Statistics and Computing (2025) 35:66
https://doi.org/10.1007/s11222-025-10597-8

ORIGINAL PAPER

Uncertainty quantification and propagation in surrogate-based Bayesian inference

Philipp Reiser[1] · Javier Enrique Aguilar[1,2] · Anneli Guthke[1] · Paul-Christian Bürkner[1,2]

Received: 9 April 2024 / Accepted: 23 February 2025

Abstract

Surrogate models are statistical or conceptual approximations for more complex simulation models. In this context, it is crucial to propagate the uncertainty induced by limited simulation budget and surrogate approximation error to predictions, inference, and subsequent decision-relevant quantities. However, quantifying and then propagating the uncertainty of surrogates is usually limited to special analytic cases or is otherwise computationally very expensive. In this paper, we propose a framework enabling a scalable, Bayesian approach to surrogate modeling with thorough uncertainty quantification, propagation, and validation. Specifically, we present three methods for Bayesian inference with surrogate models given measurement data. This is a task where the propagation of surrogate uncertainty is especially relevant, because failing to account for it may lead to biased and/or overconfident estimates of the parameters of interest. We showcase our approach in three detailed case studies for linear and nonlinear real-world modeling scenarios. Uncertainty propagation in surrogate models enables more reliable and safe approximation of expensive simulators and will therefore be useful in various fields of applications.

Keywords Surrogate modeling · Uncertainty quantification · Uncertainty propagation · Bayesian inference

1 Introduction

Simulations of complex phenomena are crucial in the natural sciences and engineering for different scenarios, e.g., for gaining system understanding, prediction of future scenarios, risk assessment, or system design. However, often they are based on complex ordinary differential equations or partial differential equations which may not have closed-form solutions and may have to be solved using expensive numerical methods. To overcome computational overhead, the field of surrogate models (Zhu and Zabaras 2018; Gramacy 2020; Lavin et al. 2021) has emerged which provide fast approximations of computationally expensive simulation. Examples are polynomial chaos expansion (Wiener 1938; Sudret 2008; Oladyshkin and Nowak 2012; Bürkner et al. 2023), Gaussian processes (Kennedy and O'Hagan 2001; Rasmussen and Williams 2005) or neural networks (Goodfellow et al. 2016). Recently, there has been a great interest in applying surrogate models in relevant areas, for example in hydrology (Mohammadi et al. 2018; Tarakanov and Elsheikh 2019; Zhang et al. 2020), in fluid dynamics (Meyer et al. 2021), in climate prediction (Kuehnert et al. 2022), or in systems biology (Renardy et al. 2018; Alden et al. 2020). Furthermore, great methodological advances have been made in the field of surrogate modeling, for example, the incorporation of physical knowledge (Raissi et al. 2019; Li et al. 2021; Brandstetter et al. 2023) or the combination of surrogate models and simulation-based inference (Radev et al. 2023).

Despite these advances, a major remaining challenge is the trustworthiness and reliability of the surrogate. Consequently, it is crucial to quantify uncertainties associated with surrogate modeling, e.g., caused by limited training data or inflexibility of the surrogate. To estimate the uncertainty in surrogate model parameters, several methods for uncertainty quantification (UQ) have been developed (e.g., Shao et al. 2017; Bürkner et al. 2023,). Uncertainty propagation (UP), a sub-field of UQ, is particularly important for addressing surrogate uncertainties in subsequent (surrogate-based) inference tasks (Smith 2013; Lavin et al. 2021; Psaros et al. 2023).

✉ Philipp Reiser
philipp-luca.reiser@simtech.uni-stuttgart.de

[1] Cluster of Excellence SimTech, University of Stuttgart, Stuttgart, Germany

[2] Department of Statistics, TU Dortmund University, Dortmund, Germany

Published online: 13 March 2025

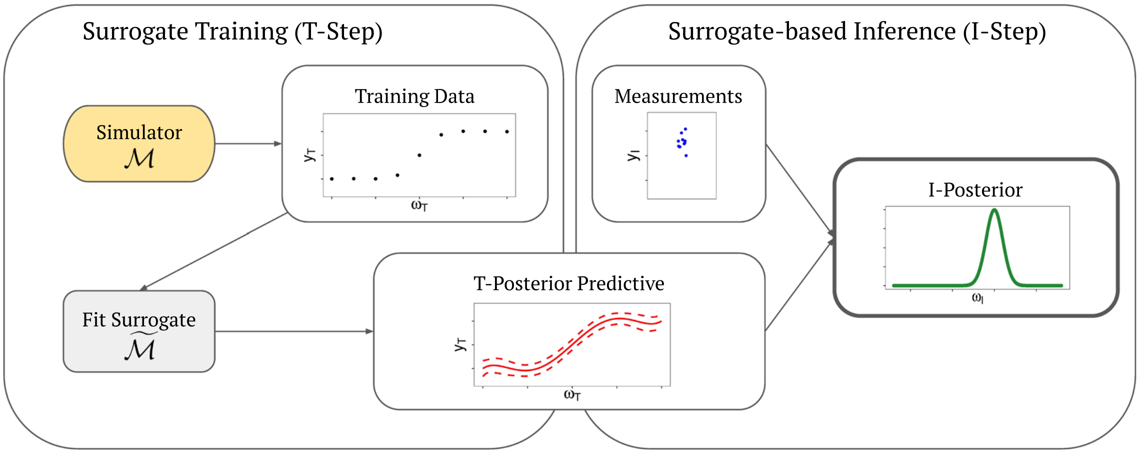

Fig. 1 Overview of two-step procedure. Left: In the surrogate training step (T-Step), training data is generated using a simulator and a surrogate model is fitted which allows to estimate the T-posterior. Right: In the surrogate-based inference step (I-Step), measurements along with the T-posterior are used to infer the I-posterior

Using probability theory as its fundamental basis, Bayesian statistics provides a rigorous way for UQ generally and specifically for UP (Gelman et al. 2013; McElreath 2020; Bürkner et al. 2023). One important use-case for UP in surrogate modeling arises from the "forward" problem, in which input parameters are uncertain and the goal is to propagate them through the surrogate while accounting for its uncertainty to compute a reliable output. For example, Ranftl and von der Linden (2021) proposed a method for UP in the forward problem, for restricted cases where conjugate analytic models are available. Similarly, Zhu and Zabaras (2018) quantified uncertainty in neural network surrogate outputs using approximate Bayesian inference.

In the "inverse" problem, when surrogates are used for Bayesian inference of parameters given observed data (e.g., Kennedy and O'Hagan 2001; Marzouk et al. 2007; Marzouk and Xiu 2009; Zeng et al. 2012; Laloy et al. 2013; Li and Marzouk 2014; Cleary et al. 2021), rigorous propagation of the surrogate uncertainty becomes even more relevant and challenging.

The framework introduced by Kennedy and O'Hagan (2001), for instance, attempts to jointly infer unknown input and surrogate parameters using data from both simulation models and observations. However, this approach leads to a loss of control over which parameters are updated by which data source, also implying complex posteriors that are hard to sample from (Bayarri et al. 2009).

To address these issues, modularization (Bayarri et al. 2009) has been introduced to surrogate modeling, aiming to update only specific parameters using selected data. Despite these advances, the uncertainty of the surrogate parameters is often neglected or simplified in the context of surrogate modeling. For example, Bayarri et al. (2009) propagated only point estimates of the surrogate parameters between modules, thus neglecting relevant uncertainty in subsequent calculations. Further, Zhang et al. (2020) applied surrogate-based Bayesian inference to hydrological systems and propagated parts of the surrogate uncertainty assuming normal surrogate posteriors. We review these methods in more detail in Sect. 2.2.3.

In this paper, our primary focus lies on UP when solving probabilistic inverse problems via surrogate models. Existing methods proposed for the same challenge are scarce and propagate the surrogate-based uncertainty only in a selective and simplified manner, which leaves a lot of room for both improved theory and improved practical methods. This not only concerns UP itself, but also diagnostic methods to assess the validity of the resulting inference. This paper aims at addressing these challenges from a fully probabilistic (Bayesian) perspective. Concretely, we make the following contributions: (i) Within a formal framework for surrogate-based Bayesian inference, we specify and categorize all relevant uncertainties. (ii) We present three distinct methods for propagating surrogate uncertainty within a two-step inference procedure, which consists of a surrogate training step (T-Step) and a surrogate-based inference step (I-Step). For a high-level overview, see Fig. 1. (iii) We adapt existing simulation-based procedures to validate the uncertainty calibration achieved via the different UP approaches. Finally, we evaluate our methods in three detailed case studies.

2 Method

In the following, we propose a framework for uncertainty propagation (UP) in surrogate-based Bayesian inference.

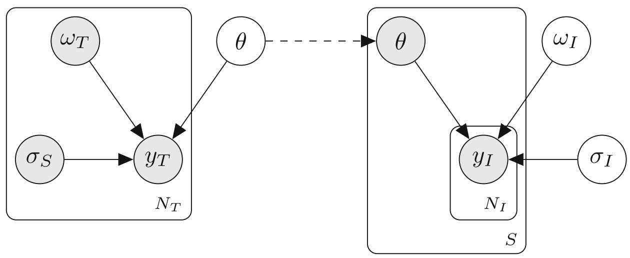

Fig. 2 Graphical model of the T-Step and I-Step. Left: In the T-Step, the observed quantities are simulation parameters ω_T, simulation output y_T, and the noise hyperparameters σ_S. The unknowns are the surrogate parameters θ. Right: In the I-Step, measurement data y_I is observed N_I times and S posterior samples of θ are propagated from the T-Step. The dashed arrow indicates that uncertainty in θ is propagated to the I-Step while θ is not updated using the data y_I. The unknowns to be inferred are the simulation parameters ω_I and the measurement error hyperparameters σ_I

This framework is applicable to general surrogate models, e.g., linear regression, polynomial chaos expansion, Gaussian processes or neural networks. The framework consists of a two-step procedure with a surrogate training step (T-Step) and an inference step (I-Step), an overview of all uncertainties occurring in surrogate-based inference, several UP schemes of selected uncertainties, and their evaluation.

2.1 Two-step procedure

We propose a two-step procedure for uncertainty propagation of surrogate models which consists of (i) the Surrogate-Training Step (T-Step), where training data is generated using a simulator and (ii) the Inference-Step (I-Step) where the trained surrogate is used to infer a quantity of interest, as illustrated in the graphical model (Jordan 1999; Ben-Gal 2008) shown in Fig. 2. Even though the general setup may look relatively simple, it becomes challenging to quantify and propagate all occurring uncertainties in a statistically rigorous manner. In the following, we differentiate between aleatoric (irreducible) and epistemic (reducible) uncertainty; for a review see Hüllermeier and Waegeman (2021); Gruber et al. (2023). In Table 1 we summarize all relevant uncertainties occurring in the two-step procedure to be discussed in detail below.

2.1.1 First step: training the surrogate (T-Step)

In the first step, we focus on training a surrogate model using artificial data generated from the complex simulator we seek to approximate. This involves computing a posterior over the surrogate parameters to account for the induced uncertainties.

Simulator We consider a given (potentially stochastic) simulator $\mathcal{M}$, which is an arbitrarily complex model, for example describing a physical or biological process, and which is hard to evaluate. Given an input simulation parameter ω_T, we obtain the output response from the simulator as follows:

$$y_T = \mathcal{M}(\omega_T; e_S) \quad \text{with} \quad e_S \sim p(e_S \mid \sigma_S), \tag{1}$$

where e_S is the noise of the simulation drawn from a simulator noise distribution $p(e_S \mid \sigma_S)$ with hyperparameters σ_S that describe the aleatoric (irreducible) uncertainty of the simulation. Note, that the simulation model itself does not necessarily have to be stochastic; it could be a deterministic physics-based or conceptual model that is complemented with a stochastic representation of measurement noise. We explicitly choose the subscript S to highlight that the noise stems from the simulation and not from the training process. This setup induces the true generating distribution $p(y_T \mid \omega_T, \sigma_S)$ of simulation responses y_T given input simulation parameters ω_T and noise hyperparameters σ_S.

Surrogate model A surrogate model $\widetilde{\mathcal{M}}$ is a statistical model that aims to approximate the simulator $\mathcal{M} \approx \widetilde{\mathcal{M}}$ while being computationally more efficient to evaluate. The parametric form of the surrogate is defined by a set of surrogate approximation parameters c. For example, if the surrogate model is a polynomial, then c are the polynomial coefficients. Given input parameters ω_T, surrogate approximation parameters c, and the simulator noise distribution $p(e_S \mid \sigma_S)$, we can calculate the surrogate response:

$$\tilde{y}_T = \widetilde{\mathcal{M}}(\omega_T, c, e_S) \quad \text{with} \quad e_S \sim p(e_S \mid \sigma_S). \tag{2}$$

Surrogate approximation error A surrogate model with a fixed architecture will always be misspecified, if the true simulator is not included in the class of surrogate models that we define. This is the standard use case, since we are specifically tailoring the surrogate model to be much simpler than the simulator. The uncertainty of the surrogate approximation parameters c only captures the (epistemic) uncertainty due to limited training data. To additionally account for the approximation error of the surrogate with respect to the simulator, caused by limited expressibility of the surrogate, we introduce an additional error term e_A. We assume e_A to follow a distribution $p(e_A \mid \sigma_A)$ with hyperparameters σ_A. This distribution captures aleatoric uncertainty which is not reducible by more training data. We can then model the true response y_T via a (potentially unknown) function $\tilde{f}$ that takes the output of the surrogate $\tilde{y}_T$ and the approximation error e_A:

$$y_T = \tilde{f}(\tilde{y}_T, e_A) \quad \text{with} \quad e_A \sim p(e_A \mid \sigma_A). \tag{3}$$

In practice, we might assume a simple additive error: $y_T = \tilde{y}_T + e_A$ and a normal distribution for e_A, but our framework is agnostic to these choices. For the sake of readability, we use $\theta = \{c, \sigma_A\}$ to combine all trainable surrogate parameters in a single vector. Together, this implies a surrogate likelihood of the simulator responses y_T:

Table 1 Uncertainties in the two-step procedure

Parameter/Posterior distribution	T-Step	I-Step	Synonyms
Simulator noise hyperparameter σ_S	Aleatoric	–	–
Surrogate approx error hyperparameter σ_A	Aleatoric	Aleatoric	T-aleatoric uncertainty
Surrogate parameter Posterior $p(c, \sigma_A \mid \mathcal{D}_T)$	Epistemic	Aleatoric	T-epistemic uncertainty / T-posterior
Measurement noise Hyperparameter σ_I	–	Aleatoric	–
Simulator-based Posterior $p(\omega_I \mid y_I; \mathcal{M})$	–	Epistemic	–
Surrogate-based Posterior $p(\omega_I \mid y_I; \widetilde{\mathcal{M}}, u)$	–	Epistemic & aleatoric*	I-posterior

* Depends on uncertainty propagation method u.

For each parameter and posterior distribution, we list the type of uncertainty (epistemic or aleatoric) in the T-/I-Step. We also list the synonyms used throughout the text

$$y_T \sim p(y_T \mid \omega_T, \sigma_S, \theta). \tag{4}$$

Surrogate training To train the surrogate, we use the simulator $\mathcal{M}$ to generate training data $\mathcal{D}_T = \{\omega_T^{(i)}, \sigma_S^{(i)}, y_T^{(i)}\}_{i=1}^{N_T}$ consisting of N_T inputs $(\omega_T^{(i)}, \sigma_S^{(i)})$ and corresponding outputs $y_T^{(i)} \sim p(y_T^{(i)} \mid \omega_T^{(i)}, \sigma_S^{(i)})$. The goal of the T-Step is to fit the parameters θ of the surrogate $\widetilde{\mathcal{M}}$ to approximate the data distribution implied by $\mathcal{M}$.

To train our surrogate parameters θ, we perform Bayesian inference using the fast-to-evaluate surrogate likelihood $p(y_T \mid \omega_T, \sigma_S, \theta)$ and a (potentially non-informative) prior $p(\theta)$. We obtain the joint posterior distribution over all surrogate parameters given the simulation training data $\mathcal{D}_T$ as:

$$p(\theta \mid \mathcal{D}_T) \propto \prod_{i=1}^{N_T} p(y_T^{(i)} \mid \omega_T^{(i)}, \sigma_S^{(i)}, \theta)\, p(\theta). \tag{5}$$

This joint posterior describes the epistemic (reducible) uncertainty in the surrogate model parameters. In the case of infinite training data, i.e. $N_T \to \infty$ the posterior $p(\theta|\mathcal{D}_T)$ converges to a point mass under regularity conditions (van der Vaart 2000). To approximate this posterior we can use sampling-based algorithms, such as Markov chain Monte Carlo (MCMC) (Robert and Casella 2005), and represent the posterior in the form of S posterior samples $\{\theta^{(1)}, \ldots, \theta^{(S)}\} \sim p(\theta \mid \mathcal{D}_T)$. The framework is in principle agnostic to the choice of estimation algorithm, as long as the algorithm can be used to obtain posterior samples. This includes MCMC but also other methods such as variational inference (Kucukelbir et al. 2017) or integrated Laplace approximation (Ruiz-Cárdenas et al. 2012; Martino and Riebler 2019). For a surrogate model with non-identifiable parameters θ, the algorithm would potentially have to deal with multimodalities (Medina-Aguayo and Christen 2022).

2.1.2 Second step: inference on real data (I-Step)

In the second step, real-world measurement data is given and our goal is to infer the unknown quantities of interest, i.e. the input simulation parameters, using the previously trained surrogate model as an efficient replacement of the complex simulator.

Measurement model We assume that the (implicit) real-world data generator is well described by the simulator $\mathcal{M}$, but with a potentially different measurement noise e_I. Measurement data y_I is then generated from an unknown underlying parameter ω_I (which has the same dimension as ω_T and serves as input to the simulator):

$$y_I = \mathcal{M}(\omega_I; e_I) \quad \text{with} \quad e_I \sim p(e_I \mid \sigma_I), \tag{6}$$

where the measurement noise e_I is drawn from a distribution $p(e_I \mid \sigma_I)$ with hyperparameters σ_I that describe the aleatoric uncertainty of the measurement. This induces a generating distribution $p(y_I \mid \omega_I, \sigma_I)$ of measurements y_I given inputs ω_I and noise hyperparameters σ_I. In contrast to the simulator setup, we only have access to the measurements y_I, but the underlying true input ω_I and σ_I are unknown.

Inference Given a set of real-world measurement data $y_I = \{y_I^{(i)}\}_{i=1}^{N_I}$ with $y_I^{(i)} \sim p(y_I^{(i)} \mid \omega_I, \sigma_I)$ for $i = 1, \ldots, N_I$, our goal is to infer the posterior of the unknown parameters ω_I, which constitute our primary quantity of interest. Additionally, we can also infer σ_I although it is only of secondary interest. To simplify the presentation, we drop σ_I from the notation in the following.

If the simulator $\mathcal{M}$ had a tractable and easy-to-evaluate likelihood function $p(y_I \mid \omega_I, \mathcal{M})$, we could simply calculate the posterior of the unknown simulation parameters ω_I given the measurement data y_I and a prior $p(\omega_I)$:

$$p(\omega_I \mid y_I, \mathcal{M}) \propto p(y_I \mid \omega_I, \mathcal{M})\, p(\omega_I). \tag{7}$$

However, for many real-world simulators, this likelihood is unavailable or highly cumbersome to evaluate (see Sect. 1), which is why we seek to replace it with simpler likelihood of the previously trained surrogate $\widetilde{\mathcal{M}}$. To incorporate the uncertainties of the surrogate parameters $\theta = (c, \sigma_A)$ from the T-Step into our real-world inference, we need to *somehow* propagate their T-Step posterior $p(\theta \mid \mathcal{D}_T)$ to the I-step. From the perspective of the I-Step, the uncertainty in $p(\theta \mid \mathcal{D}_T)$ becomes aleatoric since it is no longer reducible. The main question how to propagate $p(\theta \mid \mathcal{D}_T)$ turns out to have multiple answers – even multiple ones fully justified by probability theory. Each of these answers corresponds to an uncertainty propagation method $u \in \mathcal{U}$, for which we can obtain a surrogate-based posterior $p(\omega_I \mid y_I, \widetilde{\mathcal{M}}, u)$ as detailed below.

2.2 Uncertainty propagation in the two-step procedure

In the following, we present four different uncertainty propagation methods to perform the surrogate-based inference. These methods are namely (i) a Point Estimate, (ii) the Expected-Posterior (E-Post), (iii) the Expected-Likelihood (E-Lik), and (iv) the Expected-Log-Likelihood (E-Log-Lik). Accordingly, the set of considered propagation methods is given by $\mathcal{U} = \{\text{Point, E-Post, E-Lik, E-Log-Lik}\}$. The three latter approaches propagate the uncertainty from the T-Step using the full posterior or samples from the posterior. In E-Post, we incorporate the T-Step posterior $p(\theta \mid \mathcal{D}_T)$ directly into the posterior of the I-Step, while in the E-Lik and E-Log-Lik, the T-Step posterior is incorporated into the likelihood $p(y_I \mid \omega_I, \widetilde{\mathcal{M}}, u)$ of the I-step. In the following, all posteriors are calculated using the surrogate model $\widetilde{\mathcal{M}}$ and we will omit the dependency of $\widetilde{\mathcal{M}}$ in the posterior: $p(\omega_I \mid y_I, u) = p(\omega_I \mid y_I, \widetilde{\mathcal{M}}, u)$ to improve readability. Furthermore, for all uncertainty propagation methods, we will assume that y_I consists of conditional i.i.d. measurements $y_I^{(i)}$, such that the likelihood factorizes easily. This assumption is not necessary for our framework, but it simplifies the notation and allows for more efficient computation.

2.2.1 Point estimate

First, we describe the I-Step using only a point estimator $\hat{\theta}$ of the T-posterior $p(\theta \mid \mathcal{D}_T)$, e.g., the posterior mean, median, or mode. In this case, we condition the posterior of ω_I on $\hat{\theta}$, i.e., we reduce the full posterior to a point estimate without epistemic uncertainty. This is not a method we advocate for, but rather use it as a baseline to compare against more sophisticated methods. The posterior probability of ω_I given measurement data y_I and $\hat{\theta}$ is then (up to normalizing constants):

$$p(\omega_I \mid y_I, u = \text{Point}) \propto p(y_I \mid \omega_I, \hat{\theta}) p(\omega_I), \tag{8}$$

where the dependency of the posterior of ω_I on $\hat{\theta}$ is represented via $u = \text{Point}$. With the i.i.d. assumption, the (unnormalized) log I-posterior is

$$\begin{aligned} &\log p(\omega_I \mid y_I, u = \text{Point}) \\ &\quad \propto \sum_{i=1}^{N_I} \log p(y_I^{(i)} \mid \omega_I, \hat{\theta}) + \log p(\omega_I), \end{aligned} \tag{9}$$

where we use the $\propto$ symbol for log-probability statements to imply that an additive constant C (here, the log marginal likelihood) is not shown in the equation. Such a constant is independent of the parameters and thus irrelevant for posterior inference via MCMC or related sampling methods. The log-posterior can easily be specified in a probabilistic programming language (Gorinova et al. 2019) such as Stan (Carpenter et al. 2017) and we can then sample from the posterior with MCMC, leading to K posterior samples $\{\omega_I^{(1)}, \ldots, \omega_I^{(K)}\} \sim p(\omega_I \mid y_I, u = \text{Point})$. The benefit of the Point method is its comparably fast evaluation, since we only use a point estimate of the T-posterior and therefore can quickly evaluate the I-likelihood $p(y_I \mid \omega_I, \hat{\theta})$. While we are completely neglecting the epistemic uncertainty contained in the T-posterior, we propagate (a point estimate of) the surrogate approximation error parameter σ_A through $\hat{\theta}$. We expect the Point I-posterior $p(\omega_I \mid y_I, u = \text{Point})$ to be overconfident (too narrow); unless we have a sufficiently large amount of simulation training data $\mathcal{D}_T$ such that $p(\theta \mid \mathcal{D}_T)$ converges to a point mass and then corresponds exactly to the point estimator $\hat{\theta}$.

Related work Training a surrogate model using simulation data and subsequently employing its point estimate to infer unknown input parameters from observed data is a standard and well known approach (e.g., Marzouk et al. 2007; Laloy et al. 2013; Li and Marzouk 2014). This method has been extensively studied in the context of Bayesian surrogate models, particularly through modularization introduced by Bayarri et al. (2009).

2.2.2 Expected-posterior

Next, we present the Expected-Posterior (E-Post), a method that propagates both aleatoric and epistemic uncertainty in the T-posterior by marginalizing over the surrogate parameter posterior $p(\theta \mid \mathcal{D}_T)$ in the posterior of ω_I:

$$p(\omega_I \mid y_I, u = \text{E-Post}) = \int p(\omega_I \mid y_I, \theta) p(\theta \mid \mathcal{D}_T) \mathrm{d}\theta. \tag{10}$$

Since we only have samples $\theta^{(s)} \sim p(\theta \mid \mathcal{D}_T)$ from the T-step, we compute the Monte Carlo (MC) approximation of E-Post as:

$$p(\omega_I \mid y_I, u = \text{E-Post}) \overset{\text{MC}}{\approx} \frac{1}{S} \sum_{s=1}^{S} p(\omega_I \mid y_I, \theta^{(s)}), \tag{11}$$

which now is a finite mixture model with equal weights. This can easily be implemented in a probabilistic program by fitting a separate model for each T-posterior draw $\theta^{(s)} \sim p(\theta \mid \mathcal{D}_T)$ using the point I-posterior approach in Eq. (9), which results in K draws of ω_I, i.e., $\{\omega_I^{(s,1)}, \ldots, \omega_I^{(s,K)}\}$. The combination of these draws across all $s = 1, \ldots, S$ then represents an MC-estimate of the E-Post I-posterior as per Eq. (11), with a total of $S \cdot K$ posterior draws. We note that the number of propagated T-posterior draws S is a hyperparameter that needs to be tuned depending on the complexity and dimensionality of the problem.

Related work The idea of constructing a posterior via the aggregation of multiple posterior distributions, each approximated via samples, has been explored in multiple places in the literature. In the BayesBag method (Waddell et al. 2002; Douady et al. 2003; Bühlmann 2014; Huggins and Miller 2020), posteriors are obtained from bootstrapped copies (Efron 1979; Breiman 2004) of the original dataset and subsequently averaging the resulting bootstrapped posteriors. Similarly, in the context of missing value imputation (Little and Rubin 2019), this approach has been used to combine models fitted on multiple imputed datasets (Bürkner 2017, 2018). In terms of how many imputed data sets are needed, Austin et al. (2021) report that between 20 and 100 imputations are typically used. In both multiple data imputation and bagging, models are fitted to different datasets whereas in our E-Post method, the data is the same but the model itself changes. Furthermore, within the context of modularization of Bayesian models (Plummer 2014; Jacob et al. 2017), this approach is known as the cut distribution.

2.2.3 Expected-likelihood

Next, we present the Expected-Likelihood (E-Lik) approach, where we marginalize over the T-posterior $p(\theta \mid \mathcal{D}_T)$ in the I-likelihood:

$$p(y_I \mid \omega_I, u = \text{E-Lik}) = \int p(y_I \mid \omega_I, \theta) p(\theta \mid \mathcal{D}_T) \mathrm{d}\theta. \tag{12}$$

The full I-posterior of E-Lik is then given by

$$p(\omega_I \mid y_I, u = \text{E-Lik}) \propto p(y_I \mid \omega_I, u = \text{E-Lik}) p(\omega_I). \tag{13}$$

Assuming a factorizable likelihood, we compute the log I-posterior over the whole dataset y_I (without normalizing constant) as:

$$\begin{aligned} &\log p(\omega_I \mid y_I, u = \text{E-Lik}) \\ &\propto \log \int \prod_{i=1}^{N_I} p(y_I^{(i)} \mid \omega_I, \theta) p(\theta \mid \mathcal{D}_T) \mathrm{d}\theta + \log p(\omega_I), \end{aligned} \tag{14}$$

for which we can obtain an MC approximation using draws $\theta^{(s)} \sim p(\theta \mid \mathcal{D}_T)$:

$$\begin{aligned} &\log p(\omega_I \mid y_I, u = \text{E-Lik}) \\ &\overset{\text{MC}}{\approx} \log \left(\frac{1}{S} \sum_{s=1}^{S} \prod_{i=1}^{N_I} p(y_I^{(i)} \mid \omega_I, \theta^{(s)}) \right) + \log p(\omega_I). \end{aligned} \tag{15}$$

This representation has the problem that the product over likelihood components becomes numerically unstable as N_I grows larger, since the log operator cannot simply be pulled into sum over draws. To circumvent this, we use the `log-sum-exp` trick, i.e. calculate $\log \sum_{s=1}^{S} p(y_I|\omega_I, \theta^{(s)})$ $= \log \sum_{s=1}^{S} \exp(\log p(y_I \mid \omega_I, \theta^{(s)}))$, which has a numerically stable implementation (Carpenter et al. 2017). For given $\theta^{(s)}$, the joint log likelihood is then again a simple sum: $\log p(y_I \mid \omega_I, \theta^{(s)}) = \sum_{i=1}^{N_I} \log p(y_I^{(i)} \mid \omega_I, \theta^{(s)})$. In contrast to E-Post, E-Lik is expressed as a single probabilistic program and we can use MCMC to approximate its I-posterior with K samples $\{\omega_I^{(1)}, \ldots, \omega_I^{(K)}\}$.

More general, in contrast to E-Post, which marginalizes over θ in the I-posterior $p(\omega_I \mid y_I, \theta)$, the E-Lik method marginalizes over θ in the I-likelihood $p(y_I \mid \omega, \theta)$. E-Lik and E-Post are not identical (see Appendix Section B.1 for a counterexample), but we demonstrate in Sect. 3 that they usually yield very similar results.

Related work To our knowledge, E-Lik constitutes a novel method for full uncertainty propagation. The approach used in Zhang et al. (2020) appears related, although the details of their method are insufficiently described in the paper for a definitive assessment. Based on our understanding, their method can be seen as a special case of E-Lik, where the T-posterior is assumed to be normal and surrogate approximation error is ignored. Further, E-Lik is related to important quantities outside the area of UP: In particular, the log-likelihood constructed in E-Lik resembles the expected log predictive density (ELPD), a popular measure for predictive performance, which integrates the likelihood over the posterior of the same model before taking the logarithm outside the expectation (Vehtari and Ojanen 2012; Vehtari et al. 2017; Bürkner et al. 2023). What is more, in the context of meta-analysis, Blomstedt et al. (2019) proposed a method to combine posteriors resulting from different studies. While their sources of uncertainty are different than in our case, they also integrate them with an expected likelihood approach, rendering at least the core idea related to E-Lik.

2.2.4 Expected-log-likelihood

In the Expected-Log-Likelihood (E-Log-Lik) approach, instead of marginalizing over the likelihood as in E-Lik, we marginalize over the T-posterior in the *log*-likelihood. We define the E-Log-Lik I-likelihood as:

$$\begin{aligned} &p(y_I \mid \omega_I, u = \text{E-Log-Lik}) \\ &:= \exp\left(\int \log(p(y_I \mid \omega_I, \theta))\, p(\theta \mid \mathcal{D}_T) \mathrm{d}\theta\right). \end{aligned} \tag{16}$$

The posterior of the E-Log-Lik for ω_I is then defined as:

$$\begin{aligned} &p(\omega_I \mid y_I, u = \text{E-Log-Lik}) \\ &\propto p(y_I \mid \omega_I, u = \text{E-Log-Lik})\, p(\omega_I). \end{aligned} \tag{17}$$

When assuming a factorizable likelihood and ignoring the normalizing constant, the log posterior becomes

$$\begin{aligned} &\log p(\omega_I \mid y_I, u = \text{E-Log-Lik}) \\ &\propto \int \log\left(\prod_{i=1}^{N_I} p(y_I^{(i)} \mid \omega_I, \theta)\right) p(\theta \mid \mathcal{D}_T) \mathrm{d}\theta + \log p(\omega_I) \\ &= \int \sum_{i=1}^{N_I} \log p(y_I^{(i)} \mid \omega_I, \theta)\, p(\theta \mid \mathcal{D}_T) \mathrm{d}\theta + \log p(\omega_I), \end{aligned} \tag{18}$$

The integral is readily approximated via draws from the T-posterior:

$$\begin{aligned} &\log p(\omega_I \mid y_I, u = \text{E-Log-Lik}) \\ &\overset{\text{MC}}{\approx} \frac{1}{S} \sum_{s=1}^{S} \sum_{i=1}^{N_I} \log p(y_I^{(i)} \mid \omega_I, \theta^{(s)}) + \log p(\omega_I), \end{aligned} \tag{19}$$

which can be fitted with MCMC leading to K draws $\{\omega_I^{(1)}, \ldots, \omega_I^{(K)}\}$ in the I step. We can further rewrite the MC approximation of the log-likelihood as

$$\begin{aligned} &\log p(y_I \mid \omega_I, u = \text{E-Log-Lik}) \\ &= \sum_{s=1}^{S} \sum_{i=1}^{N_I} \log\left(p(y_I^{(i)} \mid \omega_I, \theta^{(s)})^{\frac{1}{S}}\right). \end{aligned} \tag{20}$$

This shows that the E-Log-Lik can be interpreted as power-scaling the likelihood components with equal weights $1/S$ (Geyer 1991; Kallioinen et al. 2022), which readily generalizes to unequal weights as we illustrate in Sect. 2.2.5. However, in contrast to E-Lik and E-Post, this does not strictly follow rules of probability theory as we integrate over a probability measure in the log space. Alternatively to E-Log-Lik, we could set up an Expected Log Posterior (E-Log-Post) as

$$\begin{aligned} &p(\omega_I \mid y_I, u = \text{E-Log-Post}) \\ &\propto \exp\left(\int \log p(\omega_I \mid y_I, \theta)\, p(\theta \mid \mathcal{D}_T) \mathrm{d}\theta\right), \end{aligned} \tag{21}$$

which however is equivalent to E-Log-Lik as we show in Appendix B.1

Related work The computation of the E-Log-Lik is conceptually similar to a single E-Step in an Expectation-Maximization (EM) algorithm (Dempster et al. 1977), where the log likelihood is integrated over a discretized space of latent variable values (instead of T-posterior draws as is done here). Furthermore, the Gibbs loss, a computational convenient although uncommon measure of predictive performance, also calculates an expectation over the log-likelihood (Vehtari and Ojanen 2012; Bürkner et al. 2023).

2.2.5 Clustering of the T-Posterior draws

To speed up computation of the I-Step while minimizing loss of information, we can reduce the number of propagated T-posterior draws via clustering (see Piironen et al. (2020) for a related use case in the context of variable selection). For this purpose, any clustering algorithm can in principle be used, for example, KMeans clustering (MacQueen 1967; Lloyd 1982). We apply the clustering algorithm to the set of T-posterior draws of the surrogate parameters $\{\theta^{(s)}\}_{s=1}^{S}$ to get cluster centroids $\{\mu^{(l)}\}_{l=1}^{L}$. The sufficient number of clusters L for a trustworthy approximation of the I-posterior highly depends on the complexity of the I-posterior and is a hyperparameter that needs to be tuned. Here, we carried out a visual convergence analysis, given that rigorous guidelines on the choice of the number of clusters are lacking to date and an open scientific challenge. For each of the cluster centroids we additionally store weights $\{\alpha^{(l)}\}_{l=1}^{L}$, where $\alpha^{(l)}$ is the percentage of draws associated with the cluster. While inducing an approximation error, the cluster centroids together with the corresponding weights allow for a reliable and computationally efficient processing of the T-posterior draws as we show in Sect. 3.

Clustering and re-weighting can be easily applied in all UP methods by replacing $\theta^{(s)}$ with $\mu^{(l)}$ as well as replacing the equal weight $1/S$ with $\alpha^{(l)}$ after moving it inside the sum. For example, when using E-Log-Lik, the MC-approximated integral becomes:

$$
\begin{aligned}
&\log p(y_I \mid \omega_I, u = \text{E-Log-Lik}) \\
&\overset{\text{MC}}{\approx} \frac{1}{S} \sum_{s=1}^{S} \sum_{i=1}^{N_I} \log(p(y_I^{(i)} \mid \omega_I, \theta^{(s)}) \\
&\approx \sum_{l=1}^{L} \sum_{i=1}^{N_I} \alpha^{(l)} \log p(y_I^{(i)} \mid \omega_I, \mu^{(l)}).
\end{aligned} \tag{22}
$$

2.2.6 Parallelization

Inference based on all introduced UP methods can be parallelized, but to a different degree. To compute the I-posterior using E-Post, we fit a separate model for each T-posterior draw (or each cluster of draws). This is embarrassingly parallelizable, as the models are independent so can simply be run on different cores. However, for each model, a separate MCMC warmup phase is required, which induces computational overhead.

In contrast, for E-Lik, we fit only one model during the I-step, which loops over the T-posterior draws when evaluating its likelihood. This however defines a more complicated posterior, which is substantially slower to sample from compared to the individual E-Post models. Fitting the single E-Lik model can be sped up by between-chain parallization when running multiple MCMC chains (say, one per core), but then we again create overhead due to separate warmup phases per chain. Alternatively, even when running a single chain, the E-Lik model can be parallelized via within-chain parallelization (aka threading), where the likelihood contributions of the T-posterior draws are evaluated in parallel. An overhead occurs due to variable passing and other non-parallelized model parts (e.g., the prior density evaluation). This leads to diminishing returns in terms of the number of cores used for threading. For more details on threading in Stan, see Bürkner et al. (2022).

E-Log-Lik behaves as E-Lik in terms of parallelizability as they both fit only a single model during the I-step. That said, in our experiments, E-Log-Lik models sampled substantially faster than E-Lik models, presumably for two main reasons. First, E-Log-Lik does not require the use of `log-sum-exp` in order to obtain the joint log-likelihood, since we aggregate directly on the log-scale. This reduces the number of required operations within each MCMC step. Second, presumably, the geometry of the E-Log-Lik I-posterior is simpler than that of the E-Lik I-posterior, thus implying a more efficient exploration with MCMC for the former.

2.3 Evaluation of the two-step procedure

Checking the calibration of uncertainty estimates is an important step to improve the trustworthiness of any inference algorithm. As such, it is a crucial aspect of the Bayesian workflow (Gelman et al. 2013). In our setup, we are specifically interested in the uncertainty calibration of $p(\omega_I \mid y_I, \widetilde{\mathcal{M}}, u)$, that is our I-posterior implied by the surrogate. Simulation-based Calibration (SBC) checking (Cook et al. 2006; Talts et al. 2018; Modrák et al. 2022) is a current gold-standard approach to validate Bayesian computation, jointly testing the trinity of the simulator, the probabilistic program, and the posterior approximation algorithm, e.g., a sampling algorithm such as MCMC.

Algorithm 1 SBC for the Two-Step Procedure

Choose simulator $\mathcal{M}$, surrogate $\widetilde{\mathcal{M}}$, and uncertainty propagation method u
for m in Number T-Step trials **do**
  Draw a training dataset $\mathcal{D}_T$ using the simulator $\mathcal{M}$
  Fit the surrogate $\widetilde{\mathcal{M}}$ and calculate the posterior $p(\theta \mid \mathcal{D}_T)$ (T-Step)
  for n in Number I-Step trials **do**
    Draw a prior sample $\omega_I^* \sim p(\omega_I), \sigma_I \sim p(\sigma_I)$
    Draw measurements $y_I \sim p(y_I \mid \omega_I^*, \sigma_I, \mathcal{M})$ using simulator $\mathcal{M}$
    Draw posterior samples $\{\omega_I^{(1)}, \ldots, \omega_I^{(K)}\} \sim p(\omega_I \mid y_I, \widetilde{\mathcal{M}}, u)$ (I-Step)
    Store the rank of ω_I^* within the set of posterior samples $\{\omega_I^{(1)}, \ldots, \omega_I^{(K)}\}$
  end for
end for
Perform uniformity test on the stored ranks

We extend SBC to the two-step procedure and use the notation of Bürkner et al. (2023). For any quantile $q \in (0, 1)$, let $U_q(\omega_I \mid y_I, \widetilde{\mathcal{M}}, u)$ be any uncertainty region (e.g., the quantile-based credible intervals or highest density intervals) given by the posterior $p(\omega_I \mid y_I, \widetilde{\mathcal{M}}, u)$ that depends on the surrogate $\widetilde{\mathcal{M}}$ and the uncertainty propagation method u (see Sect. 2.1 and 2.2). If the generating distribution of the assumed surrogate $\widetilde{\mathcal{M}}$ is equal to the true data-generating distribution induced by the simulator the following property holds simultaneously for all $q \in (0, 1)$:

$$
\begin{aligned}
q = \iint & \mathbb{I}[\omega_I^* \in U_q(\omega_I \mid y_I, \widetilde{\mathcal{M}}, u)]\, p(y_I \mid \omega_I^*, \mathcal{M}) \\
& \times\ p(\omega_I^*)\, dy_I\, d\omega_I^*,
\end{aligned} \tag{23}
$$

where we ignore σ_I for simplicity and $\mathbb{I}[\cdot]$ denotes the indicator function. This self-consistency property tests the correct coverage of the uncertainty region conditional on the input for every quantile. Practically, we check that the prior draws are uniformly distributed in the surrogate-based samples of ω_I. We calculate the ranks as $r(\omega_I^*, \{\omega_I^{(1)}, \ldots, \omega_I^{(K)}\}) = \sum_{k=1}^{K} \mathbb{I}\left[\omega_I^* \le \omega_I^{(k)}\right]$ and test them for uniformity using graphical tests as proposed in Säilynoja et al. (2022). In addition to graphical tests, we calculate the $\log(\gamma)$-statistic (Säilynoja et al. 2022) as a quantitative measure of uniformity allowing for a faster comparison of calibration (or strength of miscalibration) between different methods. Checking the calibration of a specific uncertainty region, e.g., $U_{0.95}(\omega_I \mid y_I, \widetilde{\mathcal{M}}, u)$, corresponds to evaluating Eq. (23) for $q = 0.95$ and is a special case of SBC.

First, as a point of comparison, we explain standard SBC for inference using the simulator $\mathcal{M}$: (i) Sample a simulation input parameter $\omega_I^* \sim p(\omega_I)$, (ii) conditioned on ω_I^* generate measurement output data $y_I \sim p(y_I \mid \omega_I^*)$, (iii) using the simulator $\mathcal{M}$ and given the measurements y_I draw samples $\{\omega_I^{(1)}, \dots, \omega_I^{(K)}\}$, and (iv) using the posterior samples, we calculate the rank statistics $r(\omega_I^*, \{\omega_I^{(1)}, \dots, \omega_I^{(K)}\})$ from the posterior $p(\omega_I \mid y_I, \mathcal{M})$. This procedure is repeated for a chosen number of I-Step trials and uniformity is tested on the stored ranks, as described above.

However, in the two-step procedure (see Sect. 2.1), where we additionally train a surrogate given simulation data (T-Step) and then propagate its uncertainty to the I-Step, we need to extend SBC as shown in Algorithm 1: We repeat the T-step multiple times (number of T-Step trials) and, in each iteration, we simulate a new training dataset $\mathcal{D}_T$ used to train the surrogate $\widetilde{\mathcal{M}}$. Within each such T-Step trial, we repeat the I-Step for SBC: First, we draw a sample of the simulation parameters ω_I^* and the simulation noise hyperparameter σ_I from their respective priors. Next, measurement data y_I is generated using the simulator $\mathcal{M}$. Using the surrogate $\widetilde{\mathcal{M}}$ and the UP method u, we draw posterior samples $\{\omega_I^{(1)}, \dots, \omega_I^{(K)}\}$ from the I-posterior and store the ranks $r(\omega_I^*, \{\omega_I^{(1)}, \dots, \omega_I^{(K)}\})$. The repetition of the T-Step and I-Step, in addition to covering the whole input space of ω_I, helps to marginalize over both the noise of the simulator and the noise in the (assumed) measurement process.

With this SBC variant of the two-step procedure, we simultaneously test six different scenarios, where a failure can indicate one or more of the following scenarios:

- Scenarios also tested in standard SBC
  - (i) Incorrect implementation of simulator $\mathcal{M}$
  - (ii) Incorrect implementation of probabilistic program of the surrogate
  - (iii) Problems with the sampling algorithm
- Additional scenarios in proposed SBC for surrogate-based inference
  - (iv) Inflexible surrogate
  - (v) Insufficient training of surrogate because of too little simulation training data
  - (vi) Inappropriate uncertainty propagation in the surrogate-based inference.

Regarding points (i)–(iii), we assume the simulator to be correct and the surrogate to be relatively simple (implementation-wise) as well as easy to fit using MCMC. Concerning points (iv) and (v), we need a sufficiently flexible surrogate in order to remove the approximation error (i.e., $\sigma_A \to 0$) and an infinite amount of training samples ($N_T \to \infty$) in order to remove the epistemic uncertainty in the posterior $p(\theta \mid \mathcal{D}_T)$. However, practically the latter will not be the case and we expect the calibration to be imperfect. Nonetheless, we will still be able to compare surrogate models $\widetilde{\mathcal{M}}$ and different UP methods u by comparing their SBC results, either graphically or via test statistics.

3 Experiments

We evaluate our surrogate-based Bayesian inference framework in three case studies: (1) A linear setup, where we propagate only epistemic uncertainty, (2) a nonlinear setup, in which we propagate both epistemic and aleatoric uncertainty, and (3) a real-world model. All code and material can be found on GitHub.[1]

3.1 Case study 1: uncertainty propagation in a linear model

In the first case study, we use a linear setup leading to partly analytic posteriors that allow us to study our framework in a simple, well-understood scenario.

Setup We consider a simple linear model as simulator:

$$y_T = \mathcal{M}(\omega_T, \sigma_S = 0) = a + b\omega_T, \tag{24}$$

with simulation input parameter ω_T and two simulation control parameters a, b set to $a = 0.5$, $b = 2$, and a simulation noise parameter σ_S set to zero, i.e., we use a deterministic simulator. For the T-Step, we consider $N_T = 2$ training points $\mathcal{D}_T = \{\omega_T^{(i)}, y_T^{(i)}\}_{i=1}^{N_T}$ (chosen according to Table 2), where $\omega_T^{(i)}$ denotes the simulation input and $y_T^{(i)} = \mathcal{M}(\omega_T^{(i)})$ the corresponding simulation output. We denote by $\Omega_T = \left[\begin{smallmatrix} 1 & \omega_T^{(1)} \\ 1 & \omega_T^{(2)} \end{smallmatrix}\right]$ the design matrix of all input parameters and by y_T the vector of all simulation outputs.

We set the surrogate model to a linear model as well, that is, we use the same model class as for the simulator:

$$\widetilde{\mathcal{M}}(\omega_T, c) = c_1 + c_2\omega_T, \tag{25}$$

where the intercept c_1 and slope c_2 form the surrogate approximation parameters $c = [c_1, c_2]^\top$. For the T-Step, we only consider the surrogate approximation parameters c as trainable surrogate parameters. Even though the surrogate can

[1] https://github.com/philippreiser/bayesian-surrogate-uncertainty-paper.

match the simulator perfectly, we fix the surrogate approximation error hyperparameter σ_A to values greater than zero. This allows us to control the width of the T-posterior and thus the epistemic uncertainty, as shown below. As prior on the surrogate approximation parameters we choose a bivariate normal distribution:

$$p(c) = \mathcal{N}(c \mid \mu_{T0}, \Sigma_{T0}), \tag{26}$$

with mean $\mu_{T0} = 0$ and a covariance matrix $\Sigma_{T0} = \sigma_{T0}^2 I$ where I is the identity matrix. We set the T-likelihood to a normal distribution as well:

$$p(y_T \mid c) = \prod_{i=1}^{N_T} \mathcal{N}(y_T^{(i)} \mid c_1 + c_2\omega_T^{(i)}, \sigma_A^2). \tag{27}$$

To generate data for the I-Step, a single measurement $y_I = \mathcal{M}(\omega_I^*)$ is obtained by inputting a true input parameter ω_I^* to the deterministic simulator $\mathcal{M}$. In the following, we will also use the augmented vector $\hat{\omega}_I = [1, \omega_I]^T$ to simplify notation. Within the I-step, we set a normal prior with mean μ_{I0} and variance σ_{I0}^2 on ω_I:

$$p(\omega_I) = \mathcal{N}(\omega_I \mid \mu_{I0}, \sigma_{I0}^2). \tag{28}$$

The I-likelihood $p(y_I \mid \omega_I)$ is assumed to be normal with fixed variance σ_I^2:

$$p(y_I \mid \omega_I, c) = \mathcal{N}(y_I \mid c_1 + c_2\omega_I, \sigma_I^2) \tag{29}$$

Deriving the posteriors In the following, we perform the T-Step (see Sect. 2.1.1) and I-Step (see Sect. 2.2) for the linear setup. In the T-Step, we calculate the T-posterior of the surrogate approximation parameters c, which contains epistemic uncertainty. Then we derive the four different I-step methods when propagating only the epistemic uncertainty from the T-posterior. The aleatoric uncertainty is zero because, in our setup, the simulator is within the approximation space of the surrogate model.

T-Step We calculate the T-posterior for the surrogate approximation parameters c given the training data $\mathcal{D}_T$ from the simulator by using the conjugate prior relation for a normal-normal model (Murphy 2007, 2012) leading to a normal posterior:

$$p(c \mid \mathcal{D}_T) = \mathcal{N}(c \mid \mu_{T1}, \Sigma_{T1}), \tag{30}$$

with

$$\Sigma_{T1} = (\Sigma_{T0}^{-1} + \sigma_A^{-2}\Omega_T^\top \Omega_T)^{-1}, \tag{31}$$

$$\mu_{T1} = \Sigma_{T1}(\Sigma_{T0}^{-1}\mu_{T0} + \sigma_A^{-2}\Omega_T y_T). \tag{32}$$

We see that by fixing σ_A we can control the width of the T-posterior and hence the epistemic uncertainty of the surrogate parameters, even if we do not propagate σ_A itself.

I-Step Below, we derive the I-posteriors of all four uncertainty propagation procedures. We compute the mean of the T-posterior: $\bar{c} = \mu_{T1}$ and use it to calculate the Point I-posterior as follows:

$$p(\omega_I \mid y_I, u = \text{Point}) \propto p(\omega_I)p(y_I \mid \omega_I, \bar{c}) \tag{33}$$

$$= \mathcal{N}(\omega_I \mid \mu_{I0}, \sigma_{I0}^2)\mathcal{N}(y_I \mid \mu_{T1}^{(1)} + \mu_{T1}^{(2)}\omega_I, \sigma_I^2) \tag{34}$$

$$\propto \mathcal{N}(\omega_I \mid \mu_{I1}, \sigma_{I1}^2), \tag{35}$$

with

$$\sigma_{I1}^2 = (\sigma_{I0}^{-2} + \sigma_I^{-2}\mu_{T1}^{(2)}\mu_{T1}^{(2)})^{-1}, \tag{36}$$

$$\mu_{I1} = \sigma_{I1}^2(\sigma_{I0}^{-2}\mu_{I0} + \sigma_I^{-2}\mu_{T1}^{(2)}(y_I - \mu_{T1}^{(1)})). \tag{37}$$

Again, this works due to the normal-normal conjugacy. We see that the I-posterior variance σ_{I1}^2 does not depend on the T-posterior covariance matrix Σ_{T1}, i.e. the uncertainty from the surrogate training step is neglected.

The E-Log-Lik I-posterior is given by

$$\begin{aligned} &p(\omega_I \mid y_I, u = \text{E-Log-Lik}) \\ &\propto p(\omega_I) \exp\left\{\int \log(p(y_I \mid \omega_I, c))p(c \mid \mathcal{D}_T)\mathrm{d}c\right\} \\ &= \mathcal{N}(\omega_I \mid \mu_{I0}, \sigma_{I0}^2) \\ &\quad \times \exp\left\{\int \log(\mathcal{N}(y_I \mid \hat{\omega}_I^\top c, \sigma_I^2))\mathcal{N}(c \mid \mu_{T1}, \Sigma_{T1})\mathrm{d}c\right\} \\ &\propto \mathcal{N}(\omega_I \mid \mu_{I1}, \sigma_{I1}^2), \end{aligned} \tag{38}$$

with

$$\begin{aligned} \sigma_{I1}^2 &= (\sigma_{I0}^{-2} + \sigma_I^{-2}(\mu_{T1}^{(2)}\mu_{T1}^{(2)} + \Sigma_{T1}^{(2,2)}))^{-1}, \\ \mu_{I1} &= \sigma_{I1}^2(\sigma_{I0}^{-2}\mu_{I0} \\ &\quad + \sigma_I^{-2}(\mu_{T1}^{(2)}y_I - \Sigma_{T1}^{(1,2)} - \mu_{T1}^{(1)}\mu_{T1}^{(2)})) \end{aligned} \tag{39}$$

Accordingly, it is also a normal distribution, but a different one from the Point I-posterior. The detailed derivation of the E-Log-Lik is given in Appendix B.3.1. If we look at the variance σ_{I1}^2 we see that E-Log-Lik produces counter-intuitive results, as $\Sigma_{T1}^{(2,2)}$ and σ_{I1}^2 are reciprocally related. That is, as the surrogate model gets more uncertain in the T-step, the I-posterior gets more certain.

The E-Lik I-posterior is computed as

$$
\begin{aligned}
& p(\omega_I \mid y_I, u = \text{E-Lik}) \\
& \quad \propto p(\omega_I) \int p(y_I \mid \omega_I, c) p(c \mid \mathcal{D}_T) \mathrm{d}c \\
& \quad = \mathcal{N}(\omega_I \mid \mu_{I0}, \sigma_{I0}^2) \\
& \qquad \times \int \mathcal{N}(y_I \mid \hat{\omega}_I^\top c, \sigma_I^2) \mathcal{N}(c \mid \mu_{T1}, \Sigma_{T1}) \mathrm{d}c \\
& \quad = \mathcal{N}(\omega_I \mid \mu_{I0}, \sigma_{I0}^2) \\
& \qquad \times \mathcal{N}(y_I \mid \hat{\omega}_I^\top \mu_{T1}, \hat{\omega}_I^\top \Sigma_{T1} \hat{\omega}_I + \sigma_I^2).
\end{aligned} \tag{40}
$$

Here, we cannot apply the normal-normal conjugacy, because $\hat{\omega}_I$ is present in both the mean and the variance in the second term of the product of the two normals. Instead, the resulting distribution is non-analytic and we have to use numerical integration to calculate the normalization constant. Looking at the derived quantity, we see that the variance of the term resulting from the marginalization over the surrogate approximation parameters c increases with the variance of the T-posterior Σ_{T1}.

Finally, we calculate the I-posterior using the E-Post approach:

$$
\begin{aligned}
& p(\omega_I \mid y_I, u = \text{E-Post}) \\
& \quad = p(\omega_I) \int \frac{p(y_I \mid \omega_I, c) p(c \mid \mathcal{D}_T)}{\int p(y_I \mid \omega_I, c) p(\omega_I) \mathrm{d}d\omega_I} \mathrm{d}c \\
& \quad = \mathcal{N}(\omega_I \mid \mu_{I0}, \sigma_{I0}^2) \\
& \qquad \times \int \frac{\mathcal{N}(y_I \mid \hat{\omega}_I^\top c, \sigma_I^2) \mathcal{N}(c \mid \mu_{T1}, \Sigma_{T1})}{\mathcal{N}(y_I \mid c_1 + c_2 \mu_{I0}, c^\top c \sigma_{I0}^2 + \sigma_I^2)} \mathrm{d}c.
\end{aligned} \tag{41}
$$

Again, this integral is non-analytic and numerical integration is required to calculate the E-Post I-posterior. Nevertheless, we can expect similar behavior to E-Lik, since we also marginalize over the T-posterior and only normalize differently.

Results

In Table 2, we provide an overview over the chosen training data, measurement data, and hyperparameters for the T and I-Step. We specifically vary the surrogate approximation error parameter $\sigma_A = \{0.1, 0.5, 1\}$ to control the epistemic uncertainty during the surrogate training, as explained above. To compare the four I-Steps, we calculate their I-posteriors under the given scenarios. The results are illustrated in Fig. 3. We see that Point produces results that are independent of the standard deviation of the T-likelihood. This is expected since it does not propagate the T-epistemic uncertainty at all. Intuitively, for the other three methods, the I-posteriors should become more uncertain as we increase the uncertainty in the T-step, since our surrogate model gets less trustworthy. This is indeed what happens for E-Lik and E-Post, which also produce very similar but not identical results. In contrast, E-Log-Lik behaves counter-intuitively since its I-posterior becomes more *certain* as the T-posterior becomes more uncertain, a behavior that we also see clearly from Eq. (39), as noted above.

3.2 Case study 2: uncertainty propagation in a logistic model

The second case study examines a nonlinear problem where we have to rely on MCMC for the posterior approximations, because analytic posteriors are unavailable. We choose the one-dimensional logistic function

$$
y = \mathcal{M}(\omega) = \frac{2}{1 + \exp(-10\omega)} - 1 \tag{42}
$$

as the (true) simulator since it is invertible and smooth everywhere. We consider two different surrogate models: The first is a parameterized generalization of the simulator and the second is a polynomial chaos expansion (PCE) surrogate. We will discuss these two cases separately below.

3.2.1 Logistic surrogate model

Setup As surrogate, we consider

$$
\widetilde{\mathcal{M}}(\omega; c) = \frac{\alpha}{1 + \exp(-\beta(\omega - \gamma))} + \delta, \tag{43}
$$

with surrogate approximation parameters $c = (\alpha, \beta, \gamma, \delta)$. Here, the true simulator is contained in the set of surrogate models (for $\alpha = 2, \beta = 10, \gamma = 0, \delta = -1$).

For the T-Step, we generate the training set by setting the input points $\omega_T^{(i)} \in [-1, 1]$ to the first N_T points of a slightly modified one-dimensional Halton sequence (Halton 1960), which starts with the boundary points (-1 and 1) and then progresses as the standard Halton sequence with center point 0. The simulated output responses $y_T^{(i)}$ for $i \in \{1, \ldots, N_T\}$ are sampled from a normal distribution with mean equal to the evaluated logistic simulator $\mathcal{M}$ at the input points and standard deviation σ_S, i.e. $y_T^{(i)} \sim \mathcal{N}(\mathcal{M}(\omega_T^{(i)}), \sigma_S^2)$. To avoid sampling issues during the training step, we induce a small simulation noise $\sigma_S = 0.01$, which is not explicitly modelled.

For the surrogate parameters c, we specify normal priors with means around the true values and a standard deviation of 1 (except for β, where we set the standard deviation to 10). For the T-likelihood, we consider a normal distribution:

$$
p(y_T \mid c) = \mathcal{N}(y_T \mid \widetilde{\mathcal{M}}(\omega_T, c), \sigma_A^2) \tag{44}
$$

with the surrogate approximation hyperparameter σ_A. On each training data set, we fit the parameters of our surrogate using Markov chain Monte Carlo (MCMC). Specifically, we use the no-U-turn-sampler (NUTS) (Hoffman and Gelman

Table 2 Parameters with realized values for case study 1

Parameters	T-Step					I-Step			
	$\begin{bmatrix}\omega_T^{(1)}\\ \omega_T^{(2)}\end{bmatrix}$	$\begin{bmatrix}a\\ b\end{bmatrix}$	μ_{T0}	σ_{T0}	σ_A	ω_I^*	μ_{I0}	σ_{I0}	σ_I
Values	$\begin{bmatrix}-0.9\\ -0.3\end{bmatrix}$	$\begin{bmatrix}0.5\\ 2\end{bmatrix}$	$\begin{bmatrix}0\\ 0\end{bmatrix}$	10	$\{0.1, 0.5, 1\}$	-0.5	0	1	0.1

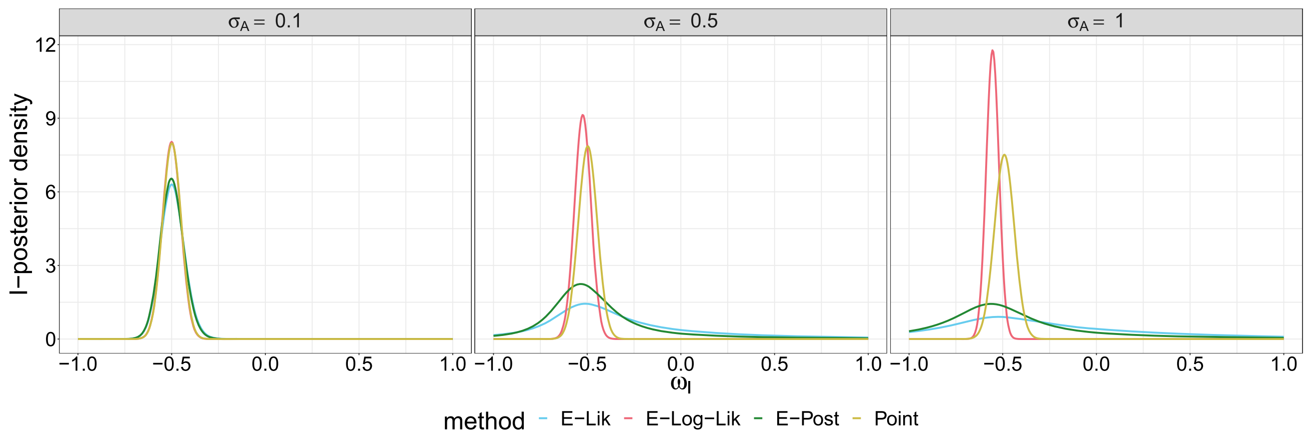

Fig. 3 I-posterior densities for the linear surrogate with normal priors/likelihoods in case study 1. We use the data and parameters as specified in Table 2. We use four different UP methods to compute the I-posterior while the surrogate approximation error $\sigma_A = \{0.1, 0.5, 1\}$ is varied

2014), an adaptive form of Hamiltonian Monte Carlo (HMC) (Neal 2011) which is a gradient based MCMC sampler via the probabilistic programming language Stan (Stan Development Team 2024). We use four chains, each running for 1250 iterations (1000 warmup and 250 post-warmup iterations), resulting in a total of 1000 T-posterior draws. We assessed convergence using standard convergence checks (i.e., the R-hat diagnostic of all parameters (Vehtari et al. 2021)).

For the I-Step, we generate $N_I = 5$ random measurements $y_I \sim \mathcal{N}(\mathcal{M}(\omega_I^*), \sigma_I^2)$, based on the simulator, true input parameters ω_I^*, and the measurement error $\sigma_I = 0.01$. As priors we set $\omega_I \sim \mathcal{N}_{[-1,1]}(0, 0.5^2)$ (with truncation bounds $[-1, 1]$) and $\sigma_I \sim$ uniform$(0, 0.05)$. These hyperparameters were chosen so that the true simulator could make valid inference about the input parameters given the measurement data.

In contrast to case study 1, we propagate the T-posterior through samples and hence use the MC approximation (see Sect. 2.2) for each of the four methods (Point, E-Lik, E-Log-Lik, E-Post). We sample from the I-posterior of ω_I and σ_I using NUTS with four chains, each running for 5000 iterations (1000 warmup and 4000 post-warmup iterations), resulting in a total of 16000 I-posterior draws.

Similar to case study 1, we propagate only the epistemic uncertainty that is encoded in the T-posterior of the surrogate parameters c. For this purpose, we either utilize all T-posterior draws or employ KMeans clustering (see Sect. 2.2.5) with $L = 25$ clusters. As the simulator is contained in the class of surrogates, no approximation error is present ($\sigma_A = 0$) and hence, there is no aleatoric uncertainty to propagate.

Posterior distributions We set $\sigma_S = 0.01$ and vary the number of training points $N_T = \{5, 7, 10\}$ to perform the T-Step. In Fig. 4, the left column shows the mean of the T-posterior predictive with the 95%-credible interval (CI) to display its epistemic uncertainty. As expected, increasing N_T leads to smaller T-epistemic uncertainty. In Appendix Fig. 11 we depict the pairs plot of the T-posterior draws of the logistic surrogate.

We compare the I-posteriors resulting from the four different methods using the measurements resulting from exemplary underlying true inputs $\omega_I^* \in \{-0.05, 0.1, 0.3\}$ in the three right columns in Fig. 4. The Point method yields I-posteriors with constant width regardless of variation in N_T. The I-posteriors of E-Lik and E-Post show similar behavior and become more uncertain as T-epistemic uncertainty increases. The E-Log-Lik follows a similar trend, but its I-posteriors have qualitatively different shapes and are narrower. For $N_T = 5$, only E-Lik and E-Post have substantial I-posterior probability mass on the true inputs ω_I^*. Notably, all uncertainty propagation methods converge to the same results as the epistemic uncertainty from the T-Step decreases.

Calibration To check if the methods for estimating the I-posteriors are calibrated, we use SBC checking (Talts et al. 2018; Modrák et al. 2022) with the `SBC` R package (Kim

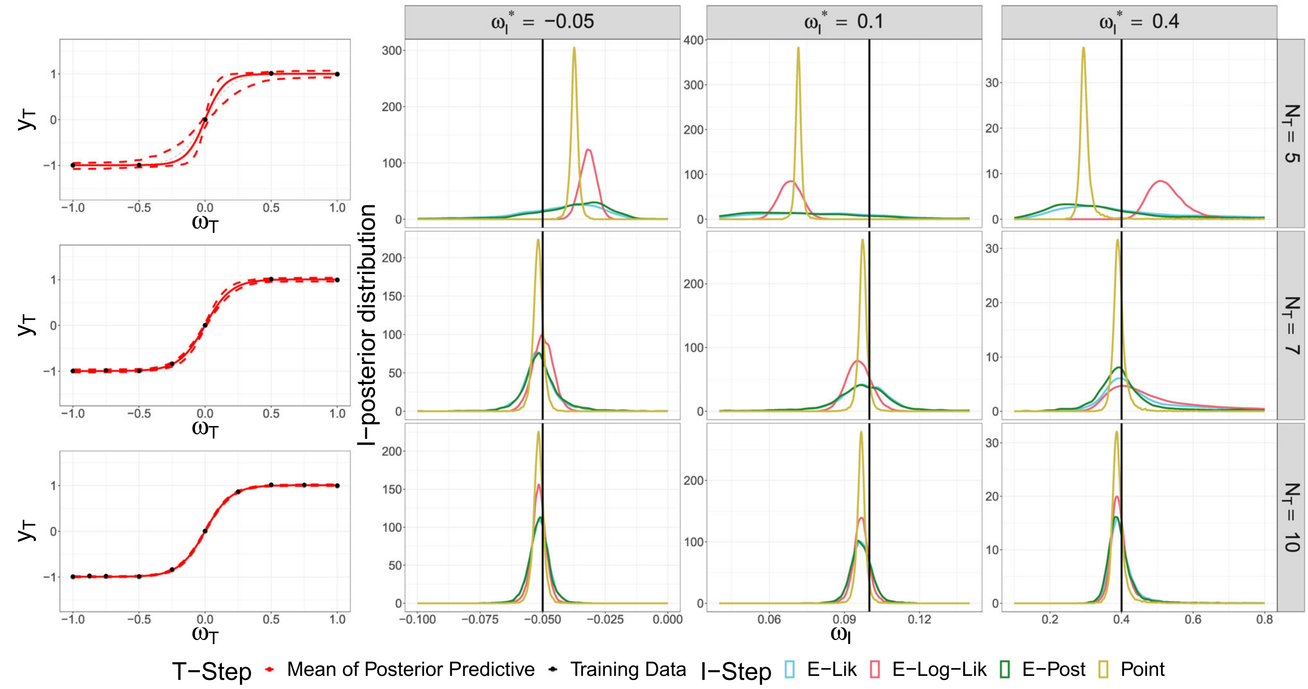

Fig. 4 Selected results for two-step procedure with the logistic surrogate in case study 2. Left: For $N_T = \{5, 7, 10\}$ the training data set $\mathcal{D}_T$ (black dots) and the mean of the T-posterior predictive distribution (red lines) is shown. Right: For each underlying true input $\omega_I^* \in \{-0.05, 0.1, 0.3\}$ (black vertical lines), we depict the I-posterior distributions for each Point, E-Lik, E-Post, and E-Log-Lik (colored lines)

et al. 2023). Concretely, we perform the adapted SBC procedure for surrogate-based inference as detailed in Sect. 2.3. For the I-Step trials, we simulated the true values of the inputs ω_I^* and measurement error σ_I from the above chosen priors. We perform 20 I-step trials within each of 10 T-Step trials resulting in a total of 200 SBC-trials.

To evaluate the calibration of the four methods graphically, we show the empirical cumulative distribution function (ECDF) difference plots (Säilynoja et al. 2022) in the upper part of Fig. 5. We choose two scenarios: high and low T-epistemic uncertainty, represented by the number of training points $N_T = \{5, 10\}$. In this graphical test, calibration is achieved if the black line lies within the blue region, i.e. the 95%-confidence envelops. We observe that, while high T-epistemic uncertainty is present, E-Post and E-Lik are almost calibrated, whereas Point and E-Log-Lik are overconfident. For $N_T = 8$ all methods show good calibration.

As an additional continuous calibration metric, we calculate the $\log(\gamma)$-statistic (Modrák et al. 2022) from our SBC results. We vary $N_T = \{5, 6, 7, 8, 9, 10\}$ to control for the amount of to-be-propagated T-epistemic uncertainty. The center part of Fig. 5 shows the corresponding results. As a general trend, we observe that E-Lik and E-Post are similarly calibrated. While first ($N_T = 5$) slightly miscalibrated, for $N_T > 5$ proper calibration is achieved. In contrast, E-Log-Lik and Point are both miscalibrated for $N_T < 7$, but with more training data, as the T-epistemic uncertainty diminishes, they also become well calibrated.

In the bottom of Fig. 5 we depict the sharpness (Gneiting et al. 2007; Bürkner et al. 2023) of the I-posteriors, here measured by the width of the 90 % CI of ω_I. For similarly calibrated methods, we say that the method with the smaller CI is sharper. We observe that Point and E-Log-Lik produce overall sharper results, but given their bad calibration, its clear that these two methods are just overconfident.

3.2.2 Polynomial surrogate model

Setup We now make the task substantially harder by not including the true simulator in the set of considered surrogate models. For this purpose, we use a polynomial chaos expansion (PCE) model as surrogate (Wiener 1938; Sudret 2008; Oladyshkin and Nowak 2012; Bürkner et al. 2023):

$$\widetilde{\mathcal{M}}(\omega; c) = \sum_{i=0}^{d} c_i \psi_i(\omega), \tag{45}$$

with the vector of surrogate coefficients $c = (c_0, ..., c_d)$, the maximum degree of polynomials d, and Legendre polynomials $\psi_i(\omega)$ (see Sudret (2008) for a detailed definition). We consider wide, independent normal priors for all surrogate coefficients: $c_i \sim \mathcal{N}(0, 5)$. In the following, we fix the max-

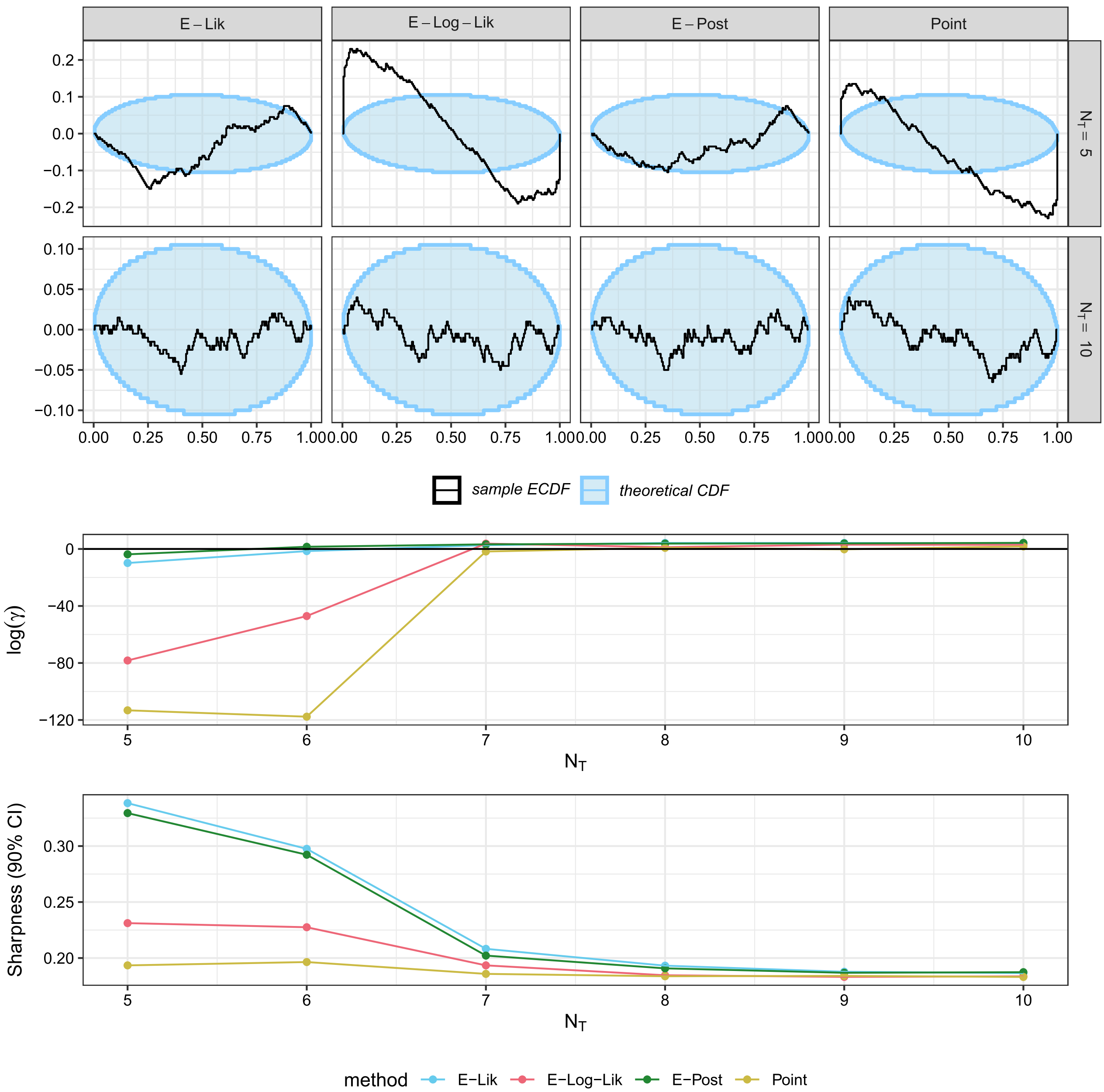

Fig. 5 Calibration and sharpness of the I-posteriors using logistic surrogate in case study 2. Top: ECDF difference plots for the I-posterior distributions of ω_I resulting from the four different methods. The blue areas in the ECDF difference plots indicate 95%-confidence envelopes and the black lines indicate the empirical cumulative distribution function (ECDF) for two different number of simulation points $N_T = \{5, 10\}$. Center: log-gamma-statistics of SBC with calibration threshold depicted as black horizontal line. Bottom: sharpness (90% CI) of I-posterior for four different I-Steps (colored dots/lines) for $N_T = \{5, 6, 7, 8, 9, 10\}$

imum polynomial degree to $d = 5$. By doing so, we create a scenario in which our surrogate is unable to fit the underlying true model appropriately such that an approximation error e_A is induced. This creates a scenario in which it is important to propagate both T-epistemic and T-aleatoric uncertainty.

Posterior distributions We perform the T- and I-Step, as previously described in Sect. 3.2.1. During the T-Step, we vary the number of training points $N_T = \{10, 20\}$ to control for the T-epistemic uncertainty. For the I-Step, we set the true input parameters to $\omega_I^* = -0.05$ and set $\sigma_S = 0.01$ to avoid sampling issues. We show the result of the T-Step in the

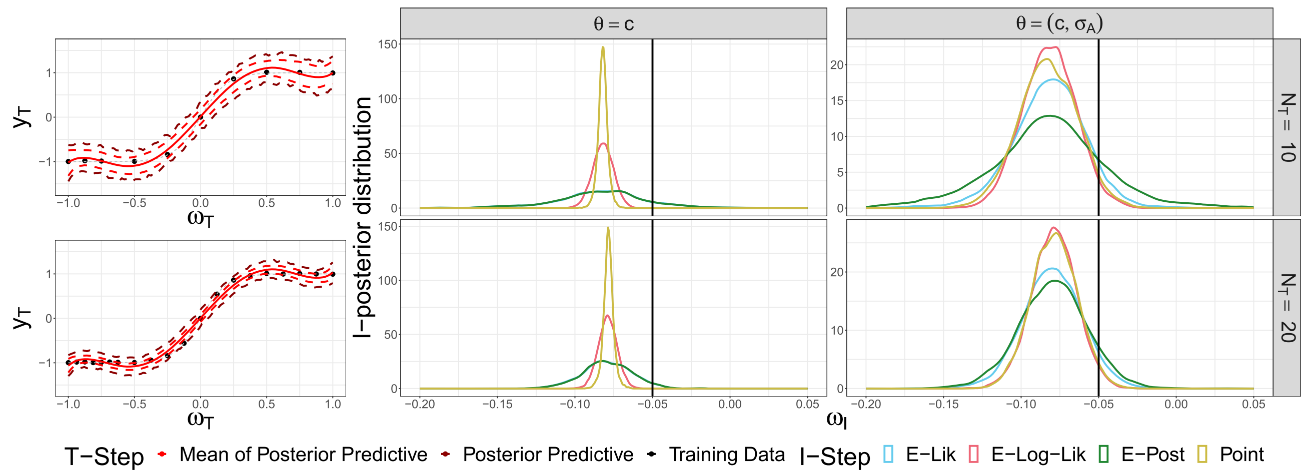

Fig. 6 Selected results for two-step procedure with PCE surrogate in case study 2. Left: For $N_T = \{10, 20\}$ the training data set $\mathcal{D}_T$ (black dots), the T-posterior predictive distribution and the mean of the T-posterior predictive distribution (dark red and red lines) is shown. Right: For the true input $\omega_I^* = -0.05$ (black vertical line), we depict the I-posterior distributions for each Point, E-Lik, E-Post, and E-Log-Lik (colored lines). In the center column we only propagate T-epistemic uncertainty via $\theta = c$ and in the right column we propagate both T-epistemic and T-aleatoric uncertainty via $\theta = (c, \sigma_A)$

left column of Fig. 6, where we present both the T-posterior predictive distribution (T-aleatoric and T-epistemic uncertainty) and the mean of the T-posterior predictive distribution (T-epistemic uncertainty only). The T-epistemic uncertainty becomes smaller with increasing number of training data points N_T, but the T-aleatoric uncertainty stays approximately constant.

In the middle column, we show the results of the I-posterior for the four methods when propagating only the T-epistemic uncertainty by considering only the T-posterior draws of the surrogate approximation parameters, i.e. $\theta = c$. For $N_T = 10$, the Point I-posterior and the E-Log-Lik I-posterior are narrow despite high T-epistemic uncertainty. In contrast, E-Lik and E-Post produce wider I-posteriors that are similar to each other. As the T-epistemic uncertainty reduces, all methods produce I-posterior distributions that converge towards a similar distribution.

In the right column, we show the I-posteriors when propagating both T-epistemic and T-aleatoric uncertainty by also propagating posterior draws of the surrogate approximation error: $\theta = (c, \sigma_A)$ (see Sect. 2.1.2). In general, all I-posteriors now tend to be wider than for $\theta = c$ and are more similar to each other. However, in the presence of substantial T-epistemic uncertainty (as for $N_T = 10$), E-Post produces the widest I-posterior, followed by E-Lik, Point, and E-Log-Lik. In Appendix C.3.1 we provide further results for different true parameter values.

Calibration We performed SBC for our PCE surrogate setup and show the results in Fig. 7. The top part shows the ECDF-difference plots for $N_T = 10$ for two cases: $\theta = c$ and $\theta = (c, \sigma_A)$. When only T-epistemic uncertainty is propagated, we see that only E-Lik and E-Post produce calibrated results, while E-Log-Lik and Point are overconfident. When T-aleatoric uncertainty is propagated as well, all calibrations improve, while E-Post still produces the best calibration results closely followed by E-Lik. In the bottom of Fig. 7 we compare the calibration via the $\log(\gamma)$-statistic of SBC under varying $N_T = \{10, 20, 30, 40, 50\}$ and confirm the observed pattern described above. As we use a surrogate model which was chosen on purpose to be inflexible and σ_A is modelled as constant over ω_I, we cannot achieve proper calibration with either method as epistemic uncertainty reduces.

3.3 Case study 3: uncertainty propagation in an SIR model

Finally, we apply our two-step procedure to a real-world case study in epidemiology. The SIR model and its variants are often used to mathematically describe the spread of infectious diseases (Hethcote 2000; Giordano et al. 2020). By considering this model, we demonstrate the applicability of the method to complex real-world problems that lack analytic solutions and require computationally expensive numerical methods. Our approach is particularly useful in such scenarios, as it allows to replace the complex simulation model while propagating relevant surrogate uncertainties.

Setup The SIR simulation model $\mathcal{M}$ is defined through the following system of differential equations:

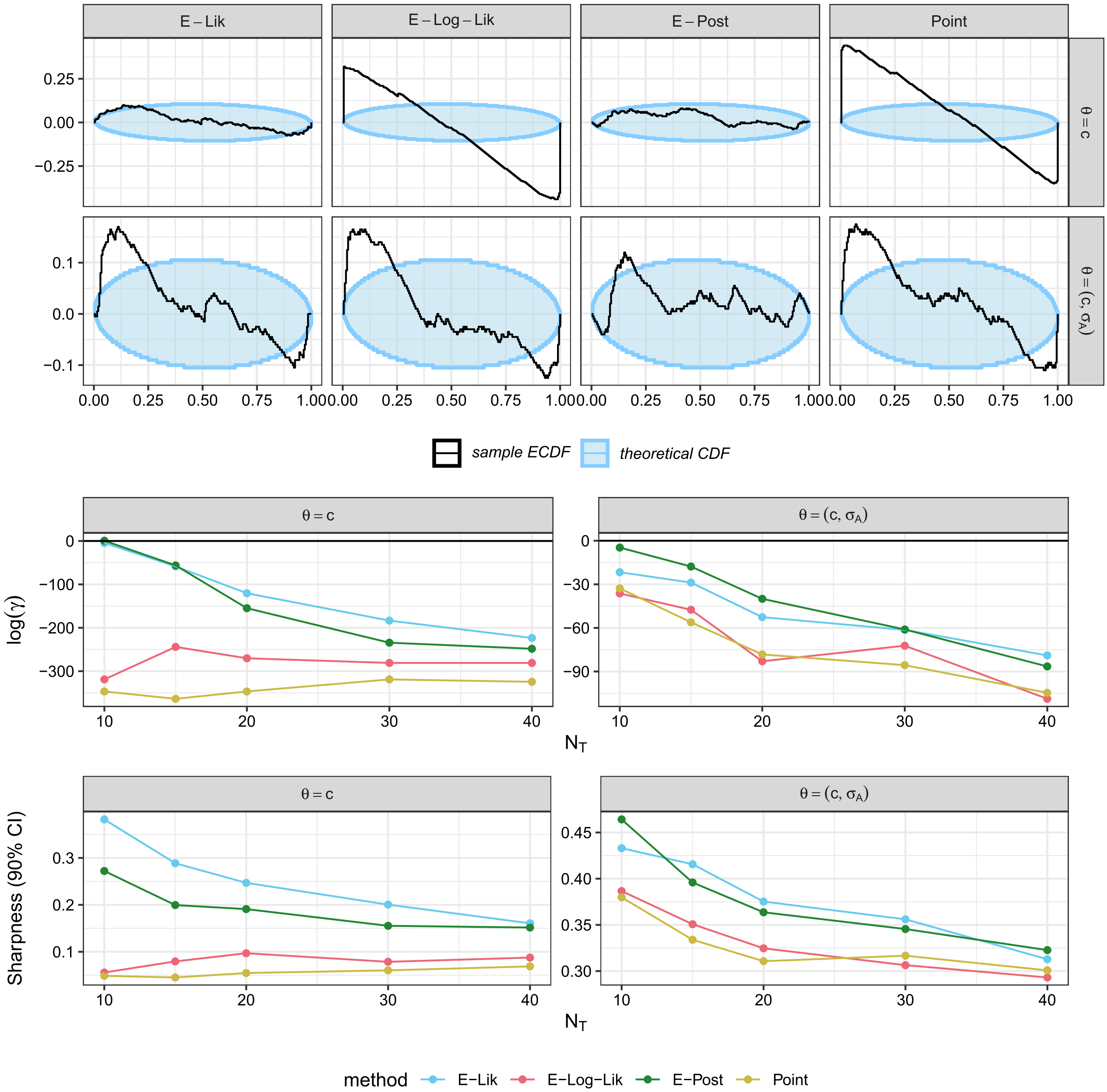

Fig. 7 Calibration and sharpness of the I-posteriors using PCE surrogate in case study 2. Top: ECDF difference plots for the I-posterior distributions of ω_I resulting from the four different methods. We set the number training points $N_T = 10$ and show the results for T-epistemic uncertainty propagation ($\theta = c$) and T-epistemic and T-aleatoric uncertainty propagation ($\theta = (c, \sigma_A)$). Center: log-gamma-statistics of SBC. Bottom: sharpness (90 % CI) of I-posterior for four different I-Steps (colored dots/lines) for $N_T = \{10, 20, 30, 40, 50\}$

$$
\begin{aligned}
\frac{\mathrm{d}S(t)}{\mathrm{d}t} &= -\beta S(t)\frac{I(t)}{N} \\
\frac{\mathrm{d}I(t)}{\mathrm{d}t} &= \beta S(t)\frac{I(t)}{N} - \gamma I(t) \\
\frac{\mathrm{d}R(t)}{\mathrm{d}t} &= \gamma I(t),
\end{aligned}
\tag{46}
$$

where $S(t)$ describes the number of susceptibles, $I(t)$ the number of infectives, and $R(t)$ the number of recovered individuals at time t. Furthermore, β describes the constant contact rate, γ the constant recovery rate, and N denotes the constant population. In the following, we set the constant population to $N = 763$, and fix the initial conditions to the exemplary values $I_0 = 1$, $S_0 = N - I_0$, $R_0 = 0$. To solve the differential equation defined in Eq. (46), we use the Dormand-Prince algorithm (Dormand and Prince 1980), a 4th/5th order Runge–Kutta method as implemented in Stan (Stan Development Team 2024).

Typically, measurement data is given for the number of infected individuals, i.e. $y = I(t)$. The unknown parameters to be inferred are $\omega = (\beta, \gamma)$. As a surrogate model, we consider again a PCE (see Sect. 3.2.2), this time with multivariate Legendre polynomials for the 3-dimensional input (t, β, γ). The maximum degree of the polynomials is set to 4, which leads to 34 unknown coefficients c by the standard truncation scheme (see Sudret (2008)). This setup is chosen to demonstrate the applicability of the method to arbitrary complex simulation and surrogate models, as long as samples can be drawn from the posteriors.

For the T-Step, we generate the training data set using a 3-dimensional Sobol sequence Sobol' (1967) with $N_T = 38$ input points $\{(t^{(i)}, \beta^{(i)}, \gamma^{(i)})\}_{i=1}^{N_T}$. The bounds of the input parameters are $t \in [1, 14]$, $\beta \in [1, 3]$, and $\gamma \in [0.1, 0.9]$. We scale all input parameters linearly to $[-1, 1]$, i.e. the standard scaling of the Legendre polynomials. The output $y_T^{(i)} = I(t^{(i)})$ is then obtained for a given time $t^{(i)}$, $\beta^{(i)}$, and $\gamma^{(i)}$ by solving Eq. (46). This creates a setup with a low simulation budget, leading to a high T-epistemic uncertainty.

The surrogate parameter priors are chosen in the same way as in 3.2.2. To learn the simulation model as efficiently as possible while still enforcing the constraint of a non-negative infection count, we choose a log-normal T-/I-likelihood. The T-model is fitted using NUTS with 4 chains of 1000 warmup and 250 post-warmup sampling iterations, resulting in a total of $S = 1000$ samples to propagate.

For the I-Step, we generate $N_I = 50$ measurements by sampling output responses $y_T^{(i)} \sim \text{NegBin}(I(t), \phi)$ given evenly spaced t, true input parameters ω_I^*, and the shape parameter $\phi = 9.6$. As priors we set $\beta \sim \mathcal{N}_{[1,3]}(2, 0.5^2)$ and $\gamma \sim \mathcal{N}_{[0.1,0.9]}(0.5, 0.25^2)$ in order to stay within the parameter bounds that were used to train the surrogate model and further set $\sigma_I \sim \text{Half-Normal}(0.5^2)$. The densities and parameterizations of the used distributions are provided in A.1.

In this setup, we propagate epistemic uncertainty contained in c through the $S = 1000$ T-posterior samples. For the Point, E-Lik, and E-Log-Lik methods, we sample from the I-posterior of ω_I and σ_I using NUTS with 32 chains, each running for 1000 warmup and 1000 post-warmup iterations, resulting in a total of 32000 I-posterior draws. Instead, for E-Post, where we fit $S = 1000$ separate models (one for each T-posterior draw), we use NUTS with 4 chains, each running for 250 iterations, resulting in a total of 10^6 posterior draws.

Results We set the underlying true input parameters to exemplary values $\omega_I^* = (\omega_{I,1}^*, \omega_{I,2}^*) = (\beta^*, \gamma^*) = (1.6, 0.4)$ and compare the I-posteriors resulting from the four different methods in Fig. 8. In the marginal I-posterior distribution plots, we observe that both the Point and E-Log-Lik methods produce an I-posterior that is overconfident in one dimension. We also observe that both the E-Lik and E-Post methods have substantial probability mass around the true values. In the scatter plots showing the joint I-posterior distribution, we observe that Point and E-Log-Lik produce samples that do not cover the true value ω_I^*, while E-Post and E-Lik do. The large uncertainty produced by these methods (E-Post and E-Lik) resembles the uncertainty in the T-posterior, as the surrogate was only trained on very limited data.

4 Conclusion

We introduced a general two-step procedure for surrogate-based Bayesian inference. Within our approach, we propagate all relevant surrogate uncertainties (both aleatoric and epistemic) from the surrogate training to the real-data inference step, thereby producing fully uncertainty aware inference on the real data. The uncertainty propagation methods developed within our framework are in principle agnostic to the chosen surrogate, but require the ability to sample from its training-step posterior given data generated from the simulator. To evaluate the uncertainty calibration of the resulting inference, we proposed an extension of simulation-based calibration suitable for our two-step surrogate approach.

As we demonstrate in our case studies, even in seemingly simple setups, complex behavior occurs in terms of posterior shape and calibration when propagating surrogate uncertainty. In particular two uncertainty propagation methods (E-Lik & E-Post) showed substantial improvements in uncertainty calibration compared to traditional surrogate-based inference that is uncertainty-unaware. What is more, our results show the importance of propagating the complete surrogate uncertainty (aleatoric and epistemic) instead of propagating only parts of it. Intuitively, one might expect that there is only one "correct" uncertainty propagation method within the bounds of probability theory. However, as we demonstrate, the two advocated methods for uncertainty propagation (E-Lik & E-Post) produce non-equivalent inference even in simple cases despite being equally justified by probability theory. That said, at least in our case studies, the produced inference was very similar. They differ in the computational requirements and ease of parallelization though (see Sect. 2.2.6), such that either or the other may be preferable depending the context and available resources.

Future Work Our surrogate-based Bayesian inference approach is agnostic to the input-parameter dimensionality and its scaling is unaffected by said dimensionality (but only by the number of draws propagated from the training to the inference step). In our case studies, we focused on simulators with up to three-dimensional input parameters in order to simplify the presentation and establish a better intuition

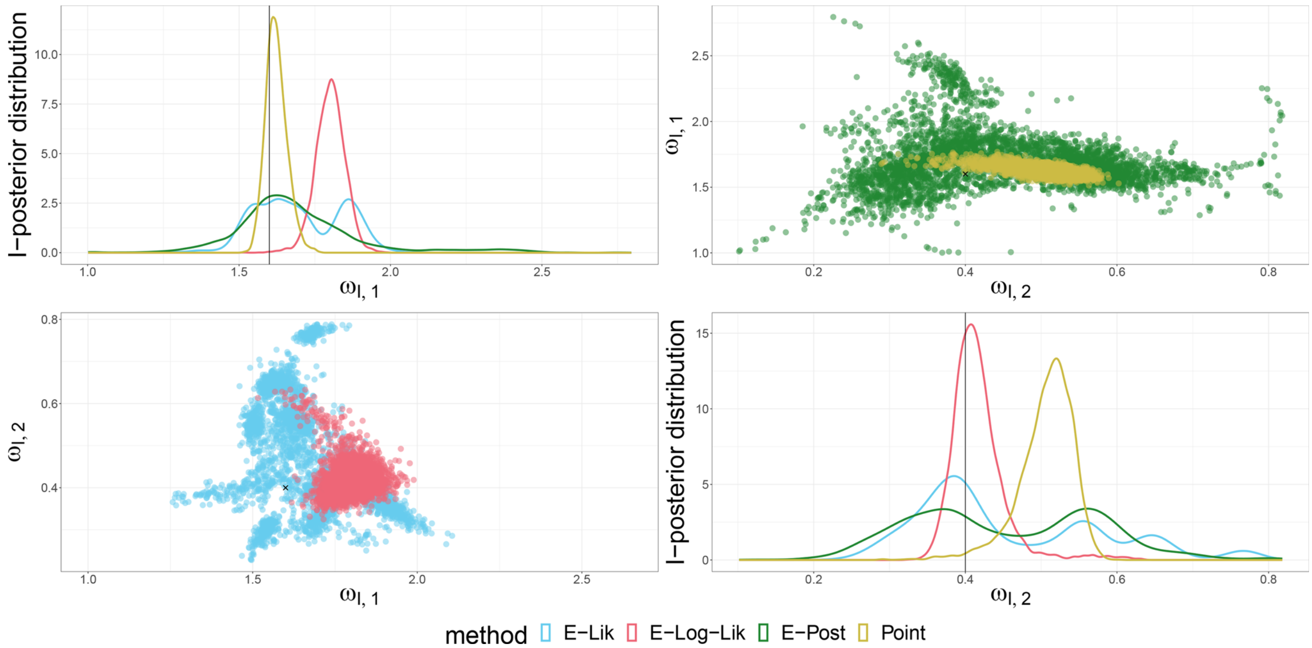

Fig. 8 Pairs plot of the I-posteriors using a PCE surrogate in case study 3. On the diagonals we depict for the true input $\omega_I^* = (\beta^*, \gamma^*) = (1.6, 0.4)$ (black vertical line) the marginal I-posterior distributions for the Point, E-Lik, E-Post, and E-Log-Lik methods (colored lines). The off-diagonals show the scatter plots of 5000 randomly subsampled I-posterior draws. We propagate T-epistemic uncertainty via $\theta = c$

about the overall approach. To study the applicability of our methods on high-dimensional challenges, biological systems requiring accurate, uncertainty-aware inference (Mitra and Hlavacek 2019) will offer an interesting class of problems for future research.

In our case studies, we modelled the surrogate approximation error to be constant, thus assuming the aleatoric uncertainty of the surrogate to be independent of the input parameters. However, this assumption is likely unjustified in practice, if there is substantial approximation error due to the surrogate inflexibility. Hence, one could improve the modeling of the surrogate approximation error by conditioning it on the input parameters, for example, as suggested in Kohlhaas et al. (2023). This will induce additional parameters to model the approximation error, which are then seamlessly propagated through our two-step procedure.

Our current implementation of uncertainty quantification relies on MCMC in both training and inference step. This may become computationally expensive, sometimes prohibitively so, as the number of (surrogate) model parameters grows (Izmailov et al. 2021). Hence, for a surrogate model with hundreds of thousands of parameters, full Bayesian UQ will become intractable and approximations are needed. Approaches like variational inference (Hinton et al. 1993; Graves 2011) and partial UQ on the last neural network layer (Kristiadi et al. 2020; Fiedler and Lucia 2023; Harrison et al. 2024) could be sensible alternatives. Our uncertainty propagation methods only require the ability to draw samples from an (approximate) posterior in the training step that can be subsequently passed to the inference step. This flexibility ensures that our approach is still applicable in scenarios where a full Bayesian UQ is infeasible, but the implication on inference validity, in particular uncertainty calibration, need to be further studied.

Appendix A. Notation

- ω: input
- y: output
- $\mathcal{M}$: simulator
- $\widetilde{\mathcal{M}}$: surrogate model
- c: surrogate approximation parameters
- d: number of surrogate parameters
- T-Step: surrogate training step
  - N_T: number of simulation training pairs
  - ω_T: simulation input
  - y_T: simulation output
  - e_S: simulation noise
  - σ_S: simulation noise hyperparamters
  - e_A: approximation error
  - σ_A: surrogate approximation error hyperparameters
  - $\theta = (c, \sigma_A)$: trainable surrogate parameters
  - S: number of T-posterior samples
- I-Step: surrogate inference step

- N_I: number of measurements
- ω_I: quantity of interest (QoI)
- y_I: measurement data
- e_I: measurement noise
- σ_I: measurement noise hyperparameters
- K: number of I-posterior samples

A.1 Negative binomial, log-normal, and half-normal distributions

Below, we show the Negative binomial, Log-normal and Half-normal distributions. The probability mass function of the Negative binomial distribution for scalar count $n \in \mathbb{N}$ with the two positive parameters $\mu \in \mathbb{R}^+$ and $\phi \in \mathbb{R}^+$ is given by:

$$p_{\text{NegBinom}}(n \mid \mu, \phi) = \binom{n+\phi-1}{n}\left(\frac{\mu}{\mu+\phi}\right)^n\left(\frac{\phi}{\mu+\phi}\right)^{\phi}. \tag{A1}$$

In this parameterization the mean and variance are given by

$$\mathbb{E}[n] = \mu \quad \text{and} \quad \text{Var}[n] = \mu + \frac{\mu^2}{\phi}. \tag{A2}$$

The probability density function of the Log-normal distribution for a positive scalar $y \in \mathbb{R}^+$ with the parameters $\mu \in \mathbb{R}$ and $\sigma \in \mathbb{R}^+$ is given by:

$$p_{\text{Log-Normal}}(y \mid \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma}\frac{1}{y}\exp\left(-\frac{1}{2}\left(\frac{\log y - \mu}{\sigma}\right)^2\right). \tag{A3}$$

In this parameterization the mean and variance are given by

$$\mathbb{E}[y] = \exp\left(\mu + \frac{\sigma^2}{2}\right) \quad \text{and} \tag{A4}$$

$$\text{Var}[y] = [\exp(\sigma^2) - 1]\exp(2\mu + \sigma^2). \tag{A5}$$

The probability density function of the Half-normal distribution for a positive scalar $y \in \mathbb{R}^+$ with the parameter $\sigma \in \mathbb{R}^+$ is given by:

$$p_{\text{Half-Normal}}(y \mid \sigma^2) = \frac{\sqrt{2}}{\sqrt{\pi}\sigma}\exp\left(-\frac{y^2}{2\sigma^2}\right). \tag{A6}$$

Appendix B: Derivations

B.1 Inequality of E-Lik and E-Post via counterexample

In the following, we show the inequality of E-Post and E-Lik (Main Sect. 2.2) using a counterexample with discrete random variables. We denote the T-posterior as $p(\theta)$ and omit the dependence on the training data $\mathcal{D}_T$ to simplify notation.

Let $\omega \in \{0, 1\}$, $y \in \{0, 1\}$, $\theta \in \{0, 1\}$. We set the probability values

$p(\omega = 0)$	1/2
$p(\omega = 1)$	1/2

$p(\theta = 0)$	1/2
$p(\theta = 1)$	1/2

	$\omega = 0, \theta = 0$	$\omega = 1, \theta = 0$
$p(y = 0 \mid \omega, \theta)$	1/4	1/2
$p(y = 1 \mid \omega, \theta)$	3/4	1/2

	$\omega = 0, \theta = 1$	$\omega = 1, \theta = 1$
$p(y = 0 \mid \omega, \theta)$	1/2	1/2
$p(y = 1 \mid \omega, \theta)$	1/2	1/2

First, we state the general formulation of E-Lik and E-Post for discrete pmf's:

$$p(\omega \mid y, u = \text{E-Lik}) = \frac{p(\omega)\sum_i p(y \mid \omega, \theta = i)p(\theta = i)}{\sum_i \sum_j p(y \mid \omega = j, \theta = i)p(\omega = j)p(\theta = i)}$$

$$p(\omega \mid y, u = \text{E-Post}) = \sum_i \frac{p(\omega)p(y \mid \omega, \theta = i)p(\theta = i)}{\sum_j p(y \mid \omega = j, \theta = i)p(\omega = j)} \times \frac{p(\omega = 0)p(y = 0 \mid \omega = 0, \theta = 1)p(\theta = 1)}{\sum_j p(y \mid \omega = j, \theta = 1)p(\omega = j)}$$

We calculate E-Post for $\omega = 0$ and $y = 0$ by plugging in the probability values:

$$p(\omega = 0 \mid y = 0, u = \text{E-Post}) = \sum_i \frac{p(\omega = 0)p(y = 0 \mid \omega = 0, \theta = i)p(\theta = i)}{\sum_j p(y = 0 \mid \omega = j, \theta = i)p(\omega = j)}$$
$$= \frac{1/2 \cdot 1/4 \cdot 1/2}{3/8} + \frac{1/2 \cdot 1/2 \cdot 1/2}{1/2} = \frac{1/16}{3/8} + \frac{1/8}{1/2}$$
$$= \frac{5}{12} \approx 0.417$$

Next, we calculate E-Lik for $\omega = 0$ and $y = 0$:

$$p(\omega = 0 \mid y = 0, u = \text{E-Lik}) = \frac{p(\omega = 0)\sum_i p(y = 0 \mid \omega = 0, \theta = i)p(\theta = i)}{\sum_i \sum_j p(y = 0 \mid \omega = j, \theta = i)p(\omega = j)p(\theta = i)}$$

$$= \frac{1/2 \cdot (1/4 \cdot 1/2 + 1/2 \cdot 1/2)}{7/16}$$
$$= \frac{3}{7} \approx 0.429.$$

We see that the results produced by E-Post and E-Lik for $\omega = 0$ and $y = 0$ are similar, but not equal.

The normalization constants for E-Post were given by:

$$\begin{aligned}
&\sum_j p(y \mid \omega = j, \theta = 0) p(\omega = j) \\
&= p(y = 0 \mid \omega = 0, \theta = 0) p(\omega = 0) \\
&\quad + p(y = 0 \mid \omega = 1, \theta = 0) p(\omega = 1) \\
&= 1/4 \cdot 1/2 + 1/2 \cdot 1/2 = 3/8
\end{aligned}$$

and

$$\begin{aligned}
&\sum_j p(y \mid \omega = j, \theta = 1) p(\omega = j) \\
&= p(y = 0 \mid \omega = 0, \theta = 1) p(\omega = 0) \\
&\quad + p(y = 0 \mid \omega = 1, \theta = 1) p(\omega = 1) \\
&= 1/2 \cdot 1/2 + 1/2 \cdot 1/2 = 1/2.
\end{aligned}$$

The normalization constant for E-Lik was given by:

$$\begin{aligned}
&p_{\text{norm}}(y = 0) \\
&= \sum_i \sum_j p(y = 0 \mid \omega = j, \theta = i) p(\omega = j) p(\theta = i) \\
&= p(y = 0 \mid \omega = 0, \theta = 0) p(\omega = 0) p(\theta = 0) \\
&\quad + p(y = 0 \mid \omega = 1, \theta = 0) p(\omega = 1) p(\theta = 0) \\
&\quad + p(y = 0 \mid \omega = 0, \theta = 1) p(\omega = 0) p(\theta = 1) \\
&\quad + p(y = 0 \mid \omega = 1, \theta = 1) p(\omega = 1) p(\theta = 1) \\
&= 1/4 \cdot 1/2 \cdot 1/2 + 1/2 \cdot 1/2 \cdot 1/2 \\
&\quad + 1/2 \cdot 1/2 \cdot 1/2 + 1/2 \cdot 1/2 \cdot 1/2 = 7/16
\end{aligned}$$

B.2 Equivalence of E-Log-Lik and E-Log-Post

Here we show that the two formulations E-Log-Post and E-Log-Lik (Main Sect. 2.2) are equivalent. The E-Log-Lik is defined as:

$$\begin{aligned}
&\log(p(\omega_I \mid y_I, u = \text{E-Log-Lik})) \\
&\propto \log(p(\omega_I)) + \int \log(p(y_I \mid \omega_I, \theta)) p(\theta \mid \mathcal{D}_T) d\theta
\end{aligned}$$

Similarly, to E-Post and E-Lik we can define the E-Log-Post:

$$\begin{aligned}
&\log(p(\omega_I \mid y_I, u = \text{E-Log-Post})) \\
&= \int \log(p(\omega_I \mid y_I)) p(\theta \mid y_T) d\theta \\
&= \int \left[\log(p(\omega_I)) + \log(p(y_I \mid \omega_I, \theta) - \log(C(\theta))\right] \\
&\quad \times p(\theta \mid y_T) d\theta \\
&= \log(p(\omega_I)) \int p(\theta \mid y_T) d\theta \\
&\quad + \int \log(p(y_I \mid \omega_I, \theta)) p(\theta \mid y_T) d\theta \\
&\quad - \int \log(C(\theta)) p(\theta \mid y_T) d\theta \\
&\propto \log(p(\omega_I)) + \int \log(p(y_I \mid \omega_I, \theta)) p(\theta \mid y_T) d\theta
\end{aligned}$$

In the last equation we used that the integral over a probability distribution is one, i.e. $\int p(\theta \mid y_T) d\theta = 1$ and since the posterior is a function of ω_I we can define a new constant $C_1(\theta) := -\int \log(C(\theta)) p(\theta \mid y_T) d\theta$.

B.3 Case study 1: slope intercept model

In this section we derive the E-Log-Lik for the slope intercept model stated in Main Sect. 3.1. Let $c = [c_1, c_2]^T$,

$$\Omega_T = \begin{bmatrix} 1 & \omega_T^{(1)} \\ 1 & \omega_T^{(2)} \\ \vdots & \vdots \\ 1 & \omega_T^{(N_T)} \end{bmatrix},$$

$$\hat{\omega}_I = [1, \omega_I]^T, \mu_{T1} = [\mu_{T1}^{(1)}, \mu_{T2}^{(2)}]^T,$$

$$\Sigma_{T1} = \begin{bmatrix} \Sigma_{T1}^{(1,1)} & \Sigma_{T1}^{(1,2)} \\ \Sigma_{T1}^{(2,1)} & \Sigma_{T1}^{(2,2)} \end{bmatrix},$$

B.3.1 E-Log-Lik

Now we derive the stated Expected-Log-Likelihood result:

$$\begin{aligned}
&p(\omega_I \mid y_I, u = \text{E-Log-Lik}) \\
&\propto p(\omega_I) \exp\left\{\int \log(p(y_I \mid \omega_I, c)) p(c \mid \mathcal{D}_T) dc\right\} \\
&= p(\omega_I) \exp\left\{\int \log(\mathcal{N}(y_I \mid \hat{\omega}_I^\top c, \sigma_I^2))\right. \\
&\quad \left.\times \mathcal{N}(c \mid \mu_{T1}, \Sigma_{T1}) dc\right\} \\
&= p(\omega_I) \exp\left\{\int -\frac{1}{2\sigma_I^2} (\hat{\omega}_I^\top c - y_I)^2\right. \\
&\quad \left.\times \mathcal{N}(c \mid \mu_{T1}, \Sigma_{T1}) dc\right\} \\
&= p(\omega_I) \exp\left\{-\frac{1}{2\sigma_I^2} \int (y_I^2 - 2 y_I \hat{\omega}_I^\top c + c^\top \hat{\omega}_I \hat{\omega}_I^\top c)\right. \\
&\quad \left.\times \mathcal{N}(c \mid \mu_{T1}, \Sigma_{T1}) dc\right\}
\end{aligned}$$

$$
\begin{aligned}
&= p(\omega_I) \exp\left\{-\frac{1}{2\sigma_I^2}(y_I^2 - 2y_I\hat{\omega}_I^\top \mathbb{E}[c]\right.\\
&\quad \left. +\mathbb{E}[c^\top \hat{\omega}_I\hat{\omega}_I^\top c])\right\}\\
&= p(\omega_I) \exp\left\{-\frac{1}{2\sigma_I^2}(y_I^2 - 2y_I\hat{\omega}_I^\top \mu_{T1}\right.\\
&\quad \left. +\mathrm{Tr}(\hat{\omega}_I\hat{\omega}_I^\top \Sigma_{T1}) + \mu_{T1}^\top\hat{\omega}_I\hat{\omega}_I^\top \mu_{T1})\right\}\\
&\quad \propto p(\omega_I) \exp\left\{-\frac{1}{2\sigma_I^2}(y_I^2 - 2y_I\hat{\omega}_I^\top \mu_{T1} + \omega_I^2\Sigma_{T1}^{(2,2)}\right.\\
&\qquad + \omega_I(\Sigma_{T1}^{(2,1)} + \Sigma_{T1}^{(1,2)}) + \mu_{T1}^{(2)}\mu_{T1}^{(2)}\omega_I^2\\
&\qquad \left. +2\mu_{T1}^{(1)}\mu_{T1}^{(2)}\omega_I)\right\}\\
&\propto p(\omega_I) \exp\left\{-\frac{1}{2\sigma_I^2}(\mu_{T1}^{(2)}\mu_{T1}^{(2)} + \Sigma_{T1}^{(2,2)})\right.\\
&\quad \left.\times(\omega_I - \frac{y\mu_{T1}^{(2)} - \Sigma_{T1}^{(1,2)} - \mu_{T1}^{(1)}\mu_{T1}^{(2)}}{\mu_{T1}^{(2)}\mu_{T1}^{(2)} + \Sigma_{T1}^{(2,2)}})^2\right\}\\
&\propto \mathcal{N}(\omega_I \mid \mu_{I1}, \sigma_{I1}^2),
\end{aligned}
$$

with

$$
\begin{aligned}
\sigma_{I1}^2 &= (\sigma_{I0}^{-2} + \sigma_I^{-2}(\mu_{T1}^{(2)}\mu_{T1}^{(2)} + \Sigma_{T1}^{(2,2)}))^{-1},\\
\mu_{I1} &= \sigma_{I1}^2(\sigma_{I0}^{-2}\mu_{I0} + \sigma_I^{-2}(\mu_{T1}^{(2)}y_I - \Sigma_{T1}^{(1,2)} - \mu_{T1}^{(1)}\mu_{T1}^{(2)}))
\end{aligned}
$$

where we used (Petersen and Pedersen 2012, p. 43):

$$
\begin{aligned}
&\mathrm{Tr}(\hat{\omega}_I\hat{\omega}_I^\top \Sigma_{T1})\\
&= \omega_I^2\Sigma_{T1}^{(2,2)} + \omega_I(\Sigma_{T1}^{(2,1)} + \Sigma_{T1}^{(1,2)}) + \Sigma_{T1}^{(1,1)}
\end{aligned}
$$

and

$$
\begin{aligned}
&\mu_{T1}^\top\hat{\omega}_I\hat{\omega}_I^\top \mu_{T1}\\
&= \mu_{T1}^{(2)}\mu_{T1}^{(2)}\omega_I^2 + 2\mu_{T1}^{(1)}\mu_{T1}^{(2)}\omega_I + \mu_{T1}^{(1)}\mu_{T1}^{(1)}
\end{aligned}
$$

Appendix C. Further results

C.1 Case study 1: slope only model

In Main Sect. 3.1 we showed the I-posteriors for the linear model with a slope and an intercept. In this section, we consider an even simpler setup, where the simulator and surrogate model are both a slope only model:

$$y = \mathcal{M}(\omega) = \widetilde{\mathcal{M}}(\omega, c) = c \cdot \omega, \tag{C7}$$

where $\omega \in \mathbb{R}$ is the input, $y \in \mathbb{R}$ the output and $c \in \mathbb{R}$ is the unknown parameter. We consider to propagate only the surrogate approximation parameter, i.e. we set $\theta = c$. We set normal priors and normal likelihoods. ***T-Step*** As our simulation data set we use one input–output pair $\mathcal{D}_T = \{\omega_T, y_T\}$ and $N_T = 1$. We set a normal prior on c with mean μ_{T0} and variance σ_{T0}^2. We calculate the posterior using the conjugate prior relation for a normal-normal model (Murphy 2007, 2012) (Fig. 9):

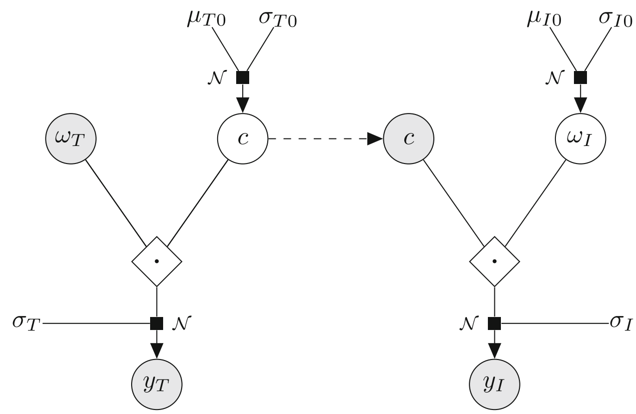

Fig. 9 Two-Step-Procedure for linear model with normal likelihoods and normal priors

$$
\begin{aligned}
p(c \mid y_T) &= \frac{p(c)p(y_T \mid c)}{p(y_T)} \qquad\qquad (C8)\\
&= \mathcal{N}(c \mid \mu_{T1}, \sigma_{T1}^2),\\
\sigma_{T1}^2 &= (\sigma_{T0}^{-2} + \sigma_A^{-2}\omega_T^2)^{-1}, \qquad\qquad (C9)\\
\mu_{T1} &= \sigma_{T1}^2(\sigma_{T0}^{-2}\mu_{T0} + \sigma_A^{-2}\omega_T y_T)
\end{aligned}
$$

I-Step: Point

We set a normal prior on ω_I:

$$p(\omega_I) = \mathcal{N}(\omega_I \mid \mu_{I0}, \sigma_{I0}^2). \tag{C10}$$

We use the following normal likelihood:

$$p(y_I \mid \omega_I, c) = \mathcal{N}(y_I \mid c \cdot \omega_I, \sigma_I^2) \tag{C11}$$

We compute the mean of the T-Step posterior: $\bar{c} = \mu_{T1}$ Now we can compute the Point I-posterior using the conjugate prior relation (Murphy 2012):

$$
\begin{aligned}
&p(\omega_I \mid y_I, u = \text{Point})\\
&\propto p(\omega_I)p(y_I \mid \omega_I, \bar{c})\\
&= \mathcal{N}(\omega_I \mid \mu_{I0}, \sigma_{I0}^2)\mathcal{N}(y_I \mid \mu_{T1} \cdot \omega_I, \sigma_I^2)\\
&\quad = \mathcal{N}(\omega_I \mid \mu_{I1}, \sigma_{I1}^2)\\
&\quad \text{, with:}\\
&\sigma_{I1}^2 = (\sigma_{I0}^{-2} + \sigma_I^{-2}\mu_{T1}^2)^{-1},\\
&\mu_{I1} = \sigma_{I1}^2(\sigma_{I0}^{-2}\mu_{I0} + \sigma_I^{-2}\mu_{T1}y_I)
\end{aligned}
$$

I-Step: E-Lik Derivation

$$
\begin{aligned}
&p(\omega_I \mid y_I, u = \text{E-Lik}) \\
&\propto p(\omega_I) \int p(y_I \mid \omega_I, c) p(c \mid \mathcal{D}_T) dc \\
&= \mathcal{N}(\omega_I \mid \mu_{I0}, \sigma_{I0}^2) \\
&\quad \times \int \mathcal{N}(y_I \mid \omega_I \cdot c, \sigma_I^2) \mathcal{N}(c \mid \mu_{T1}, \sigma_{T1}^2) dc \\
&= \mathcal{N}(\omega_I \mid \mu_{I0}, \sigma_{I0}^2) \mathcal{N}(y_I \mid \omega_I \cdot \mu_{T1}, \omega_I^2 \cdot \sigma_{T1}^2 + \sigma_I^2)
\end{aligned}
$$

I-Step: E-Log-Lik

$$
\begin{aligned}
&p(\omega_I \mid y_I, u = \text{E-Log-Lik}) \\
&\propto p(\omega_I) \exp\left\{ \int \log(p(y_I \mid \omega_I, c)) p(c \mid \mathcal{D}_T) dc \right\} \\
&= \mathcal{N}(\omega_I \mid \mu_{I0}, \sigma_{I0}^2) \exp\left\{ \int \log(\mathcal{N}(y_I \mid \omega_I \cdot c, \sigma_I^2)) \right. \\
&\quad \left. \times \mathcal{N}(c \mid \mu_{T1}, \sigma_{T1}^2) dc \right\} \\
&\propto \mathcal{N}(\omega_I \mid \mu_{I0}, \sigma_{I0}^2) \exp\left\{ \int \left(\frac{1}{2\sigma_I^2} (y_I - \omega_I c)^2 \right) \right. \\
&\quad \left. \times \mathcal{N}(c \mid \mu_{T1}, \sigma_{T1}^2) dc \right\} \\
&\quad\quad = \mathcal{N}(\omega_I \mid \mu_{I1}, \sigma_{I1}^2) \\
&\quad\quad \text{, with:} \\
&\sigma_{I1}^2 = (\sigma_{I0}^{-2} + \sigma_I^{-2}(\mu_{T1}^2 + \sigma_{T1}^2))^{-1}, \\
&\mu_{I1} = \sigma_{I1}^2(\sigma_{I0}^{-2}\mu_{I0} + \sigma_I^{-2}\mu_{T1} y_I)
\end{aligned}
$$

I-Step: E-Post

$$
\begin{aligned}
&p(\omega_I \mid y_I, u = \text{E-Post}) \\
&= p(\omega_I) \int \frac{p(y_I \mid \omega_I, c) p(c \mid \mathcal{D}_T)}{\int p(y_I \mid \omega_I, c) p(\omega_I) d\omega_I} dc \\
&= \mathcal{N}(\omega_I \mid \mu_{I0}, \sigma_{I0}^2) \\
&\quad \times \int \frac{\mathcal{N}(y_I \mid \omega_I \cdot c, \sigma_I^2) \mathcal{N}(c \mid \mu_{T1}, \sigma_{T1}^2)}{\mathcal{N}(y_I \mid c \cdot \mu_{I0}, c^2 \cdot \sigma_{I0}^2 + \sigma_I^2)} dc
\end{aligned}
$$

In Fig. 10 we compare the four uncertainty propagation methods for exemplary parameters. To control for T-epistemic uncertainty we vary σ_A. In general, we notice similar I-posteriors to the slope-intercept model (see Main Sect. 3.1). Additionally, for large σ_A we observe that both E-Lik and E-Post produce bimodal I-posteriors.

C.1.1 Derivation E-Log-Lik

We derive the Expected-Log-Likelihood result for the slope-only model:

$$
\begin{aligned}
&p(\omega_I \mid y_I, u = \text{E-Log-Lik}) \\
&\propto p(\omega_I) \exp\left\{ \int \log(p(y_I \mid \omega_I, c)) p(c \mid \mathcal{D}_T) dc \right\} \\
&\quad \propto \mathcal{N}(\omega_I \mid \mu_{I1}, \sigma_{I1}^2) \\
&\quad\quad \text{, with:} \\
&\quad \sigma_{I1}^2 = (\sigma_{I0}^{-2} + \sigma_I^{-2}(\mu_{T1}^2 + \sigma_{T1}^2))^{-1}, \\
&\quad \mu_{I1} = \sigma_{I1}^2(\sigma_{I0}^{-2}\mu_{I0} + \sigma_I^{-2}\mu_{T1} y_I)
\end{aligned}
$$

where we used:

$$
\begin{aligned}
&\int \log(p(y_I \mid \omega_I, c)) p(c \mid \mathcal{D}_T) dc \\
&= \int \log(\mathcal{N}(y_I \mid \omega_I c, \sigma_I^2)) \mathcal{N}(c \mid \mu_{T1}, \sigma_{T1}) dc \\
&= \int -\frac{1}{2\sigma_I^2} (\omega_I c - y_I)^2 \mathcal{N}(c \mid \mu_{T1}, \sigma_{T1}) dc \\
&= -\frac{1}{2\sigma_I^2} \int (y_I^2 - 2 y_I \omega_I c + c^2 \omega_I^2) \\
&\quad \times \mathcal{N}(c \mid \mu_{T1}, \sigma_{T1}) dc \\
&= -\frac{1}{2\sigma_I^2} (y_I^2 - 2 y_I \omega_I \mathbb{E}[c] + \omega_I \mathbb{E}[c^2]) \\
&= -\frac{1}{2\sigma_I^2} (y_I^2 - 2 y_I \omega_I \mu_{T1} + \mu_{T1}^2 + \sigma_{T1}^2) \\
&\propto -\frac{1}{2\sigma_I^2} (\mu_{T1}^2 + \sigma_{T1}^2)(\omega_I - \frac{\mu_{T1} y_I}{\mu_{T1}^2 + \sigma_{T1}^2})^2
\end{aligned}
$$

C.2 Case study 2: logistic model

C.3 T-posterior distribution

In Fig. 11 show the T-posterior pairs plots using the logistic surrogate model ((same setup as in Main Sect. 3.2.1) for $N_T = 7$.

C.3.1 PCE posterior distributions

We show I-posterior density plots using the PCE surrogate to approximate the logistic function (same setup as in Main Sect. 3.2) for additional true input parameters $\omega_I^* = \{-0.05, 0.1, 0.4\}$. In Fig. 12 we propagate only epistemic uncertainty and in Fig. 13 we propagate both epistemic and aleatoric uncertainty.

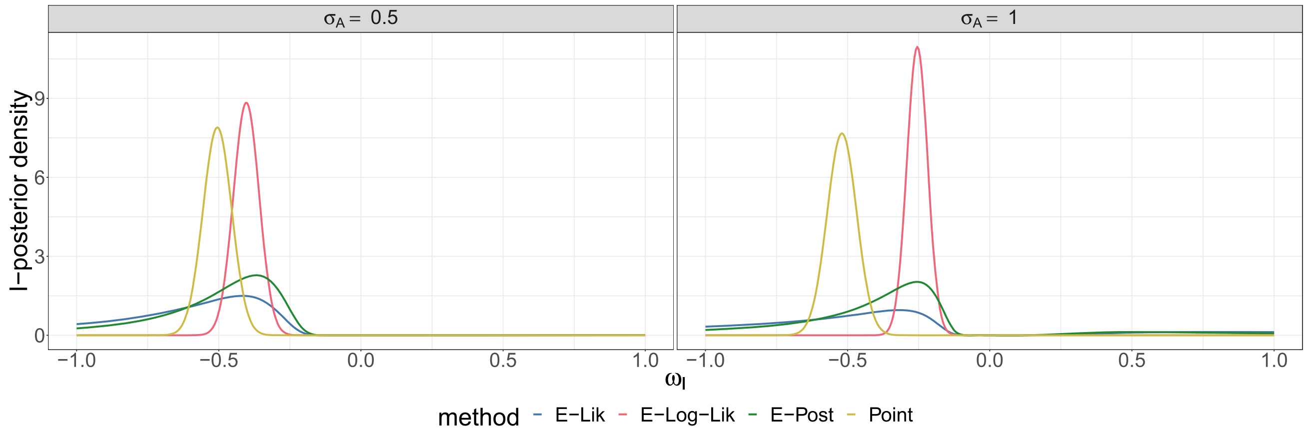

Fig. 10 I-posterior distributions for the slope-only surrogate with normal priors/likelihoods. We used exemplary hyperparameters and data. We use the four UPs to compuete the I-posterior while varying σ_A

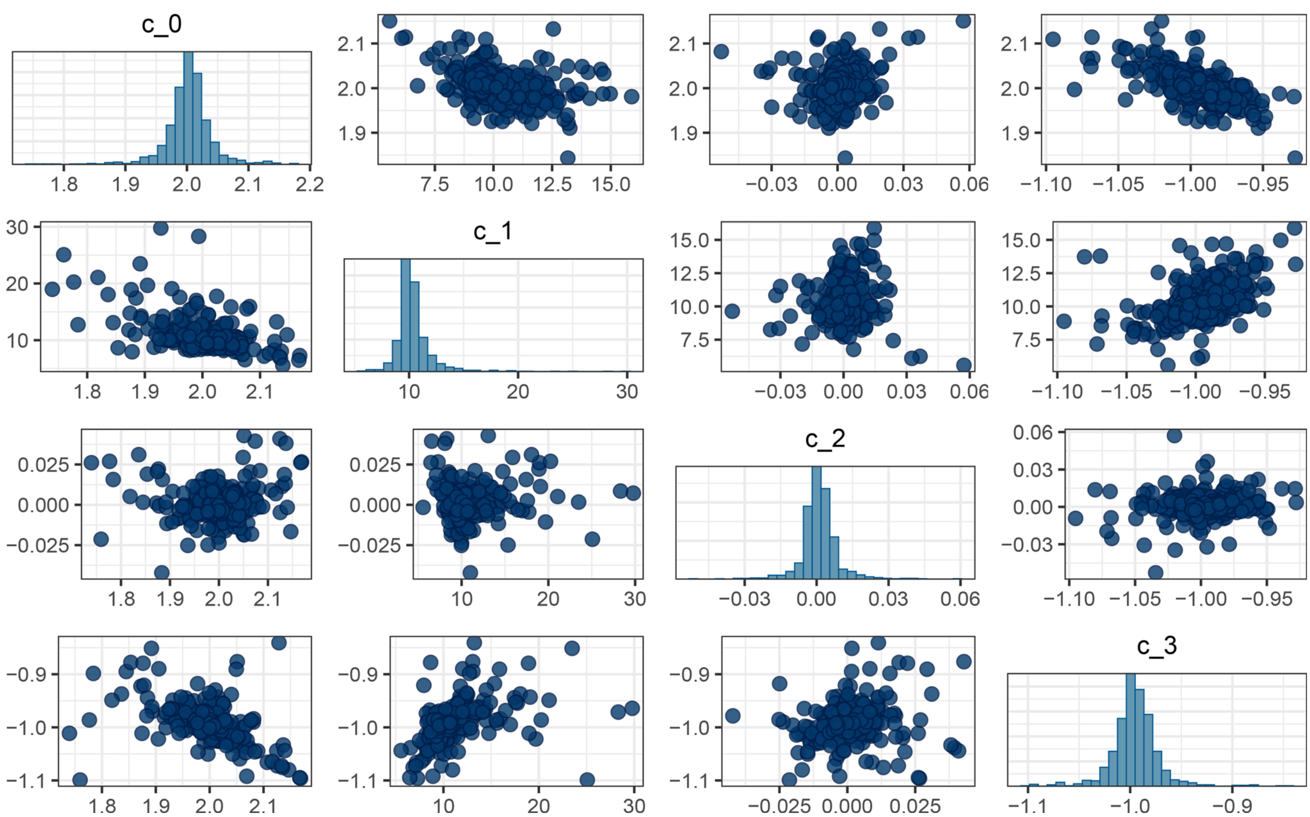

Fig. 11 Pairs plot of the T-posterior draws of the logistic surrogate in Case study 2 for $N_T = 7$

C.3.2 Computational complexity

To evaluate the computational complexity of the different UP methods, we estimate their sampling times using a PCE surrogate approximating a logistic function (see Main Sect. 3.2). For each UP method, we sample 4 chains with 1000 warmup iterations and 2000 total iterations. However, for a fair comparison, for E-Post we run for each model a full warmup phase with 1000 iterations, but sample only $2000/K$-times. The computations were not performed in parallel. We vary the maximum degree of polynomials $d \in \{2, 5, 9, 25\}$ and the number of clusters $K \in \{2, 5, 25, 100, 250, 500, 750, 1000\}$. We plot the times on a logarithmic scale in Fig. 14.

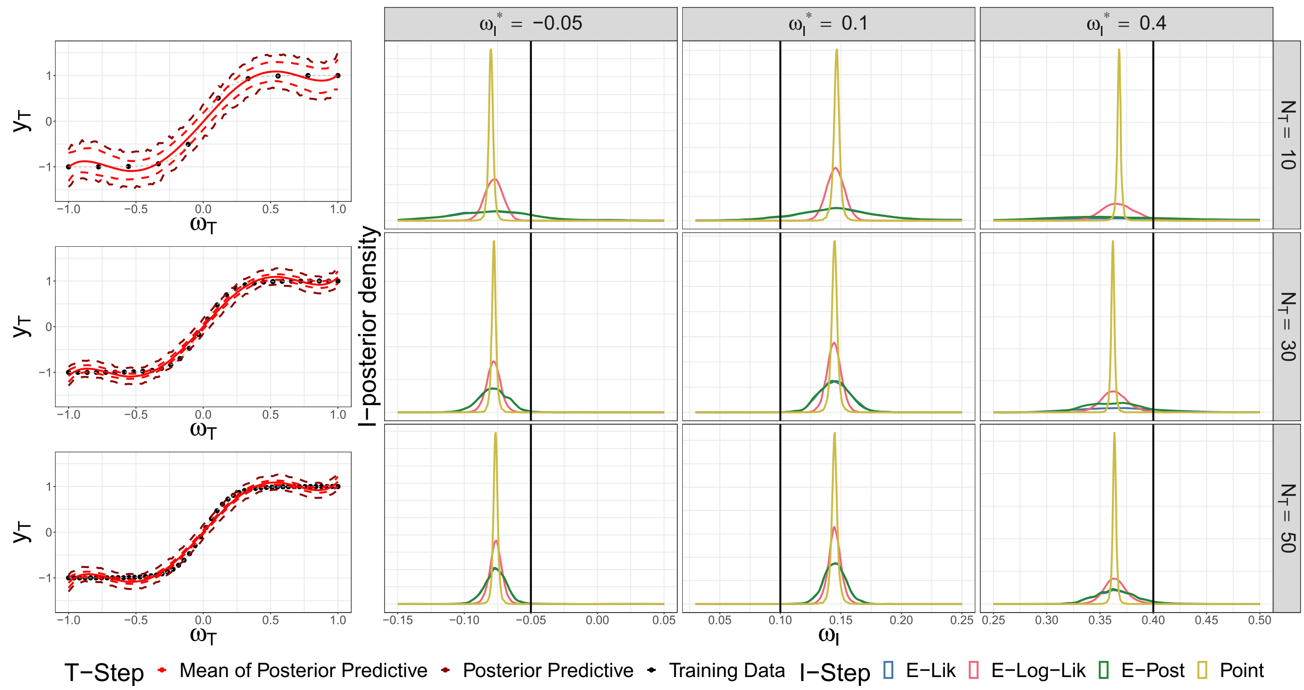

Fig. 12 Additional results for T-epistemic UP ($\theta = c$) with PCE surrogate in Case study 2. Left: For $N_T = \{10, 30, 50\}$ the training data set $\mathcal{D}_T$ (black dots), the T-posterior predictive distribution and the mean of the T-posterior predictive distribution (dark red and red lines) is shown. Right: For the true inputs $\omega_I^* = \{-0.05, 0.1, 0.4\}$ (black vertical lines) we depict the I-posterior distributions for each Point, E-Lik, E-Post, and E-Log-Lik (colored lines)

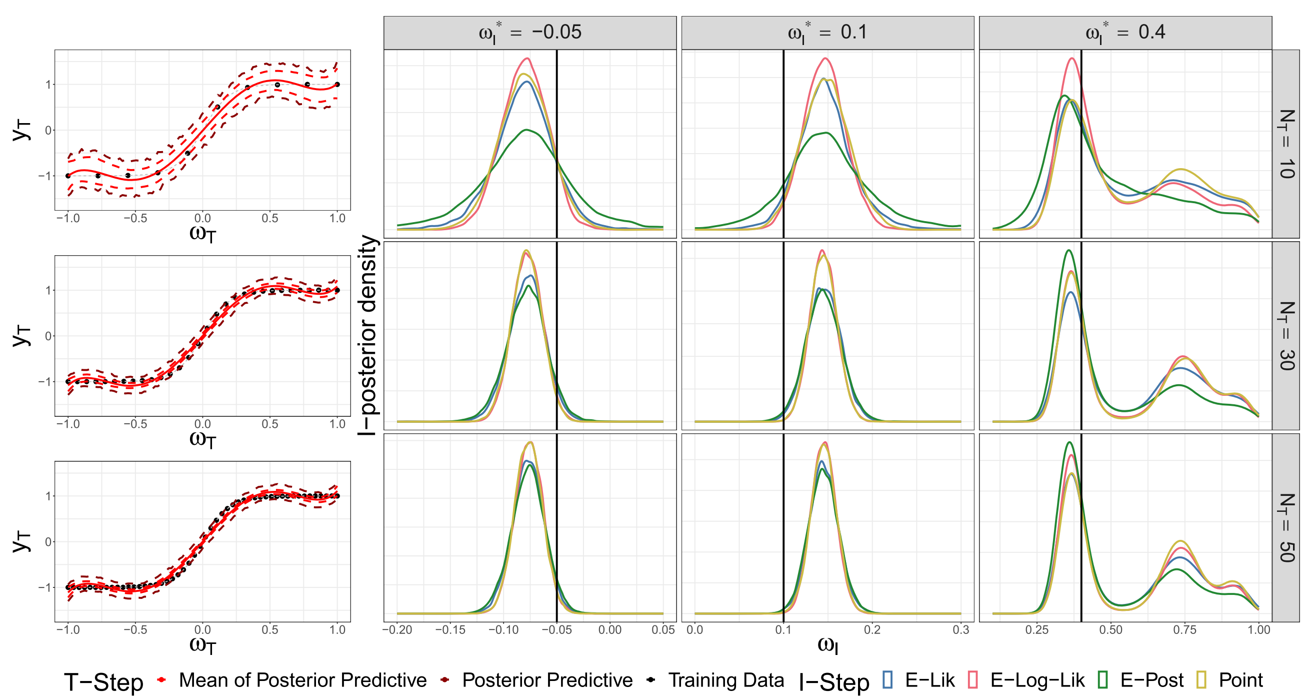

Fig. 13 Additional results for T-epistemic and T-aleatoric UP ($\theta = (c, \sigma_A)$) with PCE surrogate in Case study 2. Left: For $N_T = \{10, 30, 50\}$ the training data set $\mathcal{D}_T$ (black dots), the T-posterior predictive distribution and the mean of the T-posterior predictive distribution (dark red and red lines) is shown. Right: For the true inputs $\omega_I^* = \{-0.05, 0.1, 0.4\}$ (black vertical lines) we depict the I-posterior distributions for each Point, E-Lik, E-Post, and E-Log-Lik (colored lines)

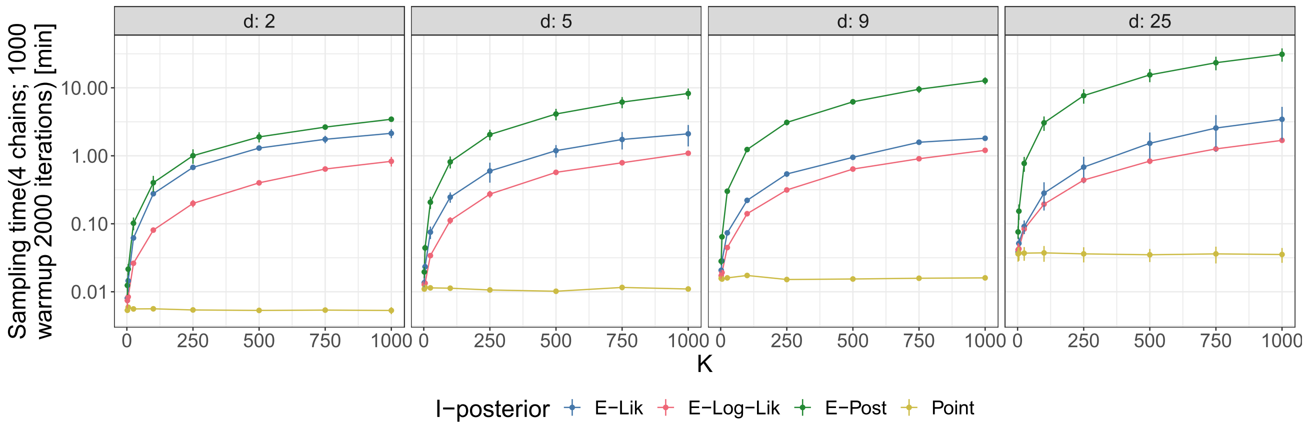

Fig. 14 Sampling times estimated for Point, E-Lik, E-Log-Lik and E-Post shown on a logarithmic scale. For each method, we sample 4 chains with 1000 warmup iterations and 2000 total iterations. We vary the maximum degree of polynomials $d \in \{2, 5, 9, 25\}$ and the number of clusters $K \in \{2, 5, 25, 100, 250, 500, 750, 1000\}$

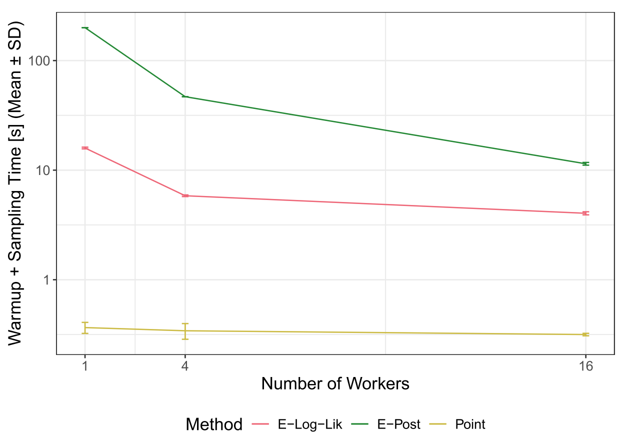

Fig. 15 Effect of parallelization on the warmup and sampling times. The runtimes for E-Log-Lik, E-Post, and Point are shown on a logarithmic scale

C.3.3 Parallelization

To examine the effectiveness of parallelization, we measure the runtime for different numbers of workers, comparing the warmup and sampling times across the Point, E-Log-Lik, and E-Post methods. We use a PCE surrogate to approximate the logistic simulator, as described in Section 3.2.2. The Point method serves as a baseline, does not propagate uncertainty, and is not parallelized. To compute the I-posteriors using the E-Log-Lik and the E-Post method, we propagate $S = 500$ T-posterior samples.

We repeat the experiment three times and report the mean sampling time along with the standard deviation. The number of workers is varied across $\{1, 4, 16\}$. For each uncertainty propagation method, we run a single MCMC chain with 1000 warm-up and 1000 post-warmup iterations. Figure 15 presents the results on a logarithmic scale.

This plot confirms that the runtime for both E-Log-Lik and E-Post decreases as the number of workers increases. Additionally, we observe that E-Post benefits more from parallelization than E-Log-Lik, which aligns with the expected degree of parallelizability of these methods as discussed in Section 2.2.6. Due to implementation constraints, we exclude E-Lik from this analysis, as its use of the `log-sum-exp` trick for numerical stability currently limits straightforward parallelization. However, given the similarities in implementation between E-Lik and E-Log-Lik, we expect that the conclusions drawn for E-Log-Lik also apply to E-Lik.

Acknowledgements Partially funded by Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy EXC 2075 - 390740016 and DFG Project 500663361. We acknowledge the support by the Stuttgart Center for Simulation Science.

Author Contributions P.B. and A.G. conceived of the idea. P.R., A.G. and P.B. developed the idea. P.R. designed and carried out the experiments and case studies. P.R. implemented the majority of the software code. J.A. helped with the derivations for case study 1. P.R. wrote the manuscript and J.A., A.G. and P.B. reviewed and edited the manuscript.

Funding Open Access funding enabled and organized by Projekt DEAL.

References

Alden, K., Cosgrove, J., Coles, M., Timmis, J.: Using emulation to engineer and understand simulations of biological systems. IEEE/ACM Trans. Comput. Biol. Bioinf. **17**(1), 302–315 (2020)

Austin, P.C., White, I.R., Lee, D.S., van Buuren, S.: Missing data in clinical research: A tutorial on multiple imputation. Can. J. Cardiol. **37**(9), 1322–1331 (2021)

Bayarri, M.J., Berger, J.O., Liu, F.: Modularization in Bayesian analysis, with emphasis on analysis of computer models. Bayesian Anal. **4**(1), 119–150 (2009)

Ben-Gal, I.: Bayesian Networks, (2008)

Blomstedt, P., Mesquita, D., Lintusaari, J., Sivula, T., Corander, J., Kaski, S.: Meta-analysis of Bayesian analyses, (2019)

Brandstetter, J., van den Berg, R., Welling, M., Gupta, J. K.: Clifford neural layers for PDE modeling. In: The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023. OpenReview.net, (2023)

Breiman, L.: Bagging predictors. Mach. Learn. **24**, 123–140 (2004)

Bühlmann, P.: Discussion of Big Bayes Stories and BayesBag. Stat. Sci. **29**(1), 91–94 (2014)

Bürkner, P.-C., Kröker, I., Oladyshkin, S., Nowak, W.: A fully Bayesian sparse polynomial chaos expansion approach with joint priors on the coefficients and global selection of terms. J. Comput. Phys. **488**, 112210 (2023)

Bürkner, P.-C., Scholz, M., Radev, S.T.: Some models are useful, but how do we know which ones? Towards a unified Bayesian model taxonomy. Statistics Surveys **17**, 216–310 (2023)

Bärkner, P.-C.: brms: An R package for Bayesian multilevel models using Stan. J. Statistical Software **80**(1), 1–28 (2017)

Bärkner, P.-C.: Advanced Bayesian Multilevel Modeling with the R Package brms. R J. **10**(1), 395–411 (2018)

Bürkner, P.-C., Gabry, J., Weber, S., Johnson, A., Modrak, M., Badr, H. S., Weber, F., Ben-Shachar, M. S., and Rabel, H.: Running brms models with within-chain parallelization. (2023). URL https://CRAN.R-Project.org/package=brms. Vignette included in R package brms, version 2.17.0

Carpenter, B., Gelman, A., Hoffman, M.D., Lee, D., Goodrich, B., Betancourt, M., Brubaker, M., Guo, J., Li, P., Riddell, A.: Stan: A probabilistic programming language. J. Stat. Softw. **76**(1), 1–32 (2017)

Cleary, E., Garbuno-Inigo, A., Lan, S., Schneider, T., Stuart, A.M.: Calibrate, emulate, sample. J. Comput. Phys. **424**, 109716 (2021)

Cook, S.R., Gelman, A., Rubin, D.B.: Validation of software for Bayesian models using posterior quantiles. J. Comput. Graph. Stat. **15**(3), 675–692 (2006)

Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the em algorithm. J. Royal Statist. Soc. Series B (Methodological) **39**(1), 1–38 (1977)

Dormand, J., Prince, P.: A family of embedded runge-kutta formulae. J. Comput. Appl. Math. **6**(1), 19–26 (1980)

Douady, C.J., Delsuc, F., Boucher, Y., Doolittle, W.F., Douzery, E.J.P.: Comparison of Bayesian and maximum likelihood bootstrap measures of phylogenetic reliability. Mol. Biol. Evol. **20**(2), 248–254 (2003)

Efron, B.: Bootstrap methods: another look at the jackknife. Ann. Stat. **7**(1), 1–26 (1979)

Fiedler, F., Lucia, S.: Improved uncertainty quantification for neural networks with Bayesian last layer. IEEE Access **11**, 123149–123160 (2023)

Gelman, A., Carlin, J., Stern, H., Dunson, D., Vehtari, A., and Rubin, D.: Bayesian Data Analysis, Third Edition. Chapman & Hall/CRC Texts in Statistical Science. Taylor & Francis, (2013)

Geyer, C. J.: Markov chain monte carlo maximum likelihood, (1991)

Giordano, G., Blanchini, F., Bruno, R., Colaneri, P., Filippo, A., Di Matteo, A., Colaneri, M.: Modelling the covid-19 epidemic and implementation of population-wide interventions in italy. Nat. Med. **26**, 1–6 (2020)

Gneiting, T., Balabdaoui, F., Raftery, A.E.: Probabilistic Forecasts, Calibration and Sharpness. J. R. Stat. Soc. Ser. B Stat Methodol. **69**(2), 243–268 (2007)

Goodfellow, I., Bengio, Y., Courville, A.: Deep learning. MIT Press, Cambridge (2016)

Gorinova, M. I., Moore, D., Hoffman, M. D.: Automatic reparameterisation of probabilistic programs, (2019)

Gramacy, R. B.: Surrogates: Gaussian Process Modeling, Design and Optimization for the Applied Sciences. Chapman Hall/CRC, Boca Raton, Florida. (2020). http://bobby.gramacy.com/surrogates/

Graves, A.: Practical variational inference for neural networks. In: Shawe-Taylor, J., Zemel, R., Bartlett, P., Pereira, F., Weinberger, K. (eds.) Advances in Neural Information Processing Systems, vol. 24. Curran Associates Inc (2011)

Gruber, C., Schenk, P. O., Schierholz, M., Kreuter, F., Kauermann, G.: Sources of uncertainty in machine learning – a statisticians' view, (2023)

Halton, J.H.: On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals. Numer. Math. **2**(1), 84–90 (1960)

Harrison, J., Willes, J., Snoek, J.: Variational Bayesian last layers. *CoRR*, abs/2404.11599, (2024)

Hethcote, H.W.: The mathematics of infectious diseases. SIAM Rev. **42**(4), 599–653 (2000)

Hinton, G. E., van Camp, D.: Keeping the neural networks simple by minimizing the description length of the weights. In: Proceedings of the Sixth Annual Conference on Computational Learning Theory, COLT '93, page 5-13, New York, NY, USA. Association for Computing Machinery, (1993)

Hoffman, M.D., Gelman, A.: The no-u-turn sampler: Adaptively setting path lengths in hamiltonian monte carlo. J. Mach. Learn. Res. **15**(47), 1593–1623 (2014)

Huggins, J. H., Miller, J. W.: Robust inference and model criticism using bagged posteriors, (2020)

Hüllermeier, E., Waegeman, W.: Aleatoric and epistemic uncertainty in machine learning: an introduction to concepts and methods. Mach. Learn. **110**(3), 457–506 (2021)

Izmailov, P., Vikram, S., Hoffman, M. D., Wilson, A. G.: What are bayesian neural network posteriors really like?. In: Meila, M. and Zhang, T., editors, Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event, volume 139 of Proceedings of Machine Learning Research, pp. 4629–4640. PMLR, (2021)

Jacob, P. E., Murray, L. M., Holmes, C. C., Robert, C. P.: Better together? statistical learning in models made of modules, (2017)

Jordan, M.I. (ed.): Learning in graphical models. MIT Press, Cambridge (1999)

Kallioinen, N., Paananen, T., Bürkner, P.-C., Vehtari, A.: Detecting and diagnosing prior and likelihood sensitivity with power-scaling, (2022)

Kennedy, M.C., O'Hagan, A.: Bayesian calibration of computer models. J. Royal Statist. Soc. Series B (Statistical Methodology) **63**(3), 425–464 (2001)

Kim, S., Moon, H., Modrák, M., Säilynoja, T.: Simulation Based Calibration for rstan/cmdstanr models. (2023). https://hyunjimoon.github.io/SBC/, https://github.com/hyunjimoon/SBC/

Kohlhaas, R., Kröker, I., Oladyshkin, S., Nowak, W.: Gaussian active learning on multi-resolution arbitrary polynomial chaos emulator:

concept for bias correction, assessment of surrogate reliability and its application to the carbon dioxide benchmark. Comput. Geosci. **27**, 1–21 (2023)
Kristiadi, A., Hein, M., Hennig, P.: Being bayesian, even just a bit, fixes overconfidence in relu networks. In: Proceedings of the 37th International Conference on Machine Learning, ICML'20. JMLR.org, (2020)
Kucukelbir, A., Tran, D., Ranganath, R., Gelman, A., Blei, D.M.: Automatic differentiation variational inference. J. Mach. Learn. Res. **18**, 14:1-14:45 (2017)
Kuehnert, J., McGlynn, D., Remy, S. L., Walcott-Bryant, A., Jones, A.: Surrogate Ensemble Forecasting for Dynamic Climate Impact Models. (2022). arXiv:2204.05795 [physics]
Laloy, E., Rogiers, B., Vrugt, J., Mallants, D., Jacques, D.: Efficient posterior exploration of a high-dimensional groundwater model from two-stage mcmc simulation and polynomial chaos expansion. Water Resources Research, 49 (2013)
Lavin, A., Zenil, H., Paige, B., Krakauer, D., Gottschlich, J., Mattson, T., Anandkumar, A., Choudry, S., Rocki, K., Baydin, A. G., Prunkl, C., Isayev, O., Peterson, E., McMahon, P. L., Macke, J. H., Cranmer, K., Zhang, J., Wainwright, H. M., Hanuka, A., Veloso, M., Assefa, S., Zheng, S., Pfeffer, A.: Simulation intelligence: Towards a new generation of scientific methods. CoRR, abs/2112.03235, (2021)
Li, J., Marzouk, Y.M.: Adaptive construction of surrogates for the Bayesian solution of inverse problems. SIAM J. Sci. Comput. **36**(3), A1163–A1186 (2014)
Li, Z., Kovachki, N. B., Azizzadenesheli, K., Liu, B., Bhattacharya, K., Stuart, A. M., Anandkumar, A.: Fourier neural operator for parametric partial differential equations. In: 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021. OpenReview.net, (2021)
Little, R.J., Rubin, D.B.: Statistical analysis with missing data, vol. 793. John Wiley & Sons (2019)
Lloyd, S.: Least squares quantization in pcm. IEEE Trans. Inf. Theory **28**(2), 129–137 (1982)
MacQueen, J. B.: Some methods for classification and analysis of multivariate observations. In: Cam, L. M. L. and Neyman, J., (eds), Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability (1967)
Martino, S., Riebler, A.: Integrated Nested Laplace Approximations (INLA). arXiv e-prints, page arXiv:1907.01248 (2019)
Marzouk, Y., Xiu, D.: A stochastic collocation approach to Bayesian inference in inverse problems, p. 6. NNSA Center for Prediction of Reliability, Integrity and Survivability of Microsystems, PRISM (2009)
Marzouk, Y.M., Najm, H.N., Rahn, L.A.: Stochastic spectral methods for efficient Bayesian solution of inverse problems. J. Comput. Phys. **224**(2), 560–586 (2007)
McElreath, R.: Statistical Rethinking: A Bayesian Course with Examples in R and STAN. Chapman & Hall/CRC Texts in Statistical Science. CRC Press, (2020)
Medina-Aguayo, F.J., Christen, J.A.: Penalised t-walk mcmc. J. Statistical Plann. Inference **221**, 230–247 (2022)
Meyer, L., Pottier, L., Ribés, A., Raffin, B.: Deep surrogate for direct time fluid dynamics. CoRR, abs/2112.10296, (2021)
Mitra, E.D., Hlavacek, W.S.: Parameter estimation and uncertainty quantification for systems biology models. Curr. Opinion Syst. Biol. **18**, 9–18 (2019)
Modrák, M., Moon, A. H., Kim, S., Bärkner, P., Huurre, N., Faltejsková, K., Gelman, A., Vehtari, A.: Simulation-based calibration checking for bayesian computation: The choice of test quantities shapes sensitivity, (2022)
Mohammadi, F., Kopmann, R., Guthke, A., Oladyshkin, S., Nowak, W.: Bayesian selection of hydro-morphodynamic models under computational time constraints. Adv. Water Resour. **117**, 53–64 (2018)
Murphy, K. P.: Conjugate Bayesian analysis of the gaussian distribution, (2007)
Murphy, K.P.: Machine learning: a probabilistic perspective. The MIT Press, Cambridge (2012)
Neal, R.: Handbook of markov chain monte carlo. Chapman and Hall/CRC, Boca Raton (2011)
Oladyshkin, S., Nowak, W.: Data-driven uncertainty quantification using the arbitrary polynomial chaos expansion. Reliab. Eng. Syst. Safety **106**, 179–190 (2012)
Petersen, K. B., Pedersen, M.S.: The matrix cookbook. Version 20121115, (2012)
Piironen, J., Paasiniemi, M., Vehtari, A.: Projective inference in high-dimensional problems: Prediction and feature selection. Electronic J. Statist. **14**(1), 2155–2197 (2020)
Plummer, M.: Cuts in Bayesian graphical models. Stat. Comput. **25**, 37–43 (2014)
Psaros, A.F., Meng, X., Zou, Z., Guo, L., Karniadakis, G.E.: Uncertainty quantification in scientific machine learning: Methods, metrics, and comparisons. J. Comput. Phys. **477**, 111902 (2023)
Radev, S. T., Schmitt, M., Pratz, V., Picchini, U., Köthe, U., Bürkner, P.: Jana: Jointly amortized neural approximation of complex Bayesian models. In Evans, R. J. and Shpitser, I., editors, Uncertainty in Artificial Intelligence, UAI 2023, July 31 - 4 August 2023, Pittsburgh, PA, USA, volume 216 of Proceedings of Machine Learning Research, pages 1695–1706. PMLR, (2023)
Raissi, M., Perdikaris, P., Karniadakis, G.: Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. J. Comput. Phys. **378**, 686–707 (2019)
Ranftl, S., von der Linden, W.: Bayesian surrogate analysis and uncertainty propagation. Phys. Sci. Forum **3**, 6 (2021)
Rasmussen, C.E., Williams, C.K.I.: Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning). The MIT Press (2005)
Renardy, M., Yi, T.-M., Xiu, D., Chou, C.-S.: Parameter uncertainty quantification using surrogate models applied to a spatial model of yeast mating polarization. PLoS Comput. Biol. **14**(5), 1–26 (2018)
Robert, C., Casella, G.: Monte carlo statistical methods. Springer texts in statistics. Springer, New York (2005)
Ruiz-Cárdenas, R., Krainski, E.T., Rue, H.: Direct fitting of dynamic models using integrated nested laplace approximations - INLA. Comput. Stat. Data Anal. **56**(6), 1808–1828 (2012)
Shao, Q., Younes, A., Fahs, M., Mara, T.A.: Bayesian sparse polynomial chaos expansion for global sensitivity analysis. Comput. Methods Appl. Mech. Eng. **318**, 474–496 (2017)
Smith, R.C.: Uncertainty Quantification: Theory, Implementation, and Applications. Society for Industrial and Applied Mathematics, USA (2013)
Sobol', I.: On the distribution of points in a cube and the approximate evaluation of integrals. USSR Comput. Math. Math. Phys. **7**(4), 86–112 (1967)
Stan Development Team: Stan Modeling Language Users Guide and Reference Manual. Version **2**, 36 (2024)
Sudret, B.: Global sensitivity analysis using polynomial chaos expansions. Reliability Eng. Syst. Safety **93**(7), 964–979 (2008)
Säilynoja, T., Bärkner, P.-C., Vehtari, A.: Graphical test for discrete uniformity and its applications in goodness-of-fit evaluation and multiple sample comparison. Statistics and Computing, 32, (2022)
Talts, S., Betancourt, M., Simpson, D., Vehtari, A., Gelman, A.: Validating Bayesian Inference Algorithms with Simulation-Based Calibration. arXiv e-prints, page arXiv:1804.06788 (2018)
Tarakanov, A., Elsheikh, A.H.: Regression-based sparse polynomial chaos for uncertainty quantification of subsurface flow models. J. Comput. Phys. **399**, 108909 (2019)
van der Vaart, A.: Asymptotic Statistics. Cambridge University Press, Cambridge (2000)

Vehtari, A., Gelman, A., Gabry, J.: Practical bayesian model evaluation using leave-one-out cross-validation and waic. Statistics and Computing, 27, (2017)
Vehtari, A., Gelman, A., Simpson, D., Carpenter, B., Bärkner, P.-C.: Rank-normalization, folding, and localization: An improved $\widehat{R}$ for assessing convergence of MCMC. Bayesian Analysis, 16(2), (2021). arXiv:1903.08008
Vehtari, A., Ojanen, J.: A survey of Bayesian predictive methods for model assessment, selection and comparison. Statistics Surveys **6**, 142–228 (2012)
Waddell, P., Kishino, H., Ota, R.: Very fast algorithms for evaluating the stability of ml and Bayesian phylogenetic trees from sequence data. Genome informatics. International Conference on Genome Informatics **13**, 82–92 (2002)
Wiener, N.: The Homogeneous Chaos. Am. J. Math. **60**(4), 897–936 (1938)
Zeng, L., Shi, L., Zhang, D., Wu, L.: A sparse grid Bayesian method for contaminant source identification. Adv. Water Resour. **37**, 1–9 (2012)
Zhang, J., Zheng, Q., Chen, D., Wu, L., Zeng, L.: Surrogate-based Bayesian inverse modeling of the hydrological system: An adaptive approach considering surrogate approximation error. Water Resources Research, 56(1):e2019WR025721. e2019WR025721 2019WR025721, (2020)
Zhu, Y., Zabaras, N.: Bayesian deep convolutional encoder-decoder networks for surrogate modeling and uncertainty quantification. J. Comput. Phys. **366**, 415–447 (2018)