The 9th European Congress on Computational Methods in Applied Sciences and Engineering
ECCOMAS Congress 2024
3–7 June 2024, Lisboa, Portugal

INVERSE UNCERTAINTY QUANTIFICATION IN AUTOMOTIVE PASSIVE SAFETY: BAYESIAN INVERSION FOR OPTIMAL SAFETY RATINGS WITH ROBUSTNESS QUANTIFICATION

Maternus Herold[1,2,3], Jonas S. Jehle[1] and Matthias Gerdts[3]

[1] BMW Research and Development Center Munich
BMW Group, Munich, Germany
email: maternus.herold@bmw.de, www.bmwgroup.com

[2] TransferLab
appliedAI Institute for Europe gGmbH, Munich, Germany
www.transferlab.ai

[3] Institute for Applied Mathematics and Scientific Computing
University of the Bundeswehr Munich, Neubiberg, Germany
www.unibw.de/lrt1

Key words: Uncertainty Quantification, Bayesian Inversion, Simulation-Based Inference, Passive Safety

Summary.

Calibrating restraint systems for crashworthiness is challenging due to the non-linear behavior of input parameters. While expert knowledge provides initial parameter estimates, complementary parameterization and dependencies complicate the task. The presented approach approximates a distribution on the parameter space that results in a desired observation distribution. We propose initializing the source distribution with the average posterior distribution and fine-tuning it against the marginal likelihood over the targeted distribution. Additionally, a weighting function is introduced to obtain stable point estimates from the source distribution. The method is applicable to various forward models and is demonstrated on a passive safety rating use case.

1 INTRODUCTION

The parameter vector θ, defining a restrating system, undergos a highly non-linear transformation to result in the observation vector x, describing the crashworthiness of a car. While expert knowledge and component specifications provide an initial guess on the range and location of the correct parameters, complementary parameterization and parameter dependencies pose challenges to the engineers. Currently, the process of calibration is forward oriented and iterative. Ideally, it could be inverted and the engineer would define a targeted distribution $p^*(x)$ on the observation space, to which a distribution on the parameter space is provided that fulfills the targeted distribution. This is the source distribution $p(\theta)$ that we aim to approximate in this study.

While desireable, the source distribution is intractable as the relationship between parameters θ and observations x is defined by a complex simulator [11], which does not provide a tractable inverse. However, in the case of stochastic simulators, the observation x can be treated as a sample from the likelihood $p(x \mid \theta)$ given a parameter θ, allowing for Bayesian inversion.

In accordance with a line of work on empirical Bayes, most notably [22, 23], we formulate the task as likelihood-free source distribution estimation as the forward model does not provide an analytical form of the likelihood in general [4, 15]. Provided a dataset of parameter-observation pairs $\mathcal{D} = \{(\theta_i, x_i)\}_{i=1}^N$ and a distribution $p^*(x)$ within the support of the prior predictive distribution of the simulator, we aim to approximate a source distribution $p(\theta)$ that yields the observations $x \sim p^*(x)$ when pushed through the forward model.

$$p^*(x) = \int p(x \mid \theta) p(\theta) \; d\theta \tag{1}$$

Due to the complexity of the forward model and the ill-posedness of the problem, finding a suitable source distribution is a challenging task. We therefore propose to initialize the source distribution by the average posterior distribution and fine-tuning it subsequently.

Estimating a source distribution is similar to Simulation-Based Inference (SBI) [4], but essentially differs in the distribution being approximated. Particularly, SBI is used here to obtain the average posterior. This is in contrast to empirical Bayes, where SBI is treated as a downstream task [22, 23].

Restraint systems require a fixed parameterization via point estimates, conditioned on the detected crash scenario. We propose to employ a weighting function in the selection process, taking the stability under perturbation into account. The weighting function can be further adapted to include economical factors as well.

In the following, we will introduce the necessary background on neural density estimation (Section 2) and Bayesian inversion (Section 3). The presented method is described subsequently (Section 4) and is finally applied to a passive safety rating (Section 5).

2 NEURAL DENSITY ESTIMATION

Conditional neural density estimation extends density estimation to model conditional distributions, where the objective is to estimate the density of a target variable x given a conditioning variable θ [15, 18, 24], with neural networks.

Normalizing flows, a key method within neural density estimation, are adapted to incorporate the conditioning variable θ into their transformation, resulting in conditional transformations of the form $f_k(z_{k-1}, \theta)$, where each f_k is a differentiable and invertible mapping (diffeomorphism) [15]. By utilizing a sequence of neural networks for their transformations, normalizing flows allow to model a complex distribution. The variable x is then derived as $x = f(z, \theta)$ where $f = f_1 \circ \cdots \circ f_k$. The conditional density $p_{\mathcal{X}}(x \mid \theta)$ is computed using the change of variables formula, adjusted for the conditional case, where $z_0 = z$, a sample from the base distribution, and $z_k = f_k(z_{k-1}, \theta)$:

$$\log p_{\mathcal{X}}(x \mid \theta) = \log p_{\mathcal{Z}}(z \mid \theta) - \sum_{k=1}^{K} \log \left| \det \left(\frac{\partial f_k(z_{k-1}, \theta)}{\partial z_{k-1}} \right) \right| . \tag{2}$$

In order to train these density estimators, the Kullback-Leibler (KL) divergence between the learned and observed conditional distributions is minimized, which is equivalent to the minimization of the negative log likelihood [1, 15]. Therefore, the weights are optimized by minimizing the conditional negative log likelihood (CNLL) over the dataset $\mathcal{D} = \{(\theta_i, x_i)\}_{i=1}^N$:

$$\mathcal{L}_{\text{CNLL}} = -\frac{1}{N} \sum_{i=1}^{N} \log p_{\mathcal{X}}(x_i|\theta_i). \tag{3}$$

3 BAYESIAN INVERSION VIA LIKELIHOOD-FREE INFERENCE

Bayes' theorem models the distribution of a system's parameters given observed data. By combining prior knowledge with observed data, the posterior distribution represents updated beliefs on the parameter vector. The posterior distribution provides an inverse relation of the parameter vector and the observed data and is often used for inversion [2, 5, 14].

The posterior distribution is defined as the conditional probability of the system's parameters θ, conditioned on the observation x. According to Bayes' theorem, the posterior is the product of the likelihood of the observation for a specific parameterization $p(x \mid \theta)$, the prior distribution on the parameter vector $\pi(\theta)$, and the evidence $p(x) = \int p(x \mid \theta)\pi(\theta)\ d\theta$. However, as the evidence is intractable in most cases, the posterior can only be established up to proportionality:

$$p(\theta \mid x) = \frac{p(x \mid \theta)\pi(\theta)}{p(x)} \propto p(x \mid \theta)\pi(\theta). \tag{4}$$

By computing the posterior, the inverse relation of the forward model $\mathcal{M}(\theta) = \tilde{x}$ is established, to which the likelihood belongs. The obtained posterior can be conditioned on a specific observation x_o to yield a distribution over parameters θ generating the observation x_o with high probability.

Bayes theorem is difficult to apply to industrial applications as the likelihood of most forward models $\mathcal{M}$ is intractable [4, 15]. To address this issue, neural Simulation-Based Inference (SBI) defines a parametric approximation $q_\phi(\theta \mid x) \approx p(\theta \mid x)$ by learning parts of Equation (4). In the case of Neural Posterior Estimation (NPE) and Neural Likelihood Estimation (NLE), conditional neural density estimators (see Section 2) are used to model the conditional densities, minimizing the KL divergence.

As minimizing the KL divergence is equivalent to maximizing the likelihood of the observed data under the model, the parameters ϕ of the conditional density estimator are trained by minimizing the conditional negative log likelihood of the data $\mathcal{D}$ under the model q_ϕ (see Section 2).

In order to minimized the loss on dataset of samples from the joint distribution, the fact that

$$p(x \mid \theta)p(\theta) = p(x, \theta) = p(\theta \mid x)p(x) \tag{5}$$

is used. Treating observations from a numerical model $\mathcal{M}$, evaluated at the parameter vector θ, as sampled from the likelihood $p(x \mid \theta)$, a dataset of samples from the joint distribution $p(\theta, x)$ can be obtained as outlined in Algorithm 1.

We would like to point to the following references for further information on the topic. Cranmer et al. [4] provide a general overview of old and new approaches to SBI. Papamakarios et al. [15, 16, 17] describe the step from Approximate Bayesian Computation (ABC) to neural SBI by using neural density estimation. Lückmann et al. [13] compare different density estimators for neural posterior, likelihood, and ratio estimation on different tasks and establishes a benchmark.

Algorithm 1 Sampling from the joint distribution in SBI.

Input: Prior distribution $\pi(\theta)$, simulator function $\mathcal{M} : \theta \to x$, number of samples N
Output: Samples from the joint distribution $\mathcal{D} = \{(\theta_i, x_i)\}_{i=1}^{N}$
Initialize an empty list of samples $\mathcal{D} \leftarrow [\ ]$
for $i \leftarrow 1$ to N **do**
    Sample parameter $\theta_i \sim \pi(\theta)$
    Generate simulated data $x_i \leftarrow \mathcal{M}(\theta_i)$
    Append (θ_i, x_i) to the list of samples $\mathcal{D}$
end for
Return $\mathcal{D}$

Finally, the improvement of neural density estimators is an ongoing line of research with recent works proposing continuous flows for density estimators [6, 12, 21], score estimation [8, 19], as well as flexible approaches [9]. Several methods for SBI are implemented in the open-source Python package sbi-toolkit [20], which is used in this work as well.

4 THE INVERSE PROBABILITY METHOD

A common engineering task is to optimize the parameters θ such that the system produces a certain anticipated or desired result x^*. Provided a forward model $\mathcal{M}$ describing our system under consideration and taking d-dimensional vectors θ to produced a n-dimensional observation $\tilde{x}$, we have $\mathcal{M}(\theta) = \tilde{x}$, where $\theta \in \mathbb{R}^d$ and $\tilde{x} \in \mathbb{R}^n$.

The difficulty in this task arises from the fact that (1) the model $\mathcal{M}$ is often complex and computationally expensive to evaluate[1]. In addition, (2) the underlying systems are non-injective, i.e. allowing for complementary parameterization that result in the same observation. Due to (1), global optimization of the parameters θ directly is infeasible and due to (2), defining an inverse mapping $\mathcal{M}^{-1}$ is an ill-posed problem. Finally, an ideal workflow would allow to incorporate the notion of robustness directly into the formulation of the targeted results x^*.

In order to address the mentioned challenges, the here presented workflow allows to compute the Bayesian inverse of the forward model $\mathcal{M}$, construct a source distribution that produces a target distribution of the anticipated results, and select robust point-estimates for final system parameterization. This way, the engineering workflow is inverted and allows to ask: *"What parameters θ result in the anticipated results x^* with high probability?"*.

4.1 Bayesian Inversion via Simulation-Based Inference

The posterior distribution for a given observation x_o is the theoretical construct that defines the distribution over parameters θ, that are likely to have lead to the observation. In order to obtain the posterior, as stated in Equation (4), the likelihood is required. Due to the complexity and dimensionality of the underlying system or the system being a black box, the likelihood is not tractable analytically [4, 15, 16]. Popular choices in this setting are manually crafting a likelihood or using a sampling-based approach, e.g. via Markov-Chain Monte-Carlo (MCMC). However, these approaches don't scale well to higher dimensions or are not feasible due to

[1] The physical crash test has a high cost incurred due to the car being damaged as well as the required setup for running such tests. This includes, among others, fire safety, crash dummy preparation, and wear of the crash-test facility. Furthermore, complete crash simulations are very complex and take several hours on large clusters.

The workflow is structured as follows.

1. Define a target distribution for the anticipated results $p^*(x)$.
2. Provide an amortized estimator for the posterior distribution $p(\theta \mid x)$.
3. Obtain a source distribution $p(\theta)$ that produces the targeted distribution $p^*(x)$.
4. Incorporate the notion of robustness into the selection of point-estimates for θ.

the high computational cost of the forward simulation. We chose to learn the likelihood in a data-drive way as detailed in Section 3. For this task, we use a neural density estimator (see Section 2). Such estimators allow to capture multi-modal distributions, which are required due to complementary solutions. Utilizing neural networks allows to capture highly-nonlinear effects within the problem. Depending on the complexity of the problem at hand, the complexity of the estimator can be adapted to facilitate the required expressiveness [9, 12, 21].

According to Section 3, we choose to compute a likelihood estimate $q_\phi(x \mid \theta) \approx p(x \mid \theta)$ instead of directly learning the posterior distribution. This is due to the facts that (1) the likelihood allows for hierarchical extensions of the problem without retraining the model. This includes the case when i.i.d. components are added to the system, which provide their own likelihood. Furthermore, (2) the likelihood can be treated as probabilistic surrogate model as well and is easier to validate, when no true posterior is available. In addition, (3) the parameter dimensionality in passive safety problems tends to be of higher dimensionality than the observation space, i.e. $\dim(\theta) \gg \dim(x)$, and (4) the likelihood is required in the robustness assessment of a point estimate in Section 4.3.

Instead of crafting or learning summary statistics as a preceding step to density estimation, we propose to use industry-standard rating functions $g : \mathbb{R}^d \to \mathbb{R}^1$. In the case of passive safety, car assessment protocols evaluate a multitude of sensor measurements $\tilde{x} \in \mathbb{R}^n$ of a single crash test to compute the relative injury risk as scalar value $x \in \mathbb{R}^1$ (Equation (6)). Incorporating this scalar score as the output effectively compresses the output dimensionality, thereby enhancing the fidelity of likelihood estimations by aligning with the preferential dimensionality relation $\dim(\theta) \gg \dim(x)$. The reduction g is defined as

$$g(\mathcal{M}(\theta)) = x, \; x \in \mathbb{R}^1. \tag{6}$$

Based on the likelihood estimate $q_\phi(x \mid \theta)$, the posterior distribution can be proportionally evaluated via Bayes' theorem (Equation (4)) for a specific combination of θ and x. A Markov-Chain Monte-Carlo sampling schema is used to sample from the posterior. As the estimated posterior is amortized, the posterior can be evaluated for any given observation x_o without the need to retrain the estimator [4].

4.2 Estimation of Source Distribution

Instead of inverting the mechanical model $\mathcal{M}$ for a single observation x_o, one is interested in finding a suitable distribution $q_\psi(\theta)$ over the parameter space $\Theta \subset \mathbb{R}^d$ that results in a desired distribution $p^*(x)$ on the target space. The targeted distribution is analytically defined[2], describing the location and shape of the observations to be made. This way, engineers can specify their targeted observations in form of a distribution, incorporating the location and variance of the allowed observation. To link the source and target distribution, we reformulate the prior predictive distribution to yield a marginal likelihood distribution $q(x) = \int p(x \mid \theta)q_\psi(\theta)\ d\theta$, depending on $q_\psi(\theta)$[3].

$$\begin{aligned} q(x) &= \int p(x \mid \theta)p(\theta)\ d\theta \\ &= \int q_\phi(x \mid \theta)p(\theta)\ d\theta \\ &= \int q_\phi(x \mid \theta)q_\psi(\theta)\ d\theta \end{aligned} \tag{7}$$

Therefore, finding a suitable source distribution for $p^*(x)$ requires to minimize the deviation of the marginal likelihood distribution $q(x)$ and the targeted distribution with respect to the source distribution. Up to our knowledge, this problem was not yet formulated in this fashion. It is however similar to problem of empirical Bayes [22, 23], where one is interested in recovering the true prior distribution with respect to observed data, given a likelihood function.

$$\psi^* = \arg\min_{\psi \in \mathbb{M}} d\left(p^*(x), q(x)\right) \tag{8}$$

In contrast to empirical Bayes, the targeted distribution $p^*(x)$ is stated analytically, rather than by observed samples. This way, we are not limited by the number of observations within a targeted region, but have an analytical function defining the targeted region. This is an advantage especially for mechanical applications as observations from expensive simulators and hardware tests are typically scarce. Furthermore, the task focussed here is not to recover a proposal distribution that led to the observed data but to find a source distribution that produces the targeted distribution. Therefore, the following assumptions are made: (1) the distance function d is differentiable w.r.t. to the parameters ψ and capable to measure the deviation between two distributions, (2) the source distribution $q_\psi(\theta)$ is flexible enough to model the required source over θ, and (3) the targeted distribution $p^*(x)$ is within the support of the marginal likelihood.

In accordance with [22] (Equation 2), we use the KL Divergence as distance function. Utilizing the estimated likelihood, obtained in Section 4.1, we minimize the negative marginal log-likelihood $\log q(x)$ of the marginal likelihood over the targeted distribution.

[2]The (multivariate) Normal distribution can be a good representation of the targeted observations as it's easily constructed. By adjusting the covariances, the shape can be adjusted to meet the required observations.

[3]As the parameterized source distribution q_ψ influences the marginal likelihood over x, we denote this dependence by adding the parameters as subscript, i.e. $q(x)$.

$$
\begin{aligned}
\log q(x) &= \log \mathbb{E}_{q_\psi(\theta)} \left[p(x \mid \theta) \right] \\
&= \log \mathbb{E}_{q_\psi(\theta)} \left[q_\omega(x \mid \theta) \right] \\
&= \log \mathbb{E}_{q_\psi(\theta)} \left[p(x \mid \theta) \right] \\
&\approx \log \sum_{k=1}^{K} p(x \mid \theta_k) \\
&= \mathrm{logSumExp} \left[\log p(x \mid \theta_k) \right] - C
\end{aligned}
\tag{9}
$$

In Equation (9), $\theta_k \sim q_\psi(\theta)$ is from the parametric density model and C is a constant independent of ψ. As Vandegar et al. [22] note, a large number of samples K is required for a good approximation of the expectation. They also show that the bias and variance of Equation (9) decreases at a rate of $\mathcal{O}(\frac{1}{K})$. In addition, the here stated formulation allows to introduce inductive bias into the model in a simple fashion [22]. This especially includes the way the source model is formulated and resembles the source distribution.

We use a normalizing flow (Section 2) to model the source distribution $q_\psi(\theta)$. The samples K to approximate Equation (9) are generated by sampling from a base distribution $z \sim p_{\mathcal{Z}}(z)$ and pushing such through the transformations defining the unconditional flow $\theta = f(z)$ where $f = f_1 \circ \cdots \circ f_k$. As f can be evaluated very effectively, samples are generated very easily and the usage of a biased estimator does not pose a problem as K can be chosen to be large.

Depending on the complexity of the underlying system under investigation the optimization of Equation (8) can be challenging. While the dimensionality of the parameter space Θ is typically high, the system often allows for complementary solutions. This artifact has to be captured by the source distribution $q_\psi(\theta)$ in a similar fashion as the (approximated) posterior distribution does. We therefore propose to initialize the source distribution with the averaged posterior distribution over the targeted distribution Equation (10). As we've approached the problem of Bayesian inversion in Section 4.1 with neural density estimators, we have an efficient approximation of the posterior.

$$
\begin{aligned}
q_\psi^{(0)}(\theta) &= \int p(\theta \mid x) p^*(x) \; dx \\
&\approx \int q_\phi(\theta \mid x) p^*(x) \; dx
\end{aligned}
\tag{10}
$$

In a following step, $q_\psi^{(0)}(\theta)$ is "fine-tuned", optimizing Equation (8), in accordance with [22, 23]. Intuitively, the posterior distribution, conditioned on the observation, defines a distribution over the parameters, that are likely to have led to the observation. By averaging over all the targets, we're interested in, we obtain a distribution that is likely to produce the targeted observations. The steps taken are summarized in Algorithm 2. We note that [23] has shown that the average posterior alone is not sufficient to yield an accurate source distribution. However, the numerical experiment in Section 5 shows that the average posterior is a good starting point for the optimization.

Algorithm 2 Optimizing a source distribution to match a targeted distribution.

1: **Input:** Prior predictive distribution $p(x)$, targeted distribution $p^*(x)$, likelihood approximation $q_\omega(x \mid \theta)$, number of Monte-Carlo samples K
2: **Output:** parameterized distribution $q_\psi(\theta)$ s.t. $q(x) \approx p^*(x)$
3: Define divergence function $d(\cdot)$ on distributions
4: Define family of parametric density estimator: $\mathcal{Q} = \{q_\psi(\theta) \mid \psi \in \Psi\}$
5: Initialize source model to average posterior: $q_\psi(\theta) = \int q_\phi(\theta \mid x)p^*(x)\ dx$
6: Optimize divergence: $\psi^* = \underset{\psi \in \mathbb{M}}{\arg\min}\ d\left(p^*(x), q(x)\right)$
7: **Return** $q_\psi(\theta)$

4.3 Distillation of Point Estimates

Based on the source distribution obtained in Section 4.2, the mechanical systems are configured using point estimates. This is also the case for systems in passive safety. Therefore, point estimates have to be distilled from the source distribution.

While the most likely parameter would be a natural choice, it is of higher interest to select robust parameters. We therefore propose a selection process incorporating a weighting function $c(\theta) \in \mathbb{R}$. Such a weighting function allows to incorporate a notion of robustness as well as economical factors. While the former is a common for parameter selection, the latter is a useful option in the application. However, due to scientific reasons, we will focus solely on the incorporation of robustness here.

The robust point estimate θ^* is formulated as the maximizer of the weighted distribution

$$\theta^* = \underset{\theta \in \mathbb{R}^d}{\arg\max}\ q_\psi(\theta)c(\theta) \tag{11}$$

of the source distribution $q_\psi(\theta)$ with respect to the weighting function $c(\theta)$. This way, both the likelihood of that parameter and its stability are taken into account.

We model this robustness by computing the likelihood of re-observing an initially obtained safety score $x' \in \mathbb{R}$ via θ', for varying parameter vectors $\theta \in \mathbb{R}^d$. By varying the parameters around the generating parameter θ', we simulate the possibility that an unseen crash scenario would've required a different set of parameters that lead to the assumed safety score. Therefore, the weighting function $c(\theta)$ incorporates the estimated likelihood from Section 4.1:

$$c(\theta') = \int q_\omega(x' \mid \theta' + \epsilon) p_{\mathcal{N}(0,\sigma I)}(\epsilon)\ d\epsilon. \tag{12}$$

In the above formulation, $\mathcal{N}(0, \sigma I)$ denotes an isotropic normal distribution with zero mean and covariance σI. In the same way as in Equation (11), and in alignment with Bayesian Decision Making [10], the weighting function can be applied to the posterior estimate, in case a robust point estimate shall be derived with respect to a specific targeted observation x^*.

5 NUMERICAL EXAMPLE

In the assessment of vehicle safety for newly designed cars, different countries state their requirements in form of new car assessment protocols (NCAPs). These include different tests that have to be conducted to assess the car's overall safety. Based on the measurements obtained from a crash test, the protocols define how the safety rating is computed.

To demonstrate the application of the proposed IPA, we consider the US NCAP[4] as stochastic simulator to highlight the key features and the steps of the approach. The approach, outlined in Section 4, is used to estimate the required measurements to obtain during a full-frontal crash, s.t. the targeted distribution over the ratings is obtained.

Let $\pi(\theta)$ be a prior distribution over the parameter space $\Theta \subset \mathbb{R}^6$ and $\epsilon \sim \mathcal{N}(0, \sigma^2)$ be Gaussian white noise with variance σ^2. The stochastic US NCAP simulator is defined as follows:

$$x = \mathcal{M}(\theta) + \epsilon, \;\; \theta \sim p(\theta), \;\; x \in \mathbb{R}. \tag{13}$$

The function $\mathcal{M}$ is defined by the US NCAP protocol [3, 11] and maps into the reals. The white noise models aleatoric uncertainty of the underlying model. This particular example was chosen as it also allows for obvious complementary solutions.

A total of $N = 10^4$ samples are drawn from the joint distribution (Algorithm 1), constructing the dataset $\mathcal{D} = \{(\theta_i, x_i)\}_{i=1}^{10^4}$. For subsequent steps, the dataset $\mathcal{D}$ is split into training and validation sets, $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{validation}}$, with 90 and 10 percent of the full data, respectively. The obtained prior predictive distribution $p(x)$ is depicted as a gray curve in Figure 1.

The first step of the IPA is to define a targeted distribution that lies in the support of the prior predictive distribution. Here, we aim for a distribution that produces mostly 5-star ratings with a low risk of obtaining upper 4-star ratings. The targeted distribution was defined with a normal, centered at $\mu = 0.6$ and a standard deviation of $\sigma = 0.05$, i.e. $p^*(x) = \mathcal{N}(0.6, 0.05^2)$. The resulting distribution w.r.t. to $p(x)$ is depicted in Figure 1 as well.

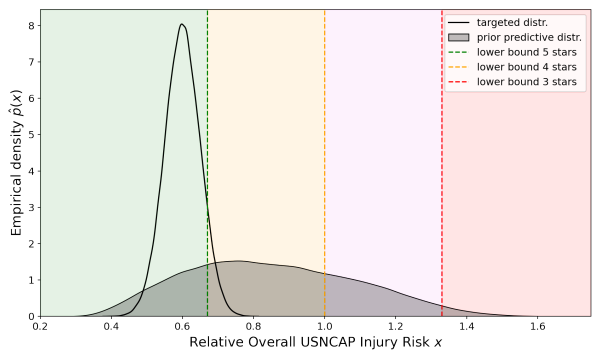


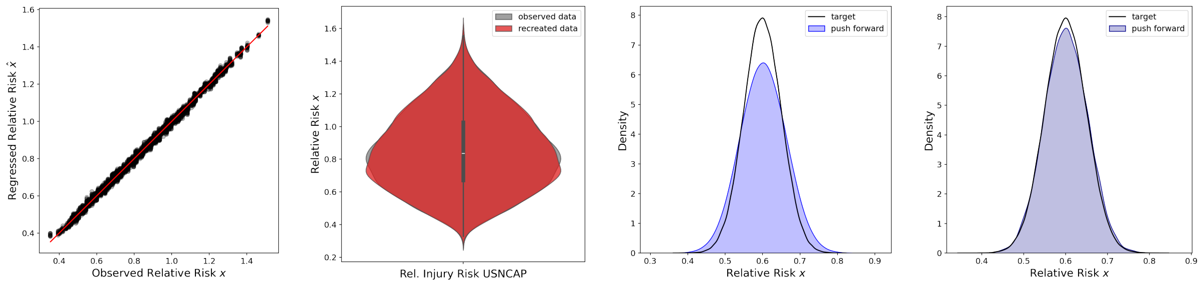
Figure 1: The targeted distribution $p^*(x)$ (black) is defined within the support of the prior predictive distribution (black). The colored areas of the plot account to the stars of the rating, relating to the relative risk obtained by the US NCAP rating.

In the second step, a Neural Spline Flow [7] $q_\omega(x \mid \theta)$ is trained to approximate the likelihood $p(x \mid \theta)$ using the training dataset $\mathcal{D}_{\text{train}}$ by minimizing the conditional negative log-likelihood (Equation (3)). The sequence of $L = 4$ transformations, with 128 units each, is conditioned on the six-dimensional parameter θ (see Section 2). The base $p_{\mathcal{Z}}$ is chosen to be standard normal distribution.

We assess the quality of the trained likelihood by regressing on and recreating the validation data. For the regression task, the likelihood is sampled 10 times per parameter θ of an input-output-pair $(\theta, x) \in \mathcal{D}_{\text{validation}}$. The prediction of the likelihood is given by the function g, computing the average of the 10 samples $g(\theta_i) = \frac{1}{10}\sum_{i=1}^{10} \hat{x}_i, \;\; \hat{x}_i \sim q_\omega(x \mid \theta_i)$.

With a coefficient of determination of $R^2 \approx 0.98$, the likelihood is able to capture the variation of the data generating process well (Figure 3a). Figure 3b compares the distribution of the

[4]The US NCAP is defined by the National Highway and Safety Administration (NHTSA) to assess the safety of new cars licensed in the United States of America.



(a) Regession task (b) Recreation task (c) Proposal (d) Fine-tuned

Figure 3: Figure 3a shows the actual-vs-predicted plot and Figure 3b shows the distribution of the observations in the validation data compared to the recreated observations. The model obtained a coefficient of determination of $R^2 \approx 0.98$ on the regression task and a value of C2ST ≈ 0.5634 on the Classifier-Two-Sample-Test. Figure 3c depicts the marginal likelihood obtained after initialization of the source model and Figure 3d after fine-tuning.

observations in the validation data to the recreated observations. The recreated data is obtained by sampling from the likelihood, conditioned on each parameter in the validation data, once. A five-fold Classifier-Two-Sample-Test [13], with an average score of C2ST ≈ 0.5634, is used to quantify the difference between the distributions. As the range of the test is $[0.5, 1.0]$, the model captures the data generating process well.

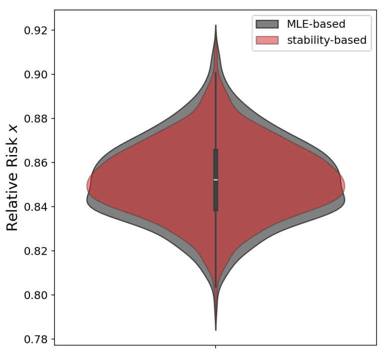


Figure 2: The samples of the perturbed likelihood under the MLE (gray) are compared to the samples obtained using the stability estimator in Equation (12) (red). The latter has less variance, indicating more stable results.

Utilizing the trained likelihood as a surrogate to the simulator in Equation (13), the third step of the approach is to minimize Equation (8) to obtain a source distribution, providing the desired targeted distribution $p^*(x)$. As source model, we choose an unconditional normalizing flow (see Section 2) with $L = 6$ transforms of 128 units each. Again, the Neural Spline Flow [7] architecture was used. According to Algorithm 2, we initialize the density model to the average posterior w.r.t. the targeted distribution. The parameters of the source model are optimized using the Adam optimizer with a decaying learning rate, starting fairly low at 10^{-8}. We depict the resulting marginal likelihoods after initialization and after fine-tuning in Figure 3c and Figure 3d. The majority of the distribution is well fit, however, the observed distribution has slightly too much mass in the tails, compared to the targeted distribution.

The final step of the approach is to obtain a stable point estimate θ^*. As mentioned in Section 4.3, we assume that real crash scenarios differ from such conducted in the lab. Therefore, the optimal parameterization for such crashes is slightly different. Still, we would like to rely on the obtained safety result. To model this scenario, we perturb the parameters and observe the likelihood of obtaining the initial observation. The standard deviation in Equation (12) was set to $\sigma = 0.1$, whilst working on standardized parameters θ. This simulates the shift between the point estimate and the actually fitting parameter for the scenario. We compare the obtained stable estimate $\hat{\theta}_{\text{stable}}$ with the Maximum-Likelihood-Estimate $\hat{\theta}_{\text{MLE}}$, which would be a typical choice. To compare the stability of the two estimates, we sample the original simulator for a total of 10^4 perturbed parameters. The resulting observations are depicted in Figure 2. We notice a narrower distribution of the observations under the stable estimate, indicating a more stable result.

6 CONCLUSION

In this study, we presented an approach to approximate a source distribution for a targeted distribution using neural density estimation. To avoid excessive queries to a costly simulator, we use the likelihood as a surrogate model. This efficient access to the likelihood is also used to obtain the average posterior and compute a stable point estimate under perturbation. We added an initialization step using the average posterior, which is beneficial for complex systems like highly non-linear simulators. For simpler simulators, direct optimization may suffice. Additionally, we used a weighting function to obtain stable point estimates, which can also include economic factors. Furthermore, the formulation of the optimization target allows for the inclusion of further constraints, which was not shown here. Sensitivity scores of the distribution could also inform the weighting function. As the distributions are provided as neural networks, sensitivity scores are easy to compute. Finally, the approach can theoretically be integrated into hierarchical problems, though specific applications remain to be demonstrated.

References

[1] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Information Science and Statistics. New York: Springer, 2006. 738 pp.

[2] Jan Bölts. “Advancing Methods and Applicability of Simulation-Based Inference in Neuroscience”. PhD thesis. Universität Tübingen, July 20, 2023.

[3] carhs. *SafetyCompanion 2024*. 2024.

[4] Kyle Cranmer, Johann Brehmer, and Gilles Louppe. “The Frontier of Simulation-Based Inference”. In: *Proceedings of the National Academy of Sciences* 117.48 (Dec. 2020), pp. 30055–30062.

[5] Miles D. Cranmer et al. “Modeling the Gaia Color-Magnitude Diagram with Bayesian Neural Flows to Constrain Distance Estimates”. In: (Aug. 21, 2019). Pre-published.

[6] Maximilian Dax et al. “Flow Matching for Scalable Simulation-Based Inference”. In: (May 26, 2023). Pre-published.

[7] Conor Durkan et al. “Neural Spline Flows”. In: *Advances in Neural Information Processing Systems*. Vol. 32. Curran Associates, Inc., 2019.

[8] Tomas Geffner, George Papamakarios, and Andriy Mnih. “Compositional Score Modeling for Simulation-Based Inference”. In: *Proceedings of the 40th International Conference on Machine Learning*. International Conference on Machine Learning. PMLR, July 3, 2023, pp. 11098–11116.

[9] Manuel Gloeckler et al. “All-in-One Simulation-Based Inference”. In: (May 31, 2024). Pre-published.

[10] Mila Gorecki, Jakob H. Macke, and Michael Deistler. “Amortized Bayesian Decision Making for Simulation-Based Models”. In: (Dec. 18, 2023). Pre-published.

[11] Jonas Siegfried Jehle. “Uncertainty Management Framework for Automotive Crash Applications”. Universität der Bundeswehr München, 2022.

[12] Yaron Lipman et al. “Flow Matching for Generative Modeling”. In: (Feb. 8, 2023). Pre-published.

[13] Jan-Matthis Lueckmann et al. “Benchmarking Simulation-Based Inference”. In: (Apr. 9, 2021). Pre-published.

[14] Joseph B. Nagel and Bruno Sudret. “A Unified Framework for Multilevel Uncertainty Quantification in Bayesian Inverse Problems”. In: *Probabilistic Engineering Mechanics* 43 (Jan. 2016), pp. 68–84.

[15] George Papamakarios. “Neural Density Estimation and Likelihood-free Inference”. In: (Oct. 29, 2019). Pre-published.

[16] George Papamakarios and Iain Murray. “Fast -Free Inference of Simulation Models with Bayesian Conditional Density Estimation”. Version 4. In: (2016).

[17] George Papamakarios, Theo Pavlakou, and Iain Murray. “Masked Autoregressive Flow for Density Estimation”. In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc., 2017.

[18] George Papamakarios et al. “Normalizing Flows for Probabilistic Modeling and Inference”. In: *arXiv e-prints* (Dec. 1, 2019).

[19] Louis Sharrock et al. “Sequential Neural Score Estimation: Likelihood-Free Inference with Conditional Score Based Diffusion Models”. In: (Nov. 21, 2022). Pre-published.

[20] Alvaro Tejero-Cantero et al. “Sbi: A Toolkit for Simulation-Based Inference”. In: *Journal of Open Source Software* 5.52 (Aug. 21, 2020), p. 2505.

[21] Alexander Tong et al. “Improving and Generalizing Flow-Based Generative Models with Minibatch Optimal Transport”. In: ICML 2023 Workshop on New Frontiers in Learning, Control, and Dynamical Systems. July 9, 2023.

[22] Maxime Vandegar et al. “Neural Empirical Bayes: Source Distribution Estimation and Its Applications to Simulation-Based Inference”. In: *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*. International Conference on Artificial Intelligence and Statistics. PMLR, Mar. 18, 2021, pp. 2107–2115.

[23] Julius Vetter et al. “Sourcerer: Sample-based Maximum Entropy Source Distribution Estimation”. In: (Feb. 12, 2024). Pre-published.

[24] Christina Winkler et al. “Learning Likelihoods with Conditional Normalizing Flows”. In: (Nov. 12, 2023). Pre-published.