Uncertainty-Aware Surrogate-based Amortized Bayesian Inference for Computationally Expensive Models

Stefania Scheurer [*†‡]
University of Stuttgart
Germany
stefania.scheurer@iws.uni-stuttgart.de

Philipp Reiser[*‡]
University of Stuttgart
Germany

Tim Brünnette[†]
University of Stuttgart
Germany

Wolfgang Nowak[†]
University of Stuttgart
Germany

Anneli Guthke[‡]
University of Stuttgart
Germany

Paul-Christian Bürkner [§]
TU Dortmund University
Germany

Abstract

Bayesian inference typically relies on a large number of model evaluations to estimate posterior distributions. Established methods like Markov Chain Monte Carlo (MCMC) and Amortized Bayesian Inference (ABI) can become computationally challenging. While ABI enables fast inference *after* training, generating sufficient training data still requires thousands of model simulations, which is infeasible for expensive models. Surrogate models offer a solution by providing *approximate* simulations at a lower computational cost, allowing the generation of large data sets for training. However, the introduced approximation errors and uncertainties can lead to overconfident posterior estimates. To address this, we propose Uncertainty-Aware Surrogate-based Amortized Bayesian Inference (UA-SABI) – a framework that combines surrogate modeling and ABI while explicitly quantifying and propagating surrogate uncertainties through the inference pipeline. Our experiments show that this approach enables reliable, fast, and repeated Bayesian inference for computationally expensive models, even under tight time constraints.

1 Introduction

Mathematical models are essential for simulating real-world processes, typically mapping parameters to observable data. Estimating these parameters from real-world observations, i.e., the inverse problem, is fundamental across scientific disciplines [e.g. 10, 45, 16, 9]. However, because models never perfectly capture reality and observed data are often sparse and imprecise, parameter estimation inherently involves uncertainty. Bayesian inference offers a systematic framework for estimating parameters while incorporating uncertainty in a statistically grounded manner [e.g. 12].

---

[*]Equal contribution
[†]Department of Stochastic Simulation and Safety Research for Hydrosystems
[‡]Cluster of Excellence SimTech
[§]Department of Statistics

Markov Chain Monte Carlo (MCMC) methods are widely used for Bayesian inference to generate high-quality samples from the posterior distribution given fixed observations [13]. However, MCMC is computationally expensive and slow, rendering it impractical in scenarios where near-instant inference is required [36]. Near-instant inference is, for example, necessary in adaptive robotic control or closed-loop medical devices, where parameters such as object mass or patient sensitivity must be estimated immediately to allow accurate prediction and safe action [e.g. 23, 44, 22]. Furthermore, each new set of observations requires restarting the entire process, further increasing costs when inference is needed for multiple datasets. Multiple inference runs may be needed for ongoing adaptation as conditions evolve or more data become available. Tracking the spread of a disease such as influenza or COVID-19 is one example for this. New case data continuously arrive, and separate datasets exist for different regions, requiring repeated Bayesian updates to infer transmission rates or reproduction numbers for each location [e.g. 31, 48]. MCMC also relies on a known likelihood function that can be evaluated either analytically or numerically.

Amortized Bayesian Inference (ABI), a deep-learning-based approach originating from simulation-based inference (SBI), addresses these limitations [4, 30, 21]. SBI methods are typically employed when evaluating the likelihood is infeasible, but simulations from the model are possible, allowing to learn a mapping from observed data to the posterior of the model parameters via simulated data. ABI in particular leverages generative neural networks to learn this mapping. Training data is generated through multiple model evaluations. Once trained, ABI enables near-instant posterior inference for new datasets, as the computational cost is incurred during training. Being likelihood-free makes ABI well suited for complex problems involving noisy or high-dimensional data, where evaluating the likelihood is infeasible or unreliable. However, the learned posterior only approximates the true posterior, and high accuracy requires well-designed network architectures and extensive training data.

Both MCMC and ABI face limitations when the simulation model is computationally expensive, since both require a large number of model evaluations. MCMC requires many likelihood evaluations per generated posterior sample; ABI requires extensive model simulations to generate sufficient amounts of training data. As a result, if computational time is limited, obtaining a good posterior becomes infeasible for both approaches.

Our goal is to enable ABI for computationally expensive models to benefit from the aforementioned advantages that ABI offers. To this end, surrogate models are the crucial tool, allowing us to reduce the computational cost of expensive simulations for training data generation. Surrogate models [42, 34], however, are only approximations (i.e., imperfect representations) of the reference simulation, and therefore not necessarily reliable. It is crucial to quantify the uncertainties associated with surrogate modeling, such as those arising from limited training data or any inherent inflexibility of the surrogate. These uncertainties then need to be propagated through the inference pipeline, ensuring consistent posterior estimation. Previous methods have integrated these uncertainties into MCMC methods with surrogate models [e.g. 20, 50, 35]. However, this requires many additional MCMC runs to incorporate the surrogate uncertainty, largely eliminating the computational advantages of surrogates.

We introduce Uncertainty-Aware Surrogate-based Amortized Bayesian Inference (UA-SABI), which enables ABI for computationally expensive models while accounting for the uncertainty inherent in surrogate approximations. By leveraging a surrogate, we eliminate the training error that arises from insufficient data during ABI training, as simulating from the surrogate is easy and fast. This comes at the cost of introducing a surrogate approximation error. However, unlike the ABI training error, the surrogate approximation error can be quantified and propagated, enabling reliable ABI for computationally expensive models. An illustration of the workflow, including training phases and inference, is given in Fig. 1. We demonstrate the workflow and substantiate the claimed advantages both theoretically and practically with a toy example and a real-world case study.
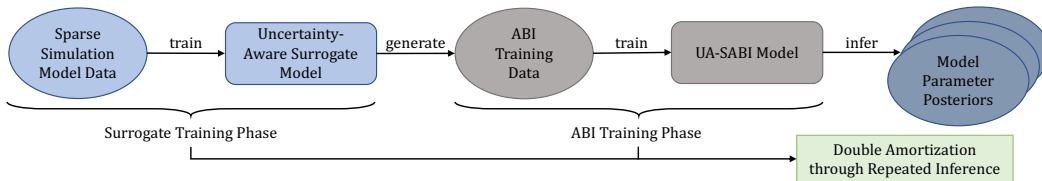

Figure 1: Illustrative workflow of UA-SABI training and inference.

2 Methods

2.1 Amortized Bayesian Inference (ABI)

This work focuses on using ABI to infer the parameters of computationally expensive simulation models. The typical ABI procedure [30] can be divided into a training phase and an inference phase. Training includes generating training data and training a neural posterior estimator (NPE) [28, 8, 14].

ABI Training Data Generation Training data is generated using a simulation model $M = M(\mathbf{x}, \boldsymbol{\omega})$, typically a realistic first-principles model. M depends on inputs $\mathbf{x}$ and parameters $\boldsymbol{\omega}$ and yields an output $\mathbf{y}$. Additionally, it can contain stochastic parts omitted in the following to simplify the notation. To learn the mapping from observations to posterior draws, a training set $D_B = \{(\mathbf{x}^{(i)}, \boldsymbol{\omega}^{(i)}, \mathbf{y}^{(i)})\}_{i=1}^{N_B}$ of N_B samples must first be generated. This can be performed by simulating N_B pairs of parameters, inputs and outputs from the simulation model:

$$\mathbf{y}^{(i)} = M(\mathbf{x}^{(i)}, \boldsymbol{\omega}^{(i)}) \quad \text{with} \quad (\mathbf{x}^{(i)}, \boldsymbol{\omega}^{(i)}) \sim p(\mathbf{x}, \boldsymbol{\omega}). \tag{1}$$

Neural Posterior Estimation After creating the training dataset, the NPE, including a summary network $S_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{y})$ and an inference network $I_{\boldsymbol{\varphi}}(\mathbf{s})$, must be trained. The goal of $S_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{y})$ is to embed (multiple) model inputs and outputs $(\mathbf{x}, \mathbf{y})$ of potentially variable length (e.g., a measurement series) in a fixed-length vector $\mathbf{s}$ that serves as input to $I_{\boldsymbol{\varphi}}(\mathbf{s})$. Then, $I_{\boldsymbol{\varphi}}(\mathbf{s})$ generates samples from the approximate posterior $q_{\boldsymbol{\varphi}}(\boldsymbol{\omega} \mid \mathbf{s})$. Here, $\boldsymbol{\theta}$ and $\boldsymbol{\varphi}$ are learnable coefficients in $S_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{y})$ and $I_{\boldsymbol{\varphi}}(\mathbf{s})$. In our case studies, we use coupling flows [29] as inference networks. Coupling flows have been empirically and theoretically shown to be expressive and allow for fast evaluation and sampling [7]. However, ABI is generally flexible in its choice for the inference network and works with any generative neural network [e.g. 1, 27, 6, 47, 38].

The summary and inference networks are trained jointly by minimizing the Kullback-Leibler (KL) divergence between the true posterior $p(\boldsymbol{\omega} \mid \mathbf{x}, \mathbf{y})$ and the approximate posterior $q_{\boldsymbol{\varphi}}(\boldsymbol{\omega} \mid \mathbf{s})$:

$$L(\boldsymbol{\varphi}, \boldsymbol{\theta}) = \mathbb{E}_{p(\mathbf{y})}[\mathrm{KL}(p(\boldsymbol{\omega} \mid \mathbf{x}, \mathbf{y}) \mid q_{\boldsymbol{\varphi}}(\boldsymbol{\omega} \mid \mathbf{s}))] \approx \sum_{i=1}^{N_B}(-\log q_{\boldsymbol{\varphi}}(\boldsymbol{\omega}^{(i)} \mid S_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}, \mathbf{y}^{(i)}))) \tag{2}$$

This expectation is approximated via the training data D_B obtained from the (potentially expensive) simulation model [30].

2.2 Uncertainty-Aware Surrogate Modeling

A surrogate model $\widetilde{M}$ aims to approximate the behavior of a simulation model M with negligible computational cost. The surrogate model is typically parametrized by learnable surrogate coefficients, denoted as $\mathbf{c}$. The simulation model output can then be approximated by

$$\mathbf{y} = M(\mathbf{x}, \boldsymbol{\omega}) \approx \widetilde{M}_{\mathbf{c}}(\mathbf{x}, \boldsymbol{\omega}) = \widetilde{\mathbf{y}}. \tag{3}$$

However, due to limited simulation data from M used for training $\widetilde{M}_{\mathbf{c}}$, the surrogate coefficients $\mathbf{c}$ can only be estimated with substantial epistemic uncertainty. For the same reason, the expressibility of $\widetilde{M}_{\mathbf{c}}$ must be restricted, i.e., the complexity of $\widetilde{M}_{\mathbf{c}}$ must be kept relatively low. This introduces an approximation error, denoted as ϵ, which captures the misspecification of the surrogate. In practice, one usually has to expect a non-negligible approximation error due to limitations in model capacity, optimization challenges, or regularization constraints. We assume the approximation error $\epsilon \sim p(\epsilon \mid \sigma)$ to follow a distribution parametrized by σ, leading to the error-adjusted surrogate output:

$$\widetilde{\mathbf{y}}_{\epsilon} = f(\widetilde{\mathbf{y}}, \epsilon) \quad \text{with} \quad \epsilon \sim p(\epsilon \mid \sigma). \tag{4}$$

This setup defines the surrogate likelihood

$$p(\widetilde{\mathbf{y}}_{\epsilon} \mid \widetilde{\mathbf{y}}, \sigma) = p(\widetilde{\mathbf{y}}_{\epsilon} \mid \mathbf{x}, \boldsymbol{\omega}, \mathbf{c}, \sigma), \tag{5}$$

that approximates the (usually unknown) likelihood of the simulation model.

Using **sparse** training data ($N_T \ll N_B$) obtained from the simulation model $D_T = \{(\mathbf{x}^{(i)}, \boldsymbol{\omega}^{(i)}, \mathbf{y}^{(i)})\}_{i=1}^{N_T}$, with $\mathbf{y}^{(i)} = M(\mathbf{x}^{(i)}, \boldsymbol{\omega}^{(i)})$ and $(\mathbf{x}^{(i)}, \boldsymbol{\omega}^{(i)})$ sampled from a given prior, we

infer the surrogate model coefficients $\mathbf{c}$ and σ via Bayesian inference. This enables capturing both the surrogate model's epistemic uncertainty and the irreducible approximation error. For this purpose, we consider the likelihood (5) and the prior $p(\mathbf{c}, \sigma)$. The joint posterior over the surrogate parameters and the approximation noise, given the training data, is then

$$p(\mathbf{c}, \sigma \mid D_T) \propto \prod_{i=1}^{N_T} p(\mathbf{y}^{(i)} \mid \mathbf{x}^{(i)}, \boldsymbol{\omega}^{(i)}, \mathbf{c}, \sigma)\, p(\mathbf{c}, \sigma), \qquad (6)$$

assuming an i.i.d sampled approximation error. The type of optimality of $\widetilde{M}$ is implied by the distributions of $\mathbf{x}$ and $\boldsymbol{\omega}$ in D_T and the shape of the error model $p(\epsilon \mid \sigma)$. Sampling-based algorithms such as MCMC can be used to estimate the surrogate posterior (6). For the considered surrogates, this is tractable since their likelihoods are fast to evaluate, and the surrogate posterior – compared to model parameter posteriors across datasets – is estimated only once. While this formulation fully captures surrogate uncertainty, we now temporarily simplify the setup by omitting this uncertainty to introduce an uncertainty-unaware baseline method – Surrogate-based ABI. This allows us to isolate and demonstrate the benefit of propagating surrogate uncertainty through to inference.

2.3 Surrogate-based ABI (SABI)

The quality of the NPE depends on a sufficient training budget D_B to ensure that $p(\boldsymbol{\omega} \mid \mathbf{x}, \mathbf{y})$ is well approximated via $q_{\boldsymbol{\varphi}}(\boldsymbol{\omega} \mid S_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{y}))$. However, for the expensive models considered here, generating sufficient ABI training data becomes infeasible. To mitigate this, we propose using a surrogate model of the expensive simulator to efficiently generate sufficient training data. Surrogate-based ABI (SABI) thus differs from standard ABI only in the generation of training data. Instead of using the expensive model, a (point) surrogate model with a point estimate $\overline{\mathbf{c}}$, e.g., the median of the posterior in Eq. (6) is used to generate the training data $D_B = \{(\mathbf{x}^{(i)}, \boldsymbol{\omega}^{(i)}, \widetilde{\mathbf{y}}^{(i)})\}_{i=1}^{N_B}$:

$$\widetilde{\mathbf{y}}^{(i)} = \widetilde{M}_{\overline{\mathbf{c}}}(\mathbf{x}^{(i)}, \boldsymbol{\omega}^{(i)}) \quad \text{with} \quad (\mathbf{x}^{(i)}, \boldsymbol{\omega}^{(i)}) \sim p(\mathbf{x}, \boldsymbol{\omega}). \qquad (7)$$

However, in general, $\widetilde{M}_{\mathbf{c}}$ is only an approximation of the simulation model M. Therefore, the SABI posterior trained with surrogate data $p(\boldsymbol{\omega} \mid \mathbf{x}, \widetilde{\mathbf{y}})$ will be systematically different from the true posterior $p(\boldsymbol{\omega} \mid \mathbf{x}, \mathbf{y})$, as no surrogate uncertainties or errors are considered in the ABI training process. Specifically, neither the epistemic uncertainty of $\widetilde{M}_{\mathbf{c}}$, nor its irreducible approximation error is propagated. As we show, this can lead to highly inaccurate posterior approximations.

2.4 Uncertainty-Aware Surrogate-based ABI (UA-SABI)

To ensure reliable inference, the uncertainty of the surrogate model needs to be propagated through the ABI training and inference procedure, as illustrated in Fig. 1. This results in our proposed Uncertainty-Aware Surrogate-based Amortized Bayesian Inference (UA-SABI) approach.

Instead of generating training data with the point surrogate model, UA-SABI utilizes the uncertainty-aware surrogate model:

$$\widetilde{\mathbf{y}}_{\epsilon}^{(i)} \sim p(\widetilde{\mathbf{y}}_{\epsilon} \mid \mathbf{x}^{(i)}, \boldsymbol{\omega}^{(i)}, \mathbf{c}^{(i)}, \sigma^{(i)}) \quad \text{with} \quad (\mathbf{x}^{(i)}, \boldsymbol{\omega}^{(i)}) \sim p(\mathbf{x}, \boldsymbol{\omega}), \, (\mathbf{c}^{(i)}, \sigma^{(i)}) \sim p(\mathbf{c}, \sigma \mid D_T).$$
$$(8)$$

To propagate uncertainty to the ABI training, we now additionally draw the surrogate coefficients $\mathbf{c}^{(i)}$ and the approximation error parameter $\sigma^{(i)}$ from the surrogate posterior (6). These coefficients are then used to evaluate the surrogate model and obtain $\widetilde{\mathbf{y}}^{(i)}$. The pair $(\widetilde{\mathbf{y}}^{(i)}, \sigma^{(i)})$ is subsequently used to generate a sample $\widetilde{\mathbf{y}}_{\epsilon}^{(i)}$ from the distribution of the error-adjusted surrogate output. We now approximate the KL divergence in Eq. (2) as:

$$L(\boldsymbol{\varphi}, \boldsymbol{\theta}) \approx \sum_{i=1}^{N_B} (-\log q_{\boldsymbol{\varphi}}(\boldsymbol{\omega}^{(i)} \mid S_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}, \widetilde{\mathbf{y}}_{\epsilon}^{(i)}))) \qquad (9)$$

A pseudocode summarizing the UA-SABI training procedure can be found in Appendix A.

3 Related Work

Jointly Amortized Neural Approximation (JANA) JANA [32] is an approach that integrates surrogate modeling with ABI. In addition to learning a neural posterior, it simultaneously learns a neural likelihood to enable marginal likelihood estimation, which is essential for tasks such as Bayesian model selection. However, its objective differs slightly, as it aims to learn both the likelihood, serving as a surrogate model, and the posterior jointly. Such a highly parameterized neural likelihood requires a sufficient amount of training data [11]. In contrast, we deal with a computationally expensive model where generating the necessary amount of training data for neural likelihood and/or posterior estimation is not feasible.

Low-Budget Simulation-based Inference with Bayesian Neural Networks Delaunoy et al. [5] also focus on the challenge of limited training data in ABI. They employ Bayesian Neural Networks (BNNs), which model uncertainty via distributions over network parameters. However, selecting an appropriate distribution and defining a meaningful prior for weights and biases remains speculative and can undermine posterior reliability, as noted in [5]. Moreover, BNNs still demand substantial training data and compute resources (~25,000 GPU hours in Delaunoy et al. [5]), thus only shifting computational resources from evaluating expensive simulation models to training BNNs.

Uncertainty-Aware Surrogate-Based Inference Reiser et al. [35] consider the same setting, replacing a computationally expensive model with an uncertainty-aware surrogate. However, their inference approach differs: they run MCMC separately for each observation set and for every surrogate sample. This results in a computational cost that scales with both the number of observations and the number of surrogate samples, making their method significantly more expensive. Their proposed method, E-Post, marginalizes over the surrogate coefficient posterior $p(\mathbf{c}, \sigma \mid D_T)$ in to obtain the posterior of the parameters of interest $\boldsymbol{\omega}$:

$$p(\boldsymbol{\omega} \mid \mathbf{y}) = \iint p(\boldsymbol{\omega} \mid \mathbf{y}, \mathbf{c}, \sigma)\, p(\mathbf{c}, \sigma \mid D_T)\, d\mathbf{c}\, d\sigma. \tag{10}$$

In practice, this integral is approximated via a Monte Carlo integration with a sufficient number of samples from the surrogate posterior $p(\mathbf{c}, \sigma \mid D_T)$. During inference when using UA-SABI, this explicit marginalization is not required, as the NPE is trained over the entire posterior space of both the outputs and the surrogate parameters. This results in a form of double amortization of the surrogate and ABI training costs through repeated inference. Since UA-SABI and E-Post share the same overall objective and framework, E-Post with MCMC inference serves as a benchmark for UA-SABI. A formal proof of asymptotic equivalence is given in Proposition 1 in Appendix B.

4 Case Studies

4.1 Objectives

We aim to validate the parameter posterior obtained with UA-SABI by assessing its quality and justifying the computational effort required to train the surrogate by the reduction in effort to train the ABI model. To show the advantage of using surrogate data instead of scarce simulation data and to demonstrate the importance of quantifying and propagating the uncertainty of the surrogate model, we compare ABI for the true model trained with full and low budget, SABI and UA-SABI (shown in blue in Fig. 2). To validate the posteriors produced by these methods, we also compare them with those obtained using the corresponding MCMC-based approach

Figure 2: Context of UA-SABI and its alternatives under tight computational constraints.

(shown in purple in Fig. 2). Further, to highlight the efficiency of our method, we compare the runtimes of UA-SABI and E-Post (shown in red in Fig. 2).

To assess posterior calibration, we generate multiple synthetic ground truth parameters and perform inference on the simulated datasets. For each dataset, we compute the rank of the ground truth within

the posterior samples and summarize the results using empirical cumulative distribution function (ECDF) difference calibration plots [43, 37, 24]. To compare the runtimes, we determined the number of observation sets required for the accumulated computational cost of repeated inference runs using E-Post to exceed the combined training time and repeated (quasi-instant) inference time of UA-SABI.

General Setup For both case studies, we use polynomial chaos expansion (PCE) for the surrogate model [46, 42, 25]. A deterministic PCE constructs the surrogate through a spectral projection onto orthogonal (w.r.t. $p(\mathbf{x}, \boldsymbol{\omega})$) polynomial basis functions, expressed as

$$\widetilde{M_{\mathbf{c}}}(\mathbf{x}, \boldsymbol{\omega}) = \sum_{d=0}^{D} c_d \cdot \Psi_d(\mathbf{x}, \boldsymbol{\omega}), \tag{11}$$

with $\boldsymbol{\Psi} = \{\Psi_d\}_{d=0}^{D}$ the multivariate orthogonal polynomial basis and $\mathbf{c} = \{c_d\}_{d=0}^{D}$ the corresponding coefficients. The number of expansion terms D is computed via the standard truncation scheme [42].

To construct a Bayesian PCE [39, 2], we define a prior distribution over the surrogate coefficients $p(\mathbf{c})$ and the approximation error parameter $p(\sigma)$. We choose a normal likelihood

$$p(\mathbf{y} \mid \mathbf{x}, \boldsymbol{\omega}, \mathbf{c}, \sigma) = \mathcal{N}(M(\mathbf{x}, \boldsymbol{\omega}) \mid \widetilde{M_{\mathbf{c}}}(\mathbf{x}, \boldsymbol{\omega}), \sigma). \tag{12}$$

Employing Hamiltonian Monte Carlo [15] yields samples from the surrogate posterior $p(\mathbf{c}, \sigma \mid D_T)$ [2]. The resulting Bayesian PCEs serve as our uncertainty-aware surrogate models.

4.2 Case Study 1: LogSin Model

We first evaluate our surrogate-based ABI approaches in a simple synthetic scenario: a simulation model with one parameter ω and a one-dimensional input x:

$$y = M(x, \omega) = \omega \log(x) + \sin(0.05x) + 0.01x + 1. \tag{13}$$

4.2.1 Setup

We generate surrogate training data by evaluating the simulation model M at the first 16 points of a two-dimensional Sobol sequence [40], scaled to $[1, 200]$ for x and $[0.6, 1.4]$ for ω. The resulting input-output pairs are used to train a Bayesian PCE. This introduces significant surrogate uncertainty and approximation error, as a perfect match to the simulation model is unattainable. The surrogate's posterior is sampled using HMC via *Stan* [3, 41]. We employ 4 chains, each with 1,000 warm-up and 250 sampling iterations, yielding 1,000 surrogate posterior samples in total to be propagated.

To perform inference on observation sets, we sampled 200 ground truth parameters ω^* from a prior $p(\omega) = \mathcal{N}(1, 0.2)$ and generated four $(\mathbf{x}, \mathbf{y})$ observations for each. We compare the inference results of our proposed surrogate-based ABI to surrogate-based MCMC methods. For ABI, we used an equivariant deep set summary network [49] and a coupling flow inference network. We employ online training, where newly sampled surrogate outputs are used at each iteration. All ABI models were trained for 100 epochs, with a batch size of 64 and 128 batches per epoch, using *BayesFlow* [33]. For given observations, ABI generates 4,000 posterior samples via the inference network. Surrogate-based MCMC (Point) draws 4,000 samples using 4 chains (1,000 warm-up and 1,000 sampling iterations). For E-Post, we run MCMC separately for each surrogate posterior draw (1,000 warm-up, 4 sampling iterations per run), a process that is embarrassingly parallelizable but still computationally expensive. Further computational details are given in Appendix C.

4.2.2 Impact of Uncertainty Propagation

First, we compare the performance of our two surrogate-based methods, SABI and UA-SABI, against each other, a low-budget ABI, trained with the same simulation model data, and a full-budget ABI trained on sufficient simulation data. Figure 3a shows corresponding recovery plots. They show the posterior median (circles) and median deviation (vertical lines) for any given ground truth value ω^*.

Figure 3a shows that a low-budget ABI model is unable to recover the ground truth values, particularly towards the boundaries of the parameter domain. Using a surrogate as part of SABI or UA-SABI, it becomes possible to generate sufficient ABI training data. Assigning the inter- and extrapolation tasks to the surrogate, specifically designed for this purpose, produces better outcomes than relying
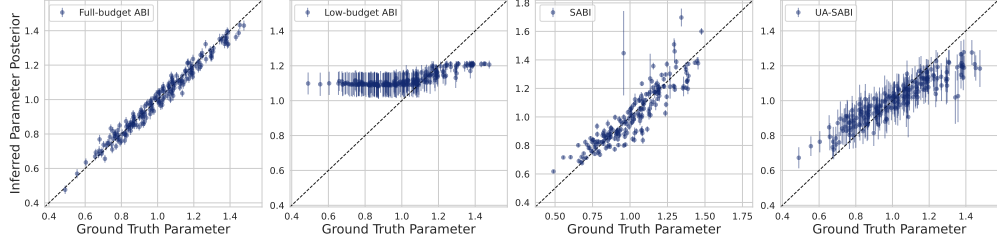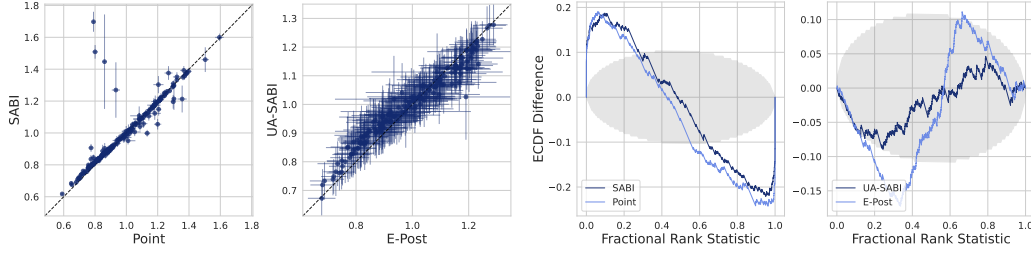
(a) Recovery plots.
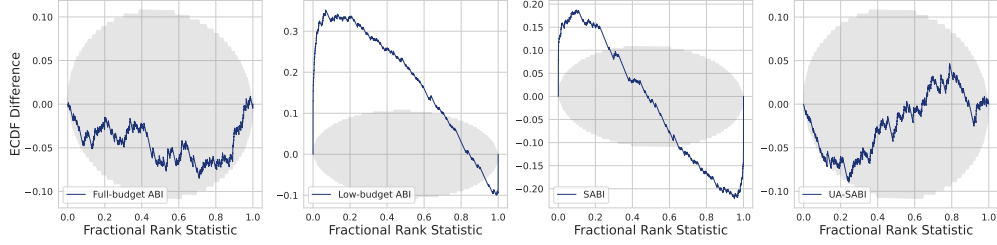
(b) ECDF difference plots. Only full-budget ABI and UA-SABI are well calibrated.

Figure 3: Recovery plots (top) and ECDF difference plots (bottom) plots for full-budget ABI, low-budget ABI, SABI, and UA-SABI over 200 ground truth samples. In the ECDF difference plots, empirical ranks are shown in blue, 95% confidence bands assuming calibration are shown in grey.

on the NPE to implicitly interpolate the posterior from scarce data. However, without propagating surrogate uncertainty (i.e., using SABI), estimated posterior uncertainty remains minimal and does not cover the true values, thus producing overconfident posteriors. Conversely, incorporating surrogate uncertainty via UA-SABI leads to an increase in posterior uncertainty, now better covering the true values. Indeed, the ECDF difference plots in Fig. 3b confirm that both low-budget ABI and SABI produce severely miscalibrated posteriors, while UA-SABI and full-budget ABI yield well-calibrated posteriors. However, achieving this calibration with full-budget ABI requires substantially more training data, highlighting the efficiency of UA-SABI under limited simulation budgets.

4.2.3 Validation of Parameter Posterior

We validate the correctness of the surrogate-based ABI approaches and benchmark their performance against a reference solution. To this end, we compare the two surrogate-based ABI methods, SABI and UA-SABI, against the corresponding surrogate-based MCMC methods, Point and E-Post. We plot the posterior medians (circles) and their median deviations (lines) for the ABI methods along with the corresponding MCMC results, as well as the ECDF differences for all the methods in Fig. 4.

(a) Recovery plots.

(b) ECDF difference plots.

Figure 4: Recovery plots (left) and ECDF difference plots (right) plots comparing ABI methods to corresponding MCMC methods over 200 ground truth samples. In the ECDF difference plots, empirical ranks are shown in blue, 95% confidence bands assuming calibration are shown in grey.

When comparing the ABI posteriors to the MCMC full reference solution, we observe that for all methods, the ABI results align closely with their MCMC counterparts. This validates the reliability

and confirms the correctness of the ABI methods. Comparing the ECDF differences between the ABI methods and their corresponding MCMC baselines shows that both Point and SABI produce similarly overconfident posteriors, while UA-SABI yields well-calibrated posteriors. In contrast, E-Post tends to produce even slightly underconfident posteriors.

4.2.4 Runtime Comparison

Next, we justify the computational effort required to train UA-SABI. Specifically, we aim to determine the break-even point – that is, the number of observation sets (i.e., measurement series) after which training an UA-SABI and performing (quasi-instant) inference becomes more efficient than repeatedly rerunning E-Post. Beyond that point, the one-time training cost of UA-SABI is amortized. All experiments were performed on a standard laptop equipped with an Intel Core i7-1185G7 CPU. For ABI, we consider both the upfront training phase and the inference time for the given sets as part of the total runtime. For E-Post, we measure the inference runtime while parallelized across 8 cores. Unlike for UA-SABI, E-Post runtimes depend on the number of cores used for parallelization.

In Fig. 6a, we show the runtime of UA-SABI and E-Post for $\{5, 10, 15, 20\}$ inference runs. We observe that UA-SABI's runtime is nearly constant for the number of inference runs, while E-Post scales linearly. Based on this comparison, UA-SABI is already justified after around 9 inference runs.

4.3 Case Study 2: Carbon Dioxide (CO_2) Storage Model

In the second case study, we test our method on a real-world problem. Specifically, we consider a CO_2 storage benchmark [19]. In this test case, a non-linear hyperbolic partial differential equation models the two-phase flow of CO_2 in brine. It describes CO_2 injection, plume migration, pressure build-up, and the influence of uncertain porous medium properties in a deep saline aquifer. Given the CO_2 saturation as measurements, our aim is to infer 3 parameters of interest: injection rate of CO_2 (IR), relative permeability degree in the fractional flux function (PM), and porosity (PR) of the formation. Five separate Bayesian PCE models with sparsity-inducing priors [2] are trained for five different instants at days $\{20, 40, 60, 80, 100\}$ and at a fixed location 30m from the injection well. A similar setup was studied in [26]. To avoid redundancy, we show results only for porosity.

4.3.1 Setup

In general, the setup for the CO_2 storage model follows the same structure as that described in Section 4.2.1. The priors of the input parameters are given as IR $\sim 6.4 \times 10^{-4} \times (1 + \text{Beta}(4, 2))$, PM $\sim 2 \times \text{Beta}(1.25, 1.25) + 2$, and PR $\sim \text{Beta}(2.4, 9)$ [19]. For PCE training, we considered 64 Sobol sequence evaluations scaled to the input parameter prior ranges and constructed the polynomial basis with arbitrary polynomial chaos (aPC) based on the priors [25, 2].

In contrast to case study 1, we used an offline ABI training set, where the training data is pre-generated from 10^4 parameter prior draws and fixed during the training process, allowing for direct comparison with a standard full-budget ABI trained on pre-generated simulation data. Further computational details are also given in Appendix C.

4.3.2 Validation of Parameter Posterior

We compare the inferred posterior samples obtained from a full-budget ABI ($N_B = 10^4$), a low-budget ABI ($N_B = 64$), SABI, and UA-SABI. Full-budget ABI was feasible due to the already available data [18, 26]. Therefore, recovery plots were chosen to validate the quality of our results relative to standard ABI. The porosity results are shown in Fig. 5.

Recovery plots in Fig. 5a again show that low-budget ABI struggles to recover the ground truth parameter, particularly near domain boundaries. SABI also produces poor estimates and, moreover, fails to capture uncertainty in the inferred parameter posteriors. This overconfidence results in miscalibrated posteriors, confirmed by the two corresponding ECDF difference plots in Fig. 5b.

In contrast, UA-SABI performs comparably to full-budget ABI while accounting for the additional uncertainty introduced by using a surrogate to generate training data. According to the ECDF difference plots, it produces a well-calibrated posterior for porosity. Recovery plots between ABI and MCMC for surrogate-based methods and corresponding ECDF difference plots are provided in Appendix D.
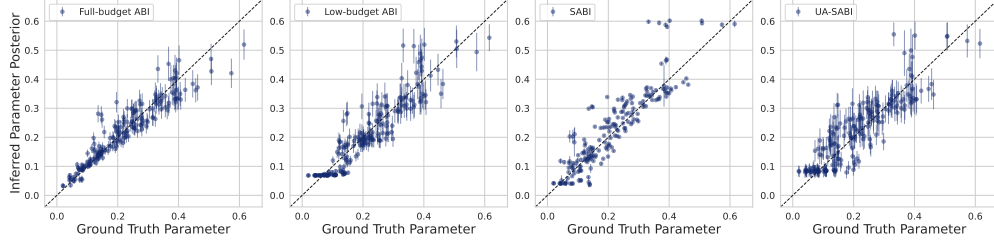
(a) Recovery plots of porosity.
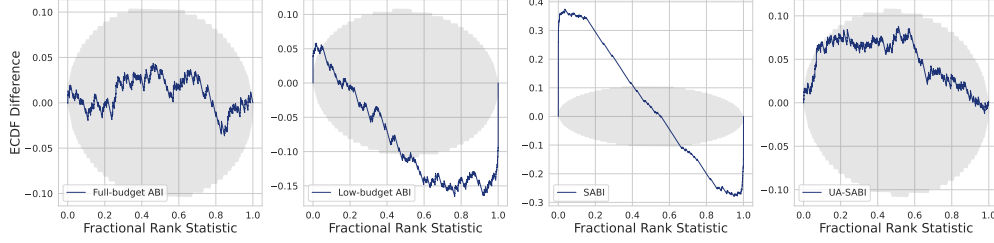
(b) ECDF difference plots of porosity. Only full-budget ABI and UA-SABI are well calibrated.

Figure 5: Recovery plots (top) and ECDF difference plots (bottom) for full-budget ABI, low-budget ABI, SABI, and UA-SABI over 200 ground truth samples. In the ECDF difference plots, empirical ranks are shown in blue, 95% confidence bands assuming calibration are shown in grey.

4.3.3 Runtime Comparison

Also, for the CO_2 storage model, we compare the runtimes, following the same approach as in Section 4.2.4. The experiments were run on a computing cluster with two AMD EPYC 7551 CPUs (totaling 64 physical cores) to speed up E-Post through parallelization.

Figure 6b presents the measured runtimes of UA-SABI and E-Post for $\{5, 6, 7, 8\}$ inference runs, whereby E-Post was parallelized on 16 cores. We observe a break-even point between 6 and 7 inference runs, indicating that UA-SABI becomes justified after 7 runs. Despite using more cores for E-Post, the training costs are amortized earlier for a more expensive model.
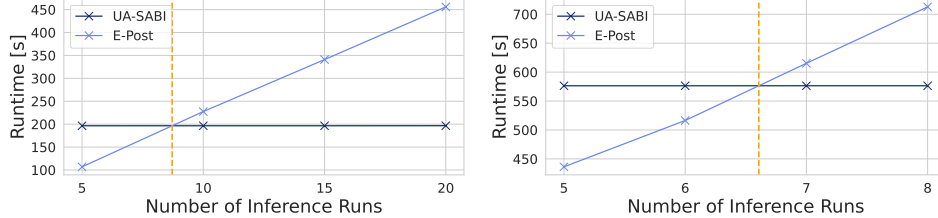
(a) Runtimes for LogSin for $\{5, 10, 15, 20\}$ runs. (b) Runtimes for CO_2 for $\{5, 6, 7, 8\}$ runs.

Figure 6: Comparison of runtimes for UA-SABI (training and inference) and E-Post (inference) with break-even point.

5 Summary and Outlook

In this work, we introduced Uncertainty-Aware Surrogate-based Amortized Bayesian Inference (UA-SABI) – a framework designed to enable efficient and reliable ABI for computationally expensive models. UA-SABI combines surrogate modeling and ABI while explicitly quantifying and propagating surrogate uncertainties through the inference process. This addresses a core limitation of existing approaches: while surrogate models can reduce the cost of generating training data for ABI, ignoring their approximation error often leads to overconfident and misleading posteriors. By

incorporating uncertainty awareness, UA-SABI enables better calibrated and reliable inference, even under tight computational constraints.

We validated UA-SABI in both a simple toy example and a real-world problem of modeling CO_2 storage, highlighting its ability to produce well-calibrated posterior estimates that match those obtained with MCMC-based methods. Our experiments demonstrate the importance of sufficient training data, as low-budget ABI produces erroneous posteriors. Also, they show the importance of uncertainty propagation when using a surrogate: SABI results in overconfident posteriors, whereas UA-SABI correctly reflects model uncertainty in its predictions. Moreover, we showed that the upfront computational effort of training UA-SABI is quickly offset in scenarios involving repeated inference, with amortization becoming beneficial already after a few inference runs.

Overall, UA-SABI offers a robust and efficient solution for more reliable and scalable inference in computationally demanding settings. Future work could include testing UA-SABI on models with higher-dimensional parameter spaces to further evaluate its scalability and performance.

Acknowledgments and Disclosure of Funding

References

[1] Lynton Ardizzone, Carsten Lüth, Jakob Kruse, Carsten Rother, and Ullrich Köthe. Guided Image Generation with Conditional Invertible Neural Networks, 2019. URL https://arxiv.org/abs/1907.02392.

[2] Paul-Christian Bürkner, Ilja Kröker, Sergey Oladyshkin, and Wolfgang Nowak. A Fully Bayesian Sparse Polynomial Chaos Expansion Approach with Joint Priors on the Coefficients and Global Selection of Terms. *Journal of Computational Physics*, 488:112210, 2023. doi: 10.1016/j.jcp.2023.112210.

[3] Bob Carpenter, Andrew Gelman, Matthew D. Hoffman, Daniel Lee, Ben Goodrich, Michael Betancourt, Marcus Brubaker, Jiqiang Guo, Peter Li, and Allen Riddell. Stan: A Probabilistic Programming Language. *Journal of Statistical Software*, 76(1):1—-32, 2017. doi: 10.18637/jss.v076.i01.

[4] Kyle Cranmer, Johann Brehmer, and Gilles Louppe. The Frontier of Simulation-based Inference. *Proceedings of the National Academy of Sciences*, 117(48):30055–30062, 2020. doi: 10.1073/pnas.1912789117.

[5] Arnaud Delaunoy, Maxence de la Brassinne Bonardeaux, Siddharth Mishra-Sharma, and Gilles Louppe. Low-Budget Simulation-based Inference with Bayesian Neural Networks, 2024. URL https://arxiv.org/abs/2408.15136.

[6] Alexander Denker, Maximilian Schmidt, Johannes Leuschner, and Peter Maass. Conditional Invertible Neural Networks for Medical Imaging. *Journal of Imaging*, 7(11):243, 2021. doi: 10.3390/jimaging7110243.

[7] Felix Draxler, Stefan Wahl, Christoph Schnörr, and Ullrich Köthe. On the Universality of Volume-Preserving and Coupling-Based Normalizing Flows, 2025. URL https://arxiv.org/abs/2402.06578.

[8] Conor Durkan, Artur Bekasov, Iain Murray, and George Papamakarios. Neural Spline Flows. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates,

Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/7ac71d433f282034e088473244df8c02-Paper.pdf.

[9] Aaron M. Ellison. Bayesian Inference in Ecology. *Ecology Letters*, 7(6):509–520, 2004. doi: 10.1111/j.1461-0248.2004.00603.x.

[10] Alexander Etz and Joachim Vandekerckhove. Introduction to Bayesian Inference for Psychology. *Psychonomic Bulletin & Review*, 25:5–34, 2018. doi: 10.3758/s13423-017-1262-3.

[11] David T. Frazier, Ryan P. Kelly, Christopher C. Drovandi, and David J. Warne. The Statistical Accuracy of Neural Posterior and Likelihood Estimation, 2024. URL https://arxiv.org/abs/2411.12068.

[12] Andrew Gelman, John B. Carlin, Hal S. Stern, and Donald B. Rubin. *Bayesian Data Analysis*. Chapman & Hall/CRC, 1995. ISBN 9780429258411.

[13] Walter R. Gilks, Sylvia Richardson, and David Spiegelhalter. *Markov Chain Monte Carlo in Practice*. Chapman & Hall/CRC, 1995. ISBN 9780429170232.

[14] David S. Greenberg, Marcel Nonnenmacher, and Jakob Macke. Automatic Posterior Transformation for Likelihood-free Inference. In K. Chaudhuri and R. Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2404–2414. PMLR, 2019. URL https://proceedings.mlr.press/v97/greenberg19a.html.

[15] Matthew D. Hoffman and Andrew Gelman. The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research*, 15(47):1593–1623, 2014. URL https://www.jmlr.org/papers/volume15/hoffman14a/hoffman14a.pdf.

[16] John P. Hülsenbeck, Fredrik Ronquist, Rasmus Nielsen, and Jonathan P. Bollback. Bayesian Inference of Phylogeny and its Impact on Evolutionary Biology. *Science*, 294(5550):2310–2314, 2001. doi: 10.1126/science.1065889.

[17] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization, 2017. URL https://arxiv.org/abs/1412.6980.

[18] Markus Köppel, Fabian Franzelin, Ilja Kröker, Sergey Oladyshkin, Dominik Wittwar, Gabriele Santin, Andrea Barth, Bernard Haasdonk, Wolfang Nowak, Dirk Pflüger, and Christian Rohde. Datasets and executables of data-driven uncertainty quantification benchmark in carbon dioxide storage, 2017.

[19] Markus Köppel, Fabian Franzelin, Ilja Kröker, Sergey Oladyshkin, Gabriele Santin, Dominik Wittwar, Andrea Barth, Bernard Haasdonk, Wolfgang Nowak, Dirk Pflüger, and Christian Rohde. Comparison of Data-Driven Uncertainty Quantification Methods for a Carbon Dioxide Storage Benchmark Scenario. *Computational Geosciences*, 23:339–354, 2019. doi: 10.1007/s10596-018-9785-x.

[20] Jan-Matthis Lückmann, Giacomo Bassetto, Theofanis Karaletsos, and Jakob H. Macke. Likelihood-free Inference with Emulator Networks. In F. J. R. Ruiz, C. Zhang, D. Liang, and T. D. Bui, editors, *Proceedings of The 1st Symposium on Advances in Approximate Bayesian Inference*, volume 96 of *Proceedings of Machine Learning Research*, pages 32–53. PMLR, 2018. URL https://proceedings.mlr.press/v96/lueckmann19a.html.

[21] Jan-Matthis Lückmann, Jan Bölts, David S. Greenberg, Pedro Goncalves, and Jakob Macke. Benchmarking Simulation-based Inference. In A. Banerjee and K. Fukumizu, editors, *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pages 343–351. PMLR, 2021. URL https://proceedings.mlr.press/v130/lueckmann21a.html.

[22] Nicolò Malagutti, Grace McGinness, and Dilip A. Nithyanandam. Real-Time Personalised Pharmacokinetic-Pharmacodynamic Modelling in Propofol Anesthesia Through Bayesian Inference. In *2023 45th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pages 1–6, 2023. doi: 10.1109/EMBC40787.2023.10339991.

[23] Norman Marlier. *Simulation-based Inference for Robotic Grasping*. PhD thesis, Universite de Liege (Belgium), 2024.

[24] Martin Modrák, Angie H. Moon, Shinyoung Kim, Paul-Christian Bürkner, Niko Huurre, Kateřina Faltejsková, Andrew Gelman, and Aki Vehtari. Simulation-based Calibration Checking for Bayesian Computation: The Choice of Test Quantities Shapes Sensitivity. *Bayesian Analysis*, pages 1–28, 2023. doi: 10.1214/23-BA1404.

[25] Sergey Oladyshkin and Wolfgang Nowak. Data-Driven Uncertainty Quantification Using the Arbitrary Polynomial Chaos Expansion. *Reliability Engineering & System Safety*, 106:179–190, 2012. doi: 10.1016/j.ress.2012.05.002.

[26] Sergey Oladyshkin, Farid Mohammadi, Ilja Kröker, and Wolfgang Nowak. Bayesian[3] Active Learning for the Gaussian Process Emulator using Information Theory. *Entropy*, 22(8):890, 2020. doi: 10.3390/e22080890.

[27] Govinda A. Padmanabha and Nicholas Zabaras. Solving Inverse Problems Using Conditional Invertible Neural Networks. *Journal of Computational Physics*, 433:110194, 2021. doi: 10.1016/j.jcp.2021.110194.

[28] George Papamakarios and Iain Murray. Fast ϵ -free Inference of Simulation Models with Bayesian Conditional Density Estimation. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL https://proceedings.neurips.cc/paper_files/paper/2016/file/6aca97005c68f1206823815f66102863-Paper.pdf.

[29] George Papamakarios, Eric Nalisnick, Danilo J Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing Flows for Probabilistic Modeling and Inference. *Journal of Machine Learning Research*, 22(57):1–64, 2021. URL http://jmlr.org/papers/v22/19-1028.html.

[30] Stefan T. Radev, Ulf K. Mertens, Andreas Voss, Lynton Ardizzone, and Ullrich Köthe. BayesFlow: Learning Complex Stochastic Models with Invertible Neural Networks. *IEEE Transactions on Neural Networks and Learning Systems*, 33(4):1452–1466, 2020. doi: 10.1109/TNNLS.2020.3042395.

[31] Stefan T. Radev, Frederik Graw, Simiao Chen, Nico T. Mutters, Vanessa M. Eichel, Till Bärnighausen, and Ullrich Köthe. OutbreakFlow: Model-based Bayesian Inference of Disease Outbreak Dynamics with Invertible Neural Networks and its Application to the COVID-19 Pandemics in Germany. *PLoS Computational Biology*, 17(10):e1009472, 2021. doi: 10.1371/journal.pcbi.1009472.

[32] Stefan T. Radev, Marvin Schmitt, Valentin Pratz, Umberto Picchini, Ullrich Köthe, and Paul-Christian Bürkner. JANA: Jointly Amortized Neural Approximation of Complex Bayesian Models. In R. J. Evans and I. Shpitser, editors, *Proceedings of the Thirty-Ninth Conference on Uncertainty in Artificial Intelligence*, volume 216 of *Proceedings of Machine Learning Research*, pages 1695–1706. PMLR, 2023. URL https://proceedings.mlr.press/v216/radev23a.html.

[33] Stefan T. Radev, Marvin Schmitt, Lukas Schumacher, Lasse Elsemüller, Valentin Pratz, Yannik Schälte, Ullrich Köthe, and Paul-Christian Bürkner. BayesFlow: Amortized Bayesian Workflows with Neural Networks. *Journal of Open Source Software*, 8(89):5702, 2023. doi: 10.21105/joss.05702.

[34] Carl E. Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2006. ISBN 026218253X.

[35] Philipp Reiser, Javier E. Aguilar, Anneli Guthke, and Paul-Christian Bürkner. Uncertainty Quantification and Propagation in Surrogate-based Bayesian Inference. *Statistics and Computing*, 35 (3):66, 2025. doi: 10.1007/s11222-025-10597-8.

[36] Christian P. Robert, Víctor Elvira, Nick Tawn, and Changye Wu. Accelerating MCMC Algorithms. *Wiley Interdisciplinary Reviews: Computational Statistics*, 10(5):e1435, 2018. doi: 10.1002/wics.1435.

[37] Teemu Säilynoja, Paul-Christian Bürkner, and Aki Vehtari. Graphical Test for Discrete Uniformity and Its Applications in Goodness-of-Fit Evaluation and Multiple Sample Comparison. *Statistics and Computing*, 32(32), 2022. doi: 10.1007/s11222-022-10090-6.

[38] Marvin Schmitt, Valentin Pratz, Ullrich Köthe, Paul-Christian Bürkner, and Stefan T. Radev. Consistency Models for Scalable and Fast Simulation-based Inference. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 126908–126945. Curran Associates, Inc., 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/e58026e2b2929108e1bd24cbfa1c8e4b-Paper-Conference.pdf.

[39] Qian Shao, Anis Younes, Marwan Fahs, and Thierry A. Mara. Bayesian Sparse Polynomial Chaos Expansion for Global Sensitivity Analysis. *Computer Methods in Applied Mechanics and Engineering*, 318:474–496, 2017. doi: 10.1016/j.cma.2017.01.033.

[40] Ilya M. Sobol'. On the Distribution of Points in a Cube and the Approximate Evaluation of Integrals. *USSR Computational Mathematics and Mathematical Physics*, 7(4):86–112, 1967. doi: 10.1016/0041-5553(67)90144-9.

[41] Stan Development Team. Stan Modeling Language Users Guide and Reference Manual, 2024. URL http://mc-stan.org/. Version 2.36.

[42] Bruno Sudret. Global Sensitivity Analysis Using Polynomial Chaos Expansions. *Reliability Engineering & System Safety*, 93(7):964–979, 2008. doi: 10.1016/j.ress.2007.04.002.

[43] Sean Talts, Michael Betancourt, Daniel Simpson, Aki Vehtari, and Andrew Gelman. Validating Bayesian Inference Algorithms with Simulation-based Calibration, 2020. URL https://arxiv.org/abs/1804.06788.

[44] Shahin Tasoujian, Saeed Salavati, Matthew A. Franchek, and Karolos M. Grigoriadis. Robust Delay-Dependent LPV Synthesis for Blood Pressure Control with Real-Time Bayesian Parameter Estimation. *IET Control Theory & Applications*, 14(10):1334–1345, 2020. doi: 10.1049/iet-cta.2019.0651.

[45] Udo Von Toussaint. Bayesian Inference in Physics. *Reviews of Modern Physics*, 83(3):943–999, 2011. doi: 10.1103/RevModPhys.83.943.

[46] Norbert Wiener. The homogeneous chaos. *American Journal of Mathematics*, 60(4):897–936, 1938. doi: 10.2307/2371268.

[47] Jonas Wildberger, Maximilian Dax, Simon Buchholz, Stephen Green, Jakob H. Macke, and Bernhard Schölkopf. Flow Matching for Scalable Simulation-based Inference. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 16837–16864. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/3663ae53ec078860bb0b9c6606e092a0-Paper-Conference.pdf.

[48] Wan Yang, Marc Lipsitch, and Jeffrey Shaman. Inference of Seasonal and Pandemic Influenza Transmission Dynamics. *Proceedings of the National Academy of Sciences*, 112(9):2723–2728, 2015. doi: 10.1073/pnas.1415012112.

[49] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R. Salakhutdinov, and Alexander J. Smola. Deep Sets. 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. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/f22e4747da1aa27e363d86d40ff442fe-Paper.pdf.

[50] Jiangjiang Zhang, Qiang Zheng, Dingjiang Chen, Laosheng Wu, and Lingzao Zeng. Surrogate-based Bayesian Inverse Modeling of the Hydrological System: An Adaptive Approach Considering Surrogate Approximation Error. *Water Resources Research*, 56(1):e2019WR025721, 2020. doi: 10.1029/2019WR025721.

A Pseudocode

Algorithm 1 Training of Surrogate and UA-SABI

    Surrogate Training Phase

Require: Simulation model $M(\mathbf{x}, \boldsymbol{\omega})$, prior $p(\mathbf{c}, \sigma)$, likelihood $p(\mathbf{y} \mid \mathbf{x}, \boldsymbol{\omega}, \mathbf{c}, \sigma)$

Ensure: Posterior $p(\mathbf{c}, \sigma \mid D_T)$

  1: Choose inputs and parameters $\{(\mathbf{x}^{(i)}, \boldsymbol{\omega}^{(i)})\}_{i=1}^{N_T}$

  2: **for** $i = 1, \ldots, N_T$ **do**

  3:     Evaluate simulation model $\mathbf{y}^{(i)} = M(\mathbf{x}^{(i)}, \boldsymbol{\omega}^{(i)})$

  4: **end for**

  5: Construct training dataset $D_T = \{(\mathbf{x}^{(i)}, \boldsymbol{\omega}^{(i)}, \mathbf{y}^{(i)})\}_{i=1}^{N_T}$

  6: Perform Bayesian inference for surrogate parameters

$$p(\mathbf{c}, \sigma \mid D_T) \propto \prod_{i=1}^{N_T} p(\mathbf{y}^{(i)} \mid \mathbf{x}^{(i)}, \boldsymbol{\omega}^{(i)}, \mathbf{c}, \sigma)\, p(\mathbf{c}, \sigma)$$

    UA-SABI Training Phase

Require: Prior $p(\mathbf{x}, \boldsymbol{\omega})$, posterior $p(\mathbf{c}, \sigma \mid D_T)$

Ensure: Trained summary and inference network parameters $\widehat{\boldsymbol{\theta}}, \widehat{\boldsymbol{\varphi}}$

  7: **for** each epoch **do**

  8:     **for** $i = 1, \ldots, N_B$ **do**

  9:         Sample inputs and parameters from prior $(\mathbf{x}^{(i)}, \boldsymbol{\omega}^{(i)}) \sim p(\mathbf{x}, \boldsymbol{\omega})$

10:         Sample surrogate parameters from posterior $(\mathbf{c}^{(i)}, \sigma^{(i)}) \sim p(\mathbf{c}, \sigma \mid D_T)$

11:         Evaluate surrogate $\widetilde{\mathbf{y}}^{(i)} = \widetilde{M}_{\mathbf{c}^{(i)}}(\mathbf{x}^{(i)}, \boldsymbol{\omega}^{(i)})$

12:         Sample corrected surrogate output $\widetilde{\mathbf{y}}_\epsilon^{(i)} \sim p(\widetilde{\mathbf{y}}_\epsilon \mid \widetilde{\mathbf{y}}^{(i)}, \sigma^{(i)})$

13:         Pass $(\mathbf{x}^{(i)}, \widetilde{\mathbf{y}}_\epsilon^{(i)})$ through summary network $\mathbf{s}^{(i)} = S_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}, \widetilde{\mathbf{y}}_\epsilon^{(i)})$

14:         Pass $(\boldsymbol{\omega}^{(i)}, \mathbf{s}^{(i)})$ through inference network $q_{\boldsymbol{\varphi}}(\boldsymbol{\omega}^{(i)} \mid \mathbf{s}^{(i)})$

15:         Compute loss from Eq. (2) and update network parameters $\boldsymbol{\theta}, \boldsymbol{\varphi}$

16:     **end for**

17: **end for**

B Proofs

Proposition 1. *The posterior targeted by UA-SABI is the same as the E-Post posterior.*

Proof. In UA-SABI, as per Eq. (8), we sample from the joint distribution

$$p(\boldsymbol{\omega}, \mathbf{y}, \mathbf{c}, \sigma \mid D_T) = p(\mathbf{y} \mid \boldsymbol{\omega}, \mathbf{c}, \sigma)\, p(\boldsymbol{\omega})\, p(\mathbf{c}, \sigma \mid D_T) \tag{14}$$

to train the conditional neural density estimator $q(\boldsymbol{\omega} \mid \mathbf{y})$. We only condition on the data $\mathbf{y}$ and treat the surrogate parameters $(\mathbf{c}, \sigma)$ as nuisance parameters ignored by $q(\boldsymbol{\omega} \mid \mathbf{y})$. This means that we (implicitly) integrate over the distribution of $(\mathbf{c}, \sigma)$. Hence, UA-SABI targets the following posterior:

$$p_{\mathrm{UA-SABI}}(\boldsymbol{\omega} \mid \mathbf{y}, D_T) \propto \iint p(\mathbf{y} \mid \boldsymbol{\omega}, \mathbf{c}, \sigma)\, p(\boldsymbol{\omega})\, p(\mathbf{c}, \sigma \mid D_T)\, d\mathbf{c}\, d\sigma \tag{15}$$

On the other hand, E-Post targets the following posterior, as per Eq. (10):

$$p_{\mathrm{E-Post}}(\boldsymbol{\omega} \mid \mathbf{y}, D_T) = \iint p(\boldsymbol{\omega} \mid \mathbf{y}, \mathbf{c}, \sigma)\, p(\mathbf{c}, \sigma \mid D_T)\, d\mathbf{c}\, d\sigma \tag{16}$$

$$\propto \iint p(\mathbf{y} \mid \boldsymbol{\omega}, \mathbf{c}, \sigma)\, p(\boldsymbol{\omega})\, p(\mathbf{c}, \sigma \mid D_T)\, d\mathbf{c}\, d\sigma, \tag{17}$$

which shows $p_{\mathrm{UA-SABI}}(\boldsymbol{\omega} \mid \mathbf{y}, D_T) = p_{\mathrm{E-Post}}(\boldsymbol{\omega} \mid \mathbf{y}, D_T)$. $\qquad\square$

C Additional Case Study Details

C.1 Surrogate Models

In case study 1, we train a Bayesian PCE with 2-dimensional Legendre polynomials and a maximum total degree of 3, resulting in $D = 10$ polynomials. We set a normal prior for the surrogate coefficients, $p(\mathbf{c}) = \mathcal{N}(0, 5)$, and a half-normal prior for the approximation error parameter, $p(\sigma) = \text{Half-}\mathcal{N}(0.5)$.

In case study 2, we consider a Bayesian PCE with 3-dimensional aPC polynomials (see Section 4.3) and a maximum total degree of 3, resulting in $D = 19$ polynomials. Following Bürkner et al. [2], we place a sparsity-inducing R2D2 prior on the surrogate coefficients $\mathbf{c}$ with $R^2 \sim \text{Beta}(0.5, 2)$. For the approximation error we set the prior as $p(\sigma) = \text{Half-}\mathcal{N}(0.1)$.

C.2 Neural Posterior Estimation

For both case studies, we used the same NPE setup. The summary network $S_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{y})$ is a *DeepSet* [49] composed of two hidden layers, each containing 10 neurons. It outputs a 10-dimensional summary vector. For the inference network $I_{\boldsymbol{\varphi}}(\mathbf{s})$ we employ a coupling flow as implemented in Radev et al. [33]. The training process employed a cosine learning rate scheduler with an initial learning rate of 5×10^{-4} and a minimum learning rate fraction of $\alpha = 10^{-6}$. The scheduler operated over a total of 12,800 steps, corresponding to 128 batches per epoch over 100 epochs. All NPE models were trained using the Adam optimizer [17] for 100 epochs.

D Additional Results for Case Study 2: CO_2 Storage Model

This section shows the recovery and ECDF difference plots for porosity of the CO_2 storage model shown for SABI vs. Point and UA-SABI vs. E-Post.

In Fig. 7 we show that SABI and Point as well as UA-SABI and E-Post yield similar results in both revovery and calibration. The notably wide intervals observed for Point, especially compared to SABI in the recovery plots suggest convergence problems in the MCMC, likely due to the low standard deviation in the likelihood. These convergence problems may explain the inconsistencies between SABI and Point. In comparison, their uncertainty-aware counterparts UA-SABI and E-Post produce highly similar estimates.
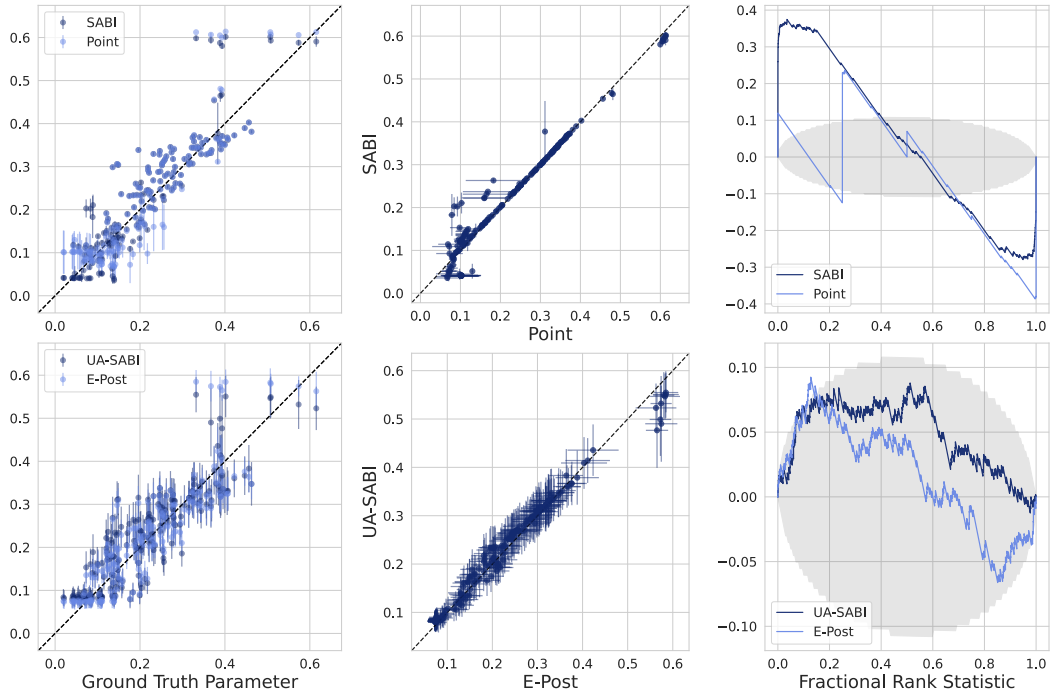
Figure 7: Recovery plots comparing to ground truth and MCMC full reference solution, ECDF difference plots (from left to right) for CO_2 storage model porosity parameter for 200 ground truth samples: SABI vs. Point (top) and UA-SABI vs. E-Post (bottom). For ECDF difference plots, empirical ranks are shown in blue, 95% confidence bands assuming calibration are shown in grey.