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

Master thesis

Completed in order to obtain the degree of Master of Science in Data Science Engineering

Conservative Simulation-Based Inference with Bayesian Deep Learning

Submitted by

Maxence de la Brassinne Bonardeaux

Academic Year: 2023-2024

University of Liège

Faculty of Applied Sciences

University supervisor: Pr. Gilles Louppe

Acknowledgements

First of all, I am deeply grateful to my supervisor, Professor Gilles Louppe. As my promoter, he consistently provided original and interesting ideas that advanced my research. His support and guidance were helpful, especially during challenging times.

I would like to extend my sincere thanks to Arnaud Delaunoy. Collaborating with him was a pleasure, and I greatly appreciated his valuable advice on both my research and the writing of this thesis.

I am also thankful to Jeanne, Gérôme, and Ophélie for their insightful comments and suggestions on the writing of my thesis. Their feedback significantly improved the quality of my work.

Lastly, I would like to thank my family for their support throughout this thesis.

Abstract

Simulation-Based Inference (SBI) involves estimating parameters $\boldsymbol{\theta}$ of a simulator that are compatible with the observations $\mathbf{x}$ without evaluating the likelihood of the data. Currently, the best solutions for SBI are neural SBI methods, which are trained using datasets built with simulations. However, simulations can be computationally expensive in fields like meteorology or cosmology. Consequently, SBI methods can operate in a data-poor regime in these fields. When only a limited number of simulations are available, traditional SBI methods tend to be overconfident due to neural methods overfitting the data. This overfitting leads to computational uncertainty, as many neural networks may fit the training data equally well but perform differently on the test data.

This thesis introduces a method using Bayesian Deep Learning (BDL) to account for computational uncertainty in SBI. We design a family of Bayesian Neural Network (BNN) priors that yield conservative results with as few as 10 samples, setting it apart from all other SBI methods. We demonstrate that the use of BDL in SBI produces informative and conservative posterior distribution estimates with only a few hundred simulations on a cosmological application. This advancement allows for drawing reliable scientific conclusions using our method, even when the number of available simulations is limited.

Table of Contents

Acronyms

ABC	Approximate Bayesian Computation.
BDL	Bayesian Deep Learning.
BMA	Bayesian Model Average.
BNN	Bayesian Neural Network.
BNN-NPE	Bayesian Neural Posterior Estimation.
BNN-NPE$_{\mathrm{T}}$	Bayesian Neural Posterior Estimation with Temperature.
BNN-NRE	Bayesian Neural Ratio Estimation.
BNN-NRE$_{\mathrm{T}}$	Bayesian Neural Ratio Estimation with Temperature.
BNNs	Bayesian Neural Networks.
BNPE	Balanced Neural Posterior Estimation.
BNRE	Balanced Neural Ratio Estimation.
CPE	Cold Posterior Effect.
EAU	Expected Aleatoric Uncertainty.
ECP	Expected Coverage Probability.
EEU	Expectec Epistemic Uncertainty.
ELBO	Evidence Lower Bound Objective.
ELP	Expected Log Posterior.
ETU	Expected Total Uncertainty.
HMC	Hamiltonian Monte Carlo.

KL	Kullback-Leibler.
MCMC	Markov Chain Monte Carlo.
NPE	Neural Posterior Estimation.
NRE	Neural Ratio Estimation.
SBI	Simulation-Based Inference.
SGHMC	Stochastic Gradient Hamiltonian Monte Carlo.
VI	Variational Inference.

1. Introduction

In many scientific domains, the ability to perform statistical inference on simulators is crucial. It can help prove or disprove scientific theories for instance. Inference involves the estimation of parameters, denoted as $\boldsymbol{\theta}$, that are compatible with the observed outputs $\mathbf{x}$ of a simulator. In other words, inference attempts to reverse-engineer the process of the simulator.

To estimate these parameters $\boldsymbol{\theta}$, traditional inference methods require computing the likelihood $p(\mathbf{x}|\boldsymbol{\theta})$, a mathematical function of how likely it is that the observed data $\mathbf{x}$ was generated if $\boldsymbol{\theta}$ are the parameters. However, in Simulation-Based Inference (SBI), evaluating the likelihood is not possible due to the complexity of the simulators used. In many applications, simulators describe an iterative process and evaluating the likelihood would require integrating over thousands of dimensions. Instead, SBI works directly with simulations. An application where SBI could be used is the expansion of the Universe (Villaescusa-Navarro et al., 2020). Observations $\mathbf{x}$ represent the evolution of particle matter in the Universe. The challenge lies in inferring key parameters of the Universe $\boldsymbol{\theta}$ such as the average density matter and how it is distributed in the Universe. Solving this problem would help explain the structure of the Universe and better understand dark matter and dark energy. The principles of SBI are detailed in Chapter 2 and are visually summarized in Figure 1.1.

To solve this likelihood-free inference problem, several SBI methods exist (Cranmer et al., 2020). These methods estimate the uncertainty in parameters $\boldsymbol{\theta}$ through the approximate posterior distribution $\hat{p}(\boldsymbol{\theta}|\mathbf{x})$. Uncertainty may arise from measurement systems, underlying physical processes, external factors, etc. However, in some cases, the approximate posterior $\hat{p}(\boldsymbol{\theta}|\mathbf{x})$ may underrepresent the true uncertainty around the parameters $\boldsymbol{\theta}$ (Figure 1.2) leading to overconfident results. This phenomenon is more pronounced in areas where simulations are expensive and limited, such as meteorology or cosmology. This

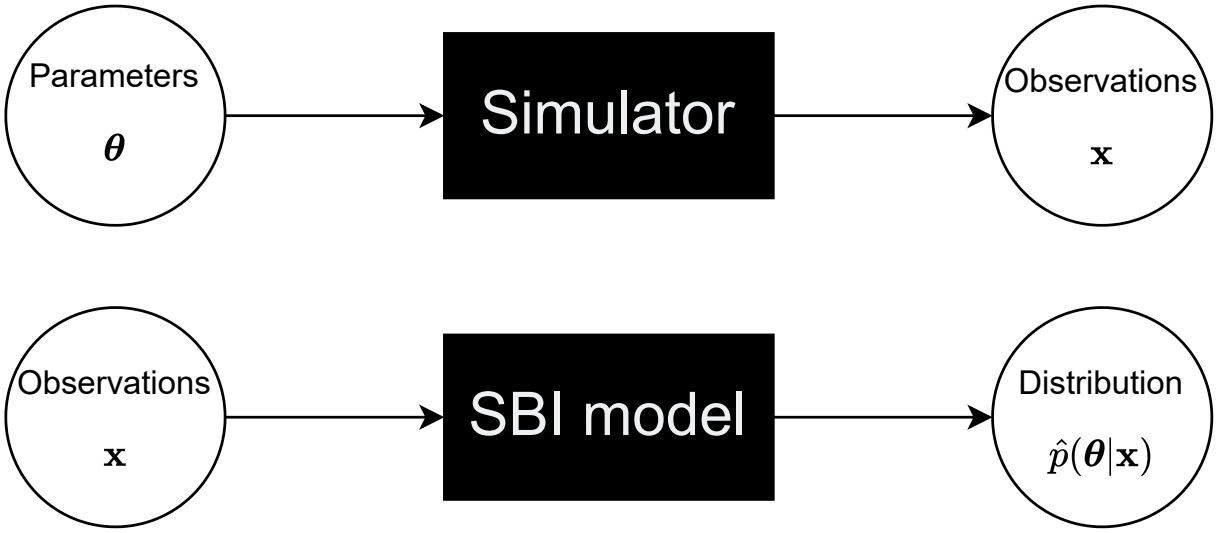


Figure 1.1: The SBI challenge: Inference of parameters $\boldsymbol{\theta}$ that are compatible with the observations $\mathbf{x}$. We rely solely on pairs $(\boldsymbol{\theta}, \mathbf{x})$ generated by the simulator or gathered from data collection.

behaviour occurs because neural networks are overfitting the limited data. Indeed, many neural networks can fit the training data equally well but may have very different predictions on the test data, leading to uncertainty as to which network best models the true posterior $p(\boldsymbol{\theta}|\mathbf{x})$. Therefore, these SBI methods can be unreliable for drawing scientific conclusions.

In response, several SBI methods have been developed to enhance conservativeness (Delaunoy et al., 2022, 2023; Falkiewicz et al., 2024; Masserano et al., 2023; Patel et al., 2023; Schmitt et al., 2023). Nonetheless, in cases where generating simulations is computationally expensive and there are only a limited number of simulations available, these methods may not provide informative distributions or may not be conservative. The specifics of these conservative methods are further discussed in Chapter 3.

In this work, we address the overfitting problem by considering all possible models fitting the training data to construct the posterior approximation $\hat{p}(\boldsymbol{\theta}|\mathbf{x})$. This is achieved using Bayesian Deep Learning (BDL) applied to traditional neural SBI methods. BDL aggregates all compatible approximations to construct a more conservative posterior approximation (Figure 1.3b). The use of BDL in SBI is detailed in Chapter 4. The results of our new method are analysed and compared to traditional methods in Chapter 5. We

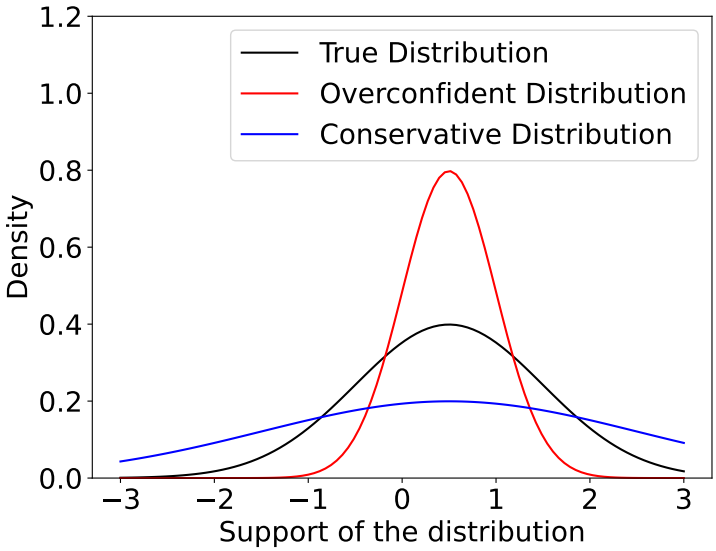
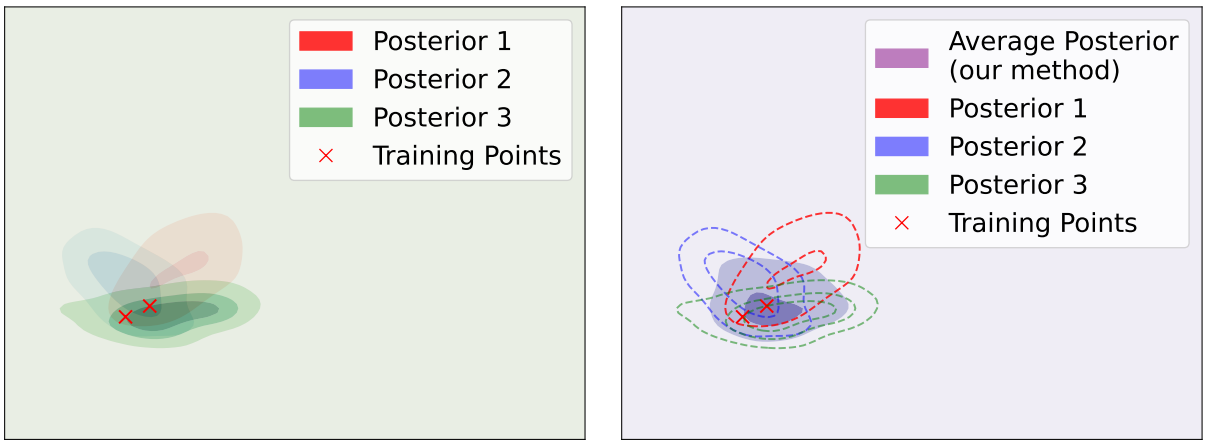


Figure 1.2: Examples of overconfident and conservative distributions. Overconfidence corresponds to distributions that are too narrow compared to the true distribution. Conservativeness corresponds to distributions that are too wide compared to the true distribution.

demonstrate the interest of our method on a practical application modelling the expansion of the Universe in Chapter 6.

Contributions: This work makes two main contributions. Firstly, it introduces a conservative method using BDL to account for computational uncertainty in SBI. The effect on the posterior of our method is empirically analysed and compared to other existing methods. The conservativeness and the information contained in the distributions are evaluated. The comparison is both quantitative and qualitative. Secondly, we design a family of Bayesian Neural Network (BNN) priors tailored for SBI which yields calibrated posteriors in the data-poor regime. These priors do not need any training data to provide calibrated posteriors.



(a) Traditional methods lead to many distributions fitting equally well the training points.

(b) Our method is the average of all possible distributions fitting the training points.

Figure 1.3: Comparison between traditional methods and our new method. Depending on randomness during training, traditional methods might lead to different posteriors as in the left figure. In scenarios with limited data, many approximations can explain the observations, yet these methods fail to account for this computational uncertainty. Our method improves conservativeness by considering all plausible approximations to construct the posterior distribution like in the right figure.

2. Simulation-Based Inference

2.1. Bayesian Inference

Inference enables scientists to draw scientific conclusions by estimating these parameters, which helps to prove or disprove scientific theories for instance. The main challenge of this thesis is the inference of parameters $\boldsymbol{\theta}$ from observations $\mathbf{x}$ (Figure 1.1) by determining the posterior distribution $p(\boldsymbol{\theta}|\mathbf{x})$.

In a typical scenario, the parameters $\boldsymbol{\theta}$ are used to generate observations $\mathbf{x}$ of a simulator through a forward process described by the distribution $p(\mathbf{x}|\boldsymbol{\theta})$. This distribution is defined by the simulator mechanism. The posterior distribution $p(\boldsymbol{\theta}|\mathbf{x})$ is defined using Bayes' rule

$$p(\boldsymbol{\theta}|\mathbf{x}) = \frac{p(\boldsymbol{\theta})p(\mathbf{x}|\boldsymbol{\theta})}{p(\mathbf{x})} \propto p(\boldsymbol{\theta})p(\mathbf{x}|\boldsymbol{\theta}), \tag{2.1}$$

where $p(\boldsymbol{\theta})$ is the prior distribution given by an expert's knowledge, and $p(\mathbf{x})$ acts as a normalizing constant. In scenarios where the likelihood $p(\mathbf{x}|\boldsymbol{\theta})$ can be computed, Markov Chain Monte Carlo (MCMC) samplers (Neal et al., 2011; Karras et al., 2022) are commonly used to estimate the posterior by sampling from it.

In many practical applications, the likelihood is computationally prohibitive to derive (Villaescusa-Navarro et al., 2020; Clemencic et al., 2011) but sampling can be done using the simulator. Such conditions render traditional MCMC samplers unusable, necessitating alternative strategies that do not rely on direct likelihood evaluations. Simulation-Based Inference (SBI) uses only likelihood samples to compute an approximate posterior distribution $\hat{p}(\boldsymbol{\theta}|\mathbf{x})$.

2.2. Simulation-Based Inference Methods

2.2.1. Overview

In SBI, several methods have been developed to estimate $p(\boldsymbol{\theta}|\mathbf{x})$ (Cranmer et al., 2020). Depending on what they approximate from Equation 2.1, they can be categorised into three groups (Figure 2.1).

1. **Likelihood Approximation** $p(\mathbf{x}|\boldsymbol{\theta})$: These methods approximate the likelihood function $p(\mathbf{x}|\boldsymbol{\theta})$ of Equation 2.1. Two popular techniques include Approximate Bayesian Computation (ABC) (Rubin, 1984; Beaumont et al., 2002) and density estimation methods (Papamakarios et al., 2019a; Durkan et al., 2018; Lueckmann et al., 2019; Alsing et al., 2019). The first one approximates the posterior distribution by using simulations to generate data and accepting parameters based on their similarity with the observed data. The second one approximates the likelihood directly. Then, MCMC can be used to sample from the approximate posterior as it only needs the approximate likelihood $\hat{p}(\mathbf{x}|\boldsymbol{\theta})$ and the prior $p(\boldsymbol{\theta})$. However, these techniques often perform poorly with high-dimensional observations $\mathbf{x}$.
2. **Likelihood-to-Evidence Ratio Approximation** $r(\mathbf{x}|\boldsymbol{\theta}) = \frac{p(\mathbf{x}|\boldsymbol{\theta})}{p(\mathbf{x})}$: This method approximates the ratio $r(\mathbf{x}|\boldsymbol{\theta})$, enabling posterior sampling with MCMC. Indeed, the approximate posterior can be computed using $\hat{p}(\boldsymbol{\theta}|\mathbf{x}) = p(\boldsymbol{\theta})\hat{r}(\mathbf{x}|\boldsymbol{\theta})$. The main method here is Neural Ratio Estimation (NRE) (Hermans et al., 2020a), detailed further in Section 2.2.3.
3. **Posterior Approximation** $p(\boldsymbol{\theta}|\mathbf{x})$: This category focuses on direct posterior approximation. The leading technique, Neural Posterior Estimation (NPE) and its variations, maintains an approximate posterior distribution $\hat{p}(\boldsymbol{\theta}|\mathbf{x})$. This method allows direct sampling from the approximate posterior (Papamakarios and Murray, 2016) and is described in Section 2.2.2.

Given the challenges associated with high-dimensional observational space, likelihood approximations are less frequently used. Currently, NRE and NPE are favoured because they achieve better performance. The following sections detail these two methods.

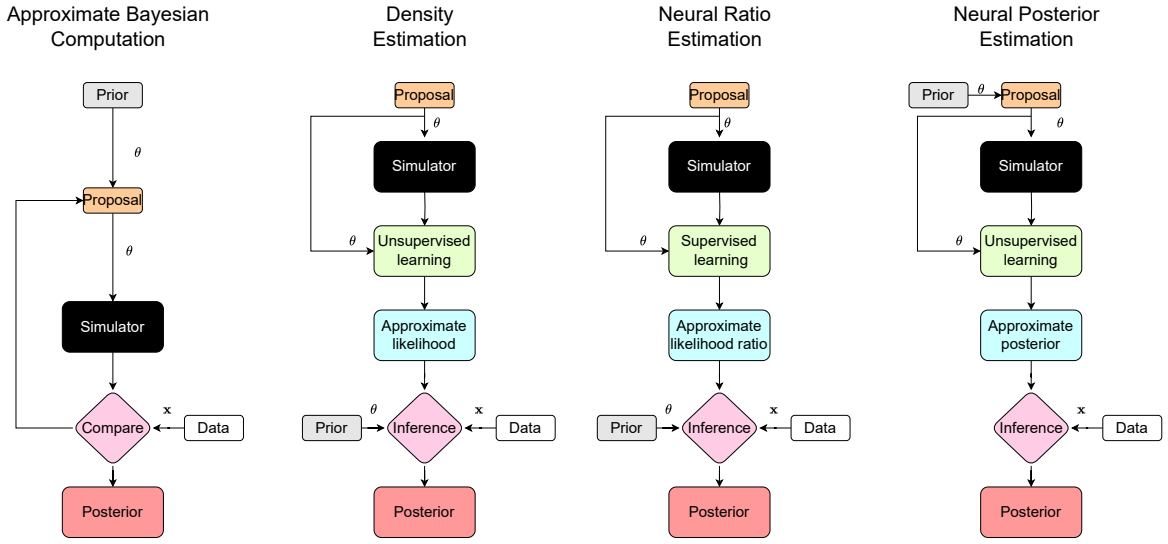


Figure 2.1: Overview of current SBI methods. The first two methods (ABC and density estimation) approximates the posterior by approximating the likelihood function $p(\mathbf{x}|\boldsymbol{\theta})$. NRE tries to approximate the likelihood-to-evidence ratio $r(\mathbf{x}|\boldsymbol{\theta}) = \frac{p(\mathbf{x}|\boldsymbol{\theta})}{p(\mathbf{x})}$ to compute the posterior distribution. Finally, NPE aims at approximating the posterior distribution directly. NRE and NPE are the most popular methods. This figure is based on (Cranmer et al., 2020).

2.2.2. Neural Posterior Estimation

NPE (Last graph in Figure 2.1) directly approximates the posterior distribution $p(\boldsymbol{\theta}|x)$ with a conditional distribution $q_\phi(\boldsymbol{\theta}|x)$ using a neural network (Papamakarios and Murray, 2016). This method is amortised, which means that after a training phase, the approximate posteriors can be efficiently evaluated for arbitrary observations $\mathbf{x}$. This method is trained to minimise the Kullback-Leibler (KL) divergence between the true posterior distribution and the conditional distribution (Greenberg et al., 2019)

$$\begin{aligned}
\phi^* &= \arg\min_\phi \mathbb{E}_\mathbf{x}[KL(p(\boldsymbol{\theta}|\mathbf{x})||q_\phi(\boldsymbol{\theta}|\mathbf{x}))] \\
&= \arg\min_\phi \mathbb{E}_\mathbf{x}\left[\mathbb{E}_{p(\boldsymbol{\theta}|\mathbf{x})}\left[\log \frac{p(\boldsymbol{\theta}|\mathbf{x})}{q_\phi(\boldsymbol{\theta}|\mathbf{x})}\right]\right] \\
&= \arg\max_\phi \mathbb{E}_{p(\boldsymbol{\theta},\mathbf{x})}[\log q_\phi(\boldsymbol{\theta}|\mathbf{x})].
\end{aligned}$$

The training leverages data generated by the simulator, optimising the conditional distribution to match the true posterior. Normalizing flows are commonly used for their efficiency in modelling complex distributions with invertible transformations (Dinh et al., 2014; Rezende and Mohamed, 2015; Papamakarios et al., 2017; Durkan et al., 2019).

2.2.3. Neural Ratio Estimation

NRE (third graph in Figure 2.1) aims at estimating the likelihood-to-evidence ratio $r(\mathbf{x}|\boldsymbol{\theta}) = \frac{p(\mathbf{x}|\boldsymbol{\theta})}{p(\mathbf{x})}$ (Hermans et al., 2020a). Approximating the ratio $r(\mathbf{x}|\boldsymbol{\theta})$ is achieved by defining a binary classifier $d(\boldsymbol{\theta}, \mathbf{x}) : \boldsymbol{\theta} \times \mathbf{X} \rightarrow [0, 1]$. It is trained to differentiate samples of the joint distribution $p(\boldsymbol{\theta}, \mathbf{x})$ from the product of the marginal distributions $p(\boldsymbol{\theta})p(\mathbf{x})$ (Cranmer et al., 2015). For the binary-cross entropy loss, the Bayes optimal classifier is (Hermans et al., 2020b)

$$\begin{aligned} d(\boldsymbol{\theta}, \mathbf{x}) &= \frac{p(\boldsymbol{\theta}, \mathbf{x})}{p(\boldsymbol{\theta}, \mathbf{x}) + p(\boldsymbol{\theta})p(\mathbf{x})} \\ &= \sigma\left(\log \frac{p(\boldsymbol{\theta}, \mathbf{x})}{p(\boldsymbol{\theta})p(\mathbf{x})}\right) \\ &= \sigma\left(\log \frac{p(\mathbf{x}|\boldsymbol{\theta})}{p(\mathbf{x})}\right) \\ &= \sigma\left(\log r(\mathbf{x}|\boldsymbol{\theta})\right), \end{aligned}$$

where $\sigma(.)$ is the sigmoid function. The output of the classifier then can be used with the inverse of the sigmoid to approximate the log likelihood ratio. Finally, using the approximate ratio, the approximate posterior can be computed using $\hat{p}(\boldsymbol{\theta}|\mathbf{x}) = p(\boldsymbol{\theta})\hat{r}(\mathbf{x}|\boldsymbol{\theta})$. This method is also amortised.

While NRE offers an alternative method to NPE, it allows only a posterior density evaluation. Other steps, such as MCMC sampling, are required to obtain posterior samples, which can be computationally intensive compared to NPE.

2.3. Summary

SBI addresses the challenge of inferring parameters $\boldsymbol{\theta}$ from simulated data $\mathbf{x}$ without direct access to the likelihood function $p(\mathbf{x}|\boldsymbol{\theta})$. The prevailing methods, NPE and NRE, use neural networks to approximate the posterior distribution $p(\boldsymbol{\theta}|\mathbf{x})$. The forthcoming chapters will outline limitations of these methods and how to overcome those limitations.

3. Conservative Simulation-Based Inference

3.1. Coverage for Conservativeness Diagnostic

In Chapter 2, Neural Posterior Estimation (NPE) and Neural Ratio Estimation (NRE) were introduced as solutions for Simulation-Based Inference (SBI). While these methods perform well with a large number of simulations, they face limitations when the number of simulations is limited, typical in fields like cosmology or meteorology. One of these limitations is overconfidence in the results (Figure 1.2). Overconfidence is an inaccurate representation of uncertainty, which is often under-represented (Hermans et al., 2021).

Overconfidence comes from neural networks, used in SBI methods, overfitting the data. To prevent overfitting, various strategies have been employed (Ying, 2019). Although these techniques may be effective in some cases, they are not specifically designed for conservativeness. For example, L2-regularisation, a common technique against overfitting, biases model weights towards zero. However, in NPE, this leads to posteriors with probability density outside the prior support. This undermines the reliability of the method to draw scientific conclusions. Consequently, there is a critical need for overfitting prevention methods tailored specifically to SBI to ensure posteriors are conservative. Such conservative methods construct broad confidence intervals to encompass the true parameters with high probability.

To evaluate conservativeness and detect overconfidence within SBI methods, the concept of Expected Coverage Probability (ECP) is used (Hermans et al., 2021).

The ECP for the $1 - \alpha$ highest posterior density regions derived from the estimated

posterior, $\hat{p}(\boldsymbol{\theta}|\mathbf{x})$, is defined as

$$\mathbb{E}_{p(\boldsymbol{\theta},\mathbf{x})}[\mathbb{I}[\boldsymbol{\theta} \in \Theta_{\hat{p}(\boldsymbol{\theta}|\mathbf{x})}(1-\alpha)]],$$

where $\Theta_{\hat{p}(\boldsymbol{\theta}|\mathbf{x})}(1-\alpha)$ denotes the $1-\alpha$ highest posterior density region of $\hat{p}(\boldsymbol{\theta}|\mathbf{x})$. This quantity can be numerically estimated (Appendix A.1.1). The true posterior exhibits an ECP of $1-\alpha$ for all $1-\alpha$ levels. Depending on the performance of an approximate posterior, three scenarios may occur:

1. The posterior is **conservative** (or underconfident) if its ECP exceeds $1-\alpha$ for all $1-\alpha$ levels.
2. The posterior is **well-calibrated** if its ECP equals $1-\alpha$ for each level.
3. The posterior is **overconfident** if its ECP falls below $1-\alpha$ for any $1-\alpha$ level.

The ECP and the three scenarios can be graphically represented (Figure 3.1). Their effects are visually illustrated in Figure 1.2.

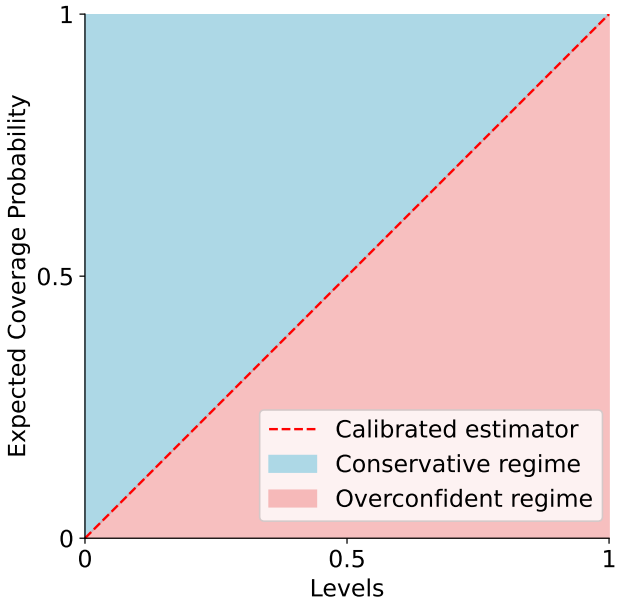
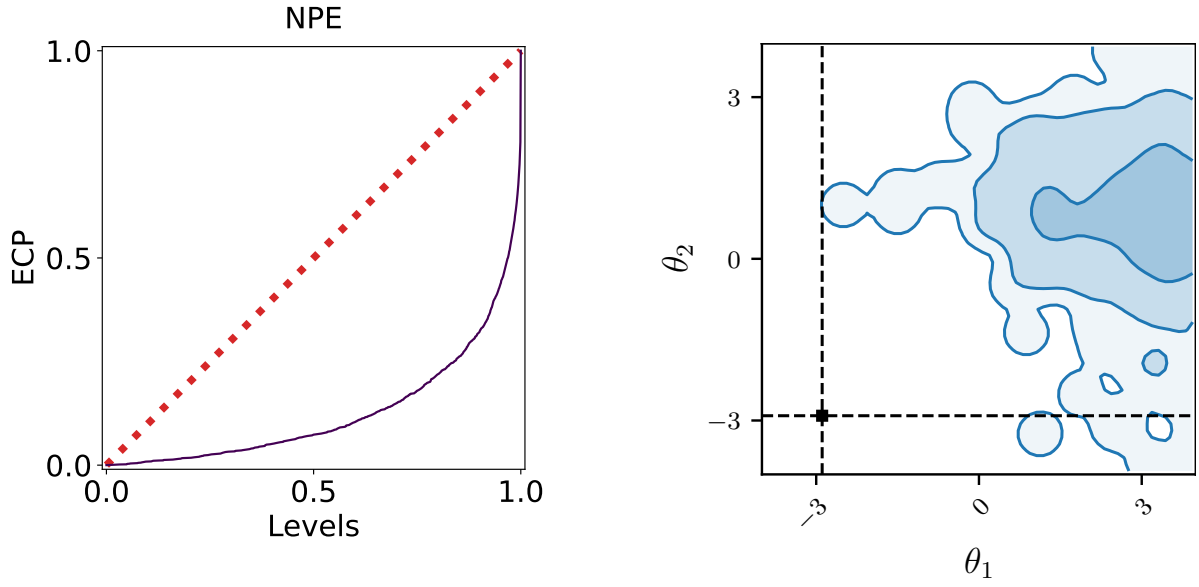


Figure 3.1: Graphical representation of the ECP. If the ECP is above the dotted red line, the posterior is conservative. If it is below, the posterior is overconfident.

One limitation of this diagnostic is that while approximate posteriors may be conservative in expectation over an entire test set, there is no guarantee of well-calibration for individual

observations $\mathbf{x}_0$. Although there exist some techniques to evaluate local coverage (Zhao et al., 2021; Linhart et al., 2024), they are outside the scope of this thesis. Nevertheless, we will show how methods perform for different observations in Section 5.3.2.

This ECP diagnostic allows us to show when a method is overconfident. Figure 3.2 demonstrates instances where NPE exhibits overconfidence on the SLCP benchmark (Appendix A.3).



(a) ECP of NPE method.

(b) Posterior distribution of NPE on a sample from the test set.

Figure 3.2: Illustrations of overconfidence in NPE on the SLCP benchmark. NPE can exhibit overconfidence, as indicated by its ECP falling below the diagonal. The poor estimation is evident as the approximate posterior has nearly no density on the true parameters.

3.2. Conservative Methods Tailored for Simulation-Based Inference

Conservative SBI methods can be categorised into three main groups:

1. **Statistical Test-Based Confidence Intervals Method:** Methods using test statistics provide theoretical guarantees for constructing confidence intervals (Masser-

ano et al., 2023; Patel et al., 2023). In these methods, parameters are included in the confidence interval if the statistical test does not reject them. However, these methods require a large number of simulations to provide informative results, rendering them less practical for cases with limited data available.

2. **Regularisation Techniques Method:** Specific regularisation methods developed for SBI, like balanced methods and differentiable ECP, offer ways to manage conservativeness (Delaunoy et al., 2022, 2023; Falkiewicz et al., 2024). While these methods are promising, they can still lead to overconfidence as they act only as a soft constraint.
3. **Ensemble Methods:** Using ensembles introduces additional uncertainty on the approximation, leading to more conservative outcomes. Ensembles approximate Bayesian methods (Hoffmann and Elster, 2021) and provide informative posteriors but they are not immune to the risk of overconfidence.

Figure 3.3 provides a graphical overview of these methods. Further discussion on Bayesian methods is reserved for Chapter 4.

3.2.1. WALDO Confidence Intervals

This method is a post-processing technique which contrasts with conventional methods by deriving conservative confidence intervals through a modified Wald test statistic (Wald, 1943), referred to as WALDO (Masserano et al., 2023). A test statistic tries to reject parameters to construct a confidence interval. Although the WALDO test statistic provides asymptotic guarantees for posterior distributions, its effectiveness diminishes with limited data. Thus, its utility is constrained in contexts where traditional methods already perform very well.

3.2.2. Balanced Neural Methods

Balanced Neural Ratio Estimation (BNRE) (Delaunoy et al., 2022) and Balanced Neural Posterior Estimation (BNPE) (Delaunoy et al., 2023) correspond to the third graph in Figure 3.3. These methods add a regularisation term in the loss function to help the method to be balanced. The balancing condition, as explained in (Delaunoy et al., 2022),

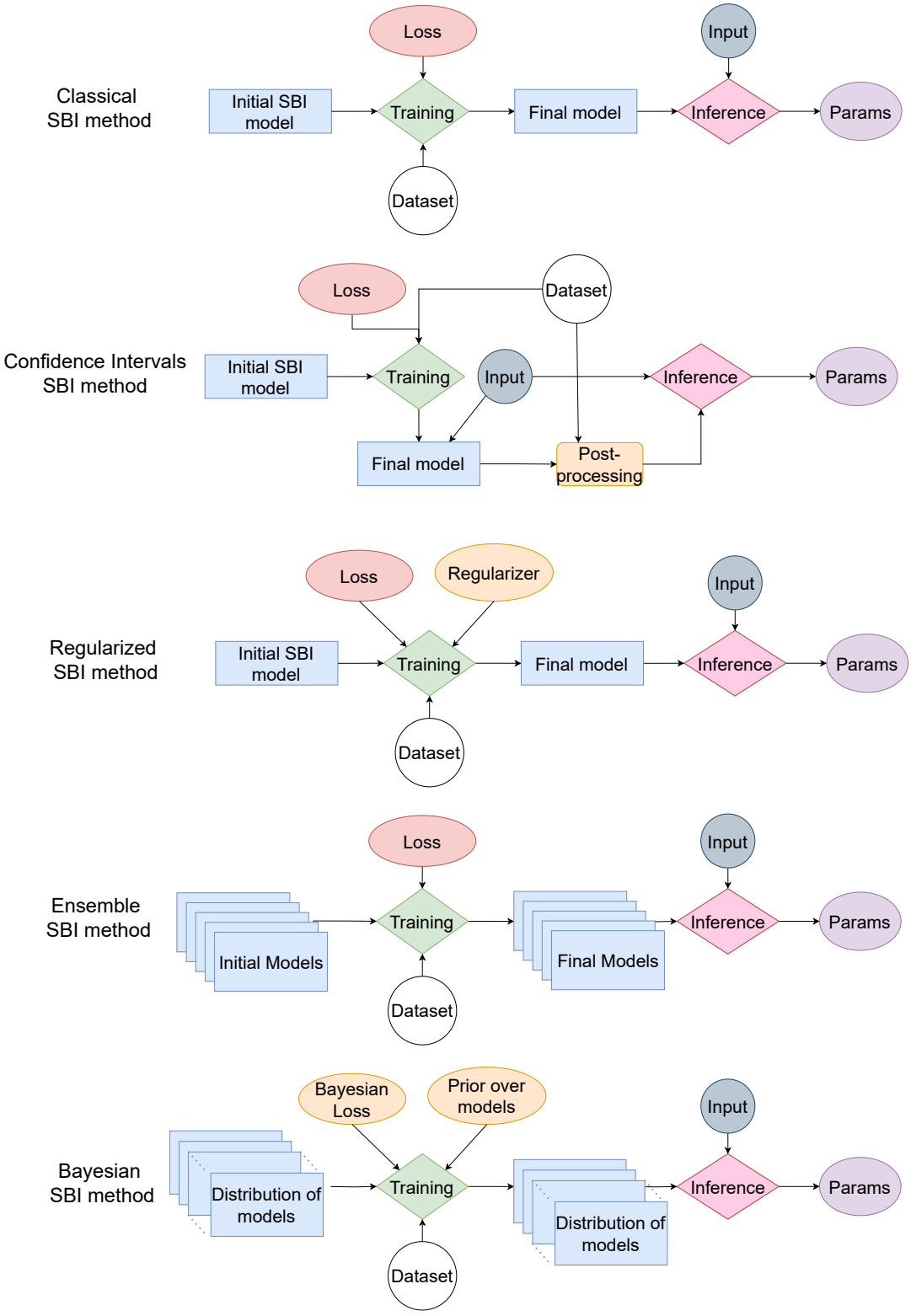


Figure 3.3: Review of SBI methods to improve conservativeness.

effectively creates larger confidence intervals than those predicted by the true posterior, thus guiding the approximate posterior $\hat{p}(\boldsymbol{\theta}|\mathbf{x})$ towards conservativeness. The Bayes optimal model remains unchanged for BNRE and BNPE.

However, substantial challenges remain in scenarios marked by limited data. Although BNRE remains conservative almost all the time, its posteriors can be uninformative without a large number of simulations. In those regimes, BNRE models the prior distribution $p(\boldsymbol{\theta})$ which is balanced and calibrated. Therefore, it is hard for the method to learn another calibrated and balanced approximate posterior distribution. For BNPE, it is the opposite. It is hard for the method to learn to be balanced with limited data and the method can still produce overconfident posteriors.

3.2.3. Regularisation with Differentiable Coverage

This method introduces a regularisation technique that incorporates a penalising term for posterior miscalibration (Falkiewicz et al., 2024). By using the ECP formula in the loss function as a soft constraint, this method tries to provide posteriors that are informative and well-calibrated. Unlike previous methods, this method specifically targets conservativeness. Despite the absence of theoretical guarantees, experimental results have demonstrated its effectiveness. However, like other methods discussed, its efficiency depends on the availability of sufficient data. In some cases, the results can still be overconfident (Falkiewicz et al., 2024).

3.2.4. Ensemble Methods

Increasing conservativeness can be achieved by using ensembles of neural networks (Hermans et al., 2021). This method corresponds to the fourth graph in Figure 3.3. Ensembles share similarities with Bayesian methods (Lakshminarayanan et al., 2017). Specifically, ensembles combine predictions from multiple approximate posteriors, increasing the overall uncertainty on the approximation with some computational uncertainty. This increased uncertainty helps for achieving a more conservative approximate posterior, as it translates to wider confidence intervals.

Despite these advantages, it is important to recognize that deep ensembles, like other

methods, are not infallible in their pursuit of conservativeness (Delaunoy et al., 2023).

3.3. Summary

Recent studies have demonstrated that both NPE and NRE can produce overconfident posterior estimations. In response, conservative methods have been developed to mitigate these issues. However, these methods present two primary drawbacks: either they maintain conservativeness consistently but require extensive data to learn informative posterior distributions, or they learn more rapidly but struggle to achieve consistent conservativeness. This dilemma highlights the need for further research to create a method able to learn efficiently and be conservative when needed.

4. Incorporating Computational Uncertainty in Conservative Simulation-Based Inference

4.1. Using Computational Uncertainty to Capture the Full Uncertainty of Approximate Posteriors

The true posterior distribution has some uncertainty about the data itself which is due to the stochastic simulator. Indeed, due to the stochasticity of the simulator, several sets of parameters are compatible with the observation.

The approximate distribution $\hat{p}(\boldsymbol{\theta}|\mathbf{x})$ introduces another source of uncertainty, known as computational uncertainty. As the approximate distributions are estimated from finite datasets, there can be many different approximations $\hat{p}(\boldsymbol{\theta}|\mathbf{x})$ that fit the training data well (Figure 1.3). However, these approximations may perform very differently on test data. This leads to uncertainty about which approximation best models the true posterior $p(\boldsymbol{\theta}|\mathbf{x})$. This uncertainty can be reduced with more data.

Our method enhances conservativeness in data-poor regimes by using a wide range of plausible approximate posteriors when estimating the posterior over parameters $\hat{p}(\boldsymbol{\theta}|\mathbf{x})$ (Figure 1.3). This method is implemented using Bayesian Deep Learning (BDL) for Neural Posterior Estimation (NPE) and Neural Ratio Estimation (NRE). Ensembles, an approximation of BDL, have already been shown to increase conservativeness (Hermans et al., 2021). Using the full BDL framework would allow to control the computational uncertainty.

4.2. Background: Bayesian Deep Learning for Computational Uncertainty

BDL in Simulation-Based Inference (SBI) represents a shift from traditional SBI methods by taking into account computational uncertainty in its framework. This is achieved using Bayesian Neural Networks (BNNs) (Papamarkou et al., 2024).

In a Bayesian Neural Network (BNN), each weight is not just a single value but a random variable with its own distribution (Figure 4.1). BDL works with a range of possible posteriors rather than a single distribution. The posterior distribution over the weights $\hat{p}(\mathbf{w}|D)$ is used to compute the Bayesian Model Average (BMA) posterior

$$\hat{p}(\boldsymbol{\theta}|\mathbf{x}) = \int_{\mathbf{w}} \hat{p}(\boldsymbol{\theta}|\mathbf{x}, \mathbf{w})\hat{p}(\mathbf{w}|D)\, d\mathbf{w} \approx \frac{1}{N}\sum_{i=1}^{N} \hat{p}(\boldsymbol{\theta}|\mathbf{x}, \mathbf{w}_i). \tag{4.1}$$

Of course, the evaluation of the integral is impossible given the dimension of the weights. Therefore, a Monte-Carlo approximation is used where the weights $\mathbf{w}_i$ from Equation 4.1 are samples from the posterior distribution $\hat{p}(\mathbf{w}|D)$.

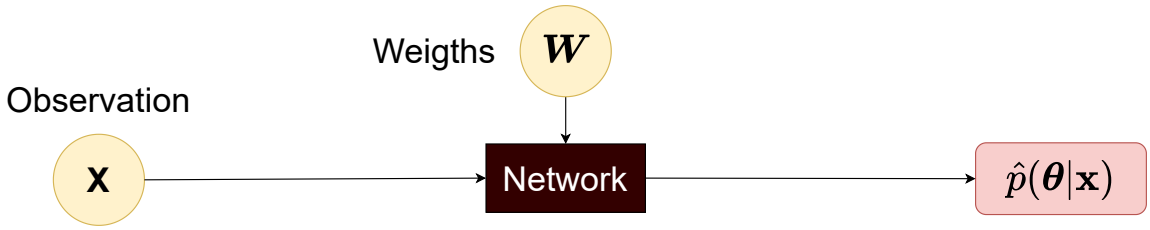
Applying Bayes' rule, we aim to infer the posterior distribution over the weights, $p(\mathbf{w}|D)$, from our dataset D defined as

$$p(\mathbf{w}|D) = \frac{p(D|\mathbf{w})p(\mathbf{w})}{p(D)}. \tag{4.2}$$

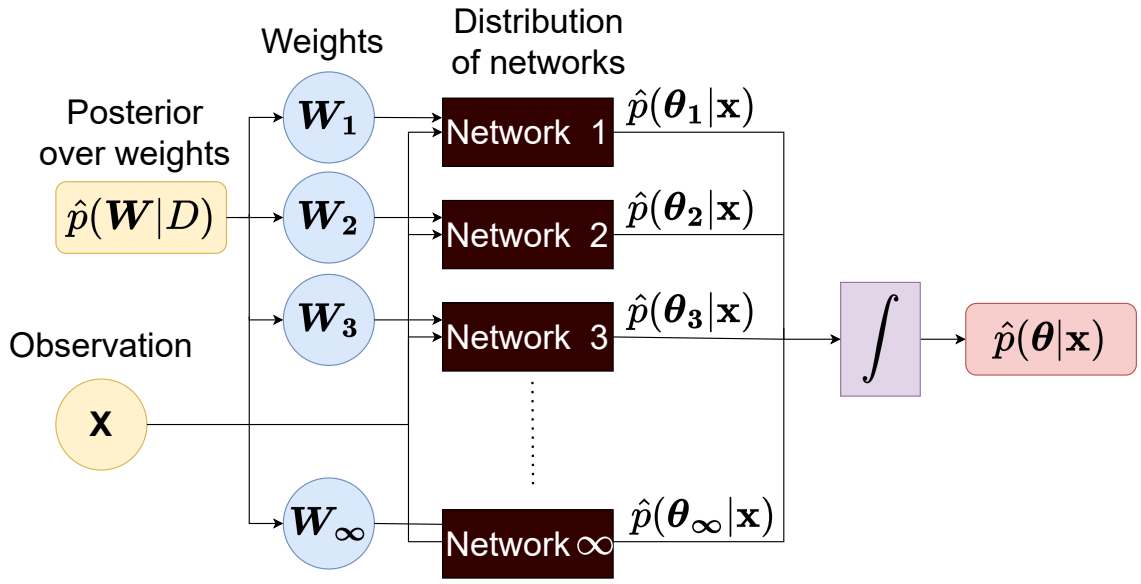
Here, $p(\mathbf{w})$ allows to incorporate prior knowledge into the neural network and acts as both a regularizer and as a mechanism to add computational uncertainty.

To compute $p(\mathbf{w}|D)$, we use approximation methods such as dropout (Gal and Ghahramani, 2016), Variational Inference (VI) (Blundell et al., 2015) or Hamiltonian Monte Carlo (HMC) method (Neal et al., 2011) among others.

It is important to distinguish between the posterior over the weights $p(\mathbf{w}|D)$ used by BNNs and the posterior over the simulator parameters $p(\boldsymbol{\theta}|\mathbf{x})$. Similarly, the prior over the weights $p(\mathbf{w})$ and the prior over the simulator parameters $p(\boldsymbol{\theta})$ must be recognized as distinct.



(a) Traditional neural network.



(b) Bayesian neural network.

Figure 4.1: Comparison of Classic Neural Networks with Bayesian Neural Networks. In a classic neural network, each weight has a single fixed value. In contrast, a Bayesian Neural Network maintains a distribution over each weight, allowing it to represent an infinite number of networks which take into account computational uncertainty.

4.2.1. Mean-Field Variational Inference Method

Mean-field VI is employed in BDL to approximate the posterior distribution over the weights $p(\mathbf{w}|D)$ with a variational family, $q_\phi(\mathbf{w})$. Typically, this family assumes independent normal distributions for the weights of the network and the parameters of the variational family are the mean and standard deviation of these weights (Graves, 2011; Feng et al., 2021). The family is trained by minimising the Kullback-Leibler (KL) divergence between the variational approximation and the true posterior over the weights.

One limitation of mean-field VI is its under-representation of epistemic uncertainty which might lead to smaller confidence intervals for the parameters $\boldsymbol{\theta}$. Mean-field VI assumes independent normal distributions for the BNN, which leads to a unimodal approximation over the weights. This simplification can be problematic in cases where the true posterior over the weights is multimodal, because the normal assumption might capture only one of the possible modes over the weights and underestimate the uncertainty of the distribution $p(\boldsymbol{\theta}|\mathbf{x})$ (Blei et al., 2017). A distinction must be made between the distribution over the weights $\mathbf{w}$ and the parameters of the simulator $\boldsymbol{\theta}$. The distribution over the weights can only be unimodal. However, the approximate distribution over the parameters $\hat{p}(\boldsymbol{\theta}|\mathbf{x})$ can be multimodal.

4.2.2. Stochastic Gradient Hamiltonian Monte Carlo Method

The Stochastic Gradient Hamiltonian Monte Carlo (SGHMC) method (Chen et al., 2014) presents an alternative for sampling from the posterior distribution $p(\mathbf{w}|D)$ in BDL. SGHMC can take into account the correlations among network weights. This advantage comes from the use of Hamiltonian dynamics used in the HMC method (Neal et al., 2011)[1]. It allows efficient exploration of complex and high-dimensional parameter spaces.

Despite its benefits, SGHMC introduces the challenge of hyperparameter tuning that plays a critical role in the effectiveness of the algorithm. Optimising these hyperparameters is often a complex and slow process, requiring careful experimentation and expertise. Nevertheless, with proper tuning, SGHMC can improve the inference in Bayesian models, making it a valuable tool for capturing the full complexity of the data and weight

[1]For the rest of the thesis, we will not differentiate HMC and SGHMC. The one used will always be SGHMC.

interactions.

4.2.3. Uncertainty Estimation in Bayesian Deep Learning

BDL allows to distinguish between aleatoric and epistemic uncertainty of the approximate posterior $\hat{p}(\boldsymbol{\theta}|\mathbf{x})$. In our case, the aleatoric uncertainty corresponds to the uncertainty about the parameters $\boldsymbol{\theta}$ measured by each individual distribution. The epistemic uncertainty corresponds to the computational uncertainty, the uncertainty about the approximation itself. The uncertainty types are quantified as follows

1. **Total Uncertainty (TU)**: quantified by the entropy of the BMA distribution

$$TU(\mathbf{x}|D) = H[\hat{p}(\boldsymbol{\theta}|\mathbf{x})] = H[\mathrm{E}_{\hat{p}(\mathbf{w}|D)}\hat{p}(\boldsymbol{\theta}|\mathbf{x}, \mathbf{w})]. \tag{4.3}$$

2. **Aleatoric Uncertainty (AU)**: quantified by the expected entropy across individual model distributions

$$AU(\mathbf{x}|D) = \mathrm{E}_{\hat{p}(\mathbf{w}|D)}H[\hat{p}(\boldsymbol{\theta}|\mathbf{x}, \mathbf{w})]. \tag{4.4}$$

3. **Epistemic Uncertainty (EU)**: quantified by the difference between total and aleatoric uncertainties

$$EU(\mathbf{x}|D) = TU(\mathbf{x}|D) - AU(\mathbf{x}|D). \tag{4.5}$$

In these expressions, D denotes the training dataset and H represents the Shannon entropy, defined as $H(p) = \mathrm{E}_p[\log_2(p)]$. There are other ways to estimate uncertainty. The motivations for using the Shannon entropy is detailed in Appendix A.1.3.

Figure 4.2 illustrates the different types of uncertainties for a distribution of two parameters. The total uncertainty corresponds to the uncertainty of the BMA distribution. The aleatoric uncertainty is the average uncertainty of each individual distribution. The epistemic uncertainty corresponds to the variation of uncertainty between each distribution.

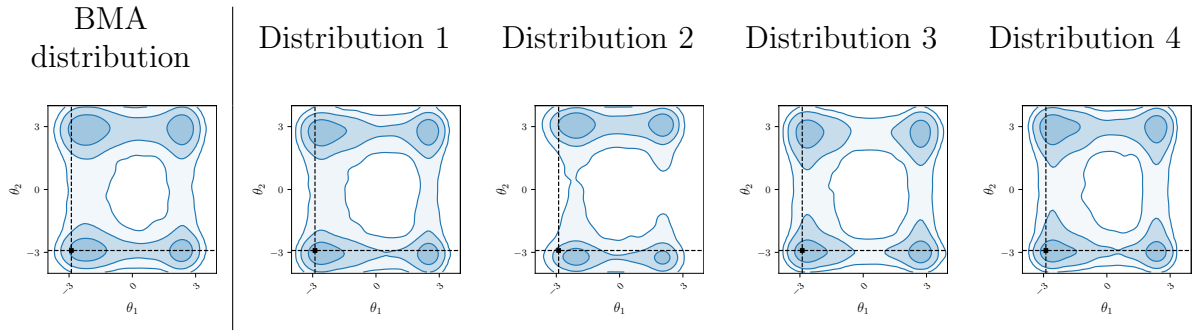


Figure 4.2: Visualisation of the effect of total, aleatoric and epistemic uncertainties in BNNs. Aleatoric uncertainty is the average uncertainty in each individual distribution. Epistemic uncertainty helps to analyse the difference between individual distributions. The total uncertainty is the uncertainty in the BMA distribution.

4.3. Using Bayesian Deep Learning for Conservative Simulation-Based Inference

4.3.1. Influence of Priors over Neural Networks on the Posterior

The choice of a good prior over weights $p(\mathbf{w})$ for BNN is crucial because it serves three purposes: the incorporation of prior knowledge about the network, regularisation of the network, and the introduction of uncertainty about the BNN predictions. Independent normal distributions are usually favoured for their simplicity and computational efficiency (Fortuin, 2021). The mean is set to zero and the standard deviation is the same for all weights. More complex parametrizations, such as matrix-valued normal distributions (Louizos and Welling, 2016) or hierarchical priors (MacKay, 1992), could improve performance, although our qualitative conclusions should not change.

Figure 4.3 illustrates the effects that the standard deviation of the prior has on the Expected Coverage Probability (ECP) described in Section 3.1. A model is considered overconfident if its ECP is below the diagonal and conservative if its ECP is above the diagonal. As it can be seen, large variances lead to overconfident predictions, while small variances can constrain the network too much, hindering its ability to learn the likelihood of the data (Figure 4.4).

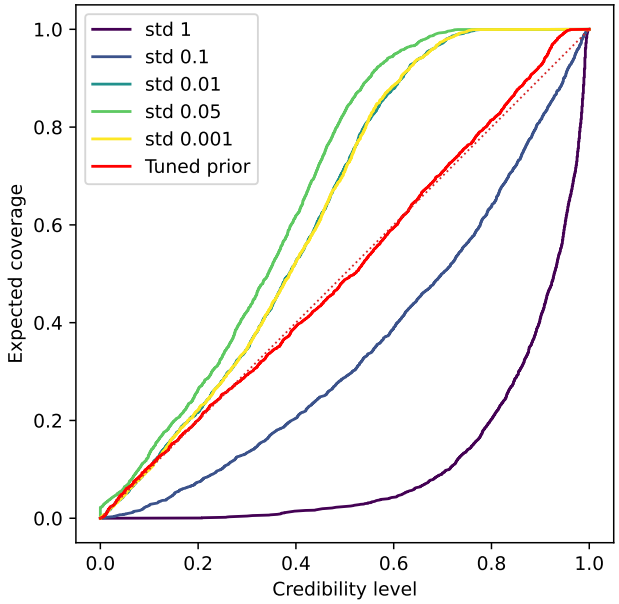


Figure 4.3: Impact of the standard deviation of the prior on the behaviour of the method. Small standard deviations (below 0.05) result in conservative predictions, while high values tend to produce overconfident results.

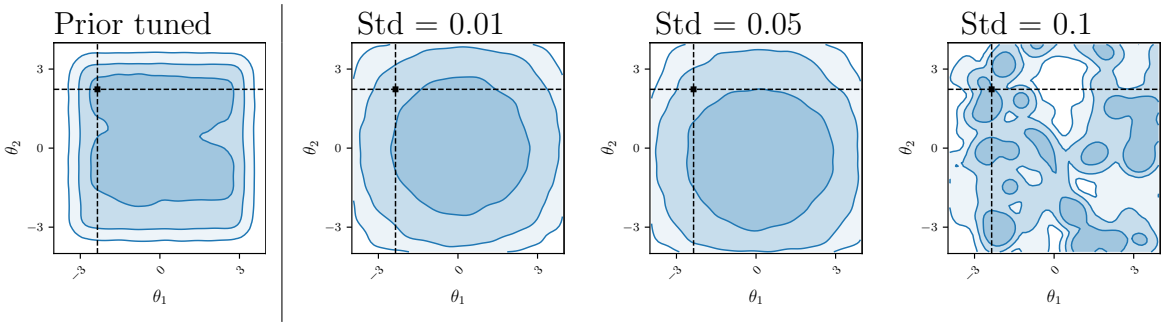


Figure 4.4: Impact of the standard deviation on the posterior a priori. High standard deviations lead to more complex distributions, complicating the learning process. Conversely, small standard deviations simplify the distributions but can restrict learning if the variance is too low.

4.3.2. Designing Priors over Neural Networks for Conservative Simulation-Based Inference

Credits: The work of Section 4.3.2 was developed by Arnaud Delaunoy as part of a joint effort to write a scientific paper.

Figure 4.4 showed that the use of independent normal distributions as priors over weights can lead to conservative results. However, the posterior distribution a priori, given by

$$p_{prior}(\boldsymbol{\theta}|\mathbf{x}) = \int_{\mathbf{w}} p(\mathbf{w}) p(\boldsymbol{\theta}|\mathbf{x}, \mathbf{w})\, d\mathbf{w}, \tag{4.6}$$

does not yield a distribution with desirable properties. Figure 4.3 shows the posteriors can be overconfident or conservative depending on the standard deviation value.

A desirable property for the prior distribution $p(\mathbf{w})$ is that Equation 4.6 provides the prior over parameters $p(\boldsymbol{\theta})$. This ensures that without data, the posterior approximation provides the expert's knowledge $p(\boldsymbol{\theta})$ such as $\hat{p}_{prior}(\boldsymbol{\theta}|\mathbf{x}) \approx p(\boldsymbol{\theta})$. In Section 4.3.1, the same standard deviation was used for all weights. Although adjusting this standard deviation could provide calibrated distributions, they would exhibit leakage (Figure 4.4). Leakage refers to the situation where the probability density distribution assigns a nonzero probability density to values outside the domain of the prior. This is undesirable because it means that the network is considering parameter values that we know are impossible.

To obtain well-calibrated posteriors a priori $\hat{p}_{prior}(\boldsymbol{\theta}|\mathbf{x})$, we propose to train a prior $p(\mathbf{w}) = q_\phi(\mathbf{w})$ such that

$$\hat{p}_{prior}(\boldsymbol{\theta}|\mathbf{x}) = \int_{\mathbf{w}} q_\phi(\mathbf{w}) \hat{p}(\boldsymbol{\theta}|\mathbf{x}, \mathbf{w})\, d\mathbf{w} \approx p(\boldsymbol{\theta}) \quad \forall \mathbf{x}. \tag{4.7}$$

The design of $q_\phi(\mathbf{w})$ is complex, as it can be difficult to ensure Equation 4.7 and learn an informative posterior $\hat{p}(\boldsymbol{\theta}|\mathbf{x})$ from it. The idea of the method is to train a variational family $q_\phi(\mathbf{w})$ to respect Equation 4.7 while being flexible enough so that the approximate posterior $\hat{p}(\mathbf{w}|D)$ can learn from it. The technical details of our method are provided in our scientific paper given in Appendix B.

Using this method, Figure 4.3 demonstrates that the resulting distributions are well-calibrated with a tuned prior, and that the uniform prior over parameters $p(\boldsymbol{\theta})$ is well

approximated (Figure 4.4).

4.3.3. Introducing Temperature to Improve Results

Following the identification of beneficial properties in weight priors in SBI, further experiments indicated that BNNs require a large number of simulations to learn an informative posterior over parameters $\hat{p}(\boldsymbol{\theta}|\mathbf{x})$ that significantly deviates from the prior $p(\boldsymbol{\theta})$. One strategy to solve this issue is the use of the Cold Posterior Effect (CPE) (Noci et al., 2021), which modifies the posterior over weights $p(\mathbf{w}|D)$ by adjusting the influence of the data likelihood with a temperature hyperparameter T

$$p(\mathbf{w}|D) \propto p(\mathbf{w}) \cdot p(D|\mathbf{w})^{1/T}.$$

The choice of T affects the sharpness of the posterior; a temperature below one increases the impact of the likelihood, thus tightening the posterior around the observed data. Conversely, a temperature greater than one increases the influence of the prior, leading to a broader posterior distribution over weights $\hat{p}(\mathbf{w}|D)$.

In VI, one maximises the Evidence Lower Bound Objective (ELBO) and not directly the posterior distribution over the weights. It can be shown that the increase of the likelihood part of the ELBO by a factor $\frac{1}{T}$ is equivalent to the previous expression (Wenzel et al., 2020). This means that the ELBO becomes

$$\text{ELBO}(\nu) = \frac{1}{T} E_{\hat{p}_\nu(\mathbf{w}|D)} \left[\log \hat{p}(\boldsymbol{\theta}|\mathbf{x}, \mathbf{w})\right] - KL(\hat{p}_\nu(\mathbf{w}|D)||p(\mathbf{w})).$$

To differentiate when we use temperature or not, we will denote the Bayesian methods without temperature as Bayesian Neural Posterior Estimation (BNN-NPE) and Bayesian Neural Ratio Estimation (BNN-NRE). The Bayesian method with temperature are denoted Bayesian Neural Posterior Estimation with Temperature (BNN-NPE$_\text{T}$) and Bayesian Neural Ratio Estimation with Temperature (BNN-NRE$_\text{T}$).

4.4. Summary

In data-poor regimes, conservative methods tend either to still be overconfident or provide uninformative approximate posteriors $\hat{p}(\boldsymbol{\theta}|\mathbf{x})$. Overfitting occurs because different approximate posteriors explain well the data which brings computational uncertainty. The integration of BDL into SBI allows to take into account computational uncertainty in the prediction of the posterior over parameters $\hat{p}(\boldsymbol{\theta}|\mathbf{x})$. In data-poor regimes, where computational uncertainty is high, this should increase the conservativeness of the posterior by aggregating all plausible approximate posteriors in the prediction.

Computational uncertainty is taken into account using BNNs which turn the fixed weights of neural networks into distributions. BDL aims at computing the posterior distribution over weights $p(\mathbf{w}|D)$, where D is the dataset. This posterior distribution requires a prior over weights to be computed $p(\mathbf{w})$. A principled way to learn the prior over weights tailored to SBI is introduced. The resulting distribution from this learned prior over weights models the prior over parameters $p(\boldsymbol{\theta})$. Currently, this new method is the only one that allows to model the prior $p(\boldsymbol{\theta})$ for NPE in data-poor regimes.

Finally, in order to be competitive with current SBI methods, the temperature hyperparameter must be used on the posterior distribution over weights $p(\mathbf{w}|D)$ to increase the importance of the data.

5. Experiments

The code used for this thesis is available at `https://github.com/MdelaBrassinne/TFE`.

5.1. Protocol

5.1.1. Simulation-Based Inference Methods

In the following sections, our new Bayesian methods, Bayesian Neural Posterior Estimation (BNN-NPE) and Bayesian Neural Ratio Estimation (BNN-NRE), are compared to classical methods. Our Bayesian methods without temperature are denoted BNN-NPE and BNN-NRE and our Bayesian methods with temperature are denoted Bayesian Neural Posterior Estimation with Temperature (BNN-NPE$_\mathrm{T}$) and Bayesian Neural Ratio Estimation with Temperature (BNN-NRE$_\mathrm{T}$). Classical methods include traditional Neural Posterior Estimation (NPE) and Neural Ratio Estimation (NRE). Conservative methods include Balanced Neural Posterior Estimation (BNPE), Balanced Neural Ratio Estimation (BNRE), Ensemble NPE and Ensemble NRE. The architectures are described in Appendix A.2.1.

For BNN-NPE and BNN-NRE, the prior for Bayesian Neural Networks (BNNs) is the tuned prior over weights described in Section 4.3.2 unless specified otherwise.

For BNN-NPE$_\mathrm{T}$ and BNN-NRE$_\mathrm{T}$, the temperature varies between 0.01 and 1. It is set such that the Expected Coverage Probability (ECP) described in Section 3.1 remains in the conservative regime.

5.1.2. Benchmarks

Four benchmarks are used to compare the methods: **SLCP** (Papamakarios et al., 2019b), **Two Moons** (Papamakarios and Murray, 2016), **Lotka Volterra** (Lotka, 1920, 1927) and **Spatial SIR** (Hermans et al., 2021). The first two benchmarks were chosen because the posterior distribution is complex which often leads to overconfident results. The others were chosen because they have high-dimensional observations which leads to many training difficulties. All of these benchmarks use a uniform distribution as a prior over the parameters $\boldsymbol{\theta}$. These benchmarks are described in more details in Appendix A.3.

5.1.3. Performance Metrics

To evaluate the performance of each method, we will use the Expected Log Posterior (ELP), the three expected uncertainties and the ECP described in Section 3.1.

1. **ELP**: The ELP is defined as

$$\text{ELP} = \mathrm{E}_{p(\boldsymbol{\theta},\mathbf{x})}[\log \hat{p}(\boldsymbol{\theta}|\mathbf{x})]. \tag{5.1}$$

   The higher the quantity, the better. Indeed, if this quantity is high, this means that the posteriors predict that the real $\boldsymbol{\theta}$ is compatible with the observation $\mathbf{x}$.

2. **Expected Total Uncertainty (ETU)**: Using Equation 4.3, the ETU is defined as

$$\text{ETU} = \mathrm{E}_{p(\boldsymbol{\theta},\mathbf{x})}[TU(\mathbf{x}|D)] = \mathrm{E}_{p(\boldsymbol{\theta},\mathbf{x})}[H[\hat{p}(\boldsymbol{\theta}|\mathbf{x})]]. \tag{5.2}$$

3. **Expected Aleatoric Uncertainty (EAU)**: Using Equation 4.4, the EAU is defined as

$$\text{EAU} = \mathrm{E}_{p(\boldsymbol{\theta},\mathbf{x})}[AU(\mathbf{x}|D)] = \mathrm{E}_{p(\boldsymbol{\theta},\mathbf{x})}[\mathrm{E}_{\hat{p}(\mathbf{w}|D)} H[\hat{p}(\boldsymbol{\theta}|\mathbf{x},\mathbf{w})]]. \tag{5.3}$$

4. **Expectec Epistemic Uncertainty (EEU)**: Using Equation 4.5, the EEU is defined as

$$\text{EEU} = \mathrm{E}_{p(\boldsymbol{\theta},\mathbf{x})}[EU(\mathbf{x}|D)] = \mathrm{E}_{p(\boldsymbol{\theta},\mathbf{x})}[TU(\mathbf{x}|D) - AU(\mathbf{x}|D)]. \tag{5.4}$$

   In practice, these quantities are evaluated using a Monte Carlo approximation.

The results were computed on three different runs using different simulations for each run. Median, minimum and maximum values are indicated when possible. The optimal results were computed by training NPE with 1 000 000 simulations.

5.2. Comparison Between Conservative Methods and Bayesian Methods

The two main factors to analyse the posteriors are the conservativeness of the posteriors using the ECP diagnostic (Section 3.1) and the information contained in the posteriors using the ELP.

As a reminder, approximate posteriors are said to be conservative if their ECP is above the diagonal for all levels α and overconfident if it is under the diagonal for all levels α.

5.2.1. Conservativeness of Posteriors

We first analyse the conservativeness with the ECP diagnostic. Figure 5.1 depicts the ECP for all methods for all 4 benchmarks. **Bayesian methods (BNN-NPE and BNN-NRE) and BNRE are the three methods which consistently are conservative** as their ECP is above the diagonal in all cases. Although there may be a slight exception towards overconfidence when very few simulations are available, these methods remain well-calibrated most of the time. All other methods can be overconfident in data-poor regimes.

In general, NPE tends to exhibit more overconfidence compared to NRE. This difference occurs because the NRE initialization models the prior distribution $p(\boldsymbol{\theta})$, thereby achieving initial well-calibration, unlike NPE. However, as expected, **when the number of simulations available increases, all methods tend to be well-calibrated and less overconfident.**

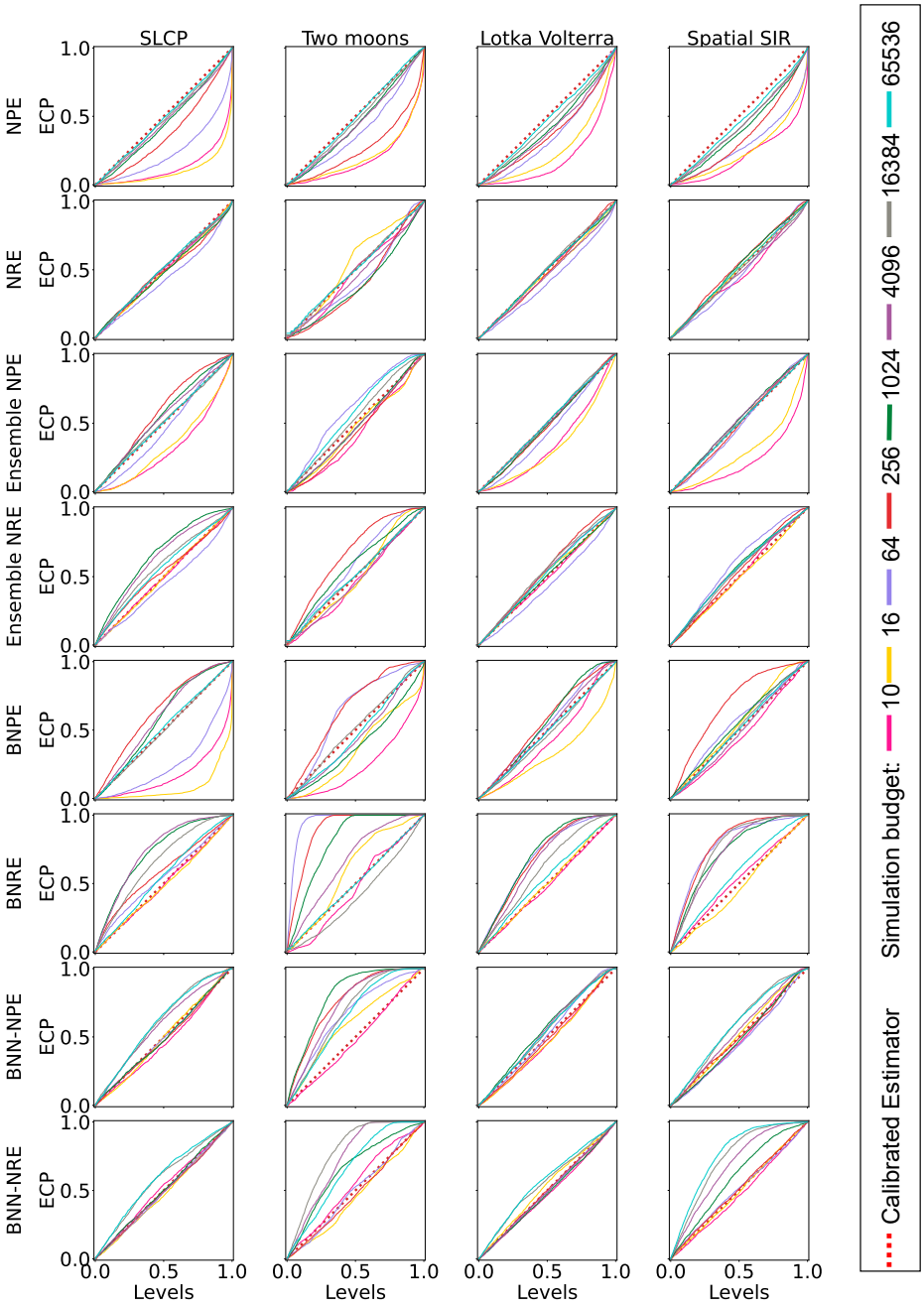


Figure 5.1: ECP of all methods on all benchmarks. Although all methods may be slightly overconfident, our Bayesian methods (BNN-NPE, BNN-NRE) and BNRE demonstrate the best reliability in terms of minimal overconfidence. Indeed, they are always above the diagonal curve. However, our method is closer to well-calibration compared to BNRE.

5.2.2. Information Contained in Posteriors

Although Figure 5.1 indicates that Bayesian methods are consistently conservative, their practical utility may be limited if the approximate posteriors $\hat{p}(\boldsymbol{\theta}|\mathbf{x})$ closely resembles the prior distribution $p(\boldsymbol{\theta})$. To quantify the information contained in the posterior, the ELP is used.

Figure 5.2 illustrates the evolution of the ELP for all methods on all 4 benchmarks where higher values indicate more informative posteriors. The plot presents the median, minimum, and maximum ELP over three runs. The ELP of the prior distribution $p(\boldsymbol{\theta})$ is shown to compare the information gain of the methods with the prior.

Bayesian methods need more simulations to achieve a similar ELP than traditional methods like NPE and NRE. This behaviour is attributed to the restrictive nature of the prior over the weights leading to broader distributions than classical methods. Consequently, the posteriors produced by Bayesian methods are not informative in data-poor regimes. However, it is important to point out that other methods like NPE perform worse than the prior $p(\boldsymbol{\theta})$ in data-poor regimes because they are overconfident while our method is at least as good as the prior $p(\boldsymbol{\theta})$.

All methods improve the ELP when using more simulations except for BNN-NRE on the SLCP, Lotka-Volterra and spatial SIR benchmarks, where no significant learning was observed. We believe this is due to the restrictive nature of the prior over weights which makes it very hard for BNN-NRE to learn the true posterior distribution $p(\boldsymbol{\theta}|\mathbf{x})$.

5.2.3. Using Temperature to Improve the Posteriors

As discussed in the previous section, Bayesian methods have a lower ELP because the prior over BNNs restricts too much the learning. The temperature hyperparameter described in Section 4.3.3 allows BNNs to focus more on the data and less on the prior. Figure 5.3 shows **BNN-NPE$_\mathbf{T}$ can achieve comparable ELP to NPE on the SLCP benchmark and stay conservative by adjusting the temperature hyperparameter.**

Compared to BNRE, the only method as conservative as Bayesian methods, BNN-NPE$_\mathrm{T}$ provides more informative posteriors for the same number of simulations available. This makes them more useful under limited simulations available.

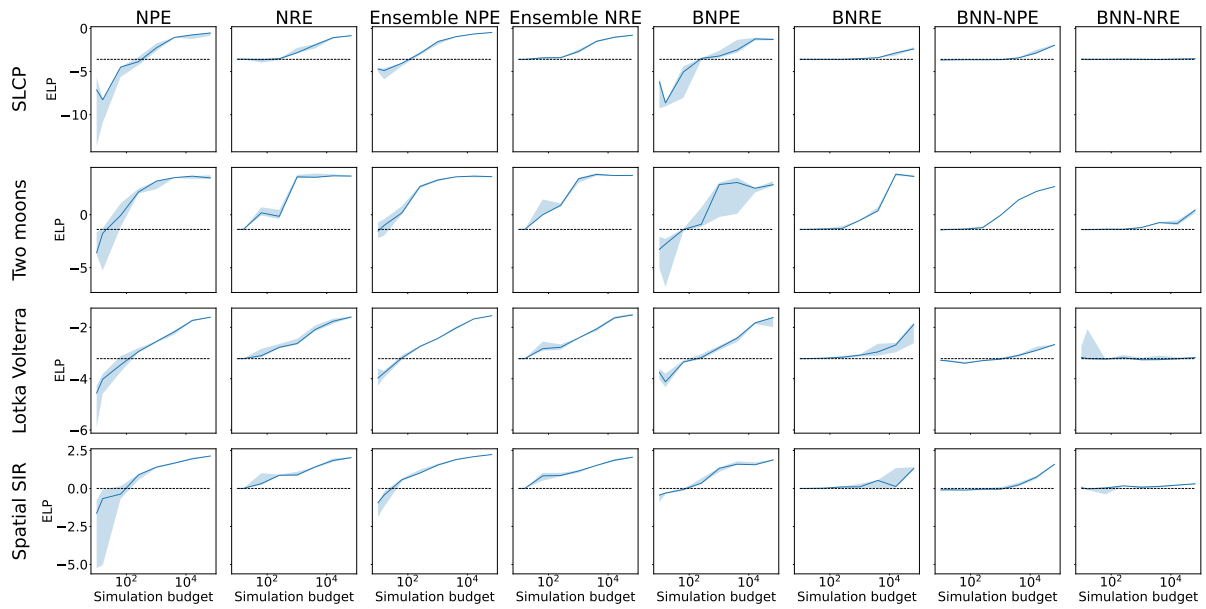


Figure 5.2: ELP of all methods for all benchmarks. The black dashed line is the ELP of the prior over parameters $p(\boldsymbol{\theta})$. Bayesian methods have a lower ELP than traditional methods (like NPE and NRE) when a large number of simulations are available. This implies distributions from Bayesian methods have less information than distributions from traditional methods. In data-poor regimes, the ELP of Bayesian methods remains close to the prior which means the approximate posteriors do not contain more information to the prior.

Overall, Bayesian methods present the best choice for assuring conservative posteriors. Nonetheless, adjusting the temperature setting when more simulations are available is necessary to obtain ELPs comparable with classical methods (Figure 5.3).

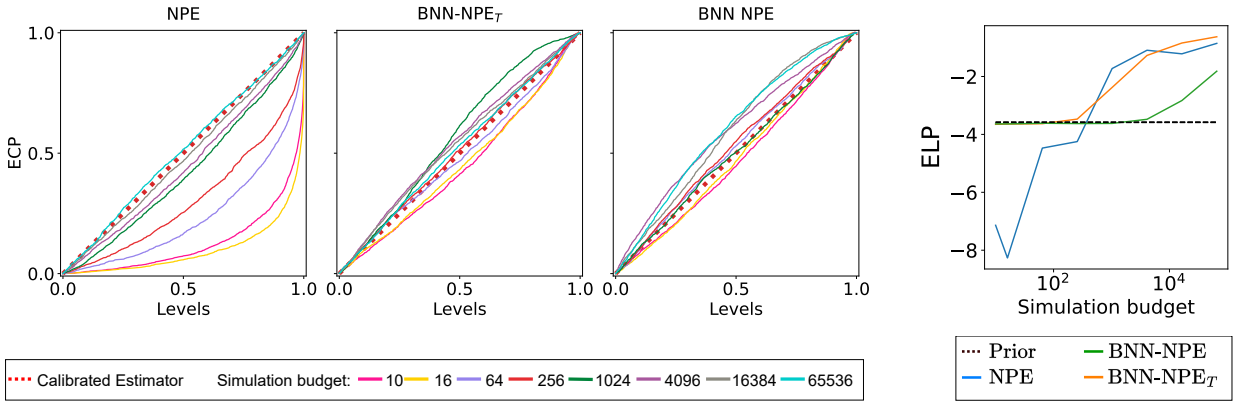


Figure 5.3: ECP and ELP comparisons for NPE and BNN-NPE on the SLCP benchmark. Bayesian methods remain conservative as their ECP are above the diagonal no matter the number of simulations available. However, without temperature adjustments, BNN-NPE does not match the ELP of NPE. With temperature, BNN-NPE$_\mathrm{T}$ maintains conservativeness and achieves comparable ELPs.

5.2.4. Uncertainty Analysis

Aleatoric and Total Uncertainty

In this analysis, we focus on the behaviour of total and aleatoric uncertainties as a function of the number of simulations available. Figure 5.4 illustrates that **both types of uncertainty decrease when more simulations are available.** This reduction is associated with improved posterior estimates, where more simulations available allow for a tighter distribution around the true parameter values, thus reducing uncertainty. This phenomenon can be qualitatively observed in Figure 5.5. In this figure, the Bayesian Model Average (BMA) distribution is tighter and more precise when more samples are used to train the network.

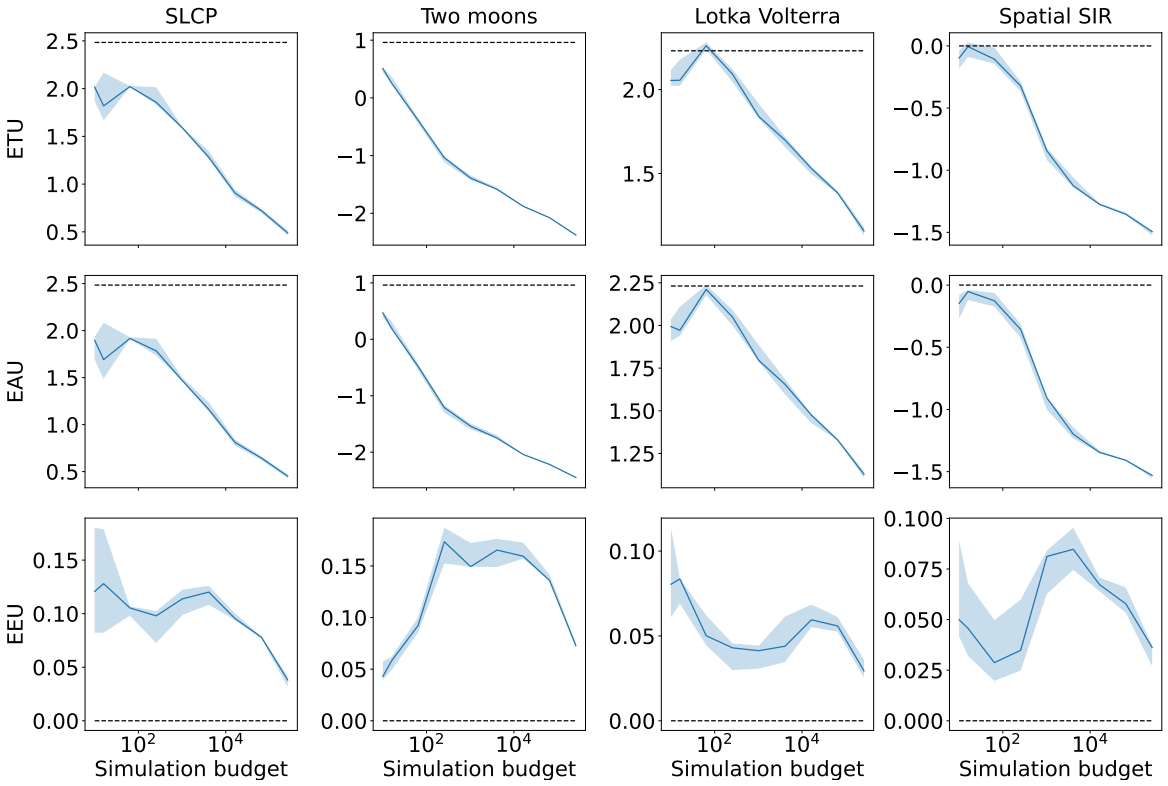


Figure 5.4: ETU, EAU, and EEU for BNN-NPE$_{\mathrm{T}}$ on all benchmarks. The black dashed line is the prior. Both total and aleatoric uncertainties decrease with more simulations available. Epistemic uncertainty presents a less predictable pattern, possibly increasing initially as the approximate posteriors are more complex, but typically decreases subsequently when many simulations are available.

Epistemic Uncertainty

The behaviour of epistemic uncertainty is potentially linked to the complexity of the posterior distributions $\hat{p}(\boldsymbol{\theta}|\mathbf{x})$. **With more simulations available, more complex posteriors are modelled, leading to greater uncertainty because the posterior is more and more complex.** However, as the number of simulations available further increases, this uncertainty diminishes, indicating the network grows confidence in these complex posteriors. Discussion about the confidence of the network in its own prediction will be detailed in Section 5.3.2.

The same trend is qualitatively shown in Figure 5.5. In data-poor regimes, the prior of the network $p(\mathbf{w})$ dominates, resulting in posteriors that are relatively flat and uniform. As the number of simulations increases, the posteriors become more complex, better fitting the data but also introducing more computational uncertainty. Nonetheless, when many simulations are available, the epistemic uncertainty reduces as the approximate posteriors align closely with the true posterior distribution.

However, although there are some variations explaining the increase of epistemic uncertainty, it is important to notice that **most of the uncertainty comes from the aleatoric uncertainty**. This is confirmed qualitatively by the distributions in Figure 5.5 where the variations between distributions are not huge. Quantitatively, Figure 5.4 shows the aleatoric uncertainty value is much higher than the epistemic uncertainty value. The reason we believe the uncertainty is mostly aleatoric comes from the prior over weights $p(\mathbf{w})$. Indeed, the initial uncertainty from the prior is mainly aleatoric. This is a strong constraint which makes it hard for the network to learn the likelihood of the data.

5.2.5. Performance Using a Tuned Prior for Bayesian Neural Networks

In this analysis, we compare BNN-NPE using the tuned prior over weights described in Section 4.3.2 against BNN-NPE using standard independent normal distributions with zero mean and the same standard deviation for all weights as the prior over the weights. Figure 5.6 illustrates that while both methods maintain conservative ECPs, **the method with the tuned prior demonstrates better calibration but requires more data**

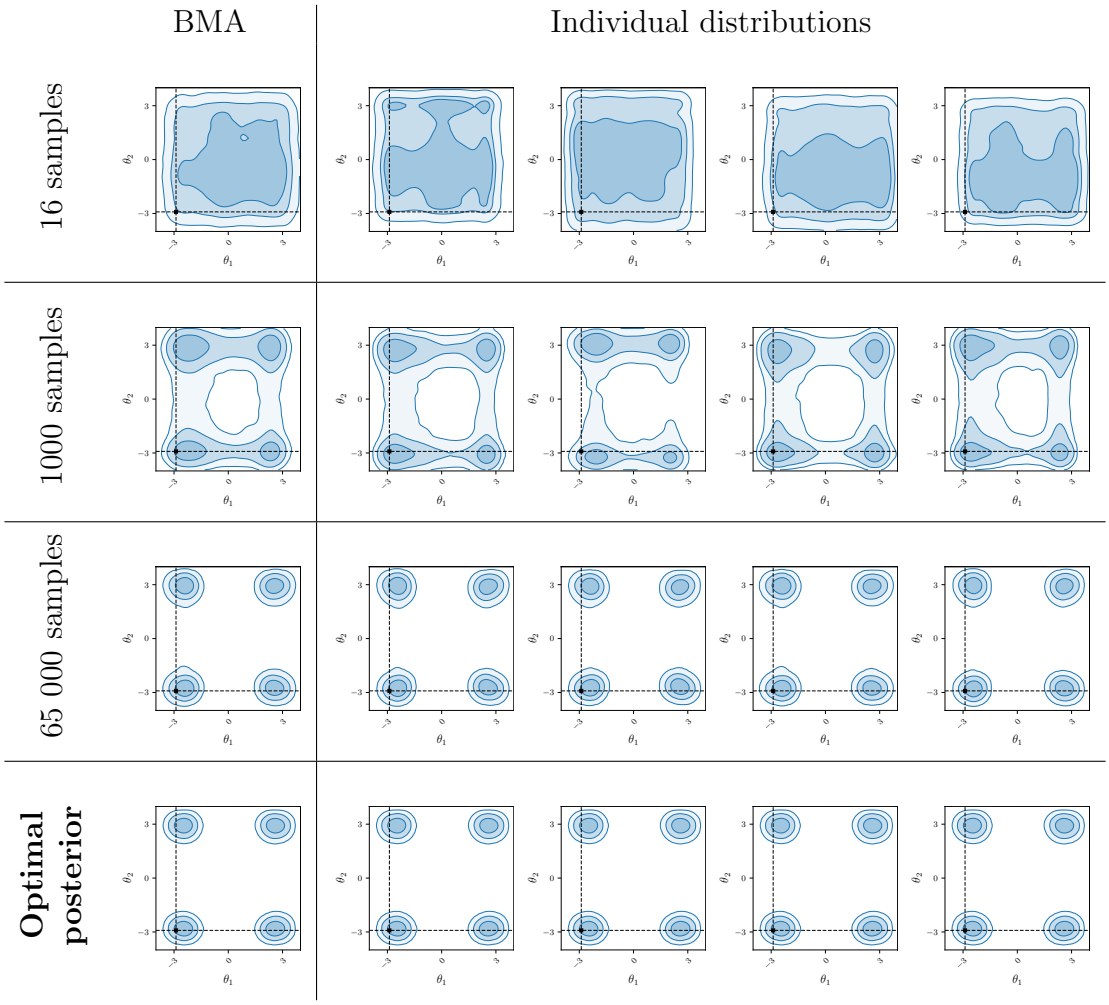


Figure 5.5: Comparison of BNN-NPE$_{\mathrm{T}}$ posteriors depending on the number of simulations available on the SLCP benchmark. As sample size increases, all methods tend to align with the true multimodal posterior, with earlier stages dominated by aleatoric uncertainty and later stages showing reduced variation and lower epistemic uncertainty.

to achieve comparable ELPs. This suggests that although the tuned prior improves calibration in Simulation-Based Inference (SBI), it may slightly slow down the learning of the network.

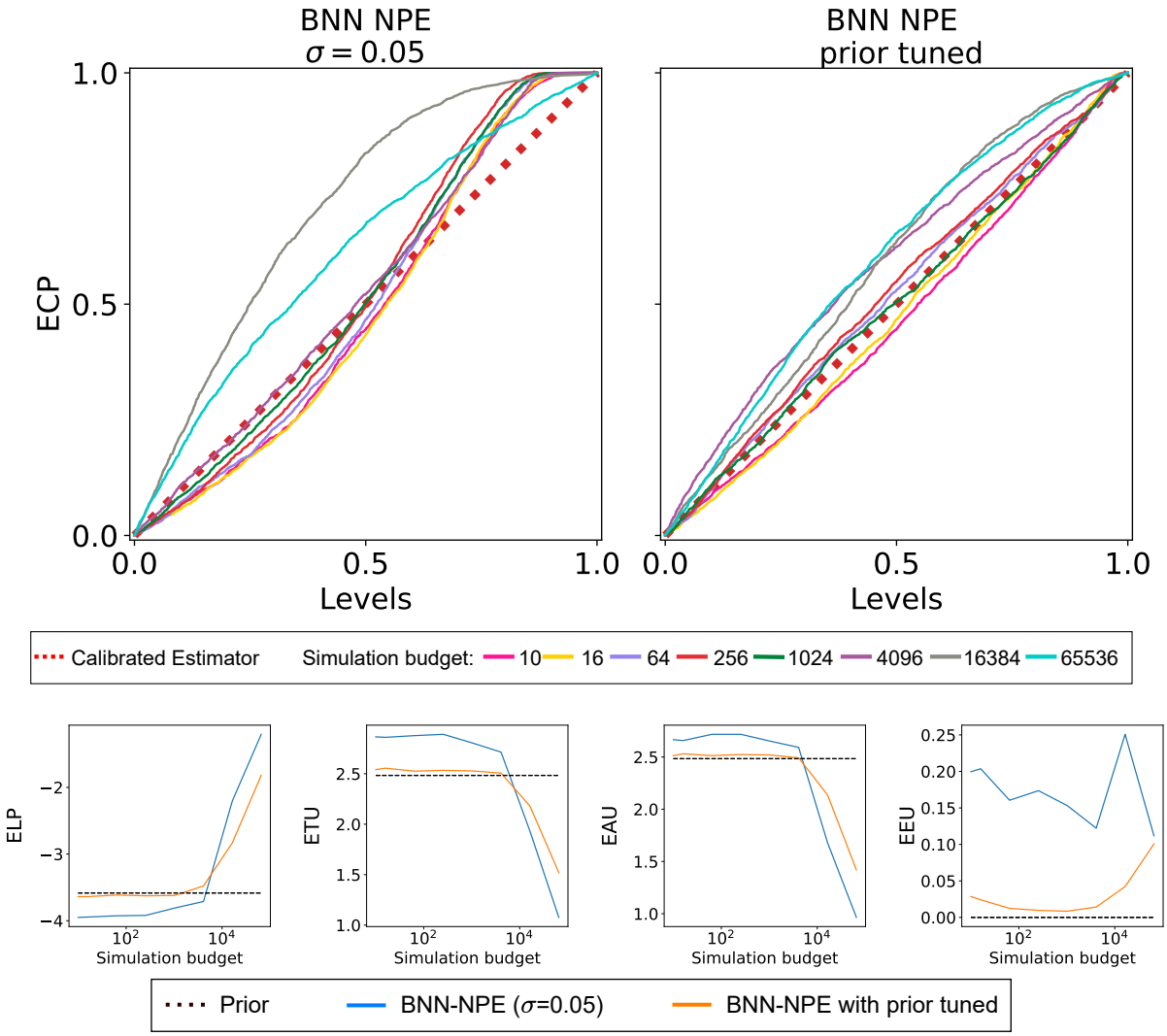


Figure 5.6: Coverage, ELP, and uncertainties for BNN-NPE with and without a tuned prior on the SLCP benchmark. Both configurations yield conservative results across all tested regimes. The tuned prior method exhibits initial better calibration but requires more data to match the ELP of the untuned method, likely due to increased total and aleatoric uncertainties.

The initial superiority in the ELP for the tuned prior method can be attributed to the leakage in the posteriors without the tuned prior. Leakage refers to the situation where the probability density distribution assigns non-zero probability density to values outside the

domain of the prior. This is undesirable because it means that the network is considering parameter values that we know are impossible. This effect diminishes as the number of simulations available increases (Figure 5.6). Indeed, when more simulations are available, the uncertainty for the tuned method becomes higher than for the untuned method. Qualitatively, this behaviour can be observed in Figure 5.7. Initially, the uniform prior over parameters $p(\boldsymbol{\theta})$ is well modelled by the method with the tuned prior. Increasing the number of simulations available, the posterior becomes more tightened around the true parameters for the method without the prior tuned.

Finally, one can observe the epistemic uncertainty for BNN-NPE with the prior tuned is always below BNN-NPE using the same standard deviation for all weights. This is mainly due to a hyperparameter in the method tuning the prior described in our scientific paper (Appendix B). This hyperparameter allows control of the epistemic uncertainty of the method. In this case, it was low but it can be increased if needed.

5.3. Posterior Analysis

In this section, we will focus exclusively on understanding the posterior distribution $\hat{p}(\boldsymbol{\theta}|\mathbf{x})$ created by BNNs on the SLCP benchmark.

5.3.1. Evolution of the Posterior for Simulation-Based Inference Methods

For Bayesian methods, the information contained in the posteriors increases with the number of simulations available (Figure 5.1). The qualitative effect of the conservativeness is shown in Figure 5.8.

In data-poor regimes, BNN-NPE and BNRE model the prior quite well while BNPE and NPE model a random distribution. This is the effect of overconfidence explained in Section 5.2.1.

When a moderate number of simulations are available (between 250 and 16 000 samples), NPE learns quite fast an informative posterior distribution although it might be overconfident. BNN-NPE$_\mathrm{T}$ and BNPE learn at a smaller pace, but the results are more

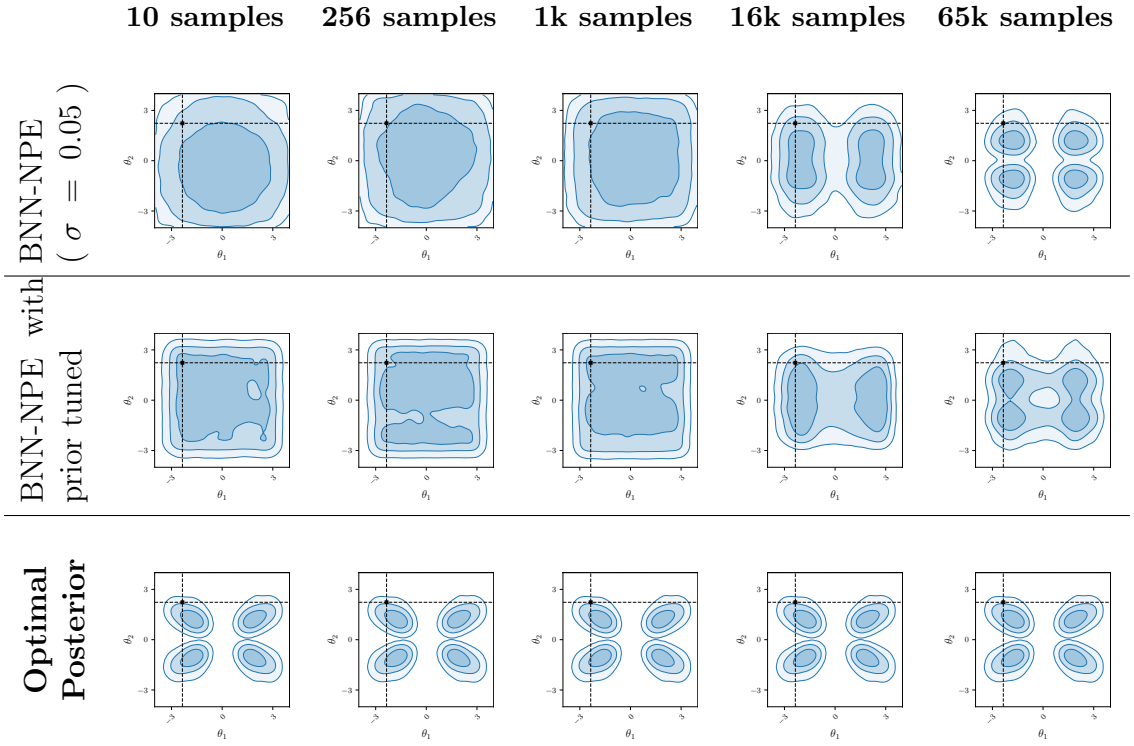


Figure 5.7: Comparison of posteriors for BNN-NPE with and without the tuned prior over weights for different data regimes on the SLCP benchmark. The initial distribution is better using a tuned prior as the posterior $p(\boldsymbol{\theta}|\mathbf{x})$ is very similar to the uniform prior $p(\boldsymbol{\theta})$. However, when increasing the number of simulations available, the posterior becomes better without the tuned prior.

conservative.

Finally, when a large number of simulations are available, almost all methods learn a similar distribution except for BNRE which still has room for improvement.

Overall, Figure 5.8 shows qualitatively the same conclusions as Figures 5.1 and 5.2. **In data-poor regimes, BNN-NPE and BNRE are the most conservative methods, but they need more simulations to achieve informative posteriors. However, introducing the temperature allows to model a posterior which is conservative and informative using the same number of simulations.**

5.3.2. Posterior Quality Depending on the Observations

In this section, we analyse how BNN-NPE$_{\mathrm{T}}$ performs depending on the distribution of the observations $\mathbf{x}$. Specifically, we investigate how the results vary for rare and common observations $\mathbf{x}$. To achieve this, we estimate the distribution of the observations $\mathbf{x}$ using kernel density estimation (Silverman, 2018) on a dataset of 1 million samples. Subsequently, we divide the test set into quantiles based on the estimated distribution. This analysis is performed using the SLCP benchmark. In Appendix A.1.2, **it is shown the true posterior distribution should still be well-calibrated for all quantiles although the prior distribution should not necessarily do.** This implies that while modelling the prior distribution initially leads to conservative results in expectation, there is no guarantee to be conservative on a specific observation $\mathbf{x_0}$.

In Figure 5.9, the ECP for NPE and BNN-NPE$_{\mathrm{T}}$ depending on the quantiles is shown. Rare observations correspond to lower quantiles while common observations correspond to higher quantiles. The results indicate that BNN-NPE$_{\mathrm{T}}$ is not perfectly calibrated for all observations $\mathbf{x}$. For the rare samples in the lower quantiles, the method is not well-calibrated. Therefore, **the posteriors are conservative in expectation, but not for every individual sample.** This can be expected as the approximate posterior $\hat{p}(\boldsymbol{\theta}|\mathbf{x})$ is very close to the prior $p(\boldsymbol{\theta})$ which is not necessarily well-calibrated for all observations $\mathbf{x}$. However, as the number of simulations increases, the method becomes well-calibrated and conservative across all quantiles. However, BNN-NPE$_{\mathrm{T}}$ remains more conservative than NPE in all cases.

Another interesting observation is that **BNN-NPE$_{\mathbf{T}}$ can recognize when it has in-**

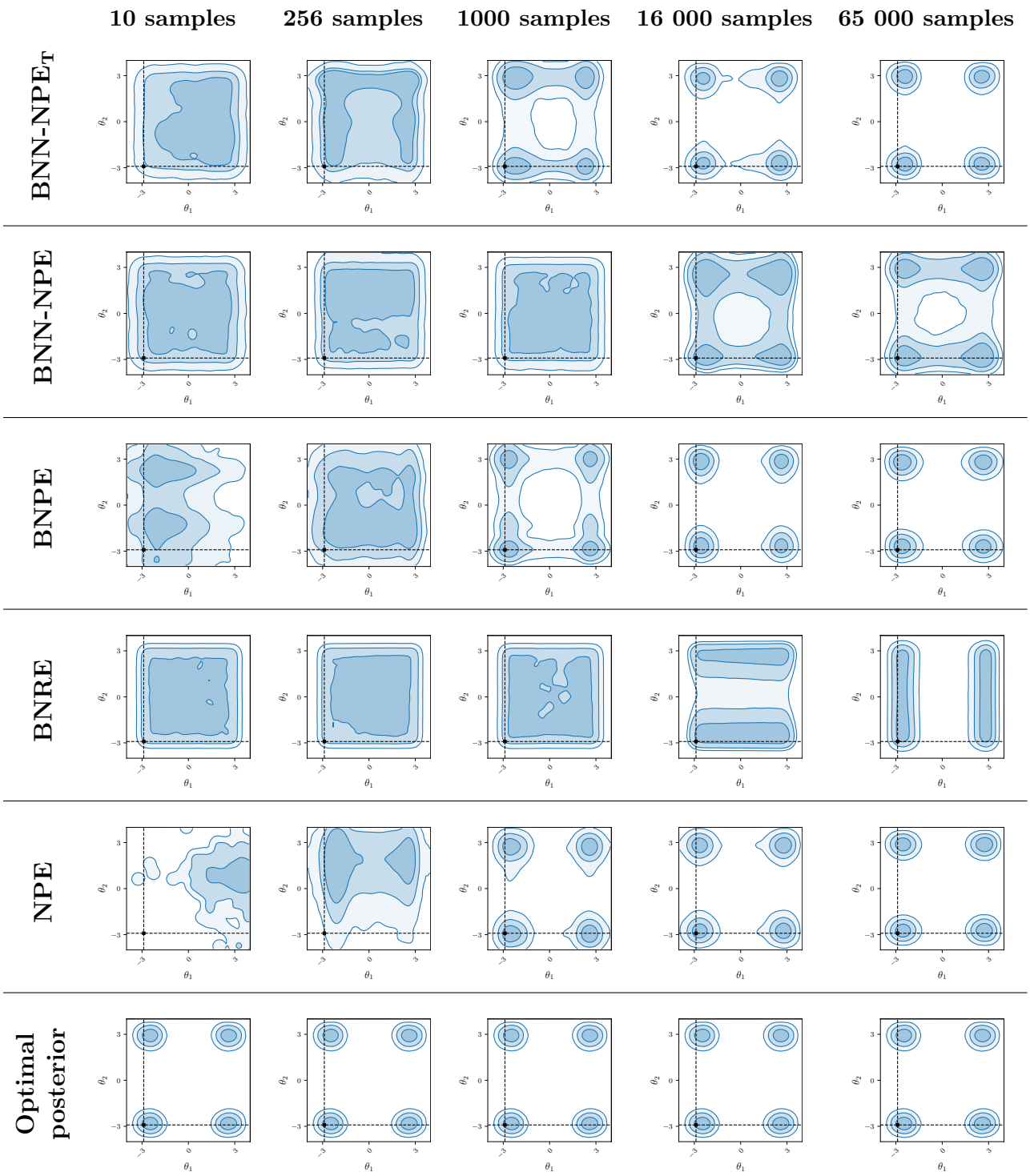


Figure 5.8: Comparison of posteriors for different methods depending on the number of simulations available on the SLCP benchmark. Adjusting the temperature in Bayesian methods enables better ELP with fewer samples while maintaining conservative posteriors.

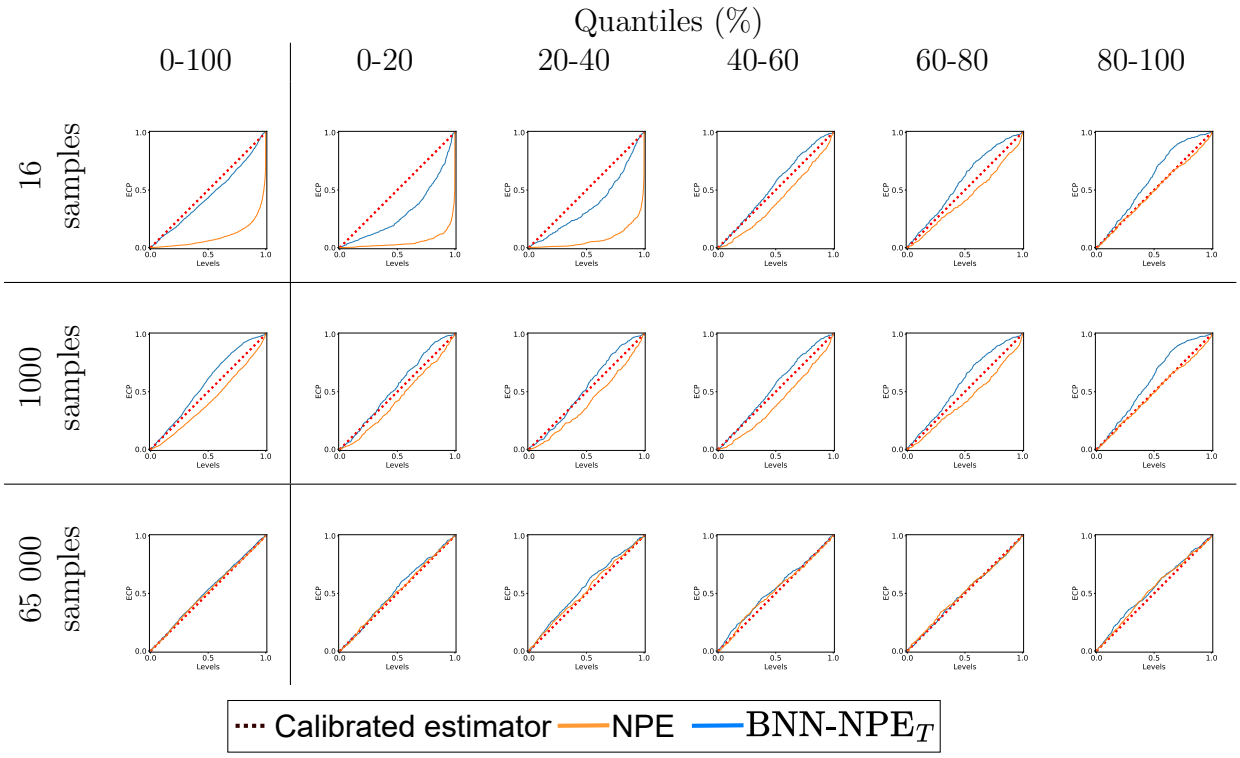
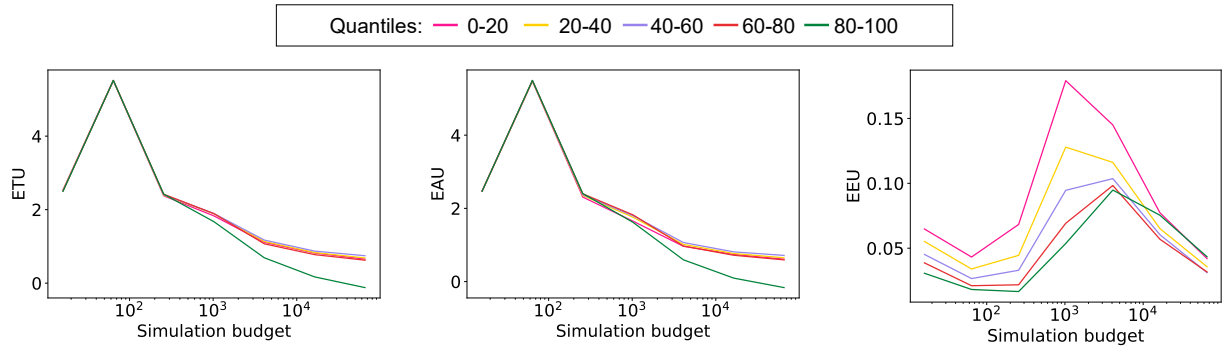


Figure 5.9: ECP of BNN-NPE and NPE depending on the distribution of $\mathbf{x}$ depending on the number of simulations available on the SLCP benchmark. Lower quantiles correspond to rare observations while higher quantiles correspond to common observations. While the ECP on rare samples (quantile 0-20) is generally worse than on common sample (quantile 80-100), no significant difference appears for moderate and large number of simulations available.

sufficient knowledge about the observation x. This knowledge can be leveraged in active learning (Haut et al., 2018). In Figure 5.10c, it is shown the epistemic uncertainty is consistently higher for rarer observations $\mathbf{x}$ as shown by lower quantiles. This indicates greater variability in individual distributions for rare observations $\mathbf{x}$. This can also be visually observed in Figure 5.11 where the BMA and individual distributions are shown for BNN-NPE$_{\mathrm{T}}$ on the SLCP benchmark for 1000 simulations available. For the common sample, almost all individual distributions are the same as there is almost no epistemic uncertainty. For the rare sample, individual distributions vary more as the epistemic uncertainty is higher. However, as already pointed out in Section 5.2.4, most of the uncertainty is aleatoric which explains why the variations are not bigger.

While the uncertainty is lower for high quantiles, this could be due to tighter posterior distributions rather than an inherent method characteristic. Figure 5.12 shows the posterior for the 80-100 quantile appears more compact than that for the 0-20 quantile.



(a) Total uncertainty (b) Aleatoric uncertainty (c) Epistemic uncertainty

Figure 5.10: Evolution of the uncertainty depending on the number of simulations available for different quantiles for BNN-NPE on the SLCP benchmark. Lower quantiles correspond to rare observations while higher quantiles correspond to common observations. The total and aleatoric uncertainty remain similar for all quantiles except for very common observations. This might be due to those observations having a tighter posterior. However, the epistemic uncertainty varies clearly depending on the quantile. The rarer the samples, the more epistemic uncertainty. This implies the network knows when it is more uncertain.

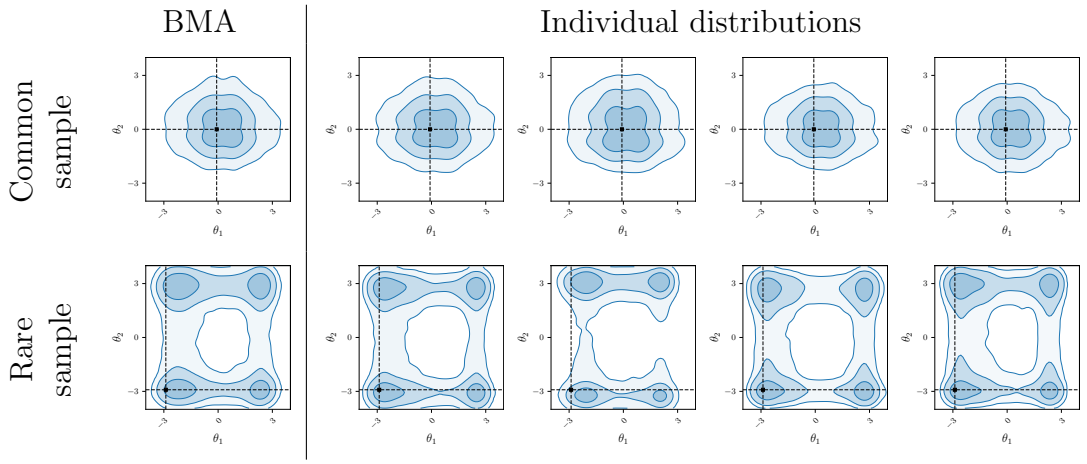


Figure 5.11: Comparison of BNN-NPE posteriors for common and rare observations $\mathbf{x}$ on the SLCP benchmark. As it can be seen, there are more variations across individual distributions for rare observations than common observations because the epistemic uncertainty is higher.

5.4. Summary

In this thesis, we have compared all SBI methods on different benchmarks with our new Bayesian methods, BNN-NPE and BNN-NRE. This comparison is based on two criteria: the conservativeness and the information contained in the posteriors produced by the methods. Qualitative and quantitative evidence of the conclusions here-below were shown in this chapter.

In terms of conservativeness, Bayesian methods (BNN-NPE and BNN-NRE) and BNRE are always conservative in expectation on the test data. All other methods fail at being conservative in data-poor regimes.

To have informative posterior approximations, Bayesian methods need more simulations than traditional methods (NPE and NRE) to reach informative posteriors. However, introducing the temperature hyperparameter allows Bayesian methods to reach posteriors at least as informative as the best methods in all data regimes.

In problems with high-dimensional inputs, BNN-NRE was unable to provide informative posteriors but BNN-NPE managed to do so. Therefore, BNN-NPE is a bit more flexible than BNN-NRE. However, when there is only a very limited number of simulations available, no method is able to provide informative posteriors. While BNN-NPE$_\mathrm{T}$ can do at

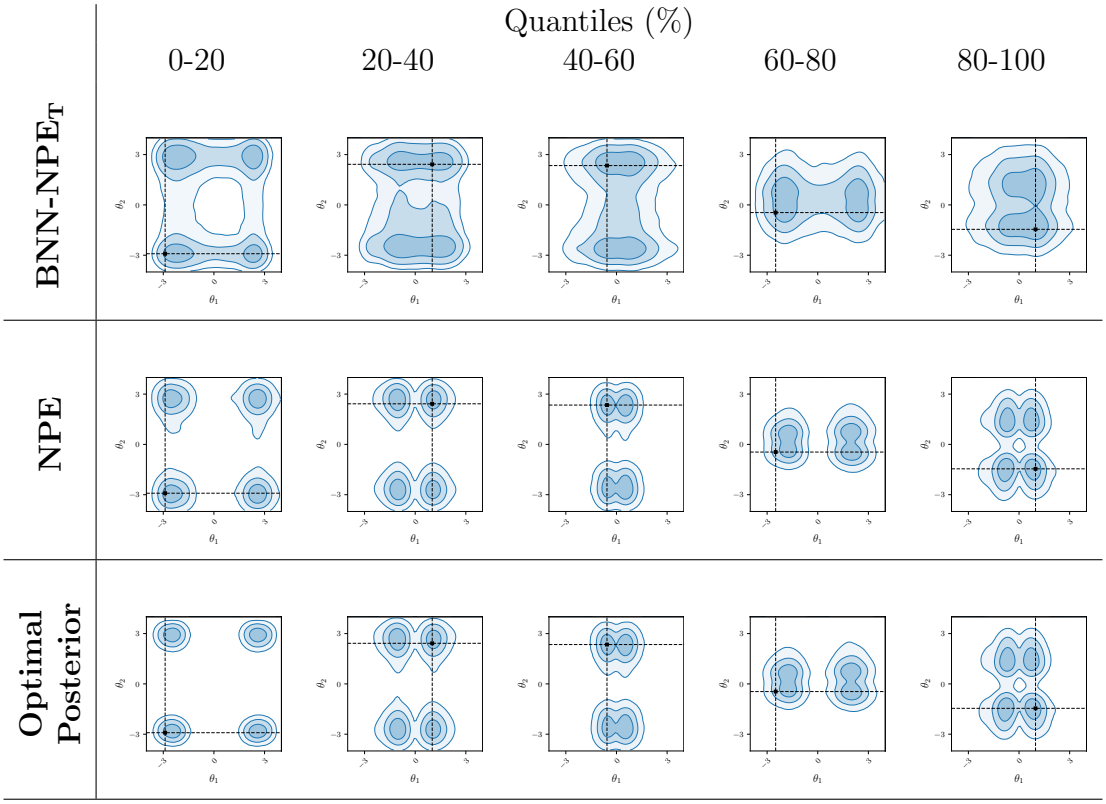


Figure 5.12: Comparison of posteriors depending on the distribution of $\mathbf{x}$ for BNN-NPE and NPE trained on a dataset of 1k simulations on the SLCP benchmark. Lower quantiles correspond to rare observations while higher quantiles correspond to common observations. While NPE seems to appear to have better posteriors, they are slightly overconfident. While BNN-NPE is more conservative, it still achieves similar results although the uncertainty is higher as expected.

least as good as any other SBI methods, the approximate posteriors are no better than the prior over parameters $p(\boldsymbol{\theta})$ in data-poor regimes.

Incorporating computational uncertainty like in BNN-NPE allows us to know what the network does not know. In particular, the network increases computational uncertainty for rare observations $\mathbf{x}$ and decreases it for common observations $\mathbf{x}$. Finally, methods are less conservative on rare observations $\mathbf{x}$ than on common samples.

Overall our best method, BNN-NPE$_{\mathrm{T}}$, was conservative in all data regimes and matches the results of the best methods in all data regimes.

6. Application in Cosmology

6.1. Description of the Cosmological Problem

Credits: Section 6.1 was written by Arnaud Delaunoy as part of a joint effort to write a scientific paper.

To showcase the utility of Bayesian Deep Learning (BDL) for Simulation-Based Inference (SBI) 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. Hence, the observable is a vector of 24 features.

6.2. Results

To showcase the results, three methods have been chosen. First, Ensemble Neural Posterior Estimation (NPE) was chosen to have a base case. Balanced Neural Ratio Estimation (BNRE) was also chosen in order to have a 100% conservative method. Finally, Bayesian Neural Posterior Estimation with Temperature (BNN-NPE$_\mathrm{T}$) was selected as it is our best method.

Figure 6.1 shows Ensemble NPE can be overconfident as it was shown in the previous chapter while the two other methods are conservative or well-calibrated. However, the information contained in the posteriors quantified by the Expected Log Posterior (ELP) is better for Ensemble NPE and BNN-NPE$_\mathrm{T}$. Therefore, BNN-NPE$_\mathrm{T}$ gives conservative and informative posteriors while the two other methods suffer from overconfident or uninformative posteriors.

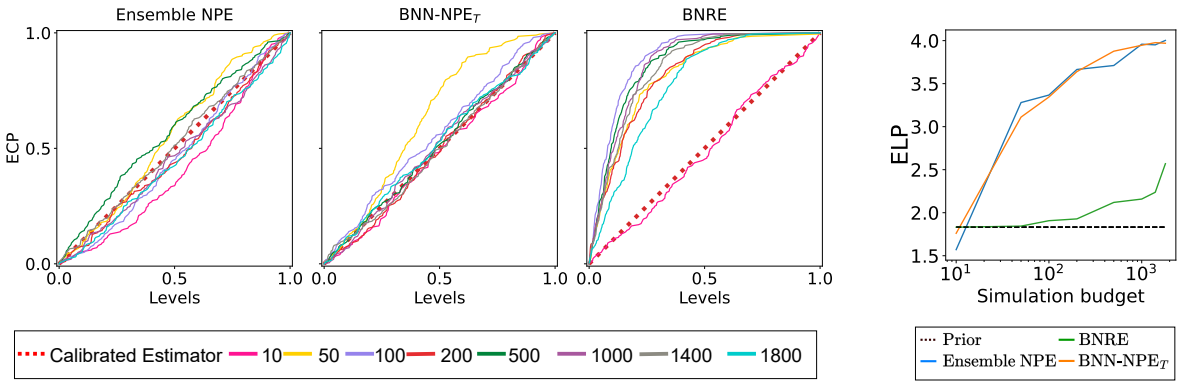


Figure 6.1: Expected Coverage Probability (ECP) and ELP comparisons for Ensemble NPE, BNRE and BNN-NPE$_\mathrm{T}$ on the cosmological application. BNRE and BNN-NPE$_\mathrm{T}$ remain conservative in all cases while Ensemble NPE can be overconfident. However, BNN-NPE$_\mathrm{T}$ and Ensemble NPE have a similar ELP indicating posteriors as informative in both cases. Therefore, BNN-NPE$_\mathrm{T}$ has the best trade-off.

In Figure 6.2, examples of posteriors depending on the number of simulations available for training is shown. Similar conclusions can be drawn, although they are qualitative in

this case. Only BNN-NPE$_\mathrm{T}$ and Ensemble NPE provide informative posteriors.

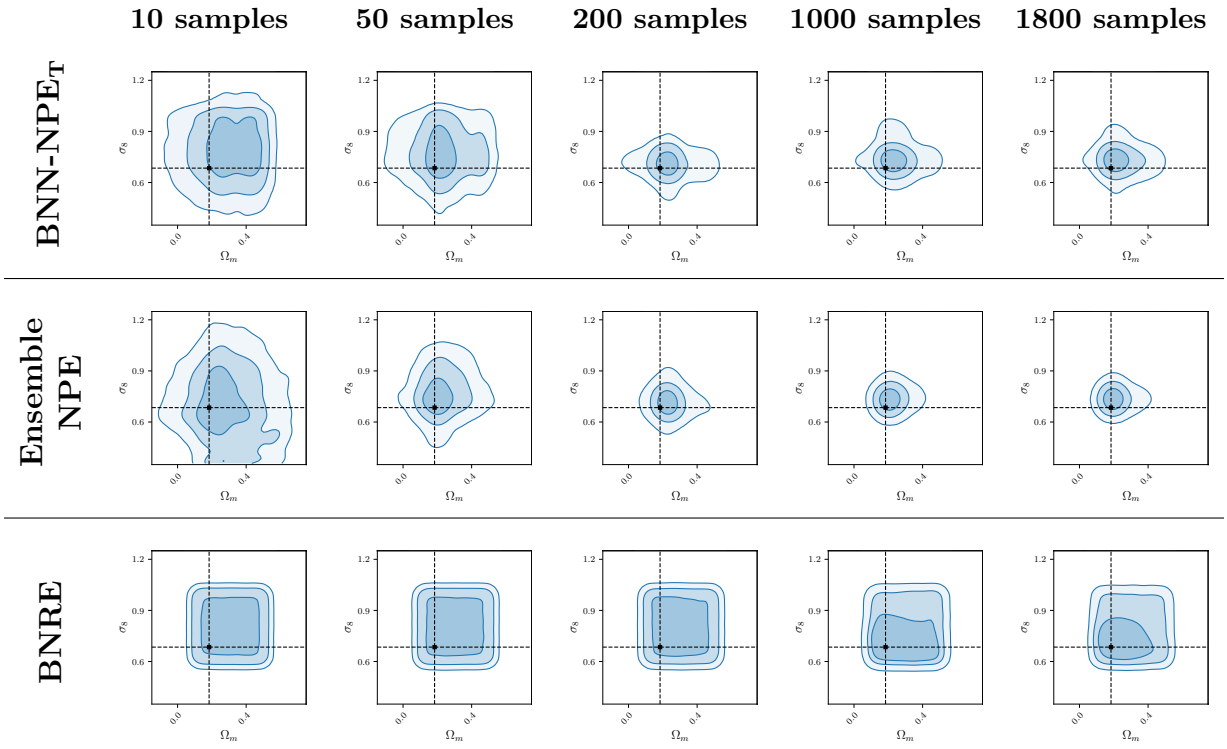


Figure 6.2: Comparison of posteriors for Ensemble NPE, BNRE and BNN-NPE$_\mathrm{T}$ on the cosmological application depending on the number of simulations available. BNRE gives uninformative posteriors while the two other methods provide informative posteriors.

7. Discussion

Conclusion

This thesis has demonstrated the effectiveness of using computational uncertainty in Simulation-Based Inference (SBI) using Bayesian Deep Learning (BDL). BDL in SBI achieves conservative results. This is particularly useful in computationally intensive fields such as cosmology and meteorology where there are only a limited number of simulations available. Our evaluation criteria included Expected Coverage Probability (ECP), which assesses the conservativeness of posterior distributions, and the Expected Log Posterior (ELP) as the amount of information contained in the posterior. We also analysed the uncertainty in the network, considering both the uncertainty from the parameters $\boldsymbol{\theta}$ and the uncertainty from the approximation itself. This analysis helps us better understand the posteriors derived from the available simulations.

The first observation is that Bayesian methods (BNN-NPE and BNN-NRE) and Balanced Neural Ratio Estimation (BNRE) are the only methods that are always providing a conservative posterior distribution. Therefore, in fields where there is a limited number of data available, Bayesian methods can be safely used to draw scientific conclusions. All other conservative methods like Balanced Neural Posterior Estimation (BNPE) or Ensemble methods can fail to be conservative in data-poor regimes (Section 5.2.1).

Furthermore, Bayesian methods are found to learn more informative posterior distributions $p(\boldsymbol{\theta}|\mathbf{x})$ than BNRE for the same amount of simulations available i.e., obtaining higher ELP values (Sections 5.2.1 and 5.2.3).

Finally, Bayesian methods need more data to achieve posteriors as informative as those of classical methods such as Neural Posterior Estimation (NPE). However, the use of temperature hyperparameter allows Bayesian methods to focus more on the likelihood

of the data and obtain posteriors that are at least as informative as the best methods (Section 5.3.1).

Limitations and Future Work

Despite theoretical motivations for using BDL to obtain conservative SBI, there is no guarantee of obtaining conservative posteriors. Nevertheless, our experimental results confirm that our methods remain conservative under all tested conditions.

Our results also revealed a predominant aleatoric uncertainty in the posterior distributions $p(\boldsymbol{\theta}|\mathbf{x})$. This results in approximate posteriors that are not always informative because they are too close to the prior distribution. To advance Bayesian methods beyond classical methods in data-poor regimes, we propose several future research directions:

1. **Active Learning**: Given the ability of the network to identify observations with higher epistemic uncertainty, the use of active learning could improve the efficiency of learning in computational intensive applications (Haut et al., 2018).
2. **Advanced Prior Modelling for Bayesian Neural Network (BNN)**: Using more complex prior distributions for BNN, such as low-rank multivariate normal distributions (Louizos and Welling, 2016) or hierarchical models (MacKay, 1992), could enable the modelling of more complex posteriors with less data. This would restrict the network less and improve learning outcomes in data-poor regimes.
3. **Enhanced Posterior Approximation Techniques**: Almost all of our experiments have been performed using mean-field Variational Inference (VI), one of the earliest and simplest methods for BDL. Using other methods for the posterior distribution $p(\mathbf{w}|D)$ on weights could increase the capacity. For instance, Stochastic Gradient Hamiltonian Monte Carlo (SGHMC) has sometimes been used, and it has improved the results at the cost of computational increases.

Finally, it is important to keep in mind that all methods developed are conservative in expectation. There is no guarantee of being conservative on a single observation. Section 5.3.2 showed that Bayesian Neural Posterior Estimation (BNN-NPE) is conservative in expectation but may be overconfident for some observations $\mathbf{x}$. To the best of our knowledge, there is no current method that can give theoretical guarantees for a single

observation $\mathbf{x_0}$.

In conclusion, this thesis underscores the conservativeness of BDL in posterior estimation in data-poor regimes. In fields such as cosmology where there are only a limited number of simulations available, our method is conservative and provides informative posteriors. Our method showed evidence of conservative and informative posteriors on a cosmological application modelling the expansion of the Universe (Chapter 6).

Bibliography

Alsing, J., Charnock, T., Feeney, S., and Wandelt, B. (2019). Fast likelihood-free cosmology with neural density estimators and active learning. *Monthly Notices of the Royal Astronomical Society*, 488(3):4440–4458.

Beaumont, M. A., Zhang, W., and Balding, D. J. (2002). Approximate bayesian computation in population genetics. *Genetics*, 162(4):2025–2035.

Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. (2017). Variational inference: A review for statisticians. *Journal of the American statistical Association*, 112(518):859–877.

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.

Clemencic, M., Corti, G., Easo, S., Jones, C., Miglioranzi, S., Pappagallo, M., Robbe, P., on behalf ofthe LHCb Collaboration, et al. (2011). The lhcb simulation application, gauss: Design, evolution and experience. In *Journal of Physics: Conference Series*, volume 331, page 032023. IOP Publishing.

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.

Cranmer, K., Pavez, J., and Louppe, G. (2015). Approximating likelihood ratios with calibrated discriminative classifiers. *arXiv preprint arXiv:1506.02169*.

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. *arXiv preprint arXiv:2304.10978*.

Dinh, L., Krueger, D., and Bengio, Y. (2014). Nice: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516*.

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

Durkan, C., Papamakarios, G., and Murray, I. (2018). Sequential neural methods for likelihood-free inference. *arXiv preprint arXiv:1811.08723*.

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.

Feng, R., Grana, D., and Balling, N. (2021). Variational inference in bayesian neural network for well-log prediction. *Geophysics*, 86(3):M91–M99.

Fortuin, V. (2021). *On the Choice of Priors in Bayesian Deep Learning*. PhD thesis, ETH Zurich.

Gal, Y. and Ghahramani, Z. (2016). Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR.

Glorot, X. and Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256. JMLR Workshop and Conference Proceedings.

Graves, A. (2011). Practical variational inference for neural networks. *Advances in neural information processing systems*, 24.

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.

Haut, J. M., Paoletti, M. E., Plaza, J., Li, J., and Plaza, A. (2018). Active learning with convolutional neural networks for hyperspectral image classification using a new bayesian approach. *IEEE Transactions on Geoscience and Remote Sensing*, 56(11):6440–6461.

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

Hermans, J., Begy, V., and Louppe, G. (2020b). 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., and Louppe, G. (2021). Averting a crisis in simulation-based inference. arxiv e-prints, art. *arXiv preprint arXiv:2110.06581*.

Hoffmann, L. and Elster, C. (2021). Deep ensembles from a bayesian perspective. *arXiv preprint arXiv:2105.13283*.

Karras, C., Karras, A., Avlonitis, M., and Sioutas, S. (2022). An overview of mcmc methods: from theory to applications. In *IFIP International Conference on Artificial Intelligence Applications and Innovations*, pages 319–332. Springer.

Kendall, A. and Gal, Y. (2017). What uncertainties do we need in bayesian deep learning for computer vision? *Advances in neural information processing systems*, 30.

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

Linhart, J., Gramfort, A., and Rodrigues, P. (2024). L-c2st: Local diagnostics for posterior approximations in simulation-based inference. *Advances in Neural Information Processing Systems*, 36.

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

Lotka, A. J. (1927). Fluctuations in the abundance of a species considered mathematically. *Nature*, 119(2983):12–12.

Louizos, C. and Welling, M. (2016). Structured and efficient variational deep learning with matrix gaussian posteriors. In *International conference on machine learning*, pages 1708–1716. PMLR.

Lueckmann, J.-M., Bassetto, G., Karaletsos, T., and Macke, J. H. (2019). Likelihood-free inference with emulator networks. In *Symposium on Advances in Approximate Bayesian Inference*, pages 32–53. PMLR.

MacKay, D. J. (1992). A practical bayesian framework for backpropagation networks. *Neural computation*, 4(3):448–472.

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.

Neal, R. M. et al. (2011). Mcmc using hamiltonian dynamics. *Handbook of markov chain monte carlo*, 2(11):2.

Noci, L., Roth, K., Bachmann, G., Nowozin, S., and Hofmann, T. (2021). Disentangling the roles of curation, data-augmentation and the prior in the cold posterior effect. *Advances in neural information processing systems*, 34:12738–12748.

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

Papamakarios, G., Pavlakou, T., and Murray, I. (2017). Masked autoregressive flow for density estimation. *Advances in neural information processing systems*, 30.

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

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

Papamakarios, G., Sterratt, D., and Murray, I. (2019c). 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*.

Rezende, D. and Mohamed, S. (2015). Variational inference with normalizing flows. In *International conference on machine learning*, pages 1530–1538. PMLR.

Rubin, D. B. (1984). Bayesianly justifiable and relevant frequency calculations for the applied statistician. *The Annals of Statistics*, pages 1151–1172.

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*.

Shaker, M. H. and Hüllermeier, E. (2020). Aleatoric and epistemic uncertainty with random forests. In *Advances in Intelligent Data Analysis XVIII: 18th International Symposium on Intelligent Data Analysis, IDA 2020, Konstanz, Germany, April 27–29, 2020, Proceedings 18*, pages 444–456. Springer.

Silverman, B. W. (2018). *Density estimation for statistics and data analysis*. Routledge.

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.

Wald, A. (1943). Tests of statistical hypotheses concerning several parameters when the number of observations is large. *Transactions of the American Mathematical society*, 54(3):426–482.

Wenzel, F., Roth, K., Veeling, B. S., 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? *arXiv preprint arXiv:2002.02405*.

Ying, X. (2019). An overview of overfitting and its solutions. In *Journal of physics: Conference series*, volume 1168, page 022022. IOP Publishing.

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

A. Appendix

A.1. Mathematical Descriptions

A.1.1. Numerical Coverage Evaluation

The Expected Coverage Probability (ECP) of a level α is

$$\mathrm{E}_{p(\boldsymbol{\theta},\mathbf{x})}[\mathbf{1}[\boldsymbol{\theta} \in \Theta_{\hat{p}(\boldsymbol{\theta}|\mathbf{x})}(1-\alpha)]]. \tag{A.1}$$

$\Theta_{\hat{p}(\boldsymbol{\theta}|\mathbf{x})}(1-\alpha)$ denotes the $1-\alpha$ highest posterior density region of the approximate posterior $\hat{p}(\boldsymbol{\theta}|\mathbf{x})$.

Using a test set of n pairs $(\boldsymbol{\theta}_i, \mathbf{x}_i) \sim p(\boldsymbol{\theta}, \mathbf{x})$, Equation A.1 can be numerically approximated with

$$\frac{1}{n}\sum_{i=1}^{n} \mathbf{1}[\boldsymbol{\theta_i} \in \Theta_{\hat{p}(\boldsymbol{\theta}|\mathbf{x}_i)}(1-\alpha)].$$

$[\mathbf{1}[\boldsymbol{\theta_i} \in \Theta_{\hat{p}(\boldsymbol{\theta}|\mathbf{x}_i)}(1-\alpha)]]$ can be numerically evaluated in two ways: by using a grid over the domain of the prior distribution $p(\boldsymbol{\theta})$, as done in Neural Ratio Estimation (NRE), or by sampling from the approximate posterior $\hat{p}(\boldsymbol{\theta}|\mathbf{x})$, as done in Neural Posterior Estimation (NPE).

A.1.2. Quantiles Coverage

The ECP of Equation A.1 can be rewritten as

$$\mathrm{E}_{p(\boldsymbol{\theta},\mathbf{x})}[\mathbf{1}[\boldsymbol{\theta} \in \Theta_{\hat{p}(\boldsymbol{\theta}|\mathbf{x})}(1-\alpha)]] = \mathrm{E}_{p(\mathbf{x})}\mathrm{E}_{p(\boldsymbol{\theta}|\mathbf{x})}[\mathbf{1}[\boldsymbol{\theta} \in \Theta_{\hat{p}(\boldsymbol{\theta}|\mathbf{x})}(1-\alpha)]]. \tag{A.2}$$

For a specific $\mathbf{x_0}$, the coverage probability becomes

$$\mathrm{E}_{p(\boldsymbol{\theta}|\mathbf{x_0})}[\mathbf{1}[\boldsymbol{\theta} \in \Theta_{\hat{p}(\boldsymbol{\theta}|\mathbf{x_0})}(1-\alpha)]]. \tag{A.3}$$

This formula should be equal to $1-\alpha$ for the true posterior. Evaluating this expression is very difficult because the true posterior distribution $p(\boldsymbol{\theta}|\mathbf{x_0})$ is unknown. Only one sample $\boldsymbol{\theta}$ is in the test set, the one used from the simulation. There exist techniques to evaluate this expression (Zhao et al., 2021) but this is out of the scope of this thesis.

As Equation A.3 should be equal to $1-\alpha$ for all observations $\mathbf{x_0}$ for the true posterior, any distribution on the observations $q(\mathbf{x})$ within the support of $p(\mathbf{x})$ should have an expected coverage of $1-\alpha$. Therefore, for the true posterior distribution, one has

$$\mathrm{E}_{q(\mathbf{x})}\mathrm{E}_{p(\boldsymbol{\theta}|\mathbf{x})}[\mathbf{1}[\boldsymbol{\theta} \in \Theta_{\hat{p}(\boldsymbol{\theta}|\mathbf{x})}(1-\alpha)]] = 1-\alpha, \tag{A.4}$$

if $\hat{p}(\boldsymbol{\theta}|\mathbf{x}) = p(\boldsymbol{\theta}|\mathbf{x})$. Therefore, if we choose $q(\mathbf{x})$ as some quantiles of $p(\mathbf{x})$, Equation A.4 should hold for the true posterior distribution.

However, although Equation A.2 is equal to $1-\alpha$ if $\hat{p}(\boldsymbol{\theta}|\mathbf{x}) = p(\boldsymbol{\theta})$, this is not necessarily the case for the local coverage in Equation A.3. Therefore, Equation A.4 does not necessarily hold if $\hat{p}(\boldsymbol{\theta}|\mathbf{x}) = p(\boldsymbol{\theta})$. As a consequence, even if a method modelling $\hat{p}(\boldsymbol{\theta}|\mathbf{x}) = p(\boldsymbol{\theta})$ is conservative in expectation, there is no guarantee it will be conservative for all observations $\mathbf{x}$.

A.1.3. Uncertainty Decomposition

Literature distinguishes between two primary measures of uncertainty: variance (Kendall and Gal, 2017) and entropy (Shaker and Hüllermeier, 2020). Variance is straightforward

and computationally efficient for single-mode distributions but falls short in capturing the complexities of multimodal distributions. In such scenarios, entropy, specifically Shannon entropy, emerges as a superior uncertainty measure due to its ability to account for multiple modes while effectively measuring the reduction in uncertainty. Therefore, the **Shannon entropy** has been used as an uncertainty measure.

The uncertainty in Bayesian Neural Networks (BNNs) can be decomposed into the total uncertainty, the uncertainty of the prediction, the aleatoric uncertainty, the uncertainty linked to the data and the epistemic uncertainty, the uncertainty linked to the model.

The uncertainty types are quantified as follows:

1. **Total Uncertainty (TU)**: Quantified by the entropy of the Bayesian Model Averaging (BMA) distribution

$$TU(\boldsymbol{\theta}|\mathbf{x}, D) = H[\hat{p}(\boldsymbol{\theta}|\mathbf{x})] = H[\mathrm{E}_{\hat{p}(\mathbf{w}|D)}\hat{p}(\boldsymbol{\theta}|\mathbf{x}, \mathbf{w})].$$

   This is the total uncertainty of the approximate posterior $\hat{p}(\boldsymbol{\theta}|\mathbf{x})$. The higher it is, the more uncertain about the prediction the model is.

2. **Aleatoric Uncertainty (AU)**: The expected entropy across individual model distributions

$$AU(\boldsymbol{\theta}|\mathbf{x}, D) = \mathrm{E}_{\hat{p}(\mathbf{w}|D)}H[\hat{p}(\boldsymbol{\theta}|\mathbf{x}, \mathbf{w})].$$

   This is the uncertainty linked to the data. It is always smaller than the total uncertainty. The higher it is, the more uncertain about the data we are. If it is very high, it means the model is unsure about the distribution and it has not learned a lot.

3. **Epistemic Uncertainty (EU)**: The difference between total and aleatoric uncer-

tainties

$$
\begin{aligned}
EU(\boldsymbol{\theta}|\mathbf{x}, D) &= TU(\mathbf{x}|D) - AU(\mathbf{x}|D) \\
&= H[\hat{p}(\boldsymbol{\theta}|\mathbf{x})] - \mathrm{E}_{\hat{p}(\mathbf{w}|D)} H[\hat{p}(\boldsymbol{\theta}|\mathbf{x}, \mathbf{w})] \\
&= -\mathrm{E}_{\hat{p}(\boldsymbol{\theta}|\mathbf{x})} \log_2[\hat{p}(\boldsymbol{\theta}|\mathbf{x})] + \mathrm{E}_{\hat{p}(\mathbf{w}|D)} \mathrm{E}_{\hat{p}(\boldsymbol{\theta}|\mathbf{x},\mathbf{w})} \log_2[\hat{p}(\boldsymbol{\theta}|\mathbf{x}, \mathbf{w})] \\
&= -\mathrm{E}_{\hat{p}(\boldsymbol{\theta}|\mathbf{x})} \log_2[\hat{p}(\boldsymbol{\theta}|\mathbf{x})] + \mathrm{E}_{\hat{p}(\mathbf{w},\boldsymbol{\theta}|\mathbf{x},D)} \log_2[\hat{p}(\boldsymbol{\theta}|\mathbf{x}, \mathbf{w})] \\
&= -\mathrm{E}_{\hat{p}(\mathbf{w},\boldsymbol{\theta}|\mathbf{x},D)} \log_2[\hat{p}(\boldsymbol{\theta}|\mathbf{x})] + \mathrm{E}_{\hat{p}(\mathbf{w},\boldsymbol{\theta}|\mathbf{x},D)} \log_2[\hat{p}(\boldsymbol{\theta}|\mathbf{x}, \mathbf{w})] \\
&= \mathrm{E}_{\hat{p}(\mathbf{w},\boldsymbol{\theta}|\mathbf{x},D)} \log_2 \left(\frac{\hat{p}(\boldsymbol{\theta}|\mathbf{x}, \mathbf{w})}{\hat{p}(\boldsymbol{\theta}|\mathbf{x})} \right) \\
&= \mathrm{E}_{\hat{p}(\mathbf{w},\boldsymbol{\theta}|\mathbf{x},D)} \log_2 \left(\frac{\hat{p}(\boldsymbol{\theta}|\mathbf{x}, \mathbf{w}) p(\mathbf{w}|D)}{\hat{p}(\boldsymbol{\theta}|\mathbf{x}) p(\mathbf{w}|D)} \right) \\
&= \mathrm{E}_{\hat{p}(\mathbf{w},\boldsymbol{\theta}|\mathbf{x},D)} \log_2 \left(\frac{\hat{p}(\mathbf{w}, \boldsymbol{\theta}|\mathbf{x}, D)}{\hat{p}(\boldsymbol{\theta}|\mathbf{x}) p(\mathbf{w}|D)} \right) \\
&= KL(\hat{p}(\mathbf{w}, \boldsymbol{\theta}|\mathbf{x}, D), \hat{p}(\boldsymbol{\theta}|\mathbf{x}) p(\mathbf{w}|D)) \\
&= I(\boldsymbol{\theta}|\mathbf{x}, \mathbf{w}|D),
\end{aligned}
$$

which is the mutual information between the parameters $\boldsymbol{\theta}$ and the weights $\mathbf{w}$.

A.2. Simulation-Based Inference Architectures

A.2.1. Architectures Description

Neural Posterior Estimation

For NPE, one needs to specify the conditional distribution $q_\phi(\boldsymbol{\theta}|\mathbf{x})$. A normalizing flow was chosen. More specifically, a neural spline flow was chosen as the family (Durkan et al., 2019).

Bayesian Neural Posterior Estimation

The same architecture as for traditional NPE was chosen which is Neural Spline Flow (Durkan et al., 2019). For mean-field Variational Inference (VI), the mean and standard deviation are the parameters.

For the standard deviation, to ensure it was positive, we predict the softplus value of the standard deviation.

Mean-field VI was used on all benchmarks. Stochastic Gradient Hamiltonian Monte Carlo (SGHMC) was used to train the tuned prior described in our scientific paper (Appendix B).

For mean-field VI, the variational family chosen is independent normal distributions for all weights.

When the tuned prior over the weights described in Section 4.3.2 is used, the initialization is set to the prior distribution.

When the prior is isotropic normal distributions with a zero-mean and the same standard deviation for all weights, the initialization slightly varies. For the VI method, the initial standard deviation is set to the prior standard deviation for all weights and the means are set using the Xavier initialization (Glorot and Bengio, 2010). For the SGHMC method, the initial guess is set to a neural network with a Xavier initialization.

100 networks were used to compute the Bayesian Model Average (BMA) distribution as using more networks did not bring any improvements. Indeed, all the performance metrics do not improve when using more than 100 networks (Figure A.1).

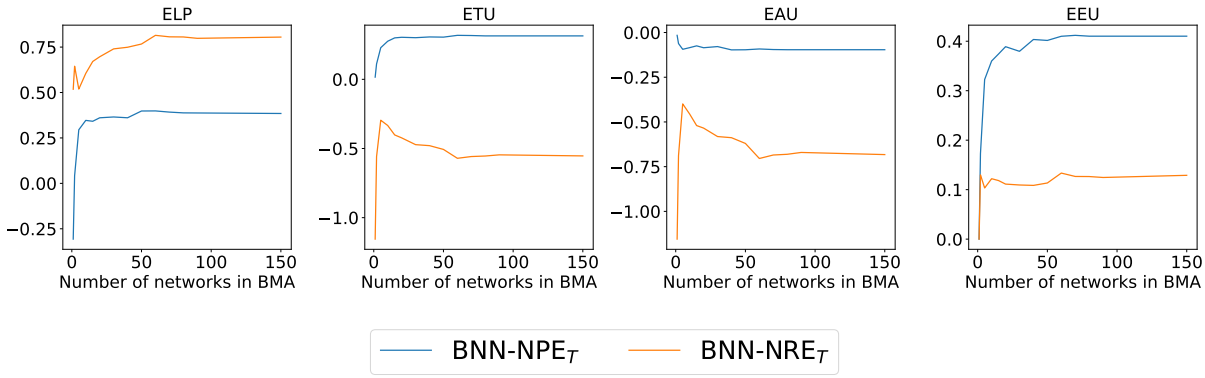


Figure A.1: Evolution of performance quantities on the BMA distribution for Bayesian Neural Posterior Estimation (BNN-NPE) with the number of networks used in the BMA on the Two Moons benchmark with 16 000 simulations available. The expected log posterior on the test set demonstrates an increase followed by a saturation point around 50 networks, a trend mirrored by the estimation of epistemic uncertainty.

Neural Ratio Estimation

NRE consists of a classificator. A simple multi-layer perceptron was chosen to classify it.

Bayesian Neural Ratio Estimation

The same architecture as for traditional NRE was chosen which is a multi-layer perceptron.

For mean-field VI, the mean and standard deviation are the parameters. For the standard deviation, to ensure it was positive, we predict the log value of the standard deviation.

Mean-field VI was used on all benchmarks. SGHMC was used to train the tuned prior described in our scientific paper (Appendix B).

For mean-field VI, the variational family chosen is independent normal distributions for all weights.

When the tuned prior over the weights described in Section 4.3.2 is used, the initialization is set to the prior distribution.

When the prior is isotropic normal distributions with a zero-mean and the same standard deviation for all weights, the initialization slightly varies. For the VI method, the initial standard deviation is set to the prior standard deviation for all weights and the means are set using the Xavier initialization (Glorot and Bengio, 2010). For the SGHMC method, the initial guess is set to a neural network with a Xavier initialization.

100 networks were used to compute the BMA distribution as using more networks did not bring any improvements. Indeed, all the performance metrics do not improve when using more than 100 networks (Figure A.1).

A.3. Benchmarks Description

Five benchmarks were used to evaluate the results:

1. **SLCP** (Papamakarios et al., 2019b):
    - **Prior over parameters**: The prior used in this benchmark is a uniform prior between -3 and 3.

- **Motivation**: This benchmark was used because the posterior distribution is complex and multimodal. Therefore, it is very easy for a model to be overconfident and is a good test for conservativeness.
- **Description**: The SLCP simulator, which models a hypothetical scenario with five parameters, generates an observable x comprising eight scalars. These scalars represent the 2D coordinates of four points, with each point's coordinate sampled from the same multivariate Gaussian distribution. The mean and covariance matrix of this distribution are parameterized by θ. This model represents a modified version of the original task focusing on inferring the marginal posterior density of two parameters. Unlike the original formulation, the likelihood in this version is non-tractable due to the process of marginalisation.

2. **Two Moons** (Papamakarios and Murray, 2016):
   - **Prior over parameters**: The prior used in this benchmark is a uniform prior between -1 and 1.
   - **Motivation**: This benchmark was used because the posterior distribution is complex and multimodal. Therefore, it is very easy for a model to be overconfident and is a good test for conservativeness.
   - **Description**: The Two Moons simulator addresses a hypothetical problem involving two parameters. The observable x consists of two scalars that denote the 2D coordinates of a point randomly drawn from a crescent-shaped distribution. This distribution undergoes shifts and rotations around the origin, influenced by the values of the parameters. Notably, these transformations are dependent on the absolute sum of the parameters, resulting in a second crescent shape in the posterior distribution and thus introducing multimodality to the model.
3. **Lotka Volterra** (Lotka, 1920, 1927):
   - **Prior over parameters**: The prior used in this benchmark is a uniform prior between -4 and 1.
   - **Motivation**: This benchmark was used because the input is a long complex time-series which leads to more training difficulties.

- **Description**: The Lotka-Volterra population model examines the dynamic interactions between predator and prey species. This model is governed by four parameters θ, which impact the reproduction and mortality rates of both species. Analysis involves deducing the marginal posterior of the parameters of the predator using a time series data of 2001 steps, which depicts the population evolution over time. The method employed is based on a Markov Jump Process, as detailed by (Papamakarios et al., 2019c).

4. **Spatial SIR** (Hermans et al., 2021):

   - **Prior over parameters**: The prior used in this benchmark is a uniform prior between 0 and 1.
   - **Motivation**: This benchmark was used because the input is a long complex time-series which leads to more training difficulties.
   - **Description**: The Spatial SIR model incorporates a grid-world consisting of susceptible, infected, and recovered individuals. The model accounts for the spatial progression of an infection influenced by initial conditions alongside infection and recovery rates, denoted by θ. The observable x represents a snapshot of the 50 by 50 grid after a predetermined period.

B. Scientific Paper Submitted to Neural Information Processing Systems 2024

This appendix provides the scientific paper submitted by Arnaud Delaunoy, Siddharth Mishra-Sharma, Gilles Louppe and myself. The day this thesis was published, the paper was still being reviewed.

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.

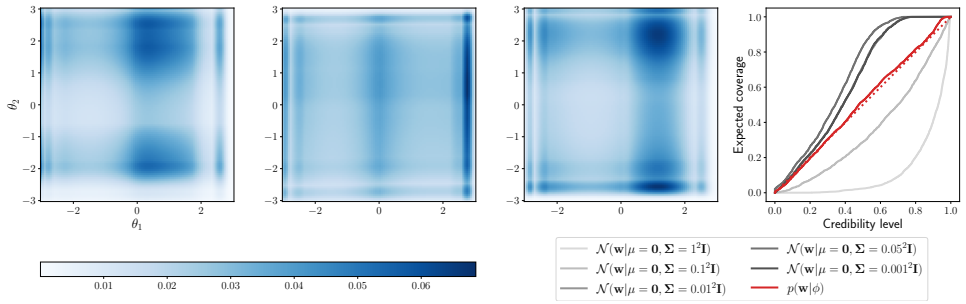


Figure 1: Visualization of the prior tuned to match the GP prior on the SLCP benchmark. Left: examples of posterior functions sampled from the tuned prior over neural network's weights. Right: expected coverage of the prior Bayesian model average with the tuned prior and normal priors for varying standard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

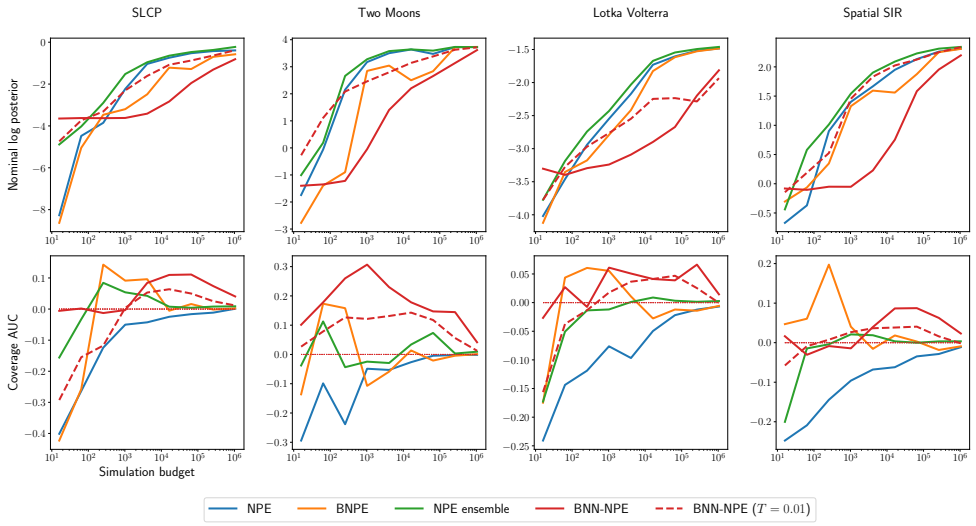


Figure 2: Comparison of different simulation-based inference methods through the nominal log probability and coverage area under the curve. The higher the nominal log probability, the more performant the method is. A calibrated posterior approximation exhibits a coverage AUC of 0. A positive coverage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

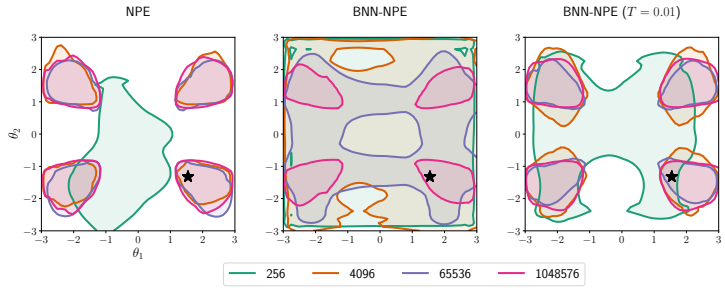


Figure 3: Examples of 95% highest posterior density regions obtained with various algorithms and simulation budgets on the SLCP benchmark for a single observation. The black star represents the ground truth used to generate the observation.

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.

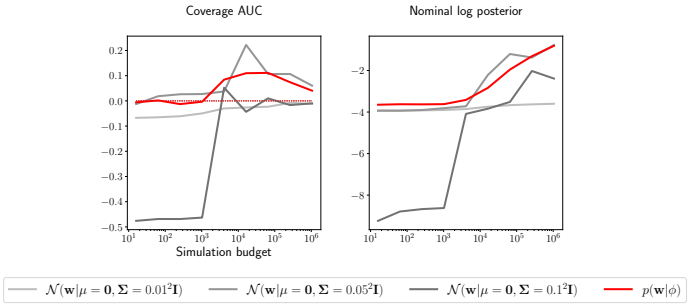


Figure 4: Comparison of posterior approximations obtained using a prior tuned to match the Gaussian process-based prior and using independent normal priors on weights with zero means and various standard deviations on the SLCP benchmark. 3 runs are performed, and the median is reported.

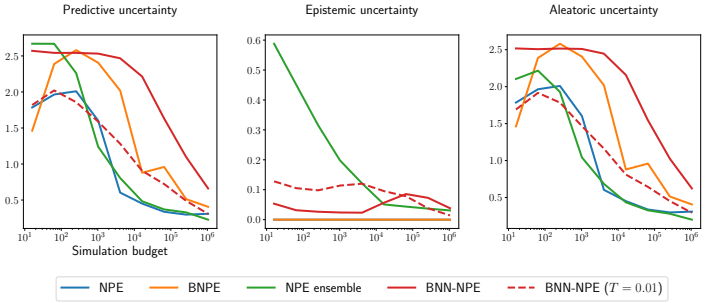


Figure 5: Quantification of the different forms of uncertainties captured by the different NPE-based methods on the SLCP benchmark. 3 runs are performed, and the median is reported.

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.

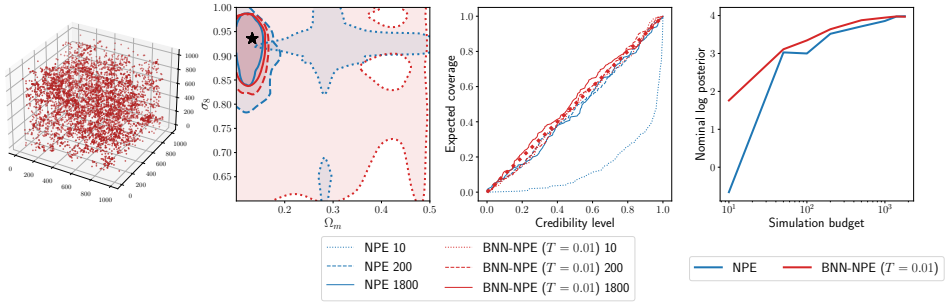


Figure 6: Comparison of the posterior approximations obtained with and without a Bayesian neural network on the cosmological application. First plot: An example observation: particles representing galaxies in a synthetic universe. Second plot: example of 95% highest posterior density regions for increasing 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 utilize 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 utilize 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.

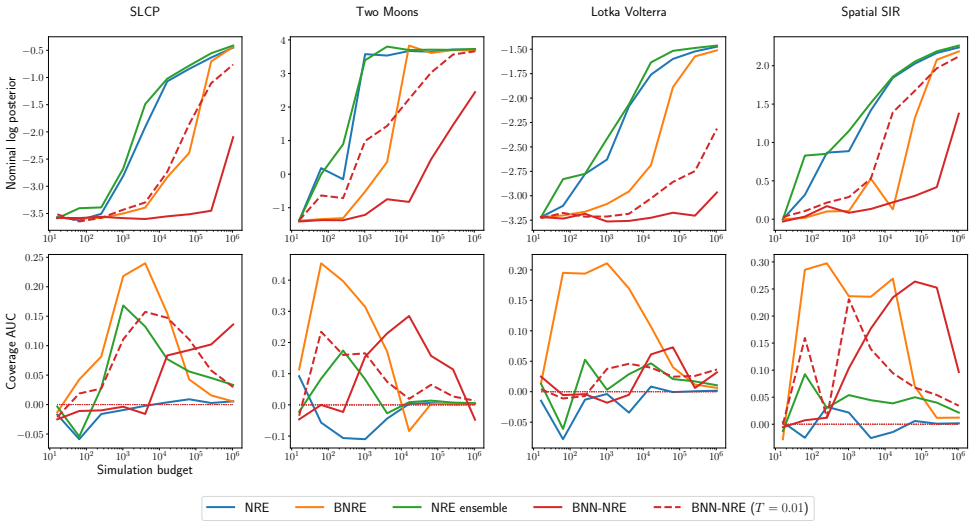


Figure 7: Comparison of different NRE simulation-based inference methods through the nominal log probability and coverage area under the curve. The higher the nominal log probability, the more performant the method is. A calibrated posterior approximation exhibits a coverage AUC of 0. A positive coverage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

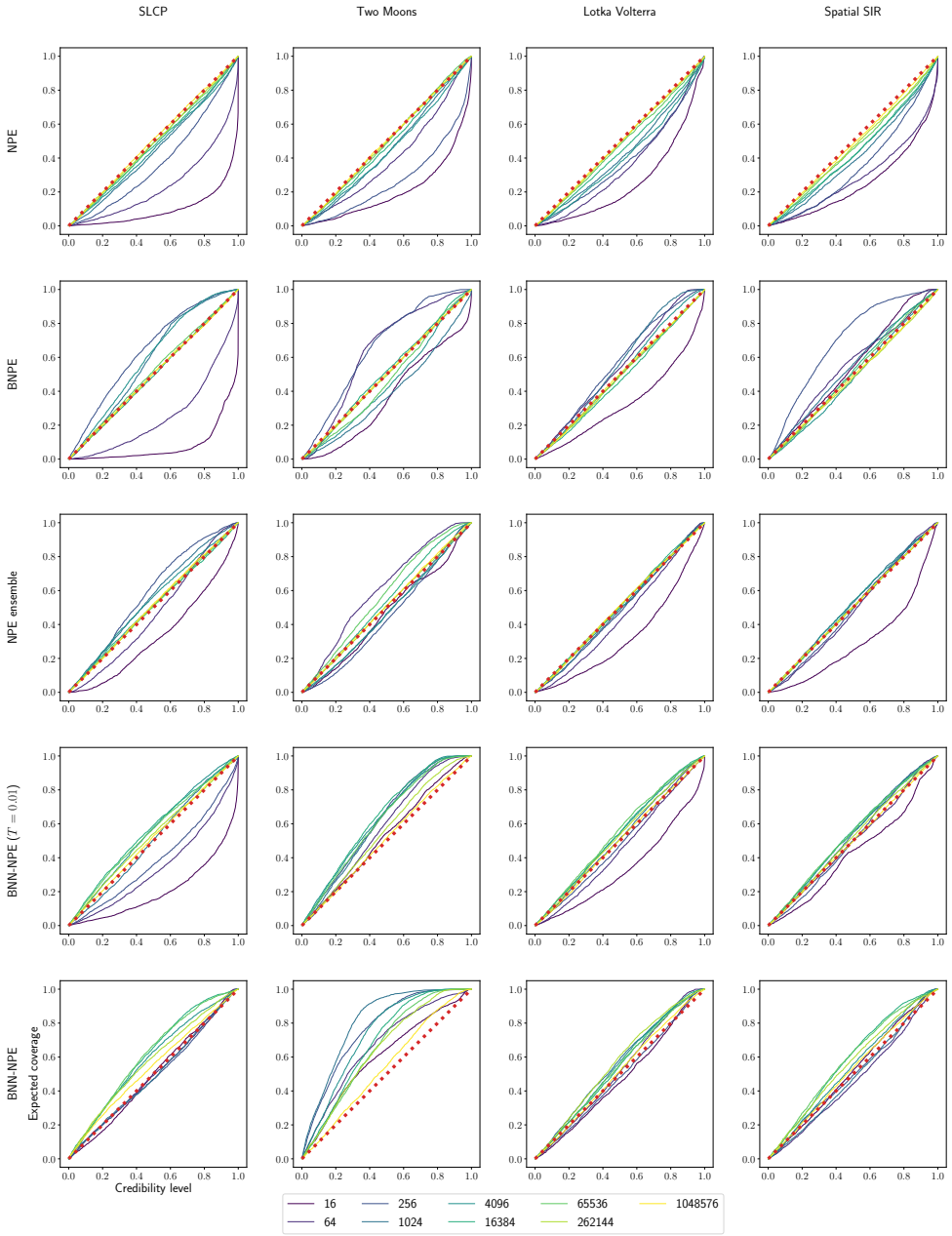


Figure 8: Coverage of different NPE simulation-based inference methods. A calibrated posterior approximation exhibits a coverage AUC of 0. A coverage curve above the diagonal indicates conservativeness and a curve below the diagonal indicates overconfidence. 3 runs are performed, and the median is reported.

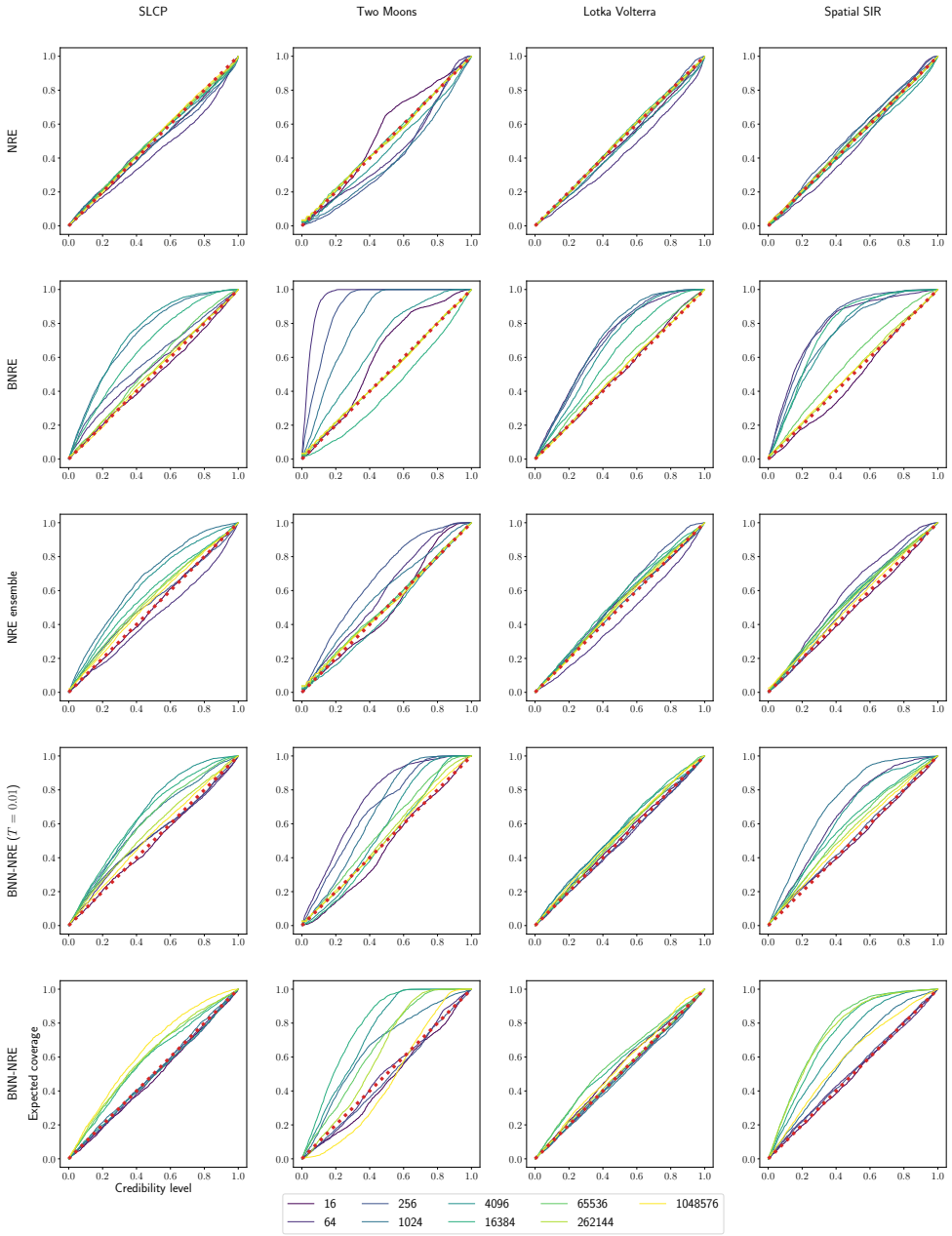


Figure 9: Coverage of different NRE simulation-based inference methods. A calibrated posterior approximation exhibits a coverage AUC of 0. A coverage curve above the diagonal indicates conservativeness and a curve below the diagonal indicates overconfidence. 3 runs are performed, and the median is reported.

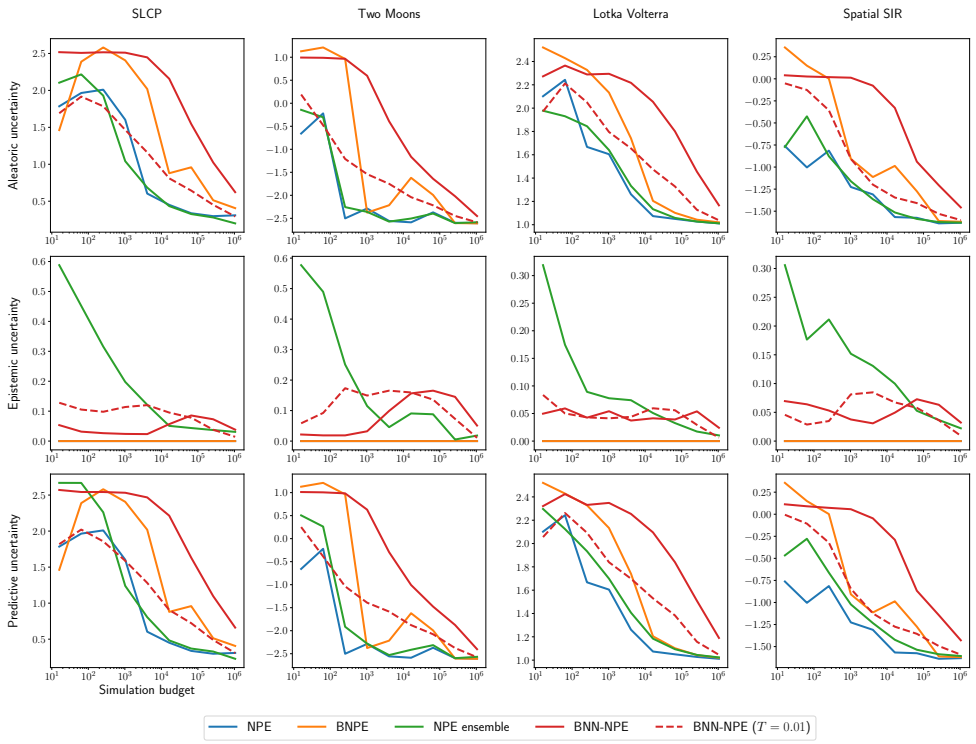


Figure 10: Quantification of the different forms of uncertainties captured by the different NPE-based methods. 3 runs are performed, and the median is reported.

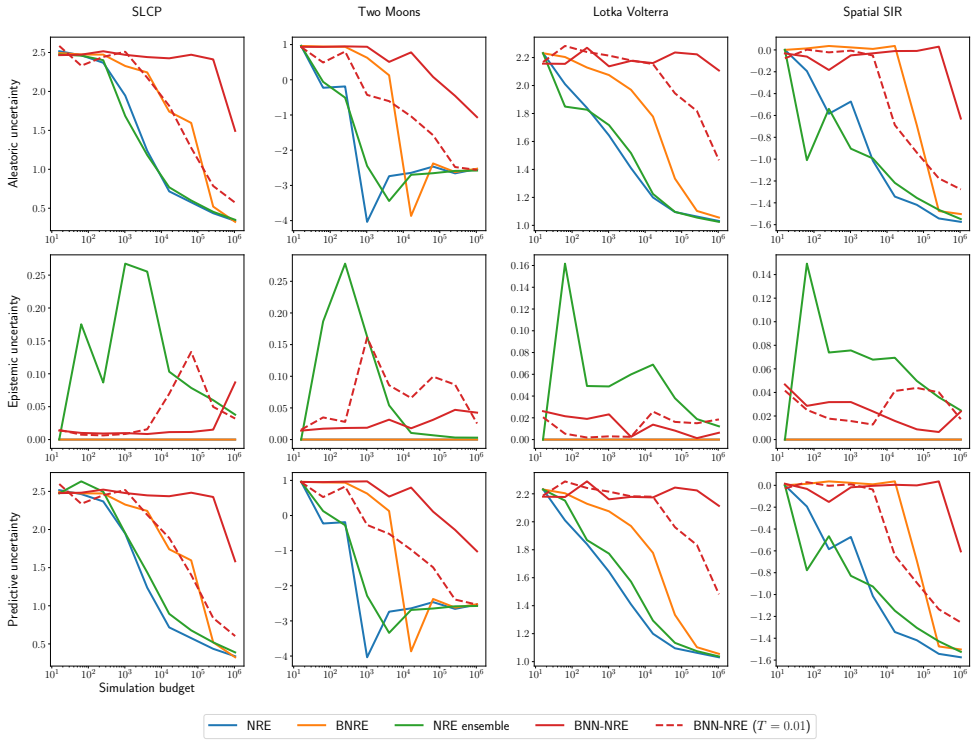


Figure 11: Quantification of the different forms of uncertainties captured by the different NRE-based methods. 3 runs are performed, and the median is reported.

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.