Master thesis : Conservative Simulation-Based Inference with Bayesian Deep Learning

Auteur : de la Brassinne Bonardeaux, Maxence
Promoteur(s) : Louppe, Gilles
Faculté : Faculté des Sciences appliquées
Diplôme : Master : ingénieur civil en science des données, à finalité spécialisée
Année académique : 2023-2024
URI/URL : http://hdl.handle.net/2268.2/20480

Low-Budget Simulation-Based Inference with Bayesian Neural Networks

Arnaud Delaunoy[*]
University of Liège

Maxence de la Brassinne Bonardeaux[*]
University of Liège

Siddharth Mishra-Sharma
The NSF AI Institute for Artificial Intelligence and Fundamental Interactions, Cambridge
Center for Theoretical Physics, Massachusetts Institute of Technology, Cambridge
Department of Physics, Harvard University, Cambridge

Gilles Louppe
University of Liège

Abstract

Simulation-based inference methods have been shown to be inaccurate in the data-poor regime, when training simulations are limited or expensive. Under these circumstances, the inference network is particularly prone to overfitting, and using it without accounting for the computational uncertainty arising from the lack of identifiability of the network weights can lead to unreliable results. To address this issue, we propose using Bayesian neural networks in low-budget simulation-based inference, thereby explicitly accounting for the computational uncertainty of the posterior approximation. We design a family of Bayesian neural network priors that are tailored for inference and show that they lead to well-calibrated posteriors on tested benchmarks, even when as few as $O(10)$ simulations are available. This opens up the possibility of performing reliable simulation-based inference using very expensive simulators, as we demonstrate on a problem from the field of cosmology where single simulations are computationally expensive. We show that Bayesian neural networks produce informative and well-calibrated posterior estimates with only a few hundred simulations.

1 Introduction

Simulation-based inference aims at identifying the parameters of a stochastic simulator that best explain an observation. In its Bayesian formulation, simulation-based inference approximates the posterior distribution of the model parameters given an observation. This approximation usually takes the form of a neural network trained on synthetic data generated from the simulator. In the context of scientific discovery, Hermans et al. (2022) stressed the need for posterior approximations that are conservative – not overconfident – in order to make reliable downstream claims. They also showed that common simulation-based inference algorithms can produce overconfident approximations that may lead to erroneous conclusions.

In the data-poor regime (Villaescusa-Navarro et al., 2020; Zhang and Mikelsons, 2023; Zeng et al., 2023), where the simulator is expensive to run and only a small number of simulations are available, training a neural network to approximate the posterior can easily lead to overfitting. With small

[*]Equal contribution

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

amounts of training data, the neural network weights are only loosely constrained, leading to high computational uncertainty (Wenger et al., 2022). That is, many neural networks can fit the training data equally well, yet they may have very different predictions on test data. For this reason, the posterior approximation is uncertain and, in the absence of a proper quantification of this uncertainty, potentially overconfident. Fortunately, computational uncertainty in a neural network can be quantified using Bayesian neural networks (BNNs) (Gal et al., 2016), which account for the uncertainty in the neural network weights. Therefore, in the context of simulation-based inference, BNNs can provide a principled way to quantify the computational uncertainty of the posterior approximation.

Hermans et al. (2022) showed empirically that using ensembles of neural networks, a crude approximation of BNNs (Lakshminarayanan et al., 2017), does improve the calibration of the posterior approximation. A few studies have also used BNNs as density estimators in simulation-based inference (Cobb et al., 2019; Walmsley et al., 2020; Lemos et al., 2023). However, these studies have remained empirical and limited in their evaluation. This lack of theoretical grounding motivates the need for a more principled understanding of BNNs for simulation-based inference. In particular, the choice of prior on the neural network weights happens to be crucial in this context, as it can strongly influence the resulting posterior approximation. Yet, arbitrary priors that convey little or undesired information about the posterior density have been used so far.

Our contributions are twofold. Firstly, we provide an improved understanding of BNNs in the context of simulation-based inference by empirically analyzing their effect on the resulting posterior approximations. Secondly, we introduce a principled way of using BNNs in simulation-based inference by designing meaningful priors. These priors are constructed to produce calibrated posteriors even in the absence of training data. We show that they are conservative in the small-data regime, for very low simulation budgets. The code is available at `https://github.com/anonymous/anonymous`.

2 Background

Simulation-based inference We consider a stochastic simulator that takes parameters $\boldsymbol{\theta}$ as input and produces synthetic observations $\boldsymbol{x}$ as ouput. The simulator implicitly defines the likelihood $p(\boldsymbol{x}|\boldsymbol{\theta})$ in the form of a forward stochastic generative model but does not allow for direct evaluation of its density due to the intractability of the marginalization over its latent variables. In this setup, Bayesian simulation-based inference aims at approximating the posterior distribution $p(\boldsymbol{\theta}|\boldsymbol{x})$ using the simulator. Among possible approaches, *neural* simulation-based inference methods train a neural network to approximate key quantities from simulated data, such as the posterior, the likelihood, the likelihood-to-evidence ratio, or a score function (Cranmer et al., 2020).

Recently, concerns have been raised regarding the calibration of the approximate posteriors obtained with neural simulation-based inference. Hermans et al. (2022) showed that, unless special care is taken, common inference algorithms can produce overconfident posterior approximations. They quantify the calibration using the expected coverage

$$\mathrm{EC}(\hat{p}, \alpha) = \mathbb{E}_{p(\boldsymbol{\theta},\boldsymbol{x})}[\mathbb{1}(\boldsymbol{\theta} \in \boldsymbol{\Theta}_{\hat{p}}(\alpha))] \tag{1}$$

where $\boldsymbol{\Theta}_{\hat{p}}(\alpha)$ denotes the highest posterior credible region at level α computed using the posterior approximate $\hat{p}(\boldsymbol{\theta}|\boldsymbol{x})$. The expected coverage is equal to α when the posterior approximate is calibrated, lower than α when it is overconfident and higher than α when it is underconfident or conservative.

The calibration of posterior approximations has been improved in recent years in various ways. Delaunoy et al. (2022, 2023) regularize the posterior approximations to be balanced, which biases them towards conservative approximations. Similarly, Falkiewicz et al. (2024) regularize directly the posterior approximation by penalizing miscalibration or overconfidence. Masserano et al. (2023) use Neyman constructions to produce confidence regions with approximate Frequentist coverage. Patel et al. (2023) combine simulation-based inference and conformal predictions. Schmitt et al. (2023) enforce the self-consistency of likelihood and posterior approximations to improve the quality of approximate inference in low-data regimes.

Bayesian deep learning Bayesian deep learning aims to account for both the aleatoric and epistemic uncertainty in neural networks. The aleatoric uncertainty refers to the intrinsic randomness of the variable being modeled, typically taken into account by switching from a point predictor to a density estimator. The epistemic uncertainty, on the other hand, refers to the uncertainty associated with

the neural network itself and is typically high in small-data regimes. Failing to account for this uncertainty can lead to high miscalibration as many neural networks can fit the training data equally well, yet they may have very different predictions on test data.

Bayesian deep learning accounts for epistemic uncertainty by treating the neural network weights as random variables and considering the full posterior over possible neural networks instead of only the most probable neural network (Papamarkou et al., 2024). Formally, let us consider a supervised learning setting where $\boldsymbol{x}$ denotes inputs, $\boldsymbol{y}$ outputs, $\boldsymbol{D}$ a dataset of N pairs $(\boldsymbol{x}, \boldsymbol{y})$, and $\mathbf{w}$ the weights of the neural network. The likelihood of a given set of weights is

$$p(\boldsymbol{D}|\mathbf{w}) \propto \prod_{i=1}^{N} p(\boldsymbol{y}_i|\boldsymbol{x}_i, \mathbf{w}), \tag{2}$$

where $p(\boldsymbol{y}_i|\boldsymbol{x}_i, \mathbf{w})$ is the output of the neural network with weights $\mathbf{w}$ and inputs $\boldsymbol{x}_i$. The resulting posterior over the weights is

$$p(\mathbf{w}|\boldsymbol{D}) = \frac{p(\boldsymbol{D}|\mathbf{w})p(\mathbf{w})}{p(\boldsymbol{D})}, \tag{3}$$

where $p(\mathbf{w})$ is the prior. Once estimated, the posterior over the neural network's weights can be used for predictions through the Bayesian model average

$$p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{D}) = \int p(\boldsymbol{y}|\boldsymbol{x}, \mathbf{w})p(\mathbf{w}|\boldsymbol{D})d\mathbf{w} \simeq \frac{1}{M}\sum_{i=1}^{M} p(\boldsymbol{y}|\boldsymbol{x}, \mathbf{w}_i), \mathbf{w}_i \sim p(\mathbf{w}|\boldsymbol{D}). \tag{4}$$

Estimating the posterior over the neural network weights is, however, a challenging problem due to the high dimensionality of the weights. Variational inference (Blundell et al., 2015) optimizes a variational family to match the true posterior, which is typically fast but requires specifying a variational family that may restrict the functions that can be modeled. Markov chain Monte Carlo methods (Welling and Teh, 2011; Chen et al., 2014), on the other hand, are less restrictive in the functions that can be modeled but require careful tuning of the hyper-parameters and are more computationally demanding. The Bayesian posterior can also be approximated by an ensemble of neural networks (Lakshminarayanan et al., 2017; Pearce et al., 2020; He et al., 2020). Laplace methods leverage geometric information about the loss to construct an approximation of the posterior around the maximum a posteriori (MacKay, 1992). Similarly, Maddox et al. (2019) use the training trajectory of stochastic gradient descent to build an approximation of the posterior.

3 Bayesian neural networks for simulation-based inference

In the context of simulation-based inference, treating the weights of the inference network as random variables enables the quantification of the computational uncertainty of the posterior approximation. In particular, the posterior approximation $\hat{p}(\boldsymbol{\theta}|\boldsymbol{x})$ can be modeled as the Bayesian model average

$$\hat{p}(\boldsymbol{\theta}|\boldsymbol{x}) = \int p(\boldsymbol{\theta}|\boldsymbol{x}, \mathbf{w})p(\mathbf{w}|\boldsymbol{D})d\mathbf{w}, \tag{5}$$

where $p(\boldsymbol{\theta}|\boldsymbol{x}, \mathbf{w})$ is the posterior approximation parameterized by the weights $\mathbf{w}$ and evaluated at $(\boldsymbol{\theta}, \boldsymbol{x})$ and where $p(\mathbf{w}|\boldsymbol{D})$ is the posterior over the weights given the dataset $\boldsymbol{D}$.

Remaining is the choice of prior $p(\mathbf{w})$. While progress has been made in designing better priors (Fortuin, 2022) in Bayesian deep learning, we argue that none of those are suitable in the context of simulation-based inference. To illustrate our point, let us consider the case of a normal prior $p(\mathbf{w}) = \mathcal{N}(\mathbf{0}, \sigma^2\boldsymbol{I})$ on the weights, in which case

$$\hat{p}_{\text{normal prior}}(\boldsymbol{\theta}|\boldsymbol{x}) = \int p(\boldsymbol{\theta}|\boldsymbol{x}, \mathbf{w})\, \mathcal{N}(\mathbf{w}|\boldsymbol{\mu} = \mathbf{0}, \boldsymbol{\Sigma} = \sigma^2\boldsymbol{I})d\mathbf{w}. \tag{6}$$

As mentioned in Section 2, a desirable property for a posterior approximation is to be calibrated. Therefore we want $\text{EC}(\hat{p}_{\text{normal prior}}, \alpha) = \alpha, \forall\alpha$. Although it might be possible for this property to be satisfied in particular settings, this is obviously not the case for all values of σ and all neural network architectures. Therefore, and as illustrated in Figure 1, the Bayesian model average is not even calibrated a priori when using a normal prior on the weights. As the Bayesian model average is not calibrated a priori, it cannot be expected that updating the posterior over weights $p(\mathbf{w}|\boldsymbol{D})$ with a small amount of data would lead to a calibrated a posteriori Bayesian model average.

3.1 Functional priors for simulation-based inference

We design a prior that induces an a priori-calibrated Bayesian model average. To achieve this, we work in the space of posterior functions instead of the space of weights. We consider the space of functions taking a pair $(\boldsymbol{\theta}, \boldsymbol{x})$ as input and producing a posterior density value $f(\boldsymbol{\theta}, \boldsymbol{x})$ as output. Each function f is defined by the joint outputs it associates with any arbitrary set of inputs, such that a posterior over functions can be viewed as a distribution over joint outputs for arbitrary inputs. Formally, let us consider M arbitrary pairs $(\boldsymbol{\theta}, \boldsymbol{x})$ represented by the matrices $\boldsymbol{\Theta} = [\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_M]$ and $\boldsymbol{X} = [\boldsymbol{x}_1, ..., \boldsymbol{x}_M]$ and let $\boldsymbol{f} = [f_1, ..., f_M]$ be the joint outputs associated with those inputs. The distribution $p(\boldsymbol{f}|\boldsymbol{\Theta}, \boldsymbol{X})$ then represents a distribution over posteriors $\boldsymbol{f} = [\tilde{p}(\boldsymbol{\theta}_1|\boldsymbol{x}_1), ..., \tilde{p}(\boldsymbol{\theta}_M|\boldsymbol{x}_M)]$. The functional posterior distribution given a dataset $\boldsymbol{D}$ is then $p(\boldsymbol{f}|\boldsymbol{\Theta}, \boldsymbol{X}, \boldsymbol{D})$ and the Bayesian model average is obtained through marginalization, that is

$$p(\boldsymbol{\theta}_i|\boldsymbol{x}_i, \boldsymbol{D}) = \int f_i \; p(\boldsymbol{f}|\boldsymbol{\Theta}, \boldsymbol{X}, \boldsymbol{D}) d\boldsymbol{f}, \quad \forall i. \tag{7}$$

Computing the posterior over functions requires the specification of a prior over functions. We first observe that the prior over the simulator's parameters is a calibrated approximation of the posterior. That is, for the prior function $p_{\text{prior}} : (\boldsymbol{\theta}, \boldsymbol{x}) \to p(\boldsymbol{\theta})$, we have that $\text{EC}(p_{\text{prior}}, \alpha) = \alpha, \forall \alpha$ (Delaunoy et al., 2023). It naturally follows that the a priori Bayesian model average with a Dirac delta prior around the prior on the simulator's parameters is calibrated

$$\begin{aligned} \hat{p}(\boldsymbol{\theta}_i|\boldsymbol{x}_i) &= \int f_i \; \delta([f_j = p_{\text{prior}}(\boldsymbol{\theta}_j, \boldsymbol{x}_j)]) \; d\boldsymbol{f}, \forall i \\ &= \int f_i \; \delta(f_i = p(\boldsymbol{\theta}_i)) \; df_i, \forall i \Rightarrow \text{EC}(\hat{p}, \alpha) = \alpha, \forall \alpha. \end{aligned} \tag{8}$$

However, this prior has limited support, and the Bayesian model average will not converge to the posterior $p(\boldsymbol{\theta}|\boldsymbol{x})$ as the dataset size increases. We extend this Dirac prior to include more functions in its support while retaining the calibration property, which we propose defining as a Gaussian process centered at p_{prior}.

A Gaussian process (GP) defines a joint multivariate normal distribution over all the outputs $\boldsymbol{f}$ given the inputs $(\boldsymbol{\Theta}, \boldsymbol{X})$. It is parametrized by a mean function μ that defines the mean value for the outputs given the inputs and a kernel function K that models the covariance between the outputs. If we have access to no data, the mean and the kernel jointly define a prior over functions as they define a joint prior over outputs for an arbitrary set of inputs. In order for this prior over functions to be centered around the prior p_{prior}, we define the mean function as $\mu(\boldsymbol{\theta}, \boldsymbol{x}) = p(\boldsymbol{\theta})$. The kernel K, on the other hand, defines the spread around the mean function and the correlation between the outputs $\boldsymbol{f}$. Its specification is application-dependent and constitutes a hyper-parameter of our method that can be exploited to incorporate domain knowledge on the structure of the posterior. We denote the Gaussian process prior over function outputs as $p_{\text{GP}}(\boldsymbol{f}|\mu(\boldsymbol{\Theta}, \boldsymbol{X}), K(\boldsymbol{\Theta}, \boldsymbol{X}))$. Proposition 1 shows that a functional prior defined in this way leads to a calibrated Bayesian model average.

Proposition 1. *The Bayesian model average of a Gaussian process centered around the prior on the simulator's parameters is calibrated. Formally, let p_{GP} be the density probability function defined by a Gaussian process, μ its mean function, and K the kernel. Let us consider M arbitrary pairs $(\boldsymbol{\theta}, \boldsymbol{x})$ represented by the matrices $\boldsymbol{\Theta} = [\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_M]$ and $\boldsymbol{X} = [\boldsymbol{x}_1, ..., \boldsymbol{x}_M]$ and represent by the vector $\boldsymbol{f} = [f_1, ..., f_M]$ the joint outputs associated with those inputs. The Bayesian model average on the i^{th} pair is expressed*

$$\hat{p}(\boldsymbol{\theta}_i|\boldsymbol{x}_i) = \int f_i \; p_{GP}(\boldsymbol{f}|\mu(\boldsymbol{\Theta}, \boldsymbol{X}), K(\boldsymbol{\Theta}, \boldsymbol{X})) \; d\boldsymbol{f}$$

If $\mu(\boldsymbol{\theta}, \boldsymbol{x}) = p(\boldsymbol{\theta}), \forall \boldsymbol{\theta}, \boldsymbol{x}$, then,

$$\text{EC}(\hat{p}, \alpha) = \alpha, \forall \alpha,$$

for all kernel K.

Proof. As p_{GP} is, by definition of a Gaussian process, a multivariate normal, the expectations of the marginals are equal to the mean parameters

$$\hat{p}(\boldsymbol{\theta}_i|\boldsymbol{x}_i) = \mu(\boldsymbol{\theta}_i, \boldsymbol{x}_i) = p(\boldsymbol{\theta}_i).$$

The joint evaluation of the Bayesian model average of the Gaussian process is hence equivalent to the joint evaluation of the prior for any matrices $\boldsymbol{\Theta}$ and $\boldsymbol{X}$. We can therefore conclude that $\hat{p}$ is equivalent to $p_{\text{prior}} : (\boldsymbol{\theta}, \boldsymbol{x}) \to p(\boldsymbol{\theta})$. Since $\text{EC}(p_{\text{prior}}, \alpha) = \alpha, \forall \alpha$ (Delaunoy et al., 2023), then, $\text{EC}(\hat{p}, \alpha) = \alpha, \forall \alpha$. □

3.2 From functional to parametric priors

In the remainder of this section, we discuss how the GP prior over posterior density functions can be used in practice to perform Bayesian inference over neural networks in the simulation-based inference setting. Let us first observe that a neural network architecture and a prior on weights jointly define a prior over functions. We parameterize the prior on weights by $\boldsymbol{\phi}$ and denote this probability density over function outputs by

$$\begin{aligned} p_{\text{BNN}}(\boldsymbol{f} \mid \boldsymbol{\phi}, \boldsymbol{\Theta}, \boldsymbol{X}) &= \int p(\boldsymbol{f} \mid \mathbf{w}, \boldsymbol{\Theta}, \boldsymbol{X})\, p(\mathbf{w}|\boldsymbol{\phi})\, d\mathbf{w} \\ &= \int \delta([f_i = p(\boldsymbol{\theta}_i|\boldsymbol{x}_i, \mathbf{w})])\, p(\mathbf{w}|\boldsymbol{\phi})\, d\mathbf{w}. \end{aligned} \tag{9}$$

To obtain a prior on weights that matches the target GP prior, we optimize $\boldsymbol{\phi}$ such that $p_{\text{BNN}}(\boldsymbol{f} \mid \boldsymbol{\phi}, \boldsymbol{\Theta}, \boldsymbol{X})$ matches $p_{\text{GP}}(\boldsymbol{f}|\mu(\boldsymbol{\Theta}, \boldsymbol{X}), K(\boldsymbol{\Theta}, \boldsymbol{X}))$. Following Flam-Shepherd et al. (2017), given a measurement set $\mathcal{M} = \{\boldsymbol{\theta}_i, \boldsymbol{x}_i\}_{i=1}^M$ at which we want the distributions to match, the KL divergence between the two priors can be expressed as

$$\begin{aligned} &\text{KL}\left[p_{\text{BNN}}(\boldsymbol{f} \mid \boldsymbol{\phi}, \mathcal{M}) \,||\, p_{\text{GP}}(\boldsymbol{f} \mid \mu(\mathcal{M}), K(\mathcal{M}))\right] \\ &= \int p_{\text{BNN}}(\boldsymbol{f} \mid \boldsymbol{\phi}, \mathcal{M}) \log \frac{p_{\text{BNN}}(\boldsymbol{f} \mid \boldsymbol{\phi}, \mathcal{M})}{p_{\text{GP}}(\boldsymbol{f} \mid \mu(\mathcal{M}), K(\mathcal{M}))} d\boldsymbol{y} \\ &= -\mathbb{H}\left[p_{\text{BNN}}(\boldsymbol{f} \mid \boldsymbol{\phi}, \mathcal{M})\right] - \mathbb{E}_{p_{\text{BNN}}(\boldsymbol{f} \mid \boldsymbol{\phi}, \mathcal{M})}\left[\log p_{\text{GP}}(\boldsymbol{f} \mid \mu(\mathcal{M}), K(\mathcal{M}))\right], \end{aligned} \tag{10}$$

where the second term $\mathbb{E}_{p_{\text{BNN}}(\boldsymbol{f} \mid \boldsymbol{\phi}, \mathcal{M})}\left[\log p_{\text{GP}}(\boldsymbol{f} \mid \mu(\mathcal{M}), K(\mathcal{M}))\right]$ can be estimated using Monte-Carlo. The first term $\mathbb{H}\left[p_{\text{BNN}}(\boldsymbol{f} \mid \boldsymbol{\phi}, \mathcal{M})\right]$, however, is harder to estimate as it requires computing $\log p_{\text{BNN}}(\boldsymbol{f} \mid \boldsymbol{\phi}, \mathcal{M})$, which involves the integration of the output over all possible weights combinations. To bypass this issue, Sun et al. (2018) propose to use Spectral Stein Gradient Estimation (SSGE) (Shi et al., 2018) to approximate the gradient of the entropy as

$$\nabla \mathbb{H}\left[p_{\text{BNN}}(\boldsymbol{f} \mid \boldsymbol{\phi}, \mathcal{M})\right] \simeq \text{SSGE}\left(\boldsymbol{f}_1, ..., \boldsymbol{f}_N \sim p_{\text{BNN}}(\boldsymbol{f} \mid \boldsymbol{\phi}, \mathcal{M})\right). \tag{11}$$

We note that the measurement set $\mathcal{M}$ can be chosen arbitrarily but should cover most of the support of the joint distribution $p(\boldsymbol{\theta}, \boldsymbol{x})$. If data from this joint distribution are available, those can be leveraged to build the measurement set. To showcase the ability to create a prior with limited data, in this work, we derive boundaries of the support of each marginal distribution and draw parameters and observations independently and uniformly over this support. If the support is known a priori, this procedure can be performed without (expensive) simulations. We draw a new measurement set at each iteration of the optimization procedure. If a fixed measurement set is available, a subsample of this measurement set should be drawn at each iteration. Also note that other methods can be used in practice to perform inference on the neural network's weights with our GP prior. Those are described in Appendix A.

As an illustrative example, we chose independent normal distributions as a variational family $p(\mathbf{w}|\boldsymbol{\phi})$ over the weights and minimize (10) w.r.t. $\mathbf{w}$. In Figure 1, we show the coverage of the resulting a priori Bayesian model average using the tuned prior, $p(\mathbf{w} \mid \boldsymbol{\phi})$, and normal priors for increasing standard deviations σ, for the SLCP benchmark. We observe that while none of the normal priors are calibrated, the trained prior achieves near-perfect calibration. This prior hence guides the obtained posterior approximation towards more calibrated solutions, even in low simulation-budget settings.

The attentive reader might have noticed that $p_{\text{BNN}}(\boldsymbol{f} \mid \boldsymbol{\phi}, \boldsymbol{\Theta}, \boldsymbol{X})$ and $p_{\text{GP}}(\boldsymbol{f}|\mu(\boldsymbol{\Theta}, \boldsymbol{X}), K(\boldsymbol{\Theta}, \boldsymbol{X}))$ do not share the same support, as the former distribution is limited to functions that represent valid densities by construction, while the latter includes arbitrarily shaped functions. This is not an issue here as the support of the first distribution is included in the support of the second distribution, and functions outside the support of the first distribution are ignored in the computation of the divergence.

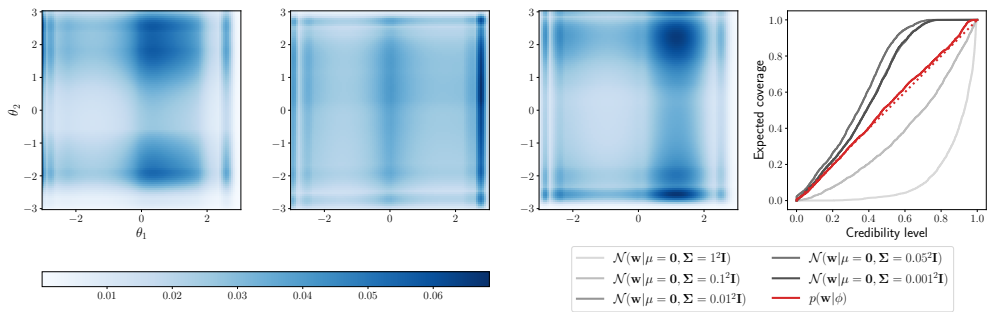

Figure 1: Visualization of the prior tuned to match the GP prior on the SLCP benchmark. Left: examples of posterior functions sampled from the tuned prior over neural network's weights. Right: expected coverage of the prior Bayesian model average with the tuned prior and normal priors for varying standard deviations.

4 Experiments

In this section, we empirically demonstrate the benefits of replacing a regular neural network with a BNN equipped with the proposed prior for simulation-based inference. We consider both Neural Posterior Estimation (NPE) with neural spline flows (Durkan et al., 2019) and Neural Ratio Estimation (NRE) (Hermans et al., 2020), along with their balanced versions (BNRE and BNPE) (Delaunoy et al., 2022, 2023) and ensembles (Lakshminarayanan et al., 2017; Hermans et al., 2022). BNNs-based methods are trained using mean-field variational inference (Blundell et al., 2015). As advocated by Wenzel et al. (2020), we also consider cold posteriors to achieve good predictive performance. More specifically, the variational objective function is modified to give less weight to the prior by introducing a temperature parameter T,

$$\mathbb{E}_{\mathbf{w}\sim p(\mathbf{w}|\boldsymbol{\tau})}\left[\sum_i \log p(\boldsymbol{\theta}_i|\boldsymbol{x}_i,\mathbf{w})\right] - T\ \mathrm{KL}[p(\mathbf{w}|\boldsymbol{\tau})||p(\mathbf{w}|\boldsymbol{\phi})], \tag{12}$$

where $\boldsymbol{\tau}$ are the parameters of the posterior variational family and T is a parameter called the temperature that weights the prior term. In the following, we call BNN-NPE, a Bayesian Neural Network posterior estimator trained without temperature ($T = 1$), and BNN-NPE ($T = 0.01$), an estimator trained with a temperature of 0.01, assigning a lower weight to the prior.

A detailed description of the Gaussian process used can be found in Appendix A. For simplicity, in this analysis, we use an RBF kernel in the GP prior. If more information on the structure of the target posterior is available, more informed kernels may be used to leverage this prior knowledge. A description of the benchmarks can be found in Appendix B, and the hyperparameters are described in Appendix C. For clarity, only NPE-based methods are shown in this section; results using NRE can be found in Appendix D.

Following Delaunoy et al. (2022), we evaluate the quality of the posterior approximations based on the expected nominal log posterior density and the expected coverage area under the curve (coverage AUC). The expected nominal log posterior density $\mathbb{E}_{\boldsymbol{\theta},\boldsymbol{x}\sim p(\boldsymbol{\theta},\boldsymbol{x})}\left[\log \hat{p}(\boldsymbol{\theta}|\boldsymbol{x})\right]$ quantifies the amount of density allocated to the nominal parameter that was used to generate the observation. The coverage AUC $\int_0^1 (\mathrm{EC}(\hat{p},\alpha)-\alpha)\, d\alpha$ quantifies the calibration of the expected posterior. A calibrated posterior approximation exhibits a coverage AUC of 0. A positive coverage AUC indicates conservativeness, and a negative coverage AUC indicates overconfidence.

BNN-based simulation-based inference Figure 2 compares simulation-based inference methods with and without accounting for computational uncertainty We observe that BNNs equipped with our prior and without temperature show positive coverage AUC even for simulation budgets as low as $O(10)$. The coverage curves are reported in Appendix D and show that this corresponds to conservative posterior approximations. We conclude that BNNs can hence be reliably used when the simulator is expensive and few simulations are available. We also observe that the nominal log posterior density is on par with other methods for very high simulation budgets but that more samples

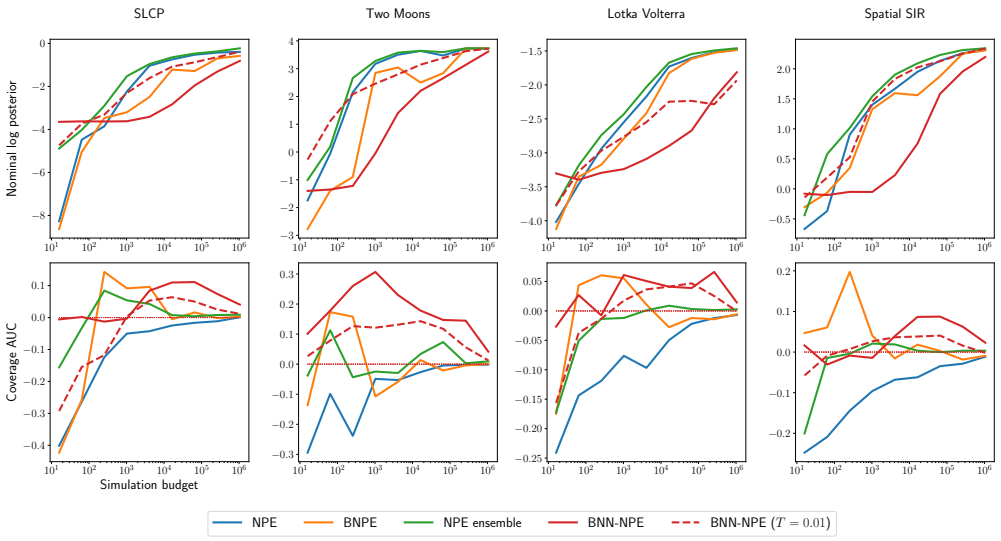

Figure 2: Comparison of different simulation-based inference methods through the nominal log probability and coverage area under the curve. The higher the nominal log probability, the more performant the method is. A calibrated posterior approximation exhibits a coverage AUC of 0. A positive coverage AUC indicates conservativeness, and a negative coverage AUC indicates overconfidence. 3 runs are performed, and the median is reported.

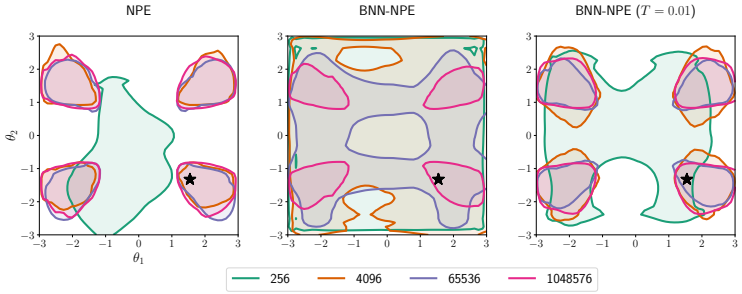

Figure 3: Examples of 95% highest posterior density regions obtained with various algorithms and simulation budgets on the SLCP benchmark for a single observation. The black star represents the ground truth used to generate the observation.

are required to achieve high values. Cold posteriors can help achieve high nominal log posterior values with fewer samples at the cost of sometimes producing overconfident posterior approximations.

Examples of posterior approximations obtained with and without using a Bayesian neural network are shown in Figure 3. Wide posteriors are observed for low budgets for BNN-NPE, while NPE produces an overconfident approximation and excludes most of the relevant parts of the posterior. As the simulation budget increases, BNN-NPE converges slowly towards the same posterior as NPE. BNN-NPE ($T = 0.01$) converges faster than BNN-NPE but, for low simulation budgets, excludes parts of the region that should be accepted according to high budget posteriors. Yet, the posterior approximate is still less overconfident than NPE's.

Comparison of different priors on weights We analyze the effect of the prior on the neural network's weights on the resulting posterior approximation. The posterior approximations obtained using our GP prior are compared to the ones obtained using independent normal priors on weights with zero means and increasing standard deviations. In Figure 4, we observe that when using a normal prior, careful tuning of the standard deviation is needed to achieve results close to the prior designed for simulation-based inference. The usage of an inappropriate prior can lead to bad calibration for low simulation budgets or can prevent learning if it is too restrictive.

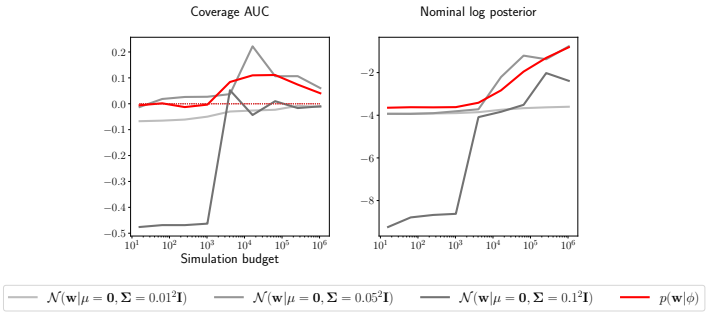

Figure 4: Comparison of posterior approximations obtained using a prior tuned to match the Gaussian process-based prior and using independent normal priors on weights with zero means and various standard deviations on the SLCP benchmark. 3 runs are performed, and the median is reported.

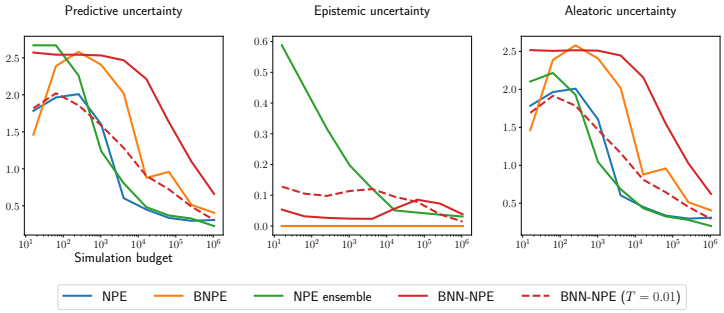

Figure 5: Quantification of the different forms of uncertainties captured by the different NPE-based methods on the SLCP benchmark. 3 runs are performed, and the median is reported.

Uncertainty decomposition We decompose the uncertainty quantified by the different methods. Following Depeweg et al. (2018), the uncertainty can be decomposed as

$$\mathbb{H}\left[\hat{p}(\boldsymbol{\theta}|\boldsymbol{x})\right] = \mathbb{E}_{q(\mathbf{w})}\left[\mathbb{H}\left[\hat{p}(\boldsymbol{\theta}|\boldsymbol{x},\mathbf{w})\right]\right] + \mathbb{I}(\boldsymbol{\theta},\mathbf{w}), \tag{13}$$

where $\mathbb{E}_{q(\mathbf{w})}\left[\mathbb{H}\left[\hat{p}(\boldsymbol{\theta}|\boldsymbol{x},\mathbf{w})\right]\right]$ quantifies the aleatoric uncertainty, $\mathbb{I}(\boldsymbol{\theta},\mathbf{w})$ quantifies the epistemic uncertainty, and the sum of those terms is the predictive uncertainty. Figure 5 shows the decomposition of the two sources of uncertainty, in expectation, on the SLCP benchmark. Other benchmarks can be found in Appendix D. We observe that BNN-NPE and NPE ensemble methods account for the epistemic uncertainty while other methods do not. BNPE artificially increases the aleatoric uncertainty to be better calibrated. The epistemic uncertainty of BNN-NPE is initially low because most of the models are slight variations of $p_{\boldsymbol{\Theta}}$. The epistemic uncertainty then increases as it starts to deviate from the prior and decreases as the training set size increases. BNN-NPE ($T = 0.01$) exhibits a higher epistemic uncertainty for low budgets as the effect of the prior is lowered.

Infering cosmological parameters from N-body simulations To showcase the utility of Bayesian deep learning for simulation-based inference in a practical setting, we consider a challenging inference problem from the field of cosmology. We consider *Quijote* N-body simulations (Villaescusa-Navarro et al., 2020) tracing the spatial distribution of matter in the Universe for different underlying cosmological models. The resulting observations are particles with different masses, corresponding to dark matter clumps, which host galaxies. We consider the canonical task of inferring the matter density (denoted Ω_m) and the root-mean-square matter fluctuation averaged over a sphere of radius $8h^{-1}$ Mpc (denoted σ_8) from an observed galaxy field. Robustly inferring the values of these parameters is one of the scientific goals of flagship cosmological surveys. These simulations are very computationally expensive to run, with over 35 million CPU hours required to generate 44100 simulations at a relatively low resolution. Generating samples at higher resolutions, or a significantly larger number of samples, is challenging due to computational constraints. These constraints necessitate methods that can be used to produce reliable scientific conclusions from a limited set of simulations – when

few simulations are available, not only is the amount of training data low, but so is the amount of test data that is available to assess the calibration of the trained model.

In this experiment, we use 2000 simulations processed as described in Cuesta-Lazaro and Mishra-Sharma (2023). These simulations form a subset of the full simulation suite run with a uniform prior over the parameters of interest. 1800 simulations are used for training and 200 are kept for testing. We use the two-point correlation function evaluated at 24 distance bins as a summary statistic. The observable is, hence, a vector of 24 features. Figure 6 compares the posterior approximations obtained with a single neural network against those obtained with a BNN trained with a temperature of 0.01. We observe from the coverage plots that while a single neural network can lead to overconfident approximations in the data-poor regime, the BNN leads to conservative approximations. BNN-NPE also exhibits higher nominal log posterior probability. Additionally, we observe that it provides posterior approximations that are calibrated and have a high nominal log probability with only a few hundred samples.

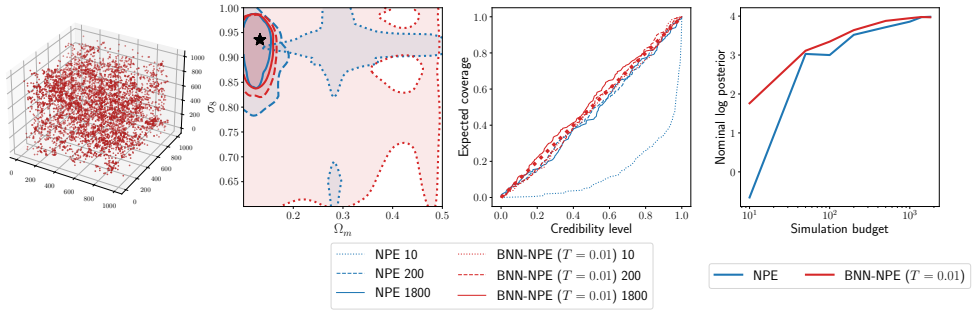

Figure 6: Comparison of the posterior approximations obtained with and without a Bayesian neural network on the cosmological application. First plot: An example observation: particles representing galaxies in a synthetic universe. Second plot: example of 95% highest posterior density regions for increasing simulation budgets. The black star represents the ground truth used to generate the observation. Third plot: Expected coverage with and without using a Bayesian neural network for increasing simulation budgets. Fourth plot: The nominal log posterior.

5 Conclusion

In this work, we use Bayesian deep learning to account for the computational uncertainty associated with posterior approximations in simulation-based inference. We show that the prior on neural network's weights should be carefully chosen to obtain calibrated posterior approximations and develop a prior family with this objective in mind. The prior family is defined in function space as a Gaussian process and mapped to a prior on weights. Empirical results on benchmarks show that incorporating Bayesian neural networks in simulation-based inference methods consistently yields conservative posterior approximations, even with limited simulation budgets of $\mathcal{O}(10)$. As Bayesian deep learning continues to rapidly advance (Papamarkou et al., 2024), we anticipate that future developments will strengthen its applicability in simulation-based inference, ultimately enabling more efficient and reliable scientific applications in domains with computationally expensive simulators.

Acknowledgments and Disclosure of Funding

Use unnumbered first level headings for the acknowledgments. All acknowledgments go at the end of the paper before the list of references. Moreover, you are required to declare funding (financial activities supporting the submitted work) and competing interests (related financial activities outside the submitted work). More information about this disclosure can be found at: `https://neurips.cc/Conferences/2024/PaperInformation/FundingDisclosure`.

Do **not** include this section in the anonymized submission, only in the final paper. You can use the `ack` environment provided in the style file to automatically hide this section in the anonymized submission.

References

Blundell, C., Cornebise, J., Kavukcuoglu, K., and Wierstra, D. (2015). Weight uncertainty in neural network. In *International conference on machine learning*, pages 1613–1622. PMLR.

Chen, T., Fox, E., and Guestrin, C. (2014). Stochastic gradient hamiltonian monte carlo. In *International conference on machine learning*, pages 1683–1691. PMLR.

Cobb, A. D., Himes, M. D., Soboczenski, F., Zorzan, S., O'Beirne, M. D., Baydin, A. G., Gal, Y., Domagal-Goldman, S. D., Arney, G. N., Angerhausen, D., et al. (2019). An ensemble of bayesian neural networks for exoplanetary atmospheric retrieval. *The astronomical journal*, 158(1):33.

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

Cuesta-Lazaro, C. and Mishra-Sharma, S. (2023). A point cloud approach to generative modeling for galaxy surveys at the field level. *arXiv preprint arXiv:2311.17141*.

Delaunoy, A., Hermans, J., Rozet, F., Wehenkel, A., and Louppe, G. (2022). Towards reliable simulation-based inference with balanced neural ratio estimation. *Advances in Neural Information Processing Systems*, 35:20025–20037.

Delaunoy, A., Miller, B. K., Forré, P., Weniger, C., and Louppe, G. (2023). Balancing simulation-based inference for conservative posteriors. In *Fifth Symposium on Advances in Approximate Bayesian Inference*.

Depeweg, S., Hernandez-Lobato, J.-M., Doshi-Velez, F., and Udluft, S. (2018). Decomposition of uncertainty in bayesian deep learning for efficient and risk-sensitive learning. In *International conference on machine learning*, pages 1184–1193. PMLR.

Durkan, C., Bekasov, A., Murray, I., and Papamakarios, G. (2019). Neural spline flows. *Advances in neural information processing systems*, 32.

Falkiewicz, M., Takeishi, N., Shekhzadeh, I., Wehenkel, A., Delaunoy, A., Louppe, G., and Kalousis, A. (2024). Calibrating neural simulation-based inference with differentiable coverage probability. *Advances in Neural Information Processing Systems*, 36.

Flam-Shepherd, D., Requeima, J., and Duvenaud, D. (2017). Mapping gaussian process priors to bayesian neural networks. In *NIPS Bayesian deep learning workshop*, volume 3.

Fortuin, V. (2022). Priors in bayesian deep learning: A review. *International Statistical Review*, 90(3):563–591.

Gal, Y. et al. (2016). Uncertainty in deep learning.

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

He, B., Lakshminarayanan, B., and Teh, Y. W. (2020). Bayesian deep ensembles via the neural tangent kernel. *Advances in neural information processing systems*, 33:1010–1022.

Hermans, J., Begy, V., and Louppe, G. (2020). Likelihood-free mcmc with amortized approximate ratio estimators. In *International conference on machine learning*, pages 4239–4248. PMLR.

Hermans, J., Delaunoy, A., Rozet, F., Wehenkel, A., Begy, V., and Louppe, G. (2022). A crisis in simulation-based inference? beware, your posterior approximations can be unfaithful. *Transactions on Machine Learning Research*.

Kozyrskiy, B., Milios, D., and Filippone, M. (2023). Imposing functional priors on bayesian neural networks. In *ICPRAM 2023, 12th International Conference on Pattern Recognition Applications and Methods*.

Lakshminarayanan, B., Pritzel, A., and Blundell, C. (2017). Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30.

Lemos, P., Cranmer, M., Abidi, M., Hahn, C., Eickenberg, M., Massara, E., Yallup, D., and Ho, S. (2023). Robust simulation-based inference in cosmology with bayesian neural networks. *Machine Learning: Science and Technology*, 4(1):01LT01.

Lotka, A. J. (1920). Analytical note on certain rhythmic relations in organic systems. *Proceedings of the National Academy of Sciences*, 6(7):410–415.

Ma, C. and Hernández-Lobato, J. M. (2021). Functional variational inference based on stochastic process generators. *Advances in Neural Information Processing Systems*, 34:21795–21807.

MacKay, D. J. (1992). Bayesian interpolation. *Neural computation*, 4(3):415–447.

Maddox, W. J., Izmailov, P., Garipov, T., Vetrov, D. P., and Wilson, A. G. (2019). A simple baseline for bayesian uncertainty in deep learning. *Advances in neural information processing systems*, 32.

Masserano, L., Dorigo, T., Izbicki, R., Kuusela, M., and Lee, A. B. (2023). Simulator-based inference with waldo: Confidence regions by leveraging prediction algorithms and posterior estimators for inverse problems. *Proceedings of Machine Learning Research*, 206.

Papamakarios, G., Sterratt, D., and Murray, I. (2019). Sequential neural likelihood: Fast likelihood-free inference with autoregressive flows. In *The 22nd international conference on artificial intelligence and statistics*, pages 837–848. PMLR.

Papamarkou, T., Skoularidou, M., Palla, K., Aitchison, L., Arbel, J., Dunson, D., Filippone, M., Fortuin, V., Hennig, P., Hubin, A., et al. (2024). Position paper: Bayesian deep learning in the age of large-scale ai. *arXiv preprint arXiv:2402.00809*.

Patel, Y., McNamara, D., Loper, J., Regier, J., and Tewari, A. (2023). Variational inference with coverage guarantees. *arXiv preprint arXiv:2305.14275*.

Pearce, T., Leibfried, F., and Brintrup, A. (2020). Uncertainty in neural networks: Approximately bayesian ensembling. In *International conference on artificial intelligence and statistics*, pages 234–244. PMLR.

Rudner, T. G., Chen, Z., Teh, Y. W., and Gal, Y. (2022). Tractable function-space variational inference in bayesian neural networks. *Advances in Neural Information Processing Systems*, 35:22686–22698.

Schmitt, M., Habermann, D., Bürkner, P.-C., Köthe, U., and Radev, S. T. (2023). Leveraging self-consistency for data-efficient amortized bayesian inference. *arXiv preprint arXiv:2310.04395*.

Shi, J., Sun, S., and Zhu, J. (2018). A spectral approach to gradient estimation for implicit distributions. In *International Conference on Machine Learning*, pages 4644–4653. PMLR.

Sun, S., Zhang, G., Shi, J., and Grosse, R. (2018). Functional variational bayesian neural networks. In *International Conference on Learning Representations*.

Tran, B.-H., Rossi, S., Milios, D., and Filippone, M. (2022). All you need is a good functional prior for bayesian deep learning. *The Journal of Machine Learning Research*, 23(1):3210–3265.

Villaescusa-Navarro, F., Hahn, C., Massara, E., Banerjee, A., Delgado, A. M., Ramanah, D. K., Charnock, T., Giusarma, E., Li, Y., Allys, E., et al. (2020). The quijote simulations. *The Astrophysical Journal Supplement Series*, 250(1):2.

Volterra, V. (1926). Fluctuations in the abundance of a species considered mathematically. *Nature*, 118(2972):558–560.

Walmsley, M., Smith, L., Lintott, C., Gal, Y., Bamford, S., Dickinson, H., Fortson, L., Kruk, S., Masters, K., Scarlata, C., et al. (2020). Galaxy zoo: probabilistic morphology through bayesian cnns and active learning. *Monthly Notices of the Royal Astronomical Society*, 491(2):1554–1574.

Welling, M. and Teh, Y. W. (2011). Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 681–688. Citeseer.

Wenger, J., Pleiss, G., Pförtner, M., Hennig, P., and Cunningham, J. P. (2022). Posterior and computational uncertainty in gaussian processes. *Advances in Neural Information Processing Systems*, 35:10876–10890.

Wenzel, F., Roth, K., Veeling, B., Swiatkowski, J., Tran, L., Mandt, S., Snoek, J., Salimans, T., Jenatton, R., and Nowozin, S. (2020). How good is the bayes posterior in deep neural networks really? In *International Conference on Machine Learning*, pages 10248–10259. PMLR.

Zeng, J., Todd, M. D., and Hu, Z. (2023). Probabilistic damage detection using a new likelihood-free bayesian inference method. *Journal of Civil Structural Health Monitoring*, 13(2):319–341.

Zhang, Y. and Mikelsons, L. (2023). Sensitivity-guided iterative parameter identification and data generation with bayesflow and pels-vae for model calibration. *Advanced Modeling and Simulation in Engineering Sciences*, 10(1):9.

A Prior tuning details

We tune the parameters $\boldsymbol{\phi}$ of a variational distribution over neural network weights $p(\mathbf{w}|\boldsymbol{\phi})$. The variational distribution is chosen to be independent normal distributions, with parameters $\boldsymbol{\phi}$ representing the means and standard deviations of each parameter of $\mathbf{w}$. This variational family defines a prior over function outputs

$$p_{\text{BNN}}(\boldsymbol{f} \mid \boldsymbol{\phi}, \boldsymbol{\Theta}, \boldsymbol{X}) = \int p(\boldsymbol{f} \mid \mathbf{w}, \boldsymbol{\Theta}, \boldsymbol{X}) p(\mathbf{w}|\boldsymbol{\phi}) d\mathbf{w}. \tag{14}$$

The parameters $\boldsymbol{\phi}$ are optimized to obtain a prior on weights that matches the target Gaussian process functional prior $p_{\text{GP}}(\boldsymbol{f}|\mu(\boldsymbol{\Theta}, \boldsymbol{X}), K(\boldsymbol{\Theta}, \boldsymbol{X}))$. To achieve this, we repeatedly sample a measurement set $\mathcal{M} = \{\boldsymbol{\theta}_i, \boldsymbol{x}_i\}_{i=1}^{M}$ and N function outputs from the BNN prior $\boldsymbol{f}_1, ..., \boldsymbol{f}_N \sim p_{\text{BNN}}(\boldsymbol{f} \mid \boldsymbol{\phi}, \mathcal{M})$ and perform a step of gradient descend to minimize the divergence

$$\text{KL}\left[p_{\text{BNN}}(\boldsymbol{f} \mid \boldsymbol{\phi}, \mathcal{M}) \,||\, p_{\text{GP}}(\boldsymbol{f} \mid \mu(\mathcal{M}), K(\mathcal{M}))\right]. \tag{15}$$

The mean function μ of the Gaussian process is selected as:

$$\mu(\boldsymbol{\theta}, \boldsymbol{x}) = p(\boldsymbol{\theta}). \tag{16}$$

The kernel K is a combination of two Radial Basis Function (RBF) kernels

$$K(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \boldsymbol{x}_1, \boldsymbol{x}_2) = \sqrt{\text{RBF}(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2)} * \sqrt{\text{RBF}(\boldsymbol{x}_1, \boldsymbol{x}_2)}. \tag{17}$$

such that the correlation between outputs is high only if $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$ as well as $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are close. The RBF kernel is defined as

$$\text{RBF}(\boldsymbol{x}_1, \boldsymbol{x}_2) = \sigma^2 \exp\left(-\frac{1}{N}\sum_{i}^{N} \frac{(x_{1,i} - x_{2,i})^2}{2l_i^2}\right), \tag{18}$$

where σ is the standard deviation and l_i is the lengthscale associated to the i^{th} feature. The lengthscale is derived from the measurement set. To determine l_i, we query observations $\boldsymbol{x}$ from the measurement set and compute the 0.1 quantile of the squared distance between different observations for each feature. We then set l_i such that $2l_i^2$ equals this quantile. All the benchmarks have a uniform prior over the simulator's parameters. The mean function is then equal to a constant C for all input values. The standard deviation is chosen to be $C/2$. To ensure stability during the inference procedure, we enforce all standard deviations defined in $\boldsymbol{\phi}$ to be at least 0.001 by setting any parameters below this threshold to this value.

Note that there are various methods that can be used to perform inference on the neural network's weights with our GP prior. Instead of minimizing the KL-divergence, the parameters $\boldsymbol{\phi}$ can be optimized using an adversarial training procedure by treating both priors as function generators and training a discriminator between the two (Tran et al., 2022). Another approach to performing inference using a functional prior is to directly use it during inference by modifying the inference algorithm to work in function space. Variational inference can be performed in the space of function (Sun et al., 2018; Rudner et al., 2022). The stochastic gradient Hamiltonian Monte Carlo algorithm (Chen et al., 2014) could also be modified to include a functional prior Kozyrskiy et al. (2023). Alternatively, a variational implicit process can be learned to express the posterior in function space (Ma and Hernández-Lobato, 2021).

B Benchmarks description

SLCP The SLCP (Simple Likelihood Complex Posterior) benchmark (Papamakarios et al., 2019) is a fictive benchmark that takes 5 parameters as input and produces an 8-dimensional synthetic observable. The observation corresponds to the 2D coordinates of 4 points that are sampled from the same multivariate normal distribution. We consider the task of inferring the marginal over 2 of the 5 parameters.

Two Moons The Two Moons simulator (Greenberg et al., 2019) models a fictive problem with 2 parameters. The observable x is composed of 2 scalars, which represent the 2D coordinates of a random point sampled from a crescent-shaped distribution shifted and rotated around the origin depending on the parameters' values. Those transformations involve the absolute value of the sum of the parameters leading to a second crescent in the posterior and, hence making it multi-modal.

Lotka Volterra The Lotka-Volterra population model (Lotka, 1920; Volterra, 1926) describes a process of interactions between a predator and a prey species. The model is conditioned on 4 parameters that influence the reproduction and mortality rate of the predator and prey species. We infer the marginal posterior of the predator parameters from a time series of 2001 steps representing the evolution of both populations over time. The specific implementation is based on a Markov Jump Process, as in Papamakarios et al. (2019).

SpatialSIR The Spatial SIR model (Hermans et al., 2022) involves a grid world of susceptible, infected, and recovered individuals. Based on initial conditions and the infection and recovery rate, the model describes the spatial evolution of an infection. The observable is a snapshot of the grid world after some fixed amount of time. The grid used is of size 50 by 50.

C Hyperparameters

All the NPE-based methods use a Neural Spline Flow (NSF) (Durkan et al., 2019) with 3 transforms of 6 layers, each containing 256 neurons. Meanwhile, all the NRE-based methods employ a classifier consisting of 6 layers of 256 neurons. For the spatialSIR and Lotka Volterra benchmarks, the observable is initially processed by an embedding network. Lotka Volterra's embedding network is a 10 layers 1D convolutional neural network that leads to an embedding of size 512. On the other hand, SpatialSIR's embedding network is an 8 layers 2D convolutional neural network resulting in an embedding of size 256.

Bayesian neural network-based methods use independent normal distributions as a variational family. During inference, 100 neural networks are sampled to approximate the Bayesian model average. Ensemble methods involve training 5 neural networks independently. The experiments were conducted on a private GPU cluster, and the estimated computational cost is around $25,000$ GPU hours.

D Additional experiments

In this section, we provide complementary results. Figure 7 illustrates the performance of the various NRE variants. Figures 8 and 9 display the coverage curves, demonstrating that a higher positive coverage AUC corresponds to coverage curves above the diagonal line. Figures 10 and 11 present the uncertainty decomposition of all methods on all the benchmarks.

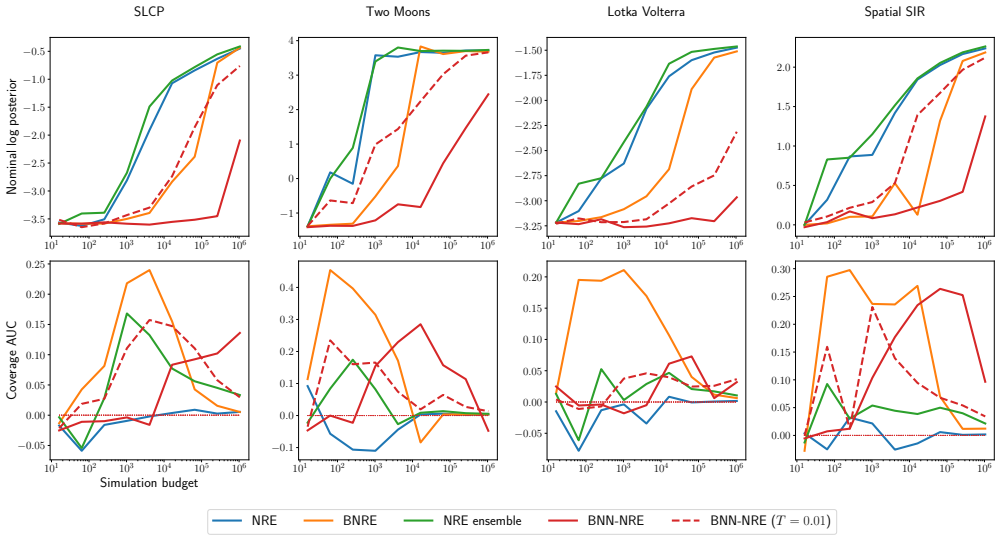

Figure 7: Comparison of different NRE simulation-based inference methods through the nominal log probability and coverage area under the curve. The higher the nominal log probability, the more performant the method is. A calibrated posterior approximation exhibits a coverage AUC of 0. A positive coverage AUC indicates conservativeness, and a negative coverage AUC indicates overconfidence. 3 runs are performed, and the median is reported

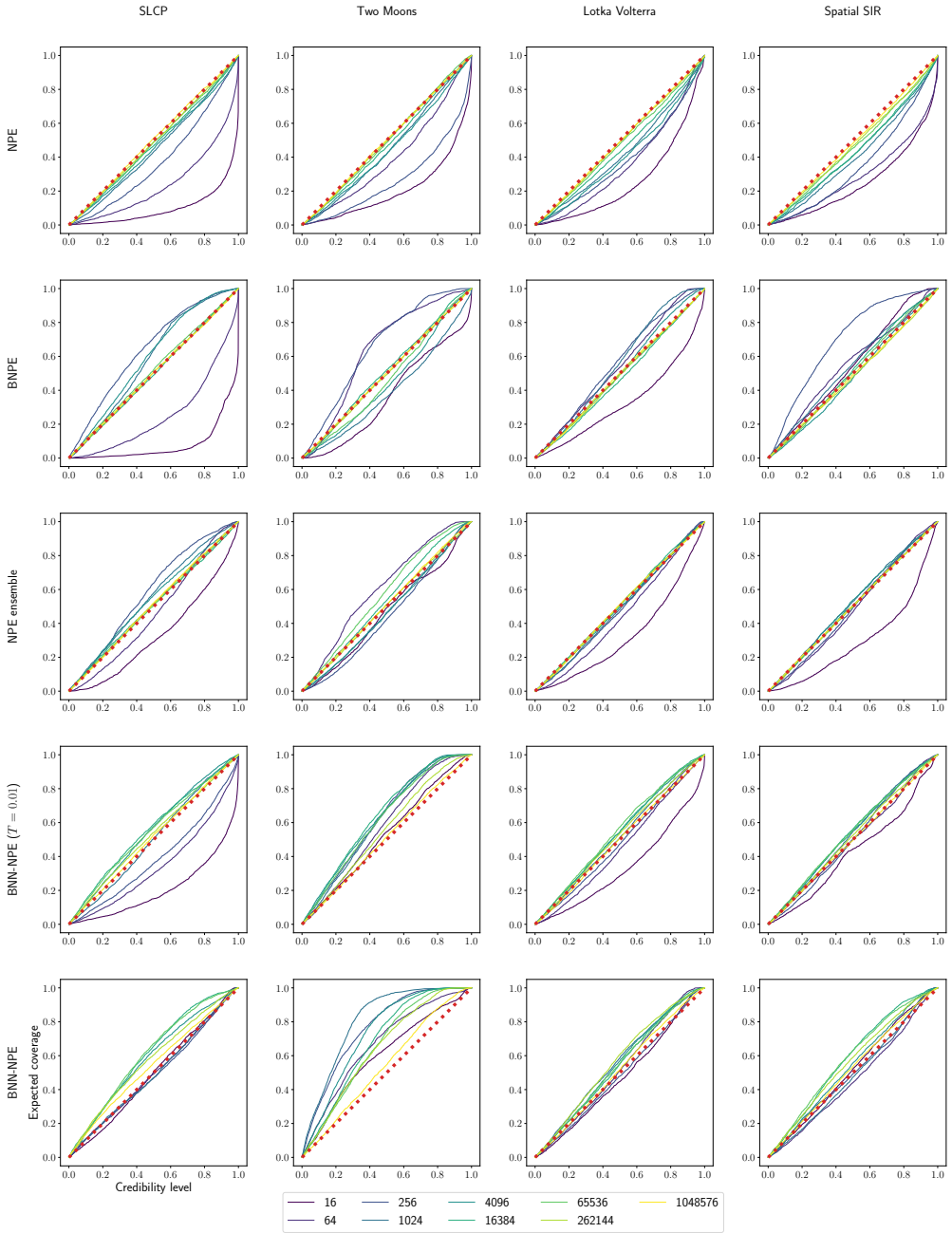

Figure 8: Coverage of different NPE simulation-based inference methods. A calibrated posterior approximation exhibits a coverage AUC of 0. A coverage curve above the diagonal indicates conservativeness and a curve below the diagonal indicates overconfidence. 3 runs are performed, and the median is reported.

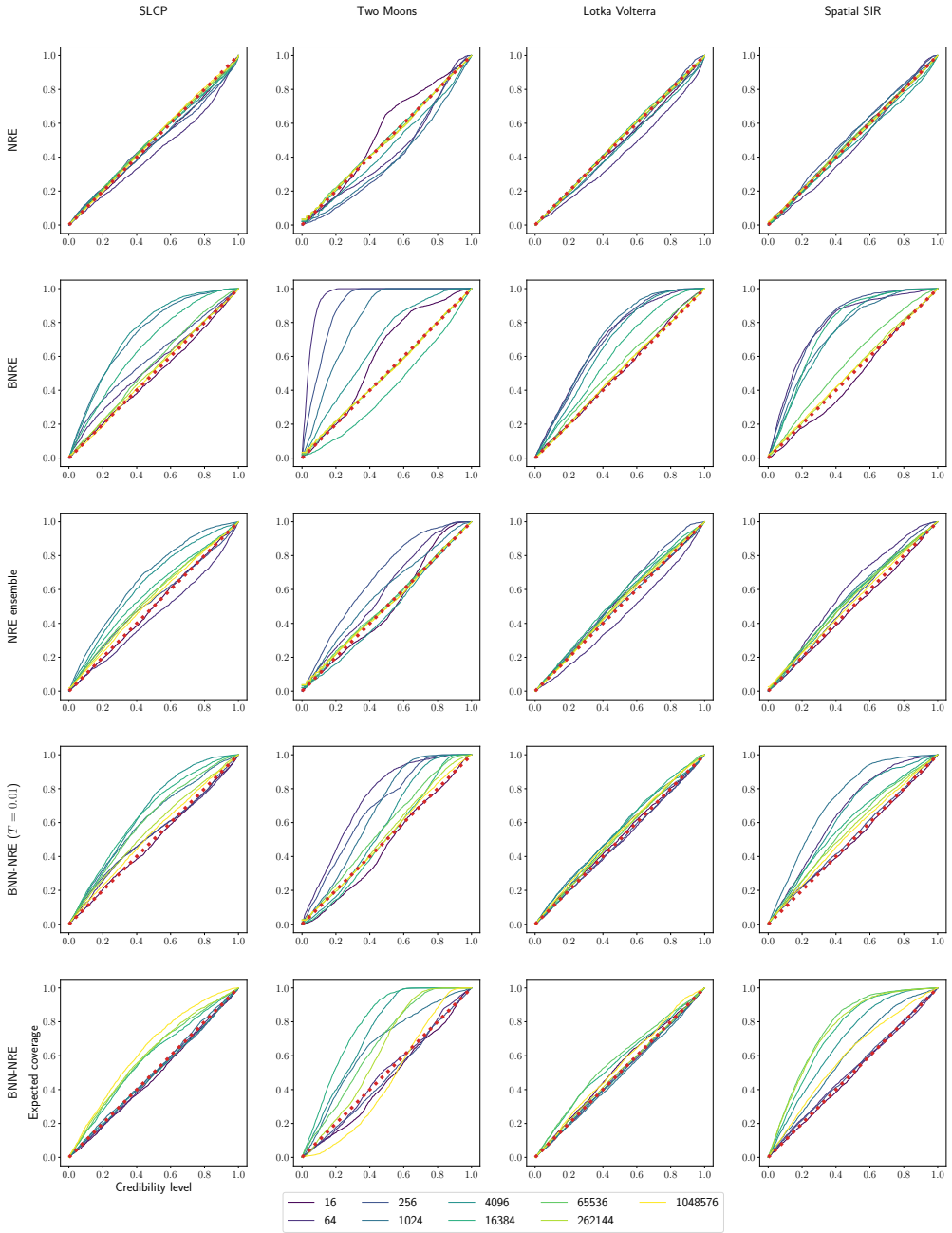

Figure 9: Coverage of different NRE simulation-based inference methods. A calibrated posterior approximation exhibits a coverage AUC of 0. A coverage curve above the diagonal indicates conservativeness and a curve below the diagonal indicates overconfidence. 3 runs are performed, and the median is reported.

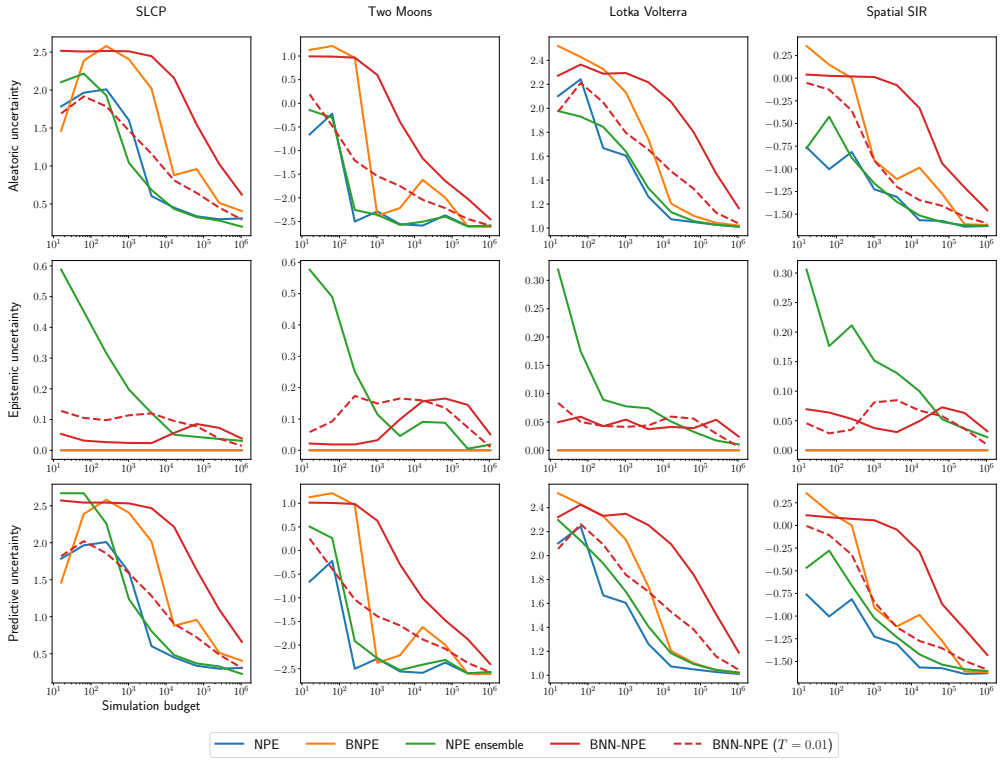

Figure 10: Quantification of the different forms of uncertainties captured by the different NPE-based methods. 3 runs are performed, and the median is reported.

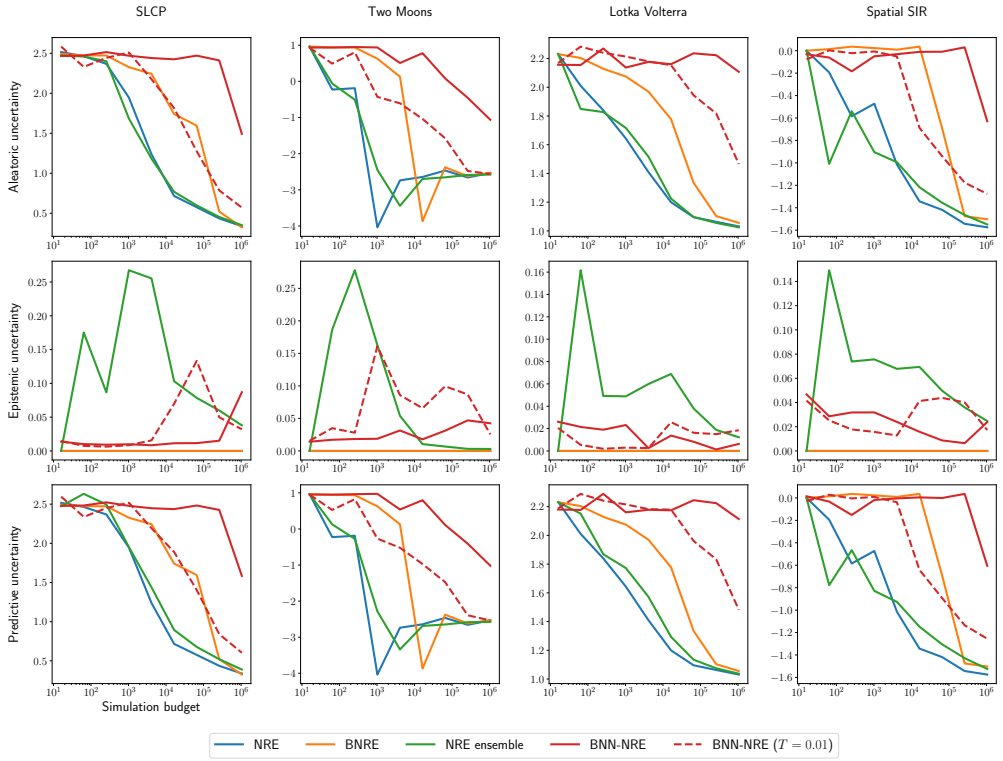

Figure 11: Quantification of the different forms of uncertainties captured by the different NRE-based methods. 3 runs are performed, and the median is reported.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: All the claims are based on either theoretical developments or empirical results.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Limitations are discussed in the experiments.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: Proposition 1 is proved and all the assumptions are clearly stated.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The code is made available. The link is in the introduction. All the hyperparameters are described in Appendix C.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The link to the code is available in the introduction.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: The code is made available, and all the hyperparameters are described in Appendix C.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Due to computational constraints, only 3 runs were made. This is not sufficient to report meaningful error bars. The median over 3 runs is always reported.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Those pieces of information are disclosed in Appendix C

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The paper respects the code of ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: There is no negative societal impact. Our method provides a way to do reliable simulation-based inference. We do not foresee any negative impact in improving the reliability of simulation-based inference.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: our method does not need safeguards.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The source of N-body simulation data is mentioned.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: Our contribution is about methodological development. No pre-trained models are released.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.