Available online at www.eccomasproceedia.org
Eccomas Proceedia UNCECOMP (2025) 301-317

ECCOMAS
Proceedia

UNCECOMP 2025
6th ECCOMAS Thematic Conference on
Uncertainty Quantification in Computational Science and Engineering
M. Papadrakakis, V. Papadopoulos, G. Stefanou (eds.)
Rhodes Island, Greece, 15-18 June 2025

BAYESIAN INVERSE UNCERTAINTY QUANTIFICATION FOR COMPUTATIONAL ENGINEERING USING SIMULATION-BASED INFERENCE

Maternus Herold[1,2,3], **Jan Boelts**[1]

[1] TransferLab, appliedAI Institute for Europe

[2] BMW Research and Development Center Munich, BMW Group

[3] Institute for Applied Mathematics and Scientifi Computing, University of the Bundeswehr Munich

Abstract. *Inverse Uncertainty Quantification (IUQ) is crucial for analyzing complex systems in applied mechanics. Bayesian inference is a common approach to IUQ, as it allows for the systematic quantification of parameter uncertainty given prior knowledge and observations. However, many simulators and experiments in applied mechanics are analytically or computationally intractable, hindering the application of traditional Bayesian inference. To overcome this, we propose using Simulation-Based Inference (SBI), which circumvents the need for explicit likelihood calculations exploiting recent advances in probabilistic machine learning. We demonstrate the application of SBI for IUQ using the open-source* `sbi` *software package, showcasing its capabilities in an automotive passive safety example. We highlight the practical SBI workflow, including prior predictive checks, posterior inference, and posterior evaluation, and discuss key considerations for its application in computational mechanics.*

Keywords: Bayesian Inference, Inverse Uncertainty Quantification, Simulation-Based Inference, Neural Density Estimation, Automotive Passive Safety

ISSN:2623-3339
Peer-review under responsibility of the organizing committee of UNCECOMP 2025.
doi: 10.7712/120225.12358.21274

1 INTRODUCTION

The target of robust and reliable system design in applied mechanics is hindered by the inherent presence of uncertainties across various facets, including material properties, geometric tolerances, and system parametrizations [26, 28, 37]. Underestimating these uncertainties can have sever consequences, particularly in safety-critical applications such as aerospace, civil engineering, and automotive passive safety. IUQ emerges as a critical methodology to address this challenge by inferring the distribution of uncertain system inputs from observed data. The ability of Bayesian inference to incorporate prior knowledge and systematically quantify uncertainty makes it a natural fi for IUQ, especially in mechanics, where prior knowledge of material properties and physical constraints is commonly available [25, 49, 50]. However, in applied mechanics models, computationally expensive simulations and the lack of an analytical likelihood often make classical approximate Bayesian methods—such as Markov Chain Monte Carlo or variational inference—impractical. To address this challenge, we propose using Simulation-Based Inference (SBI, [8]), which enables Bayesian inference without requiring an explicit likelihood. SBI trains neural networks as conditional density estimators [39] to approximate the posterior distribution directly from simulated data. This approach overcomes the limitations of likelihood-based methods in IUQ for applied mechanics, allowing systematic uncertainty estimation in model parameters.

SBI is an active research fiel that combines established methods with recent advances in probabilistic machine learning [8, 39, 12, 34]. However, practitioners often face challenges in applying these techniques, as research implementations are typically difficul to use and lack standardization [5]. To address this, several open source initiatives provide accessible and well-documented implementations of SBI methods, enabling broader adoption and reproducibility [47, 2, 41]. Such software packages play a crucial role in streamlining complex workfl ws, accelerating research cycles, and democratizing access to advanced inference techniques. By fostering collaboration and standardization, they ultimately contribute to more robust and insightful scientifi outcomes.

In the following sections, we firs discuss the application of Bayesian inference to IUQ in Section 2, highlighting the challenges posed by complex models and intractable likelihoods in applied mechanics, and thereby motivating the use of SBI. In Section 3, we introduce the concept of SBI, detailing its ability to overcome these challenges, and emphasize the need for versatile software packages that support the complete SBI workfl w. Finally, in Section 4, we demonstrate the practical application of SBI for IUQ in an automotive passive safety example, showcasing the functionality and ease of use of the `sbi` toolkit [47, 2]. This numerical example illustrates the full SBI workfl w, from prior predictive checks to posterior evaluation, using the `sbi` software package[1].

2 INVERSE UNCERTAINTY QUANTIFICATION VIA BAYESIAN INFERENCE

Relations between parameters and observations in mechanical systems are generally studied through the evaluation of a forward model $\mathcal{M} : \Theta \to \mathcal{X}$, where $\Theta \subseteq \mathbb{R}^d$ represents the parameter space and $\mathcal{X} \subseteq \mathbb{R}^m$ denotes the range of observable outputs. Depending on the fiel of application, the model $\mathcal{M}$ can be both, a simulation or a physical experiment. We denote the relationship between the parameters θ and observations $\mathbf{x}$ by the following equation:

[1]The `sbi` toolkit is an open-source Python package for the complete SBI workfl w, described in Section 3. It is accessible via Python's package index and GitHub directly.

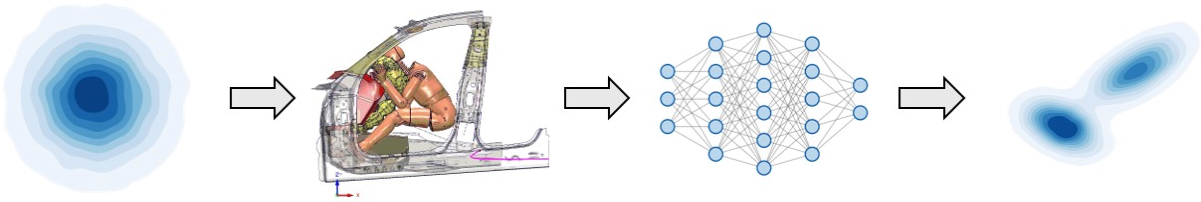


Figure 1: The general workfl w of SBI involves sampling parameter values θ from the prior $\theta \sim p(\theta)$ and evaluating the stochastic model $\mathcal{M}$ on them, generating observations $\mathbf{x}$. Based on the generated data, a neural density estimator f_ϕ (see Section 3.2) is fitted yielding a parametric posterior approximation $p_\phi(\theta \mid \mathbf{x}) \approx p(\theta \mid \mathbf{x})$.

$$\mathbf{x} = \mathcal{M}(\theta), \quad \theta \in \Theta, \mathbf{x} \in \mathcal{X}. \tag{1}$$

In many systems, noise and variability introduce uncertainty in both the parameters θ and the observations $\mathbf{x}$. This uncertainty can be described through a conditional distribution $p(\mathbf{x} \mid \theta)$, which captures how the system generates data given a set of parameters. In practice, however, the goal is often to estimate the parameters θ given observed data $\mathbf{x}$, while accounting for both system uncertainty and uncertainty of the parameter estimates. This is achieved by considering the inverse problem, i.e., obtaining the conditional distribution $p(\theta \mid \mathbf{x})$. This distribution, known as the *posterior*, is obtained using Bayes' theorem, which is given by

$$\begin{aligned} p(\theta \mid \mathbf{x}) &= \frac{p(\mathbf{x} \mid \theta)p(\theta)}{p(\mathbf{x})} \\ &\propto p(\mathbf{x} \mid \theta)p(\theta). \end{aligned} \tag{2}$$

In Equation (2), $p(\mathbf{x} \mid \theta)$ is the *likelihood*, $p(\theta)$ is the *prior* distribution on the parameters θ, and $p(\mathbf{x})$ is the marginal likelihood or evidence across the entire parameter space

$$p(\mathbf{x}) = \int_\Theta p(\mathbf{x} \mid \theta)p(\theta)\, d\theta. \tag{3}$$

The likelihood describes the probability of observing $\mathbf{x}$ given the parameters θ for the model $\mathcal{M}$. The prior distribution represents the initial belief about the parameters before observing the data. The marginal likelihood yields a constant normalization factor, which is generally a high-dimensional integral due to the dimensionality d of the parameter space Θ, making it intractable to compute exactly. Therefore, it is usually omitted and the posterior is obtained up to proportionality, as denoted in Equation (2), e.g., by obtaining posterior samples using established sampling algorithms like Markov Chain Monte Carlo (MCMC).

The approach to IUQ via the Bayesian inference is particularly suitable for applications in applied mechanics, as it provides a natural framework to incorporate expert knowledge and constraints on the parameters through the prior distribution. The likelihood (define through the model) is then used to update the *prior beliefs* about the parameters by observing data under the assumed model, resulting in the posterior distribution. The posterior distribution provides a probabilistic description of *all* possible parameter values for yielding the observation conditioned on. Crucially, access to the full posterior covariance not only provides systematic uncertainty estimates for each parameter but also unveils their interdependencies. This allows engineers to understand which parameters exert the greatest influenc on system behavior and

how they interact. This knowledge is particularly vital in applications such as vehicle safety, where the system's parameterization is constrained by its integration within a larger, more complex system.

Example 2.1 (Vehicle Restraint Systems)**.** *Consider the design of a vehicle restraint system $\mathcal{M}$, with the goal to minimize the risk of injury $\mathbf{x}$ of the occupants in the event of a collision (see Figure 2). The parameters of interest θ in this system include the deployment times of different airbags and the seatbelt characteristics. The posterior distribution $p(\theta \mid \mathbf{x})$ provides a probabilistic description of the uncertainty within the parameters, given the observed occupant safety $\mathbf{x}_o$.*

Additionally, different parameterizations $\theta_1, \theta_2, \ldots, \theta_n \in \Theta$ can lead to similar levels of occupant safety $\mathbf{x}_1 \approx \mathbf{x}_2 \approx \cdots \approx \mathbf{x}_n$, $\mathcal{M}(\theta_i) = \mathbf{x}_i \in \mathcal{X}$. The posterior distribution should ideally cover all of such solutions. In a downstream task, the different parameterizations might be assessed based on their cost or robustness [25].

While Bayesian inference provides a principled approach to IUQ, its direct application in applied mechanics is challenging. The likelihood function $p(\mathbf{x} \mid \theta)$ of a system $\mathcal{M}$ under consideration is usually not analytically available. This is due to such systems being complex mechanical models or simulations that can only be evaluated as black-box functions. Classical access to posterior samples via MCMC methods is therefore technically infeasible in the case of simulations and economically infeasible for many hardware tests [26, 28, 45]. Moreover, downstream tasks, such as those in Example 2.1, require evaluating and sampling the posterior for many observations $\mathbf{x}$. To enable this efficientl , a full parametric approximation—ideally in an amortized form—is desired, allowing rapid evaluation and sampling for any given observation.

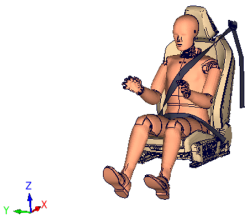


Figure 2: Illustration of a vehicle restraint system simulation depicting a crash test dummy protected by a seatbelt in a full frontal crash. Further restraint systems such as the airbags and the car's chassis are removed from the figur to reduce occlusion. Figure adapted from [26].

SBI offers a solution to these challenges. It approximates the posterior using data simulated from parameters sampled from the prior, without requiring an explicit likelihood function. Additionally, by performing conditional density estimation with neural networks, it enables amortized inference and efficien evaluation and sampling across many observations [8, 40, 18]. Unlike traditional MCMC methods, which have convergence guarantees, SBI is approximate in nature and requires careful evaluation. In the following section, we will explain the details of this approach and how it addresses the challenges in applied mechanics.

3 SIMULATION-BASED-INFERENCE

Simulation-based inference (SBI) enables Bayes ian parameter estimation in scenarios where the likelihood function is intractable and has been used in various fields such as astrophysics [9, 10], neuroscience [17, 20, 4, 3], or mechanical engineering [26, 28]. Unlike traditional methods that require an explicit likelihood function, SBI leverages methods from probabilistic machine learning to perform Bayesian inference for simulation-based models [8].

Early SBI approaches such as Approximate Bayesian Computation (ABC, [44]) address this challenge by simulating data and accepting parameters that yield close matches to observations.

However, ABC relies on comparing simulated data to observations using distance metrics and accepting parameters only if they fall below a user-define threshold, leading to inefficien y, especially in high-dimensional problems. More recent neural network-based SBI methods improve upon these methods by using neural network-based density estimation techniques to effi ciently approximate the posterior or the likelihood [39, 18, 10], making it particularly suitable for cases with high-dimensional dependencies and complex posterior shapes, such as multimodal distributions [8, 34].

The following section outlines the general workfl w of SBI, starting with the most widely used method, Neural Posterior Estimation (NPE), which directly approximates the posterior distribution. We then compare it with Neural Likelihood Estimation (NLE), which learns the likelihood function, and Neural Ratio Estimation (NRE), which estimates the likelihood ratio, highlighting their trade-offs. In addition, we discuss performance evaluation, potential failure modes, such as poor posterior coverage, and the importance of robust software implementations to ensure reliability in practice.

The application of Bayesian inference, as outlined in the established Bayesian workfl w by Gelman et al. [16], typically involves a sequence of steps: model specification prior defini tion, posterior computation, and model checking. This general framework is also applicable to simulation-based inference (SBI) [5] and it aligns closely with the common steps employed in inverse uncertainty quantificatio (IUQ). Specificall , for IUQ via Bayesian inference using SBI, one firs conducts *a priori checks* to verify the suitability of the problem and data. Subsequently, a specifi SBI method is selected and applied. Finally, *a posteriori checks* and performance evaluations are performed to assess the quality and validity of the inference results.

3.1 A Priori Checks

In order to obtain valid inference results on observed data, the setup of the prior and simulator have to be assessed. While we generally assume a slight deviation of the model from the true data-generating process, the estimation should be able to reproduce the most important aspects of the data. This is known as the *model misspecification* problem. For example, in a finit element model, neglecting certain material nonlinearities could lead to model misspecification While several methods to automatically detect and account for model misspecificatio in SBI have been proposed [14, 48, 29, 43], they often require additional computational resources. We therefore focus on the *prior predictive check*, which is a well interpretable and easy to implement alternative to assess the misspecification

Prior predictive checks are conducted to assess whether the chosen prior distributions for a model's parameters are reasonable and result in plausible data w.r.t. obtained observations $\mathbf{x}_o$. The process involves definin prior distributions $p'(\theta)$ for each model parameter $\theta \in \Theta$, incorporating any available domain knowledge and system constraints. New data is then simulated by sampling parameter values from these prior distributions and then generating datasets using the model with the sampled parameters, see Algorithm 1. This simulation creates the prior predictive distribution

$$p'(\mathbf{x}) = \int p(\mathbf{x} \mid \theta) p'(\theta) d\theta \approx \int \mathcal{M}(\theta) p'(\theta) d\theta. \tag{4}$$

The data sampled from the prior predictive distribution $\mathbf{x}' \sim p'(\mathbf{x})$ is compared to the actual observed data $\mathbf{x}_o$. If the distribution of the simulated data is realistic and contains the observed data, the priors are considered appropriate. If the simulated data is unrealistic or fails to include the observed data, the priors must be adjusted. Depending on the dimensionality of the data,

visual checks, summary statistics like mean and variance, or statistical tests may be required. Even if the simulated and observed data do not match perfectly, the priors can still be acceptable if the simulated data is generally plausible. Note that a discrepancy can also suggest an issue with the simulation model itself, in which case more advanced techniques for identifying the misspecificatio need to be employed [43, 27]. Passing the prior predictive check allows one to proceed with the training of the neural density estimator.

3.2 Posterior Approximation

At the core of neural SBI stands the idea of using access to evaluate the forward simulation model (Equation (1), $\mathcal{M}$ ($\mathcal{M}(\theta) = \mathbf{x}$) to generate synthetic data $\mathbf{x}$ and bypass the requirement of analytical likelihood for traditional Bayesian inference [42]. The three most popular approaches are NPE [39], approximating the posterior distribution $p(\theta \mid \mathbf{x})$ directly, NLE [38], approximating the likelihood $p(\mathbf{x} \mid \theta)$, and NRE [22], learning the likelihood-to-evidence ratio $r(\theta, \mathbf{x}) = \frac{p(\mathbf{x}|\theta)}{p(\mathbf{x})}$. For all three approaches, a dataset from the joint distribution $\mathcal{D} = \{(\theta_i, \mathbf{x}_i)\}_{i=1}^N$, where $(\theta_i, \mathbf{x}_i) \sim p(\theta, \mathbf{x})$, is required to train the neural density estimator f_ϕ. In order to obtain the tuples $(\theta_i, \mathbf{x}_i)$, the forward model $\mathcal{M}$ is evaluated on a set of parameter values $\theta_i \sim p(\theta)$, as outlined in Algorithm 1.

Algorithm 1: Sampling from the joint distribution.

Input: Prior distribution $p(\theta)$, simulation model $\mathcal{M} : \theta \mapsto x$, number of samples N
Output: Sampled dataset from the joint distribution
$\mathcal{D} = \{(\theta_i, \mathbf{x}_i) \mid (\theta_i, \mathbf{x}_i) \sim p(\theta, \mathbf{x}), i = 1, \ldots, N\}$
Initialize an empty list of samples $\mathcal{D} \leftarrow [\,]$;
for $i \leftarrow 0$ **to** N **do**
  Sample parameter $\theta_i \sim p(\theta)$;
  Generate simulated data $x_i \leftarrow \mathcal{M}(\theta_i)$;
  Append (θ_i, x_i) to the list of samples $\mathcal{D}$;
end

In a subsequent step, a neural density estimator f_ϕ is trained on the dataset $\mathcal{D}$ to yield the approximate conditional distribution p_ϕ, in the case of NPE and NLE, or train a classifie , in the case of NRE. In the following, we will focus on NPE, as it is the most popular and widely used method by the community. The approaches of NLE and NRE run analogously with either a different targeted distribution or training task.

The fi of the neural density estimator is achieved by minimizing the forward Kullback-Leibler (KL) divergence between the true and approximate posterior distribution

$$\mathrm{KL}\left(p(\theta \mid \mathbf{x}) \,||\, p_\phi(\theta \mid \mathbf{x})\right) = \int p(\theta \mid \mathbf{x}) \log \left(\frac{p(\theta \mid \mathbf{x})}{p_\phi(\theta \mid \mathbf{x})} \right) d\theta. \tag{5}$$

The KL-divergence cannot be optimized directly as we generally don't have access to the true posterior $p(\theta \mid \mathbf{x})$. In order to minimize the KL-divergence implicitly, the negative log likelihood (NLL) is used as a surrogate loss function [1, 38]

$$\mathcal{L}_{\mathrm{NLL}}(\phi) = -\int p(\theta, \mathbf{x}) \log p_\phi(\theta \mid \mathbf{x}) d\theta \approx -\frac{1}{N} \sum_{i=1}^{N} \log f_\phi(\theta_i \mid \mathbf{x}_i). \tag{6}$$

As the approximate posterior $p_\phi(\theta \mid \mathbf{x})$ is encoded by a neural network, the conditional distribution for arbitrary observations $\mathbf{x}'$ can be obtained by evaluating the neural network on the new data point without retraining. This is commonly referred to as *amortized* inference as the initial high cost of training the density estimator is amortized later on [8, 39]. The steps taken to obtain a posterior approximation via NPE are summarized in Figure 1.

3.3 Quality Assessment of the Posterior Approximation

Evaluating the quality of a posterior approximation is crucial for ensuring reliable inference and security in downstream tasks. We will discuss a selection of these, emphasizing their relevance to applied mechanics.

To assess the discrepancy between the approximated and true posteriors, *sample-based tests* are commonly employed. Methods such as the *classifier two-sample test* (C2ST) [15, 32, 21], Maximum Mean Discrepancy (MMD, [19]), and Kernelized Stein Discrepancy [31] compare samples from the approximated posterior with (samples from the) true posterior. However, these methods require access to the true posterior, which is typically unknown in real-world applications. Consequently, their primary utility lies in evaluating toy models or benchmarking scenarios, as seen in works like Lueckmann et al. [34].

In contrast to two-sample tests, *Posterior Predictive Checks* (PPC) offer a practical means of evaluating the inference approximation that can performed without access to a reference posterior. PPC compares observed data to data generated from the posterior predictive distribution:

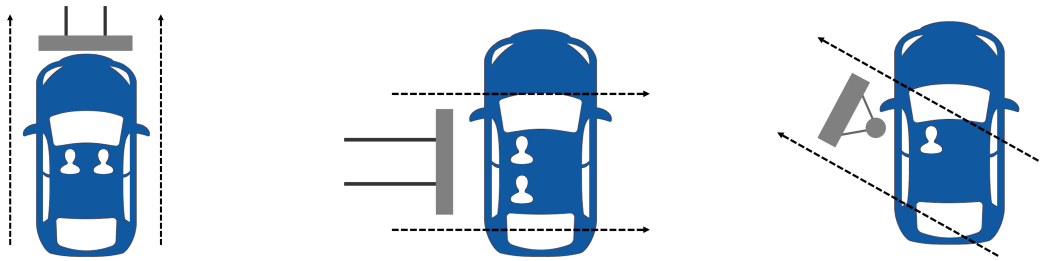
$$p(\mathbf{x}' \mid \mathbf{X}_o) = \int p(\mathbf{x}' \mid \theta) p(\theta \mid \mathbf{X}_o) d\theta \approx \int \mathcal{M}(\theta) p(\theta \mid \mathbf{X}_o) d\theta, \tag{7}$$

Samples from the posterior predictive distribution are obtained by sampling parameters from the approximated posterior and simulating (predictive) data $\mathbf{x}'$ with those parameters. The check is successful if the observed data $\mathbf{X}_o$ falls within the predictive distribution. Typically, this is checked by visual inspection or by calculating distance metrics between predictive and observed data. If the noise of the simulation model is known, the variance of the samples around the observation should match it. In computational mechanics, PPC helps assess whether the model accurately reproduces observed phenomena, such as deformation patterns or stress distributions.

Although PPC and C2ST are valuable tools for assessing the accuracy of the approximation, it is essential to additionally check whether the uncertainties represented by the posterior are well-calibrated. As Hermans et al. [23] have shown, estimators with high accuracy can still be poorly calibrated. To this end, simulation-based calibration techniques [7] allow evaluation of posterior statistical calibration without requiring access to the reference posterior. Essentially, these checks repeat the inference procedure many times on a simulated test set of parameters and corresponding (simulated) observed data, and check the coverage of the test parameters under the approximated posteriors. Common methods include *Simulation-Based Calibration* of the posterior marginals (SBC, [46]), calculation of expected coverage across all posterior dimensions [11, 13] or *Tests of Accuracy with Random Points* (TARP, [30]).

3.4 Trade-offs Between different SBI Methods

The three simulation-based inference (SBI) approaches—Neural Posterior Estimation (NPE), Neural Likelihood Estimation (NLE), and Neural Ratio Estimation (NRE)—address distinct learning problems and exhibit varying strengths in practical applications. NPE directly regresses the posterior density over the parameter space, conditioned on the observation $\mathbf{x}$. NLE,



(a) Full frontal. (b) Side impact. (c) Side impact.

Figure 3: Illustration of crash scenarios in the US NCAP protocol, according to [6]. In the full frontal crash, two crash dummies are seated in the front and the vehicle is accelerated to 56 km/h prior to impact. In the scenario of a side crash, the car is hit by a moving barrier at 55 km/h. In the last scenario, the vehicle impacts a pole at 32 km/h with one crash dummy in the driver's seat. Figure adapted from [24].

conversely, regresses the likelihood density over the observation space, conditioned on the parameter θ. NRE, unlike the others, does not perform density estimation; instead, it solves a classificatio problem to estimate the likelihood-to-evidence ratio over the joint parameter-observation space, taking both θ and $\mathbf{x}$ as input. This classificatio approach can offer computational benefit compared to density estimation.

The choice of approach depends significantl on the dimensionalities of the parameter space, $\dim(\theta)$, and the observation space, $\dim(\mathbf{x})$. When $\dim(\theta) \ll \dim(\mathbf{x})$, NPE may be preferred as it learns a density in a lower-dimensional space compared to NLE. Conversely, if $\dim(\theta) \gg \dim(\mathbf{x})$, NLE and NRE may be more suitable. However, while NLE and NRE can simplify the learning task, they require sampling techniques such as Markov Chain Monte Carlo (MCMC) or variational inference to obtain posterior samples for each new observation $\mathbf{x}$. Critically, only the training phase is amortized in NLE and NRE; posterior sampling is required for every new observation. Furthermore, MCMC methods do not allow for the direct evaluation of the posterior log probability (except for some variants of NRE [36]). In contrast, NPE is fully amortized, enabling direct sampling and evaluation of the posterior density through the learned neural network. This full amortization is particularly advantageous for simulation-based calibration methods, as it facilitates rapid repetition of inference across a test set, eliminating the need for individual sampling runs. While NLE and NRE can be beneficia when the parameter space is small relative to the observation space, MCMC in high-dimensional parameter spaces can be challenging.

For inference targeted at a specifi observation $\mathbf{x}_o$, sequential or active learning approaches can improve efficien y [39, 35, 18, 33]. However, these methods introduce greater algorithmic complexity, require careful hyperparameter selection, and often demand use-case-specifi tuning [34].

Following this overview of SBI methods, we will apply NPE, NLE, and NRE to a problem from automotive vehicle safety using the `sbi` toolkit [2, 47] in the subsequent section.

4 NUMERICAL EXAMPLES

To illustrate the application of SBI for IUQ in computational mechanics, we utilize an effi cient simulator, allowing us to perform a substantial number of simulations without incurring excessive computational costs. Our primary objective is to showcase the versatility and ease of use of SBI via the `sbi` software package, rather than conducting a comprehensive benchmark

of different SBI methods for this specifi task. For a detailed comparative analysis of various SBI methods on a set of benchmark tasks we refer the reader to Lueckmann et al. [34].

4.1 Problem Description

We employ a simulator based on the US New Car Assessment Program (US NCAP), as detailed in Herold et al. [24], to demonstrate the use of SBI for IUQ. The US NCAP encompasses a range of vehicle safety tests, collecting data from sensors on crash test dummies, vehicles, barriers, and advanced driver assistance systems. This data serves to assess the risk of severe injuries to vehicle occupants and other parties involved in accidents. An illustration of the crash scenarios is provided in Figure 3.

In this firs problem, we focus exclusively on the crash tests within the US NCAP framework, which assess the risk of injury across a series of scenarios based on measurements collected from crash test dummies. The problem is modeled as follows:

$$\mathbf{x} = \mathcal{M}_{\text{US NCAP}}(\theta) + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma^2), \tag{8}$$

where $\theta \in \Theta_{\text{US NCAP}} \subset \mathbb{R}^{21}$ represents a relatively high-dimensional parameter space, describing simulated aggregations of sensor measurements. Conversely, $\mathbf{x}_{\text{US NCAP}} \in \mathcal{X}_{\text{US NCAP}} \subset \mathbb{R}$ indicates a low-dimensional (scalar) data space, representing the fina security rating. The additive Gaussian noise, ϵ, can be controlled by adjusting the variance parameter σ. Note that the simulator implements a strongly non-linear reduction in dimensionality, mapping the 21-dimensional vehicle parameters to a single, scalar security rating. This reduction is intentional, as the security rating is the primary metric of interest. However, as a consequence, the US NCAP problem might exhibit numerous complementary solutions distributed across the parameter space. Therefore, the posterior distribution must encompass a substantial portion of the parameter space and can possess a complex shape, as described by Herold et al. [24].

4.2 A Priori Checks

Before proceeding with posterior inference, we briefl address a priori checks. In this fully simulated scenario, the simulated observations naturally fall within the prior predictive distribution, confirmin the suitability of our prior distributions (Figure 4). While this check is trivial in this context, it is a crucial firs step in practical applications with real-world data, where validating the chosen prior distributions and ensuring the suitability of the simulation model is essential.

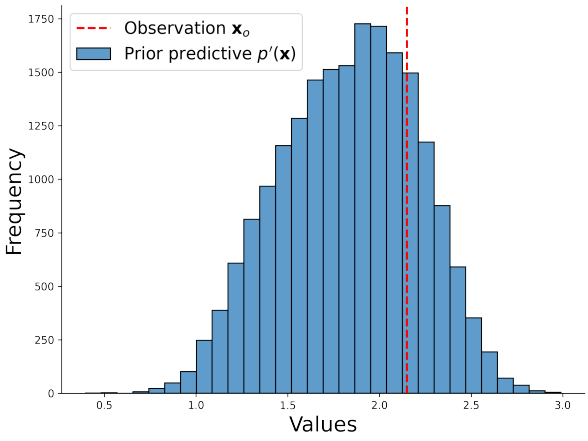


Figure 4: Prior predictive check showing the empirical distribution from samples of the prior predictive distribution Equation (4) and an observation $\mathbf{x}_o$ on top.

4.3 Posterior Approximation

After passing the a-priori check, we proceeded with the actual inference part and used NPE, NLE, and NRE to perform SBI on the US NCAP problem. To evaluate the impact of the size of the training dataset, each algo-

rithm was trained three times with different dataset sizes $N \in \{10^3, 10^4, 10^5\}$. Training was conducted until convergence as measured by a moving average of the loss on a held-out validation dataset of the same size as the training dataset. Additionally, a separate test dataset was generated for later comparison across training budgets and SBI methods (see Listing 1).

```
import sbi
import us_ncap_simulator

prior = sbi.util.torchutils.BoxUniform(...)
simulator = us_ncap_simulator.get_simulator()

thetas, thetas_val = prior.sample((N,)), prior.sample((N,))
xs, xs_val = simulator(thetas), simulator(thetas_val)

thetas_test = prior.sample((1_000,))
xs_test = simulator(thetas_test)
```

Listing 1: Exemplary code for data generation.

The posterior approximation was performed using the inference API provided by the `sbi` package (see Listing 2). When instantiating the inference object, the prior distribution must be specifie for subsequent inference. Additionally, various configuration governing the inference process can be directly passed, such as the sampling method for MCMC-based inference in the case of NLE and NRE. By simply interchanging the inference method—i.e., replacing `NPE` with `NLE` or `NRE`—the posterior distribution can be approximated according to the different methods.

```
import sbi

inference = sbi.inference.NPE(prior=prior, ...)
inference = inference.append_simulations(thetas, xs)

density_estimator = inference.train(...)

posterior = inference.build_posterior(density_estimator)
```

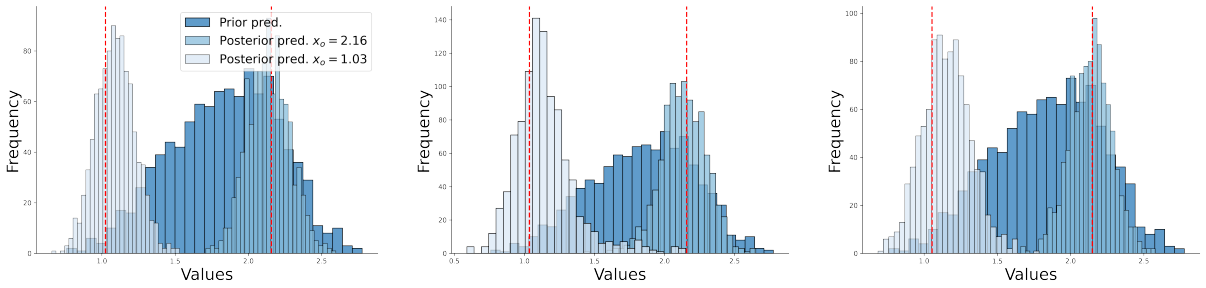
Listing 2: Exemplary code for neural posterior estimation using `sbi`.

```
from src.utils import ppc

posterior_samples = posterior.sample_batched((1000,), xs_test)

posterior_predictive_samples = simulator(posterior_samples)

median_dist = ppc(posterior_predictive_samples, xs_test)
```

Listing 3: Exemplary code for computing the PPC

4.4 Performance evaluation

We used several metrics to evaluate inference performance, tailored to each SBI method. For direct comparison between all methods and training budgets, we assessed PPC performance through visual comparison of prior and posterior predictive distributions and by calculating the median distance between test data and data generated from samples of the approximate posterior (see Listing 3). Additionally, we conducted calibration checks for all methods and budgets using SBC (see Listing 4). SBC can be performed for all marginals separately via ranking of



(a) NPE (b) NLE (c) NRE

Figure 5: Visual Posterior predictive check (PPC) for the US NCAP task using the three methods NPE, NLE, and NRE with training data budget of $N = 10^5$. Each subplot shows the prior predictive distribution as seen in Figure 4, two randomly chosen observations $\mathbf{x}_o^{(1)}$ and $\mathbf{x}_o^{(2)}$ (vertical lines), and the respective posterior predictive distributions.

parameter values [46] or via expected coverage using the log-probability of the approximate posterior [11]. We calculated SBC via the marginals in order to enable comparison across all three SBI methods, as NLE and NRE do not allow evaluation of the posterior.

```
    from sbi.analysis.plot import sbc_rank_plot
    from sbi.diagnostics import run_sbc

    ranks, _ = run_sbc(thetas_test, xs_test, posterior, \dots)

    # Plot the SBC as empirical CDFs.
    fig, ax = sbc_rank_plot(ranks, num_posterior_samples=1000, num_bins=50)
```

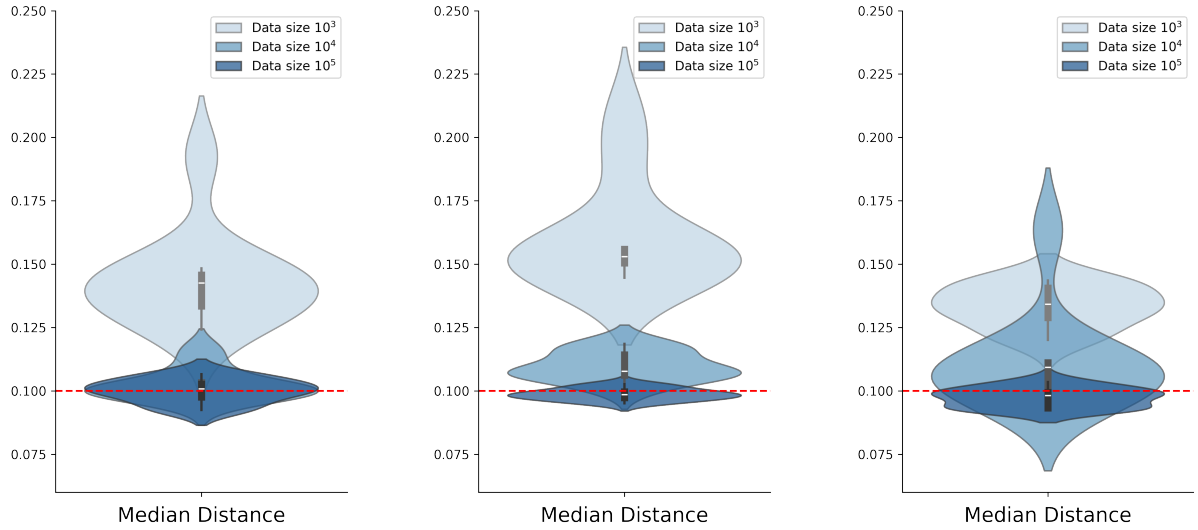
Listing 4: Exemplary code for performing SBC

Notably, all these auxiliary functions are readily available within the `sbi` package. Furthermore, the package provides additional functionality for quality analysis, including (local) classifie two-sample tests, SBC, TARP, pair plots, sensitivity analysis, and conditional density plots. For a comprehensive list of available functions, we direct the reader to the package documentation[2].

4.4.1 Posterior predictive performance

The visual Posterior Predictive Checks (PPC) demonstrated that the predictive distributions generated by all three SBI methods were concentrated around the observed data on which the posterior was conditioned (Figure 5). This indicates that the simulated data from the posterior predictive distributions closely matched the observed data. The more systematic evaluation of the median distances between the test data point and posterior predictive samples confirme this observation, showing an improvement with increasing simulation budget and converging towards the reference value (Figure 6). In this comparison, NRE demonstrated slightly faster convergence in predictive performance compared to the other two methods.

[2] https://sbi-dev.github.io/sbi



(a) NPE (b) NLE (c) NRE

Figure 6: Posterior predictive check (PPC) for the US NCAP task using the three methods NPE, NLE, and NRE with training data sizes of $N \in \{10^3, 10^4, 10^5\}$. The plot shows the median distance between the posterior predictive samples and the true data for 200 test points with 10^3 samples from the posterior predictive, each. The closer the median distance to zero, the better the posterior approximation. Note, the variance σ^2 of the additive white noise of the simulator was set to 0.1, depicted by the horizontal red line.

4.4.2 Posterior calibration performance

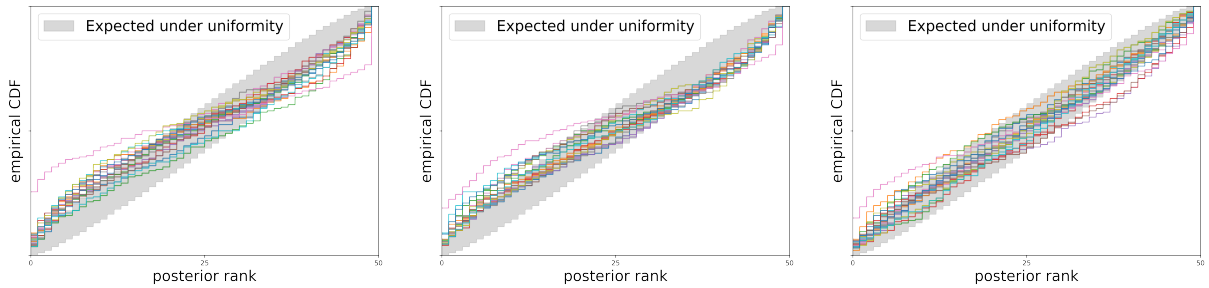
SBC results indicated that for NLE and NRE, all marginal distributions were well-calibrated. For NPE, one marginal distribution exhibited slight overconfidence characterized by an excessively narrow credible interval (Figure 7). However, with increasing training budget, this overconfidenc reduced, indicating that NPE might require more training data to achieve a well-calibrated posterior. We provide the code for all the experiments conducted for this work as repository on GitHub[3].

5 CONCLUSION

In computational mechanics, accurately quantifying uncertainties in material properties, geometric tolerances, and system parametrizations is essential for robust and reliable system design. Inverse Uncertainty Quantificatio (IUQ) is a crucial methodology for addressing these uncertainties, particularly in safety-critical applications. However, classical Bayesian inference methods often prove impractical for complex, computationally expensive models with intractable likelihoods. Simulation-Based Inference (SBI) offers a powerful alternative, enabling Bayesian inference without requiring explicit likelihood calculations.

This work demonstrated the application of Simulation-Based Inference (SBI) for Inverse Uncertainty Quantificatio (IUQ) within a Bayesian framework. A simplifie example problem, based on US New Car Assessment Program (US NCAP) automotive safety simulations, was employed to illustrate the functionality of the sbi toolkit. Due to the absence of an analytically tractable likelihood function, traditional Bayesian inference methods were not applicable to

[3]Code provided at `github.com/aai-institute/uncecomp-2025-iuq-via-sbi`.



(a) 10^3 Training samples (b) 10^4 Training samples (c) 10^5 Training samples

Figure 7: SBC results of NPE per data size. In line with the PPC, 200 test samples with 10^3 posterior samples, each, are used to compute the ranks. A perfectly calibrated posterior would yield a uniform distribution of ranks. The 99% confidenc interval of a uniform distribution is denoted by the gray-shaded area.

this problem. However, SBI facilitated the implementation of the standard Baye sian workfl w, encompassing prior predictive checks, posterior inference, and posterior evaluation.

The results obtained from this example demonstrated general SBI workfl w and showed a good predictive performance for the US NCAP model. However, we observed that the inference process resulted in only a slight constraint of the parameters, indicating that the posterior distribution remained relatively close to the prior. This is likely due to the low dimensionality of the data output compared to the high dimensionality of the parameter space (e.g., 1 vs. 21). To achieve a more constrained posterior distribution, it would be necessary to modify the simulator to generate a higher-dimensional data output. Additionally, the calibration checks revealed a slight mis-calibration for the NPE method, indicating the need for more training data to obtain fully calibrated posterior estimate. The fact that this occurred only for NPE can be explained by the difference in learning problems solved by the different approaches (see Section 3.4, e.g., NPE has to estimate a 21-dimensional posterior and NLE has to estimate a 1-dimensional likelihood.

More generally, in practical applications involving complex simulators, high-dimensional parameter spaces, or data outputs, SBI methodologies must be carefully adapted and tailored to the specifi use case. This may involve sophisticated neural network architectures, advanced sampling techniques, or customized evaluation metrics. However, it is worth noting that the SBI fiel benefit from a very active and collaborative community of researchers and open-source developers, which can significantl aid in addressing these challenges.

In summary, SBI presents a viable and developing technique for IUQ in computational mechanics and related fields Its capacity to handle complex models and intractable likelihoods offers new possibilities for uncertainty quantification facilitating the design of more robust and reliable systems. Further advancements in SBI methodologies and their application across engineering disciplines are anticipated.

Acknowledgements

The authors would like to thank the community of developers behind the sbi toolkit. The authors also thank the TransferLab at the appliedAI Institute for Europe for supporting the research and submission. MH thanks the BMW Group for supporting the research, especially Dr. Jonas S. Jehle, and Prof. Dr. Matthias Gerdts from the University of the Bundeswehr Munich for their valuable discussions on the matter.

REFERENCES

[1] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Information Science and Statistics. Springer, New York, 2006.

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

[3] Jan Boelts, Philipp Harth, Richard Gao, Daniel Udvary, Felipe Yáñez, Daniel Baum, Hans-Christian Hege, Marcel Oberlaender, and Jakob H. Macke. Simulation-based inference for efficien identificatio of generative models in computational connectomics. *PLOS Computational Biology*, 19(9):e1011406, September 2023.

[4] Jan Boelts, Jan-Matthis Lueckmann, Richard Gao, and Jakob H Macke. Flexible and efficien simulation-based inference for models of decision-making. *eLife*, 11:e77220, July 2022.

[5] Jan Bölts. *Advancing Methods and Applicability of Simulation-Based Inference in Neuroscience*. PhD thesis, Universität Tübingen, July 2023.

[6] carhs. SafetyCompanion 2024. 2024.

[7] Samantha R. Cook, Andrew Gelman, and Donald B. Rubin. Validation of Softw are for Bayesian Models Using Posterior Quantiles. *Journal of Computational and Graphical Statistics*, 15(3):675–692, 2006.

[8] Kyle Cranmer, Johann Brehmer, and Gilles Louppe. The frontier of simulation-based inference. *Proceedings of the National Academy of Sciences*, 117(48):30055–30062, December 2020.

[9] Miles D. Cranmer, Richard Galvez, Lauren Anderson, David N. Spergel, and Shirley Ho. Modeling the Gaia Color-Magnitude Diagram with Bayesian Neural Flows to Constrain Distance Estimates, August 2019.

[10] Maximilian Dax, Jonas Wildberger, Simon Buchholz, Stephen R. Green, Jakob H. Macke, and Bernhard Schölkopf. Flow Matching for Scalable Simulation-Based Inference, May 2023.

[11] Michael Deistler, Pedro J. Goncalves, and Jakob H. Macke. Truncated proposals for scalable and hassle-free simulation-based inference, November 2022.

[12] Conor Durkan, Artur Bekasov, Iain Murray, and George Papamakarios. Neural Spline Flows. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[13] Maciej Falkiewicz, Naoya Takeishi, Imahn Shekhzadeh, Antoine Wehenkel, Arnaud Delaunoy, Gilles Louppe, and Alexandros Kalousis. Calibrating Neural Simulation-Based Inference with Differentiable Coverage Probability, October 2023.

[14] David T. Frazier and Christopher Drovandi. Robust Approximate Bayesian Inference With Synthetic Likelihood. *Journal of Computational and Graphical Statistics*, 30(4):958–976, October 2021.

[15] J H Friedman. On multivariate goodness of fit and two sample testing. 2003.

[16] Andrew Gelman, Aki Vehtari, Daniel Simpson, Charles C. Margossian, Bob Carpenter, Yuling Yao, Lauren Kennedy, Jonah Gabry, Paul-Christian Bürkner, and Martin Modrák. Bayesian Workflow, November 2020.

[17] Pedro J Gonçalves, Jan-Matthis Lueckmann, Michael Deistler, Marcel Nonnenmacher, Kaan Öcal, Giacomo Bassetto, Chaitanya Chintaluri, William F Podlaski, Sara A Haddad, Tim P Vogels, David S Greenberg, and Jakob H Macke. Training deep neural density estimators to identify mechanistic models of neural dynamics. *eLife*, 9:e56261, September 2020.

[18] David S. Greenberg, Marcel Nonnenmacher, and Jakob H. Macke. Automatic Posterior Transformation for Likelihood-Free Inference, May 2019.

[19] Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. A Kernel Two-Sample Test. *Journal of Machine Learning Research*, 13(25):723–773, 2012.

[20] Lukas N. Groschner, Jonatan G. Malis, Birte Zuidinga, and Alexander Borst. A biophysical account of multiplication by a single neuron. *Nature*, 603(7899):119–123, March 2022.

[21] Michael U. Gutmann, Ritabrata Dutta, Samuel Kaski, and Jukka Corander. Likelihood-free inference via classification *Statistics and Computing*, 28(2):411–425, March 2018.

[22] Joeri Hermans, Volodimir Begy, and Gilles Louppe. Likelihood-free MCMC with Amortized Approximate Ratio Estimators, June 2020.

[23] Joeri Hermans, Arnaud Delaunoy, François Rozet, Antoine Wehenkel, Volodimir Begy, and Gilles Louppe. A Trust Crisis In Simulation-Based Inference? Your Posterior Approximations Can Be Unfaithful, December 2022.

[24] Maternus Herold, Jonas Siegfried Jehle, and Matthias Gerdts. Tractable Inverse Uncertainty Quantificatio via Probabilistic Circuits for Applied Mechanics. *under review*.

[25] Maternus Herold, Jonas Siegfried Jehle, and Matthias Gerdts. Inverse Uncertainty Quantificatio in Passive Automobile Safety: Bayesian Inversion for Optimal Safety Ratings with Robustness Quantification In *9th European Congress on Computational Methods in Applied Sciences and Engineering*, Lisbon, October 2024.

[26] Maternus Herold, Anna Veselovska, Jonas Jehle, and Felix Krahmer. Non-intrusive surrogate modelling using sparse random features with applications in crashworthiness analysis. *CoRR*, abs/2212.14507, 2022.

[27] Daolang Huang, Ayush Bharti, Amauri Souza, Luigi Acerbi, and Samuel Kaski. Learning Robust Statistics for Simulation-based Inference under Model Misspecification May 2023.

[28] Jonas Siegfried Jehle. *Uncertainty Management Framework for Automotive Crash Applications*. PhD thesis, Universität der Bundeswehr München, 2022.

[29] Ryan P. Kelly, David J. Nott, David Tyler Frazier, David J. Warne, and Christopher Drovandi. Misspecification-robust Sequential Neural Likelihood for Simulation-based Inference. *Transactions on Machine Learning Research*, March 2024.

[30] Pablo Lemos, Adam Coogan, Yashar Hezaveh, and Laurence Perreault-Levasseur. Sampling-Based Accuracy Testing of Posterior Estimators for General Inference, June 2023.

[31] Qiang Liu, Jason Lee, and Michael Jordan. A Kernelized Stein Discrepancy for Goodness-of-fit Tests. In *Proceedings of The 33rd International Conference on Machine Learning*, pages 276–284. PMLR, June 2016.

[32] David Lopez-Paz and Maxime Oquab. Revisiting Classifier Two-Sample Tests. In *International Conference on Learning Representations*, February 2017.

[33] Jan-Matthis Lueckmann, Giacomo Bassetto, Theofanis Karaletsos, and Jakob H. Macke. Likelihood-free inference with emulator networks. In *Proceedings of The 1st Symposium on Advances in Approximate Bayesian Inference*, pages 32–53. PMLR, January 2019.

[34] Jan-Matthis Lueckmann, Jan Boelts, David S. Greenberg, Pedro J. Gonçalves, and Jakob H. Macke. Benchmarking Simulation-Based Inference, April 2021.

[35] Jan-Matthis Lueckmann, Pedro J. Goncalves, Giacomo Bassetto, Kaan Öcal, Marcel Nonnenmacher, and Jakob H. Macke. Flexible statistical inference for mechanistic models of neural dynamics. arXiv, November 2017.

[36] Benjamin Kurt Miller, Christoph Weniger, and Patrick Forré. Contrastive Neural Ratio Estimation, October 2022.

[37] Joseph B. Nagel and Bruno Sudret. Bayesian Multilevel Model Calibration for Inverse Problems Under Uncertainty with Perfect Data. *Journal of Aerospace Information Systems*, 12(1):97–113, January 2015.

[38] George Papamakarios. Neural Density Estimation and Likelihood-free Inference, October 2019.

[39] George Papamakarios and Iain Murray. Fast ϵ-free Inference of Simulation Models with Bayesian Conditional Density Estimation. 2016.

[40] George Papamakarios, Theo Pavlakou, and Iain Murray. Masked autoregressive flow for density estimation. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[41] Stefan T. Radev, Marvin Schmitt, Lukas Schumacher, Lasse Elsemüller, Valentin Pratz, Yannik Schälte, Ullrich Köthe, and Paul-Christian Bürkner. Bayesfl w: Amortized bayesian workfl ws with neural networks. *Journal of Open Source Software*, 8(89):5702, 2023.

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

[43] Marvin Schmitt, Paul-Christian Bürkner, Ullrich Köthe, and Stefan T. Radev. Detecting Model Misspecificatio in Amortized Bayesian Inference with Neural Networks: An Extended Investigation, June 2024.

[44] Scott A. Sisson, Yanan Fan, and Mark Beaumont. *Handbook of Approximate Bayesian Computation*. CRC Press, September 2018.

[45] T.J. Sullivan. *Introduction to Uncertainty Quantification*, volume 63 of *Texts in Applied Mathematics*. Springer International Publishing, Cham, 2015.

[46] Sean Talts, Michael Betancourt, Daniel Simpson, Aki Vehtari, and Andrew Gelman. Validating Bayesian Inference Algorithms with Simulation-Based Calibration, October 2020.

[47] Alvaro Tejero-Cantero, Jan Boelts, Michael Deistler, Jan-Matthis Lueckmann, Conor Durkan, Pedro J. Gonçalves, David S. Greenberg, and Jakob H. Macke. Sbi: A toolkit for simulation-based inference. *Journal of Open Source Software*, 5(52):2505, August 2020.

[48] Daniel Ward, Patrick Cannon, Mark Beaumont, Matteo Fasiolo, and Sebastian M. Schmon. Robust neural posterior estimation and statistical model criticism. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, pages 33845–33859, Red Hook, NY, USA, November 2022. Curran Associates Inc.

[49] Xu Wu, Tomasz Kozlowski, Hadi Meidani, and Koroush Shirvan. Inverse uncertainty quantificatio using the modular Bayesian approach based on Gaussian process, Part 1: Theory. *Nuclear Engineering and Design*, 335:339–355, August 2018.

[50] Xu Wu, Tomasz Kozlowski, Hadi Meidani, and Koroush Shirvan. Inverse uncertainty quantificatio using the modular Bayesian approach based on Gaussian Process, Part 2: Application to TRACE. *Nuclear Engineering and Design*, 335:417–431, August 2018.