Towards Reliable Simulation-based Inference

by

Arnaud Delaunoy

Department of Electrical Engineering and Computer Science
Faculty of Applied Sciences

This dissertation has been submitted in partial fulfillment of the requirements for the Degree of Doctor of Applied Sciences.

Jury members

Prof. Gilles Louppe, University of Liège (Advisor)
Prof. Vân Anh Huynh-Thu, University of Liège (President)
Prof. Pierre Geurts, University of Liège
Prof. Anne-Françoise Donneau, University of Liège
Prof. Gaël Varoquaux, INRIA
Prof. Pedro L. C. Rodrigues, INRIA

LIÈGE université

December 2024

CONTENTS

ABSTRACT

Scientific knowledge expands by observing the world, hypothesizing some theories about it, and testing them against collected data. When those theories take the form of statistical models, statistical analyses are involved in the process of testing and refining scientific hypotheses. Those analyses can become challenging with the increased complexity of proposed theories. In this thesis, we focus on statistical models that take the form of scientific simulators and provide background about how machine learning can be used for statistical analyses in this context.

The first part of this thesis is about showing empirically that performing statistical analyses with machine learning involves a degree of approximation. Specifically, all statistical analyses involve a level of uncertainty in the conclusions drawn, and we show that approximations can lead to overconfident conclusions. We draw caution regarding such overconfident conclusions and introduce a criterion to diagnose overconfident approximations.

In the second part, we introduce balancing, a way to regularize machine learning models to reduce overconfidence and favor calibrated or underconfident approximations. Balancing is first introduced for neural ratio estimation algorithms and then extended to other algorithms. Intuition about why balancing leads to less overconfident solutions is provided, and it is shown empirically that balanced algorithms are often either close to calibrated or underconfident.

The third part shows that Bayesian neural networks can also be used to mitigate the overconfidence of approximations. Unlike balancing, no regularization is required, and this solution can then work with few training samples and, hence, computationally expensive simulators. To that end, a new Bayesian neural network prior tailored for simulation-based inference is developed, and empirical results show a reduction in overconfidence compared to similar solutions without Bayesian neural networks.

ACKNOWLEDGMENTS

PUBLICATIONS

The core part of this thesis is based on the following publications. Each publication comes with a publicly available implementation that can be found on their associated github repositories.

- Hermans, J.*, Delaunoy, A.*, Rozet, F., Wehenkel, A., & Louppe, G. (2022). A crisis in simulation-based inference? beware, your posterior approximations can be unfaithful. Transactions on Machine Learning Research.
  This paper is discussed in Chapter 4 and the implementation can be found at https://github.com/montefiore-institute/trust-crisis-in-simulation-based-inference

- Delaunoy, A.*, Hermans, J.*, Rozet, F., Wehenkel, A., & Louppe, G. (2022). Towards reliable simulation-based inference with balanced neural ratio estimation. Advances in Neural Information Processing Systems, 35, 20025-20037.
  This paper is discussed in Chapter 5 and the implementation can be found at https://github.com/montefiore-institute/balanced-nre

- Delaunoy, A.*, Miller, B. K.*, Forré, P., Weniger, C., & Louppe, G. (2023). Balancing Simulation-based Inference for Conservative Posteriors. In Fifth Symposium on Advances in Approximate Bayesian Inference.
  This paper is discussed in Chapter 6 and the implementation can be found at https://github.com/ADelau/balancing_sbi

- Delaunoy, A.*, Bonardeaux, M. D. L. B.*, Mishra-Sharma, S., & Louppe, G. (2024). Low-Budget Simulation-Based Inference with Bayesian Neural Networks. arXiv preprint arXiv:2408.15136.
  This paper is discussed in Chapter 7 and the implementation can be found at https://github.com/ADelau/low_budget_sbi_with_bnn

In addition, I authored the following publications during my thesis.

- Delaunoy, A., Wehenkel, A., Hinderer, T., Nissanke, S., Weniger, C., Williamson, A., & Louppe, G. (2020). Lightning-Fast Gravitational Wave Parameter Inference through Neural Amortization. In Machine Learning and the Physical Sciences, NeurIPS 2020 workshop.

- Delaunoy, A., & Louppe, G. (2021). SAE: Sequential Anchored Ensembles. In Bayesian Deep Learning, NeurIPS 2021 workshop.

I have also participated to the following publication.

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

1 INTRODUCTION

Scientific discoveries are driven by data, either collected through controlled experiments or by observing the world. Those scientific theories can take the form of a mathematical model for some phenomenon in the world. In such cases, the model then represents our current knowledge about that phenomenon. For example, Newton's laws capture some knowledge about motion dynamics and Maxwell's equation knowledge about electromagnetism. Scientific theories could also take other forms when not suited for mathematical modeling. In this thesis, we keep the focus on mathematical modeling. Box (1976) and Blei (2014) introduced the Box's loop, a scientific method for constructing models for phenomena from data. This method is represented in Figure 1.1. The loop contains three steps that are repeated to improve a model iteratively. The first step is to build a model. This involves some creativity and intuition to develop equations that could explain the observations made. The model might include parameters that are unknown. For example, if we consider motion under a gravitational force, it is often modeled as being proportional to object masses and the inverse of the squared distance between the objects. A multiplicative constant is then added to obtain the scale of this force. It would be hard to come up with a value for this multiplicative constant simply with intuition. The second step is then to infer hidden quantities of the model, such as this multiplicative constant, from data. Once a model has been built and its possible hidden quantities have been inferred, it can be used to make predictions about the world, and those predictions can be challenged against pieces of data. Discrepancies between predictions and data can be used to criticize the model, for example, by identifying some effects that are not taken into account by the model. This model criticism is then a feedback loop, allowing the building of more accurate models. By repeating this procedure, more and more accurate models can be built.

Box's loop provides a way to iteratively build a collection of models. Popper (1959) argue that a model cannot be verified. However, a model can be refuted empirically. This collection of models should then be pruned out by removing models that can be shown to not accurately reflect the world. This process is called falsification. Refuting models happens both on model structure and hidden quantities. If we take back the example of motion under gravitational force, one could refute that gravitational force is proportional to object masses and the inverse of the squared distance between the objects or refute some multiplicative constant values.

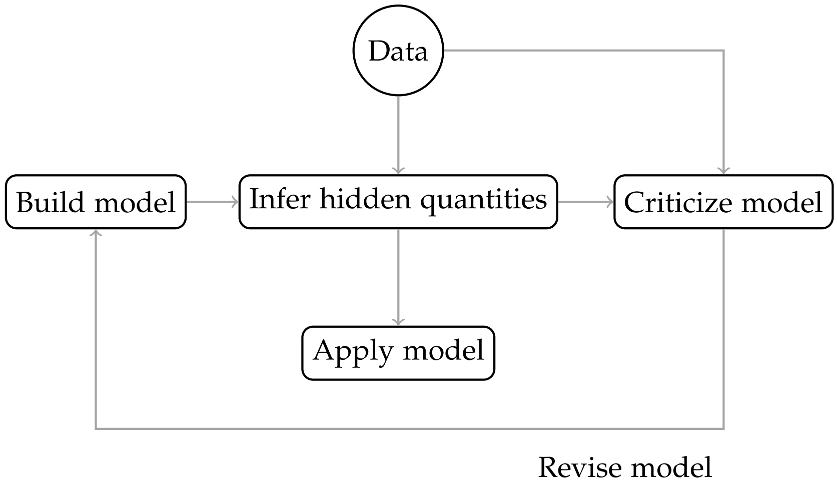

Figure 1.1: Visualization of Box's loop (Box, 1976; Blei, 2014). The loop iterates over three main elements: model building, model's hidden quantities inference, and model criticism. Iterating over those three elements aims to build better scientific models, taking into account model criticism feedback.

Historically, scientists kept models simple enough such that inferring hidden quantities and testing the model could be done analytically. Nowadays, with the increase in computational power and advances in algorithmics for numerical approximations, it is possible to deal with models that do not allow analytical inference and testing. In addition, as we keep refining models to capture all aspects, such complex models are often needed. While those advances allow the construction of more accurate models, hidden quantity inference and model testing are now often done approximately. This raises questions as to how it impacts the Popperian falsification procedure. Can a model or hidden quantities be reliably refuted by an approximate inference or testing procedure? It could be that our procedure leads to refutation only because of approximations and that if that same procedure was applied without approximations, refutation could not be obtained. Incorrect refutation of models could then ultimately prevent scientists from exploring valid models and lead them in the wrong direction.

In this thesis, we aim to adapt modern scientific reasoning to take into account the approximations made. At the root of this adapted methodology is the inclusion of the additional source of uncertainty brought by those approximations. Statistical methods are modified to produce inflated uncertainty estimates based on how the approximations are likely to affect the end result. In particular, we argue that underestimating the uncertainty is more harmful than overestimating it for Popperian falsification. Indeed, underestimation of uncertainty can lead to incorrect refutation, while overestimation can only lead to failing to refute a model. With this in mind, we design new statistical methods to try to avoid underestimation of the uncertainty and, hence, overconfidence. These methods can then be used more reliably for scientific analyses, and more trust can be given to conclusions drawn. We focus on models that take the form of a simulator. Simulators are becoming heavily used in science as they can model complex phenomena. Examples include agent-based models that can be found in the fields of epidemiology or economics. Simulators are also used in many fields of physical sciences, such as cosmology or particle physics, as well as in biology. Stochastic simulators are inherently complex to deal with as they involve many stochastic steps to model the full phenomena. If we consider agent-based models, they involve the stochastic actions of all those agents or the stochastic consequences of those actions. In addition, we focus on the problem of inferring hidden quantities of those models and on machine learning based methods for approximate inference.

The thesis is outlined as follows. In Chapter 2 and 3, we provide background in statistical modeling and simulation-based inference, respectively. In Chapter 4, we showcase how simulation-based inference methods can lead to overconfidence. In Chapter 5, we introduce balancing, the first improvement of simulation-based inference methods toward more reliability, and in Chapter 6, we extend balancing to a broader range of methods. We show in Chapter 7 that Bayesian neural networks can be used to improve the reliability of simulation-based inference methods, even in low-data regimes. We conclude and provide additional discussions in Chapter 8.

2

STATISTICAL MODELING

Statistical modeling is about specifying relationships between one or more random variables and non-random variables. Those relationships usually come from a combination of prior knowledge and inference from data. In this chapter, we describe different ways of reasoning about those relationships. We will also illustrate the concept of model using the example of a feather falling due to gravity.

2.1 PROBABILITY AND STATISTICAL MODELS

A probability model is a mathematical representation, that includes randomness, of a phenomenon. It is defined in the following way.

Definition 2.1.1. A probability model is defined by three components: a sample space Ω, a space of possible events $\mathcal{A}$ within the sample space, and a mapping P from an event $A \subset \Omega$ to a probability $P(A)$.

Hence, it encapsulates all the information about the probabilities of experiment outcomes regarding that random phenomenon. In all generality, statistics is concerned with inferring from an outcome of a probabilistic experiment interesting things about the data generating process. For example, one might be interested in extracting, from a family of models, the probability model that is the most consistent with the experimental outcome (point estimation), a range of models that are consistent with this outcome could also be extracted (interval estimation). Another interesting piece of information is whether or not a hypothesis can be refuted from data with some probability (hypothesis testing). To answer such questions, we start with a statistical model.

Definition 2.1.2. A statistical model is a mathematical model that embodies a set of statistical assumptions concerning the generation of sample data.

It often takes the form of a parametric family of probability models. In the remainder of this thesis, we will denote those parameters by $\boldsymbol{\theta}$. The statistical model with parameter values then jointly defines a probability model.

As an illustration, we will take the example of a feather falling. In this setting, we observe the time it takes for the feather to fall from a given height, and we want to model the fall.

This is illustrated in Figure 2.1. We assume that the fall follows Newtonian dynamics and that the gravitational force applied to the feather is

$$\boldsymbol{F} = m\; g\; \boldsymbol{e}, \tag{2.1}$$

where m is the mass of the feather, g is the gravity constant, and $\boldsymbol{e}$ is a unit vector pointing towards the earth. If no other forces are applied to the object, it takes $\sqrt{\frac{2h}{g}}$ seconds to fall from a height h.

We will model the measurement error as a normal distribution centered at 0:

$$\text{measurement error} \sim \mathcal{N}(0, \sigma_{\text{measurement}}). \tag{2.2}$$

If we combine the two, it gives the following statistical model for the measured fall time

$$\text{measured time} \sim \mathcal{N}\left(\sqrt{\frac{2h}{g}}, \sigma_{\text{measurement}}\right). \tag{2.3}$$

In this setting, if we assume that $\sigma_{\text{measurement}}$ and height h are known, there is only one parameter $\boldsymbol{\theta} = g$.

2.2 FREQUENTIST AND BAYESIAN STATISTICS

There are two main approaches to statistical inference: the frequentist and Bayesian paradigms. At the root of those two paradigms is the interpretation of probability. Frequentist statistics are based on the frequency interpretation of probability. The probability of an event is the observed frequency of that event in the limit of infinite observations. Bayesian statistics interpret probability as a degree of belief rather than observed frequencies.

The most direct consequence of those two philosophical approaches to probability is that, in the frequentist philosophy, we cannot assign a probability to the parameters $\boldsymbol{\theta}$ of the statistical model as those are not random events that can be observed. In the example of a feather falling, it is not possible to design a random experiment that would yield several values for g. It is an unknown parameter that is not random, and it would then make no sense to assign a probability to g in the frequentist philosophy. However, one can have a belief about what g could be, based on observations, and a probability can then be assigned to parameters in the Bayesian philosophy.

In the remainder of this thesis, we will denote the observations by $\boldsymbol{x}$ and by p the probability density function or probability mass function for continuous and discrete random variables, respectively. The data-generating process that models the fact of performing an experiment will be denoted as $p^*(\boldsymbol{x})$. Performing an experiment is, hence, equivalent to sampling an observation from this data-generating process $\boldsymbol{x} \sim p^*(\boldsymbol{x})$. A central component of statistical inference is the likelihood $p(\boldsymbol{x}|\boldsymbol{\theta})$. It defines the probability of observing $\boldsymbol{x}$ according to the statistical model with parameters $\boldsymbol{\theta}$. In our example, it defines the prob-

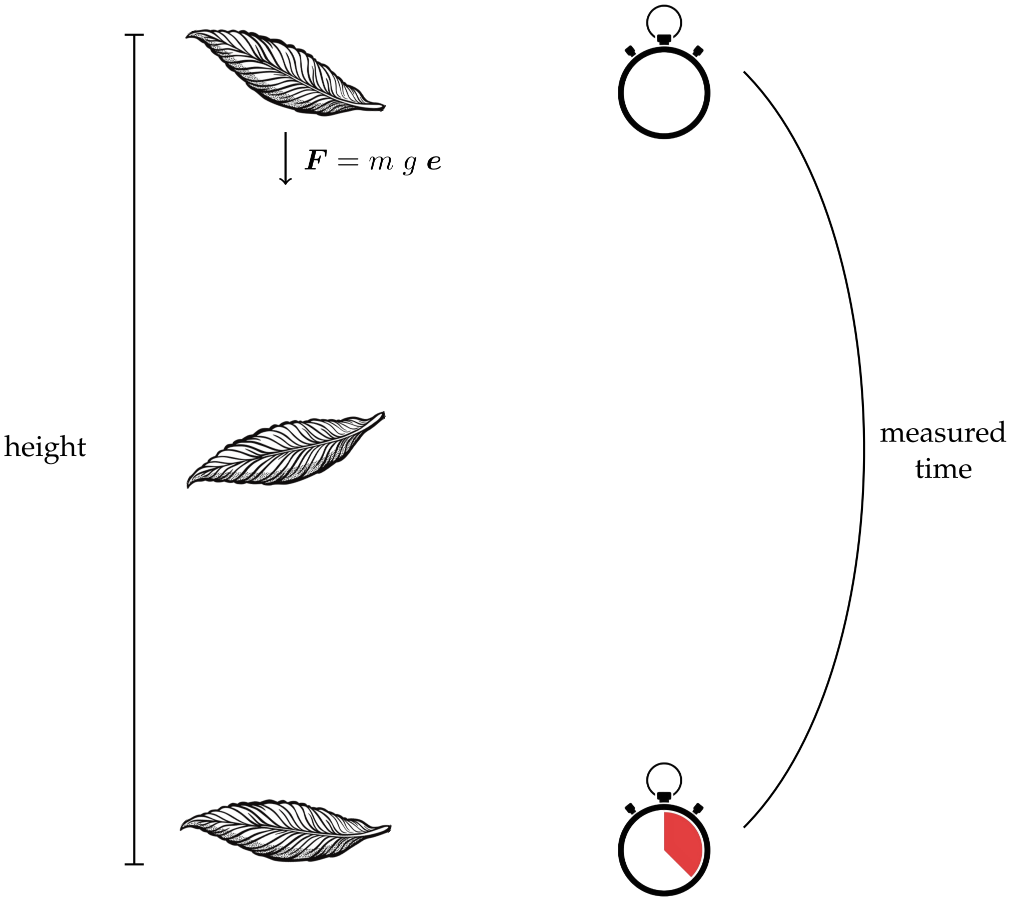

Figure 2.1: Illustration of the experiment of a feather falling. A gravitational force $\boldsymbol{F} = m\ g\ \boldsymbol{e}$ is applied on the feather and the fall time is measured.

ability of observing some fall time ($\boldsymbol{x}$) given a gravity constant g ($= \boldsymbol{\theta}$). The likelihood is used in both frequentist and Bayesian statistics.

In the Bayesian philosophy, we can go a step further and define a belief over parameters $\boldsymbol{\theta}$ given the observations $\boldsymbol{x}$. This belief is called the posterior, $p(\boldsymbol{\theta}|\boldsymbol{x})$, and is defined through the Bayes rule

$$p(\boldsymbol{\theta}|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\boldsymbol{x})}. \tag{2.4}$$

The term $p(\boldsymbol{\theta})$ is called the prior and is a part of the statistical model that represents the belief about the parameters before observing $\boldsymbol{x}$. The term $p(\boldsymbol{x})$ is called the evidence and is the marginal probability of observing $\boldsymbol{x}$. It can be computed through the law of total probability

$$p(\boldsymbol{x}) = \int p(\boldsymbol{x}|\boldsymbol{\theta})p(\boldsymbol{\theta})d\boldsymbol{\theta}. \tag{2.5}$$

The posterior is a central component of Bayesian inference as it defines all the information we have about the parameters $\boldsymbol{\theta}$. It will be used to derive all the other quantities of interest.

2.3 POINT ESTIMATION

Point estimation corresponds to the task of finding the parameters that lead to an appropriate probability model according to the data. In other words, we aim to find parameter values $\boldsymbol{\theta}$ such that $p(\boldsymbol{x}|\boldsymbol{\theta})$ is close to $p^*(\boldsymbol{x})$. If we measure how close those distributions are using the KL-divergence and assume we have access to N samples $\boldsymbol{x}_1, ..., \boldsymbol{x}_N \sim p^*(\boldsymbol{x})$, we aim for

$$\begin{aligned}
\arg\min_{\boldsymbol{\theta}} \mathrm{KL}\left[p^*(\boldsymbol{x})||p(\boldsymbol{x}|\boldsymbol{\theta})\right] &= \arg\min_{\boldsymbol{\theta}} \int \log\left(\frac{p^*(\boldsymbol{x})}{p(\boldsymbol{x}|\boldsymbol{\theta})}\right) p^*(\boldsymbol{x}) d\boldsymbol{x} \\
&\simeq \arg\min_{\boldsymbol{\theta}} \frac{1}{N}\sum_{i=1}^{N} \log\left(\frac{p^*(\boldsymbol{x}_i)}{p(\boldsymbol{x}_i|\boldsymbol{\theta})}\right) \\
&= \arg\max_{\boldsymbol{\theta}} \frac{1}{N}\sum_{i=1}^{N} \log\left(p(\boldsymbol{x}_i|\boldsymbol{\theta})\right) - \frac{1}{N}\sum_{i=1}^{N} \log\left(p(\boldsymbol{x}_i)\right) \\
&= \arg\max_{\boldsymbol{\theta}} \sum_{i=1}^{N} \log\left(p(\boldsymbol{x}_i|\boldsymbol{\theta})\right) \\
&= \arg\max_{\boldsymbol{\theta}} \prod_{i=1}^{N} p(\boldsymbol{x}_i|\boldsymbol{\theta}).
\end{aligned} \tag{2.6}$$

In the remainder, we will use $\boldsymbol{x}$ to represent both single and multiple observations from a phenomenon. The model is then a model for a single or multiple observations from the phenomenon. A model for multiple observations can be obtained from a model for single observations. $p(\boldsymbol{x}_1, ..., \boldsymbol{x}_N|\boldsymbol{\theta}) = \prod_{i=1}^{N} p(\boldsymbol{x}_i|\boldsymbol{\theta})$. An appropriate probability model can then be the one that maximizes the likelihood. This leads to the maximum likelihood estimator (MLE) of those parameters.

Definition 2.3.1. The maximum likelihood estimator $\boldsymbol{\theta}_{\mathrm{MLE}}$ is defined as

$$\boldsymbol{\theta}_{\mathrm{MLE}} = \arg\max_{\boldsymbol{\theta}} p(\boldsymbol{x}|\boldsymbol{\theta}). \tag{2.7}$$

Let us assume that we make a feather falling experiments for heights 3m, 5m and 7m and observe measured falling time of 4.68s, 6.97s and 9.69s respectively. The probability of those observations given our model is then

$$\begin{aligned}
p(\boldsymbol{x}|g) &= \frac{1}{\sigma_{\text{measurement}}\sqrt{2\pi}} \exp\left(-\frac{\left(4.68-\sqrt{\frac{6}{g}}\right)^2}{2\sigma^2_{\text{measurement}}}\right) \\
&\times \frac{1}{\sigma_{\text{measurement}}\sqrt{2\pi}} \exp\left(-\frac{\left(6.97-\sqrt{\frac{10}{g}}\right)^2}{2\sigma^2_{\text{measurement}}}\right) \\
&\times \frac{1}{\sigma_{\text{measurement}}\sqrt{2\pi}} \exp\left(-\frac{\left(9.69-\sqrt{\frac{14}{g}}\right)^2}{2\sigma^2_{\text{measurement}}}\right) \\
&= \left(\frac{1}{\sigma_{\text{measurement}}\sqrt{2\pi}}\right)^3 \exp\left(\frac{-\left(4.68-\sqrt{\frac{6}{g}}\right)^2-\left(6.97-\sqrt{\frac{10}{g}}\right)^2-\left(9.69-\sqrt{\frac{14}{g}}\right)^2}{2\sigma^2_{\text{measurement}}}\right).
\end{aligned} \tag{2.8}$$

We have that

$$\log p(\boldsymbol{x}|g) \propto -\left(4.68-\sqrt{\frac{6}{g}}\right)^2-\left(6.97-\sqrt{\frac{10}{g}}\right)^2-\left(9.69-\sqrt{\frac{14}{g}}\right)^2. \tag{2.9}$$

From that we can compute the maximum likelihood estimator

$$\begin{aligned}
g_{\text{MLE}} &= \arg\max_g p(\boldsymbol{x}|g) \\
&= \arg\max_g \log p(\boldsymbol{x}|g) \\
&= \arg\max_g -\left(4.68-\sqrt{\frac{6}{g}}\right)^2-\left(6.97-\sqrt{\frac{10}{g}}\right)^2-\left(9.69-\sqrt{\frac{14}{g}}\right)^2 \\
&\simeq 0.1849.
\end{aligned} \tag{2.10}$$

In Bayesian philosophy, it would make sense to consider instead the probability model corresponding to the parameters in which we have the highest belief. The belief over model parameters is the posterior. This leads to the maximum a posteriori estimator (MAP) of those parameters.

Definition 2.3.2. The maximum a posteriori estimator $\boldsymbol{\theta}_{\text{MAP}}$ is defined as

$$\boldsymbol{\theta}_{\text{MAP}} = \arg\max_{\boldsymbol{\theta}} p(\boldsymbol{\theta}|\boldsymbol{x}) = \arg\max_{\boldsymbol{\theta}} p(\boldsymbol{x}|\boldsymbol{\theta})p(\boldsymbol{\theta}). \tag{2.11}$$

Let us assume that we have the a priori belief that $\log g$ should follow a uniform distribution between $\log(0.01)$ and $\log(1000)$. We have

$$p(g) = \frac{1}{g(\log(1000)-\log(0.01))}. \tag{2.12}$$

Consequently,

$$\begin{aligned}\log p(g|\boldsymbol{x}) &\propto \log p(\boldsymbol{x}|g) + \log p(g)\\ &= -3\log(\sigma_{\text{measurement}}\sqrt{2\pi}) + \frac{-\left(4.68-\sqrt{\frac{6}{g}}\right)^2 - \left(6.97-\sqrt{\frac{10}{g}}\right)^2 - \left(9.69-\sqrt{\frac{14}{g}}\right)^2}{2\sigma^2_{\text{measurement}}}\\ &- \log(g) - \log(\log(1000) - \log(1)).\end{aligned} \tag{2.13}$$

If we assume that $\sigma_{\text{measurement}} = 0.5\text{s}$, then the maximum a posteriori is

$$g_{\text{MAP}} = \arg\max_g p(g|\boldsymbol{x}) \simeq 0.1838. \tag{2.14}$$

The parameters $\boldsymbol{\theta}_{\text{MLE}}$ and $\boldsymbol{\theta}_{\text{MAP}}$ could then optionally be used to obtain a probability model. This probability model can then be used to make predictions on future observations. If we consider the maximum likelihood estimator $g_{\text{MLE}} = 0.1849$, we can predict that the measured fall time from a height $h = 6\text{m}$ would follow the distribution $\mathcal{N}(\mu = \sqrt{\frac{2h}{g_{\text{MLE}}}} = 8.056, \sigma = \sigma_{\text{measurement}})$.

2.4 INTERVAL ESTIMATION

Instead of a point estimation, we might often want to obtain a range of models that are consistent with the observation. This is done through interval or region estimation. In frequentist statistics, these are called confidence regions or intervals.

Definition 2.4.1. A level α confidence region generation function $C_\alpha(.)$ is a function such that

$$P(\boldsymbol{\theta} \in C_\alpha(\boldsymbol{X})) \geq \alpha, \quad \forall \boldsymbol{\theta}, \tag{2.15}$$

where $\boldsymbol{X}$ is the random variable associated with the observation generated according to the model with parameter $\boldsymbol{\theta}$.

A level α confidence region generation function is then a function that, when given as input a random variable associated with the observation according to the model with some parameters $\boldsymbol{\theta}$, outputs a random region that contains those parameters with probability α. The probability statement is then on the constructed region and not on the parameters $\boldsymbol{\theta}$ that are not random variables. When evaluated on an observation $\boldsymbol{x}$, this yields a level α confidence region $C_\alpha(\boldsymbol{x})$.

Such regions are sometimes hard to construct analytically and must be approximated. This leads to the definition of coverage of a region generation function.

Definition 2.4.2. The frequentist coverage probability of a region generator function is the probability that generated regions based on some observations generated from the model with some parameters contain those parameters,

$$\text{coverage}(C, \boldsymbol{\theta}) = P(\boldsymbol{\theta} \in C(\boldsymbol{X})). \tag{2.16}$$

It follows that if a region generation function has coverage greater of equal to α for all $\boldsymbol{\theta}$, then this is a valid level α confidence region generation function. The coverage probability can then be used to measure the validity of a region generation function.

When the parameters $\boldsymbol{\theta}$ are one-dimensional and the generated regions are contiguous, those regions correspond to confidence intervals. Confidence intervals are defined by a lower function L_α and an upper function U_α such that

$$P(L_\alpha(\boldsymbol{X}) \leq \boldsymbol{\theta} \leq U_\alpha(\boldsymbol{X})) \geq \alpha, \quad \forall \boldsymbol{\theta}. \tag{2.17}$$

In the Bayesian setting, we assign a belief probability on parameters, and credible regions are constructed.

Definition 2.4.3. A level α credible region is defined as a region C_α such that

$$\int_{C_\alpha} p(\boldsymbol{\theta}|\boldsymbol{x}) d\boldsymbol{\theta} = \alpha. \tag{2.18}$$

It is a region in which we have a belief that the parameters have a probability α to be included in. This is then a probability statement about the unknown parameters themselves. Similarly to confidence intervals, when the parameters $\boldsymbol{\theta}$ are one-dimensional and we consider contiguous regions, those regions are defined by an upper and lower bound and are called credible intervals. There are many regions satisfying this property depending on which part of the parameter space is prioritized to be included. Although not the only valid choice, a common choice is to consider the highest posterior density credible region. This is defined as a credible region for which all included elements of the parameter space have a higher posterior density than the ones excluded. In other words, it prioritizes high posterior density elements to be included in the credible region.

While the maximum a posteriori could be computed using only the likelihood $p(\boldsymbol{x}|\boldsymbol{\theta})$ and the prior $p(\boldsymbol{\theta})$, computing a credible region requires the knowledge of the posterior $p(\boldsymbol{\theta}|\boldsymbol{x})$ explicitly. Computing the posterior requires computing the evidence term $p(\boldsymbol{x})$, which involves the computation of the integral $\int p(\boldsymbol{x}|\boldsymbol{\theta})p(\boldsymbol{\theta})d\boldsymbol{\theta}$ that might be hard to solve analytically. Therefore, we often have to rely on approximations of the posterior to derive credible regions. Two widely used techniques to approximate the posterior are Markov Chain Monte-Carlo and Variational Inference techniques.

MARKOV CHAIN MONTE-CARLO Markov Chain Monte-Carlo (MCMC) techniques consist in building a Markov chain whose stationary distribution is the posterior. A Markov chain defines a stochastic transition between states represented by a probability distribution over the next states, given the current state. In this setting, states are parameter values, and the transitions are also conditioned on the observation. The Markov chain then defines the transition $p(\boldsymbol{\theta}_{i+1}|\boldsymbol{\theta}_i, \boldsymbol{x})$. A stationary distribution of a Markov chain is a distribution over $\boldsymbol{\theta}$ such that applying the transition on elements from that distribution returns elements following the same distribution. In other words, π is a stationary distribution if, for every parameter values $\tilde{\boldsymbol{\theta}}$,

$$\int \pi(\boldsymbol{\theta})p(\tilde{\boldsymbol{\theta}}|\boldsymbol{\theta}, \boldsymbol{x})d\boldsymbol{\theta} = \pi(\tilde{\boldsymbol{\theta}}). \tag{2.19}$$

It follows that if this stationary distribution is the posterior and convergence to that stationary distribution is guaranteed, then parameters sampled from that Markov chain will asymptotically follow the posterior distribution.

The Metropolis-Hastings algorithm (Metropolis et al., 1953; Hastings, 1970) constructs a Markov chain that asymptotically follows the posterior distribution. It uses a proposal distribution $g(\tilde{\boldsymbol{\theta}}|\boldsymbol{\theta}_i, \boldsymbol{x})$ that is then corrected to get transitions leading to the posterior distribution. The transitions are defined as follows.

1. Sample $\tilde{\boldsymbol{\theta}} \sim g(\tilde{\boldsymbol{\theta}}|\boldsymbol{\theta}_i, \boldsymbol{x})$.
2. With probability $\min\left(1, \frac{p(\tilde{\boldsymbol{\theta}}|\boldsymbol{x})g(\boldsymbol{\theta}|\tilde{\boldsymbol{\theta}}, \boldsymbol{x})}{p(\boldsymbol{\theta}|\boldsymbol{x})g(\tilde{\boldsymbol{\theta}}|\boldsymbol{\theta}, \boldsymbol{x})}\right)$ set $\boldsymbol{\theta}_{i+1} = \tilde{\boldsymbol{\theta}}$, otherwise set $\boldsymbol{\theta}_{i+1} = \boldsymbol{\theta}_i$.

To show that such transitions lead to the stationary distribution being the posterior, let us first show that a distribution π is a stationary distribution if it satisfies the detailed balance equation

$$\pi(\boldsymbol{\theta})p(\tilde{\boldsymbol{\theta}}|\boldsymbol{\theta}, \boldsymbol{x}) = \pi(\tilde{\boldsymbol{\theta}})p(\boldsymbol{\theta}|\tilde{\boldsymbol{\theta}}, \boldsymbol{x}). \tag{2.20}$$

If that is the case, then

$$\int \pi(\boldsymbol{\theta})p(\tilde{\boldsymbol{\theta}}|\boldsymbol{\theta}, \boldsymbol{x})d\boldsymbol{\theta} = \int \pi(\tilde{\boldsymbol{\theta}})p(\boldsymbol{\theta}|\tilde{\boldsymbol{\theta}}, \boldsymbol{x})d\boldsymbol{\theta} = \pi(\tilde{\boldsymbol{\theta}}) \int p(\boldsymbol{\theta}|\tilde{\boldsymbol{\theta}}, \boldsymbol{x})d\boldsymbol{\theta} = \pi(\tilde{\boldsymbol{\theta}}). \tag{2.21}$$

and π is then a stationary distribution. If we assume that the transitions are built using a proposal distribution $g(\tilde{\boldsymbol{\theta}}|\boldsymbol{\theta}, \boldsymbol{x})$ and an acceptance probability $A(\tilde{\boldsymbol{\theta}}, \boldsymbol{\theta}, \boldsymbol{x})$, then

$$p(\tilde{\boldsymbol{\theta}}|\boldsymbol{\theta}, \boldsymbol{x}) \propto g(\tilde{\boldsymbol{\theta}}|\boldsymbol{\theta}, \boldsymbol{x})A(\tilde{\boldsymbol{\theta}}, \boldsymbol{\theta}, \boldsymbol{x}). \tag{2.22}$$

To satisfy the detailed balance equation, we need

$$\frac{A(\tilde{\boldsymbol{\theta}}, \boldsymbol{\theta}, \boldsymbol{x})}{A(\boldsymbol{\theta}, \tilde{\boldsymbol{\theta}}, \boldsymbol{x})} = \frac{\pi(\tilde{\boldsymbol{\theta}})g(\boldsymbol{\theta}|\tilde{\boldsymbol{\theta}}, \boldsymbol{x})}{\pi(\boldsymbol{\theta})g(\tilde{\boldsymbol{\theta}}|\boldsymbol{\theta}, \boldsymbol{x})}. \tag{2.23}$$

This is satisfied by the following acceptance probability

$$A(\tilde{\boldsymbol{\theta}}, \boldsymbol{\theta}, \boldsymbol{x}) = \min\left(1, \frac{\pi(\tilde{\boldsymbol{\theta}})g(\boldsymbol{\theta}|\tilde{\boldsymbol{\theta}}, \boldsymbol{x})}{\pi(\boldsymbol{\theta})g(\tilde{\boldsymbol{\theta}}|\boldsymbol{\theta}, \boldsymbol{x})}\right). \tag{2.24}$$

as there is always one of $A(\tilde{\boldsymbol{\theta}}, \boldsymbol{\theta}, \boldsymbol{x})$ or $A(\boldsymbol{\theta}, \tilde{\boldsymbol{\theta}}, \boldsymbol{x})$ that is equal to one and the other is equal to the right-hand side of equation 2.23 or its inverse. We aim to build a Markov chain that has the posterior as a stationary distribution and set $\pi(\boldsymbol{\theta}) = p(\boldsymbol{\theta}|\boldsymbol{x})$. The posterior ratio can be easily computed as the evidence cancels out:

$$\frac{p(\tilde{\boldsymbol{\theta}}|\boldsymbol{x})}{p(\boldsymbol{\theta}|\boldsymbol{x})} = \frac{p(\boldsymbol{x}|\tilde{\boldsymbol{\theta}})p(\tilde{\boldsymbol{\theta}})}{p(\boldsymbol{x}|\boldsymbol{\theta})p(\boldsymbol{\theta})}. \tag{2.25}$$

MCMC can, therefore, be used to draw samples from the posterior $p(\boldsymbol{\theta}|\boldsymbol{x})$ while only being able to evaluate the likelihood $p(\boldsymbol{x}|\boldsymbol{\theta})$ and the prior $p(\boldsymbol{\theta})$. Those samples can then be used to construct approximate credible regions. Let us note that the samples drawn from the Markov chain are not independently and identically distributed (i.i.d.) due to the correlation between consecutive samples and that a sufficient number of steps must be performed before reaching the stationary distribution. Those two elements hence add error to the credible region approximations. With this in mind, it is a good practice to run an MCMC algorithm with multiple chains, i.e., multiple instances of the algorithm, to decrease the correlation between samples. It is also a good practice to discard the first samples produced by each chain when the stationary distribution has not yet been attained. This is called the burn-in period.

The algorithm is correct for any proposal distribution $g(\boldsymbol{\theta}|\tilde{\boldsymbol{\theta}}, \boldsymbol{x})$. However, the closer this distribution is to the posterior, the higher the acceptance rate. In addition, the less consecutive samples are correlated, the fewer steps will be required to visit many regions of the posterior space. Much of the research in MCMC has then been directed towards finding the best proposals. Some proposals do not use information about the posterior (Goodman and Weare, 2010; Nelson et al., 2013) while others leverage quantities such as the gradient of the unnormalized posterior (Welling and Teh, 2011; Neal, 2012).

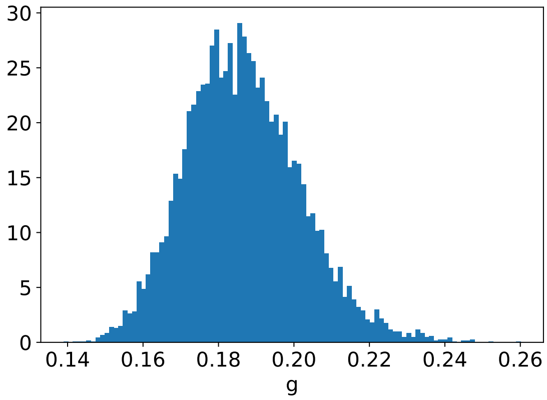

Figure 2.2: Histogram of samples obtained by running an MCMC algorithm on the feather falling model with measured time 4.68s, 6.97s and 9.69s for heights of 3m, 5m and 7m respectively.

As an illustration, Figure 2.2 shows a histogram of samples obtained by running an MCMC algorithm on the feather falling model with the same observations as for point prediction.

This histogram then approximates the posterior density. We can observe that there is a single mode and that there are samples roughly between $g = 0.14$ and $g = 0.25$.

VARIATIONAL INFERENCE While MCMC only allows drawing samples from the posterior and does not provide an analytical form of it, variational inference (VI) optimizes a family of distributions to find a member of that family that is close to the actual posterior. The idea is to start with a family of distributions over $\boldsymbol{\theta}$, also called variational family, which we will denote by $\pi_{\boldsymbol{\phi}}(\boldsymbol{\theta})$. That family has parameters $\boldsymbol{\phi}$ that will be optimized to find a distribution close to the posterior. For example, if we assume that the parameters follow a multivariate Normal distribution, $\boldsymbol{\theta} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, then the parameters are $\boldsymbol{\phi} = (\boldsymbol{\mu}, \boldsymbol{\Sigma})$.

We optimize those parameters to minimize the KL divergence between the variational distribution and the posterior:

$$\boldsymbol{\phi}^* = \arg\min_{\boldsymbol{\phi}} \mathrm{KL}\left[\pi_{\boldsymbol{\phi}}(\boldsymbol{\theta}) || p(\boldsymbol{\theta}|\boldsymbol{x})\right]. \tag{2.26}$$

Because the posterior $p(\boldsymbol{\theta}|\boldsymbol{x})$ is unknown, the KL divergence cannot be computed, and an alternative objective will be used. We first rewrite the objective as

$$\begin{aligned} \mathrm{KL}\left[\pi_{\boldsymbol{\phi}}(\boldsymbol{\theta}) || p(\boldsymbol{\theta}|\boldsymbol{x})\right] &= \int \pi_{\boldsymbol{\phi}}(\boldsymbol{\theta}) \log \frac{\pi_{\boldsymbol{\phi}}(\boldsymbol{\theta})}{p(\boldsymbol{\theta}|\boldsymbol{x})} d\boldsymbol{\theta} \\ &= \int \pi_{\boldsymbol{\phi}}(\boldsymbol{\theta}) \log \frac{\pi_{\boldsymbol{\phi}}(\boldsymbol{\theta})}{p(\boldsymbol{x}|\boldsymbol{\theta}) p(\boldsymbol{\theta})} d\boldsymbol{\theta} + \log(p(\boldsymbol{x})). \end{aligned} \tag{2.27}$$

By reorganizing the terms, we have that

$$\log(p(\boldsymbol{x})) = \mathrm{KL}\left[\pi_{\boldsymbol{\phi}}(\boldsymbol{\theta}) || p(\boldsymbol{\theta}|\boldsymbol{x})\right] + \int \pi_{\boldsymbol{\phi}}(\boldsymbol{\theta}) \log \frac{p(\boldsymbol{x}|\boldsymbol{\theta}) p(\boldsymbol{\theta})}{\pi_{\boldsymbol{\phi}}(\boldsymbol{\theta})} d\boldsymbol{\theta}, \tag{2.28}$$

and because $\mathrm{KL}\left[\pi_{\boldsymbol{\phi}}(\boldsymbol{\theta}) || p(\boldsymbol{\theta}|\boldsymbol{x})\right] \geq 0$,

$$\log(p(\boldsymbol{x})) \geq \int \pi_{\boldsymbol{\phi}}(\boldsymbol{\theta}) \log \frac{p(\boldsymbol{x}|\boldsymbol{\theta}) p(\boldsymbol{\theta})}{\pi_{\boldsymbol{\phi}}(\boldsymbol{\theta})} d\boldsymbol{\theta} = \mathbb{E}_{\pi_{\boldsymbol{\phi}}(\boldsymbol{\theta})}\left[\log \frac{p(\boldsymbol{x}|\boldsymbol{\theta}) p(\boldsymbol{\theta})}{\pi_{\boldsymbol{\phi}}(\boldsymbol{\theta})}\right]. \tag{2.29}$$

For that reason, the term $\mathbb{E}_{\pi_{\boldsymbol{\phi}}(\boldsymbol{\theta})}\left[\log \frac{p(\boldsymbol{x}|\boldsymbol{\theta}) p(\boldsymbol{\theta})}{\pi_{\boldsymbol{\phi}}(\boldsymbol{\theta})}\right]$ is often called the evidence lower bound (ELBO). From Equation 2.28, we observe that, by maximizing the ELBO and hence making it closer to $\log(p(\boldsymbol{x}))$, we push the term $\mathrm{KL}\left[\pi_{\boldsymbol{\phi}}(\boldsymbol{\theta}) || p(\boldsymbol{\theta}|\boldsymbol{x})\right]$ towards zero and hence minimize it. Maximizing the ELBO is hence equivalent to minimizing $\mathrm{KL}\left[\pi_{\boldsymbol{\phi}}(\boldsymbol{\theta}) || p(\boldsymbol{\theta}|\boldsymbol{x})\right]$. The ELBO can be approximated using Monte-Carlo

$$\mathbb{E}_{\pi_{\boldsymbol{\phi}}(\boldsymbol{\theta})}\left[\log \frac{p(\boldsymbol{x}|\boldsymbol{\theta}) p(\boldsymbol{\theta})}{\pi_{\boldsymbol{\phi}}(\boldsymbol{\theta})}\right] \simeq \frac{1}{N} \sum_{i=1}^{N} \log \frac{p(\boldsymbol{x}|\boldsymbol{\theta}_i) p(\boldsymbol{\theta}_i)}{\pi_{\boldsymbol{\phi}}(\boldsymbol{\theta}_i)}, \quad \boldsymbol{\theta}_i \sim \pi_{\boldsymbol{\phi}}(\boldsymbol{\theta}), \tag{2.30}$$

and is, hence, a valid training objective.

Once the variational parameters have been optimized, the distribution $\pi_{\boldsymbol{\phi}^*}(\boldsymbol{\theta})$ can be used as a drop-in replacement for $p(\boldsymbol{\theta}|\boldsymbol{x})$ in the downstream computations. By running variational inference on the feather falling simulator with the same observations as for point prediction and MCMC, we obtain the approximate posterior density shown in Figure 2.3. We used a log normal distribution as variational family. It is parametrized by two parameters $\boldsymbol{\phi} = (\mu, \sigma)$ such that $\log g$ follows approximately a posteriori the distribution $\mathcal{N}(\mu, \sigma)$. By maximizing the ELBO, we obtain the parameters $\mu = -1.6873$ and $\sigma = 0.0829$. It should, however, be kept in mind that it only constitutes an approximation as, on the one hand, the posterior $p(\boldsymbol{\theta}|\boldsymbol{x})$ might not be contained in the variational family, and on the other hand, the optimization might not be perfect.

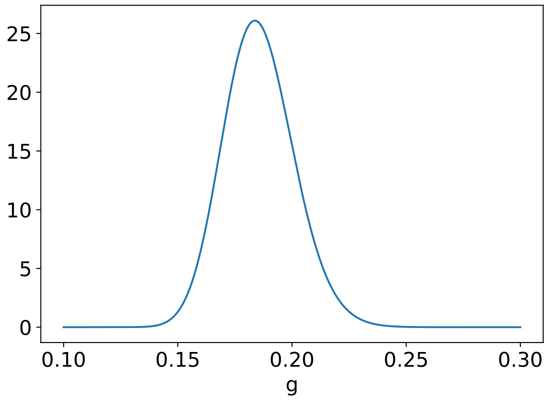

Figure 2.3: Approximate posterior density obtained by running a variational inference algorithm on the feather falling model with measured time 4.68s, 6.97s and 9.69s for heights of 3m, 5m and 7m respectively. The plotted variational distribution is a log normal distribution with parameters $\mu = -1.6873$ and $\sigma = 0.0829$.

2.5 HYPOTHESIS TESTING

Hypothesis testing is the task of assessing the compatibility between observed data and a hypothesis. The hypothesis to test is often called the null hypothesis and is denoted by H_0. The alternative hypothesis corresponds to cases where the null hypothesis does not hold and is denoted H_1. Of interest for this thesis is the fact that the hypothesis to test may be linked to a scientific theory. For example, one could want to test if the hypothesis $g = 0.5$ could be rejected from data or not. In that case, the null hypothesis is $H_0 : g = 0.5$, and the alternative hypothesis is $H_1 : g \neq 0.5$. We may also want to test if the observations made are indeed coherent with our model or not.

In the frequentist setting, we compute the adequacy between the observations and the null hypothesis. There are two outcomes from this test. If the observations are likely under the null hypothesis, no conclusion can be drawn regarding this hypothesis. Otherwise, if the observations are unlikely under the null hypothesis, we can conclude that the null hypothesis is likely not to hold and hence reject it. To test if the observations are likely under the null hypothesis, we first compress the observations to a single real number through a summary statistic function that we will call $T(.)$. That function should be designed to be, on average, higher if observations are drawn under the alternative hypothesis than under the null hypothesis. The null hypothesis is then rejected if that summary statistic is sufficiently high, i.e. above a given threshold c. The rejection region is then of the form

$$R = \{\boldsymbol{x} : T(\boldsymbol{x}) > c\}. \tag{2.31}$$

The threshold c is chosen to match a given acceptable false rejection rate α, often called the level of a test. A level α test is then a test such that the probability of erroneously rejecting the null hypothesis is lower than α. The threshold c must then satisfy

$$P(\text{reject } H_0 | H_0) = P(T(\boldsymbol{X}) > c | H_0) \leq \alpha. \tag{2.32}$$

If the null hypothesis takes the form $H_0 : \boldsymbol{\theta} \in \boldsymbol{\Theta}_0$, then this must hold for all possible values of $\boldsymbol{\theta}$ in $\boldsymbol{\Theta}_0$:

$$P(T(\boldsymbol{X}) > c | \boldsymbol{\theta}) \leq \alpha, \quad \forall \boldsymbol{\theta} \in \boldsymbol{\Theta}_0. \tag{2.33}$$

For example, we could test if the observed measured times are indeed likely to be sampled from our model with $\boldsymbol{\theta} = \boldsymbol{\theta}_{\text{MLE}}$. If that is the case, then the residuals

$$\text{measured time} - \sqrt{\frac{2h}{g_{\text{MLE}}}}$$

are sampled from the distribution $\mathcal{N}(0, \sigma_{\text{measurement}})$. We can compute the following residuals for the observations made:

$$4.68 - \sqrt{\frac{6}{g_{\text{MLE}}}} = -1.016, \quad 6.97 - \sqrt{\frac{10}{g_{\text{MLE}}}} = -0.384, \quad 9.69 - \sqrt{\frac{14}{g_{\text{MLE}}}} = 0.988.$$

To test the hypothesis, we perform a Kolmogorov-Smirnov test that uses as test statistic, T, the maximum absolute difference between the empirical cumulative distribution function (c.d.f.) defined by the residuals and the c.d.f. of the distribution $\mathcal{N}(0, \sigma_{\text{measurement}})$. We obtain a p-value of 0.47, meaning that we could only reject the hypothesis that the samples come from that distribution at a level α equal or higher to 0.47. Therefore, we cannot confidently reject that hypothesis from the observations at hand. Let us note that it does not mean that this hypothesis is correct. It could potentially be rejected by collecting more observations.

There is a direct link between hypothesis tests and confidence regions. Let us call by $R_\alpha(\boldsymbol{\theta}_0)$ the rejection region of a test $H_0 : \boldsymbol{\theta} = \boldsymbol{\theta}_0$ and $H_1 : \boldsymbol{\theta} \neq \boldsymbol{\theta}_0$ at level α. Then the region

$$C_{1-\alpha}(\boldsymbol{x}) = \{\boldsymbol{\theta} : \boldsymbol{x} \notin R_\alpha(\boldsymbol{\theta})\} \tag{2.34}$$

is a valid level $(1 - \alpha)$ confidence region. Similarly, the rejection region

$$R_\alpha(\boldsymbol{\theta}_0) = \{\boldsymbol{x} : \boldsymbol{\theta}_0 \notin C_{1-\alpha}(\boldsymbol{x})\}, \tag{2.35}$$

is a valid rejection region for the test. To prove it, let us first observe that

$$P(\boldsymbol{\theta}_0 \in C_{1-\alpha}(\boldsymbol{X})|\boldsymbol{\theta} = \boldsymbol{\theta}_0) = P(\boldsymbol{X} \notin R_\alpha(\boldsymbol{\theta}_0)|\boldsymbol{\theta} = \boldsymbol{\theta}_0) = 1 - P(\boldsymbol{X} \in R_\alpha(\boldsymbol{\theta}_0)|\boldsymbol{\theta} = \boldsymbol{\theta}_0). \quad (2.36)$$

As $P(\boldsymbol{X} \in R_\alpha(\boldsymbol{\theta}_0)|\boldsymbol{\theta} = \boldsymbol{\theta}_0) \leq \alpha$ by definition of an hypothesis test, we have

$$P(\boldsymbol{\theta}_0 \in C_{1-\alpha}(\boldsymbol{X})|\boldsymbol{\theta} = \boldsymbol{\theta}_0) \geq 1 - \alpha. \quad (2.37)$$

Similarly, if $P(\boldsymbol{\theta}_0 \in C_{1-\alpha}(\boldsymbol{X})|\boldsymbol{\theta} = \boldsymbol{\theta}_0) \geq 1 - \alpha.$, then

$$P(\boldsymbol{X} \in R_\alpha(\boldsymbol{\theta}_0)|\boldsymbol{\theta} = \boldsymbol{\theta}_0) \leq \alpha. \quad (2.38)$$

These properties then link the problem of finding a confidence region to the problem of testing a hypothesis. In addition to linking the two tasks, it also provides a way to approximate confidence regions by constructing a series of hypothesis test rejection regions. This method is called the Neyman construction (Neyman, 1937). To determine if some parameter values $\boldsymbol{\theta}_0$ should be included in the confidence region at level α, a critical value c is computed such that $P(T(\boldsymbol{X}) > c|\boldsymbol{\theta} = \boldsymbol{\theta}_0) \leq \alpha$. If the observation $\boldsymbol{x}$ is such that $T(\boldsymbol{x}) \leq c$, then $\boldsymbol{\theta}_0$ is included in the confidence region. Otherwise, it is not. By repeating this procedure for a fine grid of parameter values spanning the support of possible parameter values, an approximate confidence region can be constructed.

In the Bayesian setting, hypothesis testing takes a different form. As for the parameters of a statistical model, we can assign a probability to the different hypotheses corresponding to our belief that these hypotheses are true. The belief we have about a hypothesis is summarized by its posterior probability

$$p(H_0|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|H_0)p(H_0)}{p(\boldsymbol{x})}. \quad (2.39)$$

Hypothesis testing then boils down to choosing a hypothesis according to our belief. A common decision rule is then to favor H_0 over H_1 if

$$P(H_0|\boldsymbol{x}) \geq P(H_1|\boldsymbol{x}). \quad (2.40)$$

Depending on the application, rejecting H_0 while it is valid or accepting it while invalid might have different consequences, and we may want to avoid a given type of error as much as possible. An example that will drive this thesis is that refuting a scientific theory that is valid is much more harmful than failing to reject an invalid theory. In those cases, we can assign a cost to each error type and find the decision rule that minimizes the expected cost. We denote by C_{10} the cost of rejecting H_0 (choosing H_1) while it is valid and by C_{01} the cost of selecting H_0 while it is invalid. The expected cost is then expressed

$$C = C_{10}P(\text{reject}(H_0|H_0)) + C_{01}P(\text{not reject}H_0|H_1). \quad (2.41)$$

It is minimized when favoring H_0 over H_1 if

$$P(H_0|\boldsymbol{x})C_{10} \geq P(H_1|\boldsymbol{x})C_{01}. \tag{2.42}$$

3 SIMULATION-BASED INFERENCE

Simulators are being used more and more in science to describe complex systems that evolve in an iterative manner. For example, simulators are used in astrophysics to describe the evolution of stellar streams with various dark matter masses (Jo and Jason, 2017), to model gravitational waves (Hannam et al., 2014), or to model gravitational lensing (Wagner-Carena et al., 2024). Simulators are also used in biology to model brain activity (Destexhe and Huguenard, 2000), in epidemiology (Weligampola et al., 2023), and for particle physics (ATLAS et al., 2012). It allows for the simulation of phenomena under varying conditions and the comparison of those simulations to what is observed in reality. If we take back the example of a feather falling from the previous chapter, we can quickly identify that the proposed model is oversimplistic as it only takes gravity into account and not other forces that may be applied to the feather. We will improve that model by taking into account the drag due to the friction with air. Drag acts in the opposite direction to the fall, and its magnitude is given by the following equation

$$F_D = \frac{1}{2}\,\rho\, v^2\, C_D\, A, \tag{3.1}$$

where ρ is the air density, v is the speed of the feather, C_D is a drag coefficient, and A is the cross-sectional area, i.e., the area of the projection of the object on the plane orthogonal to the fall direction. There are three quantities in this equation that are not constant over time. The speed evolves over time due to gravity, and the cross-sectional area evolves as the object's orientation changes during the fall. If we consider that there are air movements, air density under the feather is not constant either. It is then hard to find an analytical formula for the fall time. However, we can simulate the fall by discretizing time and by modeling the fall as a succession of time steps for which different forces apply. This is illustrated in Algorithm 3.1. At every time step, forces are calculated based on current conditions. Speed and position are then updated based on the current forces applied, and a new object orientation and air density are computed. This is repeated until the feather has fallen from a given height, and the measurement of falling time is simulated.

Algorithm 3.1 Feather falling simulator.

```
Inputs:            gravity constant $g$, drag coefficient $C_D$, height $h$, mass $m$ and
                   measurement error standard deviation $\sigma_{\text{measurement}}$.
Outputs:           The measured fall time.
hyper-parameters:  Duration of a step $\Delta t$

distance = 0
speed = 0
elapsed_time = 0
$\rho = 1.225$
orientation = initial_orientation
repeat
    $A$ = cross_sectional_area(orientation)
    drag_acceleration = $\frac{\text{speed}^2\ \rho\ C_D\ A}{2m}$
    next_speed = speed + $\Delta t(g -$ drag_acceleration$)$
    distance = distance + $\Delta t \frac{\text{speed+next\_speed}}{2}$
    orientation = rotation(orientation, $\Delta t$)
    $\rho$ = air_movement($\rho, \Delta_t$)
    speed = next_speed
    elapsed_time = elapsed_time + $\Delta t$
until distance $\geq h$
measured_time $\sim \mathcal{N}$(elapsed_time, $\sigma_{\text{measurement}}$)
return measured_time
```

3.1 SIMULATORS AS STATISTICAL MODELS

In the example of a feather falling, stochasticity arises in the measurement as a random measurement error is made. It may also be hard to model the change of orientation of the feather and air movement. We might want to model those as random processes where random rotations and air movements are performed at each step. Both these elements lead to a stochastic simulator that would generate different measured fall times if run several times with the same parameters. Examples of simulator runs are provided in Figure 3.1 for various values of gravity constant g and drag coefficient C_D.

A stochastic simulator is a form of statistical model that includes latent random variables corresponding to the stochastic steps during observation generation. We will denote those latent variables by $\boldsymbol{z}$. In the example of the feather falling simulator, those latent variables correspond to the random rotations and air movements performed at each time step. The simulator is also conditioned on some parameters $\boldsymbol{\theta}$ that are g and C_D in the example. Formally, the simulator defines the distributions $p(\boldsymbol{x}|\boldsymbol{z}, \boldsymbol{\theta})$ and $p(\boldsymbol{z}|\boldsymbol{\theta})$. Consequently, it also defines the distribution $p(\boldsymbol{x}, \boldsymbol{z}|\boldsymbol{\theta}) = p(\boldsymbol{x}|\boldsymbol{z}, \boldsymbol{\theta})p(\boldsymbol{z}|\boldsymbol{\theta})$. The distribution $p(\boldsymbol{x}|\boldsymbol{\theta})$ cannot be computed analytically because it would require marginalizing and, hence, integrating over all possible latent variables, which is, in most cases, impossible. In the Bayesian setting, if we were to perform inference with variational inference or Markov chain Monte Carlo, we could use the known distributions $p(\boldsymbol{x}|\boldsymbol{z}, \boldsymbol{\theta})$ and $p(\boldsymbol{z}, \boldsymbol{\theta}) = p(\boldsymbol{z}|\boldsymbol{\theta})p(\boldsymbol{\theta})$ to produce either a variational approximation or sample from $p(\boldsymbol{z}, \boldsymbol{\theta}|\boldsymbol{x})$. If the variational distribution is such

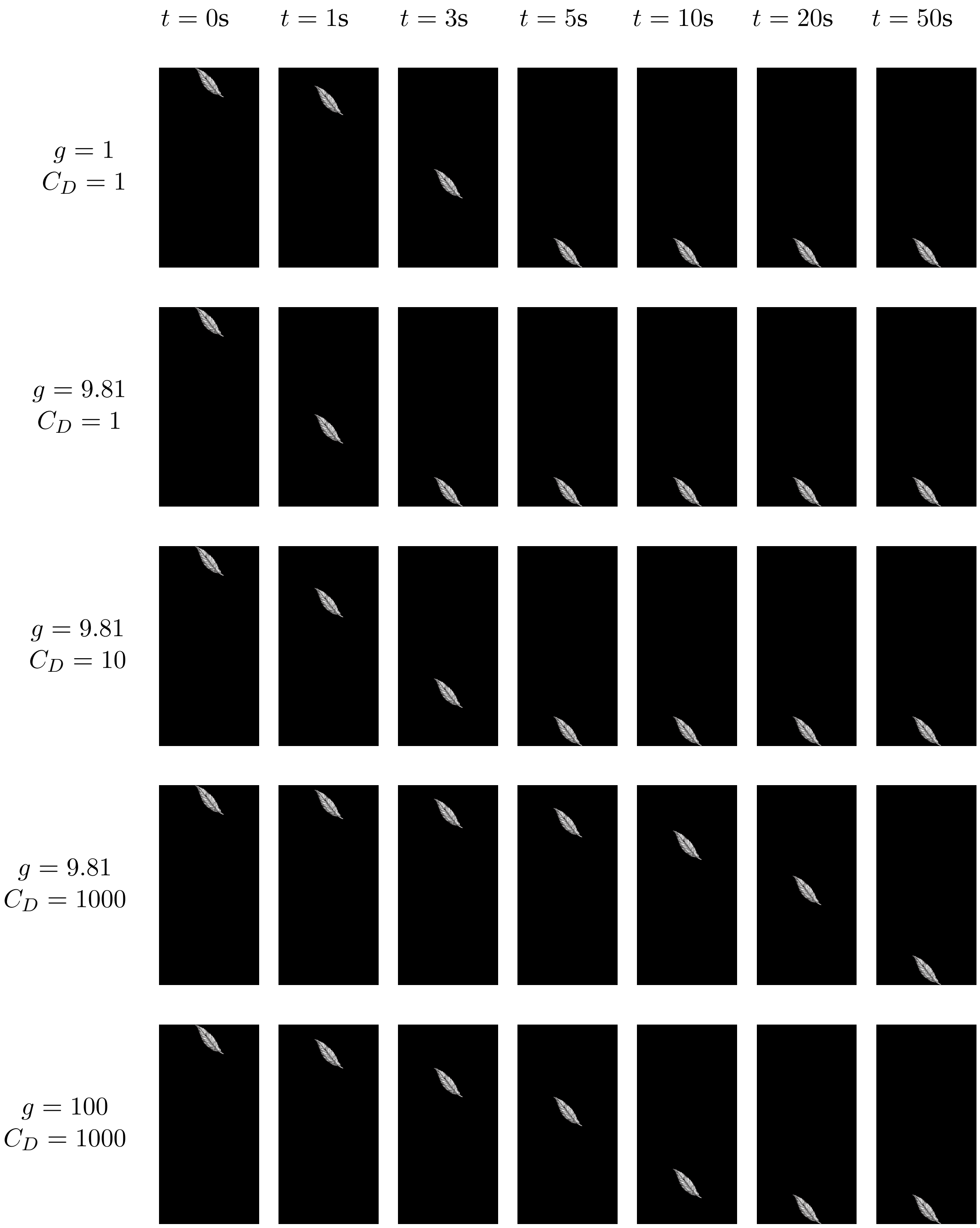

Figure 3.1: Example of feather falling simulations using various gravity and friction parameters. The feather is dropped from the same height in all those configurations, and the feather position is reported at various time steps.

that analytical marginalization is possible, the distribution $p(\boldsymbol{\theta}|\boldsymbol{x})$ can then be obtained through marginalization

$$p(\boldsymbol{\theta}|\boldsymbol{x}) = \int p(\boldsymbol{z}, \boldsymbol{\theta}|\boldsymbol{x}) \; d\boldsymbol{z}. \tag{3.2}$$

To obtain samples from the marginal, samples from $p(\boldsymbol{z}, \boldsymbol{\theta}|\boldsymbol{x})$ can be used where the $\boldsymbol{z}$ component is removed. However, to have a good enough simulator, it is usually necessary to have many internal latent variables. In the example of the feather falling, if we want

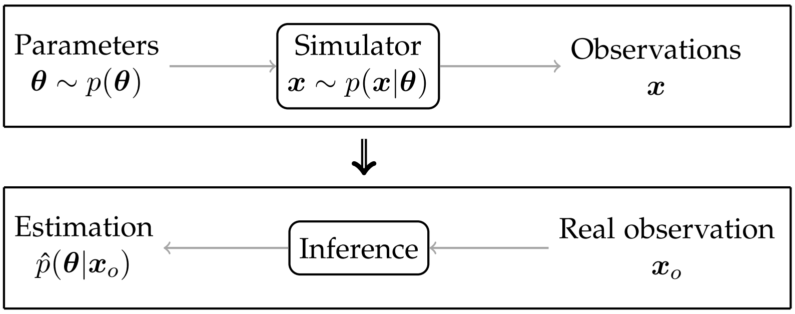
Parameters
$\boldsymbol{\theta} \sim p(\boldsymbol{\theta})$
Simulator
$\boldsymbol{x} \sim p(\boldsymbol{x}|\boldsymbol{\theta})$
Observations
$\boldsymbol{x}$
Estimation
$\hat{p}(\boldsymbol{\theta}|\boldsymbol{x}_o)$
Inference
Real observation
$\boldsymbol{x}_o$

Figure 3.2: Bayesian simulation-based inference overview

an accurate simulator, we need to choose Δt sufficiently small and hence perform many iterations and random rotations. This means that $\boldsymbol{z}$ is typically high-dimensional, and this is an issue to ensure good optimization with VI and good posterior landscape coverage with MCMC.

3.2 SIMULATION-BASED INFERENCE

To obtain more accurate inferences, we can use the fact that sampling from the likelihood $p(\boldsymbol{x}|\boldsymbol{\theta})$ is possible by sampling from $p(\boldsymbol{z}|\boldsymbol{\theta})$ and then from $p(\boldsymbol{x}|\boldsymbol{\theta}, \boldsymbol{z})$. In most settings, we are only interested in $p(\boldsymbol{\theta}|\boldsymbol{x})$, not in $p(\boldsymbol{\theta}, \boldsymbol{z}|\boldsymbol{x})$. Then, if we have inference methods that only rely on samples and not on likelihood evaluation, we can completely ignore the latent variables $\boldsymbol{z}$. This leads to the development of simulation-based inference (SBI) methods, also called likelihood-free inference (LFI), that only rely on simulations from the likelihood $p(\boldsymbol{x}|\boldsymbol{\theta})$ and not on likelihood evaluations. An overview of simulation-based inference in the Bayesian setting is provided in Figure 3.2.

A common simulation-based inference method uses histograms. This method is illustrated in Figure 3.3. The idea is to approximate a likelihood evaluation by building a histogram from samples from the likelihood. For any given configuration of parameters $\boldsymbol{\theta}_i$, a histogram can be constructed from samples $\boldsymbol{x} \sim p(\boldsymbol{x}|\boldsymbol{\theta}_i)$ to produce an approximation $\hat{p}(\boldsymbol{x}|\boldsymbol{\theta}_i)$. For a given observation $\boldsymbol{x}_o$. An approximate of the likelihood evaluated in $\boldsymbol{\theta}_i$ is then given by $\hat{p}(\boldsymbol{x}_o|\boldsymbol{\theta}_i)$. Histograms provide approximate likelihood evaluations and can hence be used in any likelihood-based inference procedure. They are usually used for the Neyman construction of confidence regions. It uses the link between hypothesis tests and confidence intervals. Parameters $\boldsymbol{\theta}_i$ are included in the confidence region if $\boldsymbol{x}_o$ is not in the rejection region of the test $H_0 : \boldsymbol{\theta} = \boldsymbol{\theta}_i, H_1 : \boldsymbol{\theta} \neq \boldsymbol{\theta}_i$. This rejection region can be approximated using the histograms $p(\boldsymbol{x}|\boldsymbol{\theta}_i)$. By repeating this procedure for many $\boldsymbol{\theta}_i$, an approximated confidence region can be computed. They are also often used to compute approximate likelihood ratios $\frac{\hat{p}(\boldsymbol{x}|\boldsymbol{\theta}_1)}{\hat{p}(\boldsymbol{x}|\boldsymbol{\theta}_2)}$ that is a key ingredient for hypothesis tests. In some cases, the observable $\boldsymbol{x}$ is high dimensional. Consequently, high-dimensional histograms must be constructed, and a too-large amount of samples might be needed to obtain good enough density approximations. To circumvent this issue, the observables can be replaced by low-

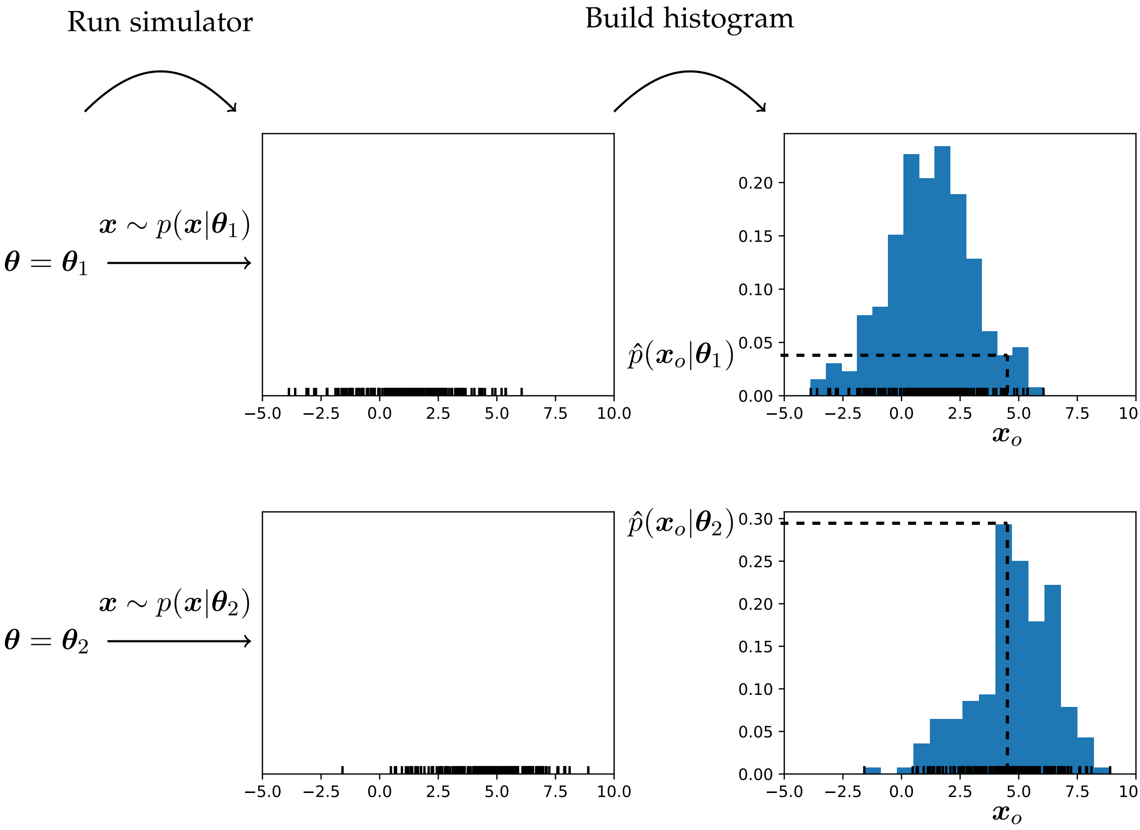

Figure 3.3: Simulation-based inference with histograms. For both configurations $\boldsymbol{\theta} = \boldsymbol{\theta}_1$ and $\boldsymbol{\theta} = \boldsymbol{\theta}_2$, samples are drawn from the simulator. Those samples are then used to construct histograms approximating the densities $p(\boldsymbol{x}_0|\boldsymbol{\theta}_1)$ and $p(\boldsymbol{x}_o|\boldsymbol{\theta}_2)$.

dimensional summary statistics that can either be handcrafted or learned. Histograms have been used, for example, for the discovery of the Higgs boson (ATLAS et al., 2012).

In the Bayesian setting, a popular method for simulation-based inference is Approximate Bayesian Computation (ABC) (Donald B. Rubin, 1984; Pritchard et al., 1999). The main idea is to perform simulations with many parameter values $\boldsymbol{\theta}$ sampled from the prior and to retain parameter values that lead to synthetic observations close to the real observation. In the limit where synthetic observations precisely equal to the real observation are retained, then the retained parameters are samples from the posterior. The algorithm in its most simple form is presented in Algorithm 3.2.

Many improvements can be made to the ABC algorithm. We provide a non-exhaustive list of those. See Beaumont (2019) for a more detailed review. If the observable $\boldsymbol{x}$ is high-dimensional, it can be compressed into automatically constructed summary statistics before being passed to the distance function (Fearnhead and Prangle, 2012). Beaumont, W. Zhang, and Balding (2002) give more weight to parameters that lead to synthetic observation closer to the real observation. Marjoram et al. (2003) extend MCMC algorithms to the simulation-based setting by replacing the likelihood-based acceptance rule with an acceptance rule computed from samples like in ABC. Toni and Stumpf (2009), Sisson, Fan, and Tanaka (2007), and Beaumont, Cornuet, et al. (2009) propose a sequential version of the

Algorithm 3.2 Approximate Bayesian Computation (ABC)

Inputs: distance function d, distance threshold ϵ, number of simulations n_sim and real observation $\boldsymbol{x}_o$.

Outputs: The retained parameters, approximately sampled from the posterior.

```
retained_parameters = List()
n_run_sim = 0
while n_run_sim < n_sim do
  $\boldsymbol{\theta} \sim p(\boldsymbol{\theta})$
  $\boldsymbol{x} \sim p(\boldsymbol{x}|\boldsymbol{\theta})$
  if $d(\boldsymbol{x}, \boldsymbol{x}_o) < \epsilon$ then
    retained_parameters.append($\boldsymbol{\theta}$)
  end if
  n_run_sim = n_run_sim + 1
end while
return retained_parameters
```

ABC algorithm. An initial population of parameter values is generated from the prior $p(\boldsymbol{\theta})$. The algorithm then iteratively refines that population to match the posterior distribution. At each step, synthetic observations are simulated using each of those parameter values. Parameter values that lead to the synthetic observations closest to the real observation are kept, and a new population is built by applying random perturbations to the kept parameter values. This procedure allows us to avoid doing simulations with parameters that are known to produce synthetic observations far from the real observation and to focus on more promising parts of the parameters space iteratively.

3.3 MACHINE LEARNING FOR SIMULATION-BASED INFERENCE

ABC and histogram-based techniques suffer from several limitations. The main limitation of histograms is the inability to scale. If the observation $\boldsymbol{x}$ is high dimensional, good summary statistics need to be constructed in order to use histograms. If the parameters $\boldsymbol{\theta}$ are high dimensional, then an unmanageable number of histograms are required to cover the parameter space. In its most simple form, ABC suffers from the same limitations as histograms. Combining sequential versions of ABC with the use of summary statistics alleviates those issues but makes the algorithm non-amortized, as opposed to amortized algorithms.

Definition 3.3.1. An amortized inference algorithm is an algorithm that, once run, allows to perform inference for any observation $\boldsymbol{x}$ without requiring heavy computations.

When using sequential versions of ABC, a population is iteratively built to match the posterior for a given observation. If we were to then perform inference on another observation, we would need to start over again from a fresh population sampled from the prior and reiterate with that new population. It would then require heavy computations as the previous computations cannot be reused. Sequential versions of ABC are hence non-amortized. In addition, handcrafted summary statistics might not contain all the information of the observation necessary to constrain the parameters. In such cases, both ABC and histograms do not yield the correct posterior asymptotically. While there exist methods for learning summary statistics, machine learning-based methods do not require the explicit construction of summary statistics to deal with high-dimensional observations.

Amortized algorithms are desired for several reasons. The first obvious scenario is when inference for observations generated in different settings is needed. If amortized, the algorithm can be run only once and not for each observation. A second scenario where amortized inference is desired is when real-time inference is required. For example, we took advantage of amortization to study gravitational waves emitted by the merge of two black holes, where real-time inference would allow to set up measurement systems to capture other types of signals from this event (Delaunoy, Wehenkel, et al., 2020). Finally, the most important advantage of amortized algorithms is that their quality can be evaluated by performing inferences on many test synthetic observations. This is of high importance to increase the trust that can be given to those approximation techniques.

Machine learning-based techniques can be designed to not suffer from the same limitations as ABC and histogram-based techniques. It allows the training of a model on several observations to then generalize to any observation and perform amortized inference. The development of neural network architectures that work with complex data such as images, time series, text, and graphs by providing appropriate inductive biases also allows the development of methods that can deal with high-dimensional observations and, to some extent, high-dimensional parameters. For those reasons, machine learning-based techniques are interesting for simulation-based inference but could also be of broad interest for statistical inference. In the remainder of this chapter, we will focus on machine learning-based methods.

NEURAL POSTERIOR AND LIKELIHOOD ESTIMATION Neural Posterior Estimation (NPE) refers to all the methods that aim to build a model approximating the posterior distribution $p(\boldsymbol{\theta}|\boldsymbol{x})$. To that end, a parametric conditional density estimator $\hat{p}_{\boldsymbol{\phi}}(\boldsymbol{\theta}|\boldsymbol{x})$ is trained, where $\boldsymbol{\phi}$ are the parameters of the density estimator, which is usually a neural network. This density estimator can be trained through variational inference to minimize expected KL divergence between the approximate and true posteriors.

$$\begin{aligned}
\boldsymbol{\phi}^* &= \arg\min_{\boldsymbol{\phi}} \mathbb{E}_{p(\boldsymbol{x})} \left[\mathrm{KL}\left[p(\boldsymbol{\theta}|\boldsymbol{x})||\hat{p}_{\boldsymbol{\phi}}(\boldsymbol{\theta}|\boldsymbol{x})\right]\right] && (3.3)\\
&= \arg\min_{\boldsymbol{\phi}} \mathbb{E}_{p(\boldsymbol{x})} \left[\mathbb{E}_{p(\boldsymbol{\theta}|\boldsymbol{x})}\left[\log \frac{p(\boldsymbol{\theta}|\boldsymbol{x})}{\hat{p}_{\boldsymbol{\phi}}(\boldsymbol{\theta}|\boldsymbol{x})}\right]\right] && (3.4)\\
&= \arg\min_{\boldsymbol{\phi}} \mathbb{E}_{p(\boldsymbol{\theta},\boldsymbol{x})} \left[\log \frac{p(\boldsymbol{\theta}|\boldsymbol{x})}{\hat{p}_{\boldsymbol{\phi}}(\boldsymbol{\theta}|\boldsymbol{x})}\right] && (3.5)\\
&= \arg\max_{\boldsymbol{\phi}} \mathbb{E}_{p(\boldsymbol{\theta},\boldsymbol{x})} \left[\log \hat{p}_{\boldsymbol{\phi}}(\boldsymbol{\theta}|\boldsymbol{x})\right] && (3.6)
\end{aligned}$$

This objective can be approximated using Monte-Carlo

$$\mathbb{E}_{p(\boldsymbol{\theta},\boldsymbol{x})}\left[\log \hat{p}_{\boldsymbol{\phi}}(\boldsymbol{\theta}|\boldsymbol{x})\right] \simeq \frac{1}{N}\sum_{i=1}^{N} \log \hat{p}_{\boldsymbol{\phi}}(\boldsymbol{\theta}_i|\boldsymbol{x}_i), \quad (\boldsymbol{\theta}_i, \boldsymbol{x}_i) \sim p(\boldsymbol{\theta}, \boldsymbol{x}). \tag{3.7}$$

Samples from the joint distribution $p(\boldsymbol{x}|\boldsymbol{\theta})$ can be obtained by first sampling parameters from the prior $p(\boldsymbol{\theta})$ and then observations using the simulator $p(\boldsymbol{x}|\boldsymbol{\theta})$. The density estimator can be of various nature. It can simply be a neural network that takes the observation $\boldsymbol{x}$ as input and outputs the parameters of a normal distribution conditioned on that observation $\boldsymbol{x}$. However, to have the flexibility to model complex distributions, normalizing flows (Dinh, Krueger, and Bengio, 2015; Rezende and Mohamed, 2015) are often used in practice. Normalizing flows are composed of a base probability distribution $q(\boldsymbol{z})$ and a neural network $f_{\boldsymbol{\phi}}$ defining a transformation from that base distribution to the target distribution. This distribution can be conditioned on $\boldsymbol{x}$ by giving it as input to the neural network performing the transformation. By the change of variable formula, the approximate posterior distribution is computed as

$$\hat{p}_{\boldsymbol{\phi}}(\boldsymbol{\theta}|\boldsymbol{x}) = q(f_{\boldsymbol{\phi}}^{-1}(\boldsymbol{\theta};\boldsymbol{x}))\left|\det\frac{\partial f_{\boldsymbol{\phi}}^{-1}(\boldsymbol{\theta};\boldsymbol{x})}{\partial \boldsymbol{\theta}}\right| \tag{3.8}$$

Note that the neural network-based transformation must be chosen such that the inverse $f_{\boldsymbol{\phi}}^{-1}(.;\boldsymbol{x})$ exists and the Jacobian $\left|\det\frac{\partial f_{\boldsymbol{\phi}}^{-1}(\boldsymbol{\theta};\boldsymbol{x})}{\partial \boldsymbol{\theta}}\right|$ can easily be computed. Normalizing flows then allow approximate posterior density evaluation, but it is also possible to sample parameters from that approximate posterior distribution. To do so, samples are first sampled from the base distribution $\boldsymbol{z} \sim q(\boldsymbol{z})$ and then transformed into samples from the target distribution $f_{\boldsymbol{\phi}}(\boldsymbol{z};\boldsymbol{x}) \sim \hat{p}_{\boldsymbol{\phi}}(\boldsymbol{\theta}|\boldsymbol{x})$.

Neural Likelihood Estimation (NLE) refers to all the methods that aim to build a model approximating the likelihood distribution $p(\boldsymbol{x}|\boldsymbol{\theta})$. Similarly to NPE, such models can be trained by minimizing the expected KL divergence

$$\begin{aligned}
\boldsymbol{\phi}^* &= \arg\min_{\boldsymbol{\phi}} \mathbb{E}_{p(\boldsymbol{\theta})}\left[\mathrm{KL}\left[p(\boldsymbol{x}|\boldsymbol{\theta})||\hat{p}_{\boldsymbol{\phi}}(\boldsymbol{x}|\boldsymbol{\theta})\right]\right] && (3.9)\\
&= \arg\min_{\boldsymbol{\phi}} \mathbb{E}_{p(\boldsymbol{\theta})}\left[\mathbb{E}_{p(\boldsymbol{x}|\boldsymbol{\theta})}\left[\log\frac{p(\boldsymbol{x}|\boldsymbol{\theta})}{\hat{p}_{\boldsymbol{\phi}}(\boldsymbol{x}|\boldsymbol{\theta})}\right]\right] && (3.10)\\
&= \arg\min_{\boldsymbol{\phi}} \mathbb{E}_{p(\boldsymbol{\theta},\boldsymbol{x})}\left[\log\frac{p(\boldsymbol{x}|\boldsymbol{\theta})}{\hat{p}_{\boldsymbol{\phi}}(\boldsymbol{x}|\boldsymbol{\theta})}\right] && (3.11)\\
&= \arg\max_{\boldsymbol{\phi}} \mathbb{E}_{p(\boldsymbol{\theta},\boldsymbol{x})}\left[\log \hat{p}_{\boldsymbol{\phi}}(\boldsymbol{x}|\boldsymbol{\theta})\right]. && (3.12)
\end{aligned}$$

The model $\hat{p}_{\boldsymbol{\phi}}(\boldsymbol{x}|\boldsymbol{\theta})$ can again be a normalizing flow or another density estimator. As NLE only provides the likelihood and not the posterior, likelihood-based inference techniques such as VI or MCMC must be applied using the approximate likelihood.

NEURAL RATIO ESTIMATION In the Bayesian setting, Neural Ratio Estimation (NRE) refers to the methods that aim to build a model for the ratio between two quantities that can then be used to approximate the posterior. The first version, often referred to as NRE-A or NRE (Hermans, Begy, and Louppe, 2020), builds a model for the likelihood to evidence ratio $r(\boldsymbol{\theta}, \boldsymbol{x}) = \frac{p(\boldsymbol{x}|\boldsymbol{\theta})}{p(\boldsymbol{x})}$. If we have access to an approximation $\hat{r}(\boldsymbol{\theta}, \boldsymbol{x})$, we can construct an approximate posterior density by multiplying it by the prior

$$\hat{p}(\boldsymbol{\theta}|\boldsymbol{x}) = \hat{r}(\boldsymbol{\theta}, \boldsymbol{x})p(\boldsymbol{\theta}), \tag{3.13}$$

because

$$p(\boldsymbol{\theta}|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\boldsymbol{x})} = r(\boldsymbol{\theta}, \boldsymbol{x})p(\boldsymbol{\theta}). \tag{3.14}$$

To obtain an approximation of the likelihood-to-evidence ratio, a classifier $\hat{d}(\boldsymbol{\theta}, \boldsymbol{x})$ is trained to distinguish pairs $(\boldsymbol{\theta}, \boldsymbol{x})$ sampled from the joint distribution $(\boldsymbol{\theta}, \boldsymbol{x}) \sim p(\boldsymbol{\theta}, \boldsymbol{x})$, with class label $y = 1$, from pairs sampled from the marginal distributions $(\boldsymbol{\theta}, \boldsymbol{x}) \sim p(\boldsymbol{\theta})p(\boldsymbol{x})$, with class label $y = 0$. It can be shown that the output of the optimal classifier trained with the binary cross-entropy loss is

$$d(\boldsymbol{\theta}, \boldsymbol{x}) = \frac{p(\boldsymbol{\theta}, \boldsymbol{x})}{p(\boldsymbol{\theta}, \boldsymbol{x}) + p(\boldsymbol{\theta})p(\boldsymbol{x})}. \tag{3.15}$$

Consequently,

$$\frac{d(\boldsymbol{\theta}, \boldsymbol{x})}{1 - d(\boldsymbol{\theta}, \boldsymbol{x})} = \frac{p(\boldsymbol{\theta}, \boldsymbol{x})}{p(\boldsymbol{\theta})p(\boldsymbol{x})} = \frac{p(\boldsymbol{x}|\boldsymbol{\theta})}{p(\boldsymbol{x})} = r(\boldsymbol{\theta}, \boldsymbol{x}). \tag{3.16}$$

An approximation of the likelihood-to-evidence ratio is then given by

$$\hat{r}(\boldsymbol{\theta}, \boldsymbol{x}) = \frac{\hat{d}(\boldsymbol{\theta}, \boldsymbol{x})}{1 - \hat{d}(\boldsymbol{\theta}, \boldsymbol{x})} = \exp(\sigma^{-1}(\hat{d}(\boldsymbol{\theta}, \boldsymbol{x}))), \tag{3.17}$$

where σ is the sigmoid function. To avoid numerically unstable computations, the approximate log posterior can be computed as

$$\log(\hat{p}(\boldsymbol{\theta}|\boldsymbol{x})) = \log(\hat{r}(\boldsymbol{\theta}, \boldsymbol{x})) + \log(p(\boldsymbol{\theta})) = \sigma^{-1}(\hat{d}(\boldsymbol{\theta}, \boldsymbol{x})) + \log(p(\boldsymbol{\theta})), \tag{3.18}$$

where $\sigma^{-1}(\hat{d}(\boldsymbol{\theta}, \boldsymbol{x}))$ can be obtained by evaluating the neural network without the final sigmoid activation layer. This formulation allows us to evaluate the approximate log posterior density. To obtain samples from this distribution, running an additional MCMC algorithm is required.

NRE-B (Durkan, Murray, and Papamakarios, 2020) employs a contrastive learning scheme by considering several samples from the product of marginal distributions. They build a model $\hat{f}(\boldsymbol{\theta}, \boldsymbol{x})$ trained on the following contrastive loss

$$L(\hat{f}) = -\frac{1}{B}\sum_{b=1}^{B} \log \frac{\exp(\hat{f}(\boldsymbol{\theta}^{(b)}, \boldsymbol{x}^{(b)}))}{\sum_{k=1}^{K} \exp(\hat{f}(\boldsymbol{\theta}^{(k)}, \boldsymbol{x}^{(b)}))}, \tag{3.19}$$

where $(\boldsymbol{\theta}^{(b)}, \boldsymbol{x}^{(b)})$ are sampled from the joint distribution $p(\boldsymbol{\theta}, \boldsymbol{x})$ and the K contrastive parameters $\boldsymbol{\theta}^{(k)}$ are each time composed of $\boldsymbol{\theta}^{(b)}$ and $K-1$ parameters set sampled independently from the prior $p(\boldsymbol{\theta})$. When trained at the optimum, the output of the model is

$$f(\boldsymbol{\theta}, \boldsymbol{x}) = \log(p(\boldsymbol{\theta}|\boldsymbol{x})) - \log(p(\boldsymbol{\theta})) + C(\boldsymbol{x}), \tag{3.20}$$

where $C(\boldsymbol{x})$ is an unknown constant. We can, therefore, recover an approximation for the unnormalized posterior density

$$\log \hat{p}(\boldsymbol{\theta}|\boldsymbol{x}) C(\boldsymbol{x}) = \hat{f}(\boldsymbol{\theta}, \boldsymbol{x}) + \log(p(\boldsymbol{\theta})). \tag{3.21}$$

NRE-C (Benjamin K Miller, Weniger, and Forré, 2022) aim to improve over NRE-B by removing the unknown constant. They do that by sampling all the pairs from the contrastive set independently from $\boldsymbol{x}$ with probability p_0 instead of always having one of the parameter sets sampled jointly with $\boldsymbol{x}$. They train their model on the following loss function

$$\begin{aligned} L(\hat{f}) = &- \frac{1}{B}(1-p_0) \sum_{b=1}^{B} \log \frac{\frac{1-p_0}{p_0} \exp(\hat{f}(\boldsymbol{\theta}^{(b)}, \boldsymbol{x}^{(b)}))}{K + \frac{1-p_0}{p_0} \sum_{k=1}^{K} \exp(\hat{f}(\boldsymbol{\theta}^{(k)} \boldsymbol{x}^{(b)}))} \\ &- \frac{1}{B} p_0 \sum_{b=B+1}^{2B} \log \frac{K}{K + \frac{1-p_0}{p_0} \sum_{k=1}^{K} \exp(\hat{f}(\boldsymbol{\theta}^{(k)}, \boldsymbol{x}^{(b)}))}, \end{aligned} \tag{3.22}$$

, where $(\boldsymbol{\theta}^{(b)}, \boldsymbol{x}^{(b)}) \sim p(\boldsymbol{\theta}, \boldsymbol{x})$ for $b = (1, ..., B)$ and $(\boldsymbol{\theta}^{(b)}, \boldsymbol{x}^{(b)}) \sim p(\boldsymbol{\theta})p(\boldsymbol{x})$ for $b = (B+1, ..., 2B)$. Again, the K contrastive parameters $\boldsymbol{\theta}^{(k)}$ are each time composed of $\boldsymbol{\theta}^{(b)}$ and $K-1$ parameters set sampled independently from the prior $p(\boldsymbol{\theta})$. In this case, when trained at the optimum, the output of the model is

$$f(\boldsymbol{\theta}, \boldsymbol{x}) = \log(p(\boldsymbol{\theta}|\boldsymbol{x})) - \log(p(\boldsymbol{\theta})). \tag{3.23}$$

We can recover an approximation for the normalized posterior density

$$\log \hat{p}(\boldsymbol{\theta}|\boldsymbol{x}) = \hat{f}(\boldsymbol{\theta}, \boldsymbol{x}) + \log(p(\boldsymbol{\theta})). \tag{3.24}$$

Let us note that NRE-A is a special case of NRE-C when $p_0 = 1/2$ and $K = 1$. In the limit of $p_0 \to 0$, NRE-C approaches NRE-B.

In the frequentist setting, K. Cranmer, Pavez, and Louppe (2015) trains a classifier to approximate likelihood ratios. Those can then be used for likelihood ratio hypothesis tests or to construct confidence intervals using Wilk's theorem or through a Neyman construction (ATLAS et al., 2024). Brehmer, Mishra-Sharma, et al. (2019) show how to use the joint score $\nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{x}, \boldsymbol{z}|\boldsymbol{\theta})$, where $\boldsymbol{z}$ are the latent variables to improve the inference quality. Stoye et al. (2018) present an improved cross-entropy estimator to train the classifier. Dalmasso, Izbicki, and Lee (2020) show how other statistics could be used and present a framework for constructing confidence sets with bounded error and performing diagnostics.

NEURAL POSTERIOR SCORE ESTIMATION Neural Posterior Score Estimation (NPSE) (Geffner et al., 2022; Sharrock et al., 2022; Linhart, Cardoso, et al., 2024) aim to compute a diffusion model score such that produced samples are approximately sampled from the posterior. As illustrated in Figure 3.4, diffusion models aim to inverse a diffusion process, also called a forward process. The diffusion process takes a sample from the target distribution as input and iteratively adds noise to it until there is only noise left. The inverse process, also called the backward process, takes a sample from the noise distribution as input and iteratively denoises it to obtain a sample from the target distribution. In our case, the target distribution is the posterior $p(\boldsymbol{\theta}|\boldsymbol{x})$. The denoising process uses a score function $\nabla_{\boldsymbol{\theta}} \log p_{T-t}(\bar{\boldsymbol{\theta}}|\boldsymbol{x})$ to control how the samples are denoised. To obtain samples from the approximate posterior $\hat{p}(\boldsymbol{\theta}|\boldsymbol{x})$, an approximation of the score is learned $s(\boldsymbol{\theta}_t, \boldsymbol{x}, t)$ and this score is used to denoise samples drawn from the noise distribution.

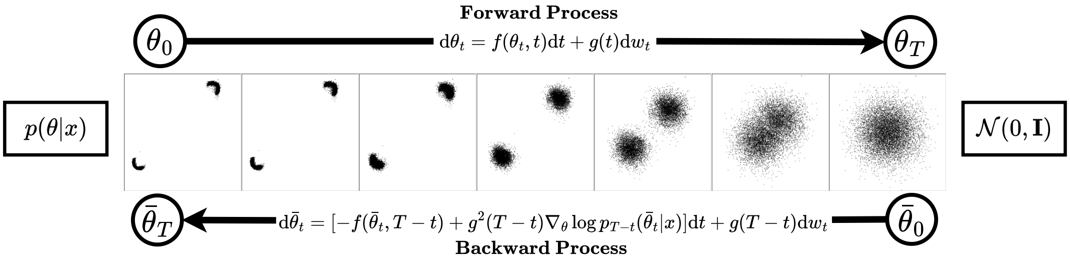
Forward Process
θ_0
$\mathrm{d}\theta_t = f(\theta_t, t)\mathrm{d}t + g(t)\mathrm{d}w_t$
θ_T
$p(\theta|x)$
$\mathcal{N}(0, \mathbf{I})$
$\bar{\theta}_T$
$\mathrm{d}\bar{\theta}_t = [-f(\bar{\theta}_t, T-t) + g^2(T-t)\nabla_\theta \log p_{T-t}(\bar{\theta}_t|x)]\mathrm{d}t + g(T-t)\mathrm{d}w_t$
$\bar{\theta}_0$
Backward Process

Figure 3.4: Visualization of a diffusion model applied on the two moons simulation-based inference benchmark. A backward process is learned to inverse a forward diffusion process. Applying the backward process to noise can then recover samples from the posterior distribution. Figure reproduced from Sharrock et al. (2022).

Diffusion techniques have some advantages over NPE techniques. First, diffusion models work well at modeling high-dimensional distributions. Those models are then useful to obtain high-quality approximations when $\boldsymbol{\theta}$ is high-dimensional. Second, the score of multiple observations can be expressed as a composition of scores for a single observation (Geffner et al., 2022; Linhart, Cardoso, et al., 2024). The key observation is that

$$p(\boldsymbol{\theta}|\boldsymbol{x}_1, ..., \boldsymbol{x}_n) \propto p(\boldsymbol{\theta})^{1-n} \prod_{i=1}^{n} p(\boldsymbol{\theta}|\boldsymbol{x}_i). \tag{3.25}$$

Geffner et al. (2022) define the forward diffusion process such that the intermediate densities are

$$p_t^f(\boldsymbol{\theta}|\boldsymbol{x}_1, ..., \boldsymbol{x}_n) \propto (p(\boldsymbol{\theta})^{1-n})^{\frac{T-t}{T}} \prod_{i=1}^{n} p_t(\boldsymbol{\theta}|\boldsymbol{x}_i), \tag{3.26}$$

where $p_t(\boldsymbol{\theta}|\boldsymbol{x}_i)$ is the distribution obtained after applying progressively some Gaussian noise for a time t. The score is then expressed

$$\nabla_{\boldsymbol{\theta}} \log p_t^f(\boldsymbol{\theta}|\boldsymbol{x}_1, ..., \boldsymbol{x}_n) = \frac{(1-n)(T-t)}{T} \nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{\theta}) + \sum_{i=1}^{n} \nabla_{\boldsymbol{\theta}} \log p_t(\boldsymbol{\theta}|\boldsymbol{x}_i). \tag{3.27}$$

As a consequence, score-based models trained on single observations could be used to approximate the posterior for any number of conditioning observations without having to retrain a model for each case. This method has the drawback of requiring Langevin dynamics to run the denoising process. This is due to the way intermediate densities are defined. To improve on that aspect, Linhart, Cardoso, et al. (2024) use the following diffusion process

$$\boldsymbol{\theta}_t = \sqrt{\alpha_t}\boldsymbol{\theta}_0 + \sqrt{1-\alpha_t}\epsilon_t, \tag{3.28}$$

where α_t is a parameter controlling noise intensity at different stages of the diffusion process, and ϵ_t are sampled from a Normal distribution. In this, case the score can be written as

$$\nabla_{\boldsymbol{\theta}} \log p_t(\boldsymbol{\theta}|\boldsymbol{x}_1,...,\boldsymbol{x}_n) = (1-n)\nabla_{\boldsymbol{\theta}} \log p_t^{\lambda}(\boldsymbol{\theta}) + \sum_{i=1}^{n} \nabla_{\boldsymbol{\theta}} \log p_t(\boldsymbol{\theta}|\boldsymbol{x}_i) + \nabla_{\boldsymbol{\theta}} \log L_{\lambda}(\boldsymbol{\theta}, \boldsymbol{x}1,...,\boldsymbol{x}_n), \tag{3.29}$$

where, λ is the prior, p_t^{λ} is the diffused prior and $\nabla_{\boldsymbol{\theta}} \log L_{\lambda}(\boldsymbol{\theta}, \boldsymbol{x}1,...,\boldsymbol{x}_n)$ is a term that needs to be approximated.

Gloeckler, Toyota, et al. (2024) showed that this compositional property of scores can be used when dealing with arbitrary length time series that satisfies the Markov property. Under that hypothesis, an arbitrary length time series can be viewed as a collection of one-step transitions of arbitrary size. The score for multiple transitions can then be recovered from the score for a single transition.

The main drawback of score-based methods compared to NPE with normalizing flows is that the denoising process only provides a way for drawing samples from an approximate posterior distribution but does not allow for approximate posterior density evaluation. This density can be computed by running the corresponding probability flow ordinary differential equation, but this has to be approximated using many neural network calls (Song et al., 2020).

ILLUSTRATIVE EXAMPLE We apply the NPE, NRE, NLE, and NPSE algorithms on the feather falling simulator to infer both the gravitational constant g and the drag coefficient C_d. We consider the same observations as in Chapter 2, being measured falling time of 4.68s, 6.97s and 9.69s for heights 3m, 5m and 7m respectively. Results are shown in Figure 3.5. Density plots are provided for NPE and NRE as those two methods allow for posterior density evaluation. For NLE and NPSE, 500 samples are drawn from the posterior by running MCMC for NLE and diffusion for NPSE. From those plots, we can see that inferring the product of g and C_d seems to be easy, but it is hard to distinguish between both g and C_d being low or both being high. All methods provide similar results, but some visible differences remain. All methods are trained on the same $10,000$ samples from the simulator $(\boldsymbol{\theta}, \boldsymbol{x}) \sim p(\boldsymbol{x}|\boldsymbol{\theta})p(\boldsymbol{\theta})$.

SEQUENTIAL MACHINE LEARNING TECHNIQUES Sequential methods aim to reduce the amount of computations required when we are interested in the posterior for a single given observation $\boldsymbol{x}_o$. In such cases, we aim to construct an approximation of the posterior

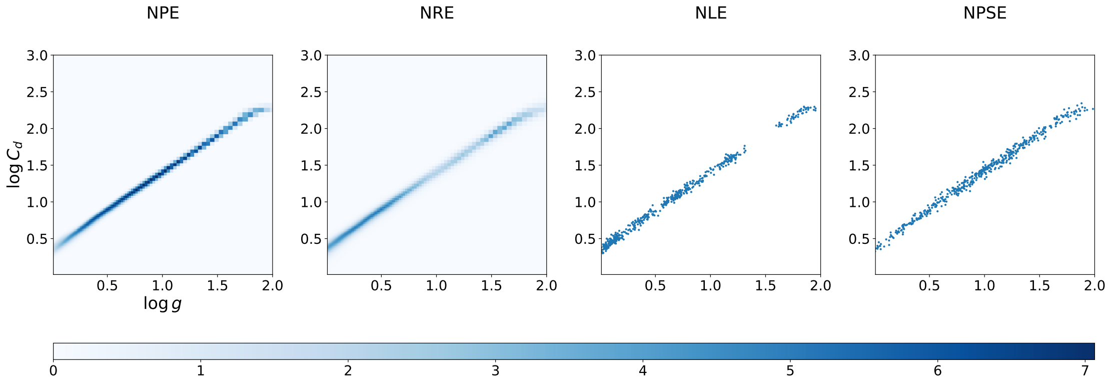

Figure 3.5: Approximate posteriors for the feather falling simulator with measured falling time of 4.68s, 6.97s and 9.69s for heights 3m, 5m and 7m respectively.

$\hat{p}(\boldsymbol{\theta}|\boldsymbol{x})$ that is close to the true posterior when $\boldsymbol{x} = \boldsymbol{x}_o$ but can be a poor approximation when $\boldsymbol{x} \neq \boldsymbol{x}_0$. To efficiently construct this approximation, the main idea is to favor sampling in regions likely to provide observations close to $\boldsymbol{x}_0$ in order for the model to be trained on many samples in the region of interest. To that end, the model training procedure operates in rounds where, at each round, the previously obtained approximate posterior is used to construct a proposal distribution $\tilde{p}(\boldsymbol{\theta})$ that should lead to simulations close to $\boldsymbol{x}_0$. In each round, a new training set is then sampled by drawing parameters $\boldsymbol{\theta}$ from the proposal $\tilde{p}(\boldsymbol{\theta})$ and observations using the simulator $p(\boldsymbol{x}|\boldsymbol{\theta})$. An overview of sequential algorithms is shown in Algorithm 3.3.

Algorithm 3.3 Sequential simulation-based inference

```
Inputs:          A prior $p(\boldsymbol{\theta})$, a simulator $p(\boldsymbol{x}|\boldsymbol{\theta})$ and a target $\boldsymbol{x}_o$
Outputs:         An approximate posterior $\hat{p}(\boldsymbol{\theta}|\boldsymbol{x})$ with focused training on $\boldsymbol{x}_o$
Hyperparameter:  Number rounds n_rounds and number of samples per round n_samples

$\tilde{p}(\boldsymbol{\theta}) = p(\boldsymbol{\theta})$
current_round = 0
while current_round < n_rounds do
    dataset = List()
    current_sample = 0
    while current_sample < n_samples do
        $\boldsymbol{\theta} \sim \tilde{p}(\boldsymbol{\theta})$
        $\boldsymbol{x} \sim p(\boldsymbol{x}|\boldsymbol{\theta})$
        dataset.append($\boldsymbol{x}, \boldsymbol{\theta}$)
        current_sample = current_sample + 1
    end while
    $\hat{p}(\boldsymbol{\theta}|\boldsymbol{x})$ = train(dataset, $\tilde{p}(\boldsymbol{\theta})$)
    $\tilde{p}(\boldsymbol{\theta}) = \hat{p}(\boldsymbol{\theta}|\boldsymbol{x} = \boldsymbol{x}_o)$
    current_round = current_round + 1
end while
return $\hat{p}(\boldsymbol{\theta}|\boldsymbol{x})$
```

Sequential training can be performed in combination with neural likelihood estimation (Papamakarios, Sterratt, and Murray, 2019), neural ratio estimation (Hermans, Begy, and

Louppe, 2020; Durkan, Murray, and Papamakarios, 2020; Benjamin K Miller, Weniger, and Forré, 2022), neural posterior estimation (Papamakarios and Murray, 2016; Lueckmann, Goncalves, et al., 2017; Greenberg et al., 2019) and neural posterior score estimation (Sharrock et al., 2022). When directly approximating posteriors or posterior scores, the training procedure should take into account the fact that the parameters $\boldsymbol{\theta}$ are not sampled from the prior $p(\boldsymbol{\theta})$ but from the proposal $\tilde{p}(\boldsymbol{\theta})$. For neural posterior estimation, this can be taken into account by reweighting the approximate posterior after the training procedure (Papamakarios and Murray, 2016), reweighting the importance of each sample in the loss (Lueckmann, Goncalves, et al., 2017) or training the model with a contrastive learning scheme (Greenberg et al., 2019). Sharrock et al. (2022) adapts those methods to the case of posterior scores.

The major drawback of sequential machine learning methods is that, as the sequential versions of the ABC algorithm, they are not amortized. This means that the trained model cannot be used to perform inference on observations $\boldsymbol{x} \neq \boldsymbol{x}_o$ as the model has not been trained on samples from that part of the observation space. Consequently, the quality of the model cannot be evaluated by performing inference on several synthetic test observations. In an attempt to get the best of both worlds, Benjamin K Miller, Cole, et al. (2021) introduced truncated marginal neural ratio estimation. The key idea is to train the model to be valid for a small subspace around $\boldsymbol{x}_o$ generated by a truncated version of the prior. The training is then still faster than training on the full observation space, and the resulting model is what they call locally amortized. The model can then be tested locally on synthetic observations generated using parameters drawn from the truncated prior. This idea has then been adapted to neural posterior estimation (Deistler et al., 2022) and neural posterior score estimation (Sharrock et al., 2022).

ADDITIONAL CONTRIBUTIONS TO SBI In this paragraph, we non-exhaustively provide short descriptions of some other pieces of work from the SBI literature. We reviewed above the use of normalizing flows for likelihood and posterior estimation, classifiers for likelihood-to-evidence ratio estimation, and the use of posterior score estimation models. Other generative models can be used. Ramesh et al. (2022) and Pacchiardi and Dutta (2022a) explore the use of generative adversarial networks. Glaser et al. (2022) propose to use energy-based models, Schmitt, Pratz, et al. (2023) explore the use of consistency models, and Wildberger et al. (2024) use flow matching. Häggström et al. (2024) use an inverse regression approach via Gaussian locally linear mappings that has the benefit of being lightweight compared to neural network approaches.

Glöckler et al. (2021), Radev et al. (2023), Schmitt, Ivanova, et al. (2024) and Häggström et al. (2024) consider jointly approximating the posterior and likelihood. Glöckler et al. (2021) leverage both approximations to avoid the need for proposal corrections in sequential methods and still enable direct posterior estimation. Radev et al. (2023) show that having both approximations allows one to run new types of tests to verify the quality of the approximations. Schmitt, Ivanova, et al. (2024) show that jointly training likelihood and posterior approximation by enforcing consistency between each other improves approximation quality in low-data regimes. Häggström et al. (2024) show that an inverse regression approach via Gaussian local linear mapping provides both the posterior and likelihood.

P. L. Rodrigues et al. (2021) and Heinrich et al. (2023) apply simulation-based inference to hierarchical Bayesian models. In those settings, both local and global parameters need to be inferred, where local parameters affect individual events and global parameters are shared by a set of events. P. L. Rodrigues et al. (2021) consider the Bayesian case and do this by learning an approximation for the posterior over global parameters given a set of observations and an approximation for the posterior over local parameters given the local observation and global parameters. Heinrich et al. (2023) adapts this methodology to frequentist inference.

Rozet and Louppe (2021a), Benjamin K Miller, Cole, et al. (2021), Rozet and Louppe (2023), and Gloeckler, Deistler, Weilbach, et al. (2024) aim to improve the flexibility of simulation-based inference methods by training models that can yield multiple distribution approximations. Rozet and Louppe (2021a) train a neural ratio estimation model that can yield any parameter marginals on the fly. Benjamin K Miller, Cole, et al. (2021) shows that using truncated sequential algorithms allows the training of an additional model for any parameter marginals using the truncated prior from the last round without restarting the sequential procedure from scratch. Gloeckler, Deistler, Weilbach, et al. (2024) use diffusion model with the transformer architecture to construct a model that can be queried to sample from all conditionals of the joint distribution $p(\boldsymbol{\theta}, \boldsymbol{x})$. Rozet and Louppe (2023) consider the problem of data assimilation where state trajectories are inferred from arbitrary observations.

3.4 diagnosing simulation-based inference

We reviewed how different quantities, such as the posterior, likelihood, likelihood-to-evidence ratio, and posterior scores, could be approximated using samples from a simulator. Those approximations can then be used to make hypothesis tests or construct credible or confidence regions. However, those approximations can be of poor quality for many reasons. If the simulator is complex, many samples might be required to capture its behavior fully. If too few samples are used due to computational constraints, then the trained model might be of poor quality. Poor quality approximations could also happen if the machine learning model is not flexible enough or poorly optimized. For all those reasons, the quality of the obtained approximations should always be checked before relying on it in downstream tasks.

The first thing to do is visually inspect the obtained posteriors on some test simulations for which the simulator's parameters are known. We sample parameters from the prior $\boldsymbol{\theta}^* \sim p(\boldsymbol{\theta})$ and then synthetic observations from the simulator using those parameters $\boldsymbol{x}^* \sim$

$p(\boldsymbol{x}|\boldsymbol{\theta}^*)$. We plot in Figure 3.6 the approximates posteriors $\hat{p}(\boldsymbol{\theta}|\boldsymbol{x}^*)$ and the parameters $\boldsymbol{\theta}^*$ used to simulate the observations $\boldsymbol{x}^*$ for the feather falling problem. We observe that there is usually a high approximate posterior mass $\hat{p}(\boldsymbol{\theta}|\boldsymbol{x}^*)$ around the parameters $\boldsymbol{\theta}^*$, indicating that the approximations do not look too bad.

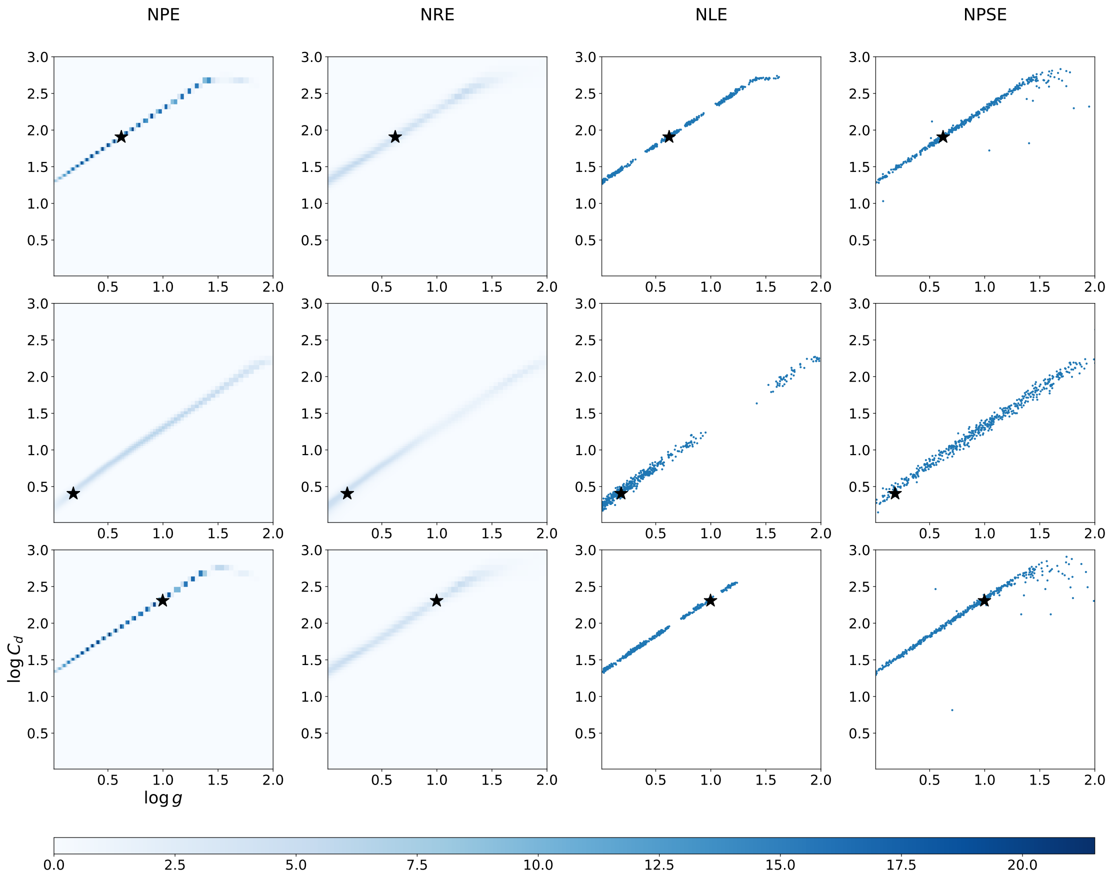

Figure 3.6: Visual inspection of the posterior quality. Either the approximate posterior density $\hat{p}(\boldsymbol{\theta}|\boldsymbol{x})$ or samples from it are shown. The black star represents the parameters $\boldsymbol{\theta}^*$ that were used to generate the synthetic test observation.

The validity of an approximation can also be tested by performing a hypothesis test of the form

- $H_0 : \hat{p}(\boldsymbol{\theta}|\boldsymbol{x}) = p(\boldsymbol{\theta}|\boldsymbol{x}), \ \forall \boldsymbol{\theta}, \boldsymbol{x}$
- $H_1 : \exists \ \boldsymbol{\theta}, \boldsymbol{x} : \hat{p}(\boldsymbol{\theta}|\boldsymbol{x}) \neq p(\boldsymbol{\theta}|\boldsymbol{x})$.

The Simulation-Based Calibration (SBC) algorithm (Talts et al., 2018) constructs a histogram of ranks of samples from the true posterior compared to L samples from the approximate posterior. Ranks are constructed as follow:

$$\begin{aligned}
\boldsymbol{\theta}^* &\sim p(\boldsymbol{\theta}) \\
\boldsymbol{x}^* &\sim p(\boldsymbol{x}|\boldsymbol{\theta}^*) \\
\{\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_L\} &\sim \hat{p}(\boldsymbol{\theta}|\boldsymbol{x}^*) \\
\text{rank} &= \sum_{i=1}^{L} \mathbb{1}[f(\boldsymbol{\theta}_i) < f(\boldsymbol{\theta}^*)],
\end{aligned}$$

where f is a function compressing parameters $\boldsymbol{\theta}$ into a single real value. Ranks are then integer in $[0, L]$. It should be noted that sampling $\boldsymbol{\theta}^* \sim p(\boldsymbol{\theta}), \boldsymbol{x}^* \sim p(\boldsymbol{x}|\boldsymbol{\theta}^*)$ is equivalent to $\boldsymbol{x}^* \sim p(\boldsymbol{x}), \boldsymbol{\theta}^* \sim p(\boldsymbol{\theta}|\boldsymbol{x}^*)$. The algorithm then compares a sample from $p(\boldsymbol{\theta}|\boldsymbol{x}^*)$ to L samples from $\hat{p}(\boldsymbol{\theta}|\boldsymbol{x}^*)$. If $\hat{p}(\boldsymbol{\theta}|\boldsymbol{x}) = p(\boldsymbol{\theta}|\boldsymbol{x}),\ \forall \boldsymbol{\theta}, \boldsymbol{x}$, then the ranks should be distributed uniformly in $[0, L]$ as the two distributions are the same. If the histogram of ranks obtained empirically is too far from a uniform distribution, we can then conclude that H_0 can be rejected and that the approximation is not perfect.

By running the SBC diagnostic for the different tested methods on the feather falling simulator, we obtain p-values of 0.000033 for NPE, 0.22 for NRE, 0.0086 for NLE and 0.0087 for NPSE. This means that we could reject the null hypothesis at a level $\alpha = 0.01$ for all the approximations but the one obtained with NRE. Note that it does not mean that NRE's approximation is trustworthy but only that the others are not. Being able to reject 3 out of the 4 approximations highlights the fact that approximations should not always be trusted and that careful analysis of their quality is required. For this analysis, 1000 samples $(\boldsymbol{\theta}^*, \boldsymbol{x}^*) \sim p(\boldsymbol{\theta}, \boldsymbol{x})$ were used, and for each observation $\boldsymbol{x}^*$, 100 samples were drawn from the approximate posterior. Parameters are compressed into a single real value through the function $f : g, C_d \rightarrow g \times C_d$.

The "Tests of Accuracy with Random Points" (TARP) algorithm (Lemos, Coogan, et al., 2023) considers the special case of f being the distance to a random reference point. They show that, in this case, asymptotic uniformity of the histogram implies that the posterior approximation is perfect. This is then not just a necessary condition but also a sufficient condition.

Linhart, Gramfort, and P. L. Rodrigues (2022) consider the special case of $\hat{p}(\boldsymbol{\theta}|\boldsymbol{x})$ being modeled by a normalizing flow with a normal distribution as base distribution. Instead of comparing the approximate and true posteriors in the space of parameters $\boldsymbol{\theta}$. They project those in the space of the base distribution of the normalizing flow by using the transformation defined by this flow. In this space, the approximate posterior is a normal distribution by definition. We can obtain samples from the true posterior in this space by applying the normalizing flow transformation on those, and they can then be compared to a normal distribution. They use the fact that the c.d.f. of a normal variable can be computed analytically to compute the c.d.f. for each dimension of the projected samples of the true posterior. If the approximation of the posterior is perfect, then those computed c.d.f. should be uniformly distributed, and all the dimensions should be independent of each other. Those two elements are then tested.

When the true posterior distribution is known, e.g., on toy problems for benchmarking purposes, the classifier two samples test (C2ST) can be used (Lopez-Paz and Oquab, 2017). It consists in training a classifier to discriminate between samples $\boldsymbol{\theta} \sim \hat{p}(\boldsymbol{\theta}|\boldsymbol{x})$ and $\boldsymbol{\theta} \sim p(\boldsymbol{\theta}|\boldsymbol{x})$. If the approximate posterior is equal to the true posterior, then even a perfectly trained classifier should have an accuracy of 50%. If we assume that a sufficient amount of samples are used to assess the accuracy and that the accuracy is not 50%, then we can conclude that the approximation is not perfect. Linhart, Gramfort, and P. Rodrigues (2024) extends this algorithm to only require samples from the joint distribution $p(\boldsymbol{\theta}, \boldsymbol{x})$ and hence be applicable in a general simulation-based inference setting. In this setting, the classifier takes both observations $\boldsymbol{x}$ and parameters $\boldsymbol{\theta}$ as input and is trained to discriminate between $\boldsymbol{\theta}, \boldsymbol{x} \sim p(\boldsymbol{\theta}, \boldsymbol{x})$ and $\boldsymbol{\theta}, \boldsymbol{x} \sim \hat{p}(\boldsymbol{\theta}|\boldsymbol{x})p(\boldsymbol{x})$. The deviation of the output of a classifier trained in such a way, from uniformity between both classes can then be used as a test statistic. Critical values for this test can be computed empirically by training classifiers on identical distributions, obtained by merging the samples from each distribution and randomly assigning a class to each sample.

Hypothesis tests provide a way to potentially reject the fact that the approximation is perfect. However, it should be kept in mind that failing to reject the null hypothesis does not necessarily mean that it is likely valid. Consequently, those tests do not guarantee the validity of the approximation. Moreover, the p-values associated with such tests should not be interpreted as distances between the true posterior and the approximation. To obtain a notion of distance, one could consider approximating the expected KL divergence between the true and approximate posteriors.

$$\begin{aligned}
\mathbb{E}_{p(\boldsymbol{x})}\left[\mathrm{KL}\left[p(\boldsymbol{\theta}|\boldsymbol{x})||\hat{p}(\boldsymbol{\theta}|\boldsymbol{x})\right]\right] &= \mathbb{E}_{p(\boldsymbol{x})}\left[\mathbb{E}_{p(\boldsymbol{\theta}|\boldsymbol{x})}\left[\log \frac{p(\boldsymbol{\theta}|\boldsymbol{x})}{\hat{p}(\boldsymbol{\theta}|\boldsymbol{x})}\right]\right] && (3.30)\\
&= \mathbb{E}_{p(\boldsymbol{x},\boldsymbol{\theta})}\left[\log \frac{p(\boldsymbol{\theta}|\boldsymbol{x})}{\hat{p}(\boldsymbol{\theta}|\boldsymbol{x})}\right] && (3.31)\\
&\propto -\mathbb{E}_{p(\boldsymbol{x},\boldsymbol{\theta})}\left[\log \hat{p}(\boldsymbol{\theta}|\boldsymbol{x})\right] && (3.32)\\
&\simeq -\frac{1}{N}\sum_{i=1}^{N} \log \hat{p}(\boldsymbol{\theta}_i|\boldsymbol{x}_i), \quad (\boldsymbol{\theta}_i, \boldsymbol{x}_i) \sim p(\boldsymbol{\theta}, \boldsymbol{x}). && (3.33)
\end{aligned}$$

While $\mathbb{E}_{p(\boldsymbol{x},\boldsymbol{\theta})}\left[\log \hat{p}(\boldsymbol{\theta}|\boldsymbol{x})\right]$ values are hard to interpret, we can conclude that an approximation $\hat{p}_1(\boldsymbol{\theta}|\boldsymbol{x})$ is better on average than an approximation $\hat{p}_2(\boldsymbol{\theta}|\boldsymbol{x})$, according to the KL divergence, if $\mathbb{E}_{p(\boldsymbol{x},\boldsymbol{\theta})}\left[\log \hat{p}_1(\boldsymbol{\theta}|\boldsymbol{x})\right] > \mathbb{E}_{p(\boldsymbol{x},\boldsymbol{\theta})}\left[\log \hat{p}_2(\boldsymbol{\theta}|\boldsymbol{x})\right]$ and hence $\mathbb{E}_{p(\boldsymbol{x})}\left[\mathrm{KL}\left[p(\boldsymbol{\theta}|\boldsymbol{x})||\hat{p}_1(\boldsymbol{\theta}|\boldsymbol{x})\right]\right] < \mathbb{E}_{p(\boldsymbol{x})}\left[\mathrm{KL}\left[p(\boldsymbol{\theta}|\boldsymbol{x})||\hat{p}_2(\boldsymbol{\theta}|\boldsymbol{x})\right]\right]$. On the feather falling simulator, we have that $\mathbb{E}_{p(\boldsymbol{x},\boldsymbol{\theta})}\left[\log \hat{p}(\boldsymbol{\theta}|\boldsymbol{x})\right]$ is equal to 1.55 for the NPE approximation and to 0.897 for the NRE approximation. We can conclude that, according to the KL divergence, the NPE approximation is closer to the posterior than the one obtained with NRE, although we were able to reject NPE's approximation and not NRE's. It cannot be evaluated for NLE and NPSE as those only provide sampling from the approximate posterior and not density evaluation.

Alternatively, if the true posterior is known for benchmarking purposes, the Maximum Mean discrepancy (MMD) (Gretton et al., 2012) can be used. The squared MMD can be expressed

$$\begin{aligned}\mathrm{MMD}^2(p(\boldsymbol{\theta}|\boldsymbol{x}), \hat{p}(\boldsymbol{\theta}|\boldsymbol{x})) = \; & \mathbb{E}_{\boldsymbol{\theta},\boldsymbol{\theta}' \sim (p(\boldsymbol{\theta}|\boldsymbol{x})}\left[(k(\boldsymbol{\theta},\boldsymbol{\theta}')\right] \\ & - 2\mathbb{E}_{\boldsymbol{\theta} \sim (p(\boldsymbol{\theta}|\boldsymbol{x}),\boldsymbol{\theta}' \sim \hat{p}(\boldsymbol{\theta}|\boldsymbol{x})}\left[(k(\boldsymbol{\theta},\boldsymbol{\theta}')\right] \\ & + \mathbb{E}_{\boldsymbol{\theta},\boldsymbol{\theta}' \sim (\hat{p}(\boldsymbol{\theta}|\boldsymbol{x})}\left[(k(\boldsymbol{\theta},\boldsymbol{\theta}')\right],\end{aligned} \tag{3.34}$$

where k is a kernel. The expectations can be approximated through Monte Carlo.

4

A CRISIS IN SIMULATION-BASED INFERENCE?

Prologue

This chapter is based on the following publication: *Hermans, J.*, Delaunoy, A.*, Rozet, F., Wehenkel, A., & Louppe, G. (2022). A crisis in simulation-based inference? beware, your posterior approximations can be unfaithful. Transactions on Machine Learning Research.*

This paper is both a position and an empirical paper. We acknowledge the fact that posterior approximations in simulation-based inference will never be perfect and discuss the impact of using those for scientific reasoning. Scientific reasoning in the tradition of Popperian falsification does not aim to verify hypotheses but to disprove false ones. Our position is that, in this setting, failing to disprove a false theory is much less detrimental than erroneously disproving a theory that might be true. By translating this to posterior approximations, an underconfident posterior approximation is much less detrimental than an overconfident approximation. While perfect approximations of posteriors are an unfeasible goal, we argue that trying to avoid overconfidence as much as possible is more realistic and is a goal that should be pursued to allow the reliable use of such approximations for scientific reasoning.

We propose to quantify the overconfidence of an approximation through a quantity that we call expected coverage. To understand how existing methods perform in that regard, we perform a large-scale empirical evaluation of the expected coverage associated with approximations from state-of-the-art methods on a wide variety of benchmarks. The first version of this paper has been published in 2021. Consequently, only methods from the ABC, NLE, NRE, and NPE families are considered. Score-based methods and other types of generative models were introduced later in the context of simulation-based inference. Similarly, only classical sequential methods are considered, as truncated methods were introduced later on.

4.1 INTRODUCTION

Many scientific disciplines rely on computer simulations to study complex phenomena under various conditions. Although modern simulators can generate realistic synthetic observables through detailed descriptions of their data generating processes, they make statistical inference more challenging. The computer code describing the data generating processes defines the likelihood function $p(\boldsymbol{x}|\boldsymbol{\theta})$ only implicitly, and its direct evaluation requires the often intractable integration of all stochastic execution paths. In this problem setting, statistical inference based on the likelihood becomes impractical. However, approximate inference remains possible by relying on likelihood-free approximations thanks to the increasingly accessible and effective suite of methods and software from the field of simulation-based inference (K. Cranmer, Brehmer, and Louppe, 2020).

While simulation-based inference targets domain sciences, advances in the field are mainly driven from a machine learning perspective. The field, therefore, inherits the quality assessments (Lueckmann, Boelts, et al., 2021) customary to the machine learning literature, primarily targeting the exactness of the approximation. In fact, domain sciences, and more specifically the physical sciences, are not necessarily interested in the exactness of an approximation. Instead, in the tradition of Popperian falsification, they often seek to constrain parameters of interest as much as possible at a given confidence level. Scientific examples include frequentist confidence intervals on the mass of the Higgs boson (Aad et al., 2012), Bayesian credible regions on cosmological parameters (Gilman et al., 2018; Aghanim et al., 2020), constraints on the intrinsic parameters of binary black hole coalescences (B. P. Abbott et al., 2016), or characterizing the space of circuit configurations giving rise to rhythmic activity in the crustacean stomatogastric ganglion (Gonçalves et al., 2020). Wrongly excluding plausible values could drive the scientific inquiry in the wrong direction, whereas failing to exclude implausible values because of too conservative estimations is much less detrimental. This implies that statistical approximations in simulation-based inference should ideally come with conservative guarantees to not produce credible regions smaller than they should be, even when the approximations are not faithful. Despite recent developments of post hoc diagnostics to inspect the quality of likelihood-free (K. Cranmer, Pavez, and Louppe, 2015; Brehmer, Mishra-Sharma, et al., 2019; Dalmasso, Izbicki, and Lee, 2020; Talts et al., 2018; Lueckmann, Boelts, et al., 2021; Brehmer, K. Cranmer, et al., 2018; Hermans, Banik, et al., 2021) and likelihood-based (Talts et al., 2018; Geweke et al., 1991; Gelman and Donald B Rubin, 1992; Raftery and Lewis, 1991; A. Dixit and Roy, 2017) approximations, assessing whether approximate inference results are sufficiently reliable for scientific inquiry remains largely unanswered whenever fitting criteria are not globally optimized or whenever the data is limited.

In this work, we measure and discuss the quality of the credible regions produced by various algorithms for Bayesian simulation-based inference. We frame our main contribution as the collection of extensive empirical evidence requiring months of computation, demonstrating that all benchmarked techniques may produce non-conservative credible regions. Our results emphasize the need for a new class of methods: conservative approximate inference algorithms. In addition, we provide empirical evidence that using an ensemble of models in place of a single model tends to produce more reliable approximations. The structure of the paper is outlined as follows. Section 4.2 describes the statistical formalism, necessary background and includes a thorough motivation for coverage. Section 4.3 highlights our main results. Section 4.4 presents several avenues of future research to enable drawing reliable scientific conclusions with simulation-based inference. All code related to this manuscript is available at `https://github.com/montefiore-ai/trust-crisis-in-simulation-based-inference`.

4.2 BACKGROUND

4.2.1 *Statistical formalism*

We evaluate posterior estimators that produce approximations $\hat{p}(\boldsymbol{\theta}|\boldsymbol{x})$ with the following semantics.

Target parameters $\boldsymbol{\theta}$ denote the parameters of interest of a simulation model, and are sometimes referred to as free or model parameters. We make the reasonable assumption that the prior $p(\boldsymbol{\theta})$ is tractable.

An **observable** $\boldsymbol{x}$ denotes a synthetic realization of the simulator, or the observed data $\boldsymbol{x}_o$ we would like to do inference on. We assume that the simulation model is correctly specified and hence is an accurate representation of the real data generation process.

The **likelihood** model $p(\boldsymbol{x}|\boldsymbol{\theta})$ is implicitly defined by the simulator's computer code. While we cannot evaluate the density $p(\boldsymbol{x}|\boldsymbol{\theta})$, we can draw samples through simulation.

The **ground truth** $\boldsymbol{\theta}^*$ specified to the simulation model whose forward evaluation produced the observable $\boldsymbol{x}_o$, i.e., $\boldsymbol{x}_o \sim p(\boldsymbol{x}|\boldsymbol{\theta} = \boldsymbol{\theta}^*)$.

A **credible region** is a space Θ within the target parameters domain that satisfies

$$\int_{\Theta} p(\boldsymbol{\theta}|\boldsymbol{x} = \boldsymbol{x}_o)d\boldsymbol{\theta} = 1 - \alpha \tag{4.1}$$

for some observable $\boldsymbol{x}_o$ and confidence level $1 - \alpha$. Because many such regions exist, we compute the credible region with the smallest volume. In the literature this credible region is known as the highest posterior density region (Box and Tiao, 1973; Hyndman, 1996). In our evaluations, we determine the credible regions by evaluating the approximated posterior density function in a discretized and empirically normalized grid of the parameter space. The credible region is the set of parameters whose density is greater than a given threshold fitted to achieve the desired credibility level $1 - \alpha$.

4.2.2 *Statistical quality assessment*

Common metrics for evaluating the quality of a posterior surrogate include the Classifier Two-sample Test (Lopez-Paz and Oquab, 2017; Lehmann and Romano, 2006) and Maximum Mean Discrepancy (Gretton et al., 2012; Bengio et al., 2014; Dziugaite et al., 2015). The main problem with these metrics is that they assess exactness of an approximation through a divergence with respect to the posterior. All approximations will diverge from

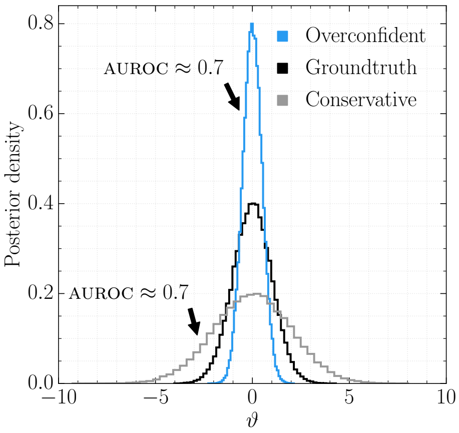

Figure 4.1: A classifier-based metric measures the divergence between posterior approximations and a ground truth by means of evaluating the classifier's discriminative performance through Area Under the Receiver Operating Characteristics curve (AUROC). The metric indicates that both the overconfident and the conservative approximations are equally accurate as it yields AUROC = 0.7 for both of them. From an inference perspective however, the conservative approximation is more suitable because it produces credible regions larger than they should be.

the posterior and there are no criteria to what constitutes an acceptable estimator. For this reason, we argue that metrics evaluating the reliability for scientific inquiry should be used alongside the divergence evaluation when evaluating estimators.

To clarify this point, consider the demonstration in Figure 4.1. A binary classifier is trained to discriminate between samples from a posterior approximation and the true posterior, as in a classifier two-sample test. The discriminative performance of the classifier is expressed through Area Under the Receiver Operating Characteristics curve (AUROC) and serves as a measure for divergence between both densities. An AUROC = 0.5 suggests an approximation that is indistinguishable from the true posterior, while AUROC = 1.0 implies that both distributions do not overlap. Although both the overconfident and the conservative approximations are of equal quality according to the AUROC metric (AUROC = 0.7), credible regions that are biased or smaller than they should be could result in the wrong exclusion of actually plausible parameter values for a given significance level, and hence to erroneous scientific conclusions. By contrast, conservative approximations, which leads to credible regions that are larger than they should be, are more scientifically reliable since they would not wrongly reject plausible parameters values but only fail to reject actually implausible parameter values. For this reason, we take the position that posterior approximations should produce overdispersed credible regions for any simulation budget. Posterior approximations do not have to closely match the true posterior to draw meaningful inferences, but they should however be conservative.

Instead of measuring the exactness of an approximation, this work directly assesses the quality of credible regions through the notion of expected coverage which probes the consistency of the posterior approximations and can be used to diagnose conservative and overconfident approximations. Similar usages of coverage diagnostics can be found in the context of standard statistical inference (Schall, 2012) and approximate Bayesian computation (Prangle et al., 2014).

Definition 4.2.1. The **expected coverage probability** of the $1-\alpha$ highest posterior density regions derived from the posterior estimator $\hat{p}(\boldsymbol{\theta}|\boldsymbol{x})$ is

$$\mathbb{E}_{p(\boldsymbol{\theta},\boldsymbol{x})}\left[\mathbb{1}\left(\boldsymbol{\theta} \in \Theta_{\hat{p}(\boldsymbol{\theta}|\boldsymbol{x})}(1-\alpha)\right)\right], \tag{4.2}$$

where the function $\Theta_{\hat{p}(\boldsymbol{\theta}|\boldsymbol{x})}(1-\alpha)$ yields the $1-\alpha$ highest posterior density region of $\hat{p}(\boldsymbol{\theta}|\boldsymbol{x})$.

Note that Equation 4.2 can be expressed either as

$$\mathbb{E}_{p(\boldsymbol{\theta})}\mathbb{E}_{p(\boldsymbol{x}|\boldsymbol{\theta})}\left[\mathbb{1}\left(\boldsymbol{\theta} \in \Theta_{\hat{p}(\boldsymbol{\theta}|\boldsymbol{x})}(1-\alpha)\right)\right], \tag{4.3}$$

which is the expected frequentist coverage probability, or alternatively as the expected Bayesian credibility

$$\mathbb{E}_{p(\boldsymbol{x})}\mathbb{E}_{p(\boldsymbol{\theta}|\boldsymbol{x})}\left[\mathbb{1}\left(\boldsymbol{\theta} \in \Theta_{\hat{p}(\boldsymbol{\theta}|\boldsymbol{x})}(1-\alpha)\right)\right], \tag{4.4}$$

whose inner expectation reduces to $1-\alpha$ whenever the posterior estimator $\hat{p}(\boldsymbol{\theta}|\boldsymbol{x})$ is well-calibrated.

The expected coverage probability can be estimated empirically given a set of n i.i.d. samples $(\boldsymbol{\theta}_i^*, \boldsymbol{x}_i) \sim p(\boldsymbol{\theta}, \boldsymbol{x})$ as

$$\frac{1}{n}\sum_{i=1}^{n}\mathbb{1}\left(\boldsymbol{\theta}_i^* \in \Theta_{\hat{p}(\boldsymbol{\theta}|\boldsymbol{x}_i)}(1-\alpha)\right). \tag{4.5}$$

Definition 4.2.2. A **conservative posterior estimator** is an estimator that has coverage at the credibility level of interest, i.e., whenever the expected coverage probability is larger or equal to the credibility level.

While coverage is necessary to assess conservativeness, it is limited in its ability to determine the information gain a posterior (approximation) has over its prior. Consider an estimator whose posteriors are identical to the prior. In this case, there is no gain in information and the expected coverage probability is equal to the credibility level. For this reason, a complete analysis should be complemented with measures such as the expected information gain $\mathbb{E}_{p(\boldsymbol{\theta},\boldsymbol{x})}\left[\log p(\boldsymbol{\theta}|\boldsymbol{x}) - \log p(\boldsymbol{\theta})\right]$ or classifier two-sample tests when the ground-truth posterior is available. This work is concerned with conservative inference and will, therefore, mainly focus on the evaluation of expected coverage. Finally, it should be noted that expected coverage is a statement about the credible regions in expectation and, therefore, does not provide any guarantee on the quality of a single posterior approximation. However, the quality of a single posterior approximation can itself be approximated with a local coverage test (Zhao et al., 2021).

4.3 EMPIRICAL OBSERVATIONS

This section covers our main contribution: the collection of empirical evidence to determine whether some simulation-based inference algorithms are conservative by nature. We are particularly interested in determining whether certain approaches should be favored over others. We do so by estimating the expected coverage of posterior estimators produced by these approaches across a broad range of hyperparameters and benchmarks of varying complexity, including two real problems. As in real use cases, the true posteriors are effectively intractable and, therefore, unknown.

4.3.1 *Methods*

We make the distinction between two paradigms. *Non-amortized* approaches are designed to approximate a single posterior, while *amortized* methods aim to learn a general-purpose estimator that attempts to approximate all posteriors supported by the prior.

4.3.1.1 *Amortized*

Neural Ratio Estimation (NRE) is an established approach in the simulation-based inference literature both from a frequentist (K. Cranmer, Pavez, and Louppe, 2015) and Bayesian (Hermans, Begy, and Louppe, 2020; Durkan, Murray, and Papamakarios, 2020; Thomas et al., 2016) perspective. In a Bayesian analysis, an amortized estimator $\hat{r}(\boldsymbol{x}|\boldsymbol{\theta})$ of the intractable likelihood-to-evidence ratio $r(\boldsymbol{x}|\boldsymbol{\theta})$ can be learned by training a binary classifier $\hat{d}(\boldsymbol{\theta}, \boldsymbol{x})$ to distinguish between samples of the joint $p(\boldsymbol{\theta}, \boldsymbol{x})$ with class label 1 and samples of the product of marginals $p(\boldsymbol{\theta})p(\boldsymbol{x})$ with class label 0, with equal label marginal probability. Similar to the density-ratio trick (Hermans, Begy, and Louppe, 2020; K. Cranmer, Pavez, and Louppe, 2015; Sugiyama et al., 2012; Goodfellow et al., 2014), the Bayes optimal classifier $d(\boldsymbol{\theta}, \boldsymbol{x})$ for the cross-entropy loss is

$$d(\boldsymbol{\theta}, \boldsymbol{x}) = \frac{p(\boldsymbol{\theta}, \boldsymbol{x})}{p(\boldsymbol{\theta}, \boldsymbol{x}) + p(\boldsymbol{\theta})p(\boldsymbol{x})} = \sigma\left(\log \frac{p(\boldsymbol{\theta}, \boldsymbol{x})}{p(\boldsymbol{\theta})p(\boldsymbol{x})}\right), \tag{4.6}$$

where $\sigma(\cdot)$ is the sigmoid function. Given a target parameter $\boldsymbol{\theta}$ and an observable $\boldsymbol{x}$ supported by $p(\boldsymbol{\theta})$ and $p(\boldsymbol{x})$ respectively, the learned classifier $\hat{d}(\boldsymbol{\theta}, \boldsymbol{x})$ approximates the log likelihood-to-evidence ratio $\log r(\boldsymbol{x}|\boldsymbol{\theta})$ through the logit function because $\sigma^{-1}(\hat{d}(\boldsymbol{\theta}, \boldsymbol{x})) \approx \log r(\boldsymbol{x}|\boldsymbol{\theta})$. The approximate log posterior density function is $\log p(\boldsymbol{\theta}) + \log \hat{r}(\boldsymbol{x}|\boldsymbol{\theta})$.

Neural Posterior Estimation (NPE) (Rezende and Mohamed, 2015) is concerned with directly learning an amortized posterior estimator $\hat{p}_\psi(\boldsymbol{\theta}|\boldsymbol{x})$ with normalizing flows. Normalizing flows define a class of probability distributions $p_\psi(\cdot)$ built from neural network-based bijective transformations (Dinh, Krueger, and Bengio, 2015; Rezende and Mohamed, 2015; Dinh, Sohl-Dickstein, and Bengio, 2017) parameterized by ψ. They are usually optimized using variational inference, by solving $\arg\min_\psi \mathbb{E}_{p(\boldsymbol{x})}\left[\textsc{kl}(p(\boldsymbol{\theta}|\boldsymbol{x}) \mid\mid \hat{p}_\psi(\boldsymbol{\theta}|\boldsymbol{x})\right]$, which is equivalent to $\arg\max_\psi \mathbb{E}_{p(\boldsymbol{\theta},\boldsymbol{x})}\left[\log \hat{p}_\psi(\boldsymbol{\theta}|\boldsymbol{x})\right]$. Once trained, the density of the modeled distribution can directly be evaluated and sampled from.

Ensembles of models constitute a standard method to improve predictive performance. In this work, we consider an ensemble model that averages the approximated posteriors of n posterior estimators that are either trained independently on the same dataset (deep ensemble) (Lakshminarayanan et al., 2017) or on bootstrap replicates of the learning set (bagging) (Breiman, 1996). While this formulation is natural for NPE, averaging likelihood-to-evidence ratios is equivalent since $\frac{1}{n}\sum_{i=1}^{n}\hat{p}_i(\boldsymbol{\theta}|\boldsymbol{x}) = p(\boldsymbol{\theta})\frac{1}{n}\sum_{i=1}^{n}\hat{r}_i(\boldsymbol{x}|\boldsymbol{\theta})$. We show in Section 4.3 that ensembles lead in average to a higher expected coverage than individual models and hence constitute a straightforward mitigation strategy against overconfidence.

4.3.1.2 *Non-amortized*

Rejection Approximate Bayesian Computation Rej-ABC (Donald B. Rubin, 1984; Pritchard et al., 1999) numerically estimates a single posterior by collecting samples $\boldsymbol{\theta} \sim p(\boldsymbol{\theta})$ whenever $\boldsymbol{x} \sim p(\boldsymbol{x}|\boldsymbol{\theta})$ is similar to $\boldsymbol{x}_o$. Similarity is expressed by means of a distance function ρ. For high-dimensional observables, the probability density of simulating an observable $\boldsymbol{x}$ such that $\boldsymbol{x} = \boldsymbol{x}_o$ is extremely small. For this reason, ABC uses a summary statistic s and an acceptance threshold ϵ. Using these components, ABC accepts samples into the approximate posterior whenever $\rho(s(\boldsymbol{x}), s(\boldsymbol{x}_o)) \leq \epsilon$. Improvements include regression adjustment (Beaumont, W. Zhang, and Balding, 2002) of the sampled parameters using local linear regression, combining ABC with MCMC (Marjoram et al., 2003) and the automatic construction of summary statistics (Fearnhead and Prangle, 2012). In our experiments, we apply Rej-ABC in its simplest form and use the identity function as a sufficient summary statistic, use no regression adjustment and set ϵ such that $\max(100, \text{simulation budget}/100)$ samples are accepted.

Sequential methods for simulation-based inference aim to approximate a single posterior by iteratively improving a posterior approximation. These methods alternate between a simulation and an exploitation phase. The latter is designed to take current knowledge into account such that subsequent simulations can be focused on parameters that are more likely to produce observables $\boldsymbol{x}$ similar to $\boldsymbol{x}_o$.

Sequential Monte-Carlo ABC (SMC-ABC) (Toni and Stumpf, 2009; Sisson, Fan, and Tanaka, 2007; Beaumont, Cornuet, et al., 2009) iteratively updates a set of proposal states to match the posterior distribution. At each iteration, accepted proposals are ranked by distance. The rankings determine whether a proposal is propagated to the next iteration. New candidates are generated by perturbing the selected ranked proposals.

Sequential Neural Posterior Estimation (SNPE) (Papamakarios and Murray, 2016; Lueckmann, Goncalves, et al., 2017; Greenberg et al., 2019) iteratively improves a normalizing flow that models the posterior. Our evaluations will specifically use the SNPE-C (Greenberg et al., 2019) variant.

Sequential Neural Likelihood (SNL) (Papamakarios, Sterratt, and Murray, 2019) models the likelihood $p(\boldsymbol{x}|\boldsymbol{\theta})$. A numerical approximation of the posterior is obtained by plugging the learned likelihood estimator into a Markov Chain Monte Carlo (MCMC) sampler as a surrogate likelihood. Similarly, Price et al., 2018; Frazier, Nott, et al., 2022 construct a synthetic normal approximation of the likelihood over summary statistics.

Sequential Neural Ratio Estimation (SNRE) (Hermans, Begy, and Louppe, 2020; Durkan, Murray, and Papamakarios, 2020) iteratively improves the modelled likelihood-to-evidence ratio.

4.3.2 *Benchmarks*

We consider 7 benchmarks, ranging from a toy problem to real scientific use cases covering various disciplines. All benchmarks and priors are available in the codebase.

The **SLCP** simulator models a fictive problem with 5 parameters. The observable $\boldsymbol{x} \in \mathbb{R}^8$ represents the 2D-coordinates of 4 points. The coordinate of each point is sampled from the same multivariate Gaussian whose mean and covariance matrix are parametrized by $\boldsymbol{\theta}$. We consider an alternative version of the original task (Papamakarios, Sterratt, and Murray, 2019) by inferring the marginal posterior density of 2 of those parameters. In contrast to its original formulation, the likelihood is not tractable due to the marginalization.

The **Weinberg** problem (K. Cranmer, Heinrich, et al., 2017) concerns a simulation of high energy particle collisions $e^+e^- \rightarrow \mu^+\mu^-$. The angular distribution of the particles can be used to measure the Weinberg angle $\boldsymbol{x}$ in the standard model of particle physics. From the scattering angle, we are interested in inferring Fermi's constant $\boldsymbol{\theta}$.

The **Spatial SIR** model involves a grid-world of susceptible, infected, and recovered individuals. Based on initial conditions and the infection and recovery rate $\boldsymbol{\theta}$, the model describes the spatial evolution of an infection. The observable $\boldsymbol{x}$ is a snapshot of the grid-world after some fixed amount of time.

M/G/1 (Blum and François, 2010) models a processing and arrival queue. The problem is described by 3 parameters $\boldsymbol{\theta}$ that influence the time it takes to serve a customer, and the time between their arrivals. The observable $\boldsymbol{x}$ is composed of 5 equally spaced quantiles of inter-departure times.

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 $\boldsymbol{\theta}$ which influence the reproduction and mortality rate of the predators and preys. We infer the marginal posterior of the predator parameters from time series representing the evolution of both populations over time.

Stellar **Streams** form due to the disruption of spherically packed clusters of stars by the Milky Way. Because of their distance from the galactic center and other visible matter, distant stellar streams are considered to be ideal probes to detect gravitational interactions with dark matter. The model (Banik et al., 2018) evolves the stellar density $\boldsymbol{x}$ of a stream over several billion years and perturbs the stream over its evolution through gravitational interactions with dark matter subhaloes parameterized by the dark matter mass $\boldsymbol{\theta}$.

Gravitational Waves (GW) are ripples in space-time emitted during events such as the collision of two black-holes. They can be detected through interferometry measurements $\boldsymbol{x}$ and convey information about celestial bodies, unlocking new ways to study the universe. We consider inferring the masses $\boldsymbol{\theta}$ of two black-holes colliding through the observation of the gravitational wave as measured by LIGO's dual detectors (LIGO Scientific Collaboration, 2018; Biwer et al., 2019).

4.3.3 *Setup*

Our evaluations consider simulation budgets ranging from 2^{10} up to 2^{17} samples and confidence levels from 0.05 up to 0.95. Within the amortized setting we train, for every simulation budget, 5 posterior estimators for 100 epochs. The expected coverage probability is estimated on at least 5,000 unseen samples from the joint $p(\boldsymbol{\theta}, \boldsymbol{x})$ and for all confidence levels under consideration. In addition, we repeat the expected coverage evaluation for ensembles of 5 estimators as well. Special care for non-amortized approaches is necessary because they approximate a single posterior and can therefore not reasonably evaluate expected coverage in the same way. Our experiments for non-amortized approaches estimate the expected coverage by repeating the inference procedure on 300 distinct observables for every simulation budget. The expected coverage probabilities are subsequently estimated based on the resulting posterior approximations. Our experiments with NPE, SNPE, SNL, SNRE, Rej-ABC and SMC-ABC rely on the implementation in the SBI package (Tejero-Cantero et al., 2020), while a custom implementation for NRE is used.

We emphasize the computational requirements necessary to generate our main contribution: the experimental observations, whose generation took months of computation. The average CPU time for evaluating an amortized procedure on all common benchmark problems is in order of 200 CPU days, while for a non-amortized approach this increases to 2800 CPU days. The bulk of the cost was associated with the repeated optimization procedure and execution of the simulator. While we considered benchmarks with low dimensional parameter spaces (up to 3 parameters), we stress that the cost for computing credible regions would grow exponentially with the number of parameters, making coverage diagnostics expensive for high-dimensional problems. Note that when using NPE, the procedure for computing the coverages could be modified to scale to higher dimensions (Rozet and Louppe, 2021a; Rozet and Louppe, 2021b). However, this procedure cannot be applied to all simulation-based inference algorithms. Nevertheless, we expect the observed behaviours to be similar or accentuated in a high-dimensional setting.

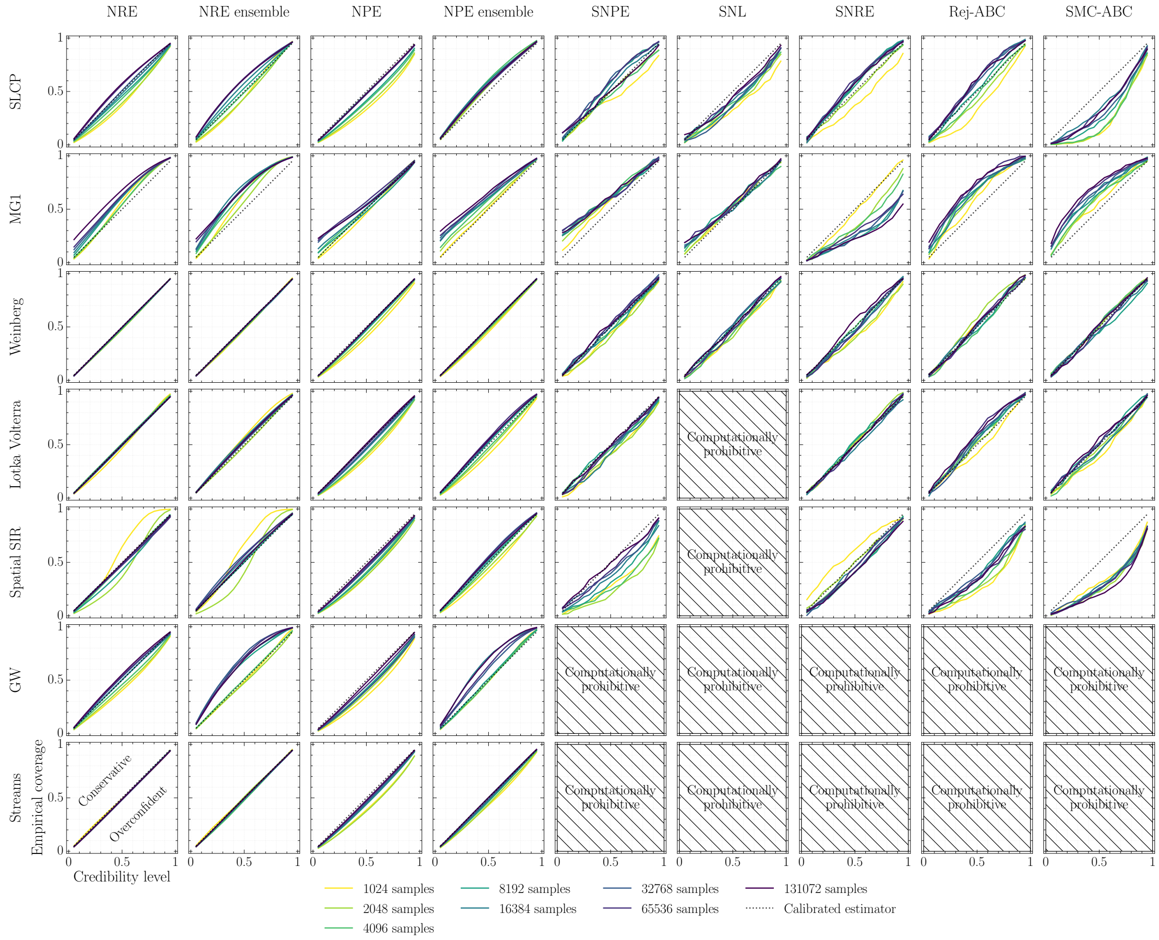

Figure 4.2: Evolution of the expected coverage w.r.t the simulation budget. A perfectly calibrated posterior has an expected coverage probability equal to the credibility level and produces a diagonal line. Conservative estimators on the other hand produce curves above the diagonal and overconfident models underneath. All algorithms can lead to non-conservative estimators. This pathology tends to be accentuated for small simulation budgets and non-amortized methods. Finally, the computationally prohibitive results indicate that the computational requirements did not allow for a coverage analysis. In the case of SNL, this was mostly due to high dimensional observables. For the astronomy benchmarks, the simulation model was simply too expensive to reasonably evaluate coverage for non-amortized methods.

4.3.4 *Results*

Figures 4.2 and 4.4 highlight our main results. Through these plots, we can directly assess whether a posterior estimator is conservative at a given confidence level and simulation budget. The figures should be interpreted as follows: a perfectly calibrated posterior has an expected coverage probability equal to the credibility level. Plotting this relation produces a diagonal line. Conservative estimators on the other hand produce curves above the diagonal and overconfident models underneath. The plots highlight an unsettling observation: all benchmarked approaches produce non-conservative posterior approximations on at least one problem setting. In general, this pathology is especially prominent in non-amortized approaches with a small simulation budget; a regime they have been specifically

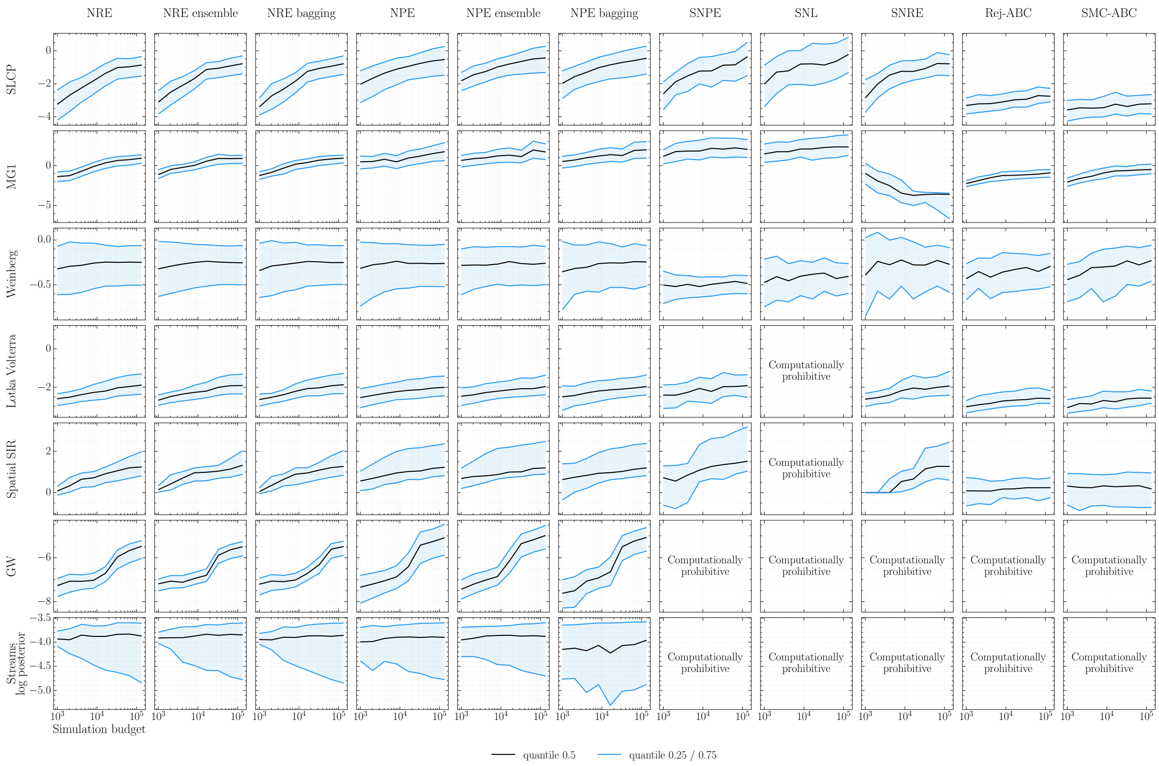

Figure 4.3: Evolution of the predicted log posterior probability of the nominal parameters with respect to the simulation budget. We evaluate the log posterior probability on $100,000$ test samples for amortized algorithms and 300 for non-amortized algorithms. We report the $0.25; 0.5; 0.75$ quantiles of this quantity over the test set. We observe that the predictive performance increases in general with the simulation budget. Exceptions are the MG1 benchmark with the SNRE algorithm and the lowest quantile of the streams benchmark.

designed for. A large simulation budget does not guarantee that a posterior estimator is conservative either. More importantly, as the reliability of the approximations computed through sequential approaches cannot practically be determined, it leaves practitioners uncertain about the reliability of their approximations. To complement this analysis, an evaluation of the predictive performance of the approximate posteriors is provided in Figure 4.3. We observe that, on most benchmark/method pairs, the predictive performance increases with the simulation budget. However, this is not the case for SNRE applied to MG1 which explains the strange behaviour of the coverage curves.

In sequential approaches, overconfidence could be explained by the alternating simulation and exploitation phases. One potential failure mode is that a non-conservative posterior approximation at a previous iteration forces the subsequent simulation phases to not produce observables that should in fact be associated with a higher posterior density, causing the posterior estimator to increase its overconfidence at each iteration.

Despite the fact that all ABC approaches use a sufficient summary statistic (by definition, the identity function), our results demonstrate that this alone is no guarantee for conservative posterior approximations. In fact, using a sufficient summary statistic with $\epsilon > 0$ does not always correspond to conservative approximations either. In such cases, ABC

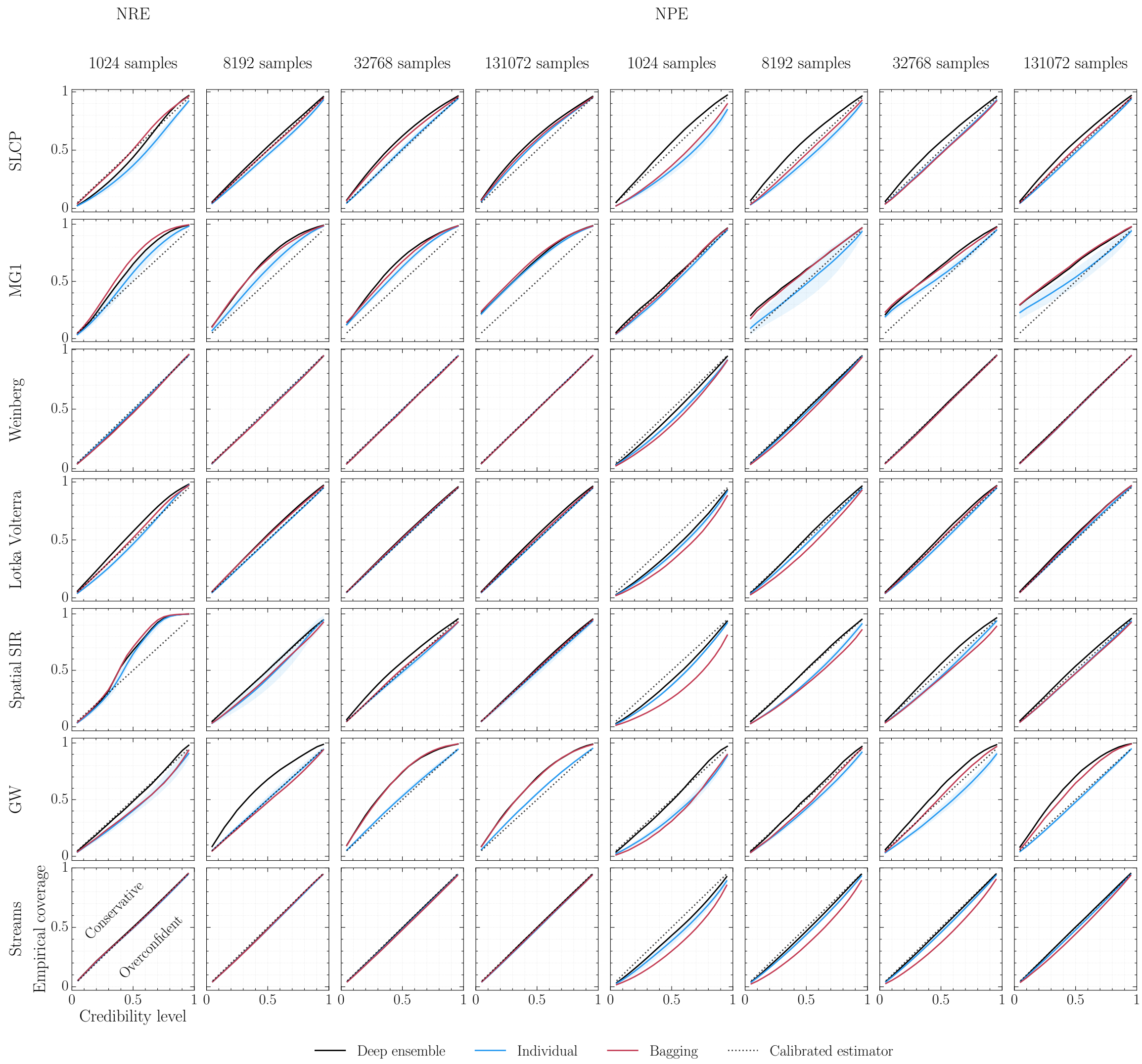

Figure 4.4: Analysis of coverage between ensemble and individual models w.r.t the various simulation budgets. The blue line represents the mean expected coverage of individual models over 5 runs, the shaded area represents its standard deviation. The black line represents the expected coverage of a single ensemble composed of 5 models. We observe that deep ensembles consistently have a higher expected coverage probability compared to the average individual model. A similar effect is not always observed with bagging, indicated by the red line. Ensembles are only evaluated for amortized approaches such as NPE and NRE.

accepts samples with larger distances, permitting the procedure to shift the mass of the approximated posterior elsewhere. In addition, a limited number of posterior samples can negatively affect the quality of the credible regions, e.g., when approximating the posterior density function with kernel density estimation. Both cases can cause the observed behaviour. ABC should therefore be applied with caution to scientific applications. Even though a handcrafted, albeit sufficient, summary statistic provides some insight into the approximated posterior, it does not imply that ABC approximations are conservative whenever the threshold $\epsilon > 0$.

In Figure 4.4, we observe that the expected coverage probability of deep ensemble models is consistently larger than the expected coverage probability of an individual posterior estimator. This highlights the fact that ensembling constitutes an immediately applicable and easy way to mitigate the overconfidence issue and build more reliable posterior estimators. However, the deep ensemble model can still be non-conservative. We hypothesize that the increase in coverage is linked to the added uncertainty captured by the ensemble model, leading to inflated credible regions. In fact, individual estimators only capture data uncertainty, while an ensemble is expected to partially capture the epistemic uncertainty as well. Surprisingly, we find that ensembles built using bagging do not always produce higher coverage than individual models while they should also capture part of the epistemic uncertainty. We could potentially attribute this behaviour to the reduced effective dataset size used to train each member of the ensemble. Figures 4.5 and 4.6 show a positive effect with respect to ensemble size.

Not evident from the figures 4.2 and 4.4 are the computational consequences of a coverage analysis on non-amortized methods. Although the figures mention a certain simulation budget, the total number of simulations for non-amortized methods should be multiplied by the number of approximated posteriors (300) to estimate the coverage. This highlights the simulation cost associated with diagnosing non-amortized approaches. This issue is not limited to coverage. Simulation-Based Calibration (SBC) (Talts et al., 2018) relies on samples of arbitrary posterior approximations as well. Diagnosing non-amortized estimators with SBC therefore requires a similar approach as we have taken in our coverage analyses. In fact, other benchmark papers such as Lueckmann, Boelts, et al. (2021) restrain from evaluating SBC because the diagnostic is computationally prohibitive for non-amortized approaches.

Our results illustrate a clear distinction between the amortized and non-amortized paradigms. Amortized methods do not require retraining or new simulations to determine the empirical expected coverage probability of a posterior estimator, while non-amortized methods do. For this reason, a global coverage analysis of non-amortized approaches is computationally prohibitive and mostly impractical. More importantly, the coverage analysis of a non-amortized approach only measures the quality of the training procedure, whereas a coverage analysis of an amortized approach diagnoses the posterior estimator itself. In addition, a global coverage analysis not only serves as diagnostic but also allows to partially alleviate the issue by performing post-training calibration. A simple way for calibrating level α credible regions is to replace those by credible regions at a level that has the desired expected coverage. Finally, non-amortized sequential algorithms have to repeat the entire simulation-training pipeline whenever architectural or hyperparameter changes are made, while amortized methods reuse previously simulated datasets. All of the above lead us to conclude that while sequential methods have the benefit of being faster to train, amortized methods should be considered for sensitive applications requiring detailed statistical validation.

Observation 1 All benchmarked algorithms may produce non-conservative posterior approximations. This pathology tends to be accentuated with small simulation budgets in both paradigms.

Observation 2 Amortized approaches tend to be more conservative in contrast to non-amortized approaches.

Observation 3 The expected coverage probability of an ensemble model is larger than the average individual model. The ensemble size positively affects the expected coverage probability as well.

Observation 4 Amortized methods are simulation-efficient, especially when taking hyper-parameter tuning and the evaluation of the expected coverage diagnostic into account.

4.4 DISCUSSION

As demonstrated empirically, simulation-based inference can be unreliable, especially whenever its approximations cannot be diagnosed. The problem of determining whether a posterior approximation is correct, or rather, suitable to a use case, is in fact not restricted to simulation-based inference specifically; the concern occurs in all of approximate Bayesian inference. The MCMC literature deals with this exact same problem in the form of determining whether a set of Markov chain samples have converged to the target distribution (Lin, 2014; Hogg and Foreman-Mackey, 2018). In this regard, empirical diagnostic tools have been proposed over the years (Talts et al., 2018; Geweke et al., 1991; Gelman and Donald B Rubin, 1992; Raftery and Lewis, 1991; A. Dixit and Roy, 2017) and have helped practitioners to apply MCMC reliably. While, we are not aware of any coverage studies similar to ours for MCMC, we would expect similar results to be found. Nonetheless, there is currently no clear solution to determine convergence with absolute certainty (A. U. Dixit, 2018; Roy, 2020), even if the likelihood function is tractable. This issue has also been studied in the ABC literature. Prangle et al. (2014) present a way for approximating the coverage in ABC by reusing the same simulations in different runs. Xing et al. (2020) provides a diagnostic for the quality of the approximation for a given observation by the mean of distortion maps. Finally, Frazier, Martin, et al. (2018) provide condition on the threshold ϵ such as to produce posterior estimates that asymptotically have proper frequentist coverage. However, the quality of the posterior estimates remains uncertain in a practical non-asymptotic regime.

We are of the opinion that theoretical and methodological advances within the field of simulation-based inference will strengthen its reliability and thereby promote its applicability in sciences. First, although all benchmarked algorithms recover the true posterior under specific optimal conditions, it is generally not possible to know whether those conditions are satisfied in practice. Therefore, the study of new objective functions that would force posterior estimators to always be conservative, regardless of optimality conditions, constitutes a valuable research avenue. From a Bayesian perspective, Rozet and Louppe

(2021b) propose using the focal and the peripheral losses to weigh down strongly classified samples as a means to tune the conservativeness of a posterior estimator. However, the technique is empirical and requires tuning to attain the desired properties in practice. Dalmasso, Izbicki, and Lee (2020) on the other hand consider the frequentist setting and introduce a theoretically-grounded algorithm for the construction of confidence intervals that are guaranteed to have calibrated coverage, regardless of the quality of the used statistic. Dalmasso, Zhao, et al. (2021) extends this work with finite sample guarantees.

Second, in light of our results that ensembles produce more conservative posteriors, model averaging constitutes another promising direction of study as a simple and directly applicable method to produce reliable posterior estimators. However, a deeper understanding of the behaviour we observe is certainly first required to further develop these methods.

Third, post-training calibration can be used to improve the reliability of posterior estimators and should certainly be considered as a way toward more conservative inference. To some extent, this has already been considered for amortized methods (K. Cranmer, Pavez, and Louppe, 2015; Brehmer, K. Cranmer, et al., 2018; Hermans, Banik, et al., 2021) and would be worth exploring further, especially for non-amortized approaches.

In this work, we assumed that the simulator perfectly models the real data generating process and focused on the computational faithfulness of the inference engine. In practice, there might be a gap between the simulation and reality. This issue, orthogonal to our work, should also be taken into account for making reliable simulation-based inference, as already initiated by several authors (Schmon et al., 2020; Matsubara et al., 2022; Pacchiardi and Dutta, 2022b; Dellaporta et al., 2022).

In summary, we show that current algorithms for simulation-based inference can produce overconfident posterior approximations, making them possibly unreliable for scientific use cases and falsificationist inquiry. Nevertheless, we remain confident and optimistic and advocate that our results are only a stepping stone toward more reliable simulation-based inference and its wider adoption in the sciences.

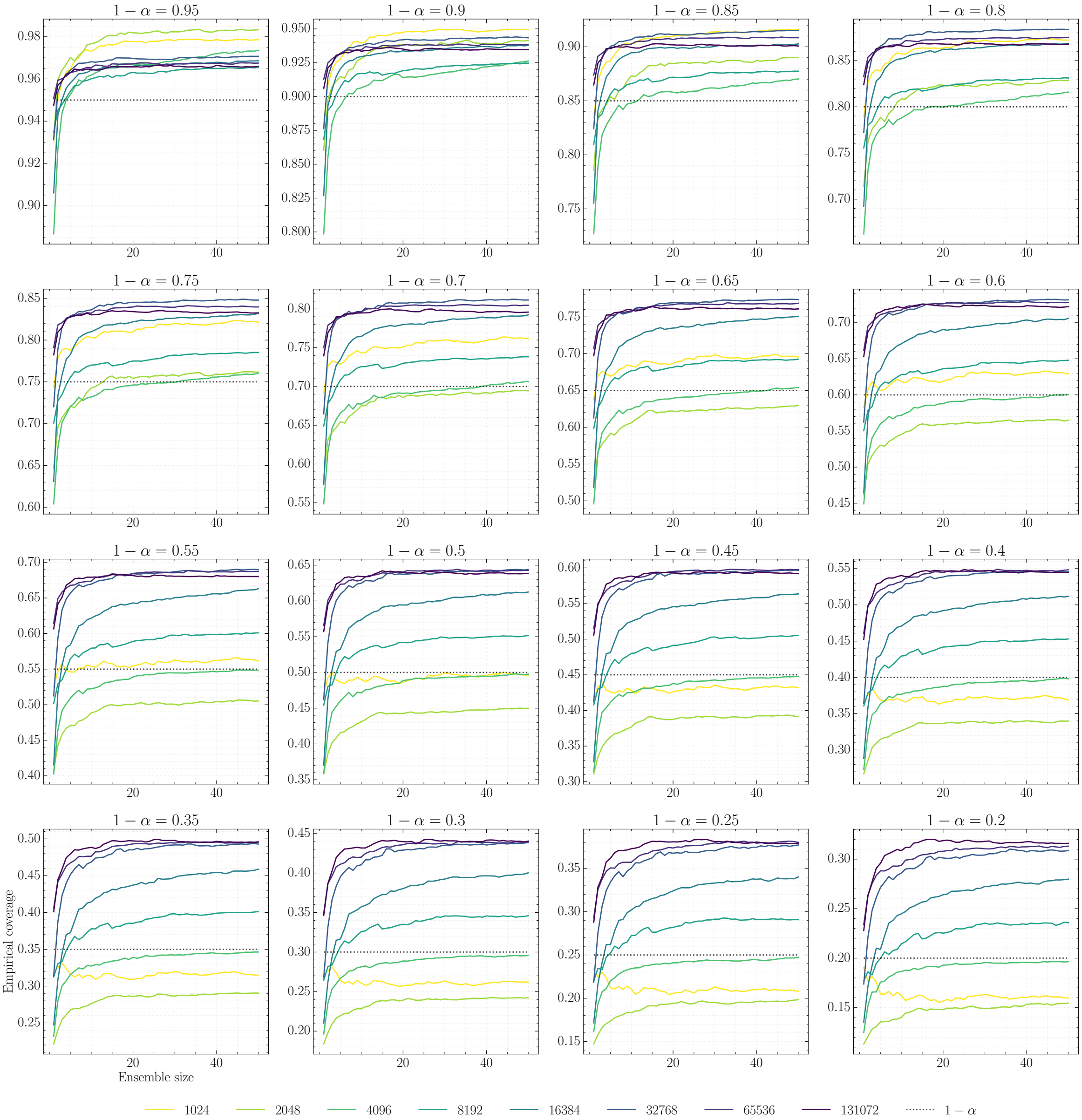

Figure 4.5: Evolution of the empirical expected coverage of deep ensembles with respect to ensemble size for various confidence levels. The results are obtained by training ratio estimators (nre) on the slcp benchmark. A positive effect is observed in terms of empirical expected coverage and ensemble size, i.e., a larger ensemble size correlates with a larger empirical expected coverage. This is unsurprising, because a larger ensemble is expected to capture more of the uncertainty that stems from the training procedure.

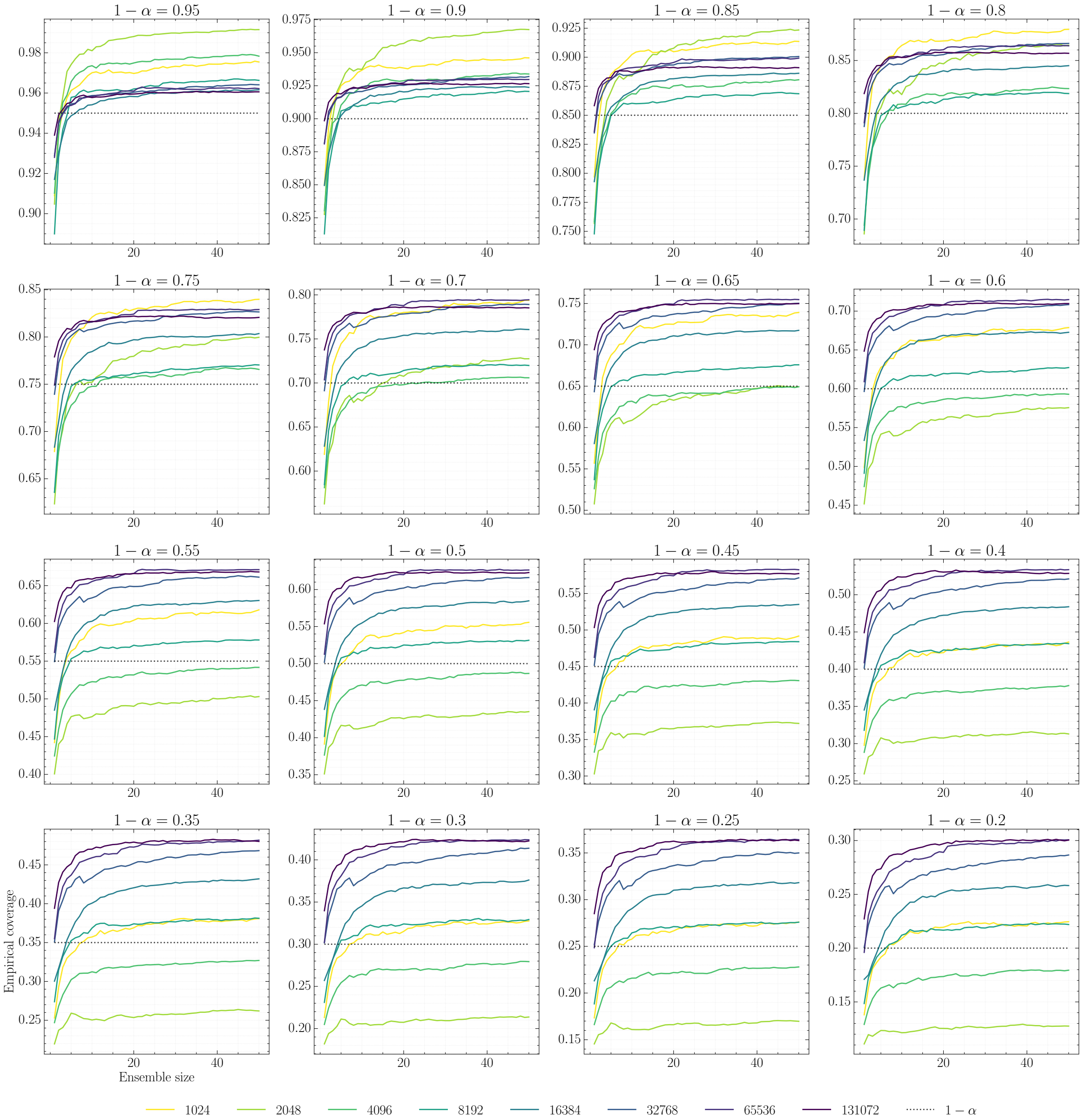

Figure 4.6: Evolution of the empirical expected coverage of bagging ensembles with respect to ensemble size for various confidence levels. The results are obtained by training ratio estimators (NRE) on the SLCP benchmark. A positive effect is observed in terms of empirical expected coverage and ensemble size, i.e., a larger ensemble size correlates with a larger empirical expected coverage. This is unsurprising, because a larger ensemble is expected to capture more of the uncertainty that stems from the training procedure.

Epilogue

In this work, we have introduced the notion of expected coverage to quantify the conservativeness of posterior approximations. From an extensive empirical evaluation, we observed that all the methods yielded overconfident posterior approximations on at least one benchmark. While not surprising, this observation illustrates some concerns with using simulation-based inference algorithms for scientific reasoning. When applying simulation-based inference algorithms in this setting, we advocate that coverage diagnostics are necessary. We observed that, nowadays, a lot of applicative simulation-based inference papers include expected coverage to analyze the quality of the produced posterior approximations (Dimitriou et al., 2023; Vasist et al., 2023; Berteaud et al., 2024; Reza et al., 2024; Swierc et al., 2024; Morandini et al., 2024).

We highlighted that such diagnostics can only be performed with amortized algorithms and raised caution regarding the use of sequential algorithms. Nowadays, this statement is no longer valid with the introduction of truncated sequential algorithms that allow local coverage testing. However, similar caution can be raised regarding newly developed models that only allow sampling from the approximate posterior, such as score-based models and generative adversarial networks. Indeed, computing the expected coverage requires posterior density estimation, which is not immediately available for such models.

In addition to diagnosing posterior approximations, it would be desired to develop algorithms that yield approximations passing the coverage diagnostic. As an attempt to easily improve conservativeness, we showed that ensembling several models empirically leads more often to conservative models. However, there are no guarantees, and ensembling leads to heavier computations. This paper also serves as a call to develop more conservative algorithms for simulation-based inference to increase its applicability to scientific domains.

5 BALANCED NEURAL RATIO ESTIMATION

Prologue

This chapter is based on the following publication: *Delaunoy, A.*, Hermans, J.*, Rozet, F., Wehenkel, A., & Louppe, G. (2022). Towards reliable simulation-based inference with balanced neural ratio estimation. Advances in Neural Information Processing Systems, 35, 20025-20037.* This piece of work is an attempt to increase the conservativeness of neural ratio estimation. We aim to derive conditions under which the trained classifier yields conservative posterior approximations and enforce those conditions during training. We introduce the notion of balanced classifier and show empirically that a balanced classifier yields more conservative approximations than a non-balanced classifier. Additionally, we derive properties of balanced classifier, providing intuition about why more conservative approximations are observed. We introduce the Balanced Neural Ratio Estimation (BNRE) algorithm, a variant of NRE that aims to produce balanced classifiers. BNRE is implemented in the LAMPE (Rozet, François, 2023) and SBI (Tejero-Cantero et al., 2020) libraries.

5.1 INTRODUCTION

Many areas of science and engineering use parametric computer simulations to describe complex stochastic generative processes. In this setting, Bayesian inference provides a principled framework to identify parameters matching empirical observations. Computer simulations, however, define the necessary likelihood function only implicitly, which prevents its evaluation and the use of classical inference algorithms. To overcome this obstacle, recent simulation-based inference (SBI) algorithms (K. Cranmer, Brehmer, and Louppe, 2020) build upon deep learning surrogates to approximate parts of the Bayes rule and enable approximate inference. For example, Papamakarios, Sterratt, and Murray (2019) and Glöckler et al. (2021) build a surrogate of the likelihood function while Hermans, Begy, and Louppe (2020), Durkan, Murray, and Papamakarios (2020), K. Cranmer, Pavez, and Louppe (2015), Benjamin K Miller, Cole, et al. (2021), and Thomas et al. (2016) approximate the likelihood-to-evidence ratio. The posterior can also be targeted directly with variational inference, as proposed by Papamakarios and Murray (2016), Greenberg et al. (2019), and Glöckler et al. (2021). These algorithms are either amortized or run sequentially to drive the training towards a target observation and improve the simulation efficiency of the procedure (Hermans, Begy, and Louppe, 2020; Durkan, Murray, and Papamakarios, 2020; Papamakarios, Sterratt, and Murray, 2019; Papamakarios and Murray, 2016; Lueckmann, Goncalves, et al., 2017; Greenberg et al., 2019; Glöckler et al., 2021). However, sequential methods have the drawback of being computationally expensive to diagnose as the surrogates are only valid for the target observation (Hermans, Delaunoy, et al., 2022). Truncated marginal neural ratio estimation (Benjamin K Miller, Cole, et al., 2021) alleviates this issue by introducing a sequential algorithm that builds a surrogate valid in a local region around the target.

Since modern simulation-based inference algorithms rely on deep learning surrogates, concerns naturally arise regarding their computational faithfulness and whether they are sufficiently adequate for the inference task of interest. In Bayesian inference, these concerns can be at least partially addressed with diagnostics designed to probe the correct behaviour of the inference method, such as $\hat{R}$ diagnostics for MCMC (Gelman and Donald B Rubin, 1992), or to assess the quality of posterior approximations directly. The latter include diagnostics such as simulation-based calibration (SBC) (Talts et al., 2018) or coverage-based diagnostics (Zhao et al., 2021; Hermans, Delaunoy, et al., 2022). As discussed by Hermans, Delaunoy, et al. (2022), posterior approximations must be conservative to guarantee reliable inferences, even when approximations are not faithful. For example, in the physical sciences, where the goal is often to constrain parameters of interest, wrongly excluding plausible values could drive the scientific inquiry in the wrong direction, whereas failing to exclude implausible values because of (too) conservative estimations is much less detrimental. Unfortunately, the same authors also demonstrate that current simulation-based inference algorithms can lead to overconfident surrogates and therefore false inferences.

Scientific use cases requiring conservative inference include for example the study of dark matter models in particle physics and astrophysics (Hermans, Banik, et al., 2021), which could be cold, warm, or hot dark matter. In general, thermal dark matter models are described by a single parameter, the dark matter thermal relic mass, which can be intuitively thought of as the energy the dark matter particle had in the Early Universe. Small values correspond to warm or hot dark matter, while high values are descriptive of cold dark matter. Applying an inference algorithm without diagnosing the learned estimator could lead to constraints that are tighter than they should be. For example, whenever an overconfident estimator produces posterior estimates that favor cold dark matter models, it could simultaneously reject alternative models, such as the extensively studied Sterile Neutrino, a potential candidate for the Warm Dark Matter particle. Making a scientific statement in this direction therefore requires the uttermost care to not wrongly exclude values of the thermal relic mass that are actually plausible.

In this work, we develop a novel algorithm that not only converges to exact inference as the simulation budget increases, but which is also more likely to produce conservative surrogates in small simulation budget regimes. Toward this objective, we propose a variant of the NRE algorithm called Balanced Neural Ratio Estimation (BNRE), which enforces a balancing condition on the binary neural classifier to increase the reliability of its posterior approximations.

The structure of the manuscript is outlined as follows. Section 5.2 describes the formalism and the necessary background. Section 5.3 describes BNRE and provides theoretical arguments towards its conservativeness and reliability. Section 5.4 illustrates our main results and provides insights regarding the behavior of the method. Finally, Section 5.5 discusses related work while Section 5.6 summarizes our contributions and hints at future work.

5.2 BACKGROUND

5.2.1 *Statistical formalism*

This work is concerned with simulation-based inference algorithms that produce posterior approximations $\hat{p}(\boldsymbol{\theta}|\boldsymbol{x})$ under the following semantics. Target parameters $\boldsymbol{\theta}$ denote the parameters of the model and we make the reasonable assumption that the prior $p(\boldsymbol{\theta})$ is tractable. The model is generically expressed as a computer program, a simulator, that describes the forward dynamics of interest based on the input parameters $\boldsymbol{\theta}$. The simulator implicitly defines the likelihood function $p(\boldsymbol{x}|\boldsymbol{\theta})$. While we cannot directly evaluate the likelihood $p(\boldsymbol{x}|\boldsymbol{\theta})$, we can execute the computer program to generate synthetic observables $\boldsymbol{x} \sim p(\boldsymbol{x}|\boldsymbol{\theta})$. Every observable $\boldsymbol{x}_o$ is tied to ground truth parameters $\boldsymbol{\theta}^*$ whose forward evaluation within the simulator produced $\boldsymbol{x}^*$.

Of special importance to Bayesians is the notion of a credible region, which is a domain $\boldsymbol{\Theta}$ within the target parameter space that satisfies $\int_{\boldsymbol{\Theta}} p(\boldsymbol{\theta}|\boldsymbol{x} = \boldsymbol{x}^*)d\boldsymbol{\theta} = 1 - \alpha$ for some observable $\boldsymbol{x}^*$ and confidence level $1 - \alpha$. Because many such regions exist, we target the credible region with the smallest volume, also known as the highest posterior density region (Box and Tiao, 1973; Hyndman, 1996).

5.2.2 *Neural ratio estimation*

Neural Ratio Estimation (NRE) is an established approach in the simulation-based inference literature both from frequentist (K. Cranmer, Pavez, and Louppe, 2015) and Bayesian (Hermans, Begy, and Louppe, 2020; Durkan, Murray, and Papamakarios, 2020; Benjamin K Miller, Cole, et al., 2021; Thomas et al., 2016) perspectives. In essence, all protocols rely on the density-ratio trick (K. Cranmer, Pavez, and Louppe, 2015; Sugiyama et al., 2012; Goodfellow et al., 2014) to construct a surrogate of the likelihood ratio. In this work, we consider an amortized estimator $\hat{r}(\boldsymbol{x}|\boldsymbol{\theta})$ of the intractable likelihood-to-evidence ratio $r(\boldsymbol{x}|\boldsymbol{\theta}) = p(\boldsymbol{\theta}, \boldsymbol{x})/p(\boldsymbol{\theta})p(\boldsymbol{x}) = p(\boldsymbol{x}|\boldsymbol{\theta})/p(\boldsymbol{x})$ that can be learned by training a binary classifier $\hat{d} : \boldsymbol{X} \times \boldsymbol{\Theta} \mapsto [0, 1]$ to distinguish between samples of the joint $p(\boldsymbol{\theta}, \boldsymbol{x})$ with class label

1 and samples of the product of marginals $p(\boldsymbol{\theta})p(\boldsymbol{x})$ with class label 0, with equal label marginal probability. For the binary cross-entropy loss, the Bayes optimal classifier is

$$d(\boldsymbol{\theta}, \boldsymbol{x}) = \frac{p(\boldsymbol{\theta}, \boldsymbol{x})}{p(\boldsymbol{\theta}, \boldsymbol{x}) + p(\boldsymbol{\theta})p(\boldsymbol{x})} = \sigma\left(\log \frac{p(\boldsymbol{\theta}, \boldsymbol{x})}{p(\boldsymbol{\theta})p(\boldsymbol{x})}\right), \tag{5.1}$$

where $\sigma(\cdot)$ is the sigmoid function. Given target parameters $\boldsymbol{\theta}$ and an observable $\boldsymbol{x}$ supported by $p(\boldsymbol{\theta})$ and $p(\boldsymbol{x})$ respectively, the learned classifier $\hat{d}$ provides an approximation for the log likelihood-to-evidence ratio $\log r(\boldsymbol{x}|\boldsymbol{\theta})$ because $\log r(\boldsymbol{x}|\boldsymbol{\theta}) = \sigma^{-1}(d(\boldsymbol{\theta}, \boldsymbol{x})) \approx \sigma^{-1}(\hat{d}(\boldsymbol{\theta}, \boldsymbol{x})) = \log \hat{r}(\boldsymbol{x}|\boldsymbol{\theta})$. The log posterior density function is approximated as $\log \hat{p}(\boldsymbol{\theta}|\boldsymbol{x}) = \log p(\boldsymbol{\theta}) + \log \hat{r}(\boldsymbol{x}|\boldsymbol{\theta})$.

5.3 BALANCED BINARY CLASSIFICATION FOR NEURAL RATIO ESTIMATION

Following Hermans, Delaunoy, et al. (2022), let us first define the expected coverage probability of the $1 - \alpha$ highest posterior density regions derived from the posterior estimator $\hat{p}(\boldsymbol{\theta}|\boldsymbol{x})$ as

$$\mathbb{E}_{p(\boldsymbol{\theta}, \boldsymbol{x})}\left[\mathbb{1}\left(\boldsymbol{\theta} \in \Theta_{\hat{p}(\boldsymbol{\theta}|\boldsymbol{x})}(1-\alpha)\right)\right], \tag{5.2}$$

where the function $\Theta_{\hat{p}(\boldsymbol{\theta}|\boldsymbol{x})}(1-\alpha)$ yields the $1 - \alpha$ highest posterior density region of $\hat{p}(\boldsymbol{\theta}|\boldsymbol{x})$. This diagnostic probes the conservativeness of the posterior estimator (or the lack thereof) and can be interpreted as the expected frequentist coverage $\mathbb{E}_{p(\boldsymbol{\theta})}\mathbb{E}_{p(\boldsymbol{x}|\boldsymbol{\theta})}\left[\mathbb{1}\left(\boldsymbol{\theta} \in \Theta_{\hat{p}(\boldsymbol{\theta}|\boldsymbol{x})}(1-\alpha)\right)\right]$.

In this work, a posterior estimator has coverage at the confidence level $1 - \alpha$ whenever the expected coverage probability is larger or equal to the nominal coverage probability, $1 - \alpha$. We say that a posterior estimator is conservative when it has coverage for all confidence levels. The expected coverage probability can be plotted for various levels α, which allows to visually identify conservative posterior estimators.

The expected coverage can also be shown to be a special case of the Simulation-based calibration (SBC) diagnostic (Talts et al., 2018), further motivating the usage of expected coverage. SBC provides a way to diagnose the faithfulness of an approximate posterior distribution $\hat{p}(\boldsymbol{\theta}|\boldsymbol{x})$. Given an observation $\boldsymbol{x}^* \sim p(\boldsymbol{x})$, Talts et al. (2018) prove that, for any one-dimensional statistic $f : \Theta \mapsto \mathbb{R}$, the rank statistic

$$r(\boldsymbol{\theta}^*) = \mathbb{E}_{p(\boldsymbol{\theta}|\boldsymbol{x}^*)}\Big[\mathbb{1}[f(\boldsymbol{\theta}) \leq f(\boldsymbol{\theta}^*)]\Big] \tag{5.3}$$

of posterior samples $\boldsymbol{\theta}^* \sim p(\boldsymbol{\theta}|\boldsymbol{x}^*)$ is uniformly distributed over the interval $[0, 1]$. Consequently, any deviation from the uniform distribution for the approximate rank statistic

$$\hat{r}(\boldsymbol{\theta}^*) = \mathbb{E}_{\hat{p}(\boldsymbol{\theta}|\boldsymbol{x}^*)}\Big[\mathbb{1}[f(\boldsymbol{\theta}) \leq f(\boldsymbol{\theta}^*)]\Big] \tag{5.4}$$

indicates some error in the approximate posterior $\hat{p}(\boldsymbol{\theta}|\boldsymbol{x}^*)$. As this holds for any statistic f, it also holds for $f(\boldsymbol{\theta}) = \hat{p}(\boldsymbol{\theta}|\boldsymbol{x}^*)$. In this special case, if $\hat{r}(\boldsymbol{\theta}^*) = \alpha$, a proportion $1 - \alpha$ of samples $\boldsymbol{\theta} \sim \hat{p}(\boldsymbol{\theta}|\boldsymbol{x}^*)$ have an approximate posterior density larger than $\boldsymbol{\theta}^*$. In other words, it means that $\boldsymbol{\theta}^*$ resides within the $1-\alpha$ highest posterior density region $\Theta_{\hat{p}(\boldsymbol{\theta}|\boldsymbol{x}^*)}(1-\alpha)$ of $\hat{p}(\boldsymbol{\theta}|\boldsymbol{x}^*)$. Therefore, we have

$$P(\hat{r}(\boldsymbol{\theta}^*) \geq \alpha) = \mathbb{E}_{p(\boldsymbol{\theta}^*|\boldsymbol{x}^*)}\left[\mathbb{1}[\boldsymbol{\theta}^* \in \Theta_{\hat{p}(\boldsymbol{\theta}|\boldsymbol{x}^*)}(1-\alpha)]\right] \tag{5.5}$$

and since $\hat{r}(\boldsymbol{\theta}^*)$ should be uniformly distributed, $P(\hat{r}(\boldsymbol{\theta}^*) \geq \alpha)$ should be equal to $1-\alpha$. In practice, this test cannot be performed locally for a given $\boldsymbol{x}^*$ as we cannot sample from the unknown posterior distribution $p(\boldsymbol{\theta}|\boldsymbol{x}^*)$. Instead, SBC checks globally that $\hat{r}(\boldsymbol{\theta}^*)$ is uniformly distributed over pairs $(\boldsymbol{\theta}^*, \boldsymbol{x}^*) \sim p(\boldsymbol{\theta}, \boldsymbol{x})$ sampled from the joint distribution, which, in the special case $f(\boldsymbol{\theta}) = \hat{p}(\boldsymbol{\theta}|\boldsymbol{x}^*)$, comes down to check that

$$\mathbb{E}_{p(\boldsymbol{\theta}^*, \boldsymbol{x}^*)}\left[\mathbb{1}[\boldsymbol{\theta}^* \in \Theta_{\hat{p}(\boldsymbol{\theta}|\boldsymbol{x}^*)}(1-\alpha)]\right] = 1-\alpha \tag{5.6}$$

is satisfied for all $\alpha \in [0,1]$. We recognize here the expected coverage diagnostic used in Hermans, Delaunoy, et al. (2022) and this work.

Our main objective is to restrict the hypothesis space of the approximate classifiers $\hat{d}$ to those leading to conservative posterior estimators, hence solving the reliability concerns of NRE. Towards this goal, we construct a hypothesis space of balanced classifiers and show both theoretically and empirically that they lead to posterior estimators that tend to be more conservative.

5.3.1 *Balanced binary classification*

Definition 5.3.1. A classifier $\hat{d}$ is balanced if $\mathbb{E}_{p(\boldsymbol{\theta},\boldsymbol{x})}\left[\hat{d}(\boldsymbol{\theta},\boldsymbol{x})\right] = \mathbb{E}_{p(\boldsymbol{\theta})p(\boldsymbol{x})}\left[1-\hat{d}(\boldsymbol{\theta},\boldsymbol{x})\right]$, or

$$\mathbb{E}_{p(\boldsymbol{\theta},\boldsymbol{x})}\left[\hat{d}(\boldsymbol{\theta},\boldsymbol{x})\right] + \mathbb{E}_{p(\boldsymbol{\theta})p(\boldsymbol{x})}\left[\hat{d}(\boldsymbol{\theta},\boldsymbol{x})\right] = 1. \tag{5.7}$$

Theorem 5.3.2. *Any balanced classifier $\hat{d}$ satisfies* $\mathbb{E}_{p(\boldsymbol{\theta},\boldsymbol{x})}\left[\frac{d(\boldsymbol{\theta},\boldsymbol{x})}{\hat{d}(\boldsymbol{\theta},\boldsymbol{x})}\right] \geq 1.$

Proof. The integral form of the balancing condition

$$\iint \Big(p(\boldsymbol{\theta},\boldsymbol{x}) + p(\boldsymbol{\theta})p(\boldsymbol{x})\Big)\hat{d}(\boldsymbol{\theta},\boldsymbol{x})d\boldsymbol{\theta}d\boldsymbol{x} = 1 \tag{5.8}$$

implies that $\Big(p(\boldsymbol{x},\boldsymbol{\theta}) + p(\boldsymbol{\theta})p(\boldsymbol{x})\Big)\hat{d}(\boldsymbol{\theta},\boldsymbol{x})$ is a valid density, both integrating to 1 and positive everywhere. Therefore, its Kullback-Leibler (KL) divergence with $p(\boldsymbol{\theta},\boldsymbol{x})$ is positive. Through Jensen's inequality, we obtain

$$\begin{aligned}
0 &\leq \mathrm{KL}\left(p(\boldsymbol{\theta},\boldsymbol{x}) \,\Big|\Big|\, \Big(p(\boldsymbol{\theta},\boldsymbol{x}) + p(\boldsymbol{\theta})p(\boldsymbol{x})\Big)\hat{d}(\boldsymbol{\theta},\boldsymbol{x})\right) \\
&\leq \mathbb{E}_{p(\boldsymbol{\theta},\boldsymbol{x})}\left[\log \frac{p(\boldsymbol{\theta},\boldsymbol{x})}{\Big(p(\boldsymbol{\theta},\boldsymbol{x}) + p(\boldsymbol{\theta})p(\boldsymbol{x})\Big)\hat{d}(\boldsymbol{\theta},\boldsymbol{x})}\right] \\
&\leq \mathbb{E}_{p(\boldsymbol{\theta},\boldsymbol{x})}\left[\log \frac{d(\boldsymbol{\theta},\boldsymbol{x})}{\hat{d}(\boldsymbol{\theta},\boldsymbol{x})}\right] \\
\Rightarrow \quad 1 &\leq \mathbb{E}_{p(\boldsymbol{\theta},\boldsymbol{x})}\left[\exp\left(\log \frac{d(\boldsymbol{\theta},\boldsymbol{x})}{\hat{d}(\boldsymbol{\theta},\boldsymbol{x})}\right)\right] = \mathbb{E}_{p(\boldsymbol{\theta},\boldsymbol{x})}\left[\frac{d(\boldsymbol{\theta},\boldsymbol{x})}{\hat{d}(\boldsymbol{\theta},\boldsymbol{x})}\right]. \qquad \square
\end{aligned}$$

Theorem 5.3.3. *Any balanced classifier $\hat{d}$ satisfies* $\mathbb{E}_{p(\boldsymbol{\theta})p(\boldsymbol{x})}\left[\frac{1-d(\boldsymbol{\theta},\boldsymbol{x})}{1-\hat{d}(\boldsymbol{\theta},\boldsymbol{x})}\right] \geq 1.$

Proof. From the integral form of the balancing condition, we have

$$\begin{aligned}
1 &= \iint \Big(p(\boldsymbol{\theta},\boldsymbol{x}) + p(\boldsymbol{\theta})p(\boldsymbol{x})\Big)\hat{d}(\boldsymbol{\theta},\boldsymbol{x})d\boldsymbol{\theta}d\boldsymbol{x} \\
&= 2 - \iint \Big(p(\boldsymbol{\theta},\boldsymbol{x}) + p(\boldsymbol{\theta})p(\boldsymbol{x})\Big)\hat{d}(\boldsymbol{\theta},\boldsymbol{x})d\boldsymbol{\theta}d\boldsymbol{x} \\
&= \iint p(\boldsymbol{\theta},\boldsymbol{x})d\boldsymbol{\theta}d\boldsymbol{x} + \iint p(\boldsymbol{\theta})p(\boldsymbol{x})d\boldsymbol{\theta}d\boldsymbol{x} - \iint \Big(p(\boldsymbol{\theta},\boldsymbol{x}) + p(\boldsymbol{\theta})p(\boldsymbol{x})\Big)\hat{d}(\boldsymbol{\theta},\boldsymbol{x})d\boldsymbol{\theta}d\boldsymbol{x} \\
&= \iint \Big(p(\boldsymbol{\theta},\boldsymbol{x}) + p(\boldsymbol{\theta})p(\boldsymbol{x})\Big)\Big(1-\hat{d}(\boldsymbol{\theta},\boldsymbol{x})\Big)d\boldsymbol{\theta}d\boldsymbol{x},
\end{aligned}$$

which implies that $\Big(p(\boldsymbol{x},\boldsymbol{\theta}) + p(\boldsymbol{\theta})p(\boldsymbol{x})\Big)\Big(1-\hat{d}(\boldsymbol{\theta},\boldsymbol{x})\Big)$ is a valid density, integrating to 1 and positive everywhere. Therefore, its Kullback-Leibler divergence with $p(\boldsymbol{\theta})p(\boldsymbol{x})$ is positive and, using Jensen's inequality, we have

$$\begin{aligned}
0 &\leq \mathrm{KL}\left(p(\boldsymbol{\theta})p(\boldsymbol{x}) \,\Big|\Big|\, \Big(p(\boldsymbol{\theta},\boldsymbol{x}) + p(\boldsymbol{\theta})p(\boldsymbol{x})\Big)\Big(1-\hat{d}(\boldsymbol{\theta},\boldsymbol{x})\Big)\right) \\
&\leq \mathbb{E}_{p(\boldsymbol{\theta})p(\boldsymbol{x})}\left[\log \frac{p(\boldsymbol{\theta})p(\boldsymbol{x})}{\Big(p(\boldsymbol{\theta},\boldsymbol{x}) + p(\boldsymbol{\theta})p(\boldsymbol{x})\Big)\Big(1-\hat{d}(\boldsymbol{\theta},\boldsymbol{x})\Big)}\right] \\
&\leq \mathbb{E}_{p(\boldsymbol{\theta})p(\boldsymbol{x})}\left[\log \frac{1-d(\boldsymbol{\theta},\boldsymbol{x})}{1-\hat{d}(\boldsymbol{\theta},\boldsymbol{x})}\right] \\
\Rightarrow \quad 1 &\leq \mathbb{E}_{p(\boldsymbol{\theta})p(\boldsymbol{x})}\left[\exp\left(\log \frac{1-d(\boldsymbol{\theta},\boldsymbol{x})}{1-\hat{d}(\boldsymbol{\theta},\boldsymbol{x})}\right)\right] = \mathbb{E}_{p(\boldsymbol{\theta})p(\boldsymbol{x})}\left[\frac{1-d(\boldsymbol{\theta},\boldsymbol{x})}{1-\hat{d}(\boldsymbol{\theta},\boldsymbol{x})}\right]. \qquad \square
\end{aligned}$$

Theorem 5.3.2 shows that, in expectation over the joint distribution $p(\boldsymbol{\theta},\boldsymbol{x})$, a balanced classifier $\hat{d}$ tends to make predictions whose probability values $\hat{d}(\boldsymbol{\theta},\boldsymbol{x})$ are smaller than the exact probability values $d(\boldsymbol{\theta},\boldsymbol{x})$. In other words, a balanced classifier $\hat{d}$ tends to be less confident than the Bayes optimal classifier d. Similarly, Theorem 5.3.3 shows that, in expectation over the product of the marginals $p(\boldsymbol{\theta})p(\boldsymbol{x})$, a balanced classifier tends to make predictions whose probability values $1 - \hat{d}(\boldsymbol{\theta}, \boldsymbol{x})$ are smaller than the exact probability values $1 - d(\boldsymbol{\theta}, \boldsymbol{x})$, hence showing that a balanced classifier $\hat{d}$ tends to also be less confident than the Bayes optimal classifier d. We note however that these two theorems hold only in expectation, which implies that neither $\hat{d}(\boldsymbol{\theta}, \boldsymbol{x}) \leq d(\boldsymbol{\theta}, \boldsymbol{x})$ for all $\boldsymbol{\theta}, \boldsymbol{x}$ nor $1 - \hat{d}(\boldsymbol{\theta}, \boldsymbol{x}) \leq 1 - d(\boldsymbol{\theta}, \boldsymbol{x})$ for all $\boldsymbol{\theta}, \boldsymbol{x}$ can generally be guaranteed.

Theorem 5.3.4. *The Bayes optimal classifier $d(\boldsymbol{\theta}, \boldsymbol{x})$ is balanced.*

Proof. Replacing the Bayes optimal classifier

$$d(\boldsymbol{\theta}, \boldsymbol{x}) \triangleq \frac{p(\boldsymbol{\theta}, \boldsymbol{x})}{p(\boldsymbol{\theta}, \boldsymbol{x}) + p(\boldsymbol{\theta})p(\boldsymbol{x})} \tag{5.9}$$

in the integral form of the balancing condition, we have

$$\begin{aligned} &\iint (p(\boldsymbol{\theta}, \boldsymbol{x}) + p(\boldsymbol{\theta})p(\boldsymbol{x}))d(\boldsymbol{\theta}, \boldsymbol{x}) d\boldsymbol{\theta} d\boldsymbol{x} \\ &= \iint \frac{\big(p(\boldsymbol{\theta}, \boldsymbol{x}) + p(\boldsymbol{\theta})p(\boldsymbol{x})\big)\, p(\boldsymbol{\theta}, \boldsymbol{x})}{p(\boldsymbol{\theta}, \boldsymbol{x}) + p(\boldsymbol{\theta})p(\boldsymbol{x})} d\boldsymbol{\theta} d\boldsymbol{x} \\ &= \iint p(\boldsymbol{\theta}, \boldsymbol{x}) d\boldsymbol{\theta} d\boldsymbol{x} = 1. \end{aligned}$$

□

Theorem 5.3.4 states that the Bayes optimal classifier is balanced. Therefore, **minimizing the cross-entropy loss while restricting the model hypothesis space to balanced classifiers results in the same Bayes optimal classifier of Eqn. 5.1.**

5.3.2 *Balanced neural ratio estimation*

We now extend the NRE algorithm to enforce the balancing condition. The previous results show that enforcing the condition should result in more conservative classifiers $\hat{d}$ and therefore to dispersed posterior approximations. Let us first note that Theorem 5.3.2 can be expressed as $\mathbb{E}_{p(\boldsymbol{x})}[\mathbb{E}_{p(\boldsymbol{\theta}|\boldsymbol{x})}[d(\boldsymbol{\theta}, \boldsymbol{x})/\hat{d}(\boldsymbol{\theta}, \boldsymbol{x}]] \geq 1$, which can (ideally) be achieved when the inner expectation is larger than 1 for all $\boldsymbol{x}$. In this case, the classifier $\hat{d}$ will be such that $\hat{d}(\boldsymbol{\theta}, \boldsymbol{x}) \leq d(\boldsymbol{\theta}, \boldsymbol{x})$ in regions of high posterior density. Then,

$$\frac{\hat{d}(\boldsymbol{\theta}, \boldsymbol{x})}{1 - \hat{d}(\boldsymbol{\theta}, \boldsymbol{x})} \leq \frac{d(\boldsymbol{\theta}, \boldsymbol{x})}{1 - d(\boldsymbol{\theta}, \boldsymbol{x})}, \text{ which is equivalent to } \hat{r}(\boldsymbol{x}|\boldsymbol{\theta}) \leq r(\boldsymbol{x}|\boldsymbol{\theta}), \tag{5.10}$$

and $\hat{p}(\boldsymbol{\theta}|\boldsymbol{x}) \leq p(\boldsymbol{\theta}|\boldsymbol{x})$ since $\hat{p}(\boldsymbol{\theta}|\boldsymbol{x}) = p(\boldsymbol{\theta})\hat{r}(\boldsymbol{x}|\boldsymbol{\theta})$. Similarly, Theorem 5.3.3 implies $1 - d(\boldsymbol{\theta}, \boldsymbol{x}) \geq 1 - \hat{d}(\boldsymbol{\theta}, \boldsymbol{x})$ in regions of high prior density, which results in $p(\boldsymbol{\theta}|\boldsymbol{x}) \leq \hat{p}(\boldsymbol{\theta}|\boldsymbol{x})$. Between those two opposite effects, the constraint on $\hat{p}(\boldsymbol{\theta}|\boldsymbol{x})$ that will dominate depends on whether $p(\boldsymbol{\theta}|\boldsymbol{x}) > p(\boldsymbol{\theta})$ or $p(\boldsymbol{\theta}|\boldsymbol{x}) < p(\boldsymbol{\theta})$. If $p(\boldsymbol{\theta}|\boldsymbol{x}) > p(\boldsymbol{\theta})$, then $\hat{p}(\boldsymbol{\theta}|\boldsymbol{x}) \leq p(\boldsymbol{\theta}|\boldsymbol{x})$, whereas if $p(\boldsymbol{\theta}|\boldsymbol{x}) < p(\boldsymbol{\theta})$ then $p(\boldsymbol{\theta}|\boldsymbol{x}) \leq \hat{p}(\boldsymbol{\theta}|\boldsymbol{x})$. Overall, imposing the balancing condition will therefore result in approximate posteriors that lie between the prior and the exact posterior, without being more confident than they should.

Practically, the balancing condition can be targeted through a regularization penalty. For the binary cross-entropy $\mathcal{L}[\hat{d}] \triangleq -\mathbb{E}_{p(\boldsymbol{\theta},\boldsymbol{x})}[\log \hat{d}(\boldsymbol{\theta},\boldsymbol{x})] - \mathbb{E}_{p(\boldsymbol{\theta})p(\boldsymbol{x})}[\log(1-\hat{d}(\boldsymbol{\theta},\boldsymbol{x}))]$ and given that the balancing condition only depends on samples from $p(\boldsymbol{x})p(\boldsymbol{\theta})$ and $p(\boldsymbol{x},\boldsymbol{\theta})$, the full loss functional including the balancing condition can be expressed as

$$\mathcal{L}_b\left[\hat{d}\right] \triangleq \mathcal{L}\left[\hat{d}\right] + \lambda\left(\mathbb{E}_{p(\boldsymbol{\theta})p(\boldsymbol{x})}\left[\hat{d}(\boldsymbol{\theta},\boldsymbol{x})\right] + \mathbb{E}_{p(\boldsymbol{\theta},\boldsymbol{x})}\left[\hat{d}(\boldsymbol{\theta},\boldsymbol{x})\right] - 1\right)^2, \tag{5.11}$$

where λ is a (scalar) hyper-parameter controlling the strength of the balancing condition's contribution. The training procedure is summarized in Algorithm 5.1. Since a classifier is balanced if the balancing condition cancels out, λ could, in principle, be set arbitrarily large. However, as the balancing condition is estimated via Monte Carlo sampling, setting λ to a large value could impair the classifier's learning ability. We found that $\lambda = 100$ works well across many problem domains with varying simulation budgets.

Algorithm 5.1 Training algorithm for Balanced Neural Ratio Estimation (BNRE).

Inputs:	Implicit generative model $p(\boldsymbol{x}\|\boldsymbol{\theta})$ (simulator) and prior $p(\boldsymbol{\theta})$
Outputs:	Approximate classifier $\hat{d}_\psi(\boldsymbol{\theta},\boldsymbol{x})$ parameterized by ψ
hyper-parameters:	Balancing condition strength λ (default = 100) and batch-size n

```
repeat
    Sample data from the joint $\{\boldsymbol{\theta}_i,\ \boldsymbol{x}_i \sim p(\boldsymbol{\theta},\boldsymbol{x}),\ y_i = 1\}_{i=1}^{n/2}$
    Sample data from the marginals $\{\boldsymbol{\theta}_i,\ \boldsymbol{x}_i \sim p(\boldsymbol{\theta})p(\boldsymbol{x}),\ y_i = 0\}_{i=n/2+1}^{n}$
    $\mathcal{L}[\hat{d}_\psi] = -\frac{1}{n}\sum_{i=1}^{n} y_i \log \hat{d}_\psi(\boldsymbol{\theta}_i,\boldsymbol{x}_i) + (1-y_i)\log(1-\hat{d}_\psi(\boldsymbol{\theta}_i,\boldsymbol{x}_i))$
    $\mathcal{B}[\hat{d}_\psi] = \frac{2}{n}\sum_{i=1}^{n/2} \hat{d}_\psi(\boldsymbol{\theta}_i,\boldsymbol{x}_i) + \frac{2}{n}\sum_{i=n/2+1}^{n} \hat{d}_\psi(\boldsymbol{\theta}_i,\boldsymbol{x}_i)$
    $\psi$ = minimizer_step(params=$\psi$, loss=$\mathcal{L}[\hat{d}_\psi] + \lambda(\mathcal{B}[\hat{d}_\psi] - 1)^2$)
until convergence
return $\hat{d}_\psi(\boldsymbol{\theta},\boldsymbol{x})$.
```

5.4 EXPERIMENTS

We start by providing an extensive validation of BNRE on a broad range of benchmarks demonstrating that the proposed method alleviates the problem. Section 5.4.2 follows up with an illustrative demonstration on the behaviour of BNRE and its hyper-parameters. Code is available at `https://github.com/montefiore-ai/balanced-nre`.

5.4.1 *Extensive validation*

SETUP We evaluate the expected coverage of posterior estimators produced by both NRE and BNRE on various problems. Those benchmarks cover a diverse set of problems from particle physics (Weinberg), epidemiology (Spatial SIR), queueing theory (M/G/1), population dynamics (Lotka Volterra, and astronomy (Gravitational Waves). They are representative of real scientific applications of simulation-based inference.

The *SLCP* simulator models a fictive problem with 5 parameters. The observable $\boldsymbol{x}$ is composed of 8 scalars which represent the 2D-coordinates of 4 points. The coordinate of each point is sampled from the same multivariate Gaussian whose mean and covariance matrix are parametrized by $\boldsymbol{\theta}$. We consider an alternative version of the original task (Papamakarios, Sterratt, and Murray, 2019) by inferring the marginal posterior density of 2 of those parameters. In contrast to its original formulation, the likelihood is not tractable due to the marginalization.

The *Weinberg* problem (K. Cranmer, Heinrich, et al., 2017) concerns a simulation of high energy particle collisions $e^+e^- \rightarrow \mu^+\mu^-$. The angular distributions of the particles can be used to measure the Weinberg angle $\boldsymbol{x}$ in the standard model of particle physics. From the scattering angle, we are interested in inferring Fermi's constant $\boldsymbol{\theta}$.

The *Spatial SIR* model (Hermans, Delaunoy, et al., 2022) involves a grid-world of susceptible, infected, and recovered individuals. Based on initial conditions and the infection and recovery rate $\boldsymbol{\theta}$, the model describes the spatial evolution of an infection. The observable $\boldsymbol{x}$ is a snapshot of the grid-world after some fixed amount of time.

M/G/1 (Shestopaloff and Neal, 2014) models a processing and arrival queue. The problem is described by 3 parameters $\boldsymbol{\theta}$ that influence the time it takes to serve a customer, and the time between their arrivals. The observable $\boldsymbol{x}$ is composed of 5 equally spaced quantiles of inter-departure times.

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 $\boldsymbol{\theta}$ which influence the reproduction and mortality rate of the predator and prey species. We infer the marginal posterior of the predator parameters from time series representing the evolution of both populations over time. The specific implementation is based on a Markov Jump Process as in Papamakarios, Sterratt, and Murray (2019).

Gravitational Waves (GW) are ripples in space-time emitted during events such as the collision of two black-holes. They can be detected through interferometry measurements $\boldsymbol{x}$ and convey information about celestial bodies, unlocking new ways to study the universe. We consider inferring the masses $\boldsymbol{\theta}$ of two black-holes colliding through the observation of the gravitational wave as measured by LIGO's dual detectors (LIGO Scientific Collaboration, 2018; Biwer et al., 2019).

Our evaluation considers simulation budgets of increasing size, ranging from $2^{10} = 1024$ to $2^{17} = 131,072$ samples, and credibility levels from 0.05 to 0.95. For every simulation budget, we train 5 posterior estimators for 500 epochs and determine the credible region by evaluating the approximated posterior density function in a discretized and empirically normalized grid of the parameter space with sufficient resolution. The subsequent credible region is the set of parameters whose estimated (and normalized) posterior density is higher or equal to an inclusion threshold fitted to obtain the desired credibility level $1 - \alpha$.

The expected coverage probability is estimated on 10000 unseen samples from the joint $p(\boldsymbol{\theta}, \boldsymbol{x})$, for each considered credibility level.

Table 5.1 summarizes the architectures and hyper-parameters used for each benchmark. The classifier architectures are separated into two parts: the embedding and the head networks. The embedding network ϕ compresses the observable into a set of features. The head network f then uses those features $\phi(\boldsymbol{x})$ concatenated with the parameters $\boldsymbol{\theta}$ to predict the class,

$$\hat{d}(\boldsymbol{\theta}, \boldsymbol{x}) = f(\boldsymbol{\theta}, \phi(\boldsymbol{x})).$$

The learning rate is scheduled during training. Table 5.1 provides the initial learning rates. Those are then divided by 10 each time no improvement was observed on the validation loss for 10 epochs. Further details can be found in the code repository attached to this manuscript.

Table 5.1: Architectures and training hyper-parameters

	SLCP	M/G/1	Weinberg	Lotka-V.	Spatial SIR	GW
Embedding network	None	None	None	CNN	Resnet-18	CNN
Embedding layers	/	/	/	8	/	13
Embedding channels	/	/	/	8	/	16
Convolution type	/	/	/	Conv1D	Conv2D	Dilated Conv1D
Head network	MLP	MLP	MLP	MLP	MLP	MLP
Head layers	6	6	6	3	3	3
Head hidden neurons	256	256	256	128	256	128
Learning rate	0.001	0.001	0.001	0.001	0.001	0.001
Epochs	500	500	500	500	500	500
Batch size	256	256	256	256	256	256

EXPECTED COVERAGE The expected coverage curves and their interpretation are detailed in Figure 5.1. We observe that NRE often produces posterior estimators that are overconfident, especially for small simulation budgets. However, NRE's reliability increases with the availability of training data. By contrast, BNRE produces posterior estimators that are conservative on all benchmarks for all simulation budgets. Figure 5.2 explores the same phenomena through a quantity which we call the coverage AUC, highlighting the effect of the simulation budget. Coverage AUC corresponds to the integrated signed area between the expected coverage curve and the diagonal of a particular simulation. From this quantity it is evident there is a clear distinction between NRE and BNRE with respect to the available simulation budget. Both methods have the tendency to converge towards 0, indicating both methods are moving closer to the Bayes optimal classifier. However, the difference between these methods lies with how this solution is approached. While NRE can approach this limit from both sides, BNRE consistently produces coverage AUC's above 0, corresponding to conservative posterior approximations, and therefore exhibits the desired behaviour (in expectation).

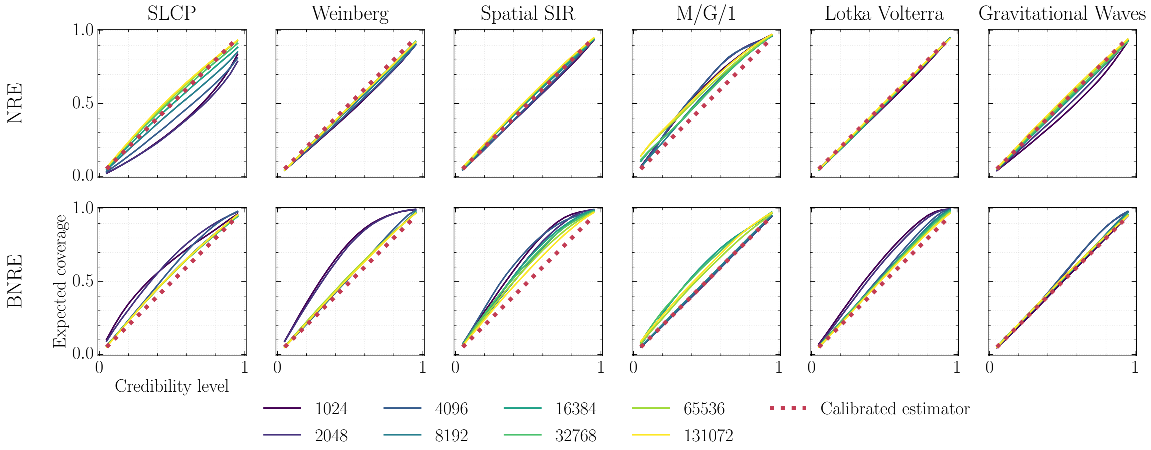

Figure 5.1: Expected coverage for increasing simulation budgets. A perfectly calibrated posterior has an expected coverage probability equal to the nominal coverage probability and hence produces a diagonal line. A conservative estimator has an expected coverage curve at or above the diagonal line, while an overconfident estimator produces curves below the diagonal line. The diagnostic therefore provides an immediate visual interpretation. We observe that NRE can produce overconfident estimators, while BNRE always produces coverage curves above the diagonal line and therefore the desired behaviour: conservative posterior approximations. The means over 5 runs are reported.

STATISTICAL PERFORMANCE In addition to the reliability of the posteriors, we evaluate and compare the statistical performance of the posterior approximations produced by NRE and BNRE. We estimate the expected approximate log posterior density $\mathbb{E}_{p(\boldsymbol{\theta},\boldsymbol{x})}\left[\log \hat{p}(\boldsymbol{\theta}|\boldsymbol{x})\right]$ over a large number of pairs $\boldsymbol{\theta}, \boldsymbol{x}$. It captures how well the posterior surrogates $\hat{p}(\boldsymbol{\theta}|\boldsymbol{x})$ approximate the true posteriors $p(\boldsymbol{\theta}|\boldsymbol{x})$ since $\mathbb{E}_{p(\boldsymbol{\theta},\boldsymbol{x})}\left[\log \hat{p}(\boldsymbol{\theta}|\boldsymbol{x})\right] = -\mathbb{E}_{p(\boldsymbol{x})}\mathrm{KL}\left[p(\boldsymbol{\theta}|\boldsymbol{x}) \mid\mid \hat{p}(\boldsymbol{\theta}|\boldsymbol{x})\right] + \mathbb{E}_{p(x)}\mathbb{E}_{p(\boldsymbol{\theta}|\boldsymbol{x})}\left[\log p(\boldsymbol{\theta}|\boldsymbol{x})\right]$ (Lueckmann, Boelts, et al., 2021).

Figure 5.3 shows our results. We observe that enforcing the balancing condition for $\lambda = 100$ is associated with a loss in statistical performance. However, the loss in statistical performance is eventually recovered by increasing the simulation budget. In fact, practitioners might be inclined to favor reliability over statistical performance (Hermans, Delaunoy, et al., 2022), although it is always a trade-off that depends on the use case. Nevertheless, it is possible to improve the statistical performance by tuning the surrogate, or by increasing the available simulation budget as we have demonstrated.

5.4.2 *In-depth analysis*

In this section, we consider the Weinberg benchmark. The quality of the posterior approximations produced by BNRE is initially discussed with respect to the simulation budget. Afterwards, the effects of the hyper-parameter λ are studied.

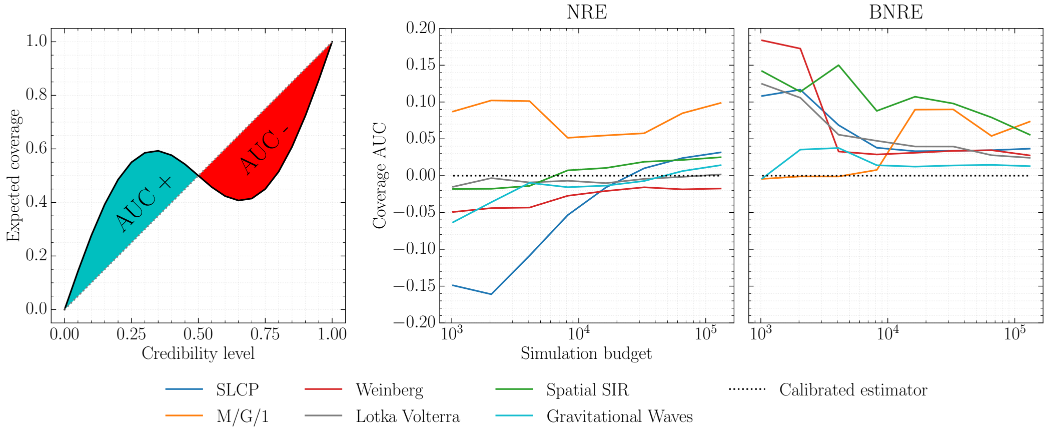

Figure 5.2: Coverage AUC measures the integrated signed area between the expected coverage curve and the diagonal. A perfectly calibrated posterior has an expected coverage probability equal to the nominal coverage probability, producing a diagonal line and has a coverage AUC of 0, as shown on the left subplot. A conservative estimator on the other hand has a coverage AUC larger than 0 and an overconfident estimator smaller than 0. We observe that while NRE can produce coverage AUC both below or above 0, BNRE always produces a coverage AUC larger than 0, implying that its posterior approximations are conservative on average. The means over 5 runs are reported. A complete overview, including standard deviations, are provided in Section 5.7.

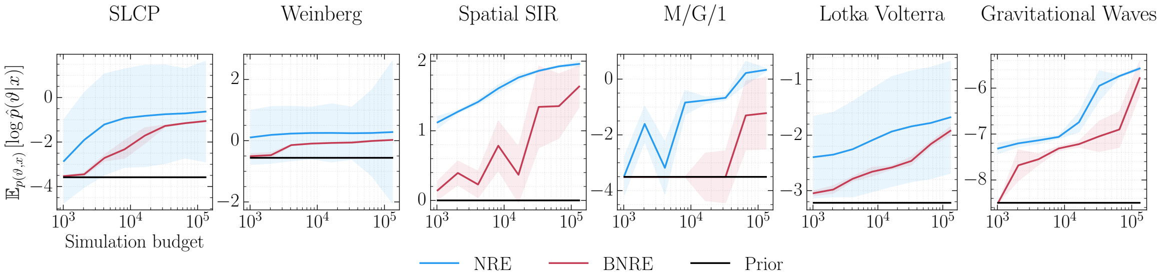

Figure 5.3: Expected value $\mathbb{E}_{p(\boldsymbol{\theta},\boldsymbol{x})}\left[\log \hat{p}(\boldsymbol{\theta}|\boldsymbol{x})\right]$ of the approximate log posterior density of the nominal parameters with respect to the simulation budget. We observe that BNRE produces log posterior densities lower than NRE. This shows that enforcing the balancing condition to have more reliable posterior approximates comes at the price of a small loss in information gain. However, BNRE improves over the prior and eventually converges towards NRE as the simulation budget increases. Solid lines represent the mean over 5 runs and shaded areas represent the standard deviation.

QUALITY ASSESSMENT Because the expected coverage does not capture the quality of an approximation in terms of information gain, we complement our assessment with a bias and variance analysis of the posterior approximations. Let us consider the expected squared error over the approximate posterior $\mathbb{E}_{\hat{p}(\boldsymbol{\theta}|\boldsymbol{x})}\left[(\boldsymbol{\theta}-\boldsymbol{\theta}^*)^2\right]$, where $\boldsymbol{\theta}^*$ is the ground truth parameter value. With $\bar{\boldsymbol{\theta}}(\boldsymbol{x}) = \mathbb{E}_{\hat{p}(\boldsymbol{\theta}|\boldsymbol{x})}\left[\boldsymbol{\theta}\right]$, we decompose $\mathbb{E}_{\hat{p}(\boldsymbol{\theta}|\boldsymbol{x})}\left[(\boldsymbol{\theta}-\boldsymbol{\theta}^*)^2\right]$ as

$$\mathbb{E}_{\hat{p}(\boldsymbol{\theta}|\boldsymbol{x})}\left[(\boldsymbol{\theta}-\bar{\boldsymbol{\theta}}(\boldsymbol{x}))^2\right] + 2\left(\bar{\boldsymbol{\theta}}(\boldsymbol{x})-\boldsymbol{\theta}^*\right)\underbrace{\mathbb{E}_{\hat{p}(\boldsymbol{\theta}|\boldsymbol{x})}\left[(\boldsymbol{\theta}-\bar{\boldsymbol{\theta}}(\boldsymbol{x}))\right]}_{=0} + \mathbb{E}_{\hat{p}(\boldsymbol{\theta}|\boldsymbol{x})}\left[(\bar{\boldsymbol{\theta}}(\boldsymbol{x})-\boldsymbol{\theta}^*)^2\right]$$

$$= \mathbb{E}_{\hat{p}(\boldsymbol{\theta}|\boldsymbol{x})}\left[(\boldsymbol{\theta}-\bar{\boldsymbol{\theta}}(\boldsymbol{x}))^2\right] + (\bar{\boldsymbol{\theta}}(\boldsymbol{x})-\boldsymbol{\theta}^*)^2 .$$

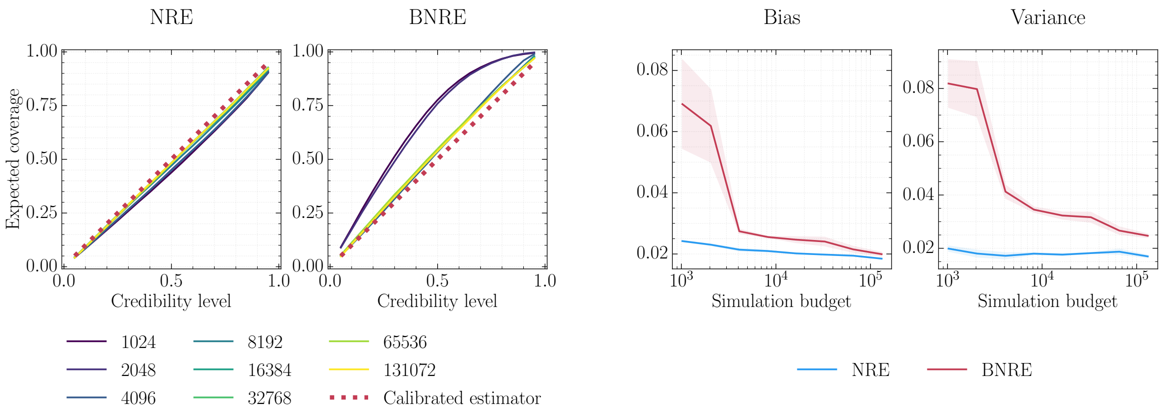

Figure 5.4: Comparison between NRE and BNRE in terms of expected coverage, bias and variance on the Weinberg benchmark. On the left side, the coverage is shown with respect to the simulation budget represented by the colormap. The bias and variance are represented on the right side of the plot. BNRE is run with $\lambda = 100$. Consistent with our previous observations in Figure 5.3, we observe that the gap in both bias and variance reduces as the simulation budget increases. Furthermore, in contrast with NRE, the posterior approximations of BNRE are tending towards being increasingly calibrated while at the same time being conservative. Solid lines represent the mean over 5 runs and shaded areas represent the standard deviation.

The expectation over the joint distribution $p(\boldsymbol{\theta}^*, \boldsymbol{x})$ of the expected squared error can hence be decomposed in a bias term defined as

$$\text{bias}(\hat{p}(\boldsymbol{\theta}|\boldsymbol{x})) \triangleq \mathbb{E}_{p(\boldsymbol{\theta}^*, \boldsymbol{x})}\left[(\bar{\boldsymbol{\theta}}(\boldsymbol{x}) - \boldsymbol{\theta}^*)^2\right], \tag{5.12}$$

which can be interpreted as the expected discrepancy between the nominal value $\boldsymbol{\theta}^*$ and the expected posterior value $\bar{\boldsymbol{\theta}}$. The variance term is

$$\text{variance}(\hat{p}(\boldsymbol{\theta}|\boldsymbol{x})) \triangleq \mathbb{E}_{p(\boldsymbol{\theta}^*, \boldsymbol{x})}\left[\mathbb{E}_{\hat{p}(\boldsymbol{\theta}|\boldsymbol{x})}\left[(\boldsymbol{\theta} - \bar{\boldsymbol{\theta}}(\boldsymbol{x}))^2\right]\right] \tag{5.13}$$

and measures the dispersion of the posterior approximations. Note that these terms differ from the typical statistical bias and variance of point estimators since we are considering full posterior estimators. In particular, the bias of the Bayes optimal model does not necessarily reduce to 0.

Figure 5.4 shows the evolution of expected coverage, bias and variance with respect to the available simulation budget. By taking all plots into consideration with respect to the simulation budget, we can validate that – as suggested by theorems 5.3.2 and 5.3.3 – the increase in expected coverage is tied to an increase in variance. However, this increase comes at the price of a slight increase in bias. Consistent with our previous observations in Figure 5.3, we observe that the gap in both bias and variance reduces as the simulation budget increases. The bias gets close to 0 for high simulation budgets, showing that the bias induced by BNRE vanishes as the simulation-budget increases. A bias and variance analysis for all remaining benchmarks is discussed in Section 5.7.

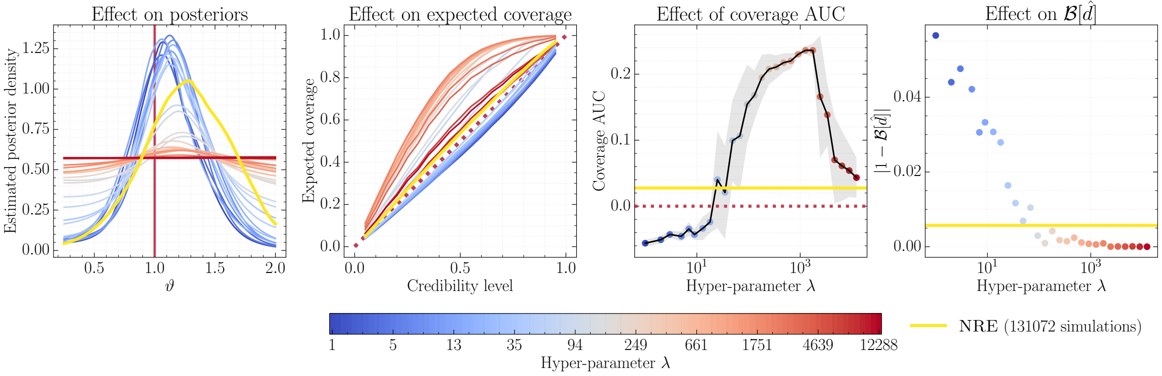

Figure 5.5: Effect of the hyper-parameter λ for a fixed simulation budget of 1024. The first plot from left to right shows the evolution of the approximate posterior for a given observation at a fixed $\boldsymbol{\theta}^*$, indicated by the red vertical line. This approximate posterior is compared to NRE trained on a large simulation budget, shown in yellow and serving as a proxy for the true posterior. The second plot illustrates the empirical expected coverage. The third plot provides a summarized view of the second plot using the coverage AUC as summary statistic. The fourth plot shows that classifiers are becoming increasingly more balanced as λ increases. In addition, the plots show that λ is directly tied to the statistical performance and reliability of the posterior approximations. Classifiers trained with small λ's are associated with (relatively) tight posteriors and overconfident approximations, while classifiers trained with larger values of λ are increasingly more dispersed and conservative until the posterior approximations reduce to the prior due to inflated statistical noise of the Monte Carlo estimation of the balancing condition. Furthermore, the expected coverage plot shows the estimator is almost perfectly calibrated and implicitly balanced. Immediately visible from the various posterior approximations in the leftmost subplot, is the fact that BNRE produces overconfident and biased approximations in the presence of a small simulation budget and a small λ, indicated by their dark blue color. However, the balancing condition can be applied to the underlying estimator to improve its reliability by increasing λ. Ideally, λ should be as small as possible to maximize predictive performance, while at the same time remain sufficiently large to guarantee coverage. From the 3rd subplot from the left, in this particular problem setting, that happens at the point where the coverage AUC transitions from being negative to positive ($\lambda \approx 25.0$).

EFFECTS OF λ Finally, Figure 5.5 shows the effect the hyper-parameter λ on the posterior approximations, their expected coverage and the balancing condition. BNRE is run 5 times for λ ranging from 1 to 2^{15} and for a fixed simulation budget of 1024. Initially, the effect on the posterior approximations is limited for small values of λ. However, once λ increases, the balancing condition forces the posterior approximations to become increasingly dispersed and conservative. Eventually, at least for this specific simulation budget, the posterior approximation reduces to the prior as the balancing condition becomes dominant over the cross-entropy term. Although the global optimum remains unchanged as stated by Theorem 5.3.4, large λ values are likely to impair the training procedure. In particular, a large λ can inflate the statistical noise of the Monte Carlo estimation of the balancing condition and make the classifier $\hat{d}$ degenerate to a classifier that is trivially balanced such as the random classifier $\hat{d}(\boldsymbol{\theta}, \boldsymbol{x}) = 0.5$ for all $\boldsymbol{\theta}, \boldsymbol{x}$. In this case, $\hat{r}(\boldsymbol{x}|\boldsymbol{\theta}) = 1$ for all $\boldsymbol{\theta}, \boldsymbol{x}$ and the approximate posterior degenerates to the prior. This effect is directly evident from Figure 5.5, starting from $\lambda \simeq 1000$. In practice, λ should be sufficiently large such that the approximate classifier is balanced, while maximizing the statistical performance of the posterior estimator. Therefore, we recommend to start with a small value for λ and to gradually increase λ until the posterior estimator becomes conservative. We empirically found $\lambda = 100$ to be a reasonably good default value leading to good performance across all considered benchmarks with various model architectures.

5.5 RELATED WORK

In the Bayesian setting, BNRE improves the reliability of NRE by constraining the classifier hypothesis space to balanced classifiers, which results in more conservative posteriors. Towards the same objective of conservative and reliable approximate posteriors, Hermans, Delaunoy, et al. (2022) have shown empirically that ensembling posterior estimators increases their expected coverage. Since the two solutions are complementary, we suggest that ensembling BNRE is a safe practice to follow. To the best of our knowledge, no other related work exists to make Bayesian simulation-based inference algorithms more conservative and reliable.

In the frequentist setting, K. Cranmer, Pavez, and Louppe (2015) make use of neural ratio estimation to learn likelihood ratio test statistics. They show that the classifier $\hat{d}$ does not need to be exact for the statistic to remain the most powerful, provided that the approximate likelihood ratio is monotonic with exact likelihood ratio. When this is not the case, robust inference remains possible by calibrating the classifier, at the price of a loss in statistical power. Similarly, for frequentist likelihood-free inference, Dalmasso, Izbicki, and Lee (2020) use classifiers to estimate likelihood ratio statistics and propose a procedure for guaranteeing valid hypothesis tests and confidence sets. Finally, Dalmasso, Zhao, et al. (2021) propose a practical procedure for the Neyman construction of confidence sets with finite-sample guarantees of nominal coverage as well as diagnostics that estimate conditional coverage over the entire parameter space.

In this work, we make the assumption that the simulator is well-specified, in the sense that it accurately models the real data generation process. However, this assumption is often violated. To overcome this issue, Generalized Bayesian inference (GBI) extends Bayesian inference by replacing the likelihood term by with arbitrary loss function (Bissiri et al., 2016). Those loss functions can be designed to mitigate specific types of misspecifications and enable robust inference, even with intractable likelihoods (Schmon et al., 2020; Matsubara et al., 2022; Pacchiardi and Dutta, 2021). Power likelihood losses have also been shown to increase robustness to model misspecification (Grünwald and Van Ommen, 2017). It consists in raising the likelihood to a power to control the impact it has over the prior. The lower the power of likelihood, the lower the importance given to the data and the higher the uncertainty of the posterior. It can either be set based on practitioner knowledge or derived from observed data (Holmes and Walker, 2017). Following the same objective, J. W. Miller and D. B. Dunson (2018) introduce coarsened posteriors that condition on a neighborhood of the empirical data distribution rather than on the data itself. This neighborhood is derived from a distance function that, when set to the relative entropy, allows the approximation of coarsened posteriors by a power posterior. Recently, Dellaporta et al. (2022) applied Bayesian non-parametric learning to SBI, making inference with misspecified simulator models both robust and computationally efficient.

5.6 CONCLUSIONS AND FUTURE WORK

In this work, we introduced Balanced Neural Ratio Estimation (BNRE), a variation of neural ratio estimation designed to produce more conservative posterior estimators, even when the likelihood-to-evidence ratio estimator is not computationally faithful. We provide theoretical arguments suggesting that enforcing the balancing condition should lead to more conservative posteriors without sacrificing exactness in the large simulation budget regime. Our theoretical results are experimentally validated on benchmarks of varying complexity.

Nevertheless, our inference algorithm comes with limitations that practitioners should keep in mind. First, we emphasize that theorems 5.3.2 and 5.3.3 hold only in expectation, which means that we cannot provide any guarantee at the level of single inferences. Second, the balancing condition is enforced through a regularization penalty that is not estimated exactly. This implies that the classifier $\hat{d}$ is rarely strictly balanced, although close to be, in which case theorems 5.3.2 and 5.3.3 do not hold. Third, the benefits of BNRE remain to be assessed in high-dimensional parameter spaces. In particular, the posterior density must be evaluated on a discretized grid over the parameter space to compute credibility regions, which currently prohibits the accurate computation of expected coverage in the high-dimensional setting. In conclusion, BNRE should not be viewed as a way to obtain conservative posterior estimators with 100% reliability, but rather as a way to increase the reliability of the posterior estimators with minimal effort and no computational overhead.

Looking forward, the balancing condition could potentially be applied to other simulation-based inference algorithms. Future works could include a generalization to neural posterior estimation (NPE). In fact, the likelihood-to-evidence ratio can be extracted from an approximate posterior by removing its dependence on the prior, $\log \hat{r}(\boldsymbol{x}|\boldsymbol{\theta}) = \log \hat{p}(\boldsymbol{\theta}|\boldsymbol{x}) - \log p(\boldsymbol{\theta})$, which in turn can be expressed as a classifier $\hat{d}(\boldsymbol{\theta}, \boldsymbol{x}) = \sigma(\log \hat{r}(\boldsymbol{x}|\boldsymbol{\theta}))$ on which the balancing condition can be evaluated and enforced. Although our work focuses on amortized approximate inference, the balancing condition could also be applied to sequential inference algorithms to increase their reliability.

Finally, although our initial motivation is framed within the field of simulation-based inference, our theoretical results are directly applicable to any binary classification task by replacing the joint and marginal distributions in the balancing condition with the distributions of the two considered classes. Therefore, it provides an easy-to-implement modification for high-risk classification problems.

5.7 additional results

Figure 5.6 shows the coverage AUC for various simulation budgets. The mean and standard deviation over 5 runs are reported.

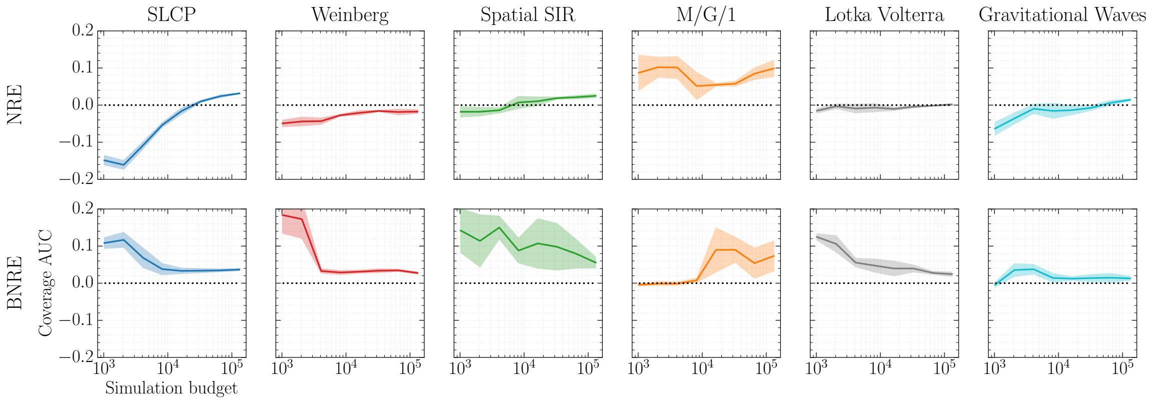

Figure 5.6: Coverage AUC measures the integrated signed area between the expected coverage curve and the diagonal. A perfectly calibrated posterior has an expected coverage probability equal to the nominal coverage probability, producing a diagonal line and has a coverage AUC of 0, as shown on the left subplot. A conservative estimator on the other hand has a coverage AUC larger than 0 and an overconfident estimator smaller than 0. We observe that while NRE can produce coverage AUC both below or above 0, BNRE always produces a coverage AUC larger than 0, implying that its posterior approximations are conservative. Solid lines represent the mean over 5 runs and shaded areas represent the standard deviation.

Figure 5.7 shows the evolution of the bias and variance w.r.t. the simulation budget on a wide variety of benchmarks. We observe that observations made on Weinberg in Section 5.4 generalize to all benchmarks. The variance obtained with BNRE is always higher or equal than the one obtained with NRE as suggested by Theorems 5.3.2 and 5.3.3. In addition, as suggested by Theorem 5.3.4, the bias and variance obtained with BNRE converge, as NRE, to the Bayes optimal solution.

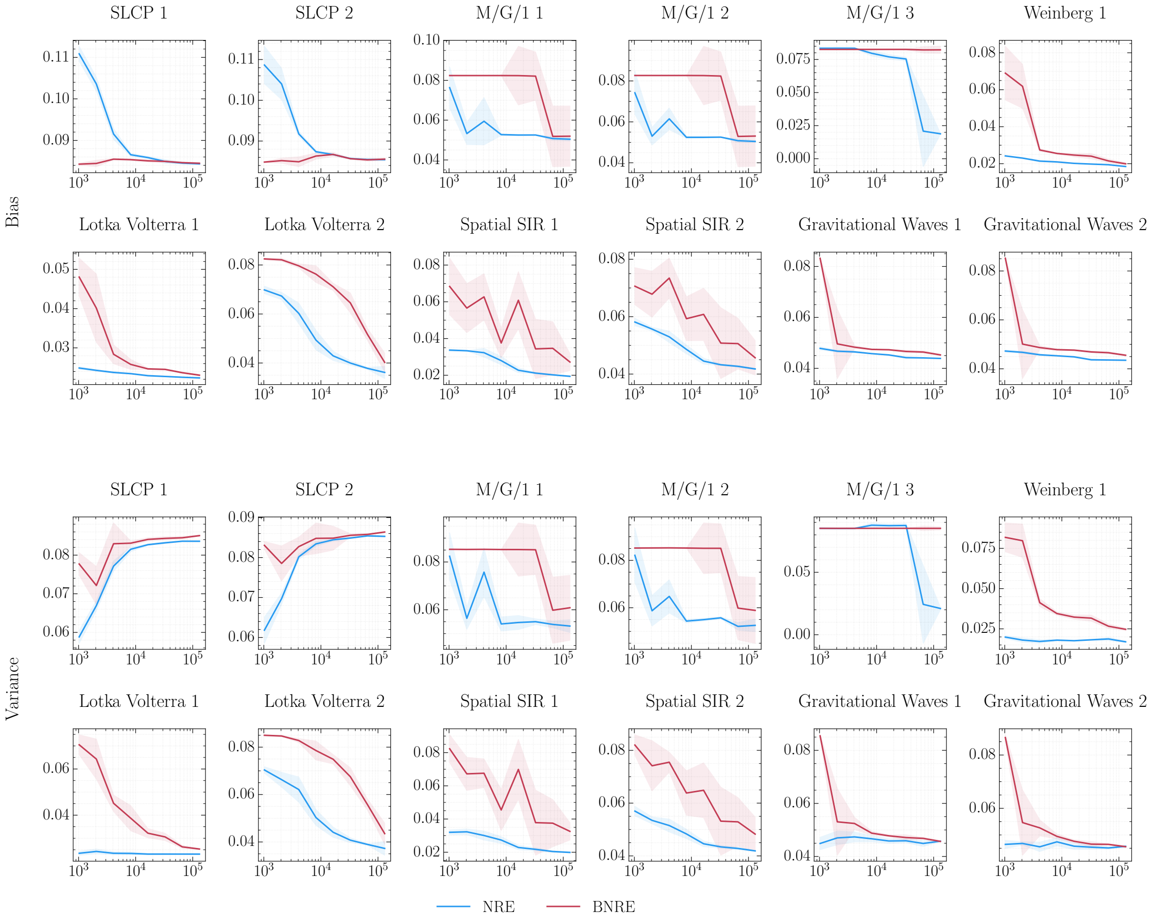

Figure 5.7: Evolution of the bias and variance w.r.t. the simulation budget. The bias and variance are estimated as described in Section 5.4 and are scaled to account for the prior's spread, permitting a direct comparison between the benchmarks. Marginals are considered when dealing with multidimensional parameter spaces. Those are denoted by an index following the benchmark name.

Epilogue

We introduced the Balanced Neural Ratio Estimation (BNRE) algorithm that enforces the balancing condition through regularization. Empirical evaluation shows that enforcing the balancing condition leads to more conservative posterior approximations. This comes at the cost of slightly hurting the predictive performance, as computed by the expected nominal log posterior density. Nevertheless, we showed that enforcing the balancing condition does not prevent attaining the Bayes optimum as the Bayes optimal classifier is always balanced. This is also observed empirically as we see that BNRE achieves similar predictive performance as NRE as the simulation budget increases. In addition to empirical results, we present in Theorems 5.3.2 and 5.3.3 some properties of balanced classifiers that provide intuition about why enforcing balancing leads to more conservative approximations. However, those properties only give intuition and do not guarantee conservativeness as they only hold in expectation, are not statements on the coverage directly, and involve expectations of ratios that are not equal to ratios of expectations. While not guaranteeing conservativeness, BNRE is very easy to implement as it only involves adding a few lines of code in the loss function to switch from NRE to BNRE, and it does not introduce any computational overhead. BNRE is, hence, a simple fix to try to improve the expected coverage of posterior approximations. However, NRE is rarely used in practice as most applicative pieces of work approximate the posterior directly with normalizing flows.

6 BALANCING SIMULATION-BASED INFERENCE

Prologue

This chapter is based on the following publication: *Delaunoy, A.*, Miller, B. K.*, Forré, P., Weniger, C., & Louppe, G. Balancing Simulation-based Inference for Conservative Posteriors. In Fifth Symposium on Advances in Approximate Bayesian Inference.*

Previous work has shown that balancing classifiers in NRE algorithms leads to more conservative posterior approximations. However, the most widely used methods at the time of writing are NPE methods. This piece of work aims to extend the applicability of balancing to other types of algorithms, including NPE and NRE-C.

6.1 INTRODUCTION

Simulation-based inference (SBI) (K. Cranmer, Brehmer, and Louppe, 2020) is a statistical inference framework that solves the inverse problem of identifying which parameter $\boldsymbol{\theta}$ generated observation $\boldsymbol{x}$ by approximating the posterior $p(\boldsymbol{\theta} \mid \boldsymbol{x})$ with a surrogate model $\hat{p}(\boldsymbol{\theta} \mid \boldsymbol{x})$. $\hat{p}$ is constructed from simulated pairs $(\boldsymbol{\theta}, \boldsymbol{x})$ produced by a generative model where the likelihood $p(\boldsymbol{x} \mid \boldsymbol{\theta})$ is only implicitly defined. A classic method to produce samples from the surrogate is a rejection sampling technique called Approximate Bayesian Computation (Sisson, Fan, and Beaumont, 2018). Recently, there has been significant development of algorithms using machine learning for estimating the posterior (Geffner et al., 2022; Sharrock et al., 2022; Papamakarios and Murray, 2016; Lueckmann, Goncalves, et al., 2017; Greenberg et al., 2019; Glöckler et al., 2021), which we call Neural Posterior Estimation (NPE); the likelihood (Papamakarios, Sterratt, and Murray, 2019; Gratton, 2017); or the likelihood-to-evidence ratio (Hermans, Begy, and Louppe, 2020; Durkan, Murray, and Papamakarios, 2020; Benjamin K Miller, Weniger, and Forré, 2022; Benjamin K Miller, Cole, et al., 2021; Thomas et al., 2016; D. Tran et al., 2017), which we call Neural Ratio Estimation (NRE).

It has been demonstrated that the estimated surrogate $\hat{p}(\boldsymbol{\theta} \mid \boldsymbol{x})$ can be more confident than $p(\boldsymbol{\theta} \mid \boldsymbol{x})$ using common SBI algorithms (Hermans, Delaunoy, et al., 2022). This poses a problem for the reliability of SBI in a scientific setting where surrogates must be conservative, i.e. avoid inaccurately excluding parameters at a given credibility level. There has been development in testing for overconfidence using empirical expected coverage and related methods (Talts et al., 2018; Hermans, Delaunoy, et al., 2022; Cook et al., 2006). Lemos, Coogan, et al. (2023) extend expected coverage testing to be a sufficient condition for posterior surrogate correctness, using only samples from the posterior surrogate. In an algorithmic approach to encourage conservativeness, Delaunoy, Hermans, et al. (2022) found that the so-called balance condition regularizes overconfidence in expectation in surrogates trained with NRE (Hermans, Begy, and Louppe, 2020). Similarly, it has been shown that ensembling reduces overconfidence (Hermans, Delaunoy, et al., 2022; Alsing et al., 2019). Zhao et al. (2021) rather test for valid local coverage. Linhart, Gramfort, and P. L. Rodrigues (2022) focuses on normalizing flows and extends this method to the multivariate setting. In a similar fashion (Dalmasso, Izbicki, and Lee, 2020; Dalmasso, Zhao, et al., 2021; Masserano et al., 2023) aim to produce valid frequentist coverage. Cannon et al. (2022) empirically show that model misspecification leads to overconfident posterior approximations and that this can be mitigated using ensembling or sharpness-aware minimization techniques.

CONTRIBUTION After providing some background, we generalize the balancing condition to NPE methods and Contrastive Neural Ratio Estimation (NRE-C) (Benjamin K Miller, Weniger, and Forré, 2022). We provide empirical evidence of the regularizing effect of the balance condition in all of these settings and the first expected coverage tests of NRE-C. Additionally, we relate the balance condition to the χ^2-divergence. Code is available at https://github.com/ADelau/balancing_sbi.

6.2 BACKGROUND

POSTERIOR ESTIMATION (NPE) A density estimator $q_{\boldsymbol{w}}(\boldsymbol{\theta} \mid \boldsymbol{x})$ with weights $\boldsymbol{w}$, such as a mixture density network (Bishop, 1994) or normalizing flow (Papamakarios, Nalisnick, et al., 2019), approximates $p(\boldsymbol{\theta} \mid \boldsymbol{x})$ when the expected Kullback-Leibler divergence

$$\mathbb{E}_{p(\boldsymbol{x})} \left[\mathrm{KL}(p(\boldsymbol{\theta} \mid \boldsymbol{x}) \mid q_{\boldsymbol{w}}(\boldsymbol{\theta} \mid \boldsymbol{x}) \right], \tag{6.1}$$

is minimized. In NPE, the surrogate model is directly $\hat{p}(\boldsymbol{\theta} \mid \boldsymbol{x}) := q_{\boldsymbol{w}}(\boldsymbol{\theta} \mid \boldsymbol{x})$.

RATIO ESTIMATION (NRE) The likelihood-to-evidence ratio $r(\boldsymbol{\theta}, \boldsymbol{x}) := \frac{p(\boldsymbol{x} \mid \boldsymbol{\theta})}{p(\boldsymbol{x})} = \frac{p(\boldsymbol{\theta}, \boldsymbol{x})}{p(\boldsymbol{\theta})p(\boldsymbol{x})}$ is estimated through a supervised learning task using classifier $\varpi(y = 1 \mid \boldsymbol{\theta}, \boldsymbol{x})$. The target conditional distribution $\pi(y = 1 \mid \boldsymbol{\theta}, \boldsymbol{x})$ comes from $\pi(\boldsymbol{\theta}, \boldsymbol{x}, y) := \pi(\boldsymbol{\theta}, \boldsymbol{x} \mid y)\pi(y)$ where the marginals are set to $\pi(y = 0) := \pi(y = 1) := \frac{1}{2}$ and the remaining conditional is defined as

$$\pi(\boldsymbol{\theta}, \boldsymbol{x} \mid y) := \begin{cases} p(\boldsymbol{\theta})p(\boldsymbol{x}) & y = 0 \\ p(\boldsymbol{\theta}, \boldsymbol{x}) & y = 1 \end{cases}. \tag{6.2}$$

The classifier $\varpi(y = 1 \,|\, \boldsymbol{\theta}, \boldsymbol{x}) := \sigma \circ f_{\boldsymbol{w}}(\boldsymbol{\theta}, \boldsymbol{x})$ is parameterized by a neural network $f_{\boldsymbol{w}}(\boldsymbol{\theta}, \boldsymbol{x})$ and σ is the sigmoid. $\varpi(y = 1 \,|\, \boldsymbol{\theta}, \boldsymbol{x})$ approximates $\pi(y = 1 \,|\, \boldsymbol{\theta}, \boldsymbol{x})$ when the NRE loss

$$\mathbb{E}_{\pi(\boldsymbol{\theta}, \boldsymbol{x})} \left[\mathrm{KL}(\pi(y \,|\, \boldsymbol{\theta}, \boldsymbol{x}) \,\|\, \varpi(y \,|\, \boldsymbol{\theta}, \boldsymbol{x})) \right], \tag{6.3}$$

is minimized. Let $\hat{p}(\boldsymbol{\theta} \,|\, \boldsymbol{x}) := \frac{\exp \circ f_{\boldsymbol{w}}(\boldsymbol{\theta}, \boldsymbol{x})}{Z_{\boldsymbol{w}}(\boldsymbol{x})} p(\boldsymbol{\theta})$, with $Z_{\boldsymbol{w}}(\boldsymbol{x}) := \int \exp \circ f_{\boldsymbol{w}}(\boldsymbol{\theta}, \boldsymbol{x}) p(\boldsymbol{\theta}) d\boldsymbol{\theta}$, define the surrogate model. When the loss is zero $f_{\boldsymbol{w}}(\boldsymbol{\theta}, \boldsymbol{x}) = \log r(\boldsymbol{\theta}, \boldsymbol{x})$, recovering the posterior.

CONTRASTIVE NEURAL RATIO ESTIMATION (NRE-C) (Benjamin K Miller, Weniger, and Forré, 2022) introduce NRE-C, an algorithm featuring a flexible, multiclass distribution $\tilde{\pi}(y)$ for $y = 0, 1, \ldots, K$.

We let marginal distribution $\tilde{\pi}(y)$ with $y \in \{0, 1, \ldots, K\}$ have probabilities $\tilde{\pi}(y = k) := \tilde{\pi}_K$ for all $k \geq 1$ and $\tilde{\pi}(y = 0) := \tilde{\pi}_0$ which yields the relationship $\tilde{\pi}_0 = 1 - K\tilde{\pi}_K$. The remaining conditional is

$$\tilde{\pi}(\boldsymbol{\Theta}, \boldsymbol{x} \,|\, y = k) := \begin{cases} p(\boldsymbol{\theta}_1) \cdots p(\boldsymbol{\theta}_K) p(\boldsymbol{x}) & k = 0 \\ p(\boldsymbol{\theta}_1) \cdots p(\boldsymbol{\theta}_K) p(\boldsymbol{x} \,|\, \boldsymbol{\theta}_k) & k = 1, \ldots, K \end{cases}, \tag{6.4}$$

where $\boldsymbol{\Theta} := (\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_K)$ are contrastive parameters, sampled from the prior $p(\boldsymbol{\theta})$. We fit the variational, multiclass classifier

$$\tilde{\varpi}(y = k \,|\, \boldsymbol{\Theta}, \boldsymbol{x}) := \begin{cases} \frac{K}{K + \sum_{i=1}^{K} \exp \circ h_{\boldsymbol{w}}(\boldsymbol{\theta}_i, \boldsymbol{x})} & k = 0 \\ \frac{\exp \circ h_{\boldsymbol{w}}(\boldsymbol{\theta}_k, \boldsymbol{x}))}{K + \sum_{i=1}^{K} \exp \circ h_{\boldsymbol{w}}(\boldsymbol{\theta}_i, \boldsymbol{x})} & k = 1, \ldots, K \end{cases} \tag{6.5}$$

by minimizing $\mathbb{E}_{\tilde{\pi}(\boldsymbol{\Theta}, \boldsymbol{x})} \left[\mathrm{KL}(\tilde{\pi}(y \,|\, \boldsymbol{\Theta}, \boldsymbol{x}) \,|\, \tilde{\varpi}(y \,|\, \boldsymbol{\Theta}, \boldsymbol{x})) \right]$, where $h_{\boldsymbol{w}}$ is a neural network parameterized by weights $\boldsymbol{w}$. We write the loss function out explicitly:

$$\begin{aligned} L[\tilde{\varpi}] := &-\frac{1}{1+\gamma} \mathbb{E}_{\tilde{\pi}(\boldsymbol{\Theta}, \boldsymbol{x} \,|\, y=0)} \left[\log \tilde{\varpi}(y = 0 \,|\, \boldsymbol{\Theta}, \boldsymbol{x}) \right] \\ &- \frac{\gamma}{1+\gamma} \mathbb{E}_{\tilde{\pi}(\boldsymbol{\Theta}, \boldsymbol{x} \,|\, y=K)} \left[\log \tilde{\varpi}(y = K \,|\, \boldsymbol{\Theta}, \boldsymbol{x}) \right], \end{aligned} \tag{6.6}$$

where we introduced $\gamma := \frac{\tilde{\pi}(y \geq 1)}{\tilde{\pi}(y=0)} = \frac{K\tilde{\pi}_K}{\tilde{\pi}_0}$. There are only two terms in this sum because we exploited symmetries in the conditionals $\tilde{\pi}(\boldsymbol{\Theta}, \boldsymbol{x} \,|\, y = k)$ when $k \neq 0$. We define the surrogate model as $\hat{p}(\boldsymbol{\theta} \,|\, \boldsymbol{x}) := \frac{\exp \circ h_{\boldsymbol{w}}(\boldsymbol{\theta}, \boldsymbol{x})}{Z_{\boldsymbol{w}}(\boldsymbol{x})} p(\boldsymbol{\theta})$ with $Z_{\boldsymbol{w}}(\boldsymbol{x}) := \int \exp \circ h_{\boldsymbol{w}}(\boldsymbol{\theta}, \boldsymbol{x}) p(\boldsymbol{\theta}) \, d\boldsymbol{\theta}$.

When the objective (6.6) becomes zero, the NRE-C surrogate model recovers the posterior. NRE-C represents a strict generalization of classifier-based, likelihood-to-evidence ratio estimation methods (Hermans, Begy, and Louppe, 2020; Durkan, Murray, and Papamakarios, 2020). In the experiments we took $\gamma := 1$ and $K := 5$.

CONSERVATIVE SURROGATES Undesirably, simulation-based inference algorithms can produce overconfident surrogate models (Hermans, Delaunoy, et al., 2022). We define overconfidence in terms of the $(1-\alpha)$ expected coverage probability of the posterior surrogate $\hat{p}(\boldsymbol{\theta} \,|\, \boldsymbol{x})$,

$$1 - \hat{\alpha}[\hat{p}; \alpha] := \mathbb{E}_{p(\boldsymbol{\theta}, \boldsymbol{x})} \left[\mathbb{1}(\boldsymbol{\theta} \in \Theta_{\hat{p}(\boldsymbol{\theta} \,|\, \boldsymbol{x})}(1-\alpha)) \right], \tag{6.7}$$

where $\mathbb{1}$ is an indicator function, and $\Theta_{\hat{p}(\boldsymbol{\theta} \,|\, \boldsymbol{x})}(1-\alpha)$ yields the $(1-\alpha)$ highest posterior density region (HPDR) of $\hat{p}(\boldsymbol{\theta} \,|\, \boldsymbol{x})$ with $\alpha \in [0,1]$. The quantity $(1-\alpha)$ is called the nominal coverage probability. When $\exists \alpha' : 1 - \hat{\alpha}[\hat{p}; \alpha'] < 1 - \alpha'$, we say that $\hat{p}(\boldsymbol{\theta} \,|\, \boldsymbol{x})$ is overconfident. Overconfidence is problematic because the surrogate tends to exclude parameter values that are actually plausible at the considered credibility level. On the other hand, extremely underconfident surrogates are not informative. Although there is a tradeoff, scientific applications take a cautious approach by favoring underconfidence. Therefore, we encourage conservative surrogates at credibility level α', which have $1 - \hat{\alpha}[\hat{p}; \alpha'] \geq 1 - \alpha'$. We imprecisely define the relative conservativeness of one posterior to another: One surrogate is more conservative than another when there are "more credibility levels" at which it is conservative than the other.

BALANCE CONDITION In an effort to produce conservative surrogates, Delaunoy, Hermans, et al. (2022) introduced the balance condition. It holds for any classifier $\varpi(y = 1 \,|\, \boldsymbol{\theta}, \boldsymbol{x})$ that satisfies

$$1 = \mathbb{E}_{p(\boldsymbol{\theta})p(\boldsymbol{x})} \left[\varpi(y = 1 \,|\, \boldsymbol{\theta}, \boldsymbol{x}) \right] + \mathbb{E}_{p(\boldsymbol{\theta}, \boldsymbol{x})} \left[\varpi(y = 1 \,|\, \boldsymbol{\theta}, \boldsymbol{x}) \right]. \tag{6.8}$$

Delaunoy, Hermans, et al. (2022) show that $\mathbb{E}_{p(\boldsymbol{\theta}, \boldsymbol{x})} \left[\frac{\pi(y=1 \,|\, \boldsymbol{\theta}, \boldsymbol{x})}{\varpi(y=1 \,|\, \boldsymbol{\theta}, \boldsymbol{x})} \right] \geq 1$ and $\mathbb{E}_{p(\boldsymbol{\theta})p(\boldsymbol{x})} \left[\frac{\pi(y=0 \,|\, \boldsymbol{\theta}, \boldsymbol{x})}{\varpi(y=0 \,|\, \boldsymbol{\theta}, \boldsymbol{x})} \right] \geq 1$ for balanced classifiers. In expectation, it implies the balanced classifier's probabilities $\varpi(y \,|\, \boldsymbol{\theta}, \boldsymbol{x})$ are closer to uniform than the target, encouraging $\hat{r}(\boldsymbol{\theta}, \boldsymbol{x})$ to be closer to 1. This brings to surrogate closer to the prior, $p(\boldsymbol{\theta})$, which is conservative.

6.3 EXTENDING THE BALANCE CONDITION

We clarify and generalize the balance criterion: Identifying it with the χ^2-divergence, and applying it to models that can evaluate the approximate posterior density.

BALANCE AS DIVERGENCE We encourage balance during training by regularizing the loss with a Lagrange multiplier; penalizing solutions that do not satisfy the balance criterion

$$B[\varpi] := B(\boldsymbol{w}) := \Big(\mathbb{E}_{p(\boldsymbol{\theta})p(\boldsymbol{x})} \left[\varpi(y = 1 \,|\, \boldsymbol{\theta}, \boldsymbol{x}) \right] + \mathbb{E}_{p(\boldsymbol{\theta}, \boldsymbol{x})} \left[\varpi(y = 1 \,|\, \boldsymbol{\theta}, \boldsymbol{x}) \right] - 1 \Big)^2, \tag{6.9}$$

where $\boldsymbol{w}$ are the classifier weights. The effects of balance are clearer when (6.9) is rewritten in the form of a χ^2-divergence. The χ^2-divergence (Sason and Verdú, 2016) is defined as

$$\chi^2(\pi(y) \,\|\, \varpi(y)) := \int \left(\frac{\varpi(y)}{\pi(y)} - 1 \right)^2 \pi(y)\, dy, \tag{6.10}$$

where $\varpi(y) := \int \varpi(y \,|\, \boldsymbol{\theta}, \boldsymbol{x}) \pi(\boldsymbol{\theta}, \boldsymbol{x}) d\boldsymbol{\theta}\, d\boldsymbol{x}$ is the marginal classifier. With the following steps, we identify that $B[\varpi] = \chi^2(\pi(y) \,\|\, \varpi(y))$.

$$\begin{aligned}
\chi^2(\pi(y) \,\|\, \varpi(y)) &:= \int \left(\frac{\varpi(y)}{\pi(y)} - 1 \right)^2 \pi(y)\, dy \\
&= \left(\frac{\varpi(y=0)}{\pi(y=0)} - 1 \right)^2 \pi(y=0) + \left(\frac{\varpi(y=1)}{\pi(y=1)} - 1 \right)^2 \pi(y=1) \\
&= \frac{(\varpi(y=0) - \pi(y=0))^2}{\pi(y=0)} + \frac{(\varpi(y=1) - \pi(y=1))^2}{\pi(y=1)} \\
&= \frac{(\varpi(y=1) - \pi(y=1))^2}{\pi(y=0)} + \frac{(\varpi(y=1) - \pi(y=1))^2}{\pi(y=1)} \\
&= (\varpi(y=1) - \pi(y=1))^2 \left(\frac{1}{\pi(y=0)} + \frac{1}{\pi(y=1)} \right) \\
&= 4(\varpi(y=1) - \pi(y=1))^2 \\
&= (2\varpi(y=1) - 1)^2 \\
&= \left(2 \int \varpi(y=1 \,|\, \boldsymbol{\theta}, \boldsymbol{x}) \pi(\boldsymbol{\theta}, \boldsymbol{x}) d\boldsymbol{\theta}\, d\boldsymbol{x} - 1 \right)^2 \\
&= \left(2 \int \varpi(y=1 \,|\, \boldsymbol{\theta}, \boldsymbol{x}) \left(\int \pi(\boldsymbol{\theta}, \boldsymbol{x} \,|\, y) \pi(y) dy \right) d\boldsymbol{\theta}\, d\boldsymbol{x} - 1 \right)^2 \\
&= \left(\int \varpi(y=1 \,|\, \boldsymbol{\theta}, \boldsymbol{x}) (\pi(\boldsymbol{\theta}, \boldsymbol{x} \,|\, y=0) + \pi(\boldsymbol{\theta}, \boldsymbol{x} \,|\, y=1)) d\boldsymbol{\theta}\, d\boldsymbol{x} - 1 \right)^2 \\
&= \left(\mathbb{E}_{p(\boldsymbol{\theta})p(\boldsymbol{x})} \left[\varpi(y=1 \,|\, \boldsymbol{\theta}, \boldsymbol{x}) \right] + \mathbb{E}_{p(\boldsymbol{\theta}, \boldsymbol{x})} \left[\varpi(y=1 \,|\, \boldsymbol{\theta}, \boldsymbol{x}) \right] - 1 \right)^2
\end{aligned} \tag{6.11}$$

where we used $\pi(y=0) = \pi(y=1) = \frac{1}{2}$ and $\varpi(\boldsymbol{\theta}, \boldsymbol{x}) := \pi(\boldsymbol{\theta}, \boldsymbol{x})$. Therefore, a balanced classifier satisfies $\varpi(y) = \pi(y)$, i.e. balancing regularizes the marginal classifier towards the target distribution for y.

BALANCE CRITERION FOR ALTERNATIVE MODELS The balance criterion regularizes marginal distribution $\varpi(y)$; this makes sense for NRE which defines $\varpi(y)$ and target marginal $\pi(y)$. However, we are interested in regularizing objectives $L(\boldsymbol{w})$ that either do not introduce a binary auxiliary variable y, or use an alternative. In order to apply the balance criterion, we propose to use the same target distribution $\pi(\boldsymbol{\theta}, \boldsymbol{x}, y)$ as NRE does and define a classifier in terms of the variational (unnormalized) posterior approximant $\hat{q}_{\boldsymbol{w}}(\boldsymbol{\theta} \,|\, \boldsymbol{x})$. We approximate $r(\boldsymbol{\theta}, \boldsymbol{x}) := \frac{p(\boldsymbol{\theta}, \boldsymbol{x})}{p(\boldsymbol{\theta})p(\boldsymbol{x})} = \frac{p(\boldsymbol{\theta} \,|\, \boldsymbol{x})}{p(\boldsymbol{\theta})}$ with $\frac{\hat{q}_{\boldsymbol{w}}(\boldsymbol{\theta} \,|\, \boldsymbol{x})}{p(\boldsymbol{\theta})}$ which yields the classifier $\varpi(y=1 \,|\, \boldsymbol{\theta}, \boldsymbol{x}; \hat{q}_{\boldsymbol{w}}) := \frac{\hat{q}_{\boldsymbol{w}}(\boldsymbol{\theta} \,|\, \boldsymbol{x})/p(\boldsymbol{\theta})}{1 + \hat{q}_{\boldsymbol{w}}(\boldsymbol{\theta} \,|\, \boldsymbol{x})/p(\boldsymbol{\theta})}$ and regularize $L(\boldsymbol{w})$ with $B(\boldsymbol{w})$ and Lagrange multiplier λ

$$L(\boldsymbol{w}) + \lambda B(\boldsymbol{w}). \tag{6.12}$$

The main contribution is reformulating $B(\boldsymbol{w})$ to be expressed in terms of $\hat{q}_{\boldsymbol{w}}$, which generalizes the balance condition to models which allow for approximate density evaluation! We consider losses $L(\boldsymbol{w})$ that go to zero if and only if $\hat{q}_{\boldsymbol{w}}(\boldsymbol{\theta} \,|\, \boldsymbol{x}) = p(\boldsymbol{\theta} \,|\, \boldsymbol{x})$. When this is true, the balance condition (6.8) also holds since $B(\boldsymbol{w})$ becomes zero:

$$\begin{aligned} B(\boldsymbol{w}) &:= \left(\int (\pi(\boldsymbol{\theta}, \boldsymbol{x} \,|\, y = 0) + \pi(\boldsymbol{\theta}, \boldsymbol{x} \,|\, y = 1)) \varpi(y = 1 \,|\, \boldsymbol{\theta}, \boldsymbol{x}; \hat{q}_{\boldsymbol{w}}) d\boldsymbol{\theta}\, d\boldsymbol{x} - 1 \right)^2 \\ &= \left(\int (p(\boldsymbol{\theta}) p(\boldsymbol{x}) + p(\boldsymbol{\theta}, \boldsymbol{x})) \frac{\hat{q}_{\boldsymbol{w}}(\boldsymbol{\theta} \,|\, \boldsymbol{x}) / p(\boldsymbol{\theta})}{1 + \hat{q}_{\boldsymbol{w}}(\boldsymbol{\theta} \,|\, \boldsymbol{x}) / p(\boldsymbol{\theta})} d\boldsymbol{\theta}\, d\boldsymbol{x} - 1 \right)^2 \end{aligned} \tag{6.13}$$

That means for loss $L(\boldsymbol{w}) = 0$ the regularized version (6.12) also goes to zero; just like BNRE. As advised in Delaunoy, Hermans, et al. (2022), we use $\lambda = 100$. In practice, this value should be tuned for best performance.

BALANCED NEURAL POSTERIOR ESTIMATION (BNPE) We propose BNPE which regularizes NPE's objective (6.1) with the balance criterion to train a normalized density estimator. We have $\hat{q}_{\boldsymbol{w}}(\boldsymbol{\theta} \,|\, \boldsymbol{x}) := q_{\boldsymbol{w}}(\boldsymbol{\theta} \,|\, \boldsymbol{x})$. The corresponding classifier becomes $\varpi(y = 1 \,|\, \boldsymbol{\theta}, \boldsymbol{x}; \hat{q}_{\boldsymbol{w}}) := \frac{q_{\boldsymbol{w}}(\boldsymbol{\theta} \,|\, \boldsymbol{x}) / p(\boldsymbol{\theta})}{1 + q_{\boldsymbol{w}}(\boldsymbol{\theta} \,|\, \boldsymbol{x}) / p(\boldsymbol{\theta})}$. Training minimizes (6.12), substituting the appropriate loss and classifier.

BALANCED CONTRASTIVE NEURAL RATIO ESTIMATION (BNRE-C) We propose BNRE-C which regularizes NRE-C's objective (6.6) with the balance criterion to train a ratio estimator. The density estimator is $\hat{q}_{\boldsymbol{w}}(\boldsymbol{\theta} \,|\, \boldsymbol{x}) := \exp \circ h_{\boldsymbol{w}}(\boldsymbol{\theta}, \boldsymbol{x}) p(\boldsymbol{\theta})$. The corresponding classifier becomes $\varpi(y = 1 \,|\, \boldsymbol{\theta}, \boldsymbol{x}; \hat{q}) := \frac{\exp \circ h_{\boldsymbol{w}}(\boldsymbol{\theta}, \boldsymbol{x}) p(\boldsymbol{\theta}) / p(\boldsymbol{\theta})}{1 + \exp \circ h_{\boldsymbol{w}}(\boldsymbol{\theta}, \boldsymbol{x}) p(\boldsymbol{\theta}) / p(\boldsymbol{\theta})} = \sigma \circ h_{\boldsymbol{w}}(\boldsymbol{\theta}, \boldsymbol{x})$. Training minimizes (6.12), substituting the appropriate loss and classifier.

6.4 REFINING BALANCED NEURAL POSTERIOR ESTIMATION

We observe that BNPE sometimes struggles to minimize the balance criterion for low simulation budgets. In this section, we provide a way to simplify learning.

6.4.1 *Improving Balanced Neural Posterior Estimation*

Let us first observe that the prior $p(\boldsymbol{\theta})$ is balanced:

$$\int (\pi(\boldsymbol{\theta}, \boldsymbol{x} \,|\, y = 0) + \pi(\boldsymbol{\theta}, \boldsymbol{x} \,|\, y = 1)) \varpi(y = 1 \,|\, \boldsymbol{\theta}, \boldsymbol{x}) d\boldsymbol{\theta}\, d\boldsymbol{x} \tag{6.14}$$

$$= \int (p(\boldsymbol{\theta}) p(\boldsymbol{x}) + p(\boldsymbol{\theta}, \boldsymbol{x})) \frac{\hat{p}(\boldsymbol{\theta} | \boldsymbol{x}) / p(\boldsymbol{\theta})}{1 + \hat{p}(\boldsymbol{\theta} | \boldsymbol{x}) / p(\boldsymbol{\theta})} d\boldsymbol{\theta}\, d\boldsymbol{x} \tag{6.15}$$

$$= \int (p(\boldsymbol{\theta}) p(\boldsymbol{x}) + p(\boldsymbol{\theta}, \boldsymbol{x})) \frac{p(\boldsymbol{\theta}) / p(\boldsymbol{\theta})}{1 + p(\boldsymbol{\theta}) / p(\boldsymbol{\theta})} d\boldsymbol{\theta}\, d\boldsymbol{x} = 1 \tag{6.16}$$

BNRE can easily model the prior as this is achieved for $\varpi(y = 1 \,|\, \boldsymbol{\theta}, \boldsymbol{x}) = 0.5 \quad \forall \boldsymbol{\theta}, \boldsymbol{x}$. In opposition, BNPE has to explicitly learn a balanced distribution. In order to ease the task of learning a balanced distribution, we propose to initialize the posterior surrogate to the prior distribution. We will refer to this algorithm as Initialized Balanced Neural Posterior Estimation (BNPE Init).

6.4.2 *Intializing neural spline flows to the prior distribution*

In this work, we use Neural Spline Flows (Durkan, Bekasov, et al., 2019). We here discuss how to initialize this architecture to the prior. A normalizing flow models a complex distribution as a sequence of transformations of some base distribution. Consequently, the flow models the prior in the two following scenarios. The transformations are all identity transformations and the base distribution is the prior, or the transformations are all identity transformations and a fixed transformation that maps the base distribution to the prior is added at the end of the flow.

In the following, we describe how to obtain the two core components: identity transformations and a transformation that maps the base distribution to the prior. We then discuss the advantages and drawbacks of the two methods. In the experiments, we used a transformation from the base distribution to the prior.

NEURAL SPLINE TRANSFORMATIONS (Durkan, Bekasov, et al., 2019) are transformations defined by K rational-quadratic functions, with boundaries set by $K+1$ knots denoted by $(x^{(k)}, y^k)_{k=0}^{K}$. The boundaries are defined by $(x^{(0)}, y^0) = (-B, -B)$ and $(x^{(K)}, y^K) = (B, B)$. The knots are parametrized by two vectors θ^w and θ^h of length K. Those vectors are passed through a softmax and multiplied by $2B$ to define the bins' width and height.

The rational-quadratic in the k^{th} bin is defined as

$$\frac{\alpha^{(k)}(\xi)}{\beta^{(k)}(\xi)} = y^{(k)} + \frac{(y^{(k+1)} - y^{(k)})[s^{(k)}\xi^2 + \delta^{(k)}\xi(1-\xi)]}{s^{(k)} + [\delta^{(k+1)} + \delta^{(k)} - 2s^{(k)}]\xi(1-\xi)}. \tag{6.17}$$

The terms $\left\{\delta^{(k)}\right\}_{k=1}^{K-1}$ define the derivatives at the internal points and are parametrized by the vector θ^d of length $K-1$. The other terms are defined as $s_k = (y^{k+1} - y^k)/(x^{k+1} - x^k)$ and $\xi = (x - x^k)/(x^{k+1} - x^k)$.

INITIALIZING TRANSFORMATIONS TO IDENTITY To let the spline transformation be close to identity, we need to initialize the knots such that all the weights and widths are the same. This can be done by initializing the vectors θ^w and θ^h to zeros for all conditionings. In addition, all $s^{(k)}$ are then equal to 1. We also need $\delta^{(k)} = 1,\ \forall k$. In the implementation used (Rozet, 2022), the vector θ^d models the log derivatives and hence must be initialized to a vector full of zeros.

To achieve the identity transform, the outputs of the neural network θ^h, θ^w and θ^d must be zeros at initialization for all conditionings. This is achieved when biases and weights are all set 0. However, this initialization is not optimal for learning via stochastic gradient descent, so we aim for a trade-off between ease of training and closeness to the prior. We initialize the neural network using some standard initialization, set the biases to 0 and divide the weights by 5 to obtain a function close to an identity transformation.

MAPPING THE BASE DISTRIBUTION TO THE PRIOR In our case, the base distribution is a normal distribution $\mathcal{N}(0,1)$ and the priors are uniform in all the benchmarks considered. Therefore, we need to define a mapping from a Uniform distribution $\mathcal{U}(a,b)$, which p.d.f. is denoted by $p_u(u)$, to a normal distribution $\mathcal{N}(0,1)$, which p.d.f. is denoted by $p_n(n)$ and c.d.f denoted $F_n(n)$. Let us first consider an intermediate mapping to a Uniform distribution $\mathcal{U}(0,1)$ which p.d.f is denoted $p_{\tilde{u}}(\tilde{u})$. Going from u to $\tilde{u}$ can be achieved with the following transformation

$$\tilde{u} = \frac{u+a}{b-a}. \tag{6.18}$$

The Jacobian linked to this transformation is then

$$\frac{1}{b-a} \tag{6.19}$$

Going from $\tilde{u}$ to n can be achieved with the following transformation

$$\tilde{u} = F_n(n) \Leftrightarrow n = F_n^{-1}(\tilde{u}). \tag{6.20}$$

The Jacobian linked to this transformation is then

$$\begin{aligned} \left|(F_n^{-1})'(\tilde{u})\right| &= \left|\frac{1}{F_n'(F_n^{-1}(\tilde{u}))}\right| &(6.21)\\ &= \frac{1}{p_n(F_n^{-1}(\tilde{u}))}. &(6.22)\end{aligned}$$

COMPARISON OF THE DIFFERENT INITIALIZATION SCHEMES We have considered two initialization schemes: using the prior as base distribution or adding a transformation that maps the base distribution to the prior distribution. Using the prior distribution as base distribution has the advantage to be easy to implement as it does not require defining a mapping between both distributions. However, we have empirically observed that modifying the base distribution can lead to worse performance. We hypothesize that this is due to the fact that all the considered benchmarks have a uniform prior while flows work better with a Gaussian base. Let us note that using such a transformation could be beneficial in the more general setting as it solves leakage! Leakage is the fact that NPE algorithms can lead to a posterior surrogate that has density outside of the prior support. This is a problem in sequential settings where this surrogate is used as a proposal for simulating new data points. The transformation from a normal distribution to the prior is a transformation that maps any distribution with infinite support to a distribution that has the same support as the prior. Therefore, leakage cannot happen when such a transformation is applied.

6.5 EXPERIMENTS

We investigate whether our proposed algorithms BNPE and BNRE-C lead to more conservative surrogate models compared to their non-balanced counterparts. Our diagnostics test the empirical expected coverage probability (6.7) of considered algorithms trained with $(\lambda > 0)$ and without $(\lambda = 0)$ the balance criterion. We also benchmark BNRE (Delaunoy, Hermans, et al., 2022), an algorithm that trains a neural network to minimize (6.12) using (6.3) as $L(\boldsymbol{w})$. The empirical expected coverage tests are performed on the following set of benchmarks of varying difficulty.

The *SLCP* simulator models a fictive problem with 5 parameters. The observable $\boldsymbol{x}$ is composed of 8 scalars which represent the 2D-coordinates of 4 points. The coordinate of each point is sampled from the same multivariate Gaussian whose mean and covariance matrix are parametrized by $\boldsymbol{\theta}$. We consider an alternative version of the original task (Papamakarios, Sterratt, and Murray, 2019) by inferring the marginal posterior density of 2 of those parameters. In contrast to its original formulation, the likelihood is not tractable due to the marginalization.

The *Weinberg* problem (K. Cranmer, Heinrich, et al., 2017) concerns a simulation of high energy particle collisions $e^+e^- \to \mu^+\mu^-$. The angular distributions of the particles can be used to measure the Weinberg angle $\boldsymbol{x}$ in the standard model of particle physics, leading to an observable composed of 20 scalars. From the scattering angle, we are interested in inferring Fermi's constant $\boldsymbol{\theta}$.

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

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 $\boldsymbol{\theta}$ which influence the reproduction and mortality rate of the predator and prey species. We infer the marginal posterior of the predator parameters from 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, Sterratt, and Murray (2019).

The *Two Moons* (Greenberg et al., 2019) simulator models a fictive problem with 2 parameters. The observable $\boldsymbol{x}$ 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.

Table 6.1 summarizes the architectures and hyper-parameters used for each benchmark. The architectures are separated into two parts: the embedding and the head networks. The embedding network compresses the observable into a set of features. The head network then uses those features concatenated with the parameters to produce the target quantity. The head can either be a classifier or a normalizing flow depending on the algorithm considered. The learning rate is scheduled during training. Table 6.1 provides the initial learning rates. Those are then divided by 10 each time no improvement was observed on the validation loss for 10 epochs.

	SLCP	Weinberg	Lotka-V.	Spatial SIR	Two Moons
Embedding network	None	None	CNN	CNN	None
Embedding layers	/	/	8	8	/
Embedding channels	/	/	8	16	/
Convolution type	/	/	Conv1D	Conv2D	/
Classifier Head	MLP	MLP	MLP	MLP	MLP
Classifier layers	6	6	6	6	6
Classifier hidden neurons	256	256	256	256	256
Flow Head	NSF	NSF	NSF	NSF	NSF
Flow layers	3	3	3	3	3
Flow hidden neurons	256	256	256	256	256
Learning rate	0.001	0.001	0.001	0.001	0.001
Epochs	500	500	500	500	500
Batch size	256	256	256	256	256
λ *(balanced algorithms)*	100	100	100	100	100
γ *(NRE-C)*	1	1	1	1	1
K *(NRE-C)*	5	5	5	5	5

Table 6.1: Architectures and training hyper-parameters

Expected coverage for these benchmarks are shown in 6.1. For both BNPE and BNRE-C, enforcing the balance criterion tends to produce conservative surrogates on most of the benchmarks, just as it did with BNRE. The only exception was Two Moons, which consistently produced overconfident surrogates for all algorithms. Taken as a whole, this provides some evidence that regularizing models allowing posterior density evaluation with the balance criterion makes them more conservative. Our results suggest that enforcing the balance criterion did not negatively impact the informativeness of the surrogate for higher simulation budgets. This is quantified in Figure 6.2 where we estimate the information contained in the surrogates through the nominal log posterior, defined as

$$\mathbb{E}_{p(\boldsymbol{\theta},\boldsymbol{x})}\left[\log \hat{p}(\boldsymbol{\theta} \mid \boldsymbol{x})\right].$$

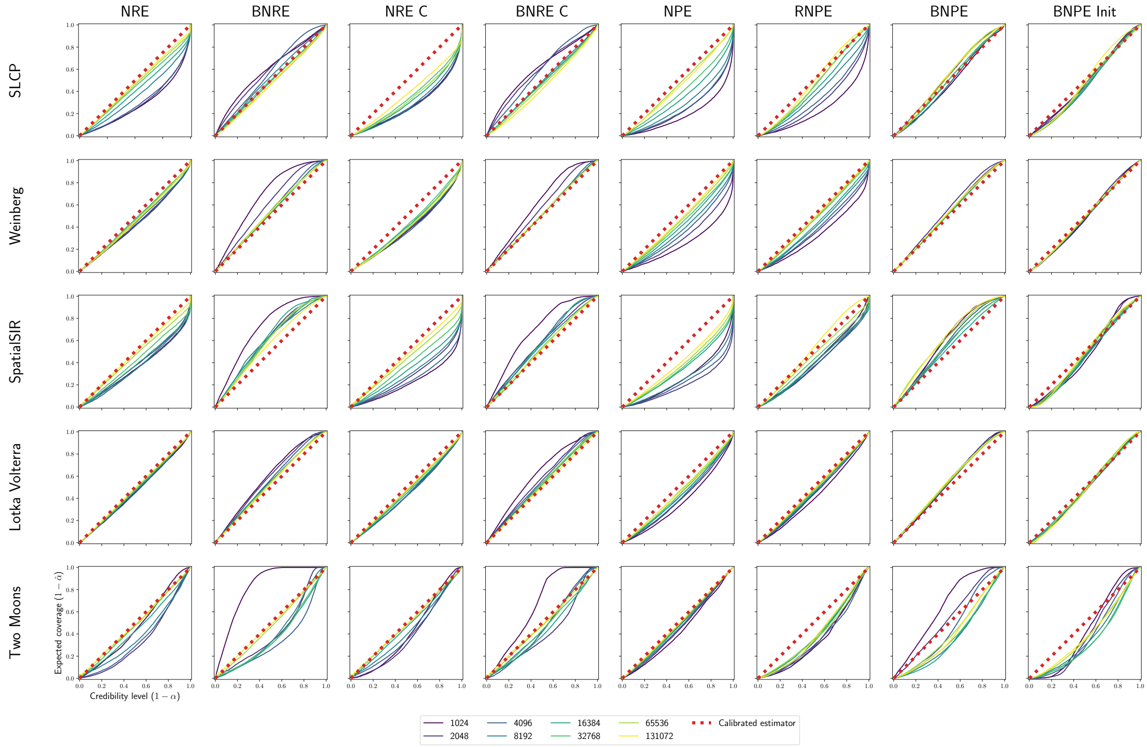

Figure 6.1: Empirical expected coverage for increasing simulation budgets. A perfectly calibrated surrogate has an expected coverage probability equal to the nominal coverage probability and produces a diagonal line. A conservative surrogate has an expected coverage curve at or above the diagonal line. An overconfident surrogate produces curves below the diagonal line. Balanced algorithms tend to produce conservative surrogates. 5 runs are performed for each simulation budget with the median $\hat{\alpha}$ at each nominal credibility reported.

We observe that balanced algorithms have a lower nominal log posterior than their corresponding non-balanced algorithms. However, they have similar nominal log posterior as the simulation budget increases. Figure 6.3 shows the balancing error, defined as

$$\begin{aligned}
&\left|\int (p(\boldsymbol{\theta})p(\boldsymbol{x}) + p(\boldsymbol{\theta},\boldsymbol{x}))\varpi(y=1\,|\,\boldsymbol{\theta},\boldsymbol{x})\,d\boldsymbol{\theta}\,d\boldsymbol{x} - 1\right| \\
=&\left|\int (p(\boldsymbol{\theta})p(\boldsymbol{x}) + p(\boldsymbol{\theta},\boldsymbol{x}))\frac{\hat{p}(\boldsymbol{\theta}|\boldsymbol{x})/p(\boldsymbol{\theta})}{1+\hat{p}(\boldsymbol{\theta}|\boldsymbol{x})/p(\boldsymbol{\theta})}\,d\boldsymbol{\theta}\,d\boldsymbol{x} - 1\right|.
\end{aligned}$$

Without surprise, we observe that enforcing the balancing condition leads to more balanced surrogates. Non-balanced algorithms show a high balancing error for low simulation budgets but this balancing error diminishes as the simulation budget gets higher. This is due to the fact that the posterior surrogate gets closer to the true posterior as the simulation increases and that the true posterior is always balanced. We also observe that normalizing flows struggle to minimize the balance criterion on the SLCP and Weinberg benchmarks for low simulation budgets. Normalizing flows must learn the prior as a multiplicative component in their density estimate, while likelihood-to-evidence ratio methods use the ground truth prior to construct the surrogate. It can be observed that initializing the posterior surrogate to the prior (BNPE init) indeed leads to a lower balancing error. Let us note that although the posterior surrogate is initialized to the prior, with a high enough

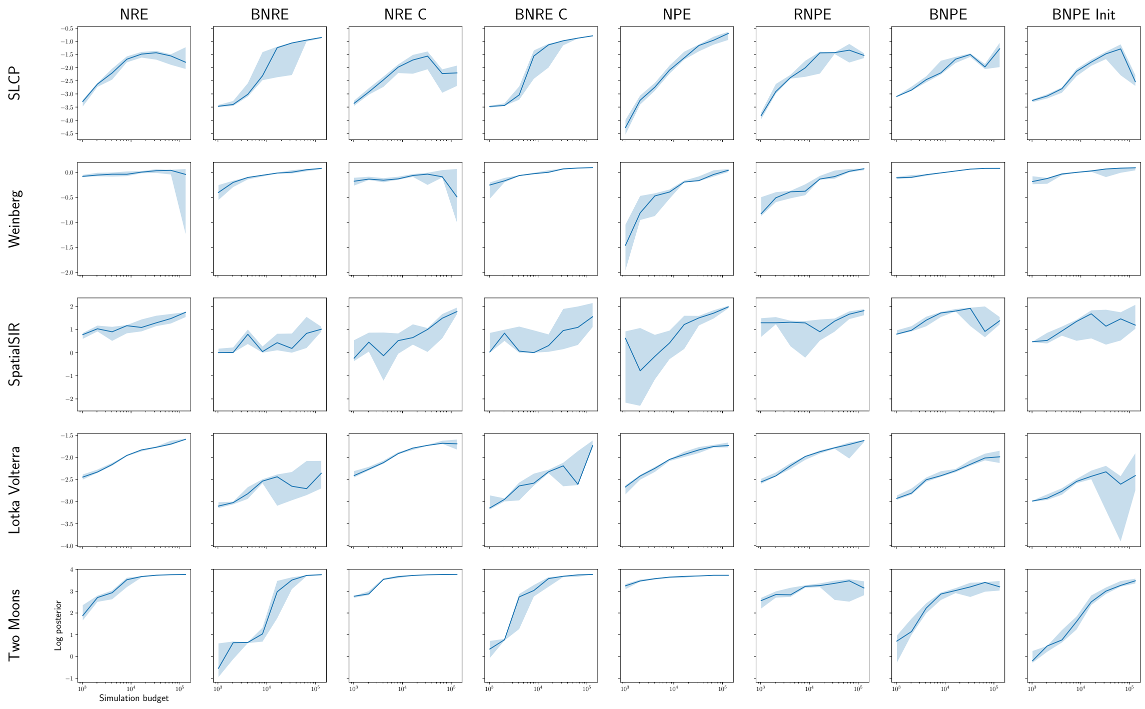

Figure 6.2: Nominal log posterior for increasing simulation budgets. 5 runs are performed for each simulation budget. Solid lines represent the median and the shaded areas represent the minimum and maximum. Larger values are desirable.

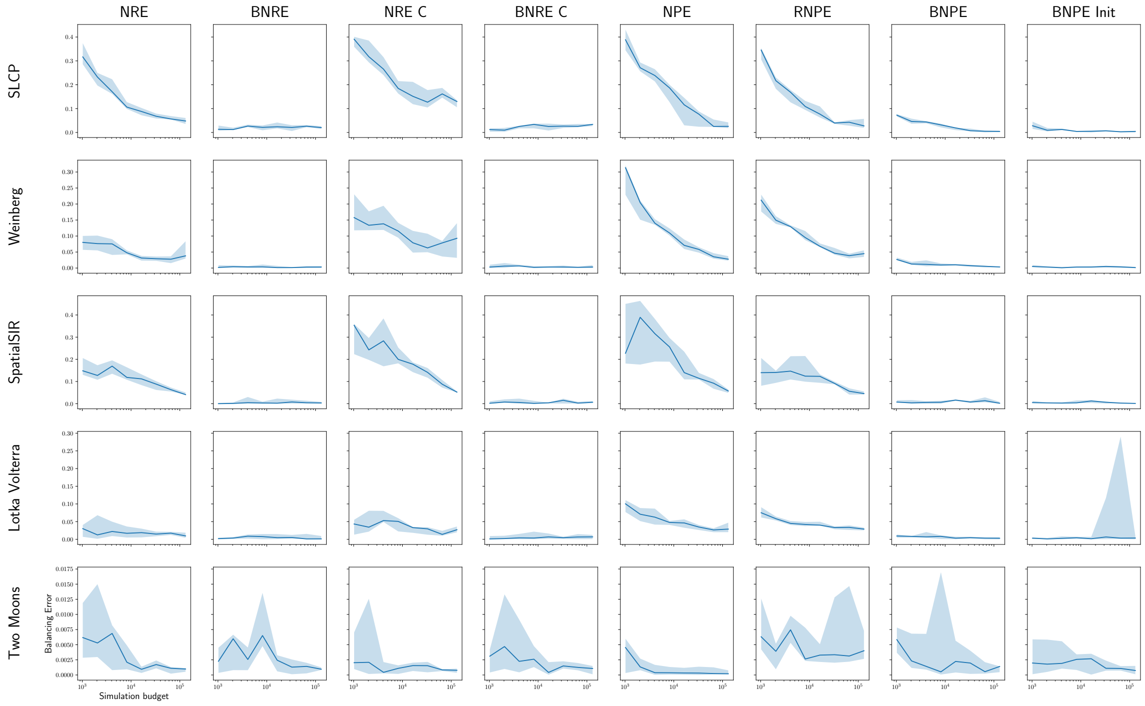

Figure 6.3: Balancing error for increasing simulation budgets. 5 runs are performed for each simulation budget. Solid lines represent the median and the shaded areas represent the minimum and maximum. Smaller values are desirable.

simulation budget, it is able to learn a balanced surrogate that carries information about the parameter as seen from the log posterior quantity in Figure 6.2.

6.6 CONCLUSIONS

In this work, we have shown that balancing can be applied to any simulation-based inference algorithm that yields a posterior density estimator, hence extending the applicability of balancing. We have also shown that the balancing condition can be expressed as a χ^2 divergence. We believe that this reformulation is a stepping stone towards a better understanding of the balancing condition and could inspire novel algorithms.

Let us note that there exist algorithms that do not directly provide the posterior density and hence do not fall into this framework. Some algorithms only aim to provide an unnormalized posterior approximation. This is the case of algorithms that model the likelihood (Papamakarios, Sterratt, and Murray, 2019) or an unnormalized version of the likelihood-to-evidence ratio (Durkan, Murray, and Papamakarios, 2020). A direct consequence is that those algorithms are not necessarily balanced at optimum. Enforcing the balancing condition may hence prevent reaching the optimal solution. Some algorithms also only allow sampling from the posterior surrogate. This is the case of score-based methods (Geffner et al., 2022; Song et al., 2020). In such a setting, the balancing condition cannot be computed. Future work would then include a reformulation of the balancing condition that uses only samples from the posterior surrogate.

Epilogue

We showed that the balancing condition can be applied to any algorithm providing explicit posterior density estimation. This includes NPE methods and NRE variants that produce normalized posterior densities, such as NRE-A and NRE-C. However, balancing is more difficult to enforce for NPE methods. To circumvent this issue, we proposed a new initialization procedure for normalizing flows, leading to balanced approximations at initialization. We observed empirically that using this initialization led to more balanced approximations. With the rise of diffusion models allowing the modeling of high-dimensional distributions, adapting balancing to methods that only provide posterior sampling could be beneficial for reliable simulation-based inference with high-dimensional parameter spaces.

While balancing has gained attention from the community and is cited in subsequent methodological papers, most applicative papers that report overconfidence do not modify the used algorithms to address the issue (Reza et al., 2024; Swierc et al., 2024; Morandini et al., 2024). This issue is not only about balancing, but all papers aiming at increasing conservativeness are at the time of writing, not used in practice. However, this line of work is still very recent. We hope that field scientists will become aware of it in the future and make use of those new tools to produce more reliable results.

7

SIMULATION-BASED INFERENCE WITH BAYESIAN NEURAL NETWORKS

Prologue

This chapter is based on the following publication: *Delaunoy, A.*, Bonardeaux, M. D. L. B.*, Mishra-Sharma, S., & Louppe, G. (2024). Low-Budget Simulation-Based Inference with Bayesian Neural Networks. arXiv preprint arXiv:2408.15136.*

While balancing provides a simple way to improve the conservativeness of many simulation-based inference algorithms, it needs to be enforced through regularization. We show that this can be problematic in settings where very few simulations are available for training. In addition, the demonstrated properties of balanced classifiers only provide intuition about why increased conservativeness is observed, but no direct link with coverage or posterior uncertainty has been established.

In this piece of work, we take a different approach and hypothesize that overconfidence is observed in classical methods because the uncertainty linked to approximation errors is not taken into account. We propose to use Bayesian neural networks to explicitly take this uncertainty into account. As opposed to the balancing line of work, this works with very low simulation budgets and, hence, has an increased range of applicability.

7.1 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, Delaunoy, 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; Y. 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 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, Delaunoy, 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, M. Cranmer, 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/ADelau/low_budget_sbi_with_bnn.

7.2 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 (K. Cranmer, Brehmer, and Louppe, 2020).

Recently, concerns have been raised regarding the calibration of the approximate posteriors obtained with neural simulation-based inference. Hermans, Delaunoy, 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

$$\mathrm{EC}(\hat{p}, \alpha) = \mathbb{E}_{p(\boldsymbol{\theta}, \boldsymbol{x})}[\mathbb{1}(\boldsymbol{\theta} \in \boldsymbol{\Theta}_{\hat{p}}(\alpha))] \tag{7.1}$$

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, Hermans, et al. (2022) and Delaunoy, Benjamin Kurt Miller, et al. (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, Ivanova, et al. (2024) 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 in all generality, where $\boldsymbol{x}$ denotes inputs, $\boldsymbol{y}$ outputs, $\boldsymbol{D}$ a dataset of N pairs $(\boldsymbol{x}, \boldsymbol{y})$, and $\boldsymbol{w}$ the weights of the neural network. The likelihood of a given set of weights is

$$p(\boldsymbol{D}|\boldsymbol{w}) \propto \prod_{i=1}^{N} p(\boldsymbol{y}_i|\boldsymbol{x}_i, \boldsymbol{w}), \tag{7.2}$$

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

$$p(\boldsymbol{w}|\boldsymbol{D}) = \frac{p(\boldsymbol{D}|\boldsymbol{w})p(\boldsymbol{w})}{p(\boldsymbol{D})}, \tag{7.3}$$

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

$$p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{D}) = \int p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{w})p(\boldsymbol{w}|\boldsymbol{D})d\boldsymbol{w} \simeq \frac{1}{M}\sum_{i=1}^{M} p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{w}_i), \boldsymbol{w}_i \sim p(\boldsymbol{w}|\boldsymbol{D}). \tag{7.4}$$

In practice, the Bayesian model average can be approximated by Monte Carlo sampling, with M samples from the posterior over the weights. The quality of the approximation depends on the number of samples M, which should be chosen high enough to obtain a good enough approximation but small enough to keep reasonable the computational costs of predictions.

Estimating the posterior over the neural network weights is 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.

7.3 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. Considering neural networks taking observations $\boldsymbol{x}$ as input and producing parameters θ as output, the posterior approximation $\hat{p}(\boldsymbol{\theta}|\boldsymbol{x})$ can be modeled as the Bayesian model average

$$\hat{p}(\boldsymbol{\theta}|\boldsymbol{x}) = \int p(\boldsymbol{\theta}|\boldsymbol{x}, \boldsymbol{w})p(\boldsymbol{w}|\boldsymbol{D})d\boldsymbol{w}, \tag{7.5}$$

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

Remaining is the choice of prior $p(\boldsymbol{w})$. While progress has been made in designing better priors in Bayesian deep learning (Fortuin, 2022), 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 $p(\boldsymbol{w}) = \mathcal{N}(\mathbf{0}, \sigma^2 \boldsymbol{I})$ on the weights, in which case

$$\hat{p}_{\text{normal prior}}(\boldsymbol{\theta}|\boldsymbol{x}) = \int p(\boldsymbol{\theta}|\boldsymbol{x}, \boldsymbol{w})\ \mathcal{N}(\boldsymbol{w}|\boldsymbol{\mu} = \mathbf{0}, \boldsymbol{\Sigma} = \sigma^2 \boldsymbol{I}) d\boldsymbol{w}. \tag{7.6}$$

As mentioned in Section 7.2, 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 7.1, 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 $p(\boldsymbol{w}|\boldsymbol{D})$ with a small amount of data would lead to a calibrated a posteriori Bayesian model average.

7.3.1 *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 over posteriors for parameters $\boldsymbol{\Theta}$ and observations $\boldsymbol{X}$ given a training 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

$$p(\boldsymbol{\theta}_i|\boldsymbol{x}_i, \boldsymbol{D}) = \int f_i\ p(\boldsymbol{f}|\boldsymbol{\Theta}, \boldsymbol{X}, \boldsymbol{D}) d\boldsymbol{f}, \quad \forall i. \tag{7.7}$$

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 $p_{\text{prior}} : (\boldsymbol{\theta}, \boldsymbol{x}) \rightarrow p(\boldsymbol{\theta})$, we have that $\text{EC}(p_{\text{prior}}, \alpha) = \alpha, \forall \alpha$ (Delaunoy, Benjamin Kurt Miller, 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

$$\begin{aligned} \hat{p}(\boldsymbol{\theta}_i|\boldsymbol{x}_i) &= \int f_i\ \delta([f_j = p_{\text{prior}}(\boldsymbol{\theta}_j, \boldsymbol{x}_j)])\ d\boldsymbol{f}, \forall i \\ &= \int f_i\ \delta(f_i = p(\boldsymbol{\theta}_i))\ df_i, \forall i \Rightarrow \text{EC}(\hat{p}, \alpha) = \alpha, \forall \alpha. \end{aligned} \tag{7.8}$$

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. For example, periodic kernels could be used if periodicity is observed. Kernel's hyperparameters can also be chosen such as to incorporate what would be a reasonable deviation of the approximated posterior from the prior. 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 7.3.1 shows that a functional prior defined in this way leads to a calibrated Bayesian model average.

Proposition 7.3.1. *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, Benjamin Kurt Miller, et al., 2023), then, $\text{EC}(\hat{p}, \alpha) = \alpha, \forall \alpha$. □

7.3.2 *From functional to parametric priors*

In this section, we now discuss how the functional GP prior over posterior density functions proposed in Section 3.1 can be used in the simulation-based inference setting. We follow Flam-Shepherd et al. (2017) and Sun et al. (2018) for mapping the functional prior to a distribution over neural network weights. We note that other methods for functional Bayesian deep learning have been proposed in the literature. These methods could also be used in our setting and are discussed at the end of the section.

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

$$\begin{aligned} p_{\text{BNN}}(\boldsymbol{f} \mid \boldsymbol{\phi}, \boldsymbol{\Theta}, \boldsymbol{X}) &= \int p(\boldsymbol{f} \mid \boldsymbol{w}, \boldsymbol{\Theta}, \boldsymbol{X}) \; p(\boldsymbol{w}|\boldsymbol{\phi}) \; d\boldsymbol{w} \\ &= \int \delta([f_i = p(\boldsymbol{\theta}_i|\boldsymbol{x}_i, \boldsymbol{w})]) \; p(\boldsymbol{w}|\boldsymbol{\phi}) \; d\boldsymbol{w}. \end{aligned} \tag{7.9}$$

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

$$\begin{aligned} &\text{KL}\left[p_{\text{BNN}}(\boldsymbol{f} \mid \boldsymbol{\phi}, \mathcal{M}) \,||\, p_{\text{GP}}(\boldsymbol{f} \mid \mu(\mathcal{M}), K(\mathcal{M}))\right] \\ &= \int p_{\text{BNN}}(\boldsymbol{f} \mid \boldsymbol{\phi}, \mathcal{M}) \log \frac{p_{\text{BNN}}(\boldsymbol{f} \mid \boldsymbol{\phi}, \mathcal{M})}{p_{\text{GP}}(\boldsymbol{f} \mid \mu(\mathcal{M}), K(\mathcal{M}))} d\boldsymbol{y} \\ &= -\mathbb{H}\left[p_{\text{BNN}}(\boldsymbol{f} \mid \boldsymbol{\phi}, \mathcal{M})\right] - \mathbb{E}_{p_{\text{BNN}}(\boldsymbol{f} \mid \boldsymbol{\phi}, \mathcal{M})}\left[\log p_{\text{GP}}(\boldsymbol{f} \mid \mu(\mathcal{M}), K(\mathcal{M}))\right], \end{aligned} \tag{7.10}$$

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

$$\nabla \mathbb{H}\left[p_{\text{BNN}}(\boldsymbol{f} \mid \boldsymbol{\phi}, \mathcal{M})\right] \simeq \text{SSGE}\left(\boldsymbol{f}_1, ..., \boldsymbol{f}_N \sim p_{\text{BNN}}(\boldsymbol{f} \mid \boldsymbol{\phi}, \mathcal{M})\right). \tag{7.11}$$

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.

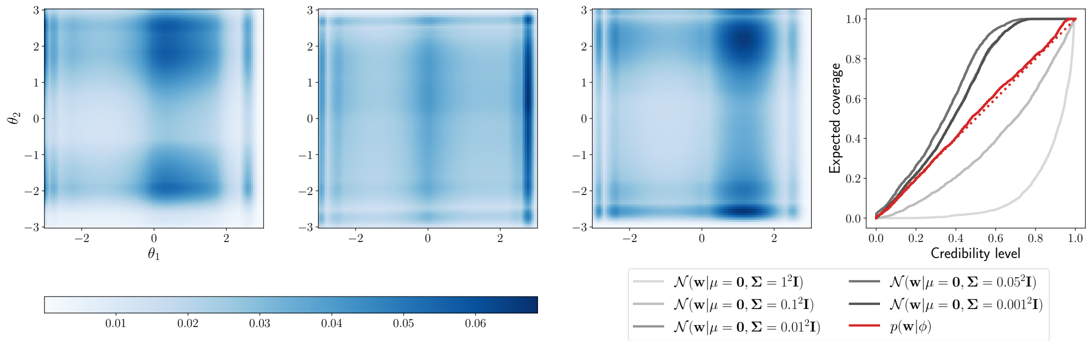

Figure 7.1: Visualization of the prior tuned to match the GP prior on the SLCP benchmark. Left: examples of posterior functions sampled from the tuned prior over neural network's weights. Right: expected coverage of the prior Bayesian model average with the tuned prior and normal priors for varying standard deviations.

As an illustrative example, we chose independent normal distributions as a variational family $p(\boldsymbol{w}|\boldsymbol{\phi})$ over the weights and minimize (7.10) w.r.t. $\boldsymbol{w}$. In Figure 7.1, we show the coverage of the resulting a priori Bayesian model average using the tuned prior, $p(\boldsymbol{w} \mid \boldsymbol{\phi})$, 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.

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 (B.-H. 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).

7.4 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, Bekasov, et al., 2019) and Neural Ratio Estimation (NRE) (Hermans, Begy, and Louppe, 2020), along with their balanced versions (BNRE and BNPE) (Delaunoy, Hermans, et al., 2022; Delaunoy, Benjamin Kurt Miller, et al., 2023) and ensembles (Lakshminarayanan et al., 2017; Hermans, Delaunoy, 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,

$$\mathbb{E}_{\boldsymbol{w} \sim p(\boldsymbol{w}|\boldsymbol{\tau})} \left[\sum_i \log p(\boldsymbol{\theta}_i | \boldsymbol{x}_i, \boldsymbol{w}) \right] - T \, \mathrm{KL}[p(\boldsymbol{w}|\boldsymbol{\tau}) || p(\boldsymbol{w}|\boldsymbol{\phi})], \tag{7.12}$$

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.

SETUP All the NPE-based methods use a Neural Spline Flow (NSF) (Durkan, Bekasov, 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. All the models are trained for 500 epochs which we observed to be enough to reach convergence.

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.

We evaluate the different methods of the SLCP, Two Moons, Lotka-Volterra and Spatial SIR benchmarks.

The *SLCP* (Simple Likelihood Complex Posterior) benchmark (Papamakarios, Sterratt, and Murray, 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.

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.

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, Sterratt, and Murray (2019).

The *Spatial SIR* model (Hermans, Delaunoy, 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.

In the experiment, The kernel K used in the GP prior is a combination of two Radial Basis Function (RBF) kernels. If more information on the structure of the target posterior is available, more informed kernels may be used to leverage this prior knowledge. The kernel is defined by

$$K(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \boldsymbol{x}_1, \boldsymbol{x}_2) = \sqrt{\text{RBF}(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2)} * \sqrt{\text{RBF}(\boldsymbol{x}_1, \boldsymbol{x}_2)}. \tag{7.13}$$

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

$$\text{RBF}(\boldsymbol{x}_1, \boldsymbol{x}_2) = \sigma^2 \exp\left(-\frac{1}{N}\sum_i^N \frac{(x_{1,i} - x_{2,i})^2}{2l_i^2}\right), \tag{7.14}$$

where σ is the standard deviation and l_i is the lengthscale associated to the i^{th} feature. The lengthscale is derived from the measurement set. To determine l_i, we query observations $\boldsymbol{x}$ from the measurement set and compute the 0.1 quantile of the squared distance between different observations for each feature. We then set l_i such that $2l_i^2$ equals this quantile. All the benchmarks have a uniform prior over the simulator's parameters. The mean function is then equal to a constant C for all input values. The standard deviation is chosen to be $C/2$.

Following Delaunoy, Hermans, 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 $\int_0^1 (\text{EC}(\hat{p},\alpha)-\alpha)\, d\alpha$ 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.

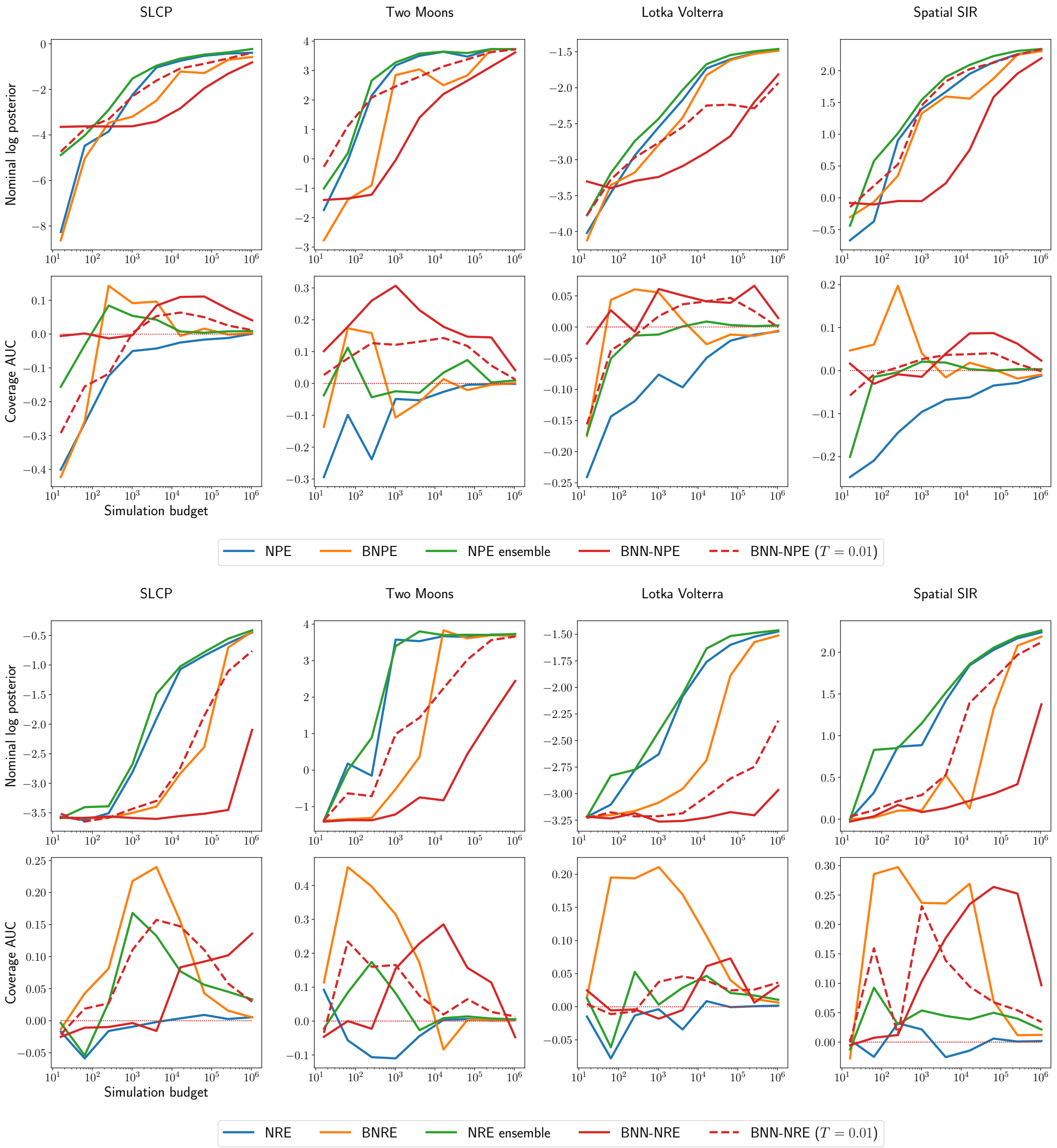

Figure 7.2: Comparison of simulation-based inference methods through the nominal log probability and coverage area under the curve. The higher the nominal log probability, the more performant the method is. A calibrated posterior approximation exhibits a coverage AUC of 0. A positive coverage AUC indicates conservativeness, and a negative coverage AUC indicates overconfidence. 3 runs are performed, and the median is reported. The plot at the top shows the results for NPE simulation-based inference methods, and the one at the bottom shows NRE methods.

BNN-based simulation-based inference Figure 7.2 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, or only slightly negative, coverage AUC even for simulation budgets as low as $O(10)$. Negative coverage AUC is still observed, and hence conservativeness is not strictly guaranteed. However, this constitutes a significant improvement over the other methods in that regard. The coverage curves are reported in Section 7.6. We conclude that BNNs can hence be more reliably used than the other benchmarked methods when the simulator is expensive and few simulations are available. We observe that increasing the reliability comes with the drawback of requiring more simulations than the other methods to reach similar nominal log posterior density values. Without temperature, a few orders of magnitude more samples might be needed. However, in theory, as the amount of sample increases, the effect of the prior diminishes, and BNNs should reach the same nominal log posterior density as standard methods. By adding a temperature to the prior, its effect is diminished and better nominal log posterior density values are observed From these observations, general guidelines to set the temperature include increasing T if overconfidence is observed and decreasing it if low predictive performance is observed.

Examples of posterior approximations obtained with and without using a Bayesian neural network are shown in Figure 7.3. 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. Finally, Figure 7.2 shows that BNN-NRE is more conservative than BNN-NPE. This comes at the cost of lower nominal log posterior density for a given simulation budget while still reaching comparable values as the simulation budgets grow.

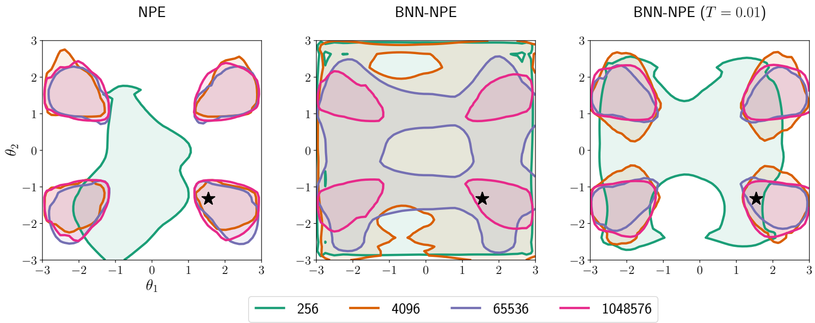

Figure 7.3: Examples of 95% highest posterior density regions obtained with various algorithms and simulation budgets on the SLCP benchmark for a single observation. The black star represents the ground truth used to generate the observation.

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 7.4, 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.

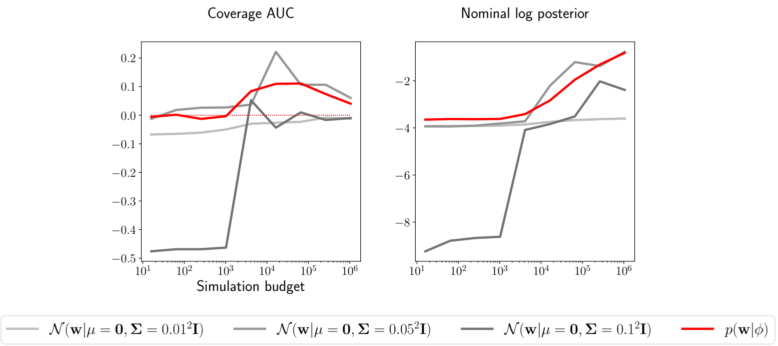

Figure 7.4: Comparison of posterior approximations obtained using a prior tuned to match the Gaussian process-based prior and using independent normal priors on weights with zero means and various standard deviations on the SLCP benchmark. 3 runs are performed, and the median is reported.

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

$$\mathbb{H}\left[\hat{p}(\boldsymbol{\theta}|\boldsymbol{x})\right] = \mathbb{E}_{q(\boldsymbol{w})}\left[\mathbb{H}\left[\hat{p}(\boldsymbol{\theta}|\boldsymbol{x},\boldsymbol{w})\right]\right] + \mathbb{I}(\boldsymbol{\theta},\boldsymbol{w}), \tag{7.15}$$

where $\mathbb{E}_{q(\boldsymbol{w})}\left[\mathbb{H}\left[\hat{p}(\boldsymbol{\theta}|\boldsymbol{x},\boldsymbol{w})\right]\right]$ quantifies the aleatoric uncertainty, $\mathbb{I}(\boldsymbol{\theta},\boldsymbol{w})$ quantifies the epistemic uncertainty, and the sum of those terms is the predictive uncertainty. Figure 7.5 shows the decomposition of the two sources of uncertainty, in expectation, on the SLCP benchmark. Other benchmarks can be found in Section 7.6. 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

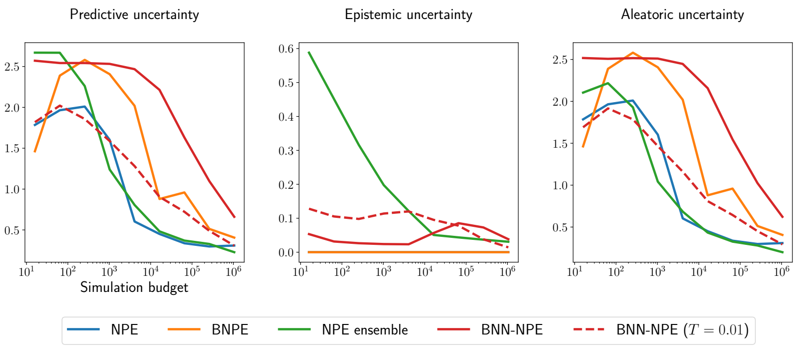

Figure 7.5: Quantification of the different forms of uncertainties captured by the different NPE-based methods on the SLCP benchmark. 3 runs are performed, and the median is reported.

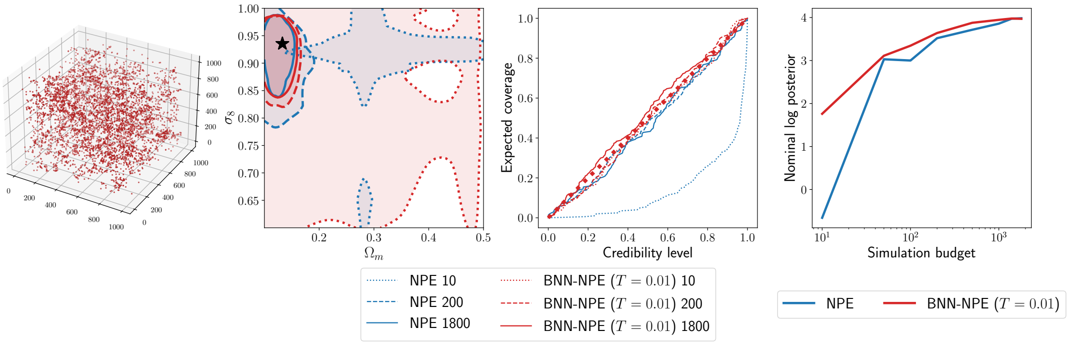

Figure 7.6: Comparison of the posterior approximations obtained with and without a Bayesian neural network on the cosmological application. First plot: An example observation: particles representing galaxies in a synthetic universe. Second plot: example of 95% highest posterior density regions for increasing 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.

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. We observed that setting a temperature lower than 1 was needed to reach reasonable predictive performance with Bayesian neural networks in this setting. Figure 7.6 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.

7.5 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.

Using BNNs for simulation-based inference also comes with limitations. The first observed limitation is that the Bayesian neural network based methods might need orders of magnitude more simulated data in order to reach a predictive power similar to methods that do not use BNNs, such as NPE. While we showed that this limitation can be mitigated by tuning the temperature appropriately, this is something that might require trials and errors. A second limitation is the computational cost of predictions. When training a BNN using variational inference, the training cost remains on a similar scale as standard neural networks. At prediction time, however, the Bayesian model average described in Equation 7.4 must be approximated, and this requires a neural network evaluation for each Monte Carlo sample in the approximation. The computational cost of predictions then scales linearly with the number M of Monte Carlo samples. Finally, although our method significantly improves the reliability over standard methods for low simulation budgets, conservativeness is not strictly guaranteed. There are no theoretical guarantees and negative coverage AUC may still be observed.

7.6 additional experiments

In this section, we provide complementary results. Figures 7.7 and 7.8 display the coverage curves, demonstrating that a higher positive coverage AUC corresponds to coverage curves above the diagonal line. Figures 7.9 and 7.10 present the uncertainty decomposition of all methods on all the benchmarks. Figures 7.11 and 7.12 illustrate how the performance of the different algorithms varies across different runs.

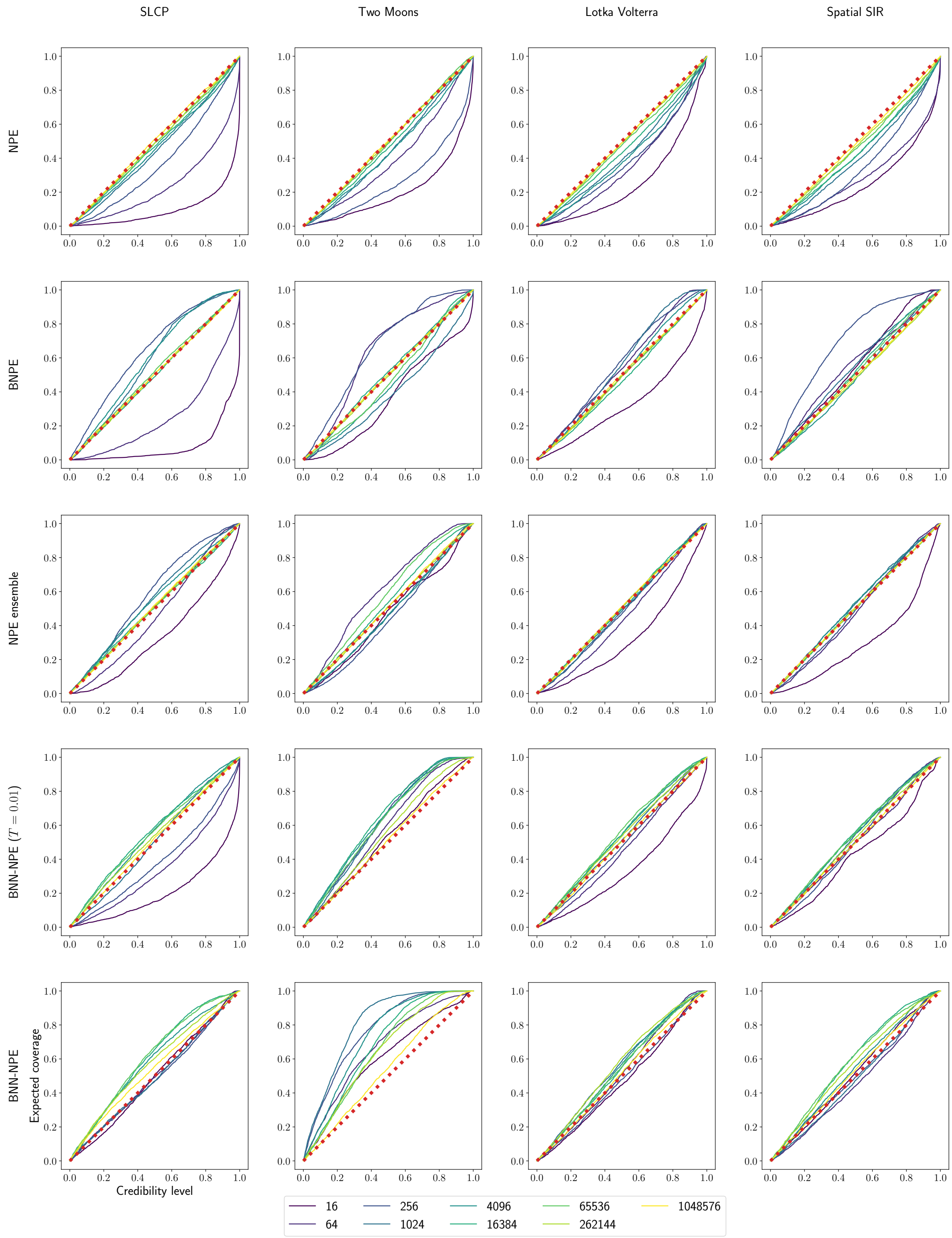

Figure 7.7: Coverage of different NPE simulation-based inference methods. A calibrated posterior approximation exhibits a coverage AUC of 0. A coverage curve above the diagonal indicates conservativeness and a curve below the diagonal indicates overconfidence. 3 runs are performed, and the median is reported.

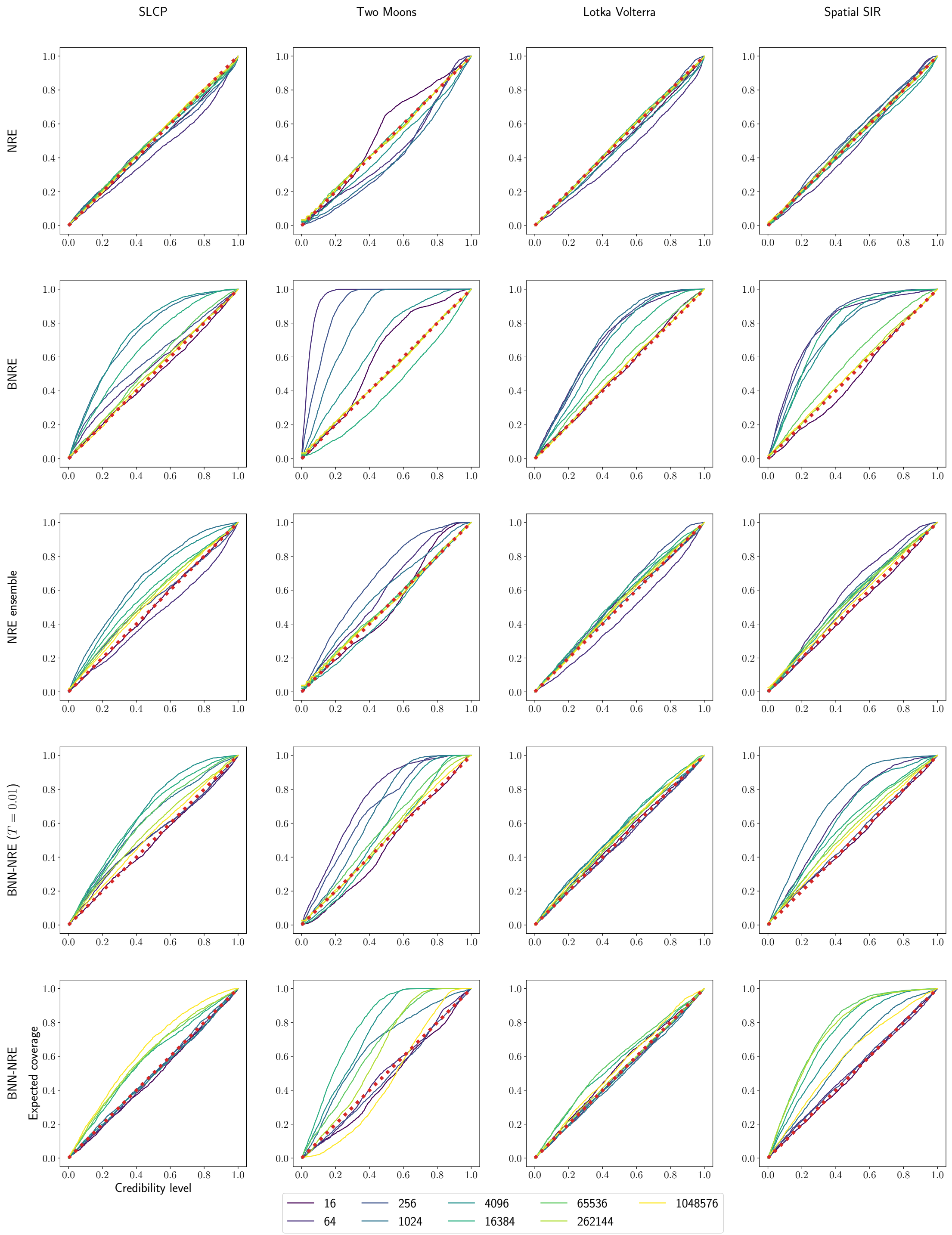

Figure 7.8: Coverage of different NRE simulation-based inference methods. A calibrated posterior approximation exhibits a coverage AUC of 0. A coverage curve above the diagonal indicates conservativeness and a curve below the diagonal indicates overconfidence. 3 runs are performed, and the median is reported.

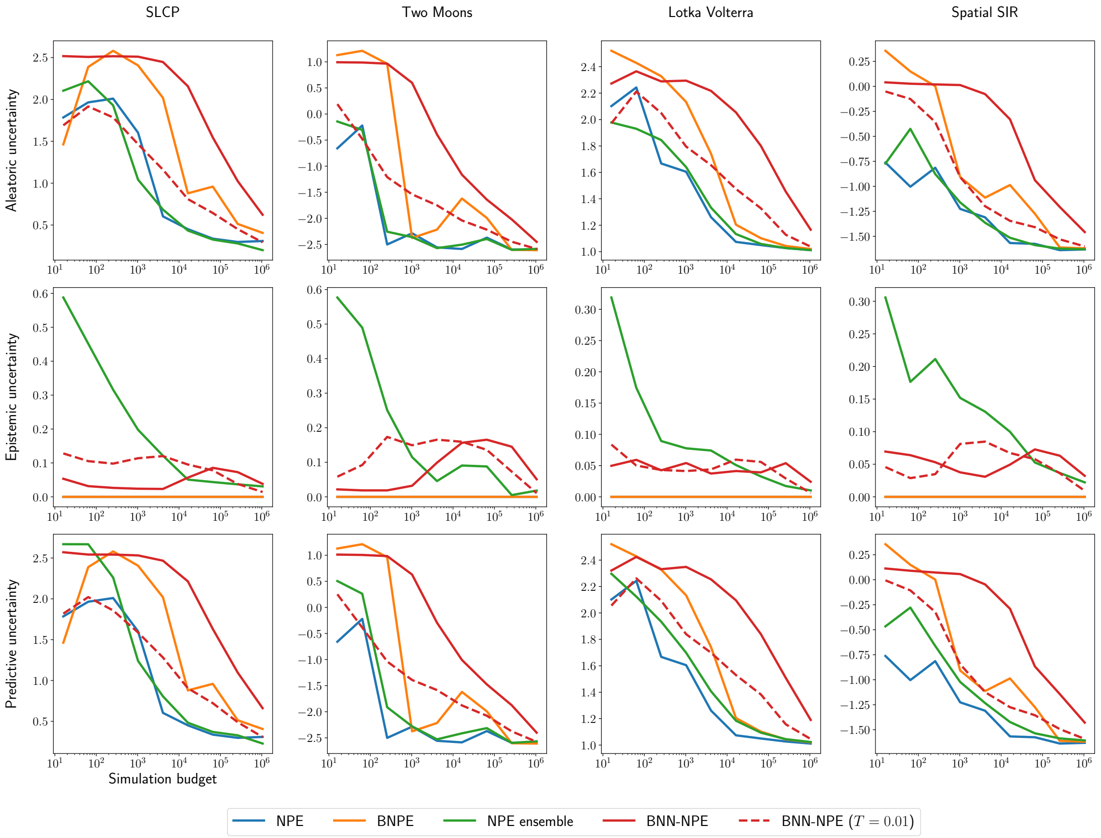

Figure 7.9: Quantification of the different forms of uncertainties captured by the different NPE-based methods. 3 runs are performed, and the median is reported.

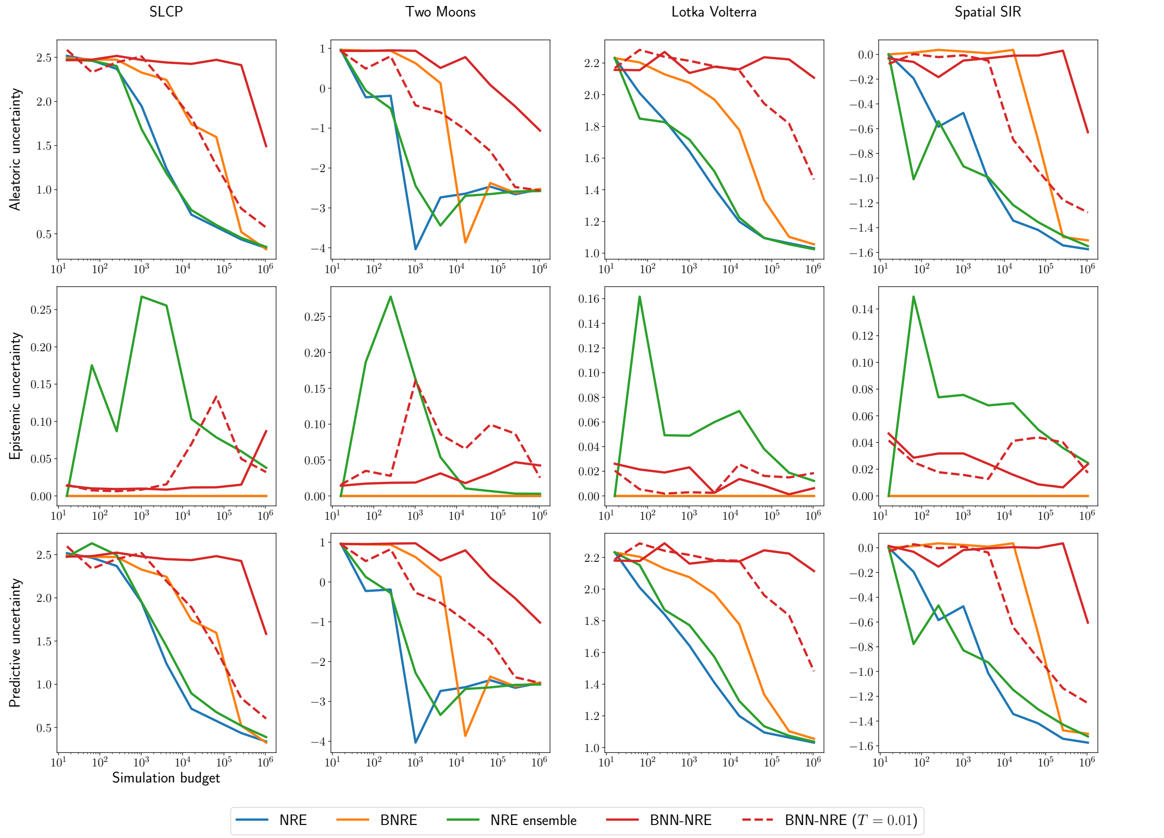

Figure 7.10: Quantification of the different forms of uncertainties captured by the different NRE-based methods. 3 runs are performed, and the median is reported.

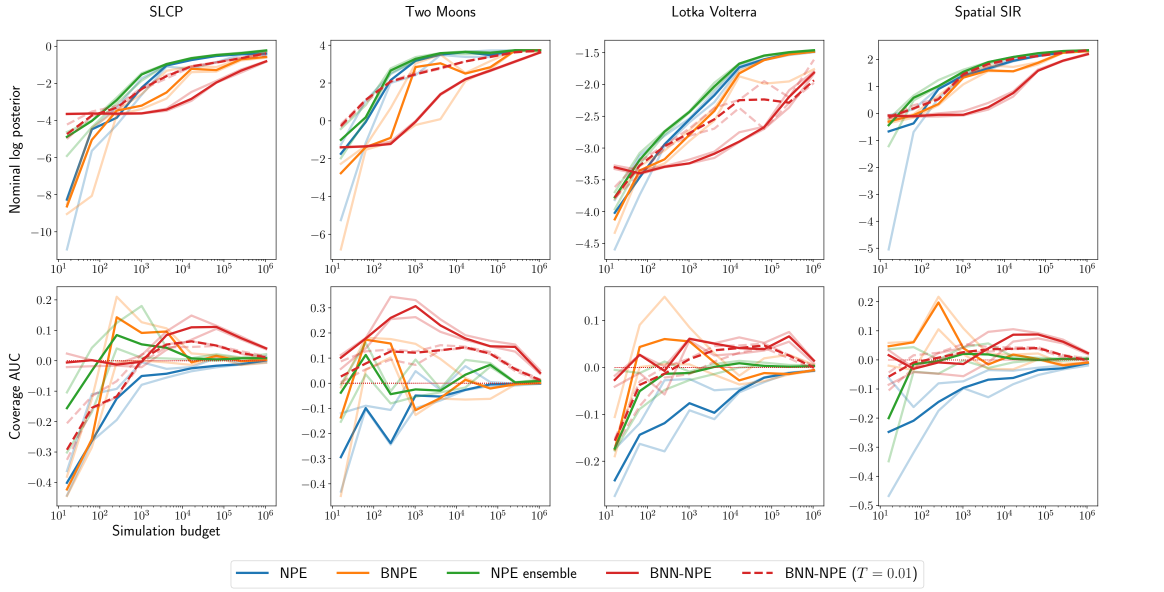

Figure 7.11: Comparison of different NPE simulation-based inference methods through the nominal log probability and coverage area under the curve. The higher the nominal log probability, the more performant the method is. A calibrated posterior approximation exhibits a coverage AUC of 0. A positive coverage AUC indicates conservativeness, and a negative coverage AUC indicates overconfidence. 3 runs are performed. The median run is reported in plain, and the shaded lines represent the other two runs.

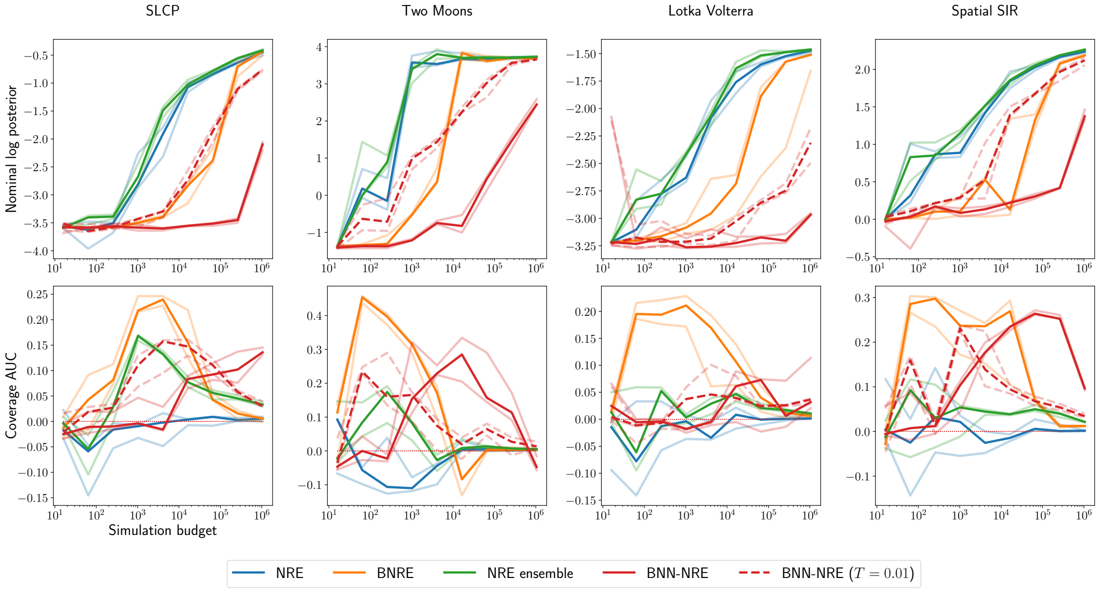

Figure 7.12: Comparison of different NRE simulation-based inference methods through the nominal log probability and coverage area under the curve. The higher the nominal log probability, the more performant the method is. A calibrated posterior approximation exhibits a coverage AUC of 0. A positive coverage AUC indicates conservativeness, and a negative coverage AUC indicates overconfidence. 3 runs are performed. The median run is reported in plain, and the shaded lines represent the other two runs.

Epilogue

We showed that using Bayesian neural networks allows to take computational uncertainty into account and leads to more conservative posterior approximations. We illustrated the importance of the prior choice for Bayesian neural networks in the context of simulation-based inference and defined a family of priors tailored for this field. Limitations of this work include the increased computational cost as a prior must first be optimized, a posterior over neural networks must be approximated, and several neural network evaluations are required to make predictions. However, in settings where the simulator is very expensive to run, this cost will be negligible compared to the cost of making many more simulations. Another limitation is the loss of predictive power, requiring much more data to reach a predictive power similar to NPE. We showed that this can be improved by tuning the prior temperature. In addition, as the focus was on discussing the use of Bayesian deep learning for simulation-based inference and the design of an appropriate prior, we used mean field variational inference, a rather simple Bayesian deep learning algorithm that requires little tuning. Using more advanced inference algorithms and careful tuning could improve results in that regard. We also hope that BNNs' predictive power improves with new advances in the field of Bayesian deep learning.

8 DISCUSSION

Throughout this thesis, we motivated the need for conservative posterior approximations, introduced the expected coverage to diagnose conservativeness, and introduced two ways to improve conservativeness: balancing and the use of Bayesian neural networks. In this chapter, we discuss other lines of work related to the conservativeness of posterior approximations and some limitations that remain with current simulation-based inference algorithms. We end the discussion with a general conclusion.

8.1 RELATED WORK

Another way of improving conservativeness in cases where the expected coverage can be approximated is to perform explicit calibration on the coverage. If we are solely interested in credible regions and not in the posterior itself, this can be done post-training. The expected coverage of a posterior approximation can be computed for a range of levels using a calibration set sampled from $p(\boldsymbol{\theta}, \boldsymbol{x})$, yielding a coverage plot. Given a posterior approximation $\hat{p}(\boldsymbol{\theta}|\boldsymbol{x})$, to obtain credible intervals with expected coverage approximately of α, credible intervals at level $\tilde{\alpha}$ can be used, where $\tilde{\alpha}$ is such that

$$\mathrm{EC}(\hat{p}, \tilde{\alpha}) = \alpha, \tag{8.1}$$

where EC denotes the expected coverage. If an infinite amount of calibration samples are used, then the produced credible regions at level $\tilde{\alpha}$ have an expected coverage of exactly α. For a finite amount of calibration samples, correct coverage is not guaranteed.

Patel et al. (2023) proposed to use conformal prediction as a way to produce regions with some guarantees about the expected coverage for a finite calibration set. Specifically, they use inductive conformal prediction that selects points to include in a region by comparing their non-conformity score to the ones of others in the calibration set. In this case, the authors propose to use

$$s(\boldsymbol{x}, \boldsymbol{\theta}) = \frac{1}{\hat{p}(\boldsymbol{\theta}|\boldsymbol{x})} \tag{8.2}$$

as non-conformity score. This score is computed for all samples from the calibration set $(\boldsymbol{\theta}, \boldsymbol{x}) \sim p(\boldsymbol{\theta}, \boldsymbol{x})$. If we aim for a region generator that leads to an expected coverage of α, the threshold $s_C(\alpha)$ is the $\lceil (N_c + 1)/(1 - \alpha) \rceil / N_c$ score quantile on the calibration set, where N_c is the size of the calibration set. The prediction region for an observation $\boldsymbol{x}$ is then constructed as

$$C(\boldsymbol{x}) = \{\boldsymbol{\theta} : s(\boldsymbol{\theta}, \boldsymbol{x}) \leq s_C(\alpha)\}. \tag{8.3}$$

If the calibration samples and the observation used for prediction are all sampled independently from the same distribution (i.i.d.), the expected coverage of this prediction region generator is guaranteed to not be lower than ϵ with some probability above δ, with ϵ and δ being varying with the size of the calibration set N_c (Vovk, 2012). Inductive conformal prediction then bounds miscoverage with some probability. Note that this statement is about expected coverage and that those prediction regions might not be valid credible regions for all observations $\boldsymbol{x}$ or valid confidence region generators for all parameters $\boldsymbol{\theta}$.

In order to obtain not only calibrated credibility regions in expectation but posterior approximations leading to such regions, Falkiewicz et al. (2024) propose to add a regularization term based on the coverage that penalizes for either overconfidence or miscalibration. This is then similar to what is done for balanced algorithms, with the exception that regularization is directly performed on coverage instead of enforcing the balancing condition. While regularizing for coverage is more desirable than balancing, approximating the coverage at each batch introduces a computational overhead, while the balancing condition is cheap to approximate.

K. Cranmer, Pavez, and Louppe (2015) show that an approximate likelihood ratio can be calibrated to converge to the real likelihood ratio provided that the approximation is a monotonic function of the likelihood ratio. Under this condition, they show that

$$r(\boldsymbol{x}; \boldsymbol{\theta}_0, \boldsymbol{\theta}_1) = \frac{p(\boldsymbol{x}|\boldsymbol{\theta}_0)}{p(\boldsymbol{x}|\boldsymbol{\theta}_1)} = \frac{p(\hat{r}(\boldsymbol{x}; \boldsymbol{\theta}_0, \boldsymbol{\theta}_1)|\boldsymbol{\theta}_0)}{p(\hat{r}(\boldsymbol{x}; \boldsymbol{\theta}_0, \boldsymbol{\theta}_1)|\boldsymbol{\theta}_1)}, \tag{8.4}$$

where $p(\hat{r}(\boldsymbol{x}; \boldsymbol{\theta}_0, \boldsymbol{\theta}_1)|\boldsymbol{\theta}_0)$ and $p(\hat{r}(\boldsymbol{x}; \boldsymbol{\theta}_0, \boldsymbol{\theta}_1)|\boldsymbol{\theta}_1)$ can be approximated on a calibration set.

Dalmasso, Izbicki, and Lee (2020) and Masserano et al. (2023) aim for tractable valid frequentists coverage for all parameter values $\boldsymbol{\theta}$. They base their method on the Neyman construction that uses the relationship between confidence regions and hypothesis tests presented in Equation 2.34. A classical Neyman construction requires estimating critical values of a hypothesis test for each parameter value on a fine grid, which can be computationally intensive. Instead of estimating critical values for each parameter value in the grid independently, they propose to train a quantile regressor on test statistics computed on $(\boldsymbol{\theta}, \boldsymbol{x})$ pairs sampled from $p(\boldsymbol{\theta}, \boldsymbol{x})$. The regressor then takes parameters $\boldsymbol{\theta}$ as input and outputs approximate quantiles of the distribution of test statistics for various observations $\boldsymbol{x} \sim p(\boldsymbol{x}|\boldsymbol{\theta})$. If the regressor is assumed to be perfectly trained, those quantiles are valid critical values, and the Neyman construction using such critical values yields valid Confidence regions. Zhao et al. (2021) introduce a local coverage test. Similarly to expected coverage computation, test pairs are sampled $\boldsymbol{\theta}^*, \boldsymbol{x}^* \sim p(\boldsymbol{\theta}, \boldsymbol{x})$, and the rank, i.e., the minimal confidence level α required for the approximate credible region based on $\boldsymbol{x}^*$ to include the nominal parameters $\boldsymbol{\theta}^*$, is computed. Instead of reporting the marginal rank distributions, they train regressors for various confidence levels α, taking $\boldsymbol{x}^*$ as input and outputting 1 if the rank is lower than α, 0 otherwise. They use this regressor to construct a statistic for testing local coverage.

8.2 LIMITATIONS

In this thesis, we focused on the conservativeness aspect of approximate posteriors. While we provided ways to improve the conservativeness of current methods, limitations remain both on these aspects and on other aspects to obtain reliable simulation-based inference. One of those limitations is the lack of theoretical guarantees regarding conservativeness. Of all the discussed methods, conformal prediction (Patel et al., 2023) is the only one providing guarantees in the finite samples regime. These guarantees are, however, in expectation, and the produced regions are neither valid credible regions for all $\boldsymbol{x}$ nor valid confidence regions for all $\boldsymbol{\theta}$. Vovk (2012) showed that object conditional coverage cannot be guaranteed for rich observational spaces, such as $\mathbb{R}$, without any distributional assumptions. In the remainder of this section, we expand on two other limitations, namely the non-invariance to reparametrization of expected coverage based on high posterior density regions and possible misspecification of the simulator.

High posterior density regions are not invariant to reparametrizations. Let us assume a transformation f applied to parameters $\boldsymbol{\theta}$, then $p(f(\boldsymbol{\theta})|\boldsymbol{x})$ is not necessarily equivalent to $p(\boldsymbol{\theta}|\boldsymbol{x})$. For example, if we consider the transformation $f : \boldsymbol{\theta} \rightarrow 2\boldsymbol{\theta}$, then we have

$$p(f(\boldsymbol{\theta})|\boldsymbol{x}) = \frac{p(\boldsymbol{\theta}|\boldsymbol{x})}{2}, \quad \forall \boldsymbol{\theta}. \tag{8.5}$$

In this case, the density is divided by 2 but the resulting high posterior density region would remain the same. This is not the case if the transformation f is such that some parts of the space are more dilated or contracted than others. In those cases, we might have $p(f(\boldsymbol{\theta})|\boldsymbol{x}) > p(\boldsymbol{\theta}|\boldsymbol{x})$ for some parameters $\boldsymbol{\theta}$ and $p(f(\boldsymbol{\theta})|\boldsymbol{x}) < p(\boldsymbol{\theta}|\boldsymbol{x})$ for some other parameters $\boldsymbol{\theta}$. Consequently, high posterior density regions might be different after applying the reparametrization. Figure 8.1 shows an example of this for the transformation $f : \boldsymbol{\theta} \rightarrow \exp(\boldsymbol{\theta})$. The 95% high posterior density credible interval for $\boldsymbol{\theta}$ is $[-1.95, 1.95]$ and the one for $\exp(\boldsymbol{\theta})$ is $[0.03, 5.23]$. We have that $\exp(-1.95) = 0.14 \neq 0.03$ and $\exp(1.95) = 7.02 \neq 5.23$. Different parts of the parameter space are then included in the high posterior density credible interval depending on the used parametrization.

This raises some questions regarding the definition of a conservative posterior approximation. Indeed, any conservative posterior approximation can be turned into an overconfident approximation by reparametrizing $\boldsymbol{\theta}$, according to the definition based on high posterior density region expected coverage. To make sense of this definition of conservativeness, we argue that parametrization should be chosen by taking domain knowledge into account such that flat distributions correspond to distributions that domain experts would consider uncertain. To obtain quantities that are invariant to reparametrization, ideas can be

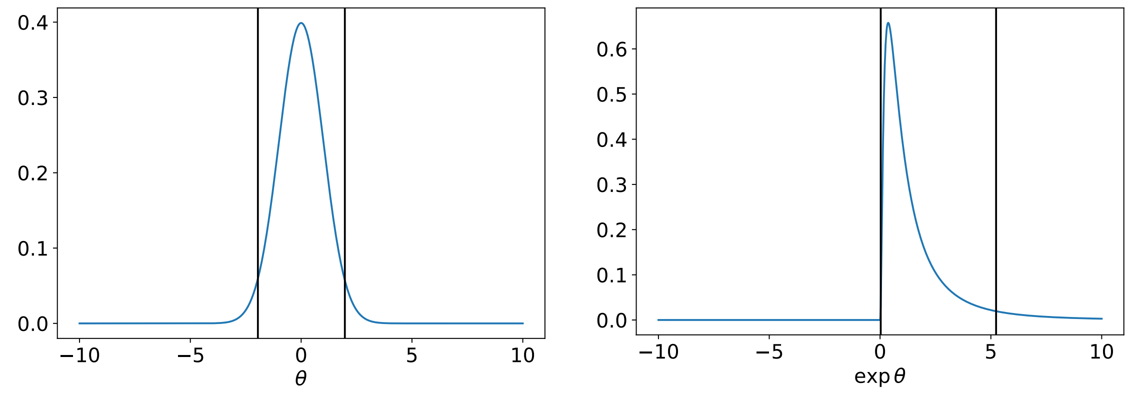

Figure 8.1: Comparison of high posterior density credible intervals obtained using various parameterizations. We consider that one parametrization is the exponential of the other. The blue lines represent the p.d.f. and the black lines represent the 95% high posterior density credible intervals.

taken from the objective Bayesian inference literature. In particular, high-posterior density regions can be replaced by low-loss credible regions that favor parts of the parameters space that have a low loss value according to some loss function (Bernardo, 2005). If that loss function is invariant to a reparametrization, then so are the resulting credible regions.

In addition to issues linked to posterior approximations, scientific models can be misspecified, in the sense that it does not perfectly model reality. Let us consider that real observations are drawn from the data-generating process $p^*(\boldsymbol{x})$. Following Cannon et al. (2022), a model is misspecified if there are no parameters $\boldsymbol{\theta}$ such that $p^*(\boldsymbol{x}) = p(\boldsymbol{x}|\boldsymbol{\theta})$. A consequence of misspecification when using machine learning models for approximate inference is that the machine learning model is used to make predictions on observations from the distribution $p^*(\boldsymbol{x})$ while being trained on samples from the distribution $\int p(\boldsymbol{x}|\boldsymbol{\theta})p(\boldsymbol{\theta})d\boldsymbol{\theta}$. The model might then perform poorly on observations from $p^*(\boldsymbol{x})$. This is illustrated by Gloeckler, Deistler, and Macke (2023), showing that machine learning models are vulnerable to adversarial attacks in the context of simulation-based inference. Adversarial attacks are small perturbations to the observation $\boldsymbol{x}$ such that the resulting posterior approximation changes drastically. Those can then be seen as a worst-case scenario of model misspecification. Model misspecification may be detected by making posterior predictive checks, i.e., by comparing samples $\boldsymbol{x}_o \sim p^*(\boldsymbol{x})$ to samples from $\int p(\boldsymbol{x}|\boldsymbol{\theta})p(\boldsymbol{\theta}|\boldsymbol{x}_o)$

Wehenkel et al. (2024) consider an alternative definition. They assume that this data generation process can be expressed as $p^*(\boldsymbol{x}) = \int p(\boldsymbol{\theta})p^*(\boldsymbol{x}|\boldsymbol{\theta})d\boldsymbol{\theta}$, where $p(\boldsymbol{\theta})$ is a known prior and $p^*(\boldsymbol{x}|\boldsymbol{\theta})$ is an unknown process. They then define a misspecified model as a model such that

$$p^*(\boldsymbol{\theta}|\boldsymbol{x}) = \frac{p^*(\boldsymbol{x}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p^*(\boldsymbol{x})} \neq \frac{p(\boldsymbol{x}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\boldsymbol{x})} = p(\boldsymbol{\theta}|\boldsymbol{x}). \tag{8.6}$$

They assume having some samples from the process $p^*(\boldsymbol{x}|\boldsymbol{\theta})$ through, for example, more costly procedures and use those samples to correct the posterior approximation.

8.3 CONCLUSION

The use of approximate methods for statistical inference raises questions about the reliability of the obtained results for scientific reasoning. In particular, it might lead to the incorrect refutation of scientific models and lead scientists in the wrong direction. In chapter 4, we shed light on the impact of approximate methods for scientific reasoning and provided the expected coverage diagnostic as a way to identify issues with approximations. We defined approximations passing the expected coverage diagnostic, i.e., having an expected coverage higher than the desired credibility level as conservative, and argued that conservative approximations are desired for more reliable scientific reasoning. An empirical evaluation showed that state-of-the-art methods, at that time, did not always lead to conservative approximations. We introduced in Chapters 5 and 6 balancing as a way to increase the conservativeness of the resulting approximations. Balancing is a technique that is simple to implement and does not highly increase computational requirements. It is a simple first thing to try to improve conservativeness if overconfident approximations are observed. Balancing can fall short in cases where very few samples from a simulator model are available to approximate the desired quantities. In Chapter 7, we used Bayesian neural networks to improve the conservativeness of simulation-based inference algorithms in such regimes.

While we proposed methods for improving conservativeness, the limitations shown in Section 8.2 highlight that the resulting methods still do not guarantee reliable Popperian falsification. The proposed methods aim for conservativeness as defined by the expected coverage but expected coverage only provides information about produced credible regions in expectation and is not invariant to reparametrization unless special care is taken. While we believe improving expected coverage conservativeness increases the reliability, those considerations suggest taking a step back and defining stronger objectives to aim for, with the objective of achieving reliable scientific reasoning with numerical approximations. For example, conservativeness should ideally not only be achieved in expectation. It should be ensured for the particular observation at hand.

It might, however, be unrealistic to aim for conservativeness guarantees for a given observation and, in general, to create algorithms having sufficiently strong guarantees to be used for Popperian falsification while dealing with numerical approximations. Most current methods aiming for conditional coverage guarantees involve another machine learning model for calibration that itself leads to another layer of approximations. A more realistic aim could be to make the scientific method evolve to deal with the presence of approximations while at the same time improving those approximations. A way would be to no longer consider that models can be definitively refuted, as the procedure leading to refutation involves approximations. Such models that would be considered as refuted under the hypothesis that approximations are perfect could, however, be considered as improbable. The quality of the approximations could then themselves be criticized, leading to considering those models again.

This thesis focused on the implications of numerical approximations on the process of scientific discovery and aimed to improve the reliability of approximate statistical inference methods in that regard. It should be noted that what constitutes an appropriate scientific methodology is a complex open philosophical debate even when no approximations are made in the statistical pipeline. Our work is only a stepping stone for improving an aspect of the scientific methodology. Drawing reliable scientific conclusions goes beyond having high-quality approximations, and the definition of reliable is up for debate.

BIBLIOGRAPHY

Box, G. E. (1976). Science and statistics. *Journal of the American Statistical Association*, *71*(356), 791–799.

Blei, D. M. (2014). Build, compute, critique, repeat: Data analysis with latent variable models. *Annual Review of Statistics and Its Application*, *1*(1), 203–232.

Popper, K. R. (1959). *The logic of scientific discovery*. Routledge.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., & Teller, E. (1953). Equation of state calculations by fast computing machines. *The journal of chemical physics*, *21*(6), 1087–1092.

Hastings, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, *57*(1), 97–109. https://doi.org/10.1093/biomet/57.1.97

Goodman, J., & Weare, J. (2010). Ensemble samplers with affine invariance. *Communications in Applied Mathematics and Computational Science*, *5*(1), 65–80. https://doi.org/10.2140/camcos.2010.5.65

Nelson, B., Ford, E. B., & Payne, M. J. (2013). Run dmc: An efficient, parallel code for analyzing radial velocity observations using n-body integrations and differential evolution markov chain monte carlo. *The Astrophysical Journal Supplement Series*, *210*(1), 11.

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

Neal, R. M. (2012). Mcmc using hamiltonian dynamics. *arXiv preprint arXiv:1206.1901*.

Neyman, J. (1937). Outline of a theory of statistical estimation based on the classical theory of probability. *Philosophical Transactions of the Royal Society A*, *236*, 333–380. https://api.semanticscholar.org/CorpusID:19584450

Jo, B., & Jason, S. (2017). Streamgap-pepper.

Hannam, M., Schmidt, P., Bohé, A., Haegel, L., Husa, S., Ohme, F., Pratten, G., & Pürrer, M. (2014). Simple model of complete precessing black-hole-binary gravitational waveforms. *Physical review letters*, *113*(15), 151101.

Wagner-Carena, S., Lee, J., Pennington, J., Aalbers, J., Birrer, S., & Wechsler, R. H. (2024). A strong gravitational lens is worth a thousand dark matter halos: Inference on small-scale structure using sequential methods. *arXiv preprint arXiv:2404.14487*.

Destexhe, A., & Huguenard, J. R. (2000). Nonlinear thermodynamic models of voltage-dependent currents. *Journal of Computational Neuroscience*, *9*, 259–270.

Weligampola, H., Ramanayake, L., Ranasinghe, Y., Ilangarathna, G., Senarath, N., Samarakoon, B., Godaliyadda, R., Herath, V., Ekanayake, P., Ekanayake, J., Maheswaran, M., Theminimulle, S., Rathnayake, A., Dharmaratne, S., Pinnawala, M., Yatigammana, S., & Tilakaratne, G. (2023). Pandemic simulator: An agent-based framework with human behavior modeling for pandemic-impact assessment to build sustainable communities. *Sustainability*, *15*(14). https://doi.org/10.3390/su151411120

ATLAS, C., et al. (2012). Observation of a new particle in the search for the standard model higgs boson with the atlas detector at the lhc.

Rubin, D. B. [Donald B.]. (1984). Bayesianly Justifiable and Relevant Frequency Calculations for the Applied Statistician. *The Annals of Statistics*, *12*(4), 1151–1172.

Pritchard, J. K., Seielstad, M. T., Perez-Lezaun, A., & Feldman, M. W. (1999). Population growth of human y chromosomes: A study of y chromosome microsatellites. *Molecular biology and evolution*, *16*(12), 1791–1798.

Beaumont, M. A. (2019). Approximate bayesian computation. *Annual review of statistics and its application*, *6*(1), 379–403.

Fearnhead, P., & Prangle, D. (2012). Constructing summary statistics for approximate bayesian computation: Semi-automatic approximate bayesian computation. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *74*(3), 419–474.

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

Marjoram, P., Molitor, J., Plagnol, V., & Tavaré, S. (2003). Markov chain monte carlo without likelihoods. *Proceedings of the National Academy of Sciences*, *100*(26), 15324–15328.

Toni, T., & Stumpf, M. P. H. (2009). Simulation-based model selection for dynamical systems in systems and population biology. *Bioinformatics*, *26*(1), 104–110. https://doi.org/10.1093/bioinformatics/btp619

Sisson, S. A., Fan, Y., & Tanaka, M. M. (2007). Sequential monte carlo without likelihoods. *Proceedings of the National Academy of Sciences*, *104*(6), 1760–1765.

Beaumont, M. A., Cornuet, J.-M., Marin, J.-M., & Robert, C. P. (2009). Adaptive approximate bayesian computation. *Biometrika*, *96*(4), 983–990.

Delaunoy, A., Wehenkel, A., Hinderer, T., Nissanke, S., Weniger, C., Williamson, A. R., & Louppe, G. (2020). Lightning-fast gravitational wave parameter inference through neural amortization. *arXiv preprint arXiv:2010.12931*.

Dinh, L., Krueger, D., & Bengio, Y. (2015). NICE: non-linear independent components estimation. *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Workshop Track Proceedings*.

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

Hermans, J., Begy, V., & Louppe, G. (2020). Likelihood-free MCMC with amortized approximate ratio estimators. In H. D. III & A. Singh (Eds.), *Proceedings of the 37th international conference on machine learning* (pp. 4239–4248, Vol. 119). PMLR.

Durkan, C., Murray, I., & Papamakarios, G. (2020). On contrastive learning for likelihood-free inference. *International Conference on Machine Learning*, 2771–2781.

Miller, B. K. [Benjamin K], Weniger, C., & Forré, P. (2022). Contrastive neural ratio estimation. *Advances in Neural Information Processing Systems*, *35*, 3262–3278.

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

ATLAS, C., et al. (2024). An implementation of neural simulation-based inference for parameter estimation in atlas. *arXiv preprint arXiv:2412.01600*.

Brehmer, J., Mishra-Sharma, S., Hermans, J., Louppe, G., & Cranmer, K. (2019). Mining for dark matter substructure: Inferring subhalo population properties from strong lenses with machine learning. *The Astrophysical Journal*, *886*(1), 49.

Stoye, M., Brehmer, J., Louppe, G., Pavez, J., & Cranmer, K. (2018). Likelihood-free inference with an improved cross-entropy estimator. *arXiv preprint arXiv:1808.00973*.

Dalmasso, N., Izbicki, R., & Lee, A. (2020). Confidence sets and hypothesis testing in a likelihood-free inference setting. *International Conference on Machine Learning*, 2323–2334.

Geffner, T., Papamakarios, G., & Mnih, A. (2022). Score modeling for simulation-based inference. *arXiv preprint arXiv:2209.14249*.

Sharrock, L., Simons, J., Liu, S., & Beaumont, M. (2022). Sequential neural score estimation: Likelihood-free inference with conditional score based diffusion models. *arXiv preprint arXiv:2210.04872*.

Linhart, J., Cardoso, G. V., Gramfort, A., Corff, S. L., & Rodrigues, P. L. (2024). Diffusion posterior sampling for simulation-based inference in tall data settings. *arXiv preprint arXiv:2404.07593*.

Gloeckler, M., Toyota, S., Fukumizu, K., & Macke, J. H. (2024). Compositional simulation-based inference for time series. *arXiv preprint arXiv:2411.02728*.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., & Poole, B. (2020). Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*.

Papamakarios, G., Sterratt, D., & Murray, I. (2019). Sequential neural likelihood: Fast likelihood-free inference with autoregressive flows. *The 22nd International Conference on Artificial Intelligence and Statistics*, 837–848.

Papamakarios, G., & Murray, I. (2016). Fast ε-free inference of simulation models with bayesian conditional density estimation. *Advances in neural information processing systems*, 1028–1036.

Lueckmann, J.-M., Goncalves, P. J., Bassetto, G., Öcal, K., Nonnenmacher, M., & Macke, J. H. (2017). Flexible statistical inference for mechanistic models of neural dynamics. *Advances in Neural Information Processing Systems*, *30*.

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

Miller, B. K. [Benjamin K], Cole, A., Forré, P., Louppe, G., & Weniger, C. (2021). Truncated marginal neural ratio estimation. *Advances in Neural Information Processing Systems*, *34*, 129–143.

Deistler, M., Goncalves, P. J., & Macke, J. H. (2022). Truncated proposals for scalable and hassle-free simulation-based inference. *Advances in Neural Information Processing Systems*, *35*, 23135–23149.

Ramesh, P., Lueckmann, J.-M., Boelts, J., Tejero-Cantero, Á., Greenberg, D. S., Gonçalves, P. J., & Macke, J. H. (2022). Gatsbi: Generative adversarial training for simulation-based inference. *arXiv preprint arXiv:2203.06481*.

Pacchiardi, L., & Dutta, R. (2022a). Likelihood-free inference with generative neural networks via scoring rule minimization. *arXiv preprint arXiv:2205.15784*.

Glaser, P., Arbel, M., Hromadka, S., Doucet, A., & Gretton, A. (2022). Maximum likelihood learning of unnormalized models for simulation-based inference. *arXiv preprint arXiv:2210.14756*.

Schmitt, M., Pratz, V., Köthe, U., Bürkner, P.-C., & Radev, S. T. (2023). Consistency models for scalable and fast simulation-based inference. *CoRR*.

Wildberger, J., Dax, M., Buchholz, S., Green, S., Macke, J. H., & Schölkopf, B. (2024). Flow matching for scalable simulation-based inference. *Advances in Neural Information Processing Systems*, *36*.

Häggström, H., Rodrigues, P. L., Oudoumanessah, G., Forbes, F., & Picchini, U. (2024). Fast, accurate and lightweight sequential simulation-based inference using gaussian locally linear mappings. *arXiv preprint arXiv:2403.07454*.

Glöckler, M., Deistler, M., & Macke, J. H. (2021). Variational methods for simulation-based inference. *International Conference on Learning Representations*.

Radev, S. T., Schmitt, M., Pratz, V., Picchini, U., Köthe, U., & Bürkner, P.-C. (2023). Jana: Jointly amortized neural approximation of complex bayesian models. *Uncertainty in Artificial Intelligence*, 1695–1706.

Schmitt, M., Ivanova, D. R., Habermann, D., Koethe, U., Bürkner, P.-C., & Radev, S. T. (2024). Leveraging self-consistency for data-efficient amortized bayesian inference. *Forty-first International Conference on Machine Learning*.

Rodrigues, P. L., Moreau, T., Louppe, G., & Gramfort, A. (2021). Leveraging global parameters for flow-based neural posterior estimation. *arXiv preprint arXiv:2102.06477*.

Heinrich, L., Mishra-Sharma, S., Pollard, C., & Windischhofer, P. (2023). Hierarchical neural simulation-based inference over event ensembles. *arXiv preprint arXiv:2306.12584*.

Rozet, F., & Louppe, G. (2021a). Arbitrary marginal neural ratio estimation for simulation-based inference. *arXiv preprint arXiv:2110.00449*.

Rozet, F., & Louppe, G. (2023). Score-based data assimilation. *Advances in Neural Information Processing Systems*, *36*, 40521–40541.

Gloeckler, M., Deistler, M., Weilbach, C., Wood, F., & Macke, J. H. (2024). All-in-one simulation-based inference. *arXiv preprint arXiv:2404.09636*.

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

Lemos, P., Coogan, A., Hezaveh, Y., & Perreault-Levasseur, L. (2023). Sampling-based accuracy testing of posterior estimators for general inference. *arXiv preprint arXiv:2302.03026*.

Linhart, J., Gramfort, A., & Rodrigues, P. L. (2022). Validation diagnostics for sbi algorithms based on normalizing flows. *arXiv preprint arXiv:2211.09602*.

Lopez-Paz, D., & Oquab, M. (2017). Revisiting classifier two-sample tests. *International Conference on Learning Representations*.

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

Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B., & Smola, A. (2012). A kernel two-sample test. *The Journal of Machine Learning Research*, *13*(1), 723–773.

Cranmer, K., Brehmer, J., & Louppe, G. (2020). The frontier of simulation-based inference. *Proceedings of the National Academy of Sciences*.

Lueckmann, J.-M., Boelts, J., Greenberg, D., Goncalves, P., & Macke, J. (2021). Benchmarking simulation-based inference. In A. Banerjee & K. Fukumizu (Eds.), *Proceedings of the 24th international conference on artificial intelligence and statistics* (pp. 343–351, Vol. 130). PMLR.

Aad, G., Abajyan, T., Abbott, B., Abdallah, J., Khalek, S. A., Abdelalim, A. A., Aben, R., Abi, B., Abolins, M., AbouZeid, O., et al. (2012). Observation of a new particle in the search for the standard model higgs boson with the atlas detector at the lhc. *Physics Letters B*, *716*(1), 1–29.

Gilman, D., Birrer, S., Treu, T., Keeton, C. R., & Nierenberg, A. (2018). Probing the nature of dark matter by forward modelling flux ratios in strong gravitational lenses. *Monthly Notices of the Royal Astronomical Society*, *481*(1), 819–834.

Aghanim, N., et al. (2020). Planck 2018 results. VI. Cosmological parameters [[Erratum: Astron.Astrophys. 652, C4 (2021)]]. *Astron. Astrophys.*, *641*, A6. https://doi.org/10.1051/0004-6361/201833910

Abbott, B. P., Abbott, R., Abbott, T., Abernathy, M., Acernese, F., Ackley, K., Adams, C., Adams, T., Addesso, P., Adhikari, R., et al. (2016). Gw151226: Observation of gravitational waves from a 22-solar-mass binary black hole coalescence. *Physical review letters*, *116*(24), 241103.

Gonçalves, P. J., Lueckmann, J.-M., Deistler, M., Nonnenmacher, M., Öcal, K., Bassetto, G., Chintaluri, C., Podlaski, W. F., Haddad, S. A., Vogels, T. P., et al. (2020). Training deep neural density estimators to identify mechanistic models of neural dynamics. *Elife*, *9*, e56261.

Brehmer, J., Cranmer, K., Louppe, G., & Pavez, J. (2018). A Guide to Constraining Effective Field Theories with Machine Learning. *Phys. Rev. D*, *98*(5), 052004. https://doi.org/10.1103/PhysRevD.98.052004

Hermans, J., Banik, N., Weniger, C., Bertone, G., & Louppe, G. (2021). Towards constraining warm dark matter with stellar streams through neural simulation-based inference. *Monthly Notices of the Royal Astronomical Society*, *507*(2), 1999–2011.

Geweke, J. F., et al. (1991). *Evaluating the accuracy of sampling-based approaches to the calculation of posterior moments* (tech. rep.). Federal Reserve Bank of Minneapolis.

Gelman, A., & Rubin, D. B. [Donald B]. (1992). Inference from iterative simulation using multiple sequences. *Statistical science*, *7*(4), 457–472.

Raftery, A. E., & Lewis, S. (1991). *How many iterations in the gibbs sampler?* (Tech. rep.).

Dixit, A., & Roy, V. (2017). Mcmc diagnostics for higher dimensions using kullback leibler divergence. *Journal of Statistical Computation and Simulation*, *87*(13), 2622–2638.

Box, G. E., & Tiao, G. C. (1973). *Bayesian inference in statistical analysis* (Vol. 40). John Wiley & Sons.

Hyndman, R. J. (1996). Computing and graphing highest density regions. *The American Statistician*, *50*(2), 120–126.

Lehmann, E. L., & Romano, J. P. (2006). *Testing statistical hypotheses*. Springer Science & Business Media.

Bengio, Y., Yao, L., & Cho, K. (2014). Bounding the test log-likelihood of generative models. *International Conference on Learning Representations*.

Dziugaite, G. K., Roy, D. M., & Ghahramani, Z. (2015). Training generative neural networks via maximum mean discrepancy optimization. *Proceedings of the Thirty-First Conference on Uncertainty in Artificial Intelligence*, 258–267.

Schall, R. (2012). The empirical coverage of confidence intervals: Point estimates and confidence intervals for confidence levels. *Biometrical journal*, *54*(4), 537–551.

Prangle, D., Blum, M. G., Popovic, G., & Sisson, S. (2014). Diagnostic tools for approximate bayesian computation using the coverage property. *Australian & New Zealand Journal of Statistics*, *56*(4), 309–329.

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

Thomas, O., Dutta, R., Corander, J., Kaski, S., Gutmann, M. U., et al. (2016). Likelihood-free inference by ratio estimation. *Bayesian Analysis*.

Sugiyama, M., Suzuki, T., & Kanamori, T. (2012). Density-ratio matching under the bregman divergence: A unified framework of density-ratio estimation. *Annals of the Institute of Statistical Mathematics*, *64*(5), 1009–1044.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative adversarial nets. *Advances in neural information processing systems*, *27*.

Dinh, L., Sohl-Dickstein, J., & Bengio, S. (2017). Density estimation using real NVP. *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*.

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

Breiman, L. (1996). Bagging predictors. *Machine learning*, *24*(2), 123–140.

Price, L. F., Drovandi, C. C., Lee, A., & Nott, D. J. (2018). Bayesian synthetic likelihood. *Journal of Computational and Graphical Statistics*, *27*(1), 1–11.

Frazier, D. T., Nott, D. J., Drovandi, C., & Kohn, R. (2022). Bayesian inference using synthetic likelihood: Asymptotics and adjustments. *Journal of the American Statistical Association*, (just-accepted), 1–28.

Cranmer, K., Heinrich, L., Head, T., & Louppe, G. (2017). "Active Sciencing" with Reusable Workflows.

Blum, M. G., & François, O. (2010). Non-linear regression models for approximate bayesian computation. *Statistics and computing*, *20*(1), 63–73.

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

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

Banik, N., Bertone, G., Bovy, J., & Bozorgnia, N. (2018). Probing the nature of dark matter particles with stellar streams. *Journal of Cosmology and Astroparticle Physics*, *2018*(07), 061.

LIGO Scientific Collaboration. (2018). LIGO Algorithm Library - LALSuite. https://doi.org/10.7935/GT1W-FZ16

Biwer, C. M., Capano, C. D., De, S., Cabero, M., Brown, D. A., Nitz, A. H., & Raymond, V. (2019). PyCBC Inference: A Python-based parameter estimation toolkit for compact binary coalescence signals. *Publ. Astron. Soc. Pac.*, *131*(996), 024503. https://doi.org/10.1088/1538-3873/aaef0b

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

Rozet, F., & Louppe, G. (2021b). *Arbitrary marginal neural ratio estimation for likelihood-free inference* [Master's thesis, University of Liège, Belgium].

Lin, J. (2014). An integrated procedure for bayesian reliability inference using mcmc. *Journal of Quality and Reliability Engineering*, *2014*.

Hogg, D. W., & Foreman-Mackey, D. (2018). Data analysis recipes: Using markov chain monte carlo. *The Astrophysical Journal Supplement Series*, *236*(1), 11.

Dixit, A. U. (2018). *Developments in mcmc diagnostics and sparse bayesian learning models* [Doctoral dissertation, Iowa State University].

Roy, V. (2020). Convergence diagnostics for markov chain monte carlo. *Annual Review of Statistics and Its Application*, *7*, 387–412.

Xing, H., Nicholls, G., & Lee, J. K. (2020). Distortion estimates for approximate bayesian inference. *Conference on Uncertainty in Artificial Intelligence*, 1208–1217.

Frazier, D. T., Martin, G. M., Robert, C. P., & Rousseau, J. (2018). Asymptotic properties of approximate bayesian computation. *Biometrika*, *105*(3), 593–607.

Dalmasso, N., Zhao, D., Izbicki, R., & Lee, A. B. (2021). Likelihood-free frequentist inference: Bridging classical statistics and machine learning in simulation and uncertainty quantification. *arXiv preprint arXiv:2107.03920*.

Schmon, S. M., Cannon, P. W., & Knoblauch, J. (2020). Generalized posteriors in approximate bayesian computation. *arXiv preprint arXiv:2011.08644*.

Matsubara, T., Knoblauch, J., Briol, F.-X., & Oates, C. J. (2022). Robust generalised bayesian inference for intractable likelihoods. *Journal of the Royal Statistical Society: Series B*.

Pacchiardi, L., & Dutta, R. (2022b). Score matched neural exponential families for likelihood-free inference. *Journal of Machine Learning Research*, *23*(38), 1–71.

Dellaporta, C., Knoblauch, J., Damoulas, T., & Briol, F.-X. (2022). Robust bayesian inference for simulator-based models via the mmd posterior bootstrap. *International Conference on Artificial Intelligence and Statistics*, 943–970.

Dimitriou, A., Figueroa, D. G., & Zaldivar, B. (2023). Fast likelihood-free reconstruction of gravitational wave backgrounds. *arXiv preprint arXiv:2309.08430*.

Vasist, M., Rozet, F., Absil, O., Mollière, P., Nasedkin, E., & Louppe, G. (2023). Neural posterior estimation for exoplanetary atmospheric retrieval. *Astronomy & Astrophysics*, *672*, A147.

Berteaud, J., Eckner, C., Calore, F., Clavel, M., & Haggard, D. (2024). Simulation-based inference of radio millisecond pulsars in globular clusters. *arXiv preprint arXiv:2405.15691*.

Reza, M., Zhang, Y., Avestruz, C., Strigari, L. E., Shevchuk, S., & Villaescusa-Navarro, F. (2024). Constraining cosmology with simulation-based inference and optical galaxy cluster abundance. *arXiv preprint arXiv:2409.20507*.

Swierc, P., Tamargo-Arizmendi, M., Ćiprijanović, A., & Nord, B. D. (2024). Domain-adaptive neural posterior estimation for strong gravitational lens analysis. *arXiv preprint arXiv:2410.16347*.

Morandini, A., Ferber, T., & Kahlhoefer, F. (2024). Reconstructing axion-like particles from beam dumps with simulation-based inference. *The European Physical Journal C*, *84*(2), 1–26.

Rozet, François. (2023). *Lampe*. https://doi.org/10.5281/zenodo.8405783

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

Shestopaloff, A. Y., & Neal, R. M. (2014). On bayesian inference for the m/g/1 queue with efficient mcmc sampling. *arXiv preprint arXiv:1401.5548*.

Bissiri, P. G., Holmes, C. C., & Walker, S. G. (2016). A general framework for updating belief distributions. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *78*(5), 1103–1130.

Pacchiardi, L., & Dutta, R. (2021). Generalized bayesian likelihood-free inference using scoring rules estimators. *arXiv preprint arXiv:2104.03889*.

Grünwald, P., & Van Ommen, T. (2017). Inconsistency of bayesian inference for misspecified linear models, and a proposal for repairing it. *Bayesian Analysis*, *12*(4), 1069–1103.

Holmes, C. C., & Walker, S. G. (2017). Assigning a value to a power likelihood in a general bayesian model. *Biometrika*, *104*(2), 497–503.

Miller, J. W., & Dunson, D. B. (2018). Robust bayesian inference via coarsening. *Journal of the American Statistical Association*.

Sisson, S. A., Fan, Y., & Beaumont, M. A. (2018). Overview of abc. In *Handbook of approximate bayesian computation* (pp. 3–54). Chapman; Hall/CRC.

Gratton, S. (2017). Glass: A general likelihood approximate solution scheme. *arXiv preprint arXiv:1708.08479*.

Tran, D., Ranganath, R., & Blei, D. M. (2017). Hierarchical implicit models and likelihood-free variational inference. *arXiv preprint arXiv:1702.08896*.

Cook, S. R., Gelman, A., & Rubin, D. B. (2006). Validation of software for bayesian models using posterior quantiles. *Journal of Computational and Graphical Statistics*, *15*(3), 675–692.

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

Alsing, J., Charnock, T., Feeney, S., & 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.

Masserano, L., Dorigo, T., Izbicki, R., Kuusela, M., & 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*.

Cannon, P., Ward, D., & Schmon, S. M. (2022). Investigating the impact of model misspecification in neural simulation-based inference. *arXiv preprint arXiv:2209.01845*.

Bishop, C. M. (1994). Mixture density networks. *Technical Report*.

Papamakarios, G., Nalisnick, E., Rezende, D. J., Mohamed, S., & Lakshminarayanan, B. (2019). Normalizing flows for probabilistic modeling and inference. *arXiv preprint arXiv:1912.02762*.

Sason, I., & Verdú, S. (2016). f-divergence inequalities. *IEEE Transactions on Information Theory*, *62*(11), 5973–6006.

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

Rozet, F. (2022). Zuko. https://doi.org/10.5281/zenodo.7625672

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.

Zhang, Y., & 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.

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

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

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

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.

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.

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

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

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

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

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*.

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

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

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

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

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

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

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

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

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

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

Tran, B.-H., Rossi, S., Milios, D., & 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.

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

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

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

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

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

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

Vovk, V. (2012). Conditional validity of inductive conformal predictors. *Asian conference on machine learning*, 475–490.

Bernardo, J. M. (2005). Intrinsic credible regions: An objective bayesian approach to interval estimation. *Test*, *14*, 317–384.

Gloeckler, M., Deistler, M., & Macke, J. H. (2023). Adversarial robustness of amortized bayesian inference. *Proceedings of the 40th International Conference on Machine Learning*, 11493–11524.

Wehenkel, A., Gamella, J. L., Sener, O., Behrmann, J., Sapiro, G., Cuturi, M., & Jacobsen, J.-H. (2024). Addressing misspecification in simulation-based inference through data-driven calibration. *arXiv preprint arXiv:2405.08719*.