Structural Safety 114 (2025) 102579

Contents lists available at ScienceDirect

Structural Safety

journal homepage: www.elsevier.com/locate/strusafe

Response probability distribution estimation of expensive computer simulators: A Bayesian active learning perspective using Gaussian process regression

Chao Dang [a],*, Marcos A. Valdebenito [a], Nataly A. Manque [a], Jun Xu [b,c], Matthias G.R. Faes [a]

[a] Chair for Reliability Engineering, TU Dortmund University, Leonhard-Euler-Str. 5, Dortmund 44227, Germany
[b] College of Civil Engineering, Hunan University, Changsha 410082, PR China
[c] State Key Laboratory of Bridge Engineering Safety and Resilience, Hunan University, Changsha 410082, PR China

ARTICLE INFO

Keywords:
Probability distribution estimation
Computer simulator
Bayesian inference
Bayesian active learning
Gaussian process regression

ABSTRACT

Estimation of the response probability distributions of computer simulators subject to input random variables is a crucial task in many fields. However, achieving this task with guaranteed accuracy remains an open computational challenge, especially for expensive-to-evaluate computer simulators. In this work, a Bayesian active learning perspective is presented to address the challenge, which is based on the use of the Gaussian process (GP) regression. First, estimation of the response probability distributions is conceptually interpreted as a Bayesian inference problem, as opposed to frequentist inference. This interpretation provides several important benefits: (1) it quantifies and propagates discretization error probabilistically; (2) it incorporates prior knowledge of the computer simulator, and (3) it enables the effective reduction of numerical uncertainty in the solution to a prescribed level. The conceptual Bayesian idea is then realized by using the GP regression, where we derive the posterior statistics of the response probability distributions in semi-analytical form and also provide a numerical solution scheme. Based on the practical Bayesian approach, a Bayesian active learning (BAL) method is further proposed for estimating the response probability distributions. In this context, the key contribution lies in the development of two crucial components for active learning, i.e., stopping criterion and learning function, by taking advantage of the posterior statistics. It is empirically demonstrated by five numerical examples that the proposed BAL method can efficiently estimate the response probability distributions with desired accuracy.

1. Introduction

Computer simulators are widely used across various fields of science and engineering to model, analyze, and predict the behavior of complex systems in the presence of randomness. For example, in physics, simulating quantum systems aids in understanding particle behavior, interactions, and probabilistic outcomes. In applied mechanics and engineering, finite element models are employed extensively for studying the performance of structures, considering randomness in their internal structural properties and external operating conditions. In the latter context, typical research topics include: (1) Reliability analysis, which assesses the probability that a system produces an undesired response; (2) Statistical moment evaluation, focused on determining the statistical moments of the system's response; and (3) Probability distribution estimation, which involves estimating the probability distribution of the system response. Among these, a central problem is the estimation of the response probability distributions, such as the cumulative distribution function (CDF), complementary CDF (CCDF), and probability density function (PDF). This task is crucial because it provides a complete characterization of the uncertain system response, allowing a more comprehensive understanding of the underlying system behavior under random influences without the need for costly and time-consuming physical experiments. However, the long runtime of a single evaluation of computer simulations often makes this process particularly challenging.

Over the past few decades, a variety of methods have been developed for estimating the response probability distributions of computer simulators. Commonly used methods can be broadly classified into three types: simulation-based methods, statistical moment-based methods and surrogate-assisted methods. Often considered as the most

* Corresponding author.
E-mail address: chao.dang@tu-dortmund.de (C. Dang).

https://doi.org/10.1016/j.strusafe.2025.102579
Received 13 August 2024; Received in revised form 29 November 2024; Accepted 10 February 2025
Available online 19 February 2025

straightforward approach, simulation-based methods estimate the response probability distribution by first generating numerous samples of the response of interest. Examples of such methods include Monte Carlo simulation (MCS) [1] and its various variants such as stratified sampling [2,3], Latin hypercube sampling [2,4] and quasi-MCS [5].Using the response samples, regular density estimation approaches (such as histogram and kernel denstiy estimation) can then be performed to approximate the underlying probability distribution. In general, simulation-based methods are less or not sensitive to the input dimensionality and non-linearity of the computer simulator under consideration. Nonetheless, they often suffer from slow convergence rates, necessitating a significant number of simulations. This drawback becomes particularly pronounced when dealing with an expensive-to-evaluate computer simulator. As an alternative, statistical moment-based methods approximate the probability distribution of the response of interest from knowledge of its estimated statistical moments through a prescribed distribution model. Note that most, but not all, of the methods in this category were developed in the context of reliability analysis, but are equally applicable to the topic of this work. A non-exhaustive list includes high-order moments-based methods [6–10], fractional moments-based maximum entropy methods [11,12] and fractional moments-based mixture distribution methods [13,14]. In addition, recent advances have explored approaches based on harmonic moments [15,16] and characteristic function [17–19]. Nonetheless, a common criticism of such methods is that the numerical errors behind those probability distribution estimates are rarely known and remain challenging to derive. Last but not least, surrogate-assisted methods have also been developed for response probability distribution estimation. The key idea is to use a simplified model to substitute the original expensive computer simulator based on a set of carefully selected input–output data points. Representative methods include polynomial chaos expansions [20–22], Gaussian process (GP) regression or Kriging [23–26]. The interested reader is referred to [27] for a comprehensive study of active learning based surrogates for estimating response probability distributions. It is shown that surrogate-assisted methods have the potential to reduce the computational burden if well designed. In addition to the three types of methods mentioned above, it is worth noting that many other probability distribution estimation methods have been developed in specific field of stochastic dynamics, for example, path integrals [28], globally-evolving-based generalized density evolution equation [29], probability density evolution method [30,31], direct probability integral method [32,33], to just name a few. It is noteworthy that the latter two methods can also be applied to stochastic static systems. In summary, despite considerable efforts, it remains an open challenge to efficiently and accurately estimate the response probability distributions of expensive computer simulators.

To address the research gap, this work aims to present a Bayesian active learning perspective on the response probability distribution estimation of expensive computer simulators using GP regression. The main contributions can be summarized as follows:

- The estimation of response probability distributions is conceptually interpreted as a Bayesian inference problem. This interpretation brings several important benefits: (1) it provides a principled approach to quantifying and propagating discretization error as a source of epistemic uncertainty in a probabilistic way through a computational pipeline; (2) it allows the incorporation of our prior knowledge about the computer simulator into the estimation; and (3) it enables the effective reduction of the numerical uncertainty in the solution of the response probability distribution to a prescribed level.
- The conceptual Bayesian idea is then realized by the virtue of the GP regression, an easy-to-use Bayesian model to define a distribution over functions. A GP prior is first assigned to the computer simulator under consideration, and then conditioning the GP prior on several computer simulator evaluations yields a posterior GP over the computer simulator. We derive the posterior statistics of the response CDF, CCDF and PDF in semi-analytical form and also provide the MCS solutions. The developed Bayesian approach can be seen as an extension of the Bayesian approach for failure probability estimation reported in [34–36] and belongs to a probabilistic numerical method [37].
- The problem of response probability distribution estimation is finally framed in a Bayesian active learning setting. Specifically, a Bayesian active learning method is proposed for estimating the response probability distributions, based on the above practical Bayesian approach. In this context, the key contribution is the development of two crucial components for active learning by making use of the posterior statistics: a stopping criterion and a learning function.

The rest of the paper is structured as follows. Section 2 presents the formulation of the problem to be solved in this work. The conceptual Bayesian framework for response probability distribution estimation is given in Section 3, followed by the practical one. In Section 4, we introduce the proposed Bayesian active learning method for response probability distribution estimation. Five numerical examples are studied in Section 5 to illustrate the proposed method. Section 6 concludes the paper with some final remarks.

2. Problem formulation

Consider a physical model represented by a real-valued deterministic computer simulator $g : \mathbb{R}^d \mapsto \mathbb{R}$. The input to the model is a vector of d continuous random variables, i.e., $\boldsymbol{X} = [X_1, X_2, \dots, X_d] \in \mathcal{X} \subseteq \mathbb{R}^d$, with the prescribed joint PDF $f_{\boldsymbol{X}}(\boldsymbol{x})$. As a consequence, the response of the model is also a random variable denoted by $Y \in \mathcal{Y} \subseteq \mathbb{R}$, i.e., $Y = g(\boldsymbol{X})$. The CDF of Y is defined by:

$$F_Y(y) = \mathbb{P}(Y \le y) = \mathbb{P}(g(\boldsymbol{X}) \le y) = \int_{\mathcal{X}} I_A(\boldsymbol{x}) f_{\boldsymbol{X}}(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}, \tag{1}$$

where $\mathbb{P}$ is the probability measure; $A = \{\boldsymbol{x} \in \mathcal{X} | g(\boldsymbol{x}) \le y\}$; $I_A(\boldsymbol{x})$ is the indicator function: $I_A(\boldsymbol{x}) = 1$ if $\boldsymbol{x} \in A$, and $I_A(\boldsymbol{x}) = 0$ otherwise. The CCDF of Y is given by:

$$\overline{F}_Y(y) = \mathbb{P}(Y > y) = \mathbb{P}(g(\boldsymbol{X}) > y) = \int_{\mathcal{X}} I_{\overline{A}}(\boldsymbol{x}) f_{\boldsymbol{X}}(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}, \tag{2}$$

where $\overline{A}$ is the complement set of A, i.e., $\overline{A} = \{\boldsymbol{x} \in \mathcal{X} | g(\boldsymbol{x}) > y\}$. Note that $\overline{F}_Y(y) = 1 - F_Y(y)$ holds. Assuming that $F_Y(y)$ is almost everywhere differentiable, Y admits a PDF, which is expressed as:

$$f_Y(y) = \frac{\mathrm{d}F_Y(y)}{\mathrm{d}y} = \int_{\mathcal{X}} \delta(y - g(\boldsymbol{x})) f_{\boldsymbol{X}}(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} = \int_{\mathcal{X}} \zeta_B(\boldsymbol{x}) f_{\boldsymbol{X}}(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}, \tag{3}$$

where $\delta(\cdot)$ is the Dirac delta function; $B = \{\boldsymbol{x} \in \mathcal{X} | y = g(\boldsymbol{x})\}$; $\zeta_B(\boldsymbol{x}) = \delta(y - g(\boldsymbol{x}))$.

The analytical solutions of Eqs. (1)–(3) are not available, except for some rather simple cases. Therefore, in practice, one has to resort to numerical methods. A numerical solution scheme often requires the computer simulator to be evaluated many times. This can be computationally prohibitive if the computer simulator is expensive to evaluate, as is often the case.

3. Bayesian inference about response probability distributions

This section treats the estimation of response probability distributions as a Bayesian inference problem, as opposed to frequentist inference. First, a conceptual Bayesian framework is created, which provides the philosophical foundations of this work. Second, a practical Bayesian framework is developed based on the GP regression.

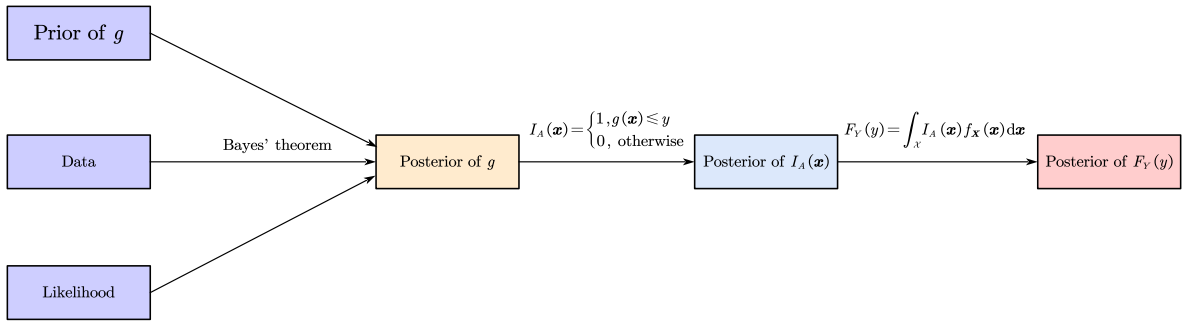


Fig. 1. Conceptual illustration of the Bayesian inference about the response CDF.

3.1. Conceptual Bayesian framework

As in many existing probabilistic numerical methods (e.g., Bayesian optimization [38,39] and Bayesian quadrature [40–42]), the premise of a Bayesian inference treatment of the response probability distribution estimation is that the g function should be thought of as being random. This is because, even though g is, by our definition, a deterministic mapping, it remains numerically unknown until we actually evaluate it. Furthermore, even when it can be evaluated, it is computationally impractical to compute it at every possible location. Once we admit that we do have epistemic uncertainty about g due to discretization, it becomes natural to use a Bayesian approach to the problem of response probability distribution estimation. Take the estimation of the response CDF (Eq. (1)) as an example. A Bayesian approach starts by placing a prior distribution on the g function. Conditioning this prior on several evaluations of g provides a posterior distribution over g according to the Bayes' law. This posterior distribution subsequently induces a posterior distribution over the indicator function I_A, and ultimately, a posterior distribution over the sought response CDF F_Y. The conceptual idea is illustrated in Fig. 1. It should be noted that the response CCDF and PDF can be treated in a similar way.

The conceptual Bayesian framework offers several significant advantages, among which the following are particularly noteworthy:

- It provides a principled approach to quantifying and propagating the discretization error in a fully probabilistic manner through a computational pipeline. The Bayesian framework treats the discretization errorarising from observing g at several discrete locations as a source of epistemic uncertainty, enabling probabilistic analysis and reasoning. This allows for the systematic quantification and propagation of the discretization error through a computational pipeline. Note that the posterior distribution over the response probability distribution reflects the fact that g has been discretized.
- It allows the integration of our prior beliefs of the g function into the estimation process. Specifically, prior knowledge regarding properties of g, such as smoothness, periodicity, or other structural characteristics, can be encoded through its prior distribution. By doing so, the resulting method not only aligns with existing domain knowledge but also often achieves faster convergence and reduces computational cost. This capability stems from the Bayesian nature of the framework, offering a flexibility that is not possible in frequentist inference.
- It facilitates the effective reduction of numerical uncertainty in the solution of the response probability distribution to a specified level. The well-quantified uncertainty can provide useful information for designing computer experiments in order to effectively reduce the numerical uncertainty to an accepted level. An illustrative exploration of this feature is provided in Section 4, highlighting its significance and potential benefits.

3.2. Practical Bayesian framework

3.2.1. Prior distribution

A stochastic process can be assigned to g as a prior distribution to express our uncertainty associated with it. Among many possible options, this study adopts the widely-used GP prior such that:

$$g_0(\boldsymbol{x}) \sim \mathcal{GP}(m_{g_0}(\boldsymbol{x}), k_{g_0}(\boldsymbol{x}, \boldsymbol{x}')), \tag{4}$$

where g_0 denotes the prior distribution of g before seeing any observations; $m_{g_0}(\boldsymbol{x}) : \mathcal{X} \mapsto \mathbb{R}$ is prior mean function and $k_{g_0}(\boldsymbol{x}, \boldsymbol{x}') : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$ is the prior covariance (also known as the kernel) function. The GP prior is fully specified by $m_{g_0}(\boldsymbol{x})$ and $k_{g_0}(\boldsymbol{x}, \boldsymbol{x}')$. The prior mean function reflects the general trend of g, while the prior covariance function encodes key assumptions about the smoothness, periodicity, and other properties of g. Without loss of generality, we employ a constant prior mean and a Gaussian kernel (though other choices can be employed if specific prior knowledge is available):

$$m_{g_0}(\boldsymbol{x}) = \beta, \tag{5}$$

$$k_{g_0}(\boldsymbol{x}, \boldsymbol{x}') = \sigma_0^2 \exp\left(-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{x}')\boldsymbol{\Sigma}^{-1}(\boldsymbol{x} - \boldsymbol{x}')^\top\right), \tag{6}$$

where $\beta \in \mathbb{R}$; $\sigma_0 > 0$ is the standard deviation of g_0; $\boldsymbol{\Sigma} = \mathrm{diag}\left\{l_1^2, l_2^2, \ldots, l_d^2\right\}$ is a diagonal matrix with $l_i > 0$ being the characteristic length-scale in the ith dimension. The parameters collected in $\boldsymbol{\vartheta} = \left\{\beta, \sigma_0, l_1, l_2, \ldots, l_d\right\}$ are known as hyper-parameters.

3.2.2. Estimating hyper-parameters

Assume that we now have a dataset $\boldsymbol{\mathcal{D}} = \{\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{Y}}\}$, where $\boldsymbol{\mathcal{X}} = \left\{\boldsymbol{x}^{(i)}\right\}_{i=1}^{n}$ is an n-by-d matrix with its ith row being $\boldsymbol{x}^{(i)}$, and $\boldsymbol{\mathcal{Y}} = \left\{y^{(i)}\right\}_{i=1}^{n_0}$ is an n-by-1 vector with $y^{(i)} = g(\boldsymbol{x}^{(i)})$. The hyper-parameters can be learned from data $\boldsymbol{\mathcal{D}}$ by maximizing the log marginal likelihood:

$$\log p(\boldsymbol{\mathcal{Y}}|\boldsymbol{\mathcal{X}}, \boldsymbol{\vartheta}) = -\frac{1}{2}\left[(\boldsymbol{\mathcal{Y}} - \beta)^\top \boldsymbol{K}_{g_0}^{-1}(\boldsymbol{\mathcal{Y}} - \beta) + \log|\boldsymbol{K}_{g_0}| + n\log(2\pi)\right], \tag{7}$$

where $\boldsymbol{K}_{g_0}$ is an n-by-n covariance matrix with its (i, j)-th entry is $\left[\boldsymbol{K}_{g_0}\right]_{i,j} = k_{g_0}(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)})$. In addition to maximum likelihood estimation, methods such as cross-validation, maximum a posteriori estimation, and the fully Bayesian approach can also be used to select the hyperparameters. Among these, the fully Bayesian approach is conceptually appealing; however, it may present challenges in obtaining the analytical posterior distribution of the GP regression.

3.2.3. Posterior statistics

Conditioning the GP prior on the data $\boldsymbol{\mathcal{D}}$ gives the posterior distribution of g, which also follows a GP:

$$g_n(\boldsymbol{x}) \sim \mathcal{GP}(m_{g_n}(\boldsymbol{x}), k_{g_n}(\boldsymbol{x}, \boldsymbol{x}')), \tag{8}$$

where g_n represents the posterior distribution of g after seeing n observations; $m_{g_n}(\boldsymbol{x}) : \mathcal{X} \mapsto \mathbb{R}$ is the posterior mean function, and $k_{g_n}(\boldsymbol{x}, \boldsymbol{x}') : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$ is the posterior covariance function:

$$m_{g_n}(\boldsymbol{x}) = m_{g_0}(\boldsymbol{x}) + \boldsymbol{k}_{g_0}(\boldsymbol{x}, \boldsymbol{\mathcal{X}})^\top \boldsymbol{K}_{g_0}^{-1} \left(\boldsymbol{\mathcal{Y}} - \boldsymbol{m}_{g_0}(\boldsymbol{\mathcal{X}}) \right), \tag{9}$$

$$k_{g_n}(\boldsymbol{x}, \boldsymbol{x}') = k_{g_0}(\boldsymbol{x}, \boldsymbol{x}') - \boldsymbol{k}_{g_0}(\boldsymbol{x}, \boldsymbol{\mathcal{X}})^\top \boldsymbol{K}_{g_0}^{-1} \boldsymbol{k}_{g_0}(\boldsymbol{\mathcal{X}}, \boldsymbol{x}'), \tag{10}$$

where $\boldsymbol{m}_{g_0}(\boldsymbol{\mathcal{X}})$ is an n-by-1 mean vector with its ith element being $m(\boldsymbol{x}^{(i)})$; $\boldsymbol{k}_{g_0}(\boldsymbol{x}, \boldsymbol{\mathcal{X}})$ is an n-by-1 covariance vector with its ith element being $k_{g_0}(\boldsymbol{x}, \boldsymbol{x}^{(i)})$; $\boldsymbol{k}_{g_0}(\boldsymbol{\mathcal{X}}, \boldsymbol{x}')$ is an n-by-1 vector with its ith element being $k_{g_0}(\boldsymbol{x}^{(i)}, \boldsymbol{x}')$. For further detailed information on the above standard GP regression, please refer to [43].

The induced posterior distribution of I_A conditional on $\boldsymbol{\mathcal{D}}$ follows a generalized Bernoulli process (GBP):

$$I_{A,n}(\boldsymbol{x}) \sim \mathcal{GBP}(m_{I_{A,n}}(\boldsymbol{x}), k_{I_{A,n}}(\boldsymbol{x}, \boldsymbol{x}')), \tag{11}$$

where $I_{A,n}$ denotes the posterior distribution of I_A after seeing n observations; $m_{I_{A,n}}(\boldsymbol{x}) : \mathcal{X} \mapsto \mathbb{R}$ and $k_{I_{A,n}}(\boldsymbol{x}, \boldsymbol{x}') : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$ are the posterior mean and covariance functions respectively, which can be derived as follows:

$$\begin{aligned} m_{I_{A,n}}(\boldsymbol{x}) =& \mathbb{E}\left[I_{A,n}(\boldsymbol{x})\right] \\ =& 1 \cdot \mathbb{P}\left(I_{A,n}(\boldsymbol{x}) = 1\right) + 0 \cdot \mathbb{P}\left(I_{A,n}(\boldsymbol{x}) = 0\right) \\ =& \mathbb{P}\left(g_n(\boldsymbol{x}) \le y\right) \\ =& \boldsymbol{\Phi}\left(\frac{y - m_{g_n}(\boldsymbol{x})}{\sigma_{g_n}(\boldsymbol{x})}\right), \end{aligned} \tag{12}$$

$$\begin{aligned} k_{I_{A,n}}(\boldsymbol{x}, \boldsymbol{x}') =& \mathbb{E}\left[\left(I_{A,n}(\boldsymbol{x}) - m_{I_{A,n}}(\boldsymbol{x})\right)\left(I_{A,n}(\boldsymbol{x}') - m_{I_{A,n}}(\boldsymbol{x}')\right)\right] \\ =& \mathbb{E}\left[I_{A,n}(\boldsymbol{x}) I_{A,n}(\boldsymbol{x}')\right] - \mathbb{E}\left[I_{A,n}(\boldsymbol{x})\right]\mathbb{E}\left[I_{A,n}(\boldsymbol{x}')\right] \\ =& \mathbb{P}\left(g_n(\boldsymbol{x}) \le y, g_n(\boldsymbol{x}') \le y\right) - m_{I_{A,n}}(\boldsymbol{x}) m_{I_{A,n}}(\boldsymbol{x}') \\ =& \boldsymbol{\Phi}_2\left(\begin{bmatrix} y \\ y \end{bmatrix}; \begin{bmatrix} m_{g_n}(\boldsymbol{x}) \\ m_{g_n}(\boldsymbol{x}') \end{bmatrix}, \begin{bmatrix} \sigma_{g_n}^2(\boldsymbol{x}) & k_{g_n}(\boldsymbol{x}, \boldsymbol{x}') \\ k_{g_n}(\boldsymbol{x}', \boldsymbol{x}) & \sigma_{g_n}^2(\boldsymbol{x}') \end{bmatrix}\right) \\ & - \boldsymbol{\Phi}\left(\frac{y - m_{g_n}(\boldsymbol{x})}{\sigma_{g_n}(\boldsymbol{x})}\right)\boldsymbol{\Phi}\left(\frac{y - m_{g_n}(\boldsymbol{x}')}{\sigma_{g_n}(\boldsymbol{x}')}\right), \end{aligned} \tag{13}$$

where $\boldsymbol{\Phi}$ is the CDF of a standard normal variable; $\boldsymbol{\Phi}_2$ is the bivariate normal CDF; $\sigma_{g_n}(\boldsymbol{x}) : \mathcal{X} \mapsto \mathbb{R}$ is the posterior standard deviation function of g, i.e., $\sigma_{g_n}(\boldsymbol{x}) = \sqrt{k_{g_n}(\boldsymbol{x}, \boldsymbol{x})}$. The posterior variance function of I_A, denoted by $\sigma_{I_{A,n}}^2(\boldsymbol{x}) : \mathcal{X} \mapsto \mathbb{R}$, is given by:

$$\sigma_{I_{A,n}}^2(\boldsymbol{x}) = \boldsymbol{\Phi}\left(\frac{y - m_{g_n}(\boldsymbol{x})}{\sigma_{g_n}(\boldsymbol{x})}\right)\boldsymbol{\Phi}\left(-\frac{y - m_{g_n}(\boldsymbol{x})}{\sigma_{g_n}(\boldsymbol{x})}\right). \tag{14}$$

Similarly, the posterior distribution of $I_{\overline{A}}$ conditional on $\boldsymbol{\mathcal{D}}$ also follows a GBP:

$$I_{\overline{A},n}(\boldsymbol{x}) \sim \mathcal{GBP}(m_{I_{\overline{A},n}}(\boldsymbol{x}), k_{I_{\overline{A},n}}(\boldsymbol{x}, \boldsymbol{x}')), \tag{15}$$

where $I_{\overline{A},n}$ denotes the posterior distribution of $I_{\overline{A}}$; $m_{I_{\overline{A},n}}(\boldsymbol{x}) : \mathcal{X} \mapsto \mathbb{R}$ and $k_{I_{\overline{A},n}}(\boldsymbol{x}, \boldsymbol{x}') : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$ are the posterior mean and covariance functions respectively, which are given by:

$$m_{I_{\overline{A},n}}(\boldsymbol{x}) = \boldsymbol{\Phi}\left(-\frac{y - m_{g_n}(\boldsymbol{x})}{\sigma_{g_n}(\boldsymbol{x})}\right), \tag{16}$$

$$\begin{aligned} k_{I_{\overline{A},n}}(\boldsymbol{x}, \boldsymbol{x}') =& \boldsymbol{\Phi}_2\left(\begin{bmatrix} y \\ y \end{bmatrix}; \begin{bmatrix} m_{g_n}(\boldsymbol{x}) \\ m_{g_n}(\boldsymbol{x}') \end{bmatrix}, \begin{bmatrix} \sigma_{g_n}^2(\boldsymbol{x}) & k_{g_n}(\boldsymbol{x}, \boldsymbol{x}') \\ k_{g_n}(\boldsymbol{x}', \boldsymbol{x}) & \sigma_{g_n}^2(\boldsymbol{x}') \end{bmatrix}\right) \\ & - \boldsymbol{\Phi}\left(\frac{y - m_{g_n}(\boldsymbol{x})}{\sigma_{g_n}(\boldsymbol{x})}\right)\boldsymbol{\Phi}\left(\frac{y - m_{g_n}(\boldsymbol{x}')}{\sigma_{g_n}(\boldsymbol{x}')}\right). \end{aligned} \tag{17}$$

The posterior variance function of $I_{\overline{A}}$, denoted by $\sigma_{I_{\overline{A},n}}^2(\boldsymbol{x}) : \mathcal{X} \mapsto \mathbb{R}$, reads:

$$\sigma_{I_{\overline{A},n}}^2(\boldsymbol{x}) = \boldsymbol{\Phi}\left(\frac{y - m_{g_n}(\boldsymbol{x})}{\sigma_{g_n}(\boldsymbol{x})}\right)\boldsymbol{\Phi}\left(-\frac{y - m_{g_n}(\boldsymbol{x})}{\sigma_{g_n}(\boldsymbol{x})}\right). \tag{18}$$

We note that $k_{I_{A,n}}(\boldsymbol{x}, \boldsymbol{x}') = k_{I_{\overline{A},n}}(\boldsymbol{x}, \boldsymbol{x}')$ and $\sigma_{I_{A,n}}^2(\boldsymbol{x}) = \sigma_{I_{\overline{A},n}}^2(\boldsymbol{x})$ hold.

The posterior distribution of ζ_B conditional on $\boldsymbol{\mathcal{D}}$ follows a Dirac delta process (DDP):

$$\zeta_{B,n}(\boldsymbol{x}) \sim \delta(y - g_n(\boldsymbol{x})), \tag{19}$$

where $\zeta_{B,n}$ denotes the posterior distribution of ζ_B. The posterior mean and covariance functions of ζ_B can be derived as follows:

$$\begin{aligned} m_{\zeta_{B,n}}(\boldsymbol{x}) =& \mathbb{E}\left[\zeta_{B,n}(\boldsymbol{x})\right] \\ =& \mathbb{E}\left[\delta(y - g_n(\boldsymbol{x}))\right] \\ =& \int_{-\infty}^{+\infty} \delta(y - g_{n,i}(\boldsymbol{x})) f_{g_n(\boldsymbol{x})}(g_{n,i}(\boldsymbol{x})) \mathrm{d}g_{n,i}(\boldsymbol{x}) \\ =& f_{g_n(\boldsymbol{x})}(y) \\ =& \frac{1}{\sigma_{g_n}(\boldsymbol{x})}\phi\left(\frac{y - m_{g_n}(\boldsymbol{x})}{\sigma_{g_n}(\boldsymbol{x})}\right), \end{aligned} \tag{20}$$

$$\begin{aligned} k_{\zeta_{B,n}}(\boldsymbol{x}, \boldsymbol{x}') =& \mathbb{E}\left[\left(\zeta_{B,n}(\boldsymbol{x}) - m_{\zeta_{B,n}}(\boldsymbol{x})\right)\left(\zeta_{B,n}(\boldsymbol{x}') - m_{\zeta_{B,n}}(\boldsymbol{x}')\right)\right] \\ =& \mathbb{E}\left[\zeta_{B,n}(\boldsymbol{x})\zeta_{B,n}(\boldsymbol{x}')\right] - \mathbb{E}\left[\zeta_{B,n}(\boldsymbol{x})\right]\mathbb{E}\left[\zeta_{B,n}(\boldsymbol{x}')\right] \\ =& \mathbb{E}\left[\delta(y - g_n(\boldsymbol{x}))\delta(y - g_n(\boldsymbol{x}'))\right] - m_{\zeta_{B,n}}(\boldsymbol{x}) m_{\zeta_{B,n}}(\boldsymbol{x}') \\ =& \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} \delta(y - g_{n,i}(\boldsymbol{x}))\delta(y - g_{n,j}(\boldsymbol{x}')) f_{g_n(\boldsymbol{x}), g_n(\boldsymbol{x}')} \\ & \times \left(g_{n,i}(\boldsymbol{x}), g_{n,j}(\boldsymbol{x}')\right) \mathrm{d}g_{n,i}(\boldsymbol{x}) \mathrm{d}g_{n,i}(\boldsymbol{x}') \\ & - m_{\zeta_{B,n}}(\boldsymbol{x}) m_{\zeta_{B,n}}(\boldsymbol{x}') \\ =& f_{g_n(\boldsymbol{x}), g_n(\boldsymbol{x}')}(y, y) - m_{\zeta_{B,n}}(\boldsymbol{x}) m_{\zeta_{B,n}}(\boldsymbol{x}') \\ =& \phi_2\left(\begin{bmatrix} y \\ y \end{bmatrix}; \begin{bmatrix} m_{g_n}(\boldsymbol{x}) \\ m_{g_n}(\boldsymbol{x}') \end{bmatrix}, \begin{bmatrix} \sigma_{g_n}^2(\boldsymbol{x}) & k_{g_n}(\boldsymbol{x}, \boldsymbol{x}') \\ k_{g_n}(\boldsymbol{x}', \boldsymbol{x}) & \sigma_{g_n}^2(\boldsymbol{x}') \end{bmatrix}\right) \\ & - \frac{1}{\sigma_{g_n}(\boldsymbol{x})}\phi\left(\frac{y - m_{g_n}(\boldsymbol{x})}{\sigma_{g_n}(\boldsymbol{x})}\right)\frac{1}{\sigma_{g_n}(\boldsymbol{x}')}\phi\left(\frac{y - m_{g_n}(\boldsymbol{x}')}{\sigma_{g_n}(\boldsymbol{x}')}\right), \end{aligned} \tag{21}$$

where $g_{n,i}$ and $g_{n,j}$ represents two realizations of g_n; $f_{g_n(\boldsymbol{x})}$ is the PDF of $g_n(\boldsymbol{x})$; $f_{g_n(\boldsymbol{x}), g_n(\boldsymbol{x}')}$ is the joint PDF of $g_n(\boldsymbol{x})$ and $g_n(\boldsymbol{x}')$; ϕ_2 is the bi-variate normal PDF. The posterior variance function of ζ_B is expressed as:

$$\sigma_{\zeta_{B,n}}^2(\boldsymbol{x}) = \left[\delta(0) - \frac{1}{\sigma_{g_n}(\boldsymbol{x})}\phi\left(\frac{y - m_{g_n}(\boldsymbol{x})}{\sigma_{g_n}(\boldsymbol{x})}\right)\right]\frac{1}{\sigma_{g_n}(\boldsymbol{x})}\phi\left(\frac{y - m_{g_n}(\boldsymbol{x})}{\sigma_{g_n}(\boldsymbol{x})}\right). \tag{22}$$

As $\delta(0) = \infty$, the posterior variance function of ζ_B does not exist.

The posterior distribution (denoted by $F_{Y,n}$) of F_Y can be generated by considering the push-forward of $I_{A,n}$ by the integration operator. However, the exact type of this distribution is not known. Fortunately, we can derive the posterior mean and variance functions by applying Fubini's theorem as follows:

$$\begin{aligned} m_{F_{Y,n}}(y) =& \mathbb{E}\left[F_{Y,n}(y)\right] \\ =& \mathbb{E}\left[\int_{\mathcal{X}} I_{A,n}(\boldsymbol{x}) f_{\boldsymbol{X}}(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}\right] \\ =& \int_{\mathcal{X}} \mathbb{E}\left[I_{A,n}(\boldsymbol{x})\right] f_{\boldsymbol{X}}(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} \\ =& \int_{\mathcal{X}} \boldsymbol{\Phi}\left(\frac{y - m_{g_n}(\boldsymbol{x})}{\sigma_{g_n}(\boldsymbol{x})}\right) f_{\boldsymbol{X}}(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}, \end{aligned} \tag{23}$$

$$
\begin{aligned}
&\sigma^2_{F_{Y,n}}(y) = \mathbb{V}\left[F_{Y,n}(y)\right] \\
&= \mathbb{E}\left[\left(F_{Y,n}(y) - m_{F_{Y,n}}(y)\right)^2\right] \\
&= \mathbb{E}\left[\left(\int_{\mathcal{X}} I_{A,n}(\boldsymbol{x}) f_{\boldsymbol{X}}(\boldsymbol{x})\mathrm{d}\boldsymbol{x} - \int_{\mathcal{X}} \mathbb{E}\left[I_{A,n}(\boldsymbol{x})\right] f_{\boldsymbol{X}}(\boldsymbol{x})\mathrm{d}\boldsymbol{x}\right)^2\right] \\
&= \mathbb{E}\left[\left(\int_{\mathcal{X}} \left(I_{A,n}(\boldsymbol{x}) - \mathbb{E}\left[I_{A,n}(\boldsymbol{x})\right]\right) f_{\boldsymbol{X}}(\boldsymbol{x})\mathrm{d}\boldsymbol{x}\right)^2\right] \\
&= \mathbb{E}\left[\left(\int_{\mathcal{X}} \left[I_{A,n}(\boldsymbol{x}) - \mathbb{E}\left[I_{A,n}(\boldsymbol{x})\right]\right] f_{\boldsymbol{X}}(\boldsymbol{x})\mathrm{d}\boldsymbol{x}\right)\left(\int_{\mathcal{X}} \left[I_{A,n}(\boldsymbol{x}') - \mathbb{E}\left[I_{A,n}(\boldsymbol{x}')\right]\right] f_{\boldsymbol{X}}(\boldsymbol{x}')\mathrm{d}\boldsymbol{x}'\right)\right] \\
&= \mathbb{E}\left[\left(\int_{\mathcal{X}}\int_{\mathcal{X}} \left(I_{A,n}(\boldsymbol{x}) - \mathbb{E}\left[I_{A,n}(\boldsymbol{x})\right]\right)\left(I_{A,n}(\boldsymbol{x}') - \mathbb{E}\left[I_{A,n}(\boldsymbol{x}')\right]\right) f_{\boldsymbol{X}}(\boldsymbol{x}) f_{\boldsymbol{X}}(\boldsymbol{x}')\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{x}'\right)\right] \\
&= \int_{\mathcal{X}}\int_{\mathcal{X}} \mathbb{E}\left[\left(I_{A,n}(\boldsymbol{x}) - \mathbb{E}\left[I_{A,n}(\boldsymbol{x})\right]\right)\left(I_{A,n}(\boldsymbol{x}') - \mathbb{E}\left[I_{A,n}(\boldsymbol{x}')\right]\right)\right] f_{\boldsymbol{X}}(\boldsymbol{x}) f_{\boldsymbol{X}}(\boldsymbol{x}')\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{x}' \\
&= \int_{\mathcal{X}}\int_{\mathcal{X}} k_{I_{A,n}}(\boldsymbol{x},\boldsymbol{x}') f_{\boldsymbol{X}}(\boldsymbol{x}) f_{\boldsymbol{X}}(\boldsymbol{x}')\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{x}' \\
&= \int_{\mathcal{X}}\int_{\mathcal{X}} \left\{ \begin{array}{c} \Phi_2\left(\begin{bmatrix} y \\ y \end{bmatrix}; \begin{bmatrix} m_{g_n}(\boldsymbol{x}) \\ m_{g_n}(\boldsymbol{x}') \end{bmatrix}, \begin{bmatrix} \sigma^2_{g_n}(\boldsymbol{x}) & k_{g_n}(\boldsymbol{x},\boldsymbol{x}') \\ k_{g_n}(\boldsymbol{x}',\boldsymbol{x}) & \sigma^2_{g_n}(\boldsymbol{x}') \end{bmatrix}\right) \\ -\Phi\left(\frac{y-m_{g_n}(\boldsymbol{x})}{\sigma_{g_n}(\boldsymbol{x})}\right)\Phi\left(\frac{y-m_{g_n}(\boldsymbol{x}')}{\sigma_{g_n}(\boldsymbol{x}')}\right) \end{array} \right\} \\
&\quad \times f_{\boldsymbol{X}}(\boldsymbol{x}) f_{\boldsymbol{X}}(\boldsymbol{x}')\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{x}',
\end{aligned}
\tag{24}
$$

where $\mathbb{V}$ is the variance operator. To avoid the complexity of calculating the posterior variance, an upper bound on $\sigma^2_{F_{Y,n}}(y)$ is derived using the Cauchy–Schwarz inequality (i.e., $k_{I_{A,n}}(\boldsymbol{x},\boldsymbol{x}') \le \sigma_{I_{A,n}}(\boldsymbol{x})\sigma_{I_{A,n}}(\boldsymbol{x}')$):

$$
\begin{aligned}
\overline{\sigma}^2_{F_{Y,n}}(y) &= \int_{\mathcal{X}}\int_{\mathcal{X}} \sigma_{I_{A,n}}(\boldsymbol{x})\sigma_{I_{A,n}}(\boldsymbol{x}') f_{\boldsymbol{X}}(\boldsymbol{x}) f_{\boldsymbol{X}}(\boldsymbol{x}')\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{x}' \\
&= \left(\int_{\mathcal{X}} \sigma_{I_{A,n}}(\boldsymbol{x}) f_{\boldsymbol{X}}(\boldsymbol{x})\mathrm{d}\boldsymbol{x}\right)^2 \\
&= \left(\int_{\mathcal{X}} \sqrt{\Phi\left(\frac{y-m_{g_n}(\boldsymbol{x})}{\sigma_{g_n}(\boldsymbol{x})}\right)\Phi\left(-\frac{y-m_{g_n}(\boldsymbol{x})}{\sigma_{g_n}(\boldsymbol{x})}\right)} f_{\boldsymbol{X}}(\boldsymbol{x})\mathrm{d}\boldsymbol{x}\right)^2.
\end{aligned}
\tag{25}
$$

Analogously, the posterior mean and variance functions of $\overline{F}_Y$ are given by:

$$
m_{\overline{F}_{Y,n}}(y) = \int_{\mathcal{X}} \Phi\left(-\frac{y-m_{g_n}(\boldsymbol{x})}{\sigma_{g_n}(\boldsymbol{x})}\right) f_{\boldsymbol{X}}(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x},
\tag{26}
$$

$$
\begin{aligned}
&\sigma^2_{\overline{F}_{Y,n}}(y) \\
&= \int_{\mathcal{X}}\int_{\mathcal{X}} \left\{ \begin{array}{c} \Phi_2\left(\begin{bmatrix} y \\ y \end{bmatrix}; \begin{bmatrix} m_{g_n}(\boldsymbol{x}) \\ m_{g_n}(\boldsymbol{x}') \end{bmatrix}, \begin{bmatrix} \sigma^2_{g_n}(\boldsymbol{x}) & k_{g_n}(\boldsymbol{x},\boldsymbol{x}') \\ k_{g_n}(\boldsymbol{x}',\boldsymbol{x}) & \sigma^2_{g_n}(\boldsymbol{x}') \end{bmatrix}\right) \\ -\Phi\left(\frac{y-m_{g_n}(\boldsymbol{x})}{\sigma_{g_n}(\boldsymbol{x})}\right)\Phi\left(\frac{y-m_{g_n}(\boldsymbol{x}')}{\sigma_{g_n}(\boldsymbol{x}')}\right) \end{array} \right\} \\
&\quad \times f_{\boldsymbol{X}}(\boldsymbol{x}) f_{\boldsymbol{X}}(\boldsymbol{x}')\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{x}'.
\end{aligned}
\tag{27}
$$

Furthermore, an upper bound on $\sigma^2_{\overline{F}_{Y,n}}(y)$ is available:

$$
\overline{\sigma}^2_{\overline{F}_{Y,n}}(y) = \left(\int_{\mathcal{X}} \sqrt{\Phi\left(\frac{y-m_{g_n}(\boldsymbol{x})}{\sigma_{g_n}(\boldsymbol{x})}\right)\Phi\left(-\frac{y-m_{g_n}(\boldsymbol{x})}{\sigma_{g_n}(\boldsymbol{x})}\right)} f_{\boldsymbol{X}}(\boldsymbol{x})\mathrm{d}\boldsymbol{x}\right)^2.
\tag{28}
$$

It should be noted that there exist $\sigma^2_{\overline{F}_{Y,n}}(y) = \sigma^2_{F_{Y,n}}(y)$ and $\overline{\sigma}^2_{\overline{F}_{Y,n}}(y) = \overline{\sigma}^2_{F_{Y,n}}(y)$.

Likewise, we can obtain the posterior mean and variance functions of f_Y:

$$
m_{f_{Y,n}}(y) = \int_{\mathcal{X}} \frac{1}{\sigma_{g_n}(\boldsymbol{x})}\phi\left(\frac{y-m_{g_n}(\boldsymbol{x})}{\sigma_{g_n}(\boldsymbol{x})}\right) f_{\boldsymbol{X}}(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x},
\tag{29}
$$

$$
\begin{aligned}
&\sigma^2_{f_{Y,n}}(y) \\
&= \int_{\mathcal{X}}\int_{\mathcal{X}} \left\{ \begin{array}{c} \phi_2\left(\begin{bmatrix} y \\ y \end{bmatrix}; \begin{bmatrix} m_{g_n}(\boldsymbol{x}) \\ m_{g_n}(\boldsymbol{x}') \end{bmatrix}, \begin{bmatrix} \sigma^2_{g_n}(\boldsymbol{x}) & k_{g_n}(\boldsymbol{x},\boldsymbol{x}') \\ k_{g_n}(\boldsymbol{x}',\boldsymbol{x}) & \sigma^2_{g_n}(\boldsymbol{x}') \end{bmatrix}\right) \\ -\frac{1}{\sigma_{g_n}(\boldsymbol{x})}\phi\left(\frac{y-m_{g_n}(\boldsymbol{x})}{\sigma_{g_n}(\boldsymbol{x})}\right)\frac{1}{\sigma_{g_n}(\boldsymbol{x}')}\phi\left(\frac{y-m_{g_n}(\boldsymbol{x}')}{\sigma_{g_n}(\boldsymbol{x}')}\right) \end{array} \right\} \\
&\quad \times f_{\boldsymbol{X}}(\boldsymbol{x}) f_{\boldsymbol{X}}(\boldsymbol{x}')\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{x}'.
\end{aligned}
\tag{30}
$$

In addition, an upper bound on $\sigma^2_{f_{Y,n}}(y)$ can also be derived:

$$
\begin{aligned}
&\overline{\sigma}^2_{f_{Y,n}}(y) \\
&= \left(\int_{\mathcal{X}} \sqrt{\left[\delta(0) - \frac{1}{\sigma_{g_n}(\boldsymbol{x})}\phi\left(\frac{y-m_{g_n}(\boldsymbol{x})}{\sigma_{g_n}(\boldsymbol{x})}\right)\right]\frac{1}{\sigma_{g_n}(\boldsymbol{x})}\phi\left(\frac{y-m_{g_n}(\boldsymbol{x})}{\sigma_{g_n}(\boldsymbol{x})}\right)} f_{\boldsymbol{X}}(\boldsymbol{x})\mathrm{d}\boldsymbol{x}\right)^2.
\end{aligned}
\tag{31}
$$

Eq. (31) implies that the upper bound of $\sigma^2_{f_{Y,n}}(y)$ goes to infinity, so it is not meaningful.

The posterior mean functions of the response CDF, CCDF and PDF can naturally provide point estimates, while the posterior variance functions (or their upper bound functions, if they exist) can serve as measures of uncertainty for the response probability distributions.

3.2.4. Numerical treatment of posterior statistics

Note that all posterior statistics of F_Y, $\overline{F}_Y$ and f_Y are not analytically tractable (if exist) and thus require numerical treatment. In this study, Monte Carlo simulation (MCS) is employed. Taking the posterior statistics of F_Y as an example, the estimators of $m_{F_{Y,n}}(y)$ and $\overline{\sigma}_{F_{Y,n}}(y)$ are given by:

$$
\hat{m}_{F_{Y,n}}(y) = \frac{1}{N}\sum_{j=1}^{N} \Phi\left(\frac{y-m_{g_n}\left(\boldsymbol{x}^{(j)}\right)}{\sigma_{g_n}\left(\boldsymbol{x}^{(j)}\right)}\right),
\tag{32}
$$

$$
\hat{\overline{\sigma}}_{F_{Y,n}}(y) = \frac{1}{N}\sum_{j=1}^{N} \sqrt{\Phi\left(\frac{y-m_{g_n}\left(\boldsymbol{x}^{(j)}\right)}{\sigma_{g_n}\left(\boldsymbol{x}^{(j)}\right)}\right)\Phi\left(-\frac{y-m_{g_n}\left(\boldsymbol{x}^{(j)}\right)}{\sigma_{g_n}\left(\boldsymbol{x}^{(j)}\right)}\right)},
\tag{33}
$$

where $\left\{\boldsymbol{x}^{(j)}\right\}_{j=1}^{N}$ is a set of N random samples generated according to $f_{\boldsymbol{X}}(\boldsymbol{x})$. The variances of $\hat{m}_{F_{Y,n}}(y)$ and $\hat{\overline{\sigma}}_{F_{Y,n}}(y)$ can be expressed as:

$$
\mathbb{V}\left[\hat{m}_{F_{Y,n}}(y)\right] = \frac{1}{N(N-1)}\sum_{j=1}^{N}\left[\Phi\left(\frac{y-m_{g_n}\left(\boldsymbol{x}^{(j)}\right)}{\sigma_{g_n}\left(\boldsymbol{x}^{(j)}\right)}\right) - \hat{m}_{F_{Y,n}}(y)\right]^2,
\tag{34}
$$

$$
\begin{aligned}
&\mathbb{V}\left[\hat{\overline{\sigma}}_{F_{Y,n}}(y)\right] = \\
&\frac{1}{N(N-1)}\sum_{j=1}^{N}\left[\sqrt{\Phi\left(\frac{y-m_{g_n}\left(\boldsymbol{x}^{(j)}\right)}{\sigma_{g_n}\left(\boldsymbol{x}^{(j)}\right)}\right)\Phi\left(-\frac{y-m_{g_n}\left(\boldsymbol{x}^{(j)}\right)}{\sigma_{g_n}\left(\boldsymbol{x}^{(j)}\right)}\right)} - \hat{\overline{\sigma}}_{F_{Y,n}}(y)\right]^2.
\end{aligned}
\tag{35}
$$

Following the practical Bayesian framework described above, one can obtain point estimates for the response probability distributions, along with their associated uncertainty measure estimates, given the data $\boldsymbol{\mathcal{D}}$. The design of computer experiments thus is important for the accuracy and efficiency, which involves the determination of the number and locations of the input samples.

4. Bayesian active learning of response probability distributions

In this section, a Bayesian active learning method is developed based on the practical Bayesian framework for estimating response probability distributions. This implies that the response probability distributions are estimated in an iterative way until a certain criterion is fulfilled. Since an upper bound on the posterior variance function for the response PDF does not exist, we will focus on the response CDF and CCDF in the active learning process, with the response PDF being obtained as a by-product at the end.

4.1. Stopping criterion

One of the critical challenges in Bayesian active learning is determining when to stop the iterative process, known as stopping criterion. In general, formulating a stopping criterion might depend on several considerations, such as the primary goal, available resources, and other factors. Hereto, the accuracy of the response CDF and CCDF is of great interest.

A natural measure of the accuracy of the response CDF is its posterior coefficient of variation (CoV) function. However, such a measure involves the posterior variance function of the response CDF, which can be computationally demanding. Alternatively, we use the upper bound of the posterior CoV function of the response CDF as a measure of its accuracy, which is defined as:

$$\overline{\mathrm{CoV}}_{F_{Y,n}}(y) = \frac{\overline{\sigma}_{F_{Y,n}}(y)}{m_{F_{Y,n}}(y)}, \tag{36}$$

where $m_{F_{Y,n}}(y)$ is given in Eq. (23) and $\overline{\sigma}_{F_{Y,n}}(y)$ is given in Eq. (25). Likewise, the upper bound of the posterior CoV function of the response CCDF is used as a measure of its accuracy:

$$\overline{\mathrm{CoV}}_{\overline{F}_{Y,n}}(y) = \frac{\overline{\sigma}_{\overline{F}_{Y,n}}(y)}{m_{\overline{F}_{Y,n}}(y)}, \tag{37}$$

where $m_{\overline{F}_{Y,n}}(y)$ is given in Eq. (26) and $\overline{\sigma}_{\overline{F}_{Y,n}}(y)$ is given in Eq. (28).

In order to ensure the accuracy of both the response CDF and CCDF, we introduce the following stopping criterion in this study:

$$\max_{y\in\mathcal{Y}} H_n(y) < \epsilon, \tag{38}$$

where $H_n(y) = \max\left(\overline{\mathrm{CoV}}_{F_{Y,n}}(y), \overline{\mathrm{CoV}}_{\overline{F}_{Y,n}}(y)\right)$; ϵ is a user-specified threshold. This stopping criterion means that the iterative process stops as soon as the upper bounds of the posterior CoV functions of both the response CDF and CCDF for any $y \in \mathcal{Y}$ are less than a predefined threshold ϵ.

Remark 1. If greater accuracy in the left tail of the sought distribution is of particular interest, the stopping criterion can be defined as $\max_{y\in\mathcal{Y}} \overline{\mathrm{CoV}}_{F_{Y,n}}(y) < \epsilon$. On the contrary, if the right tail is of greater interest, the stopping criterion can instead be defined as $\max_{y\in\mathcal{Y}} \overline{\mathrm{CoV}}_{\overline{F}_{Y,n}}(y) < \epsilon$.

4.2. Learning function

Another critical component in Bayesian active learning is the mechanism or strategy used to select the most informative data points to evaluate next, known as the learning (or acquisition) function. This function comes into play when the stopping criterion (Ineq. (38)) is not met. The desired learning function should be able to suggest promising points which, once evaluated, are expected to decrease the value of the left-hand side term of Ineq. (38).

First, we identify a critical location at which $H_n(y)$ has the largest value:

$$y^\star = \operatorname*{argmax}_{y\in\mathcal{Y}} H_n(y). \tag{39}$$

Then, we can define a new learning function, denoted by $L_n : \mathcal{X} \mapsto \mathbb{R}$:

$$L_n(\boldsymbol{x}|y^\star) = \sqrt{\Phi\left(\frac{y^\star - m_{g_n}(\boldsymbol{x})}{\sigma_{g_n}(\boldsymbol{x})}\right)\Phi\left(-\frac{y^\star - m_{g_n}(\boldsymbol{x})}{\sigma_{g_n}(\boldsymbol{x})}\right)} f_{\boldsymbol{X}}(\boldsymbol{x}). \tag{40}$$

Note that this learning function is taken from the integrand of the upper bound on the posterior standard deviation function of F_Y and $\overline{F}_Y$ at $y = y^\star$, i.e., $\int_{\mathcal{X}} L_n(\boldsymbol{x}|y^\star)\mathrm{d}\boldsymbol{x} = \overline{\sigma}_{F_{Y,n}}(y^\star) = \overline{\sigma}_{\overline{F}_{Y,n}}(y^\star)$. Therefore, it can be interpreted as a measure of the contribution at $\boldsymbol{x}$ to the total value of $\overline{\sigma}_{F_{Y,n}}(y^\star)$ or $\overline{\sigma}_{\overline{F}_{Y,n}}(y^\star)$.

Having defined the learning function, the next best point to evaluate the g function can be selected by:

$$\boldsymbol{x}^{(n+1)} = \operatorname*{argmax}_{\boldsymbol{x}\in\mathcal{X}} L_n(\boldsymbol{x}). \tag{41}$$

Remark 2. The proposed strategy involves a two-step learning function, similar to the one in [24]. However, in our method, both the learning function for identifying the critical location and the one for selecting the next best input point are directly related to the accuracy measure derived from the posterior statistics of the response CDF and CCDF (outlined in Section 4.1). Furthermore, our approach eliminates the need for kernel selection, simplifying implementation while maintaining effectiveness.

Remark 3. It is also possible to bypass the two-step strategy by directly maximizing $L_n(\boldsymbol{x}|y)$. However, this approach is not explored further in this study.

4.3. Implementation procedure of the proposed method

The implementation procedure of the proposed Bayesian active learning method can be summarized in seven main steps, accompanied by a flowchart in Fig. 2.

Step 1: Generate N samples according to $f_{\boldsymbol{X}}(\boldsymbol{x})$

Since several intractable integrals (i.e., $m_{F_{Y,n}}(y)$, $m_{\overline{F}_{Y,n}}(y)$, $m_{f_{Y,n}}(y)$, $\overline{\sigma}_{F_{Y,n}}(y)$ ($\overline{\sigma}_{\overline{F}_{Y,n}}(y)$)) entail numerical integration under the same density $f_{\boldsymbol{X}}(\boldsymbol{x})$, we first generate a set of N samples according to $f_{\boldsymbol{X}}(\boldsymbol{x})$ using a suitable low-discrepancy sequence (Sobol sequence in this study), which are denoted as $\left\{\boldsymbol{x}^{(j)}\right\}_{j=1}^{N}$. The number of samples N depends on the expected statistical error for approximating these integrals.

Step 2: Define an initial observation dataset

The Bayesian active learning process needs to be initialized with an initial set of observations. First, a small number (say n_0) of uniform points, $\boldsymbol{\mathcal{X}} = \left\{\boldsymbol{x}^{(i)}\right\}_{i=1}^{n_0}$, are generated within a hyper-rectangular $\Lambda_1 = \prod_{r=1}^{d}\left[a_r, b_r\right] \subseteq \mathcal{X}$ using an appropriate low-discrepancy sequence (Hammersley sequence in this study). In this study, the lower and upper bounds in the rth dimension are specified by: $a_r = F_{X_r}^{-1}(\rho)$ and $b_r = F_{X_r}^{-1}(1-\rho_1)$, where F_{X_r} denotes the marginal CDF of X_r and ρ_1 is a small truncation probability. Second, the g-function is evaluated at $\boldsymbol{\mathcal{X}}$ to produce the corresponding response values, i.e., $\boldsymbol{\mathcal{Y}} = \left\{y^{(i)}\right\}_{i=1}^{n_0}$ with $y^{(i)} = g(\boldsymbol{x}^{(i)})$. Finally, the initial dataset is formed by $\boldsymbol{\mathcal{D}} = \{\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{Y}}\}$. Let $n = n_0$.

Step 3: Obtain the posterior statistics of g

This step involves obtaining the posterior GP of g conditional on $\boldsymbol{\mathcal{D}}$, i.e., $g_n(\boldsymbol{x}) \sim \mathcal{GP}(m_{g_n}(\boldsymbol{x}), k_{g_n}(\boldsymbol{x}, \boldsymbol{x}'))$. Such a task can be performed by many well-developed GP regression toolboxes. In this study, we employ the *fitrgp* function available in the Statistics and Machine Learning Toolbox of Matlab R2024a.

Step 4: Evaluate the posterior statistics of F_Y and $\overline{F}_Y$

Step 4.1 Evaluate the posterior mean and standard deviation functions of g at $\left\{\boldsymbol{x}^{(j)}\right\}_{j=1}^{N}$, i.e., $\boldsymbol{M} = \left\{m_{g_n}(\boldsymbol{x}^{(j)})\right\}_{j=1}^{N}$ and $\boldsymbol{\Xi} = \left\{\sigma_{g_n}\left(\boldsymbol{x}^{(j)}\right)\right\}_{j=1}^{N}$;

Step 4.2 Discretize the range of interest $\left[y_{\min}, y_{\max}\right]$ into h equally spaced sub-intervals, i.e., $y_{\min} = y_0 < y_1 < \cdots < y_h = y_{\max}$ with spacing $(y_{\max} - y_{\min})/h$. Note that the lower and upper bounds $y_{\min}$ and $y_{\max}$ are difficult to know a priori. Therefore, we propose to estimate them from the posterior GP of g. More specifically, $y_{\min}$ and $y_{\max}$ are specified by the p and $1-p$ quantiles of $\boldsymbol{M} - \lambda\boldsymbol{\Xi}$ and $\boldsymbol{M} + \lambda\boldsymbol{\Xi}$, respectively, where λ is introduced to account for the posterior standard deviation. In this manner, the region of interest of the response CDF and CCDF is expected to be that with probability within p and $1-p$, where the value of p can be specified according to the requirements of practical

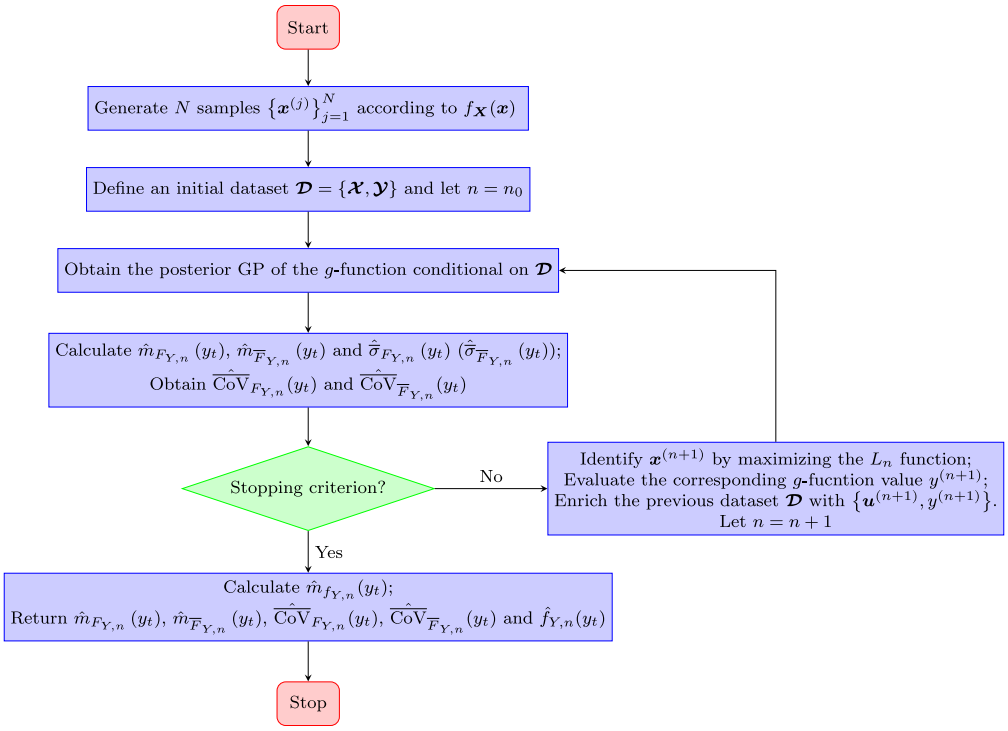


Fig. 2. Flowchart of the proposed BAL method.

applications.

Step 4.3 First, obtain the estimates $\hat{m}_{F_{Y,n}}(y_t)$, $\hat{m}_{\overline{F}_{Y,n}}(y_t)$ and $\hat{\bar{\sigma}}_{F_{Y,n}}(y_t)$ ($\hat{\bar{\sigma}}_{\overline{F}_{Y,n}}(y_t)$) by using $\boldsymbol{M}$ and $\boldsymbol{\Xi}$, $t = 0, 1, \ldots, h$. Then, calculate the upper bound estimates on the posterior CoV functions of the response CDF and CCDF by $\widehat{\overline{\mathrm{CoV}}}_{F_{Y,n}}(y_t) = \frac{\hat{\bar{\sigma}}_{F_{Y,n}}(y_t)}{\hat{m}_{F_{Y,n}}(y_t)}$ and $\widehat{\overline{\mathrm{CoV}}}_{\overline{F}_{Y,n}}(y_t) = \frac{\hat{\bar{\sigma}}_{\overline{F}_{Y,n}}(y_t)}{\hat{m}_{\overline{F}_{Y,n}}(y_t)}$.

Step 5: Check the stopping criterion

If the stopping criterion $\max_{y_t} \hat{H}_n(y_t) < \epsilon$ is satisfied twice in a row, where $\hat{H}_n(y_t) = \max\left(\widehat{\overline{\mathrm{CoV}}}_{F_{Y,n}}(y_t), \widehat{\overline{\mathrm{CoV}}}_{\overline{F}_{Y,n}}(y_t)\right)$, then go to **Step 7**; otherwise, go to **Step 6**.

Step 6: Enrich the observation dataset

In this stage, the previous observation dataset needs to be enriched. First, identify the critical location $y^\star = \mathrm{argmax}_{y_t} \hat{H}_n(y_t)$ and identify the next best point $\boldsymbol{x}^{(n+1)}$ by $\boldsymbol{x}^{(n+1)} = \mathrm{argmax}_{\boldsymbol{x}\in\Lambda_2} L_n(\boldsymbol{x})$ (Genetic algorithm is used in this study), Λ_2 is specified in similar with Λ_1 by replacing ρ_1 with ρ_2. Then, the g-function is evaluated at $\boldsymbol{x}^{(n+1)}$ to obtain the corresponding output $y^{(n+1)}$, i.e., $y^{(n+1)} = g(\boldsymbol{x}^{(n+1)})$. At last, the previous dataset $\mathcal{D}$ is enriched with $\{\boldsymbol{x}^{(n+1)}, y^{(n+1)}\}$, i.e., $\mathcal{D} = \mathcal{D} \cup \{\boldsymbol{x}^{(n+1)}, y^{(n+1)}\}$. Let $n = n + 1$ and go to **Step 3**.

Step 7: Evaluate the posterior mean of f_Y and return the results

Calculate $\hat{m}_{f_{Y,n}}(y_t)$ using the current $\boldsymbol{M}$ and $\boldsymbol{\Xi}$, $t = 0, 1, \ldots, h$. Return $\hat{m}_{F_{Y,n}}(y_t)$, $\hat{m}_{\overline{F}_{Y,n}}(y_t)$, $\widehat{\overline{\mathrm{CoV}}}_{F_{Y,n}}(y_t)$, $\widehat{\overline{\mathrm{CoV}}}_{\overline{F}_{Y,n}}(y_t)$ and $\hat{f}_{Y,n}(y_t)$ as the final results.

5. Numerical examples

In this section, five numerical examples are investigated to demonstrate the performance of the proposed BAL method for estimating the response CDF, CCDF and PDF. The parameters of our method is set as follows: $N = 5 \times 10^5$, $n_0 = 10$, $\rho_1 = 10^{-5}$, $\rho_2 = 10^{-8}$, $h = 100$, $p = 5 \times 10^{-5}$, $\lambda = 2$ and $\epsilon = 0.20$. For comparison, one of the representative existing methods, called active learning- based GP (AL-GP) metamodelling method [24], is also conducted for the first three examples. The number of training candidates is set to 5×10^5 with other parameters specified in each example. For both the proposed BAL method and the AL-GP method, 20 independent runs are performed to assess their robustness. When no analytical (or semi-analytical) solution is available, MCS is used to produce a reference result.

5.1. Example 1: A toy example

The first example considers a toy example taken from [24]:

$$Y = g(\boldsymbol{X}) = \min\left[X_1 - X_2, X_1 + X_2\right], \tag{42}$$

where X_1 and X_2 are two independent standard normal variables. The CDF, CCDF and PDF of Y can be expressed as:

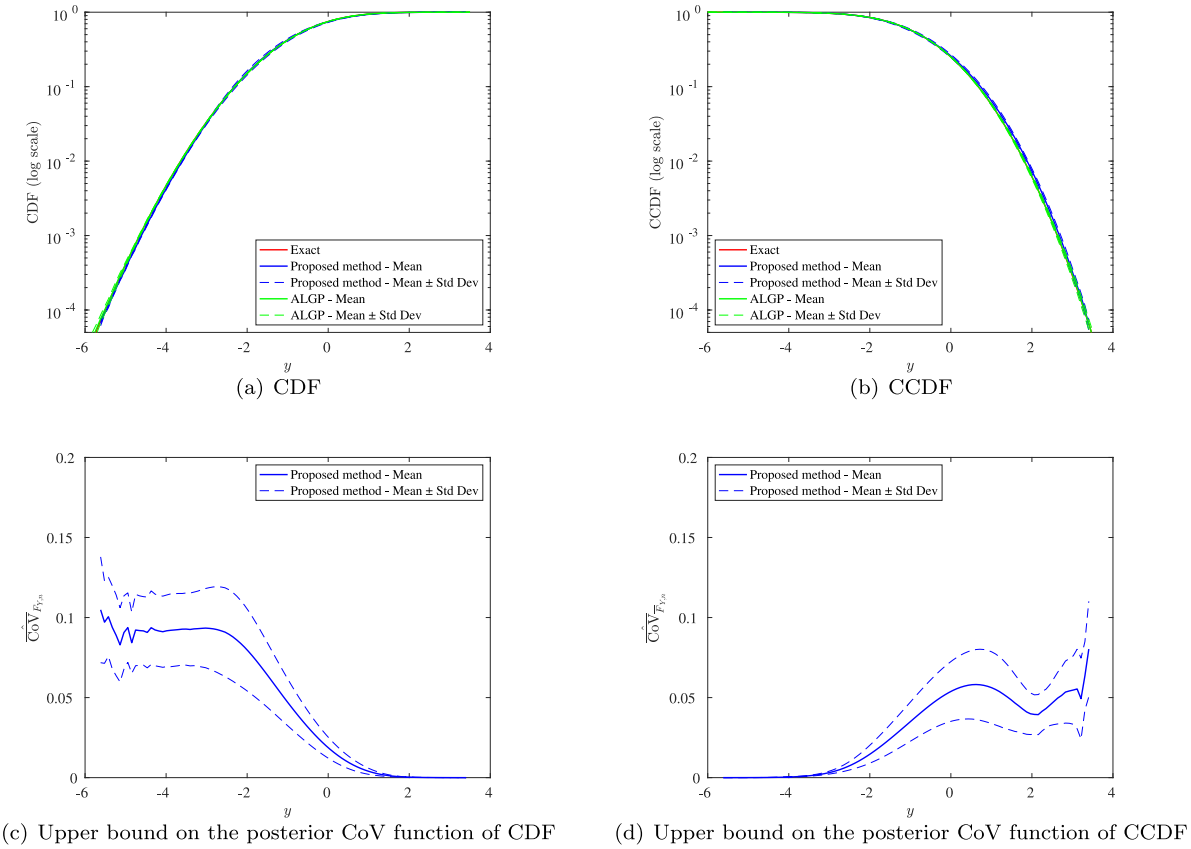
$$F_Y(y) = \boldsymbol{\Phi}\left(\frac{y}{\sqrt{2}}\right)\left(2 - \boldsymbol{\Phi}\left(\frac{y}{\sqrt{2}}\right)\right), \tag{43}$$

$$\overline{F}_Y(y) = 1 - \boldsymbol{\Phi}\left(\frac{y}{\sqrt{2}}\right)\left(2 - \boldsymbol{\Phi}\left(\frac{y}{\sqrt{2}}\right)\right), \tag{44}$$

$$f_Y(y) = \sqrt{2}\phi\left(\frac{y}{\sqrt{2}}\right)\left(1 - \boldsymbol{\Phi}\left(\frac{y}{\sqrt{2}}\right)\right). \tag{45}$$

Although $\boldsymbol{\Phi}$ does not have an analytical expression in terms of elementary functions, it can be computed accurately using numerical methods or approximations. Therefore, we still consider the results of Eqs. (43)–(45) as 'exact'.

For the AL-GP method, the range of interest is set to $[-6.0, 3.5]$ and the stopping criterion threshold is set to $\bar{\epsilon} = 0.15$. Fig. 3 shows the results of the response CDF and CCDF. As can be observed in Figs. 3(a) and 3(b), both the AL-GP method and the proposed BAL method produce CDF/CCDF mean curves that are close to the exact ones, along with notably narrow mean ± std dev bands. It is worth noting that the proposed method can provide the error measures for the response CDF/CCDF, i.e., the upper bounds of the CoV functions of the response CDF and CCDF, as depicted in Figs. 3(c) and 3(d), respectively. It can be seen that the mean curves and also the mean + std dev curves are all well below the $\epsilon = 0.20$ threshold. Additionally, the proposed method can also produce the response PDF as a by-product, with statistical results shown in Fig. 4. The mean curve is close to the



(a) CDF (b) CCDF

(c) Upper bound on the posterior CoV function of CDF (d) Upper bound on the posterior CoV function of CCDF

Fig. 3. Response CDF and CCDF for Example 1.

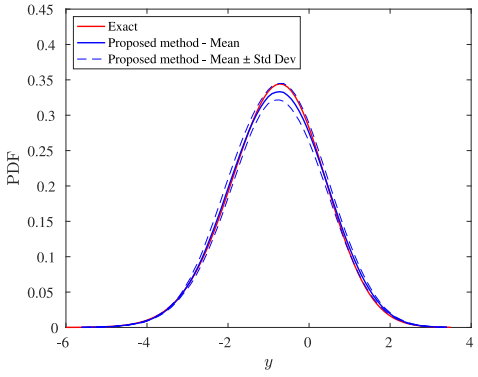


Fig. 4. Response PDF for Example 1.

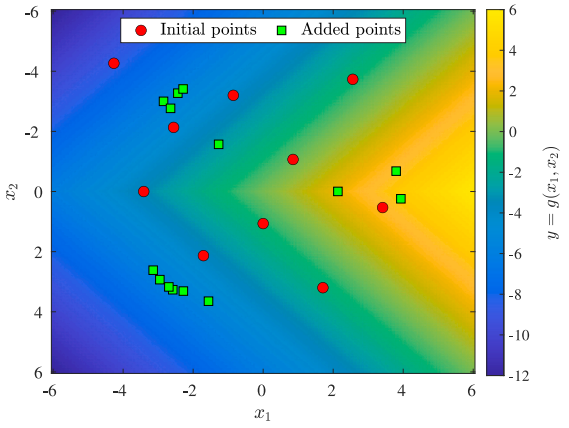


Fig. 5. Design of computer experiments for Example 1.

Table 1
Comparison of the number of g-function calls for Example 1.

Method	N_{call}	
	Mean	CoV
Proposed method	10 + 15.00 = 25.00	12.91%
AL-GP	12 + 33.20 = 45.20	50.74%

exact one and the mean ± std dev band is quite narrow. On average, the proposed method requires significantly fewer g-function calls than the AL-GP method (see Table 1). Additionally, the AL-GP method exhibits a higher CoV for the number of g-function evaluations compared to the proposed method.

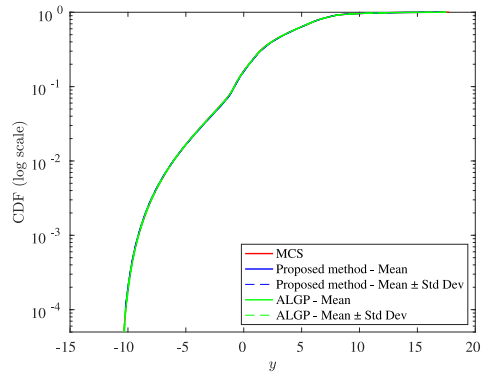
For illustrative purposes, the points selected during an arbitrary run of our approach are shown in Fig. 5. As can be seen, the 10 initial points are evenly distributed in the input space, as expected. During the active learning phase, only 14 additional points are identified before the proposed stopping criterion is met, which strategically chosen by maximizing the proposed learning function.

5.2. Example 2: The Ishigami function

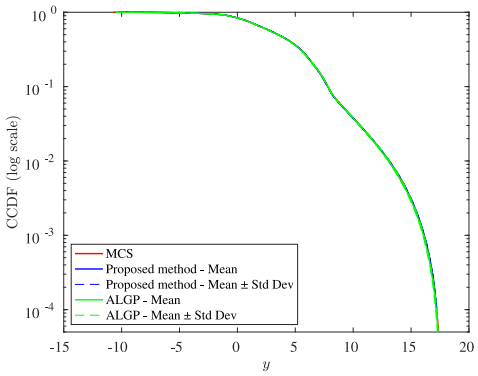
As a second example, consider the Ishigami function:

$$Y = \sin(X_1) + a\sin^2(X_2) + bX_3^4\sin(X_1), \tag{46}$$

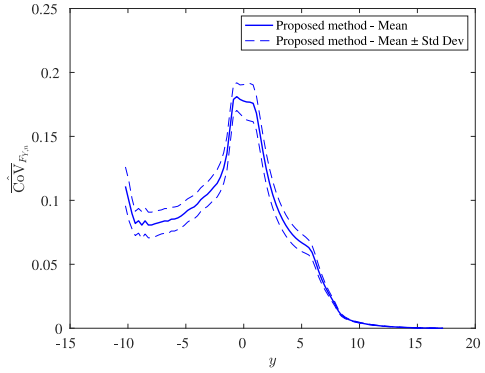
where $a = 7$ and $b = 0.1$; X_1, X_2 and X_3 are three independent uniform random variables within $[-\pi, \pi]$.



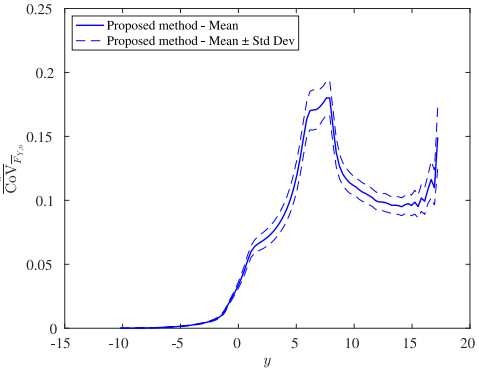
(a) CDF



(b) CCDF



(c) Upper bound on the posterior CoV function of CDF



(d) Upper bound on the posterior CoV function of CCDF

Fig. 6. Response CDF and CCDF for Example 2.

Table 2
Comparison of the number of g-function calls for Example 2.

Method	N_{call}	
	Mean	CoV
Proposed method	10 + 187.00 = 197.00	2.88%
AL-GP	12 + 226.55 = 238.55	3.93%

The reference solutions of the response CDF, CCDF and PDF are generated using MCS with 10^7 samples. The range of interest and stopping criterion threshold of the AL-GP method are set to $[-10.5, 17.5]$ and $\bar{\epsilon} = 0.15$, respectively. The results for the response CDF and CCDF are shown in Fig. 6. Figs. 6(a) and 6(b) indicate that both the AL-GP method and the proposed method can yield highly accurate response CDF and CCDF, even in the low-probability range. The proposed method is capable of providing the upper bounds of the posterior CoV functions of the response CDF and CCDF, as presented in Figs. 6(c) and 6(d) respectively. As can be observed, the mean curves are situated behind the threshold value $\epsilon = 0.20$, and the mean ± std dev bands are notably narrow. Fig. 7 depicts the by-product PDF generated by the proposed method, which is compared to the histogram produced by MCS. It is noteworthy that the proposed method is able to accurately capture the bi-modal shape. Table 2 indicates that the proposed method necessitates fewer g-function evaluations on average and exhibits a slightly smaller CoV than the AL-GP method.

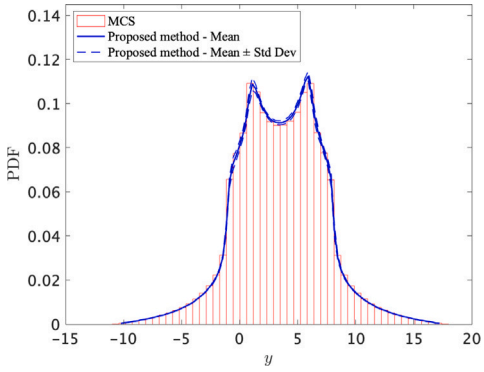


Fig. 7. Response PDF for Example 2.

5.3. Example 3: A nonlinear oscillator

The third numerical example involves a single-degree-of-freedom nonlinear oscillator under a rectangular pulse load [44], as shown in Fig. 8. The response function is defined in terms of the performance function of the oscillator:

$$Y = g(\boldsymbol{X}) = 3r - \left| \frac{2F_1}{k_1 + k_2} \sin\left(\frac{t_1}{2} \sqrt{\frac{k_1 + k_2}{m}} \right) \right|, \tag{47}$$

where m, k_1, k_2, r, F_1 and t_1 are six independent random variables, as listed in Table 3.

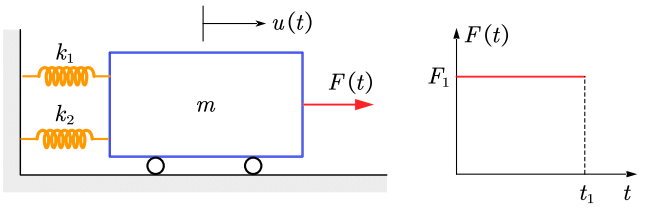


Fig. 8. A nonlinear oscillator under a rectangular pulse load.

Table 3
Input random variables for Example 3.

Variable	Distribution	Mean	Std Dev
m	Normal	1.0	0.05
k_1	Normal	1.0	0.10
k_2	Normal	0.2	0.01
r	Normal	0.5	0.05
F_1	Normal	1.0	0.20
t_1	Normal	1.0	0.20

Table 4
Comparison of the number of g function calls for Example 3.

Method	N_{call}	
	Mean	CoV
Proposed method	10 + 14.75 = 24.75	8.68%
AL-GP	12 + 21.60 = 33.60	7.87%

To provide the reference results for the response CDF, CCDF and PDF, MCS with 10^7 samples is conducted. For the AL-GP method, the range of interest and the stopping criterion are set to $[-0.65, 1.65]$ and $\bar{\epsilon} = 0.10$ respectively. Figs. 9(a) and 9(b) demonstrate that both the AL-GP method and the proposed method are capable of generating CDF/CCDF mean curves that are in close alignment with the reference curves, as well as mean ± std dev bounds that are notably narrow. The upper bounds of the posterior CoV functions for the response CDF and CCDF are presented statistically in Figs. 9(c) and 9(d) respectively, as a result of the proposed method. Fig. 10 shows the statistical results of the response PDF, produced by the proposed method as a by-product. It can be observed that the PDF mean curve aligns closely with the histogram produced by MCS, and the mean ± std dev band is relatively narrow. Table 4 compares the required number of g-function calls. The proposed method, on average, necessitates a reduced number of g-function evaluations, while exhibiting a marginally greater CoV than the AL-GP method.

5.4. Example 4: A space truss structure

The fourth numerical example consists of a 52-bar space truss structure, as shown in Fig. 11. This structure is modeled as a three-dimensional finite element model using the open source software OpenSees. The model consists of 21 nodes and 52 truss elements. All elements have the same cross-sectional area A and Young's modulus E. Concentrated vertical loads along the negative z-axis, denoted P_0 - P_{12}, are applied to nodes 0–12. Of interest is the vertical displacement of node 0:

$$Y = g(A, E, P_0, P_1, \dots, P_{12}), \tag{48}$$

where A, E, P_0, P_1, $\cdots$, P_{12} are treated as 15 independent random variables, as described in Table 5.

The reference solutions for the response CDF, CCDF and PDF are provided by MCS with 10^6 samples. Figs. 12(a) and 12(b) shows that the proposed method can produce an almost unbiased CDF/CCDF mean and a narrow mean ± std dev band. In addition, the proposed method can also provide the local error measures, i.e. the upper bounds on the posterior CoV of the response CDF and CCDF, where the mean curves and mean ± std dev bands are plotted in Figs. 12(c) and 12(d), respectively. The response PDF can be obtained as a by-product for the proposed method, the statistical results of which are shown in Fig. 13. Again, the results are very favorable. Note that the proposed method only requires an average of 10 + 29.50 = 39.50 g-function evaluations (with a CoV of 9.16%).

Table 5
Input random variables for Example 4.

Variable	Distribution	Mean	CoV
A	Normal	2×10^3 mm^2	0.10
E	Normal	2.06×10^5 MPa	0.10
P_0	Log-normal	2×10^2 kN	0.20
$P_1, P_2, \dots, P_{12}$	Log-normal	1×10^2 kN	0.15

Table 6
Basic random variables for Example 5.

Variable	Distribution	Parameter 1	Parameter 2
h_D (m)	Uniform	7	10
$k_{xx,1}$ (10^{-7} m/s)	Log-normal	5	0.20
$k_{yy,1}$ (10^{-7} m/s)	Log-normal	2	0.20
$k_{xx,2}$ (10^{-6} m/s)	Log-normal	5	0.20
$k_{yy,2}$ (10^{-6} m/s)	Log-normal	2	0.20

Note: For a uniform distribution, parameter 1 and parameter 2 are the lower and upper bounds, respectively; otherwise, parameter 1 and parameter 2 are the mean and CoV, respectively.

5.5. Example 5: A seepage problem

The last numerical example investigates the steady-state confined seepage under an impervious dam (adopted from [45]), as shown in Fig. 14. The dam rests on an impermeable rock layer, above which are two permeable soil layers: a 15 m layer of silty sand and a 5 m layer of silty gravel. The horizontal and vertical permeabilities of the ith layer are given as $k_{xx,i}$ and $k_{yy,i}$, respectively. A water column of height h_D m is retained at the upstream of the dam. The hydraulic head h_W over the impermeable layer is $h_W = h_D + 20$ m. At the bottom of the dam, a cut-off wall is installed to prevent excessive seepage. It is assumed that water flows only from segment AB to segment CD through the two permeable layers, with no flow occurring along any other boundary of the system. We consider h_D, $k_{xx,1}$, $k_{yy,1}$, $k_{xx,2}$ and $k_{yy,2}$ as five independent random variables, as listed in Table 6. The hydraulic head h_W is governed by the following partial differential equation:

$$k_{xx,i}\frac{\partial^2 h_W}{\partial x^2} + k_{yy,i}\frac{\partial^2 h_W}{\partial y^2} = 0, i = 1, 2, \tag{49}$$

where x and y are the horizontal and vertical coordinates, respectively. The boundary conditions for this equation include the hydraulic head across segments AB and CD, with zero flow across the remaining boundaries, as previously described. Eq. (49) is solved using the finite element method with 3413 nodes and 1628 quadratic triangular elements, as shown in Fig. 14. For illustrative purposes, the hydraulic head solved by fixing the random variables at their mean values is shown in Fig. 15. Once the hydraulic head is solved, the quantity of interest, i.e. the seepage flow q on the downstream side of the dam, can be calculated:

$$q = -\int_{\text{CD}} k_{yy,2}\frac{\partial h_W}{\partial y}dx, \tag{50}$$

which is measured in units of volume over time over distance. The flow q is a function of the five basic random variables, so it is a random variable as well.

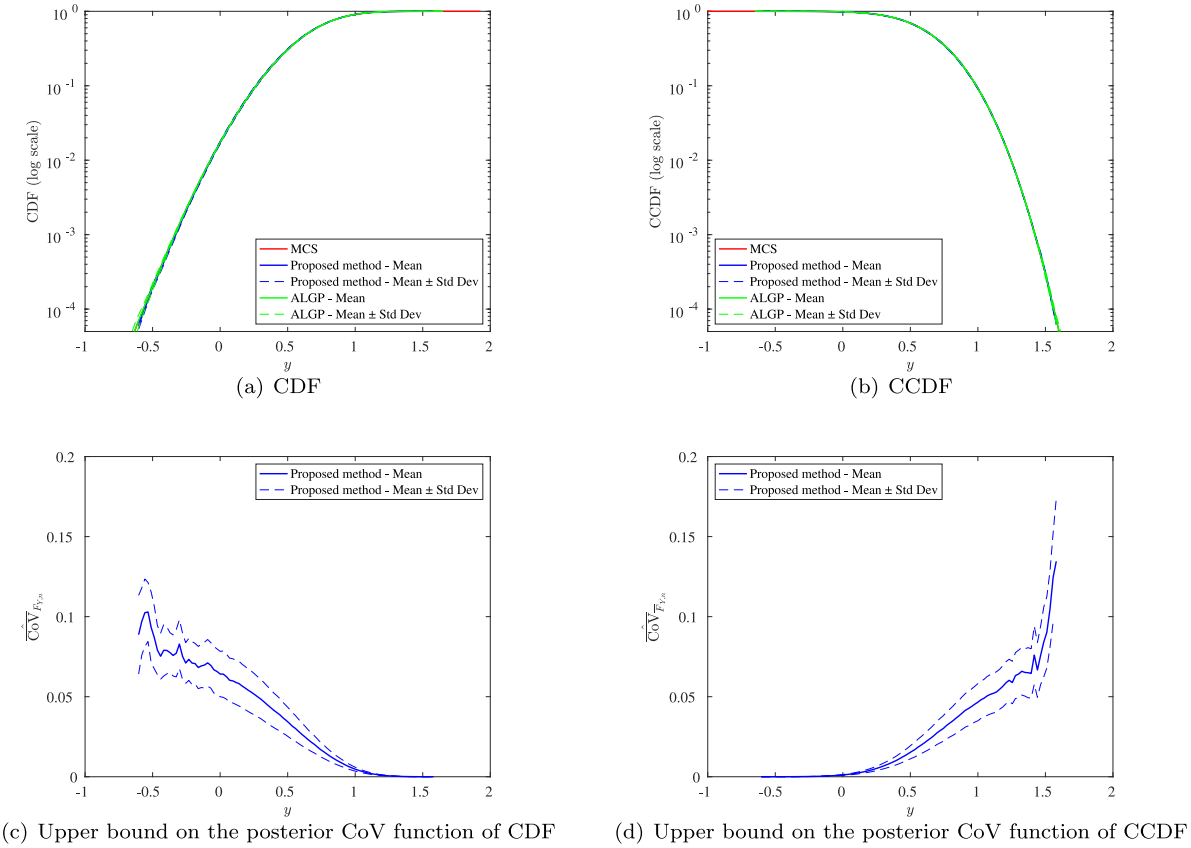


Fig. 9. Response CDF and CCDF for Example 3.

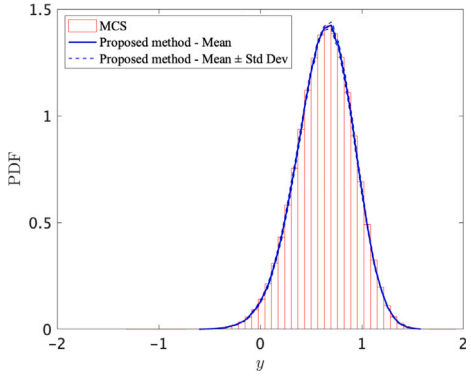


Fig. 10. Response PDF for Example 3.

The reference results for the response CDF, CCDF, and PDF of q are generated by MCS with 10^6 runs. The CDF and CCDF related results from the proposed BAL method are shown in Fig. 16. It can be seen from Figs. 16(a) and 16(b) that the proposed method produces CDF and CCDF mean curves that are close to the corresponding reference solutions, with very narrow mean ± std dev bands. Our method can also provide the upper bounds on the posterior CoV of the CDF and CCDF, whose statistical results are depicted in Figs. 16(c) and 16(d). In addition, the response PDF can be obtained as a by-product, as shown in Fig. 17, with the mean curve close to the reference result and a narrow mean ± std dev band. Remarkably, the proposed method only needs on average 10 + 24.65 = 34.65 model evaluations (with a CoV of 8.44%).

6. Concluding remarks

This paper presents a Bayesian active learning perspective using Gaussian process (GP) regression on estimating the response probability distributions of expensive computer simulators in the presence of randomness. First, the estimation of response probability distributions is conceptually interpreted as a Bayesian inference problem, in contrast to frequentist inference. This conceptual Bayesian idea has several important advantages, for example, quantifying numerical error as a source of epistemic uncertainty, incorporating prior knowledge and enabling the reduction of numerical error in an active learning way. By virtue of the well-established GP regression, a practical Bayesian approach is then developed for estimating the response probability distributions. In this context, a GP prior is assigned over the computer simulator, conditioning this GP prior on several computer simulator evaluations gives rise to a GP posterior for the computer simulator. We derive the posterior statistics of the response cumulative distribution function (CDF), complementary CDF (CCDF) and probability density function (PDF) in semi-analytical form, and provide the numerical solution scheme. At last, a Bayesian active learning method is proposed for response probability distribution estimation, where two key components for active learning are devised by making use of the

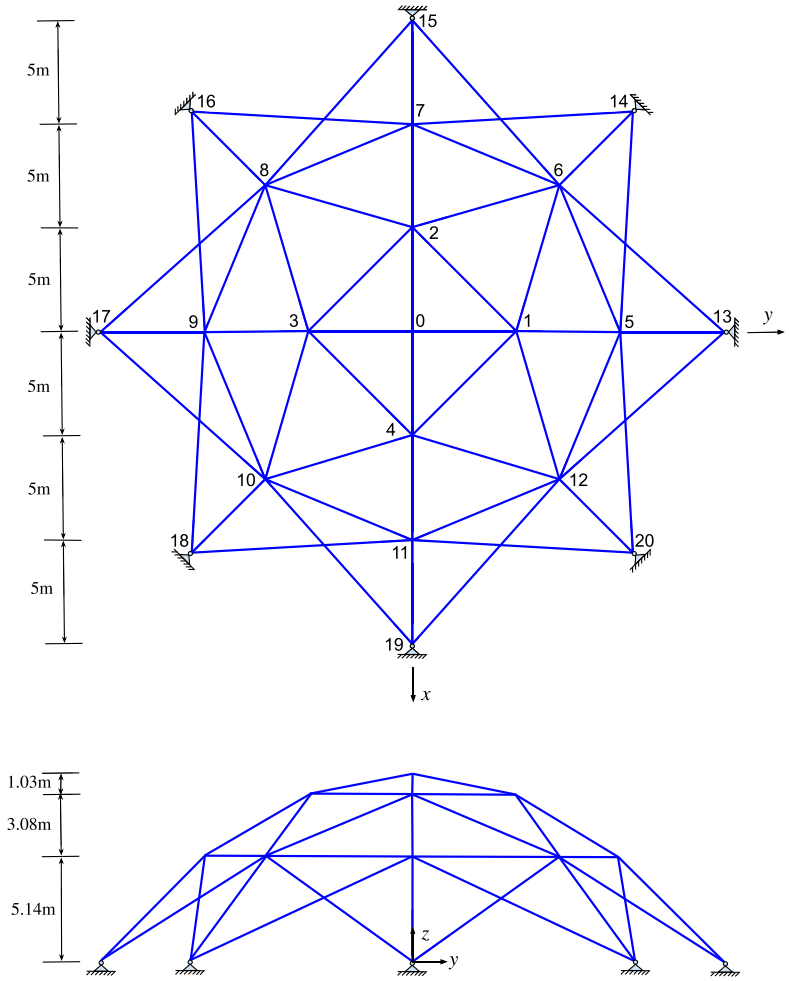


Fig. 11. A 52-bar space truss structure.

posterior statistics. Five numerical examples are studied to demonstrate the performance of the proposed Bayesian active learning method. The results show that our method can produce the response CDF and CCDF with quantified uncertainty using only a small number of calls to the computer simulator. Additionally, the method provides the response PDF as a by-product without numerically differentiating the response CDF.

The findings of this study can be used as a starting point for a number of future studies. In particular, two possible research directions are suggested here. One is to develop a Bayesian active learning scheme that can operate directly on the response PDF, which remains challenging. The other is to develop a strategy that can identify multiple informative points from the learning function, thus allowing parallel computation.

CRediT authorship contribution statement

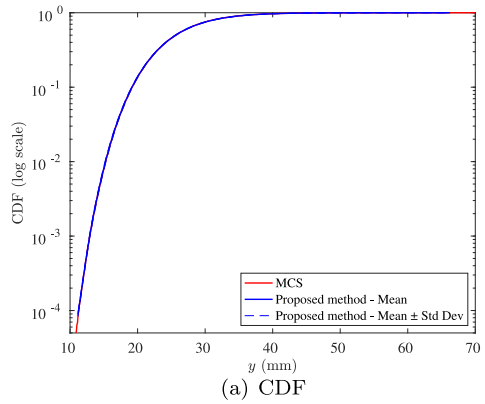
Chao Dang: Writing – review & editing, Writing – original draft, Visualization, Validation, Methodology, Investigation, Formal analysis, Conceptualization. **Marcos A. Valdebenito:** Writing – review & editing, Validation, Supervision, Resources, Investigation. **Nataly A. Manque:** Writing – review & editing, Visualization, Software, Resources. **Jun Xu:** Writing – review & editing, Project administration, Funding acquisition. **Matthias G.R. Faes:** Writing – review & editing, Supervision, Resources, Project administration, Funding acquisition.

Declaration of competing interest

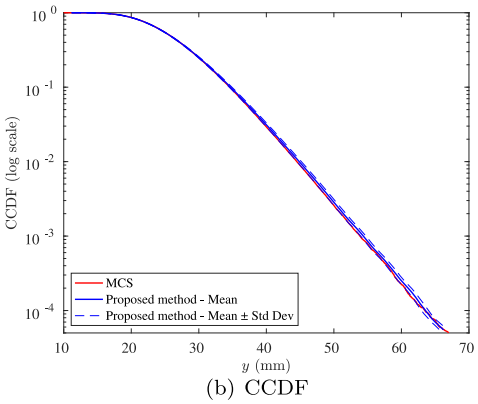
The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

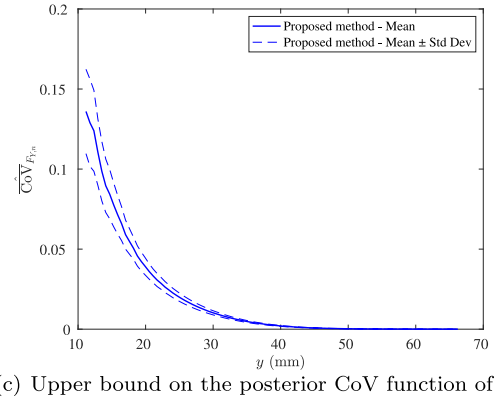
Chao Dang is grateful for the financial support of the German Research Foundation (DFG), Germany (Grant number 530326817). Jun Xu would like to appreciate the financial support of National Natural Science Foundation of China (No. 52278178).



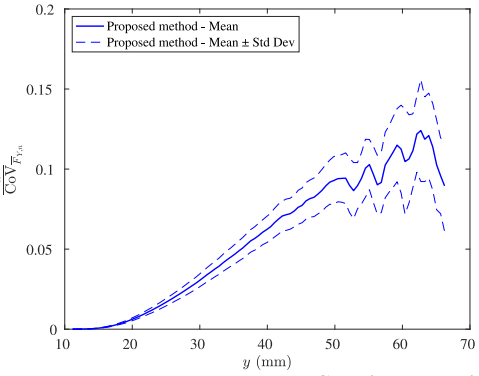
(a) CDF



(b) CCDF



(c) Upper bound on the posterior CoV function of CDF



(d) Upper bound on the posterior CoV function of CCDF

Fig. 12. Response CDF and CCDF for Example 4.

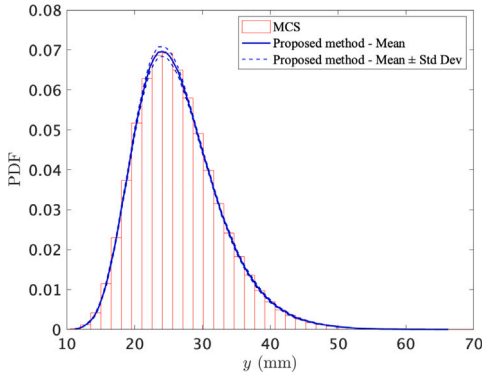


Fig. 13. Response PDF for Example 4.

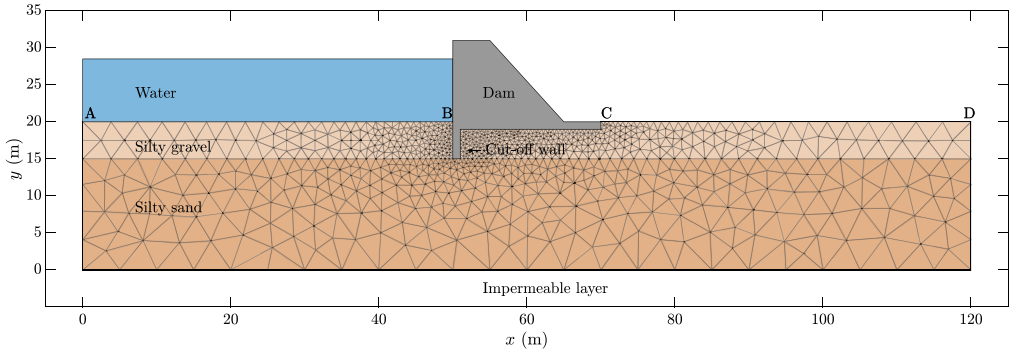


Fig. 14. Schematic representation of the confined seepage under an impermeable dam.

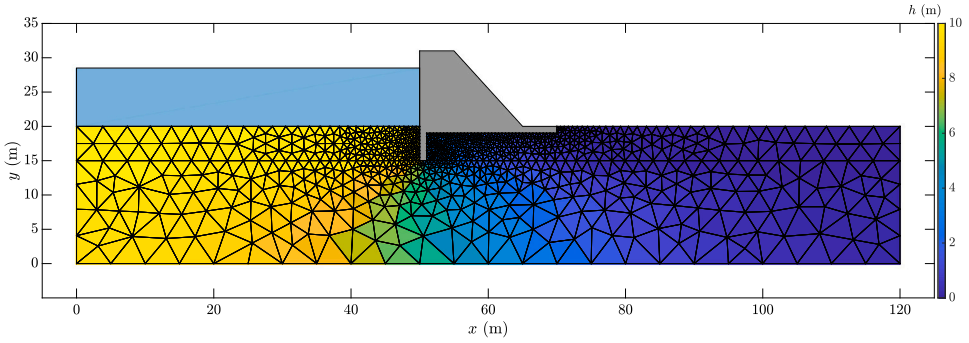


Fig. 15. Hydraulic head solved by fixing the random variables at their mean values.

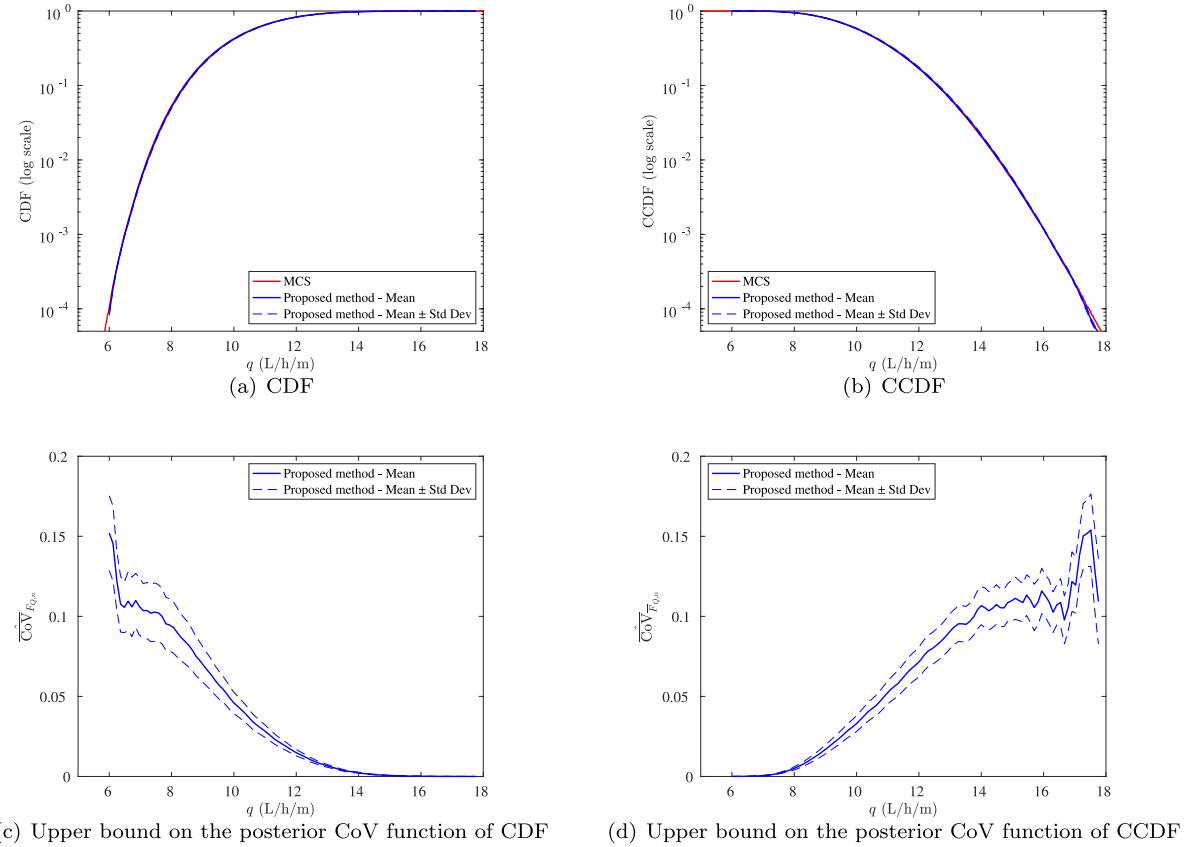


Fig. 16. Response CDF and CCDF for Example 5.

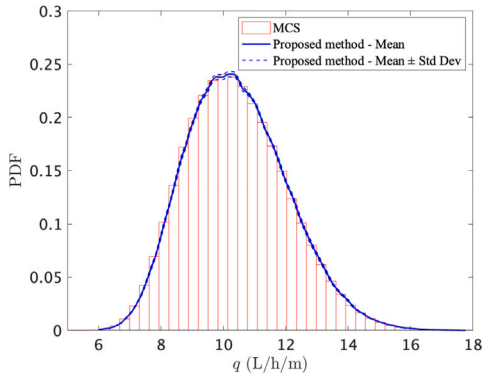


Fig. 17. Response PDF for Example 5.

Data availability

No data was used for the research described in the article.

References

[1] Kroese DP, Taimre T, Botev ZI. Handbook of Monte Carlo methods. John Wiley & Sons; 2013.

[2] Helton JC, Davis FJ. Latin hypercube sampling and the propagation of uncertainty in analyses of complex systems. Reliab Eng Syst Saf 2003;81(1):23–69. http://dx.doi.org/10.1016/S0951-8320(03)00058-9.

[3] Shields MD, Teferra K, Hapij A, Daddazio RP. Refined stratified sampling for efficient Monte Carlo based uncertainty quantification. Reliab Eng Syst Saf 2015;142:310–25. http://dx.doi.org/10.1016/j.ress.2015.05.023.

[4] Shields MD, Zhang J. The generalization of Latin hypercube sampling. Reliab Eng Syst Saf 2016;148:96–108. http://dx.doi.org/10.1016/j.ress.2015.12.002.

[5] Lemieux C. Monte Carlo and Quasi-Monte Carlo sampling. Springer; 2009.

[6] Zhao Y-G, Lu Z-H. Structural reliability: approaches from perspectives of statistical moments. John Wiley & Sons; 2021.

[7] Lee SH, Chen W. A comparative study of uncertainty propagation methods for black-box-type problems. Struct Multidiscip Optim 2009;37:239–53. http://dx.doi.org/10.1007/s00158-008-0234-7.

[8] Xu J, Dang C. A new bivariate dimension reduction method for efficient structural reliability analysis. Mech Syst Signal Process 2019;115:281–300. http://dx.doi.org/10.1016/j.ymssp.2018.05.046.

[9] He W, Li G, Hao P, Zeng Y. Maximum entropy method-based reliability analysis with correlated input variables via hybrid dimension-reduction method. J Mech Des 2019;141(10):101405. http://dx.doi.org/10.1115/1.4043734.

[10] Zhou T, Peng Y. Adaptive Bayesian quadrature based statistical moments estimation for structural reliability analysis. Reliab Eng Syst Saf 2020;198:106902. http://dx.doi.org/10.1016/j.ress.2020.106902.

[11] Zhang X, Pandey MD. Structural reliability analysis based on the concepts of entropy, fractional moment and dimensional reduction method. Struct Saf 2013;43:28–40. http://dx.doi.org/10.1016/j.strusafe.2013.03.001.

[12] Xu J, Kong F. Adaptive scaled unscented transformation for highly efficient structural reliability analysis by maximum entropy method. Struct Saf 2019;76:123–34. http://dx.doi.org/10.1016/j.strusafe.2018.09.001.

[13] Dang C, Xu J. A mixture distribution with fractional moments for efficient seismic reliability analysis of nonlinear structures. Eng Struct 2020;208:109912. http://dx.doi.org/10.1016/j.engstruct.2019.109912.

[14] Ding C, Dang C, Valdebenito MA, Faes MG, Broggi M, Beer M. First-passage probability estimation of high-dimensional nonlinear stochastic dynamic systems by a fractional moments-based mixture distribution approach. Mech Syst Signal Process 2023;185:109775. http://dx.doi.org/10.1016/j.ymssp.2022.109775.

[15] Xu J, Yu Q. Harmonic transform-based non-parametric density estimation method for forward uncertainty propagation and reliability analysis. Struct Saf 2023;103:102331. http://dx.doi.org/10.1016/j.strusafe.2023.102331.

[16] Yu Q, Xu J. Harmonic transform-based density estimation method for uncertainty propagation and reliability analysis involving multi-modal distributions. Mech Syst Signal Process 2023;190:110113. http://dx.doi.org/10.1016/j.ymssp.2023.110113.

[17] Xu J, Song J, Yu Q, Kong F. Generalized distribution reconstruction based on the inversion of characteristic function curve for structural reliability analysis. Reliab Eng Syst Saf 2023;229:108768. http://dx.doi.org/10.1016/j.ress.2022.108768.

[18] Wan Z, Gao Y, Tao W. Structural reliability analysis using generalized distribution reconstruction method with novel improvements. ASCE- ASME J Risk Uncertain Eng Syst Part A: Civ Eng 2024;10(2):04024028. http://dx.doi.org/10.1061/AJRUA6.RUENG-1298.

[19] Zhang F, Xu J, Wan Z. An adaptive Gaussian Mixture Model for structural reliability analysis using convolution search technique. Struct Saf 2024;102548. http://dx.doi.org/10.1016/j.strusafe.2024.102548.

[20] Blatman G, Sudret B. An adaptive algorithm to build up sparse polynomial chaos expansions for stochastic finite element analysis. Probab Eng Mech 2010;25(2):183–97. http://dx.doi.org/10.1016/j.probengmech.2009.10.003.

[21] Zeng X, Ghanem R. Projection pursuit adaptation on polynomial chaos expansions. Comput Methods Appl Mech Engrg 2023;405:115845. http://dx.doi.org/10.1016/j.cma.2022.115845.

[22] Weng Y-Y, Liu T, Zhang X-Y, Zhao Y-G. Probability density estimation of polynomial chaos and its application in structural reliability analysis. Reliab Eng Syst Saf 2025;253:110537. http://dx.doi.org/10.1016/j.ress.2024.110537.

[23] Oakley J, O'hagan A. Bayesian inference for the uncertainty distribution of computer model outputs. Biometrika 2002;89(4):769–84. http://dx.doi.org/10.1093/biomet/89.4.769.

[24] Wang Z, Broccardo M. A novel active learning-based Gaussian process metamodelling strategy for estimating the full probability distribution in forward UQ analysis. Struct Saf 2020;84:101937. http://dx.doi.org/10.1016/j.strusafe.2020.101937.

[25] Song J, Zhang Y, Cui Y, Yue T, Dang Y. Bayesian active learning approach for estimation of empirical copula-based moment-independent sensitivity indices. Eng Comput 2024;40(2):1247–63. http://dx.doi.org/10.1007/s00366-023-01865-0.

[26] Xiang Y, Han T, Li Y, Shi L, Pan B. A multi-region active learning Kriging method for response distribution construction of highly nonlinear problems. Comput Methods Appl Mech Engrg 2024;419:116650. http://dx.doi.org/10.1016/j.cma.2023.116650.

[27] Su M, Wang Z, Bursi OS, Broccardo M. Surrogate modeling for probability distribution estimation: uniform or adaptive design? 2024, arXiv preprint arXiv:2404.07323.

[28] Kougioumtzoglou IA, Psaros AF, Spanos PD. Path integrals in stochastic engineering dynamics. Springer Cham.

[29] Lyu M-Z, Chen J-B. A unified formalism of the GE-GDEE for generic continuous responses and first-passage reliability analysis of multi-dimensional nonlinear systems subjected to non-white-noise excitations. Struct Saf 2022;98:102233. http://dx.doi.org/10.1016/j.strusafe.2022.102233.

[30] Li J, Chen J. Probability density evolution method for dynamic response analysis of structures with uncertain parameters. Comput Mech 2004;34(5):400–9. http://dx.doi.org/10.1007/s00466-004-0583-8.

[31] Li J, Chen J. Stochastic Dynamics of Structures. John Wiley & Sons; 2009.

[32] Chen G, Yang D. Direct probability integral method for stochastic response analysis of static and dynamic structural systems. Comput Methods Appl Mech Engrg 2019;357:112612. http://dx.doi.org/10.1016/j.cma.2019.112612.

[33] Tao T, Zhao G, Yu Y, Huang B, Zheng H. A fully adaptive method for structural stochastic response analysis based on direct probability integral method. Comput Methods Appl Mech Engrg 2022;396:115066. http://dx.doi.org/10.1016/j.cma.2022.115066.

[34] Dang C, Wei P, Song J, Beer M. Estimation of failure probability function under imprecise probabilities by active learning–augmented probabilistic integration. ASCE- ASME J Risk Uncertain Eng Syst Part A: Civ Eng 2021;7(4):04021054. http://dx.doi.org/10.1061/AJRUA6.0001179.

[35] Dang C, Wei P, Faes MG, Valdebenito MA, Beer M. Parallel adaptive Bayesian quadrature for rare event estimation. Reliab Eng Syst Saf 2022;225:108621. http://dx.doi.org/10.1016/j.ress.2022.108621.

[36] Dang C, Valdebenito MA, Faes MG, Wei P, Beer M. Structural reliability analysis: A Bayesian perspective. Struct Saf 2022;99:102259. http://dx.doi.org/10.1016/j.strusafe.2022.102259.

[37] Hennig P, Osborne MA, Kersting HP. Probabilistic numerics: Computation as machine learning. Cambridge University Press; 2022.

[38] Jones DR, Schonlau M, Welch WJ. Efficient global optimization of expensive black-box functions. J Global Optim 1998;13:455–92. http://dx.doi.org/10.1023/A:1008306431147.

[39] Garnett R. Bayesian optimization. Cambridge University Press; 2023.

[40] O'Hagan A. Bayes–Hermite quadrature. J Statist Plann Inference 1991;29(3):245–60. http://dx.doi.org/10.1016/0378-3758(91)90002-V.

[41] Rasmussen CE, Ghahramani Z. Bayesian Monte Carlo. Adv Neural Inf Process Syst 2003;505–12.

[42] Briol F-X, Oates CJ, Girolami M, Osborne MA, Sejdinovic D. Probabilistic integration: A role in statistical computation? Statist Sci 2019;34(1):1–22. http://dx.doi.org/10.1214/18-STS660.

[43] Williams CK, Rasmussen CE. Gaussian processes for machine learning, vol. 2, (no. 3). MA: MIT press Cambridge; 2006.

[44] Bucher CG, Bourgund U. A fast and efficient response surface approach for structural reliability problems. Struct Saf 1990;7(1):57–66. http://dx.doi.org/10.1016/0167-4730(90)90012-E.

[45] Valdebenito MA, Jensen HA, Hernández H, Mehrez L. Sensitivity estimation of failure probability applying line sampling. Reliab Eng Syst Saf 2018;171:99–111. http://dx.doi.org/10.1016/j.ress.2017.11.010.