Robust Neural Posterior Estimation and Statistical Model Criticism

Anonymous Author(s)
Affiliation
Address
email

Abstract

Computer simulations have proven a valuable tool for understanding complex phenomena across the sciences. However, the utility of simulators for modelling and forecasting purposes is often restricted by low data quality, as well as practical limits to model fidelity. In order to circumvent these difficulties, we argue that modellers must treat simulators as idealistic representations of the true data generating process, and consequently should thoughtfully consider the risk of *model misspecification*. In this work we revisit neural posterior estimation (NPE), a class of algorithms that enable black-box parameter inference in simulation models, and consider the implication of a simulation-to-reality gap. While recent works have demonstrated reliable performance of these methods, the analyses have been performed using synthetic data generated by the simulator model itself, and have therefore only addressed the well-specified case. In this paper, we find that the presence of misspecification, in contrast, leads to unreliable inference when NPE is used naïvely. As a remedy we argue that principled scientific inquiry with simulators should incorporate a *model criticism* component, to facilitate interpretable identification of misspecification and a *robust inference* component, to fit "wrong but useful" models. We propose robust neural posterior estimation (RNPE), an extension of NPE to simultaneously achieve both these aims, through explicitly modelling the discrepancies between simulations and the observed data. We assess the approach on a range of artificially misspecified examples, and find RNPE performs well across the tasks, whereas naïvely using NPE leads to misleading and erratic posteriors.

1 Introduction

Stochastic simulators have become a ubiquitous modelling tool across the sciences and are regularly applied to some of the most complex and challenging problems of scientific interest, including climate change (see e.g. Randall et al., 2007), particle physics (e.g. Brehmer et al., 2020), and the Covid-19 pandemic (e.g. Ferguson et al., 2020). Simulators implicitly define a likelihood function $p(\boldsymbol{x} \mid \boldsymbol{\theta})$, where $\boldsymbol{x}$ is the simulator output and $\boldsymbol{\theta}$ are the simulator parameters. Although running the simulator to sample from the model is straightforward, the inherent complexity of simulators often makes analytic calculation of the likelihood intractable. As a result, classical inference techniques to find the parameter posterior $p(\boldsymbol{\theta} \mid \boldsymbol{x})$ such as Markov chain Monte Carlo (MCMC) (Metropolis et al., 1953) are infeasible. To overcome this issue, a large family of simulation-based inference (SBI) methods have been developed that allow parameter inference to be performed on arbitrary "black-box" simulators (see Cranmer et al., 2020). Broadly, these approaches estimate a function that allows access to an approximate posterior. This can be achieved by approximating the posterior directly (Greenberg et al., 2019; Lueckmann et al., 2017), or indirectly, via approximating the likelihood (Papamakarios et al., 2019b) or likelihood-to-evidence ratio (Hermans et al., 2020; Thomas et al., 2022), from which the

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

posterior can be sampled using MCMC. For estimating the posterior or likelihood, neural posterior estimation (NPE) and neural likelihood estimation (NLE) have shown to be powerful approaches (Lueckmann et al., 2021), which rely on neural network-based conditional density estimators, such as normalising flows, to approximating the likelihood or posterior (Papamakarios et al., 2019a).

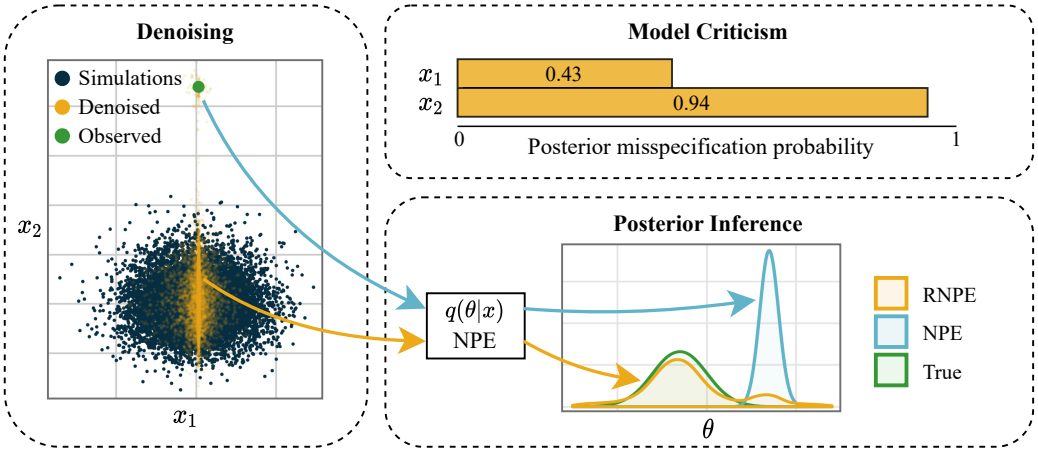


Figure 1: Overview of the robust neural posterior estimation (RNPE) framework. Through "denoising" the observed data, we can simultaneously perform *model criticism* to identify how the model is misspecified, and perform inference *robust* to model misspecification. See Gaussian example in section 4.2 for details of the experiment.

To define misspecification, it is necessary to make a clear distinction between the true data generating process and the simulator: we will write $\boldsymbol{y}_o$ for observed data, following some unknown data generating process with distribution denoted p^*, and we will write $\boldsymbol{x} \sim p(\boldsymbol{x} \mid \boldsymbol{\theta})$ to denote samples from the simulator. In many applications of SBI, summary statistics are used to reduce the dimensionality of the problem. If this is the case, we use $\boldsymbol{x}$ and $\boldsymbol{y}_o$ to denote the simulated and observed summary statistics, rather than the raw data, and treat the raw data as an unobserved latent of the model. The simulator is said to be *misspecified* if the true data generating process does not fall within the family of distributions defined by the simulator, i.e. $p^* \notin \{p(\boldsymbol{x} \mid \boldsymbol{\theta});\ \boldsymbol{\theta} \in \boldsymbol{\Theta}\}$. Simulators generally form idealised and simplistic representations of a more complex data generating process, and hence – like all statistical models – will be "wrong" to some extent. This misspecification often results in a discrepancy between simulated and observed data, sometimes termed a "simulation-to-reality gap". Despite this, the general approach in SBI is to learn a posterior approximation $q(\boldsymbol{\theta} \mid \boldsymbol{x})$ (either directly, or indirectly), using simulated data, $(\boldsymbol{x}, \boldsymbol{\theta}) \sim p(\boldsymbol{x} \mid \boldsymbol{\theta})p(\boldsymbol{\theta})$, and to condition on the observed data, $q(\boldsymbol{\theta} \mid \boldsymbol{x} = \boldsymbol{y}_o)$. This approach is only theoretically justified under the assumption that the simulator is well-specified; relying on this approach for misspecified simulators leads to two issues:

(i) As the posterior approximation $q(\boldsymbol{\theta} \mid \boldsymbol{x})$ is learned using samples from the simulator, this implies that we can only expect it to form a reasonable approximation of $p(\boldsymbol{\theta} \mid \boldsymbol{x})$ for regions well covered by simulations. For misspecified simulators, the observed data $\boldsymbol{y}_o$ may be very unlikely in $p(\boldsymbol{x}) = \int p(\boldsymbol{x}|\boldsymbol{\theta})p(\boldsymbol{\theta})d\boldsymbol{\theta}$, in which case the approximation $q(\boldsymbol{\theta} \mid \boldsymbol{x} = \boldsymbol{y}_o)$ will likely be poor. In some cases, $\boldsymbol{y}_o$ may even fall outside the support of $p(\boldsymbol{x})$, which results in NPE attempting to extrapolate to estimate the undefined posterior.

(ii) It has been demonstrated more generally that Bayesian inference is frequently problematic under misspecification. Even for simple models, misspecification can lead to posterior concentration around "bad" models (Grünwald and Langford, 2007; Grünwald and Van Ommen, 2017) and poorly calibrated credible regions (Kleijn and van der Vaart, 2012; Syring and Martin, 2019). These issues will persist even in the unlikely case that $q(\boldsymbol{\theta} \mid \boldsymbol{x})$ perfectly approximates $p(\boldsymbol{\theta} \mid \boldsymbol{x})$.

One approach to improving robustness is to incorporate an "error model" within the simulator itself, such as by applying additive noise to the output to ensure that the observed data has reasonable marginal likelihood under the model (Cranmer et al., 2020). However, this approach has some

limitations: *i)* it is challenging to interpret how the model is misspecified - an obvious choice is to use posterior predictive checks, but poor predictive performance could result from failed inference, limitations of simulator, or both; *ii)* retraining of the approximate posterior would be required to investigate different error models, which can be computationally expensive, *iii)* the error model itself is likely tractable, which can be utilised to improve inference.

Our contribution. We propose robust neural posterior estimation (RNPE), an SBI framework that incorporates an error model, $p(\boldsymbol{y} \mid \boldsymbol{x})$ to account for discrepancies between simulations and the observed data, whilst avoiding the limitations of naïvely incorporating it directly into the simulator. We explicitly learn to invert the error model for the observed data prior to inference of the parameter posterior, a process we call "denoising". This approach yields the following advantages:

Efficient Model Criticism. Understanding a models limitations is crucial for principled model development (Box, 1980; Gelman and Shalizi, 2013). By probabilistically inverting the error process for the observed data, and inferring any associated error model parameters, we can detect and interpret discrepancies between the simulated and observed data, allowing criticism of the simulator. This process can be achieved in a manner decoupled from inference, which provides efficiency benefits, whilst avoiding confounding inference failures with problems arising due to limitations of the simulator.

Robust Inference. Even after several rounds of model development and model criticism, the "best" available model will still frequently be misspecified to an appreciable degree. However, it is often still desirable to use "wrong but useful" models for parameter inference and predictions, and hence the inference process should be robust to misspecification. By exploiting the amortisation[1] property of NPE, we can ensemble posterior estimates over a set of "denoised" observations, leading to posterior estimates robust to misspecification.

An overview of the RNPE framework can be seen in Fig. 1. We examine the practical value of this approach for *model criticism* and *robust inference* on examples in which we artificially introduce realistic levels of misspecification.

2 Method

2.1 Error Model

As outlined above, we can explicitly model discrepancies between the observed data and simulations by introducing an error model $p(\boldsymbol{y} \mid \boldsymbol{x})$, such that the assumed generative model is

$$p(\boldsymbol{\theta}, \boldsymbol{x}, \boldsymbol{y}) = p(\boldsymbol{y} \mid \boldsymbol{x}) p(\boldsymbol{x} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta}), \tag{1}$$

where we assume that the error is independent of the simulator parameters $\boldsymbol{\theta}$, and we treat $\boldsymbol{x}$ as an unobserved latent of the model. Using the equivalent factorisation $p(\boldsymbol{\theta}, \boldsymbol{x}, \boldsymbol{y}) = p(\boldsymbol{y}) p(\boldsymbol{x} \mid \boldsymbol{y}) p(\boldsymbol{\theta} \mid \boldsymbol{x})$, we can marginalise over the latent $\boldsymbol{x}$ and find an expression for the parameter posterior

$$p(\boldsymbol{\theta}, \boldsymbol{y}) = p(\boldsymbol{y}) \int p(\boldsymbol{x} \mid \boldsymbol{y}) p(\boldsymbol{\theta} \mid \boldsymbol{x}) \, \mathrm{d}\boldsymbol{x}, \tag{2}$$

$$p(\boldsymbol{\theta} \mid \boldsymbol{y}) = \int p(\boldsymbol{x} \mid \boldsymbol{y}) p(\boldsymbol{\theta} \mid \boldsymbol{x}) \, \mathrm{d}\boldsymbol{x} = \mathbb{E}_{\boldsymbol{x} \sim p(\boldsymbol{x} \mid \boldsymbol{y})}[p(\boldsymbol{\theta} \mid \boldsymbol{x})]. \tag{3}$$

Equation (3), implies that if we had access to $p(\boldsymbol{x} \mid \boldsymbol{y})$ and $p(\boldsymbol{\theta} \mid \boldsymbol{x})$, we could sample from the parameter posterior by first sampling the posterior over the latent variables $\boldsymbol{x} \sim p(\boldsymbol{x} | \boldsymbol{y})$, and then sampling from $p(\boldsymbol{\theta} \mid \boldsymbol{x})$. This approach reverses the new assumed data generating process from Equation (1) whilst propagating uncertainty throughout, first removing the error from the observed data by "denoising", and then finding the associated simulator parameters given the "denoised" data. A Monte Carlo approximation of the expectation in Equation (3) could also be used to estimate the posterior density $p(\boldsymbol{\theta} \mid \boldsymbol{y})$, i.e.

$$p(\boldsymbol{\theta} \mid \boldsymbol{y}) \approx \frac{1}{M} \sum_{m=1}^{M} p(\boldsymbol{\theta} \mid \tilde{\boldsymbol{x}}_m), \qquad \tilde{\boldsymbol{x}}_1, \ldots, \tilde{\boldsymbol{x}}_M, \overset{\text{iid}}{\sim} p(\boldsymbol{x} \mid \boldsymbol{y}), \tag{4}$$

[1] Amortisation refers to the ability of a conditional distribution approximation to condition on arbitrary instances of the conditioning variable, rather than being specialised to a particular instance.

In practice, $p(\boldsymbol{x} \mid \boldsymbol{y})$ and $p(\boldsymbol{\theta} \mid \boldsymbol{x})$ are unknown. We approximate $p(\boldsymbol{\theta} \mid \boldsymbol{x})$ by training a normalising flow $q(\boldsymbol{\theta} \mid \boldsymbol{x})$ on simulated data $(\boldsymbol{x}, \boldsymbol{\theta}) \sim p(\boldsymbol{x} \mid \boldsymbol{\theta})p(\boldsymbol{\theta})$, as is commonly done for NPE. To approximately sample from $p(\boldsymbol{x} \mid \boldsymbol{y})$, we first train a normalising flow, $q(\boldsymbol{x})$, fitted to samples from the prior predictive distribution of the simulator $\boldsymbol{x} \sim p(\boldsymbol{x})$. Given an error model $p(\boldsymbol{y} \mid \boldsymbol{x})$ specified *a priori*, we can then approximately sample $\tilde{\boldsymbol{x}} \sim p(\boldsymbol{x} \mid \boldsymbol{y})$ using MCMC, as $\hat{p}(\boldsymbol{x} \mid \boldsymbol{y}) \propto p(\boldsymbol{y} \mid \boldsymbol{x})q(\boldsymbol{x})$. Sampling of any error model parameters can also be included at this step if desired. We denote samples from $\hat{p}(\boldsymbol{x} \mid \boldsymbol{y})$ and $q(\boldsymbol{\theta} \mid \boldsymbol{x})$ as $\tilde{\boldsymbol{x}}$ and $\tilde{\boldsymbol{\theta}}$ to avoid confusion with prior samples and simulations. Pseudo-code for the overall approach is given in Algorithm 1. Note that variations of the algorithm are possible. For example, instead of the Monte Carlo composition approach in line 6, one could train a neural posterior directly on a simulator equipped with our noise model. We refer to the latter as noisy neural posterior estimation (NNPE). In practice, RNPE has some advantages including the fact that the underlying NPE algorithm does not require retraining whenever the noise model is changed.

Algorithm 1: Robust neural posterior estimation (RNPE)

for i *in* $1 : N$ **do**
1 Sample $\boldsymbol{\theta}_i \sim p(\boldsymbol{\theta})$
2 Simulate $\boldsymbol{x}_i \sim p(\boldsymbol{x} \mid \boldsymbol{\theta})$
end
3 Train $q(\boldsymbol{x})$ on $\{\boldsymbol{x}_i\}_{i=1}^N$
4 Sample $\tilde{\boldsymbol{x}}_m \sim \hat{p}(\boldsymbol{x} \mid \boldsymbol{y}_o) \propto p(\boldsymbol{y}_o \mid \boldsymbol{x})q(\boldsymbol{x}),\ m = 1, \ldots, M$ using MCMC
5 Train NPE $q(\boldsymbol{\theta} \mid \boldsymbol{x})$ on $\{(\boldsymbol{\theta}_i, \boldsymbol{x}_i)\}_{i=1}^N$
6 Sample $\tilde{\boldsymbol{\theta}}_m \sim q(\boldsymbol{\theta} \mid \tilde{\boldsymbol{x}}_m),\ m = 1, \ldots, M$
return $\{(\tilde{\boldsymbol{\theta}}_m, \tilde{\boldsymbol{x}}_m)\}_{m=1}^M$, samples approximately from $p(\boldsymbol{\theta}, \boldsymbol{x} \mid \boldsymbol{y}_o)$

Under this framework, the standard SBI approach can be retrieved as the special case in which $p(\boldsymbol{y} \mid \boldsymbol{x}) = p(\boldsymbol{x} \mid \boldsymbol{y}) = \delta(\boldsymbol{x} - \boldsymbol{y})$, where δ is the Dirac delta distribution, meaning that we revert to assuming no discrepancy between the simulator and data generating process.

The idealised nature of simulation models implies that frequently there will be some characteristics of the observed data that are well captured by simulations, and some aspects in which there exists a discrepancy. Using a set of summary statistics for inference, rather than the raw data naturally captures an interpretable, low dimensional and diverse set of characteristics of the simulated and observed data. Because of this, we find that in general, reducing the raw data to summary statistics is advantageous for *model criticism*, as the practitioner can choose summary statistics which align with their belief about important model characteristics, and easily interpret any discrepancies (see Section 4.4). The requirement for domain specific knowledge to develop summary statistics has been alleviated by recent work (e.g. Fearnhead and Prangle, 2012; Chan et al., 2018; Chen et al., 2020), which seek to automatically embed the raw data to summary statistics; however, these embeddings often lack a tangible interpretation. Alternatively, both hand-crafted and automatic summary statistics could be used in tandem. Hereafter, however, we perform *model criticism* and *inference* using hand-crafted summary statistics, such that $\boldsymbol{x}$ and $\boldsymbol{y}_o$ refer to simulated and observed summary statistics, *not* the raw data.

2.2 Spike and Slab

In practice, the true error model is unknown, and hence the error model should describe our prior belief over the discrepancy between the data generating process and simulations. Additionally, to facilitate model criticism, it should ideally allow easy assessment of which summary statistics are approximately well-specified and which are misspecified. Inspired by sparsity-inducing priors used in Bayesian regression literature (George and McCulloch, 1993), we use a "spike and slab" error model on each summary statistic

$$\boldsymbol{x} \sim q(\boldsymbol{x}), \tag{5}$$

$$z_j \sim \text{Bernoulli}(\rho), \qquad j = 1, \ldots, D \tag{6}$$

$$y_j \mid x_j, z_j = 0 \sim N(x_j, \sigma^2), \qquad j = 1, \ldots, D \tag{7}$$

$$y_j \mid x_j, z_j = 1 \sim \text{Cauchy}(x_j, \tau), \qquad j = 1, \ldots, D \tag{8}$$

where $\boldsymbol{x} \in \mathbb{R}^D$. As the summary statistics have varying scales, we standardise them to have mean zero and variance one. The Bernoulli variables indicate whether the j-th summary statistic is approximately well-specified; we use the probability of $\rho = 0.5$ to express the belief that it is equally likely a summary statistic will be misspecified or well-specified *a priori*. When a summary statistic is well-specified (i.e., $z_j = 0$), we take the error distribution to be a tight Gaussian (a spike) centred on x_j , choosing $\sigma = 0.01$ to enforce consistency between the simulator and the observed data for the j-th statistic. In contrast, in the misspecified case (i.e., $z_j = 1$), we take the error model to be the much wider and heavy tailed Cauchy distribution (a slab). Generally, we might expect misspecification to be relatively subtle, but some model inadequacies can lead to catastrophically misspecified summary statistics. We chose the Cauchy scale $\tau = 0.25$, to reflect this, as it places half the mass within ± 0.25 standard deviations of x_j, but the long tails accommodate summary statistics that are very misspecified. The sparsity inducing effect of this error model implies that we expect this choice to perform well when misspecification itself is sparse, that is that a proportion of the summary statistics could reasonably be jointly consistent with the simulator, whereas others may be incompatible. By marginalising over $\boldsymbol{z}$, the spike and slab error model can also be written as

$$p(\boldsymbol{y} \mid \boldsymbol{x}) = \prod_{j=1}^{D} \left[(1-\rho) \cdot p(y_j \mid x_j, z_j = 0) + \rho \cdot p(y_j \mid x_j, z_j = 1) \right], \tag{9}$$

and as such can be viewed as an equally weighted mixture of the Gaussian spike, $p(y_j \mid x_j, z_j = 0)$, and the Cauchy slab, $p(y_j \mid x_j, z_j = 1)$, for each summary statistic. We note that error model hyperparameter tuning approaches could be considered, e.g. a reasonable heuristic would be to choose an error model that is broad enough that the denoised data $\tilde{\boldsymbol{x}}$ tend not to be outliers in $q(\boldsymbol{x})$, compared to simulations[2]. However, we chose to keep the hyperparameters consistent across tasks, to avoid the risk of overfitting and to demonstrate that neither strong prior knowledge on the error model, nor careful hyperparameter tuning is necessary to yield substantial improvements in performance.

A key advantage of the spike and slab error model is given by the latent variable $\boldsymbol{z}$. Similar to posterior inclusion probabilities in Bayesian regression, the posterior frequency of $z_j = 1$, $\Pr(z_j = 1 \mid \boldsymbol{y})$, can be used as an indicator of the *posterior misspecification probability* for the j-th summary statistic. By comparing to the prior misspecification probability ρ, we can say that if $\Pr(z_j = 1 \mid \boldsymbol{y}) > \rho$, the model provides evidence of misspecification for the j-th summary statistic, and if $\Pr(z_j = 1 \mid \boldsymbol{y}) < \rho$, it provides evidence it is well-specified (Talbott, 2016). For the purpose of generality, the latent variable $\boldsymbol{z}$ was not included when describing RNPE thus far. However, we can jointly infer the posterior $\hat{p}(\boldsymbol{x}, \boldsymbol{z} \mid \boldsymbol{y})$ in step 4 of algorithm 1, by using an MCMC algorithm that supports sampling both continuous and discrete variables. We used mixed Hamiltonian Monte Carlo (HMC) (Zhou, 2020, 2022) from the NumPyro python package (Phan et al., 2019), which is an adaptation of HMC (Neal et al., 2011; Duane et al., 1987), for this purpose.

3 Related Work

3.1 Model Criticism in SBI

Posterior predictive checks have been widely used in SBI, to compare the predictive samples to the observed data (e.g. Durkan et al., 2020; Greenberg et al., 2019; Papamakarios et al., 2019b). Although this is a form of model criticism, we note a key limitation of this approach is that if discrepancy is discovered, it may not be clear whether this is due to the failure of the inference procedure, or due to simulator misspecification. In general, it may be possible to identify the presence of misspecification using anomaly/novelty detection. One such approach was suggested by Schmitt et al. (2021) in the context of SBI. However, this approach does not facilitate interpreting misspecification, and it is not clear how to extend it to perform robust inference.

There are a few related methods that criticise the a posterior estimate by assessing its calibration using simulated data (e.g. Hermans et al., 2021; Prangle et al., 2014; Talts et al., 2018); however, these results are only applicable under the assumption the model is well-specified.

[2]This could for example be assessed by estimating highest density regions of $q(\boldsymbol{x})$, using the density quantile approach from Hyndman (1996)

3.2 Robust Inference in SBI

Approximate Bayesian computation (ABC) is a family of SBI methods, characterised by their use of rejection sampling alongside a kernel function to compare the simulated and observed data (Tavaré et al., 1997; Pritchard et al., 1999; Beaumont et al., 2002). The simplest form of ABC uses simulations with $\boldsymbol{\theta} \sim p(\boldsymbol{\theta})$, and accepts parameters with probability proportional to the similarity of the corresponding simulations to the observed data. The use of a kernel is often seen as a necessary evil, as enforcing an exact match between the observed and simulated data generally results in all samples being rejected. However, it has been noted that the kernel function can often be interpreted as specifying an error model distribution $p(\boldsymbol{y} \mid \boldsymbol{x})$ (Wilkinson, 2013). This idea has been expanded using generalised Bayesian inference, in which the kernel is replaced with a loss function (Schmon et al., 2020), which can be chosen to use a robust measure of discrepancy such as the maximum mean discrepancy (MMD) (Chérief-Abdellatif and Alquier, 2020; Frazier et al., 2020a; Park et al., 2015).

Synthetic likelihood is an SBI method that uses a Gaussian approximation of the likelihood (Wood, 2010). Frazier and Drovandi (2021) investigated a robust synthetic likelihood algorithm, in which the variance components of the Gaussian likelihood are expanded based on additional free parameters, inferred in a nested MCMC sampling scheme. In the context of the paper, they interpret this approach as "likelihood adjustment"; however, we note this could equivalently be reframed as the addition of an independent Gaussian error model, with a prior on the variances[3].

In contrast to more traditional SBI methods, few solutions have been proposed for performing robust inference with neural SBI methods. Perhaps the most common approach is to augment the model with additional nuisance parameters, to expand the range of values the simulator can reasonably produce (Cranmer et al., 2020). This again can often be interpreted as a method of incorporating an error model within the simulator. However, as previously mentioned, the primary limitation of this approach is that it does not easily facilitate *model criticism*.

4 Experiments

On three misspecified tasks, we benchmark the performance of three methods for parameter inference: *i)* RNPE, *ii)* NNPE in which we directly approximate $p(\boldsymbol{\theta} \mid \boldsymbol{y})$ by incorporating the error model into the simulator, and *iii)* NPE, in which we assume no model error is present. To reliably assess performance, we repeat the inference procedure with 1000 different observations and ground truth parameter pairs for each misspecified task, and assess the performance based on two metrics: *i)* the log probability of the true parameter vector, $\boldsymbol{\theta}^*$, and *ii)* the posterior coverage properties (see Section 4.3). We further show examples of RNPE on a single observation for each task to demonstrate how RNPE can be applied in practice for model criticism and robust posterior inference. The code required to reproduce all the results from this manuscript is available at `REDACTED_FOR_ANONYMITY`.

4.1 Setup

For all experiments, we used $N = 50,000$ simulations, with $M = 100,000$ MCMC samples following $20,000$ warm up steps. The MCMC chains were initialised using a random simulation, and $z_j = 1$ for $j = 1, \ldots, D$. To build the approximation $q(\boldsymbol{x})$, we used block neural autoregressive flows (De Cao et al., 2020), as we do not require the ability to sample directly from $q(\boldsymbol{x})$. For the approximation of $q(\boldsymbol{\theta} \mid \boldsymbol{x})$, we utilised neural spline flows (Durkan et al., 2019). For all tasks the hyperparameters were kept consistent; information on hyperparameter choices can be found in Appendix A.3.

4.2 Tasks

Detailed descriptions of each task can be found in Appendix A.2.

Gaussian. A simple Gaussian example from Frazier et al. (2020b). The task involves predicting a single parameter μ, using the sample mean and variance of 100 i.i.d. samples drawn from $N(\mu, 1)$, where 1 is the variance of the Gaussian distribution. To artificially produce misspecified observations,

[3] This results due the fact that the addition of two independent Gaussian variables yields another Gaussian random variable, with covariance equal to the sum of the individual covariance matrices.

the 100 i.i.d. are instead drawn from $N(\mu, 2)$. Despite its simplicity, this task usefully demonstrates that issues with misspecification can arise even in the simplest of models.

SIR. A stochastic model of epidemic spread with a time-varying infection rate. We attempt to infer two parameters, the infection rate β and the recovery rate γ. We draw daily infection counts for a period of 365 days and compute the following summary statistics: 1-3) `Mean`, `Median` and `Max` - the mean, median and maximum number of infections; 4) `Max Day` - the day of occurrence of the maximum number of infections, 5) `Half Day` - the day at which half the total number of infections was reached, and 6) `Autocor` - the mean autocorrelation (lag 1) of infections. To artificially produce misspecified observations, we chose to introduce a small degree of reporting delays, whereby weekend infection counts are reduced by 5%, which are subsequently recouped on the following Monday. Fig. 5 in the appendix shows an example misspecified simulation and a corresponding artificially produced observation.

CS. A marked point process model of cancer and stromal (CS) cell development in 2D-space, in which the locality of a cell relative to unobserved parent points determines whether it is a cancer or a stromal cell. Similar point process models can be found in Jones-Todd et al. (2019). There are three Poisson rate parameters, corresponding to the cell rate λ_c, the parent rate λ_p and the daughter rate λ_d, where λ_d controls the number of nearest cells to each parent which become cancerous. We used the following summary statistics: 1-2) `N Cancer` and `N Stromal` - the number of cancer and stromal cells; 3-4) `Mean Min Dist` and `Max Min Dist` - the mean and maximum distance from stromal cells to their nearest cancer cell. Summary statistics 3-4 were approximated empirically through sampling 50 stromal cells, as the full distance matrix was too expensive to compute. To artificially produce misspecified observations, cancer cells were removed if they fell too close to the cancer parent, mimicking necrosis that often occurs in core regions of tumours. A plot of an example misspecified simulation and a corresponding artificially produced observation can be found in Fig. 6 in the appendix.

4.3 Performance Metrics

Log probability of $\boldsymbol{\theta}^*$. The mean posterior log probability of the true parameters over multiple observations is an extensively used performance metric (e.g. Papamakarios and Murray, 2016; Hermans et al., 2020; Durkan et al., 2020), and is closely related to the average KL divergence between the true and approximate posteriors (Lueckmann et al., 2021). We found that NPE would occasionally fail catastrophically, placing negligible posterior mass on the true parameters, skewing the mean estimates. Due to this, we instead use box plots to visualise the distribution over the log-probabilities of the true parameters, rather than reporting the mean.

Posterior Coverage. Given a confidence level α and a posterior approximation $\hat{p}(\boldsymbol{\theta} \mid \boldsymbol{y})$, let $\mathrm{HDR}_{\hat{p}(\boldsymbol{\theta}|\boldsymbol{y})}(1-\alpha)$ represent its $1-\alpha$ highest posterior density region, i.e. the smallest region that contains at least $100(1-\alpha)\%$ of the mass of $\hat{p}(\boldsymbol{\theta} \mid \boldsymbol{y})$. The expected posterior coverage is the frequency with which the true parameter value falls within this highest density region

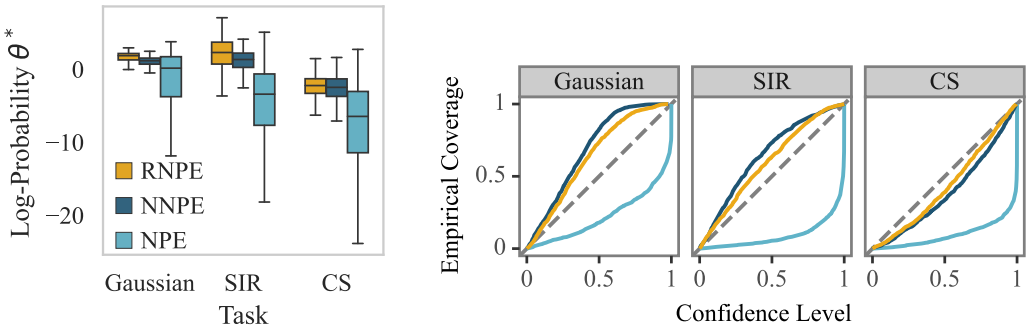
$$\mathbb{E}_{\boldsymbol{\theta}^*, \boldsymbol{y} \sim p(\boldsymbol{\theta}, \boldsymbol{y})} \left[\mathbb{1}\{\boldsymbol{\theta}^* \in \mathrm{HDR}_{\hat{p}(\boldsymbol{\theta}|\boldsymbol{y})}(1-\alpha)\} \right] \tag{10}$$

where $\mathbb{1}$ is the indicator function. Posterior coverage is useful for assessing whether an inference procedure is likely to yield overconfident or conservative posterior estimates (Hermans et al., 2021). We empirically estimate this expectation, approximating the highest density regions using the density quantile approach outlined in Hyndman (1996), with 10,000 samples from the approximate posterior.

4.4 Results and Discussion

Overall Performance.

The results shown in Fig. 2 show that both RNPE and NNPE perform better than NPE, both in terms of tending to place more mass on $\boldsymbol{\theta}^*$ (Fig. 2a), and producing posteriors with more conservative coverage properties (Fig. 2b). It is important to note that having conservative posterior estimates is generally considered preferable to having overconfident posterior estimates, as the latter could lead to drawing of erroneous scientific conclusions. These results highlight the importance of accounting for model misspecification when performing parameter inference, even if doing so introduces a misspecified error model. RNPE slightly outperforms NNPE in terms of the mass placed on $\boldsymbol{\theta}^*$. We hypothesise this increased performance is due to RNPE exploiting the fact that the error model is known and tractable, rather than rather than naïvely relying on NPE to account for it.



(a) The log-probability of the true parameters $\boldsymbol{\theta}^*$ in the approximate posteriors.

(b) The coverage of each approach at different confidence levels. A well-calibrated posterior follows the dotted line, with conservative posteriors lying above the dotted line, and overconfident posteriors falling below the dotted line.

Figure 2: Comparative performance of RNPE, NNPE and NPE on the three misspecified tasks, using 1000 different observation and true parameter pairs.

Gaussian. An example for the Gaussian task can be seen in Fig. 1, in which x_1 corresponds to the mean statistic, and x_2 the variance. From observing the posterior misspecification probabilities and the denoised samples, the results clearly identify a discrepancy in which the observed variance statistic is higher than the simulated variance statistics, allowing identification of source of misspecification. As this is a simple example, we can analytically access the true posterior (obtained using the true data generating model). RNPE produces a posterior close to the ground-truth, whereas NPE confidently places most of the posterior mass far from the ground-truth posterior.

SIR. In the SIR model, the observed data were artificially corrupted by mimicking reporting delays over the weekend. With this in mind, we can consider how the *model criticism* aspect of RNPE could facilitate identification of the issue. Fig. 3a shows that most summary statistics are more likely to be well-specified than was believed *a priori*, with the clear exception of the `Autocor` summary statistic, which is inferred to be misspecified. Fig. 3b shows a density plot of the denoised data for the `Autocor` summary statistic, showing that the observed autocorrelation is much lower relative to what would be expected under the model. This may prompt the practitioner to investigate the day-to-day changes in infections, in which case the discontinuities introduced by the reporting delays could be identified. Fig. 3c shows the posteriors for RNPE and NPE. RNPE yields a posterior with far lower variance than NPE, demonstrating an example where it does not achieve robustness simply by producing broader posterior estimates.

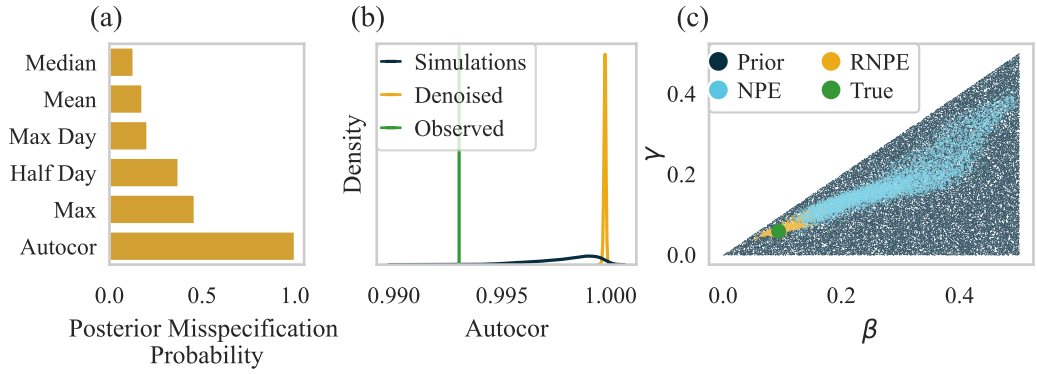


Figure 3: The results for an example of the SIR task: a) the inferred posterior misspecification probabilities for each summary statistic; b) density plot for the denoised samples of the autocorrelation summary statistic; c) The posteriors for both NPE and RNPE.

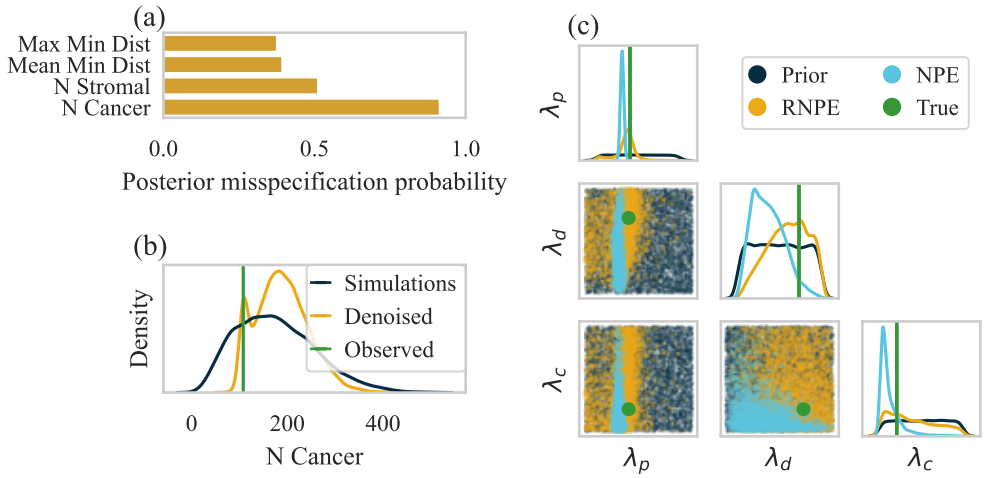


Figure 4: The results for an example of the CS task: a) the inferred posterior misspecification probabilities for each summary statistic; b) density plot for the denoised samples of the `N Cancer` summary statistic; c) a pair plot of the posteriors for both NPE and RNPE.

CS. In the CS model, the observed data was corrupted by removing cells in the core regions of tumours, mimicking necrosis (see Fig. 6). This misspecification primarily impacts the `N Cancer` summary statistic (the number of cancer cells). From Figs. 4a,4b, we can see that the results provide evidence that the `N Cancer` statistic is misspecified, with the observed data having fewer cancer cells than would be expected if it was generated from the simulator. Once again, alongside domain knowledge, this information points the practitioner to the source of misspecification (necrosis), and hence the model could be adjusted appropriately. For this example, NPE produced an overconfident posterior with little mass on the true parameters, whereas RNPE does not exhibit this issue (Fig. 4c).

5 Conclusions

We described RNPE, an approach that can be applied to any black-box simulator to simultaneously perform *model criticism*, and perform *robust inference*. RNPE is philosophically similar to existing robust SBI approaches for ABC and synthetic likelihood (Wilkinson, 2013; Frazier and Drovandi, 2021). However, this work introduces two novel contributions: *i)* we propose the first strategy for performing robust inference with a neural SBI method, which have advantages in simulation efficiency, flexibility and scaling to high dimensional problems (Lueckmann et al., 2021), and *ii)* we use a spike-and-slab error model to get explicit misspecification probability diagnostics for summary statistics, to facilitate *model criticism*. However, our work is not without limitations. Firstly, RNPE is more computationally costly than NPE, requiring an additional MCMC step and fitting of $q(\boldsymbol{x})$. Secondly, we only evaluated performance of a single choice of error model on a limited number of tasks. It would be beneficial in the future to have a larger set of realistic misspecified tasks for assessing performance, similar to the work by Lueckmann et al. (2021) for the well-specified case. Finally, we restricted focus to NPE - in future work it would be interesting to develop robust extensions for other SBI algorithms and to compare results with other existing methods.

We demonstrated with three examples, that RNPE consistently outperforms NPE for parameter inference under misspecification, tending to put more posterior mass on the true parameters, whilst also exhibiting better coverage properties. The tasks were designed to mimic levels of misspecification that could be realistically encountered, and hence the poor performance of NPE calls into question its use in practice, particularly when the level of misspecification could be significant. The substantial difference in performance highlights, as suggested by Box (1980), that we should not have blind faith that a model is sufficiently accurate, and therefore we must devise methods for performing *model criticism* and *robust inference*. With further research and wider adoption of these principles in SBI, we believe simulators will only become more powerful tools for scientific discovery.

References

Atkeson, A., Kopecky, K., and Zha, T. (2020). Estimating and forecasting disease scenarios for COVID-19 with an SIR model. Technical report, National Bureau of Economic Research.

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

Bingham, E., Chen, J. P., Jankowiak, M., Obermeyer, F., Pradhan, N., Karaletsos, T., Singh, R., Szerlip, P., Horsfall, P., and Goodman, N. D. (2019). Pyro: Deep universal probabilistic programming. *The Journal of Machine Learning Research*, 20(1):973–978.

Box, G. E. (1980). Sampling and Bayes' inference in scientific modelling and robustness. *Journal of the Royal Statistical Society: Series A (General)*, 143(4):383–404.

Bradbury, J., Frostig, R., Hawkins, P., Johnson, M. J., Leary, C., Maclaurin, D., Necula, G., Paszke, A., VanderPlas, J., Wanderman-Milne, S., and Zhang, Q. (2018). JAX: composable transformations of Python+NumPy programs.

Brehmer, J., Kling, F., Espejo, I., and Cranmer, K. (2020). Madminer: Machine learning-based inference for particle physics. *Computing and Software for Big Science*, 4(1):1–25.

Chan, J., Perrone, V., Spence, J., Jenkins, P., Mathieson, S., and Song, Y. (2018). A likelihood-free inference framework for population genetic data using exchangeable neural networks. *Advances in neural information processing systems*, 31.

Chen, Y., Zhang, D., Gutmann, M., Courville, A., and Zhu, Z. (2020). Neural approximate sufficient statistics for implicit models. *arXiv preprint arXiv:2010.10079*.

Chérief-Abdellatif, B.-E. and Alquier, P. (2020). MMD-Bayes: Robust Bayesian estimation via maximum mean discrepancy. In *Symposium on Advances in Approximate Bayesian Inference*, pages 1–21. PMLR.

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

De Cao, N., Aziz, W., and Titov, I. (2020). Block neural autoregressive flow. In *Uncertainty in artificial intelligence*, pages 1263–1273. PMLR.

Duane, S., Kennedy, A. D., Pendleton, B. J., and Roweth, D. (1987). Hybrid Monte Carlo. *Physics letters B*, 195(2):216–222.

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

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

Fearnhead, P. and 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.

Ferguson, N. M., Laydon, D., Nedjati-Gilani, G., Imai, N., Ainslie, K., Baguelin, M., Bhatia, S., Boonyasiri, A., Cucunubá, Z., Cuomo-Dannenburg, G., et al. (2020). Impact of non-pharmaceutical interventions (NPIs) to reduce covid-19 mortality and healthcare demand.

Frazier, D. T. and Drovandi, C. (2021). Robust approximate Bayesian inference with synthetic likelihood. *Journal of Computational and Graphical Statistics*, 30(4):958–976.

Frazier, D. T., Drovandi, C., and Loaiza-Maya, R. (2020a). Robust approximate Bayesian computation: An adjustment approach. *arXiv preprint arXiv:2008.04099*.

Frazier, D. T., Robert, C. P., and Rousseau, J. (2020b). Model misspecification in approximate Bayesian computation: consequences and diagnostics. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 82(2):421–444.

Gelman, A. and Shalizi, C. R. (2013). Philosophy and the practice of Bayesian statistics. *British Journal of Mathematical and Statistical Psychology*, 66(1):8–38.

George, E. I. and McCulloch, R. E. (1993). Variable selection via Gibbs sampling. *Journal of the American Statistical Association*, 88(423):881–889.

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

Grünwald, P. and Langford, J. (2007). Suboptimal behavior of Bayes and MDL in classification under misspecification. *Machine Learning*, 66(2):119–149.

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

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

Hermans, J., Delaunoy, A., Rozet, F., Wehenkel, A., and Louppe, G. (2021). Averting a crisis in simulation-based inference. *arXiv preprint arXiv:2110.06581*.

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

Jones-Todd, C. M., Caie, P., Illian, J. B., Stevenson, B. C., Savage, A., Harrison, D. J., and Bown, J. L. (2019). Identifying prognostic structural features in tissue sections of colon cancer patients using point pattern analysis. *Statistics in medicine*, 38(8):1421–1441.

Kidger, P. and Garcia, C. (2021). Equinox: neural networks in JAX via callable pytrees and filtered transformations. *arXiv preprint arXiv:2111.00254*.

Kleijn, B. J. and van der Vaart, A. W. (2012). The Bernstein-von-Mises theorem under misspecification. *Electronic Journal of Statistics*, 6:354–381.

Lam, S. K., Pitrou, A., and Seibert, S. (2015). Numba: A LLVM-based python JIT compiler. In *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC*, pages 1–6.

Lueckmann, J.-M., Boelts, J., Greenberg, D., Goncalves, P., and Macke, J. (2021). Benchmarking simulation-based inference. In *International Conference on Artificial Intelligence and Statistics*, pages 343–351. PMLR.

Lueckmann, J.-M., Goncalves, P. J., Bassetto, G., Öcal, K., Nonnenmacher, M., and Macke, J. H. (2017). Flexible statistical inference for mechanistic models of neural dynamics. *Advances in neural information processing systems*, 30.

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

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

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

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

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

Park, M., Jitkrittum, W., and Sejdinovic, D. (2015). K2-ABC: approximate Bayesian computation with kernel embeddings. *arXiv preprint arXiv:1502.02558*.

Perla, J., Sargent, T. J., Stachurski, J., and Rackauckas, C. (2022). Quantitative economics with julia: Modeling shocks in covid 19 with stochastic differential equations.

Phan, D., Pradhan, N., and Jankowiak, M. (2019). Composable effects for flexible and accelerated probabilistic programming in NumPyro. *arXiv preprint arXiv:1912.11554*.

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

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

Rackauckas, C. and Nie, Q. (2017). Differentialequations. jl–a performant and feature-rich ecosystem for solving differential equations in julia. *Journal of Open Research Software*, 5(1).

Randall, D. A., Wood, R. A., Bony, S., Colman, R., Fichefet, T., Fyfe, J., Kattsov, V., Pitman, A., Shukla, J., Srinivasan, J., et al. (2007). Climate models and their evaluation. In *Climate change 2007: The physical science basis. Contribution of Working Group I to the Fourth Assessment Report of the IPCC (FAR)*, pages 589–662. Cambridge University Press.

Schmitt, M., Bürkner, P.-C., Köthe, U., and Radev, S. T. (2021). Bayesflow can reliably detect model misspecification and posterior errors in amortized Bayesian inference. *arXiv preprint arXiv:2112.08866*.

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

Syring, N. and Martin, R. (2019). Calibrating general posterior credible regions. *Biometrika*, 106(2):479–486.

Talbott, W. (2016). Bayesian Epistemology. In Zalta, E. N., editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, Winter 2016 edition.

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

Tavaré, S., Balding, D. J., Griffiths, R. C., and Donnelly, P. (1997). Inferring coalescence times from DNA sequence data. *Genetics*, 145(2):505–518.

Thomas, O., Dutta, R., Corander, J., Kaski, S., and Gutmann, M. U. (2022). Likelihood-free inference by ratio estimation. *Bayesian Analysis*, 17(1):1–31.

Wilkinson, R. D. (2013). Approximate Bayesian computation (ABC) gives exact results under the assumption of model error. *Statistical applications in genetics and molecular biology*, 12(2):129–141.

Wood, S. N. (2010). Statistical inference for noisy nonlinear ecological dynamic systems. *Nature*, 466(7310):1102–1104.

Zhou, G. (2020). Mixed Hamiltonian Monte Carlo for mixed discrete and continuous variables. *Advances in Neural Information Processing Systems*, 33:17094–17104.

Zhou, G. (2022). Metropolis augmented hamiltonian monte carlo. *arXiv preprint arXiv:2201.08044*.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Section 5.
   (c) Did you discuss any potential negative societal impacts of your work? [No] We see robust inference approaches allowing for better informed decision making, which should generally have positive societal consequences.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] We used a box plot (Fig. 2a) to visualise the distribution of results over seeds.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] NumPyro Python package (Section 2.2).
   (b) Did you mention the license of the assets? [No]
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]