelly, Christopher Drovandi, and David J. Warne. The Statistical Accuracy of Neural Posterior and Likelihood Estimation, November 2024. URL http://arxiv.org/abs/2411.12068. arXiv:2411.12068 [stat].

J Friedman. On Multivariate Goodness-of-Fit and Two-Sample Testing. Technical Report SLAC-PUB-10325, 826696, Stanford, January 2004. URL http://www.osti.gov/servlets/purl/826696/.

Richard Gao, Michael Deistler, and Jakob H. Macke. Generalized Bayesian Inference for Scientific Simulators via Amortized Cost Estimation, November 2023. URL http://arxiv.org/abs/2305.15208.

Wulfram Gerstner and Werner M. Kistler. Mathematical formulations of Hebbian learning. *Biological Cybernetics*, 87(5):404–415, December 2002. ISSN 1432-0770. doi: 10.1007/s00422-002-0353-y. URL https://doi.org/10.1007/s00422-002-0353-y.

Wulfram Gerstner, Werner M. Kistler, Richard Naud, and Liam Paninski. *Neuronal Dynamics: From Single Neurons to Networks and Models of Cognition*. Cambridge University Press, Cambridge, 2014. ISBN 978-1-107-06083-8. doi: 10.1017/CBO9781107447615. URL https://www.cambridge.org/core/books/neuronal-dynamics/7537509004673376559 6191E23B2959D.

Michael B. Giles. Multilevel Monte Carlo Path Simulation. *Operations Research*, 56(3):607–617, June 2008. ISSN 0030-364X. doi: 10.1287/opre.1070.0496. URL https://pubsonline.informs.org/doi/abs/10.1287/opre.1070.0496.

Manuel Gloeckler, Michael Deistler, Christian Weilbach, Frank Wood, and Jakob H. Macke. All-in-one simulation-based inference, May 2024. URL http://arxiv.org/abs/2404.09636.

Manuel Gloeckler, Shoji Toyota, Kenji Fukumizu, and Jakob H. Macke. Compositional simulation-based inference for time series, March 2025. arXiv:2411.02728 [cs].

Pedro J Gonçalves, Jan-Matthis Lueckmann, Michael Deistler, Marcel Nonnenmacher, Kaan Öcal, Giacomo Bassetto, Chaitanya Chintaluri, William F Podlaski, Sara A Haddad, Tim P Vogels, David S Greenberg, and Jakob H Macke. Training deep neural density estimators to identify mechanistic models of neural dynamics. *eLife*, 9:e56261, September 2020. ISSN 2050-084X. doi: 10.7554/eLife.56261. URL https://doi.org/10.7554/eLife.56261.

David Greenberg, Marcel Nonnenmacher, and Jakob Macke. Automatic Posterior Transformation for Likelihood-Free Inference. In *Proceedings of the 36th International Conference on Machine Learning*, pp. 2404–2414. PMLR, May 2019. URL https://proceedings.mlr.press/v97/greenberg19a.html.

Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. A Kernel Two-Sample Test. *Journal of Machine Learning Research*, 13(25):723–773, 2012. ISSN 1533-7928. URL http://jmlr.org/papers/v13/gretton12a.html.

Sam Griesemer, Defu Cao, Zijun Cui, Carolina Osorio, and Yan Liu. Active Sequential Posterior Estimation for Sample-Efficient Simulation-Based Inference. In *Neural Information Processing Systems*, November 2024. URL https://openreview.net/forum?id=fkuseU0nJs&noteId=AzIKRHiQD4.

Michael U. Gutmann and Jukka Corander. Bayesian Optimization for Likelihood-Free Inference of Simulator-Based Statistical Models. *Journal of Machine Learning Research*, 17(125):1–47, 2016. ISSN 1533-7928.

Ehsan Haghighat, Maziar Raissi, Adrian Moure, Hector Gomez, and Ruben Juanes. A physics-informed deep learning framework for inversion and surrogate modeling in solid mechanics. *Computer Methods in Applied Mechanics and Engineering*, 379:113741, June 2021. ISSN 0045-7825. doi: 10.1016/j.cma.2021.113741. URL https://www.sciencedirect.com/science/article/pii/S0045782521000773.

Zhong-Hua Han, Stefan Görtz, and Ralf Zimmermann. Improving variable-fidelity surrogate modeling via gradient-enhanced kriging and a generalized hybrid bridge function. *Aerospace Science and Technology*, 25(1):177–189, March 2013. ISSN 1270-9638. doi: 10.1016/j.ast.2012.01.006. URL https://www.sciencedirect.com/science/article/pii/S127096381200017X.

Isaac M. Held. The Gap between Simulation and Understanding in Climate Modeling. *ametsoc*, November 2005. doi: 10.1175/BAMS-86-11-1609. URL https://journals.ametsoc.org/view/journals/bams/86/11/bams-86-11-1609.xml.

Joeri Hermans, Volodimir Begy, and Gilles Louppe. Likelihood-free MCMC with Amortized Approximate Ratio Estimators. In *Proceedings of the 37th International Conference on Machine Learning*, pp. 4239–4248. PMLR, November 2020. URL https://proceedings.mlr.press/v119/hermans20a.html.

Yuga Hikida, Ayush Bharti, Niall Jeffrey, and François-Xavier Briol. Multilevel neural simulation-based inference, August 2025. arXiv:2506.06087 [stat] version: 2.

A. L. Hodgkin and A. F. Huxley. A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of Physiology*, 117(4):500–544, August 1952. ISSN 0022-3751. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1392413/.

Vladimír Holý and Petra Tomanová. Estimation of Ornstein-Uhlenbeck Process Using Ultra-High-Frequency Data with Application to Intraday Pairs Trading Strategy, July 2022. arXiv:1811.09312 [q-fin].

Mathias Hoppe, Ola Embreus, and Tünde Fülöp. DREAM: A fluid-kinetic framework for tokamak disruption runaway electron simulations. *Computer Physics Communications*, 268:108098, November 2021. ISSN 0010-4655. doi: 10.1016/j.cpc.2021.108098. URL https://www.sciencedirect.com/science/article/pii/S0010465521002101.

Marko Järvenpää, Michael U. Gutmann, Arijus Pleska, Aki Vehtari, and Pekka Marttinen. Efficient Acquisition Rules for Model-Based Approximate Bayesian Computation. *Bayesian Analysis*, 14(2):595–622, June 2019. ISSN 1936-0975, 1931-6690. doi: 10.1214/18-BA1121.

Kirthevasan Kandasamy, Gautam Dasarathy, Jeff Schneider, and Barnabás Póczos. Multi-fidelity Bayesian Optimisation with Continuous Approximations. In *Proceedings of the 34th International Conference on Machine Learning*, pp. 1799–1808. PMLR, July 2017. URL https://proceedings.mlr.press/v70/kandasamy17a.html.

Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization, January 2017. URL http://arxiv.org/abs/1412.6980. arXiv:1412.6980 [cs].

Supeng Kou, Benjamin Olding, Martin Lysy, and Jun Liu. A Multiresolution Method for Parameter Estimation of Diffusion Processes. *Journal of The American Statistical Association - J AMER STATIST ASSN*, 107:4, December 2012. doi: 10.1080/01621459.2012.720899.

Andrew K Lampinen and Surya Ganguli. An analytic theory of generalization dynamics and transfer learning in deep linear networks. *arXiv preprint arXiv:1809.10374*, 2018.

Diane Larsen-Freeman. Transfer of Learning Transformed. *Language Learning*, 63(s1):107–129, 2013. ISSN 1467-9922. doi: 10.1111/j.1467-9922.2012.00740.x. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-9922.2012.00740.x.

Pablo Lemos, Adam Coogan, Yashar Hezaveh, and Laurence Perreault-Levasseur. Sampling-Based Accuracy Testing of Posterior Estimators for General Inference, June 2023. URL http://arxiv.org/abs/2302.03026. arXiv:2302.03026 [stat].

David Lopez-Paz and Maxime Oquab. Revisiting Classifier Two-Sample Tests. In *International Conference on Learning Representations*, Toulon, France, 2017. International Conference on Learning Representations. doi: 10.48550/arXiv.1610.06545.

Jan-Matthis Lueckmann, Pedro J Goncalves, Giacomo Bassetto, Kaan Öcal, Marcel Nonnenmacher, and Jakob H Macke. Flexible statistical inference for mechanistic models of neural dynamics. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://papers.nips.cc/paper_files/paper/2017/hash/addfa9b7e234254d26e9c7f2af1005cb-Abstract.html.

Jan-Matthis Lueckmann, Giacomo Bassetto, Theofanis Karaletsos, and Jakob H. Macke. Likelihood-free inference with emulator networks. In *Proceedings of The 1st Symposium on Advances in Approximate Bayesian Inference*, pp. 32–53. PMLR, January 2019. URL https://proceedings.mlr.press/v96/lueckmann19a.html.

Jan-Matthis Lueckmann, Jan Boelts, David Greenberg, Pedro Goncalves, and Jakob Macke. Benchmarking Simulation-Based Inference. In *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, pp. 343–351. PMLR, March 2021. URL https://proceedings.mlr.press/v130/lueckmann21a.html.

Andrew J. Majda and Boris Gershgorin. Quantifying uncertainty in climate change science through empirical information theory. *Proceedings of the National Academy of Sciences of the United States of America*, 107(34):14958–14963, 2010. ISSN 0027-8424. URL https://www.jstor.org/stable/27862175.

Luca Martino, David Luengo, and Joaquín Míguez. Accept–Reject Methods. In Luca Martino, David Luengo, and Joaquín Míguez (eds.), *Independent Random Sampling Methods*, pp. 65–113. Springer International Publishing, Cham, 2018. ISBN 978-3-319-72634-2. doi: 10.1007/978-3-319-72634-2_3. URL https://doi.org/10.1007/978-3-319-72634-2_3.

Aimee Maurais, Terrence Alsup, Benjamin Peherstorfer, and Youssef Marzouk. Multifidelity Covariance Estimation via Regression on the Manifold of Symmetric Positive Definite Matrices. *SIAM Journal on Scientific Computing*, 2023. doi: 10.48550/ARXIV.2307.12438. URL https://arxiv.org/abs/2307.12438.

Benjamin Kurt Miller, Alex Cole, Patrick Forré, Gilles Louppe, and Christoph Weniger. Truncated marginal neural ratio estimation. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, NIPS '21, pp. 129–143, Red Hook, NY, USA, December 2021. Curran Associates Inc. ISBN 978-1-7138-4539-3.

Barry L. Nelson and Linda Pei. *Foundations and Methods of Stochastic Simulation: A First Course*, volume 316 of *International Series in Operations Research & Management Science*. Springer International Publishing, Cham, 2021. ISBN 978-3-030-86193-3 978-3-030-86194-0. doi: 10.1007/978-3-030-86194-0. URL https://link.springer.com/10.1007/978-3-030-86194-0.

Fabio Nobile and Francesco Tesei. A Multi Level Monte Carlo method with control variate for elliptic PDEs with log-normal coefficients. *Stochastic Partial Differential Equations: Analysis and Computations*, 3(3):398–444, September 2015. ISSN 2194-041X. doi: 10.1007/s40072-015-0055-9. URL https://doi.org/10.1007/s40072-015-0055-9.

Santisudha Panigrahi, Anuja Nanda, and Tripti Swarnkar. A Survey on Transfer Learning. In Debahuti Mishra, Rajkumar Buyya, Prasant Mohapatra, and Srikanta Patnaik (eds.), *Intelligent and Cloud Computing*, pp. 781–789, Singapore, 2021. Springer. ISBN 9789811559716. doi: 10.1007/978-981-15-5971-6_83.

George Papamakarios and Iain Murray. Fast \epsilon -free Inference of Simulation Models with Bayesian Conditional Density Estimation. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL https://proceedings.neurips.cc/paper/2016/hash/6aca97005c68f1206823815f66102863-Abstract.html.

George Papamakarios, David Sterratt, and Iain Murray. Sequential Neural Likelihood: Fast Likelihood-free Inference with Autoregressive Flows. In *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, pp. 837–848. PMLR, April 2019. URL https://proceedings.mlr.press/v89/papamakarios19a.html.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/hash/bdbca288fee7f92f2bfa9f7012727740-Abstract.html.

Benjamin Peherstorfer, Karen Willcox, and Max Gunzburger. Optimal Model Management for Multifidelity Monte Carlo Estimation. *SIAM Journal on Scientific Computing*, 38(5):A3163–A3194, January 2016. ISSN 1064-8275. doi: 10.1137/15M1046472. URL https://epubs.siam.org/doi/abs/10.1137/15M1046472.

Benjamin Peherstorfer, Karen Willcox, and Max Gunzburger. Survey of Multifidelity Methods in Uncertainty Propagation, Inference, and Optimization. *SIAM Review*, 60(3):550–591, January 2018. ISSN 0036-1445. doi: 10.1137/16M1082469. URL https://epubs.siam.org/doi/10.1137/16M1082469.

Gabriel Peyré and Marco Cuturi. Computational Optimal Transport. *Working Papers*, October 2017. Number: 2017-86 Publisher: Center for Research in Economics and Statistics.

Jonathan Pillow and James Scott. Fully Bayesian inference for neural models with negative-binomial spiking. In *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012.

Lutz Prechelt. Early Stopping - But When? In *Lecture Notes in Computer Science, vol 7700*. Montavon, G., Orr, G.B., Müller, KR. (eds) Neural Networks: Tricks of the Trade. Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, January 2002. doi: 10.1007/3-540-49430-8_3. URL https://link.springer.com/chapter/10.1007/3-540-49430-8_3.

Thomas P. Prescott and Ruth E. Baker. Multifidelity Approximate Bayesian Computation. *SIAM/ASA Journal on Uncertainty Quantification*, 8(1):114–138, January 2020. doi: 10.1137/18M1229742.

Thomas P. Prescott and Ruth E. Baker. Multifidelity Approximate Bayesian Computation with Sequential Monte Carlo Parameter Sampling. *SIAM/ASA Journal on Uncertainty Quantification*, 9 (2):788–817, January 2021. doi: 10.1137/20M1316160. URL https://epubs.siam.org/doi/abs/10.1137/20M1316160.

Thomas P. Prescott, David J. Warne, and Ruth E. Baker. Efficient multifidelity likelihood-free Bayesian inference with adaptive computational resource allocation. *Journal of Computational Physics*, 496:112577, January 2024. ISSN 0021-9991. doi: 10.1016/j.jcp.2023.112577.

J K Pritchard, M T Seielstad, A Perez-Lezaun, and M W Feldman. Population growth of human Y chromosomes: a study of Y chromosome microsatellites. *Molecular Biology and Evolution*, 16 (12):1791–1798, December 1999. ISSN 0737-4038. doi: 10.1093/oxfordjournals.molbev.a026091. URL https://doi.org/10.1093/oxfordjournals.molbev.a026091.

Wilfrid Rall. *The Theoretical Foundation of Dendritic Function: Selected Papers of Wilfrid Rall with Commentaries*. MIT Press, 1995. ISBN 978-0-262-19356-6. Google-Books-ID: Nx5fb82827oC.

Poornima Ramesh, Jan-Matthis Lueckmann, Jan Boelts, Alvaro Tejero-Cantero, David S. Greenberg, Pedro J. Goncalves, and Jakob H. Macke. GATSBI: Generative Adversarial Training for Simulation-Based Inference. In *International Conference on Learning Representations*, October 2021. URL https://openreview.net/forum?id=kR1hC6j48Tp.

Francois Roset. Zuko - Normalizing flows in PyTorch, November 2024. URL https://github.com/probabilists/zuko.

DB RUBIN. Using the SIR algorithm to simulate posterior distributions. *Bayesian statistics 3. Proceedings of the third Valencia international meeting, 1-5 June 1987*, pp. 395–402, 1988. Publisher: Clarendon Press.

Alex A. Saoulis, Davide Piras, Niall Jeffrey, Alessio Spurio Mancini, Ana M. G. Ferreira, and Benjamin Joachimi. Transfer learning for multifidelity simulation-based inference in cosmology, May 2025. arXiv:2505.21215 [astro-ph].

Marvin Schmitt, Desi R. Ivanova, Daniel Habermann, Ullrich Köthe, Paul-Christian Bürkner, and Stefan T. Radev. Leveraging Self-Consistency for Data-Efficient Amortized Bayesian Inference, July 2024a. arXiv:2310.04395 [cs].

Marvin Schmitt, Valentin Pratz, Ullrich Köthe, Paul-Christian Bürkner, and Stefan T. Radev. Consistency Models for Scalable and Fast Simulation-Based Inference, November 2024b. arXiv:2312.05440 [cs].

Jialin Song, Yuxin Chen, and Yisong Yue. A General Framework for Multi-fidelity Bayesian Optimization with Gaussian Processes. In *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, pp. 3158–3167. PMLR, April 2019. ISSN: 2640-3498.

Javan Tahir, Surya Ganguli, and Grant M. Rotskoff. Features are fate: a theory of transfer learning in high-dimensional regression, October 2024. URL http://arxiv.org/abs/2410.08194.

Sean Talts, Michael Betancourt, Daniel Simpson, Aki Vehtari, and Andrew Gelman. Validating Bayesian Inference Algorithms with Simulation-Based Calibration, October 2020. URL http://arxiv.org/abs/1804.06788.

S. Tavaré, D. J. Balding, R. C. Griffiths, and P. Donnelly. Inferring coalescence times from DNA sequence data. *Genetics*, 145(2):505–518, February 1997. ISSN 0016-6731. doi: 10.1093/genetics/145.2.505.

Leander Thiele, Adrian E. Bayer, and Naoya Takeishi. Simulation-Efficient Cosmological Inference with Multi-Fidelity SBI, July 2025. arXiv:2507.00514 [astro-ph].

Nilesh Tripuraneni, Michael Jordan, and Chi Jin. On the theory of transfer learning: The importance of task diversity. *Advances in neural information processing systems*, 33:7852–7862, 2020.

Werner Van Geit, Michael Gevaert, Giuseppe Chindemi, Christian Rössert, Jean-Denis Courcol, Eilif B. Muller, Felix Schürmann, Idan Segev, and Henry Markram. BluePyOpt: Leveraging Open Source Software and Cloud Infrastructure to Optimise Model Parameters in Neuroscience. *Frontiers in Neuroinformatics*, 10, 2016. ISSN 1662-5196. URL https://www.frontiersin.org/articles/10.3389/fninf.2016.00017.

Huy D. Vo, Zachary Fox, Ania Baetica, and Brian Munsky. Bayesian Estimation for Stochastic Gene Expression Using Multifidelity Models. *The Journal of Physical Chemistry. B*, 123(10):2217–2234, March 2019. ISSN 1520-5207. doi: 10.1021/acs.jpcb.8b10946.

T. P. Vogels, H. Sprekeler, F. Zenke, C. Clopath, and W. Gerstner. Inhibitory plasticity balances excitation and inhibition in sensory pathways and memory networks. *Science (New York, N.Y.)*, 334(6062):1569–1573, December 2011. ISSN 1095-9203. doi: 10.1126/science.1211095.

Xinming Wang, Simon Mak, John Miller, and Jianguo Wu. Local transfer learning Gaussian process modeling, with applications to surrogate modeling of expensive computer simulators, October 2024. URL http://arxiv.org/abs/2410.12690.

David J. Warne, Thomas P. Prescott, Ruth E. Baker, and Matthew J. Simpson. Multifidelity multilevel Monte Carlo to accelerate approximate Bayesian parameter inference for partially observed stochastic processes. *Journal of Computational Physics*, 469:111543, November 2022. ISSN 00219991. doi: 10.1016/j.jcp.2022.111543.

Chulhee Yun, Shankar Krishnan, and Hossein Mobahi. A unifying view on implicit bias in training linear neural networks. *arXiv preprint arXiv:2010.02501*, 2020.

Zahra Zanjani Foumani, Mehdi Shishehbor, Amin Yousefpour, and Ramin Bostanabad. Multi-fidelity cost-aware Bayesian optimization. *Computer Methods in Applied Mechanics and Engineering*, 407:115937, March 2023. ISSN 0045-7825. doi: 10.1016/j.cma.2023.115937.

Xiaoshu Zeng, Gianluca Geraci, Michael S. Eldred, John D. Jakeman, Alex A. Gorodetsky, and Roger Ghanem. Multifidelity uncertainty quantification with models based on dissimilar parameters. *Computer Methods in Applied Mechanics and Engineering*, 415:116205, October 2023. ISSN 00457825. doi: 10.1016/j.cma.2023.116205. URL http://arxiv.org/abs/2304.08644.

Friedemann Zenke and Wulfram Gerstner. Limits to high-speed simulations of spiking neural networks using general-purpose computers. *Frontiers in Neuroinformatics*, 8:76, 2014. ISSN 1662-5196. doi: 10.3389/fninf.2014.00076.

A USAGE OF LLMS

LLM usage was minimal, limited to grammar refinement, sentence shortening, code cleanup and discovering papers outside our main domain.

B PROOF OF CONVERGENCE OF THE NPE LOG-LIKELIHOOD LOSS

Let $\theta_i \sim p(\theta_i)$ be samples from the prior of a high-fidelity model, and $x_i \sim p(x|\theta_i)$ be the respective high-fidelity simulations. In NPE, we define the loss function as the negative log likelihood:

$$\mathcal{L}(\phi) = -\frac{1}{N}\sum_{i}^{N} \log q_\phi(\theta_i|x_i), \tag{3}$$

where θ_i are samples from the prior distribution, x_i are the respective simulations (i.e., samples from $p(x|\theta_i)$), and $\boldsymbol{\phi}$ are the parameters of the neural density estimator to be optimized. If we let the number of samples θ_i (and respective simulations) $N \to \infty$:

$$\begin{aligned}
\mathcal{L}(\phi) &= \mathbb{E}_{p(\theta)p(x|\theta)}\left[-\log q_\phi(\theta|x)\right] \\
&= \mathbb{E}_{p(x)p(\theta|x)}\left[-\log q_\phi(\theta|x)\right] \\
&= \mathbb{E}_{p(x)}\left[\mathbb{E}_{p(\theta|x)}\left[\log \frac{p(\theta|x)}{q_\phi(\theta|x)}\right]\right] + C \\
&= \mathbb{E}_{p(x)}[D_{KL}\left(p(\theta|x), q_\phi(\theta|x)\right)] + C
\end{aligned} \tag{4}$$

where C is a constant with respect to ϕ. Minimizing $\mathcal{L}(\phi)$ with respect to ϕ is thus equivalent to minimizing the KL divergence between the true posterior distribution and the estimated posterior in the limit of an infinite number of high-fidelity samples.

C FURTHER EXPERIMENTAL DETAILS

C.1 TRAINING PROCEDURE

All methods and evaluations were implemented in PyTorch (Paszke et al., 2019). We used the Zuko package (version 1.4.0, MIT License)[1] (Roset, 2024) to implement the normalizing flow, based on the Neural Spline Flows (NSF) architecture (Durkan et al., 2019), and the SBI package (version 0.24.0, Apache 2.0 license)[2] (Boelts et al., 2024) for additional functions. The parameters used to generate simulations were logit-transformed for numerical stability, and the summary statistics were z-scored to improve the performance of the normalizing flows. The loss function is the negative-log likelihood, and the optimization function is the *Adam optimizer* (Kingma & Ba, 2017).

The Neural Spline Flow (NSF) architecture consists of 5 transformations, each parametrized with 50 hidden units and 8 bins. The batch size was set to 200, and the learning rate to 5×10^{-4}. The train-validation fraction is 0.1, and training of the NSF utilized an early stopping criterion with a patience of 20 epochs for the early stopping criterion. The settings described above are all default settings of the SBI package at the time of the method's development (Boelts et al., 2024).

Note, the stopping criterion follows the default configuration of the SBI package, which is defined as follows: Let E be the error function of the training algorithm (negative log likelihood), $E_{val}(t)$ the validation error at epoch t, which is used by the stopping criterion. The value $E_{opt}(t)$ is the lowest validation set error obtained in epochs up to t:

$$E_{opt}(t) := \min_{t' \leq t} E_{val}(t') \tag{5}$$

The early stopping criterion S terminates training once the validation error $E_{\text{val}}(t)$ has increased for p consecutive epochs (the patience parameter). At this point, the model corresponding to the lowest validation error observed that far, $E_{\text{opt}}(t)$, is selected and returned.

Rather than fixing the number of training epochs, the idea behind early stopping is that when the validation error has increased not only once, but over p consecutive steps, such an increase indicates a stage of overfitting (Prechelt, 2002). Note that if the patience is too small, underfitting might occur, and training may terminate too early due to stochastic fluctuations in the loss. Similarly, overfitting might likely occur when the patience is set to excessively high numbers (especially with a low number of simulations, since the loss function is typically more variable in this setting).

For the fine-tuning step of `MF-NPE`, no network weights were frozen. This choice has been purposely made to maintain full flexibility of the network to adapt to the high-fidelity model.

For the evaluation of `MF-TSNPE-AF`, we used 5 rounds of active sampling, where 80% of the high-fidelity dataset was used for standard MF-NPE training, and 20% was split across the rounds of active sampling. The active samples were selected using the acquisition function over an ensemble of 5 networks.

For a fair performance comparison, all methods were trained on the same datasets and evaluated on the same observations x_o. All amortized results were obtained over 10 network initializations, and all non-amortized results over 1 or 10 network initializations (depending on the computational cost of the task). We evaluated the methods over 30 observations for the C2ST metric, more than the 10 observations chosen previously for benchmarking (Lueckmann et al., 2021). This choice is motivated by our focus on evaluating the methods in low-data regimes, where greater certainty is required. The performance on the L5PC neuron task was evaluated with the metric NLTP and over 100 x_o's. Here, the performance of the amortized methods was averaged over 10 network initializations, and in the non-amortized methods over 1 network initialization, since training had to be performed for each individual x_o. The performance of the methods on the recurrent spiking network task was averaged over 10 network initializations and evaluated over 262,008 observations, which was the maximum number of available samples for this high-dimensional problem.

[1] `https://github.com/probabilists/zuko`
[2] `https://github.com/sbi-dev/sbi`

D TASKS

D.1 OU PROCESS

The Ornstein-Uhlenbeck (OU) process is a high-fidelity model with 2 to 4 free parameters that contains a temporal structure in the observations. As a low-fidelity model, we chose i.i.d. samples from a Gaussian distribution (unstructured vector), parametrized by the mean and standard deviation. This setting makes it well-suited to examine the impact of parameter space overlap between the low- and high-fidelity models, as well as the impact of a systematic bias in the posterior of the low-fidelity model on transfer learning.

High-fidelity model The Ornstein-Uhlenbeck process models a drift-diffusion process of a particle starting at position X(0) and drifting towards an equilibrium state. The model has two main components: a *drift* term and a *diffusion* term:

$$dX_t = \underbrace{\gamma(\mu - X_t)dt}_{\text{drift}} + \underbrace{\sigma dW_t}_{\text{diffusion}},$$

where μ is the mean of the asymptotic distribution over positions X, σ is the magnitude of the stochasticity of the process and γ is the convergence speed. $X(0)$ is the initial position of the process, which we assume to be stochastic: $X(0) \sim \mathcal{N}(\mu + \mu_{\text{offset}}, 1)$. The parameters of interest that we aim to estimate are $\mu, \sigma, \gamma, \mu_{\text{offset}}$.

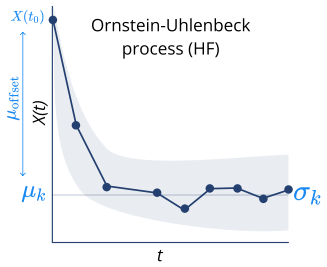


Figure 5: The four parameters of the Ornstein-Uhlenbeck process: the mean μ, standard deviation σ, convergence rate γ, and μ_{offset}, which is the difference between the initial condition $X(0)$ and mean μ.

The Ornstein-Uhlenbeck process was approximated with the Euler-Maruyama method:

$$X(t+\delta t) = X(t) + f_{\text{drift}}(t, X)\,\delta t + f_{\text{diffusion}}(t, X)\,\sqrt{\delta t}\,\mathcal{N}(0,1).$$

Starting from the exact likelihood for the Ornstein-Uhlenbeck process given by Kou et al. (2012):

$$f_{\text{exact hi}}(\boldsymbol{X} \mid \mu, \gamma, \sigma) = \prod_{t=1}^{n} \frac{1}{\sqrt{\pi g}\sigma} \exp\left\{-\frac{1}{g\sigma^2}\left((\mu - X_t) - \sqrt{1-\gamma g}\,(\mu - X_{t-1})\right)^2\right\},$$

where $g = (1 - \exp(-2\gamma\Delta t))/\gamma$, we modify it by incorporating an additional parameter μ_{offset} to account for a stochastic $X(0)$.

The full likelihood $f_{\text{exact hi}}(\boldsymbol{X} \mid \mu, \sigma, \gamma, \mu_{\text{offset}})$ is given by

$$f_{\text{exact hi}}(\boldsymbol{X} \mid \mu, \sigma, \gamma, \mu_{\text{offset}}) = \frac{1}{\sqrt{2\pi}} \exp\left\{-\frac{(x - (\mu + \mu_{\text{offset}}))^2}{2}\right\} f_{\text{exact hi}}(\boldsymbol{X} \mid \mu, \gamma, \sigma)$$

Low-fidelity model As a low-fidelity model, we use i.i.d. Gaussian Samples. At convergence, the distribution over X_t approaches a Gaussian distribution with mean μ and standard deviation $\frac{\sigma}{\sqrt{2\gamma}}$. In our setup, we chose a low-fidelity model that corresponds to time-independent random draws from a Gaussian distribution with mean μ_{lo} and standard deviation σ_{lo}:

$$X_t \sim \mathcal{N}(\mu_{\text{lo}}, \sigma_{\text{lo}}^2) \tag{6}$$

The posterior distribution over the parameters of the low-fidelity model has a biased mean influenced by the initial position μ_{offset} and convergence speed γ.

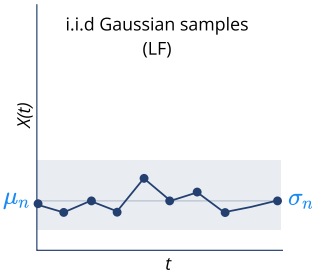


Figure 6: i.i.d. Gaussian samples with mean μ_{L} and standard deviation σ_{L}.

Prior	$\mu \sim \mathcal{U}(0.1, 3), \quad \sigma \sim \mathcal{U}(0.1, 0.6), \quad \gamma \sim \mathcal{U}(0.1, 1), \quad \mu_{\text{offset}} \sim \mathcal{U}(0, 4)$
HF Simulator	$\mathbf{x}\|\theta = (x_1, \ldots, x_{101}), \quad x_0 \sim \mathcal{N}(\mu + \mu_{\text{offset}}, 1)$, where $dx_t = \gamma(\mu - x_t)dt + \sigma dW_t$
LF Simulator	$\mathbf{x}\|\theta = (x_1, \ldots, x_{10}), \quad x_i \sim \mathcal{N}(\mu_{\text{lo}}, \sigma^2_{\text{lo}}),$
HF Dimensionality	$\theta \in \mathbb{R}^{2-4}, \quad \mathbf{x} \in \mathbb{R}^{101}, \quad U(\mathbf{x}) \in \mathbb{R}^{10}$
LF Dimensionality	$\theta \in \mathbb{R}^{2}, \quad \mathbf{x} \in \mathbb{R}^{10}, \quad U(\mathbf{x}) \in \mathbb{R}^{10}$
References	(Holý & Tomanová, 2022; Carter & Strey, 2023; Kou et al., 2012)

For the two-dimensional experiment, the free parameters $\gamma, \mu_{\text{offset}}$ have been fixed to $\gamma = 0.5$ and $\mu_{\text{offset}} = 3.0$. For the three-dimensional-experiment, only $\mu_{\text{offset}} = 3.0$. The **summary statistics** $U(x)$ from the high-fidelity model consists of 10 uniformly distributed subsamples drawn from a trace of 101 timesteps. Parameters and summary statistics are illustrated in Figures 5 and 6.

D.2 Posterior distributions over OU process

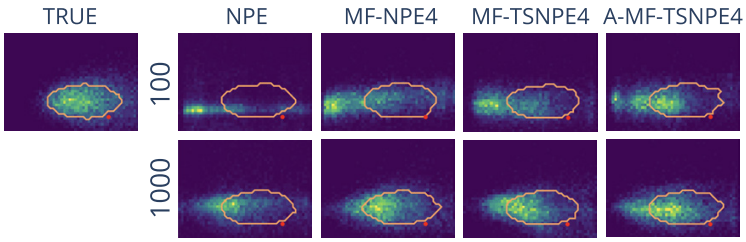


Figure 7: Posterior density estimates for a single observation from the OU process with two free parameters (OU2). The orange contour lines contain 68% of the probability mass of the true posterior distribution.

D.3 OU process with varying parameter space

We present a comparison of our multifidelity approaches to NPE and MF-ABC, with different numbers of pre-trained low-fidelity simulations. MF-NPE3 is pre-trained on a low-fidelity dataset of size 10^3, while MF-NPE4 and MF-NPE5 use datasets of 10^4 and 10^5 low-fidelity simulations, respectively. The MF-ABC results suggest that neural density approaches scale better to complex problems (Frazier et al., 2024).

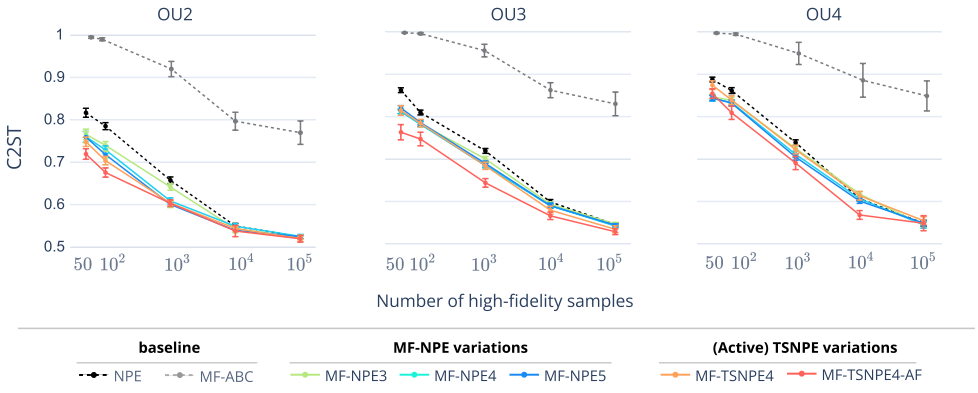


Figure 8: MF-NPE benefits from larger low-fidelity datasets. We ran MF-ABC with hyperparameters $\boldsymbol{\epsilon} = (1, 1)$ and $\boldsymbol{\eta} = (0.9, 0.3)$ (more details in Appendix E.1.1). All variants of our method perform better than MF-ABC and NPE.

D.4 Inferring the parameters of a Gaussian model pretrained on the OU3 model

In this example, we examine how the performance changes when the low-fidelity model has a larger number of parameters than the high-fidelity model: the low-fidelity model is the Ornstein-Uhlenbeck process with three parameters, and the high-fidelity model corresponds to i.i.d. Gaussian samples parameterised by a mean and variance (so, only two parameters). To accomplish that, the density estimator pre-trained on the low-fidelity model was fine-tuned only on the dimensions of the high-fidelity and the extra dimension was kept as a dummy dimension. NPE was directly trained on the 2-dimensional parameter space of the high-fidelity model. At inference time, the posterior evaluation was performed only on the high-fidelity parameter dimensions. We observe that when the dimension of θ is smaller than the dimension of θ_L, transfer learning provides a significant improvement in performance.

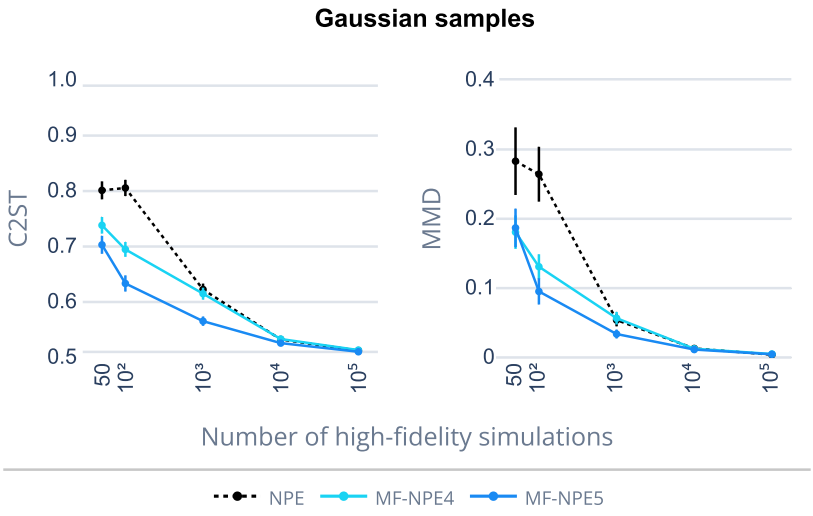


Figure 9: Evaluation with C2ST and MMD over a two-dimensional Gaussian Samples model, pretrained on the three-dimensional OU process model.

D.5 SLCP

Simple Likelihood Complex Posterior (SLCP) is a benchmark inference task that has been artificially designed to have a simple likelihood, but a very non-trivial 5-dimensional posterior to infer. In this example, we study the impact of multifidelity in cases where the dimensionality of the parameter space differs between the low-fidelity and high-fidelity models.

High-fidelity model The SLCP problem involves five parameters. The prior distribution is uniform across a five-dimensional parameter space, and the observations consist of four two-dimensional samples drawn from a Gaussian distribution. Both the mean and the variance of this Gaussian depend on the parameters through nonlinear mappings. The high-fidelity model follows the code in the SBI benchmarking paper (Lueckmann et al., 2021).

Low-fidelity model In the low-fidelity model, we experimented with the effect of different numbers of parameters on the inference quality. We fixed $m_\theta = 0$, and kept the parameters of S_θ free.

Prior	$\mathcal{U}(-3, 3)$
HF Simulator	$\mathbf{x}\|\theta = (x_1, \ldots, x_4), \quad x_i \sim \mathcal{N}(\mathbf{m}_\theta, \mathbf{S}_\theta),$ where $\mathbf{m}_\theta = \begin{bmatrix}\theta_1 \\ \theta_2\end{bmatrix}, \quad \mathbf{S}_\theta = \begin{bmatrix}s_1^2 & \rho s_1 s_2 \\ \rho s_1 s_2 & s_2^2\end{bmatrix},$ with $s_1 = \theta_3^2,\ s_2 = \theta_4^2,\ \rho = \tanh(\theta_5).$
LF Simulator	$\mathbf{x}\|\theta = (x_1, \ldots, x_4), \quad x_i \sim \mathcal{N}(0, \mathbf{S}_\theta),$ where $\mathbf{S}_\theta = \begin{bmatrix}s_1^2 & \rho s_1 s_2 \\ \rho s_1 s_2 & s_2^2\end{bmatrix},$ with $s_1 = \theta_3^2,\ s_2 = \theta_4^2,\ \rho = \tanh(\theta_5).$
HF Dimensionality	$\theta \in \mathbb{R}^5, \quad \mathbf{x} \in \mathbb{R}^8$
LF Dimensionality	$\theta \in \mathbb{R}^3, \quad \mathbf{x} \in \mathbb{R}^8$
References	(Papamakarios et al., 2019; Hermans et al., 2020) (Durkan et al., 2020; Greenberg et al., 2019; Lueckmann et al., 2021) (Thiele et al., 2025)

D.6 SIR

The Susceptible, Infected, and Recovered (SIR) model is a classical epidemiological benchmark example that captures the spread of infectious diseases through three interacting compartments: Susceptible (S), Infectious (I), and Recovered (R). Its dynamics are governed by the system of ordinary differential equations. The model is parameterized by two rates: the **infection rate** β and the **recovery rate** γ. We investigate how multifidelity addresses the partly observed dynamics of the model. Rather than observing the three dynamics of the SIR model (following the setup of the SBI benchmarking (Lueckmann et al., 2021), we assume that no dynamics regarding the recovered subjects are known (SI model).

Low-fidelity model In the low-fidelity model, we assume no information is available about the dynamics of recovered individuals. The total population size and the initial conditions are kept consistent with the high-fidelity model.

Bounded domain	$[0.001, 3]^2$
Prior	$\beta \sim \text{LogNormal}(\log(0.4), 0.5), \quad \gamma \sim \text{LogNormal}(\log(0.125), 0.2)$
HF Simulator	$\mathbf{x}\|\theta = (x_1, \ldots, x_{50}), \quad x_i = I_i/N$ equally spaced, I is simulated from $\frac{dS}{dt} = -\beta\frac{SI}{N}, \quad \frac{dI}{dt} = \beta\frac{SI}{N} - \gamma I, \quad \frac{dR}{dt} = \gamma I$
LF Simulator	$\mathbf{x}\|\theta = (x_1, \ldots, x_{50}), \quad x_i = I_i/N$ equally spaced, I is simulated from $\frac{dS}{dt} = -\beta\frac{SI}{N}, \quad \frac{dI}{dt} = \beta\frac{SI}{N} - \gamma I,$
Dimensionality	$\theta \in \mathbb{R}^2,\ x \in \mathbb{R}^{3\times 161},\ U(\mathbf{x}) \in \mathbb{R}^{10}$
Fixed parameters	Population size $N = 10^6$, duration of task $T = 160$ days. Initial conditions: $(S(0), I(0), R(0)) = (N - 1, 1, 0)$
References	(Lueckmann et al., 2021; Greenberg et al., 2019) (Hermans et al., 2020; Durkan et al., 2020)

Summary statistics $U(x)$ are 10 subsamples from the I trace.

D.7 IMAGE EXAMPLE

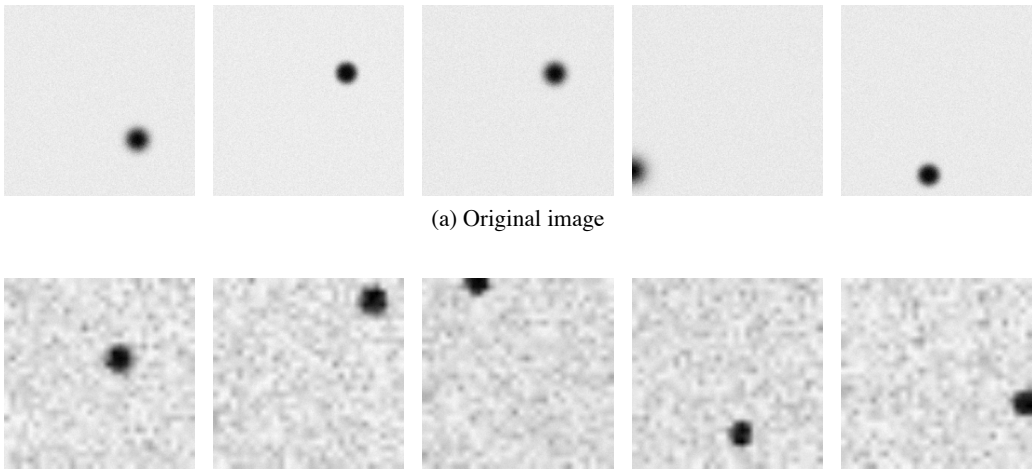
We apply our method to a problem with **high-dimensional observations**, and explore the benefits of transfer learning in combination with **embedding networks**. The high-fidelity model is a 256x256 image, while the low-fidelity model has a resolution of 32x32. An example of both simulator outputs is shown in Fig. 10.

High-fidelity model The Gaussian Blob image example contains high-dimensional observations that have been embedded with a CNN embedding from the SBI package (Boelts et al., 2024). The model renders a 2D image, which we modeled as a 256 x 256 pixel image of a Gaussian blob, and aiming to infer three parameters ($\mu_{\text{off}}, \sigma_{\text{off}}, \gamma$): the horizontal and vertical displacements of the blob, and its contrast (Lueckmann et al., 2019). The image is a grey-scale and is generated through a binomial distribution with a total count of 255 and probability p_{ij}, as described in Lueckmann et al. (2019).

Low-fidelity model In our setup, the low-fidelity model generates a spatially low-resolution dataset (32x32 image). We upscale these images using interpolation techniques and provide the resulting low-resolution inputs to the embedding network $U(x)$.

Prior HF	$x_{\text{off}}, y_{\text{off}} \sim \mathcal{U}(0, 256), \quad \gamma \sim \mathcal{U}(0.2, 2)$
Prior LF	$x_{\text{off}}, y_{\text{off}} \sim \mathcal{U}(0, 32), \quad \gamma \sim \mathcal{U}(0.2, 2)$
Simulator	$\mathbf{x}\|\theta = (x_1, \ldots, x_{1024}), \quad$ where, $I_{xy} \sim \text{Bin}(\cdot\|255, p_{xy})$ $p_{xy} = 0.9 - 0.8 \exp^{-0.5(r_{xy}/\sigma^2)^\gamma}$ $r_{xy} = (x - x_{\text{off}})^2 + (y - y_{\text{off}})^2$
Dimensionality HF	$\theta \in \mathbb{R}^3,\ x \in \mathbb{R}^{256\times256}, U(x) \in \mathbb{R}^{32}$
Dimensionality LF	$\theta \in \mathbb{R}^3,\ x \in \mathbb{R}^{32\times32}, U(x) \in \mathbb{R}^{32}$
Fixed parameters	Standard deviation $\sigma_{\text{lf}} = 2$, $\sigma_{\text{hf}} = 12$
References	(Lueckmann et al., 2019)

E GAUSSIAN BLOB EVALUATION



(a) Original image

(b) low-fidelity simulations

Figure 10: Five examples of generated images with the Gaussian Blob across the two fidelities, with (a) the original 256x256 high-fidelity simulations, (b) the upsampled 32x32 low-fidelity simulations.

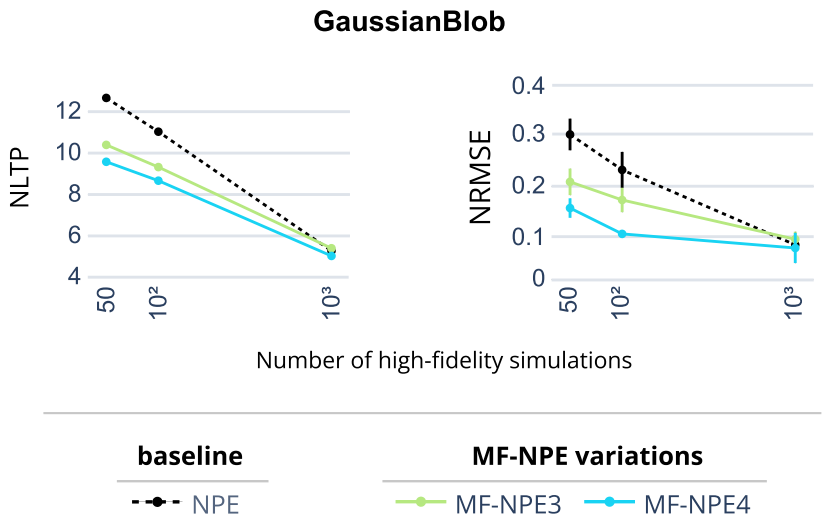


Figure 11: Method comparison with NLTP and NRMSE for the Gaussian Blob task. Evaluated over 10000 observations.

E.1 Data generation and transformations for increased network performance

During the performance evaluation, we encountered numerical instabilities, particularly with NPE in low-simulation budgets: a substantial proportion of the estimated probability density was placed outside of the uniform prior bounds, a phenomenon dubbed 'leakage' that has been previously documented (Greenberg et al., 2019; Deistler et al., 2022). Logit-transforming the model parameters before training the density estimator resolved the issue.

This transformation creates a mapping from a bounded to an unbounded space, resulting in a density estimation within the prior bounds after the inverse transformation. In addition, the summary statistics of the simulations were z-scored for improved performance of the density estimator, the default setting in the SBI package (Boelts et al., 2024).

E.1.1 Multifidelity Approximate Bayesian Computation (MF-ABC)

We translated into Python a publicly available Julia implementation of the multifidelity ABC algorithm (Prescott & Baker, 2020). In our setup, the adaptive sampling scheme of MF-ABC selected approximately 30% of the batch size as high-fidelity samples in the OU2 and OU3 tasks, and 50% in the OU4 task. To ensure consistency with our neural network experiments, we z-scored the simulator output before inference. We also explored the effect of varying the acceptance threshold ϵ. We found that the hyperparameters slightly affect the performance of MF-ABC, but that MF-NPE always shows superior performance than MF-ABC (Figure 12). However, MF-ABC has several other hyperparameters to tune. We cannot exclude the hypothesis that larger performance gains could be obtained from such an approach by a more extensive hyperparameter search.

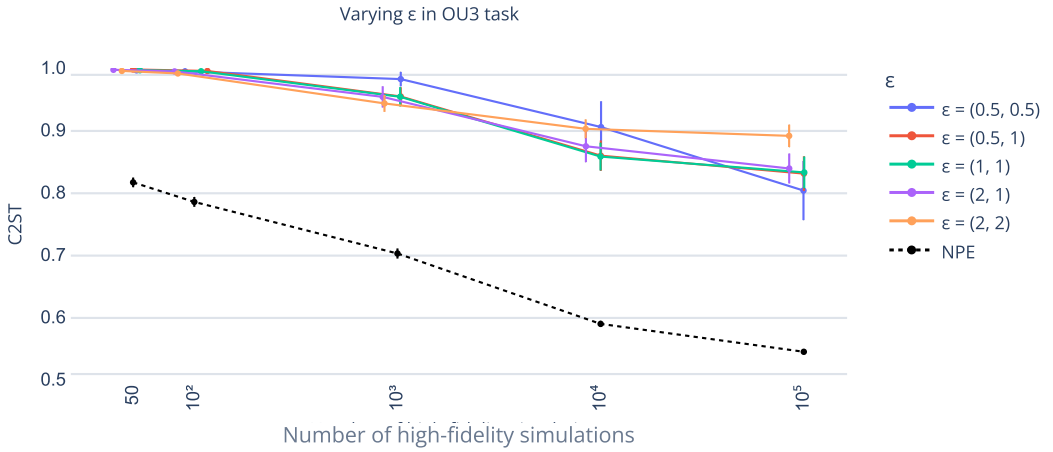


Figure 12: **C2ST results for MF-ABC with varying hyperparameters ϵ.** Mean and 95% confidence interval.

MF-ABC posteriors ABC-based methods typically require a significantly larger number of samples for convergence (Lueckmann et al., 2021; Frazier et al., 2024). In line with previous studies, we find that 10^4 samples are not yet enough for MF-ABC to converge to a good estimate of the posterior in the OU2 task.

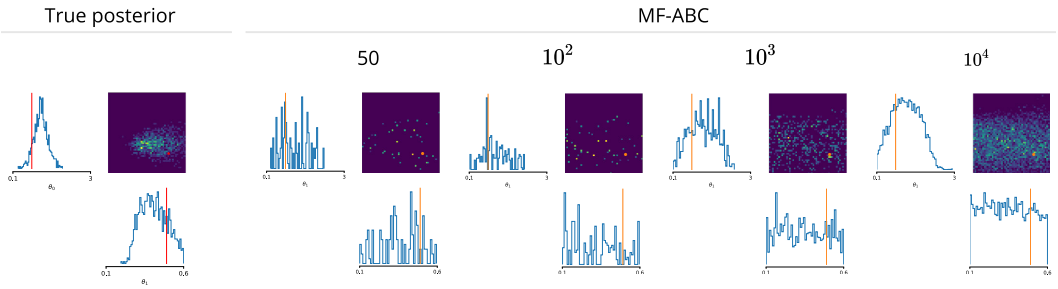


Figure 13: **Comparison between MF-ABC posterior estimates and the true posterior.** Results for the Ornstein-Uhlenbeck process with two free parameters. Posterior estimates are shown for varying numbers of high-fidelity simulations (50, 100, 10^3, and 10^4).

F TASK 2: MULTICOMPARTMENTAL SINGLE NEURON MODEL

The response of a morphologically detailed neuron to an input current is typically modeled with a multicompartmental neuron model wherein the voltage dynamics of each compartment μ are based on Hodgkin-Huxley equations (Hodgkin & Huxley, 1952):

$$\begin{aligned} c_{\mathrm{m}} \frac{dV_\mu}{dt} = &- i_{\mathrm{m}}^\mu + \frac{I_{\mathrm{e}}^\mu}{A_\mu} + g_{\mu,\mu+1}\left(V_{\mu+1} - V_\mu\right) \\ &+ g_{\mu,\mu-1}\left(V_{\mu-1} - V_\mu\right). \end{aligned} \tag{7}$$

The total membrane current i_{m} for a specific compartment is the sum over different types of ion channels i, such as sodium, potassium and leakage channels:

$$i_{\mathrm{m}} = \bar{g}_{\mathrm{Na}} m^3 h (V - E_{\mathrm{Na}}) + \bar{g}_{\mathrm{K}} n^4 (V - E_{\mathrm{K}}) + \bar{g}_{\mathrm{L}} (V - E_{\mathrm{L}}) + \bar{g}_{\mathrm{M}} p (V - E_{\mathrm{M}}) \tag{8}$$

We are interested in inferring the densities of two prominent ion channels $\bar{g}_{\mathrm{Na}}$ and $\bar{g}_{\mathrm{K}}$.

The low- and high-fidelity models differ in the number of compartments per branch: the low-fidelity model has a single compartment per branch, while the high-fidelity model consists of eight compartments per branch.

All simulations were performed using Jaxley (V 0.8.2) (Deistler et al., 2024) over 120 ms. The injection current is a step current of 0.55mV over 100 ms, with a delay of 10ms. The step size of the simulator is 0.025.

When sampling from the prior distribution over parameters, approximately $0.05 - 0.1\%$ of the respective simulations had clearly unrealistic summary statistics: these simulations were iteratively replaced by random draws from the prior distribution/proposal or active learning list (depending on the algorithm) until we collected a desired number of valid simulations.

F.1 NRMSE EVALUATION

In addition to the NLTP metric, we demonstrate that the NRMSE metric yields results that support our conclusions.

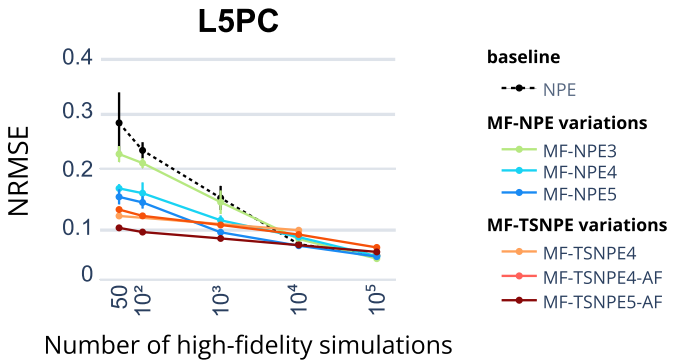


Figure 14: NRMSE evaluation for the multicompartmental neuron model.

F.2 SIMULATION-BASED CALIBRATION AND POSTERIOR DISTRIBUTIONS

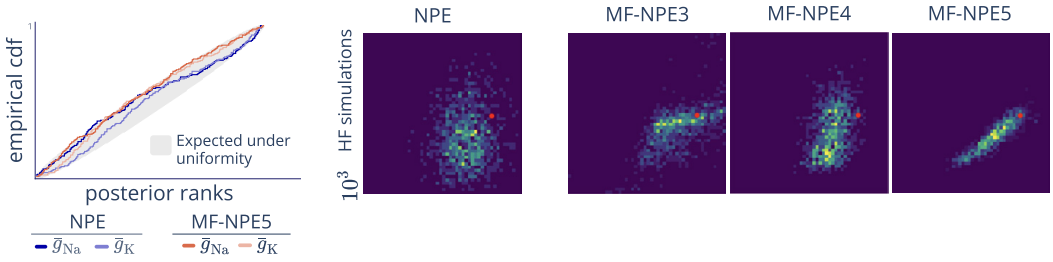


Figure 15: Simulation-based calibration (left) and respective posterior distributions for NPE and MF-NPE (right) for the multicompartmental neuron model task. MF-NPE is respectively, pretrained on $10^3, 10^4, 10^5$ low-fidelity simulations (dubbed as MF-NPE3, MF-NPE4, and MF-NPE5). All models were trained on 10^3 high-fidelity simulations.

F.3 Posterior Predictive Checks

With only 50 high-fidelity simulations, MF-NPE gives similar accuracy to NPE trained on 1000 simulations (Fig. 16), and for a fixed number of 1000 high-fidelity simulations, MF-NPE5 outperforms NPE (Fig. 17).

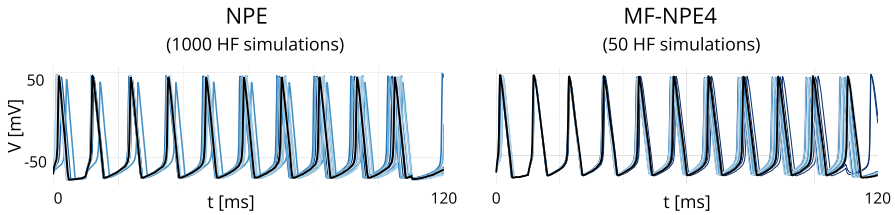


Figure 16: Posterior predictives for the multicompartmental neuron model with varying number of high-fidelity simulations.

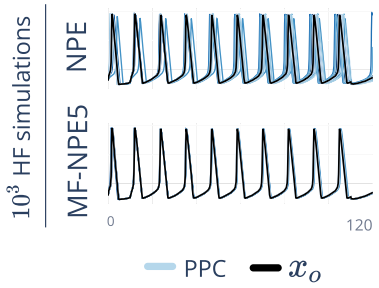


Figure 17: Posterior predictives for the multicompartmental neuron model for a fixed number of high-fidelity simulations.

F.4 Low and high-fidelity traces

We present simulations with the models with 1- and 8-compartments per dendritic branch (low- and high-fidelity models, respectively) to illustrate that the model outputs are different, given the same parameters.

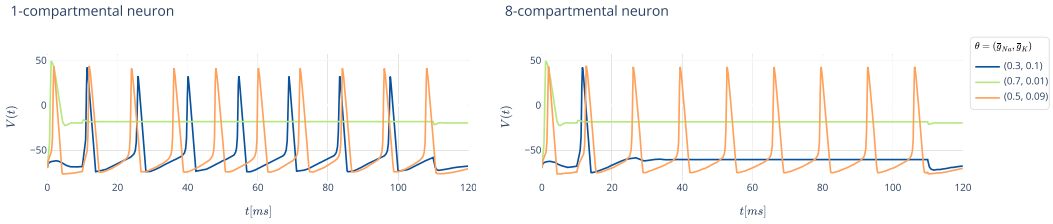


Figure 18: Simulated membrane potential traces of an L5 pyramidal cell (L5PC) model with Jaxley (Deistler et al., 2024). The low- and high-fidelity models are, respectively, a single-compartment model per dendritic branch versus an eight-compartment model per branch.

G TASK 3: SPIKING NETWORK MODEL

G.1 HIGH-FIDELITY MODEL

We considered a recurrent spiking network of 5120 neurons (4096 excitatory, 1024 inhibitory), with parameters taken from Confavreux et al. (2023). The membrane potential dynamics of neuron j, excitatory (E) or inhibitory (I), followed

$$\tau_m \frac{\mathrm{d}V_j}{\mathrm{d}t} = -\left(V_j - V_{\text{rest}}\right) - g_j^{\mathrm{E}}(t)\left(V_j - E_{\mathrm{E}}\right) - g_j^{\mathrm{I}}(t)\left(V_j - E_{\mathrm{I}}\right), \tag{9}$$

A postsynaptic spike was generated whenever the membrane potential $V_j(t)$ crossed a threshold $V_j^{\text{th}}(t)$, with an instantaneous reset to V_{reset}. This threshold $V_j^{\text{th}}(t)$ was incremented by $V_{\text{spike}}^{\text{th}}$ every time neuron j spiked and otherwise decayed following

$$\tau_{\text{th}} \frac{\mathrm{d}V_j^{\text{th}}}{\mathrm{d}t} = V_{\text{base}}^{\text{th}} - V_j^{\text{th}}. \tag{10}$$

The excitatory and inhibitory conductances, g^{E} and g^{I} evolved such that

$$\begin{aligned} g_j^{\mathrm{E}}(t) &= a g_j^{\text{AMPA}}(t) + (1-a) g_j^{\text{NMDA}}(t) \quad \text{and} \quad \frac{\mathrm{d}g_j^{\mathrm{I}}}{\mathrm{d}t} = -\frac{g_j^{\mathrm{I}}}{\tau_{\text{GABA}}} + \sum_{i \in \text{Inh}} w_{ij}(t)\delta_i(t) \\ \text{with} \quad \frac{\mathrm{d}g_j^{\text{AMPA}}}{\mathrm{d}t} &= -\frac{g_j^{\text{AMPA}}}{\tau_{\text{AMPA}}} + \sum_{i \in \text{Exc}} w_{ij}(t)\delta_i(t) \quad \text{and} \quad \frac{\mathrm{d}g_j^{\text{NMDA}}}{\mathrm{d}t} = \frac{g_j^{\text{AMPA}}(t) - g_j^{\text{NMDA}}}{\tau_{\text{NMDA}}}, \end{aligned} \tag{11}$$

with $w_{ij}(t)$ the connection strength between neurons i and j (unitless), $\delta_k(t) = \sum \delta(t - t_k^*)$ the spike train of pre-synaptic neuron k, where t_k^* denotes the spike times of neuron k, and δ the Dirac delta. All neurons received input from 5k Poisson neurons, with 5% random connectivity and constant rate $r_{\text{ext}} = 10$Hz in each simulation. The recurrent connectivity was instantiated with random sparse connectivity (10%). All recurrent synapses in the network (E-to-E and E-to-I, I-to-E, I-to-I) underwent variations of spike-timing dependent plasticity (STDP) (Gerstner & Kistler, 2002; Confavreux et al., 2023). Given the learning rate η, the weights between the neurons i and j of connection type X-to-Y evolved over time as:

$$\begin{aligned} \frac{dw_{ij}}{dt} =& \eta \left[\delta_{\text{pre}}(t)\left(\alpha + \kappa x_{\text{post}}(t)\right)\right. \\ &\left. + \delta_{\text{post}}(t)\left(\beta + \gamma x_{\text{pre}}(t)\right)\right]. \end{aligned} \tag{12}$$

with variables $x_i(t)$ and $x_j(t)$ describing the pre- and postsynaptic spikes over time:

$$\frac{\mathrm{d}x_i}{\mathrm{d}t} = -\frac{x_i}{\tau_{\text{XY}}^{\text{pre}}} + \delta_i(t) \quad \text{and} \quad \frac{\mathrm{d}x_j}{\mathrm{d}t} = -\frac{x_j}{\tau_{\text{XY}}^{\text{post}}} + \delta_j(t) \tag{13}$$

with $\tau_{\text{XY}}^{\text{pre}}$ and $\tau_{\text{XY}}^{\text{post}}$ the time constants of the traces associated with the pre- and postsynaptic neurons, respectively.

The 24 free parameters of interest were $\tau_{\text{pre}}, \tau_{\text{post}}, \alpha, \beta, \kappa, \gamma$ multiplied by the number of synapse types (e.g., $\alpha_{EE}, \alpha_{II}, \alpha_{EI}, \alpha_{IE}$), following previous work (Confavreux et al., 2023).

G.2 LOW-FIDELITY MODEL

Following previous work (Confavreux et al., 2023; Vogels et al., 2011; Dayan & Abbott, 2001), a (partial) mean-field theory applied to the E-to-E and E-to-I connections in the model described above gave:

$$r_{\mathrm{E}}^* = \frac{-\alpha_{\text{EE}} - \beta_{\text{EE}}}{\lambda_{\text{EE}}} \quad \text{and} \quad r_{\mathrm{I}}^* = \frac{-\alpha_{\text{EI}} r_{\mathrm{E}}^*}{\beta_{\text{EI}} + \lambda_{\text{EI}} r_{\mathrm{E}}^*} \tag{14}$$

with r_{E}^* and r_{I}^* the firing rates of the excitatory (resp. inhibitory) population at steady state, and

$$\lambda_{\text{XY}} = \kappa_{\text{XY}} \tau_{\text{XY}}^{\text{post}} + \gamma_{\text{XY}}^{\text{pre}} \tag{15}$$

With type $(X, Y) \in \{E, I\}$. For all synapse types, we assume $(-\alpha_{\text{XY}} - \beta_{\text{XY}}) > 0$ and $\lambda_{\text{XY}} > 0$, as a second-order stability condition (Confavreux et al., 2023). Note that in this low-fidelity model, we only considered 2 of the 4 plastic conditions, and thus 12 of the 24 free parameters of the high-fidelity model.

G.3 Synaptic Plasticity with varying parameter space

We investigated how inference performance changes as the discrepancy between the low- and high-fidelity models increases. To this end, we varied the dimensionality of the low-fidelity model between 3, 6, and 12 parameters, while keeping the high-fidelity model fixed at 24 parameters. Parameters that were excluded from inference in the low-fidelity settings were fixed to the following values for each connection type: $\tau_{\text{pre}} = \tau_{\text{post}} = 0.05, \gamma = -1.9, \alpha = \beta = \kappa = 0.5$. The value of γ should be smaller than other parameters to fulfill the second-order stability condition (Confavreux et al., 2023).

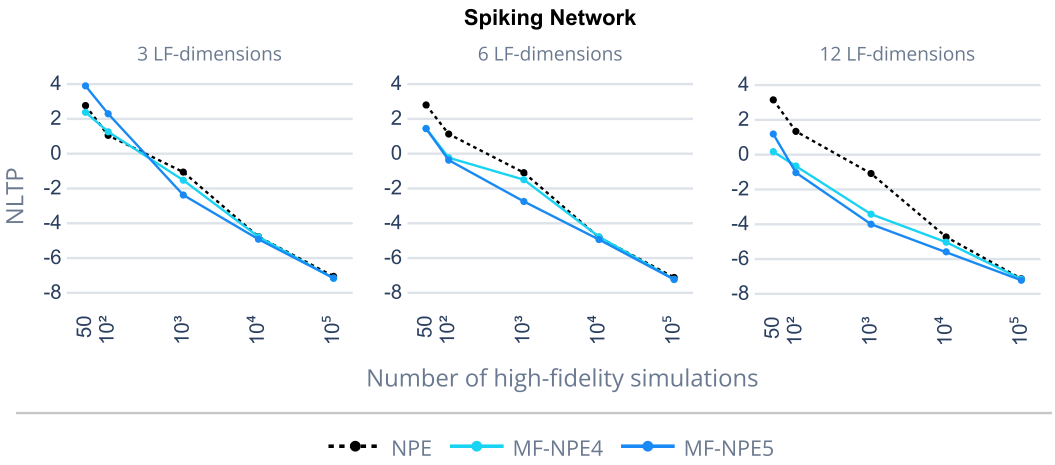


Figure 19: Negative-log-likelihood over true parameters, with different numbers of free parameters in the low-fidelity model.

We observe that the performance of MF-NPE degrades as the number of parameters in the low-fidelity model decreases as compared to the high-fidelity model. In particular, unlike in all our other experiments, when the low-fidelity model had only 3 parameters, pretraining on 10^5 low-fidelity samples led to worse MF-NPE performance: in this regime, using 10^5 samples (MF-NPE5) resulted in negative transfer, whereas pretraining on 10^4 samples (MF-NPE4) resulted in a performance close to standard NPE.

G.4 Discussion on alternative solutions

We consider the following strategies:

- pretraining on solely low-fidelity simulations,
- pretraining on the joint of low- and high-fidelity simulations.

G.4.1 Pretraining on low fidelity samples

This approach follows the main discussion in Sec. 3.1.1, and has also been the main method employed in the paper. We purposefully do not freeze the weights after transfer, allowing the network to retain the flexibility to adapt to high-fidelity simulations.

G.4.2 Pretraining on the joint of LF and HF samples

We examined whether pretraining on the joint distribution of low- and high-fidelity simulations could provide a better initialization for subsequent fine tuning. As shown in Fig. 20, this strategy yields no significant improvement on the first two benchmarking tasks compared to standard MF-NPE. However, we encourage further work to investigate additional variations on this approach to improve the domain adaptation (e.g., domain adaptation through MMD Elsemüller et al. (2025), importance weighting for extremely unbalanced datasets, adversarial discriminative domain adaptation, training a single multifidelity inference network).

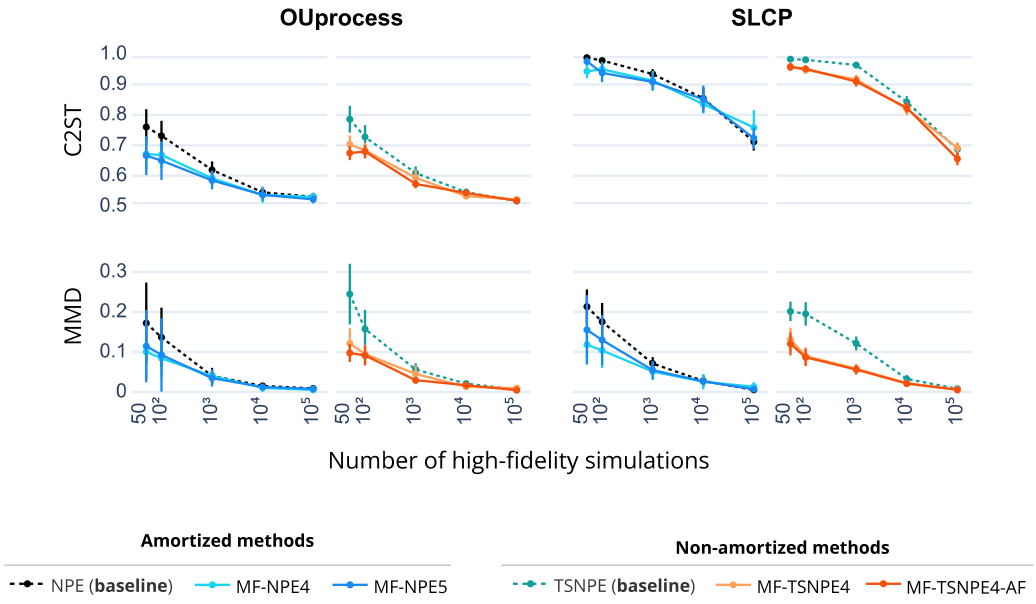


Figure 20: MF-(TS)NPE (joint) has been pretrained on both low- and high-fidelity samples.

H PRIOR BOUNDS ACROSS NEUROSCIENCE TASKS

For the OU process task, we chose a uniform prior with bounds that would lead to a range of different outputs. For the multicompartment neuron model task, we chose a uniform prior with bounds based on the work of Deistler et al. (2022). For the spiking network model task, we chose a uniform prior with bounds based on the work of Confavreux et al. (2023).

Table 1: Prior bounds for the single- and multicompartmental neuron model.

PARAMETER NAME	LOWER BOUND	UPPER BOUND
$\bar{g}_{\mathrm{NA}}$	0.005	0.8
$\bar{g}_{\mathrm{K}}$	10^{-6}	0.15

Table 2: Prior bounds for each synapse type (E-to-E, E-to-I, I-to-E and I-to-I) for the spiking neural network and mean-field model.

PARAMETER NAME	LOWER BOUND	UPPER BOUND
τ_{pre}	0.01	0.1
τ_{post}	0.01	0.1
α	-2	2
β	-2	2
γ	-2	2
κ	-2	2

I DISTANCE BETWEEN THE LF AND HF POSTERIOR

Both the low and high-fidelity posterior distributions have been trained on 10^5 simulations and evaluated over 10 true observations. In the table below, we focus on cases with two fidelities and measure the distance between the low and high-fidelity models with the MMD and C2ST metrics. We observe that the distance between the posterior distributions is not a direct measure of success in transfer learning. For instance, the posterior distributions of the low-and high-fidelity models of the L5PC neuron are significantly different. However, the network still manages to leverage information between the two simulators (Figure 3), supporting the theoretical results of Tahir et al. (2024).

Transfer learning seems to work less well on the OU process task when the dimensionality of the parameters differs between the low- and high-fidelity models (see Sec. 8). This is observed despite the fact that the distance between the low and high-fidelity posteriors is lower for the OU4 case than for the OU2 case, as the low-fidelity OU2 posterior is highly biased (Fig. 21).

Table 3: Distance between low- and high-fidelity posterior (mean $\pm$ std) for different tasks.

Task	**MMD**	**C2ST**
SLCP	0.13 ± 0.05	0.91 ± 0.03
SIR	0.04 ± 0.03	0.57 ± 0.03
OU2	1.00 ± 0.11	0.98 ± 0.02
OU3	0.69 ± 0.087	0.98 ± 0.01
OU4	0.24 ± 0.05	0.90 ± 0.04
L5PC	0.76 ± 0.23	0.99 ± 0.00
SynapticPlasticity	0.01 ± 0.00	0.70 ± 0.02

I.1 PAIRPLOTS

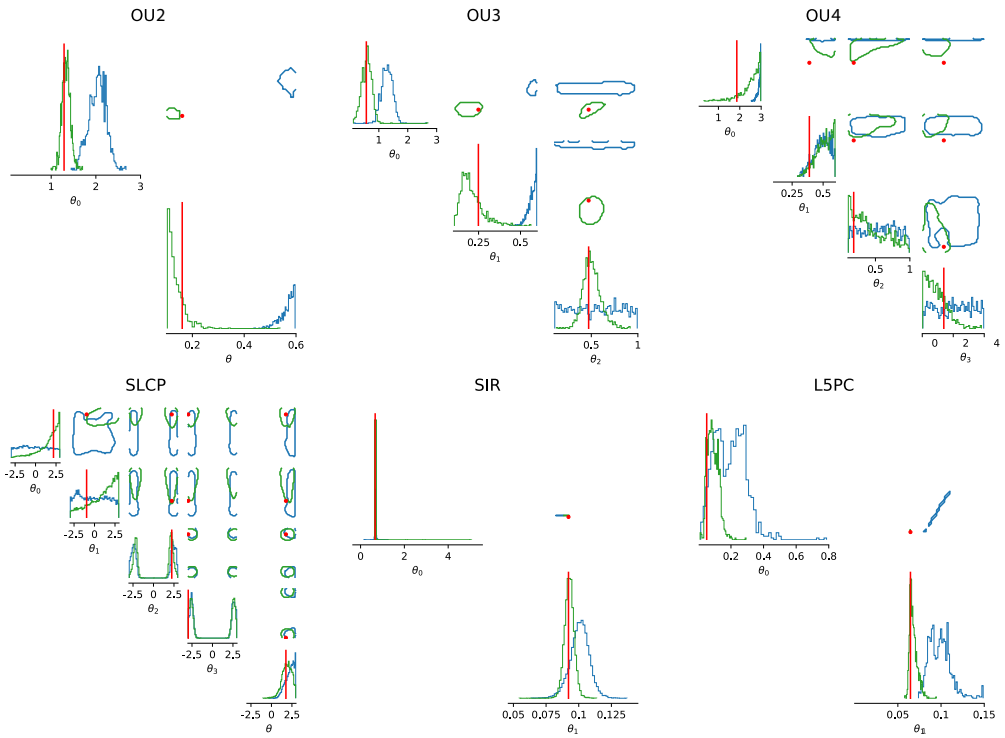


Figure 21: Posterior distributions of the low-fidelity posterior (blue) and high-fidelity posterior (green). Contours contain 68% of the true posterior mass for the low-fidelity model. Vertical bars and dots correspond to the value of the true parameters.

J SIMULATION VERSUS TRAINING COST

We tracked the wall-clock run-time for training and simulation stages of the neural density estimator. Computations were performed on nodes each equipped with 4× Intel Xeon Gold 6448H CPUs (32 cores per socket, 128 physical cores, 256 logical CPUs) and approximately 2TB RAM, running Linux 5.14.0. We compare the costs in regimes where the performance of NPE is similar to MF-NPE and MF-TSNPE-AF (Fig. 3). Details about the network architecture and hyperparameters are described in Appendix C.1. In cases where many samples had to be generated for active non-amortized schemes (e.g., 10^5 HF samples for the L5PC task; Figure 3), we used multiprocessing over CPUs. The simulations for the third task were parallelized over 913 CPUs.

Table 4: Comparison of methods for the real-world tasks in terms of the number of simulations and computational cost. Total training cost is reported as mean ± standard deviation over 5 network runs.

	method	# simulations		CPU (seconds)		
		LF	HF	tot. cost (sim.)	tot. cost (train)	total cost
L5PC	NPE	NA	10^4	4940	70.39 ± 18.32	5010.39 ± 18.32
	MF-NPE	10^4	10^3	1032	96.94 ± 15.19	1128.94 ± 15.19
	MF-TSNPE-AF	10^4	50	607	557.44 ± 52.5	1164.44 ± 52.5
Network	NPE	NA	10^4	3×10^6	120.43	3,000,120
	MF-NPE	10^5	10^3	3×10^5	94.54	300,094

Table 5: Comparison of methods across models in terms of the number of simulations and accuracy. Evaluated using the NLTP metric.

	Method	# Simulations		Accuracy (C2ST/NLTP)
		LF	HF	
L5PC	NPE	NA	10^4	-5.87 ± 0.04
	MF-NPE	10^4	10^3	-5.73 ± 0.05
	MF-TSNPE-AF	10^4	50	-5.08 ± 0.27
Network	NPE	NA	10^4	-4.72 ± 0.01
	MF-NPE	10^5	10^3	-4.08 ± 0.01

Table 4 shows that the multifidelity approaches make sense when the training cost is significantly lower than the simulation cost, such as in the L5PC and the spiking network model. For instance, in the spiking network task, a single high-fidelity simulation requires approximately 5 CPU minutes, whereas a low-fidelity simulation takes only 0.0008 seconds.

K TARP EVALUATION FOR ALL TASKS

We performed additional evaluations on the calibration of all experiments with TARP (Lemos et al., 2023).

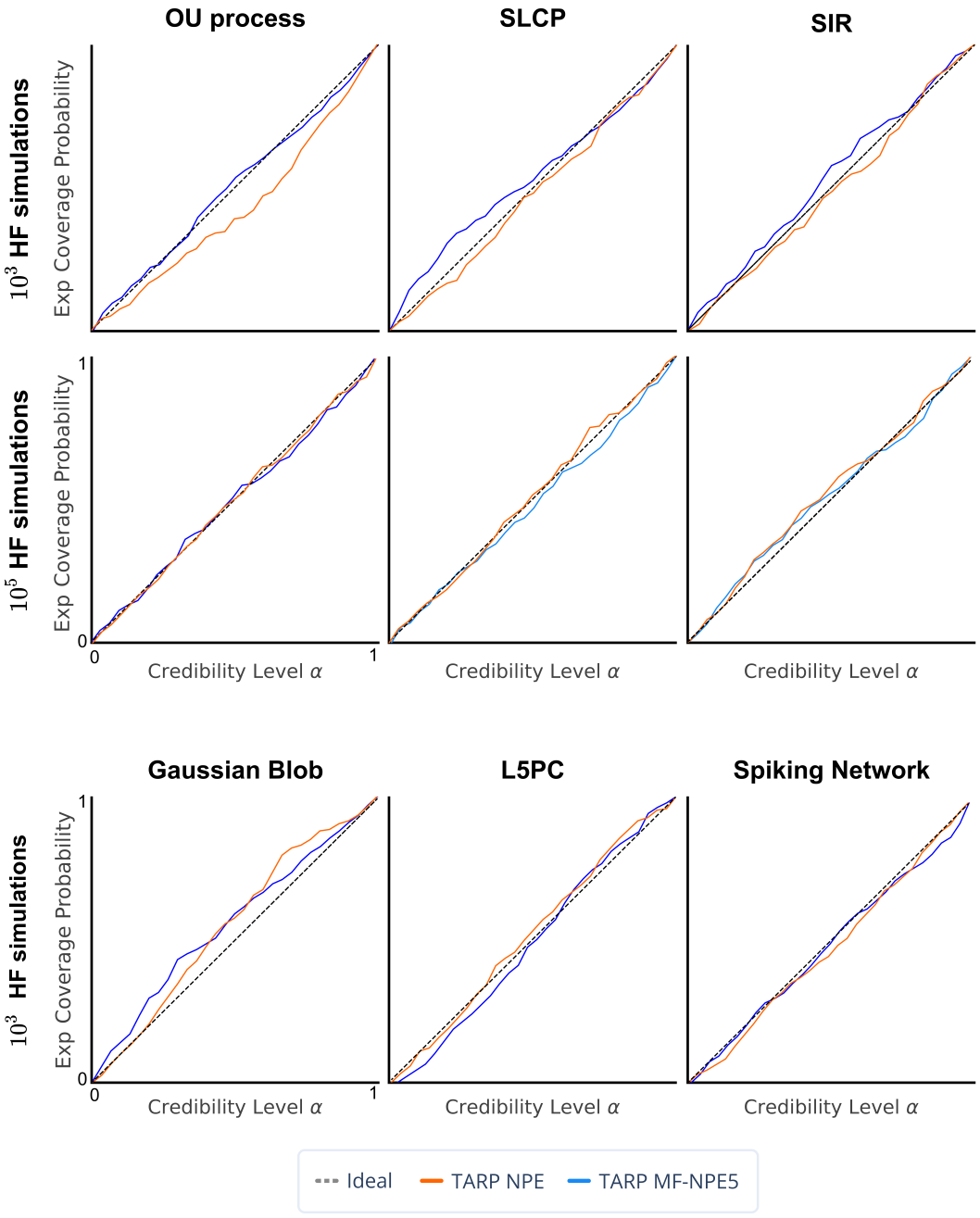


Figure 22: TARP calibration test across 10^5 LF simulations (10^4 for the Gaussian blob example). The calibration test was performed over 200 runs.

L MF-NPE FOR MULTIPLE LOWER-FIDELITY SIMULATORS

Algorithm 2 MF-NPE with multiple fidelities

Input: Simulations $\{(\boldsymbol{\theta}, \boldsymbol{x}^{(f)})\}_{f=1}^{F}$ over F fidelities; Early stopping criterion S; conditional density estimators $\{q_\psi^{(f)}(\boldsymbol{\theta}|\boldsymbol{x}^{(f)})\}_{f=1}^{F}$ with features ψ.
for $f = 1$ **to** F **do**
  $\mathcal{L}(\psi^{(f)}) = \frac{1}{N^{(f)}} \sum_{i=1}^{N^{(f)}} -\log q_\psi^{(f)} \left(\boldsymbol{\theta}_i | \boldsymbol{x}_i^{(f)}\right)$.
  $\mathrm{opt}^{(f)} \leftarrow \mathrm{Adam}(\cdot)$
  if $f > 1$ **then**
    Initialize $q_\psi^{(f)}$ with features of trained $q_\psi^{(f-1)}$.
  end if
  for epoch in epochs **do**
    train $q_\psi^{(f)}$ to minimize $\mathcal{L}(\psi^{(f)})$ until S is reached.
  end for
end for

M SEQUENTIAL ALGORITHMS

M.1 MF-TSNPE

Algorithm 3 MF-TSNPE

Input: N pairs of $(\boldsymbol{\theta}, \boldsymbol{x}_{\mathrm{L}})$; conditional density estimators $q_\psi(\boldsymbol{\theta}|\boldsymbol{x}_{\mathrm{L}})$ and $q_\phi(\boldsymbol{\theta}|\boldsymbol{x})$ with learnable parameters ψ and ϕ; early stopping criterion S; simulator $p(\boldsymbol{x}|\boldsymbol{\theta})$; prior $p(\boldsymbol{\theta})$; number of rounds R; ϵ that defines the highest-probability region (HPR_ϵ); number of high-fidelity simulations per round M.
Output: posterior estimate $q_\phi(\boldsymbol{\theta}|\boldsymbol{x})$
$\mathcal{L}(\psi) = \frac{1}{N}\sum_{i=1}^{N} -\log q_\psi\left(\boldsymbol{\theta}_i|\boldsymbol{x}_i^{\mathrm{L}}\right)$.
for epoch in epochs **do**
    train q_ψ to minimize $\mathcal{L}(\psi)$ until S is reached.
end for
Initialize $\tilde{p}(\boldsymbol{\theta})$ as $p(\boldsymbol{\theta})$
Initialize q_ϕ with weights and biases of trained q_ψ.
for r in R **do**
    $\boldsymbol{\theta}^{(r)} \sim \tilde{p}(\boldsymbol{\theta})$, sample parameters from proposal
    $\boldsymbol{x}^{(r)} \sim p(\boldsymbol{x}|\boldsymbol{\theta}^{(r)})$, generate high-fidelity simulations
    for epoch in epochs **do**
        $\mathcal{L}(\phi) = \frac{1}{M}\sum_{i=1}^{M} -\log q_\phi\left(\boldsymbol{\theta}_i^{(r)}|\boldsymbol{x}_i^{(r)}\right)$.
        train q_ϕ to minimize $\mathcal{L}(\phi)$ until S is reached.
    end for
    Compute expected coverage ($\tilde{p}(\boldsymbol{\theta})$, q_ϕ)
    $\tilde{p}(\boldsymbol{\theta}) \propto p(\boldsymbol{\theta}) \cdot \mathbb{1}_{\theta \in \mathrm{HPR}_\epsilon}$
end for

All experiments were run with $R = 5$ rounds and $\epsilon = 1e^{-6}$. More details about TSNPE at Deistler et al. (2022).

M.2 MF-TSNPE-AF

Algorithm 4 MF-TSNPE-AF

Input: N pairs of $(\boldsymbol{\theta}, \boldsymbol{x}_{\mathrm{L}})$; conditional density estimator $q_\psi(\boldsymbol{\theta}|\boldsymbol{x}_{\mathrm{L}})$ with learnable parameters ψ and ensemble of conditional density estimators $\{q_\phi^e(\boldsymbol{\theta}|\boldsymbol{x})\}_E^e$, each with independent ϕ; early stopping criterion S; simulator $p(\boldsymbol{x}|\boldsymbol{\theta})$; prior $p(\boldsymbol{\theta})$; number of rounds R; ϵ that defines the highest-probability region (HPR_ϵ); number of high-fidelity simulations per round M.
Output: Ensemble posterior estimate $q_\phi(\boldsymbol{\theta}|\boldsymbol{x}) = \frac{1}{E}\sum_{e=1}^{E} q_\phi^e(\boldsymbol{\theta}|\boldsymbol{x})$
$\mathcal{L}(\psi) = \frac{1}{N}\sum_{i=1}^{N} -\log q_\psi\left(\boldsymbol{\theta}_i|\boldsymbol{x}_i^{\mathrm{L}}\right)$.
for epoch in epochs **do**
    train q_ψ to minimize $\mathcal{L}(\psi)$ until S is reached.
end for
for e $\in$ Ensemble **do**
    Initialize q_ϕ^e with weights and biases of trained q_ψ.
end for
$\boldsymbol{\theta}_{\mathrm{pool}} \sim p(\boldsymbol{\theta})$
Initialize $\tilde{p}(\boldsymbol{\theta})$ as $p(\boldsymbol{\theta})$
for r in R **do**
    $\boldsymbol{\theta}_{\mathrm{prop}}^{(r)} \sim \tilde{p}(\boldsymbol{\theta})$, generate $M - B$ samples from proposal
    $\boldsymbol{\theta}_{\mathrm{active}}^{(r)}$ = top B values from $\boldsymbol{\theta}_{\mathrm{pool}}$ using the acquisition function eq. equation 2
    $\boldsymbol{\theta}^{(r)} = \{\boldsymbol{\theta}_{\mathrm{prop}}^{(r)} \cup \boldsymbol{\theta}_{\mathrm{active}}^{(r)}\}$
    $\boldsymbol{x}^{(r)} \sim p(\boldsymbol{x}|\boldsymbol{\theta}^{(r)})$, generate high-fidelity simulations
    for e $\in$ Ensemble **do**
        for epoch in epochs **do**
            $\mathcal{L}(\phi) = \frac{1}{M}\sum_{i=1}^{M} -\log q_\phi^e\left(\boldsymbol{\theta}_i^{(r)}|\boldsymbol{x}_i^{(r)}\right)$.
            train q_ϕ to minimize $\mathcal{L}(\phi)$ until S is reached.
        end for
    end for
    Compute expected coverage ($\tilde{p}(\boldsymbol{\theta})$, $\frac{1}{E}\sum q_\phi^e(\boldsymbol{\theta}|\boldsymbol{x})$)
    $\tilde{p}(\boldsymbol{\theta}) \propto p(\boldsymbol{\theta}) \cdot \mathbb{1}_{\theta \in \mathrm{HPR}_\epsilon}$
end for

All experiments were run with $R = 5$ rounds, $\epsilon = 1e^{-6}$, and an ensemble of 5 networks. The addition of an acquisition function biases the proposal distribution, causing the density estimate to diverge from the true posterior. In principle, this could be addressed by using atomic proposals (Greenberg et al., 2019), but given that such an approach suffers from posterior leakage, we do not introduce a proposal correction in order to retain the well-behaved loss function in TSNPE. We argue that the benefit of informative samples would outweigh the potential bias, as long as the percentage of samples selected from the acquisition function would be small compared to the proposal samples. Therefore, we set $B = .2M$ to mitigate the concern of biasing the posterior with parameters selected with the acquisition function.