Structural reliability analysis with extremely small failure probabilities: A quasi-Bayesian active learning method

Chao Dang[a,*], Alice Cicirello[b], Marcos A. Valdebenito[c], Matthias G.R. Faes[c], Pengfei Wei[d], Michael Beer[a,e,f]

[a]*Institute for Risk and Reliability, Leibniz University Hannover, Callinstr. 34, Hannover 30167, Germany*
[b]*Department of Engineering, University of Cambridge, Trumpington Street, Cambridge CB2 1PZ, United Kingdom*
[c]*Chair for Reliability Engineering, TU Dortmund University, Leonhard-Euler-Str. 5, Dortmund 44227, Germany*
[d]*School of Power and Energy, Northwestern Polytechnical University, Xi'an 710072, PR China*
[e]*Institute for Risk and Uncertainty, University of Liverpool, Peach Street, Liverpool L69 7ZF, United Kingdom*
[f]*International Joint Research Center for Resilient Infrastructure & International Joint Research Center for Engineering Reliability and Stochastic Mechanics, Tongji University, Shanghai 200092, PR China*

Abstract

The concept of Bayesian active learning has recently been introduced from machine learning to structural reliability analysis. Although several specific methods have been successfully developed, significant efforts are still needed to fully exploit their potential and to address existing challenges. This work proposes a quasi-Bayesian active learning method, called 'Quasi-Bayesian Active Learning Cubature', for structural reliability analysis with extremely small failure probabilities. The method is established based on a cleaver use of the Bayesian failure probability inference framework. To reduce the computational burden associated with the exact posterior variance of the failure probability, we propose a quasi posterior variance instead. Then, two critical elements for Bayesian active learning, namely the stopping criterion and the learning function, are developed subsequently. The stopping criterion is defined based on the quasi posterior coefficient of variation of the failure probability, whose numerical solution scheme is also tailored. The learning function is extracted from the quasi posterior variance, with the introduction of an additional parameter that allows multi-point selection and hence parallel distributed processing. By testing on four numerical examples, it is empirically shown that the proposed method can assess extremely small failure probabilities with desired accuracy and efficiency.

Keywords: Structural reliability analysis; Small failure probability; Bayesian active learning; Stopping criterion; Learning function; Parallel computing

1. Introduction

Structural reliability analysis aims at quantifying the likelihood that a structure will achieve certain undesired performance, taking into account uncertainties in material properties, geometric dimensions and applied loads, etc. If these uncertainties are modeled in a purely probabilistic context, an essential task is to calculate the so-called failure probability P_f, which is mathematically defined as a multi-dimensional integral:

$$P_f = \int_{\mathcal{X}} I(g(\boldsymbol{x})) f_{\boldsymbol{X}}(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}, \tag{1}$$

where $\boldsymbol{X} = [X_1, X_2, \cdots, X_d]^\top \in \mathcal{X} \subseteq \mathbb{R}^d$ is a vector of d random variables with known joint probability density function (PDF) $f_{\boldsymbol{X}}(\boldsymbol{x})$; $g(\cdot) : \mathbb{R}^d \to \mathbb{R}$ denotes the performance function (also known as the limit state function), which takes a negative value when a failure event occurs; $I(\cdot) : \mathbb{R} \to \{0, 1\}$ represents the failure indicator function: $I = 1$ if $g(\boldsymbol{x}) < 0$ and $I = 0$ otherwise. In many practical applications, such a task has the following common characteristics: (1) it is most unlikely that the failure probability can be solved analytically, despite the simplicity of its definition; (2) the failure probability of interest is very small, close to zero; (3) each evaluation of the g-function can be quite computationally demanding. The combination of these characteristics makes probabilistic structural reliability analysis very challenging from a numerical point of view.

To meet the computational challenge, a variety of numerical methods have been developed over the last few decades. They can be roughly classified into five main groups: (1) stochastic simulation methods (e.g., Monte Carlo simulation (MCS) and its variants [1]), (2) asymptotic approximation methods (e.g., first-/second- order reliability method [2]), (3) moment based methods (e.g., fourth-order moment method [3] fractional moment method [4]), (4) probability conservation based methods (e.g., probability density evolution method [5] and globally-evolving-based generalized density evolution equation method [6]) and (5) surrogate-assisted methods (e.g., response surface method [7], polynomial chaos expansion method [8] and Kriging-based method [9]). It should be noted that these classifications are not strictly mutually exclusive

*Corresponding author
Email address: chao.dang@irz.uni-hannover.de (Chao Dang)

and may overlap and intersect. Among the existing developments, surrogate-assisted methods have received increasing attention in the structural reliability analysis community, especially those that are empowered with an active learning paradigm. The credit for introducing active learning from the field of machine learning to the field of structural reliability analysis is generally attributed to Bichon et al. [10] and Echard et al. [11], who developed the well-known efficient global reliability method and active learning Kriging Monte Carlo simulation (AK-MCS) method respectively. Since then, a large number of active learning reliability methods have been proposed by researchers and engineers from various fields. The interested reader is referred to [12, 13] for the recent advances of active learning methods for structural reliability analysis.

Another active learning paradigm, called Bayesian active learning (as a type of active learning that particularly emphasizes the use of Bayesian principles), has also been recently introduced from machine learning to structural reliability analysis. The first work was reported in [14], where: (1) the problem of failure probability estimation is first interpreted as a Bayesian inference problem; (2) the posterior mean and an upper bound on the posterior variance of the failure probability are derived, given that a Gaussian process prior is placed over the performance function; (3) a numerical method, called 'Active Learning Probabilistic Integration' (ALPI), is developed for failure probability estimation, with a stopping criterion and a learning function being directly derived from the known posterior statistics of the failure probability. The ALPI method was further enhanced by the 'Parallel Adaptive Bayesian Quadrature' (PABQ) method [15] to facilitate parallel distributed processing and assessing small failure probabilities. A principled 'Bayesian failure probability inference' (BFPI) framework was then developed in [16], where the exact posterior variance of the failure probability is obtained. Although the BPFI provides a complete Bayesian treatment of the failure probability integral in terms of second-order posterior statistics, it is still challenging to perform Bayesian active learning of the failure probability using its known posterior statistics, largely due to the computational burden associated with the exact posterior variance.

To overcome this obstacle, several efforts have been made to develop Bayesian active learning reliability analysis methods without using the posterior variance of the failure probability. In the work [17], the authors

introduced three partially Bayesian active learning methods under the name of 'Partially Bayesian active learning cubature'. These methods use only the posterior mean of the failure probability to design the two critical components for Bayesian active learning, namely the stopping criterion and the learning function. In a similar spirit, a method called 'semi-Bayesian active learning quadrature' (SBALQ) was developed in [18], which allows multi-point selection and thus parallel distributed processing. In addition, another method called 'Parallel Bayesian Probabilistic Integration' (PBPI) [19] was also proposed, based on the development of a pseudo posterior variance for the failure probability. As a side remark, the Bayesian active learning idea has also been scusesfully perused in the context of line sampling for structural reliability analysis, see for example [20–22]. Although many efforts have been made to advance the development of Bayesian active learning reliability methods, there is still much room for progress to fully exploit their potential and effectively address existing challenges.

The objective of this work is to present another Bayesian active learning method, called 'Quasi-Bayesian Active Learning Cubature' (QBALC), for structural reliability analysis based on the BFPI framework. This method is expected to be capable of evaluating extremely small failure probabilities, which is one of the main challenges in the realm of structural reliability analysis. The main contributions can be summarized as follows. First, we develop a quasi posterior variance for the failure probability by simplifying the exact one. It may therefore be more conservative than the upper bound given in [14, 15], less computationally expensive than the exact posterior variance given in [16], and less empirical than the pseudo posterior variance [19]. Second, a stopping criterion is proposed, which is based on the quasi posterior coefficient of variation (COV) of the failure probability, in contrast to existing stopping criteria [14, 15, 17, 19]. Third, a numerical integration technique is introduced to approximate the two analytical intractable integrals involved in the stopping criterion, similar to [17, 19]. Fourth, a learning function derived from the quasi posterior variance is proposed, which itself allows for multi-point selection, and hence parallel computing. The multi-point section strategy is significantly different our previous studies [15, 18, 19].

The rest of this paper is structured as follows. Section 2 briefly reviews the BFPI framework. The proposed QBALC method is presented in Section 3. Four numerical examples are studied in Section 4 to

validate the proposed method. Section 5 concludes the present study.

2. Bayesian failure probability inference

In this section, we give a general overview of the BFPI framework originally developed in [16]. It should be noted that the framework in [16] is set up in the physical space (i.e., $\mathcal{X}$). Here it is presented in standard normal space (i.e., $\mathcal{U}$) instead. To do so, we first introduce a transformation T that can transform the physical random variables into standard normal variables, i.e., $\boldsymbol{U} = T(\boldsymbol{X})$, where $\boldsymbol{U} = [U_1, U_2, \cdots, U_d]^\top \in \mathcal{U} \subseteq \mathbb{R}^d$ represents a vector of d standard normal variables. This can be achieved by using some widely-used transformations, such as Rosenblatt transformation and Nataf transformation. A transformed performance function can be defined such that $\mathcal{G}(\boldsymbol{U}) = g(T^{-1}(\boldsymbol{U}))$, where T^{-1} denotes the inverse transformation. The failure indicator function corresponding to the transformed performance function $\mathcal{G}$ is denoted as $\mathcal{I}$, which is equal to 1 if $\mathcal{G}(\boldsymbol{u}) < 0$ and 0 otherwise. The failure probability can be rewritten as $\mathcal{P}_f = \int_{\mathcal{U}} \mathcal{I}(\mathcal{G}(\boldsymbol{u}))\phi_{\boldsymbol{U}}(\boldsymbol{u})\mathrm{d}\boldsymbol{u}$, where $\phi_{\boldsymbol{U}}(\boldsymbol{u})$ denotes the joint PDF of $\boldsymbol{U}$. For a schematic diagram of the BFPI framework in standard normal space, see Fig. 1.

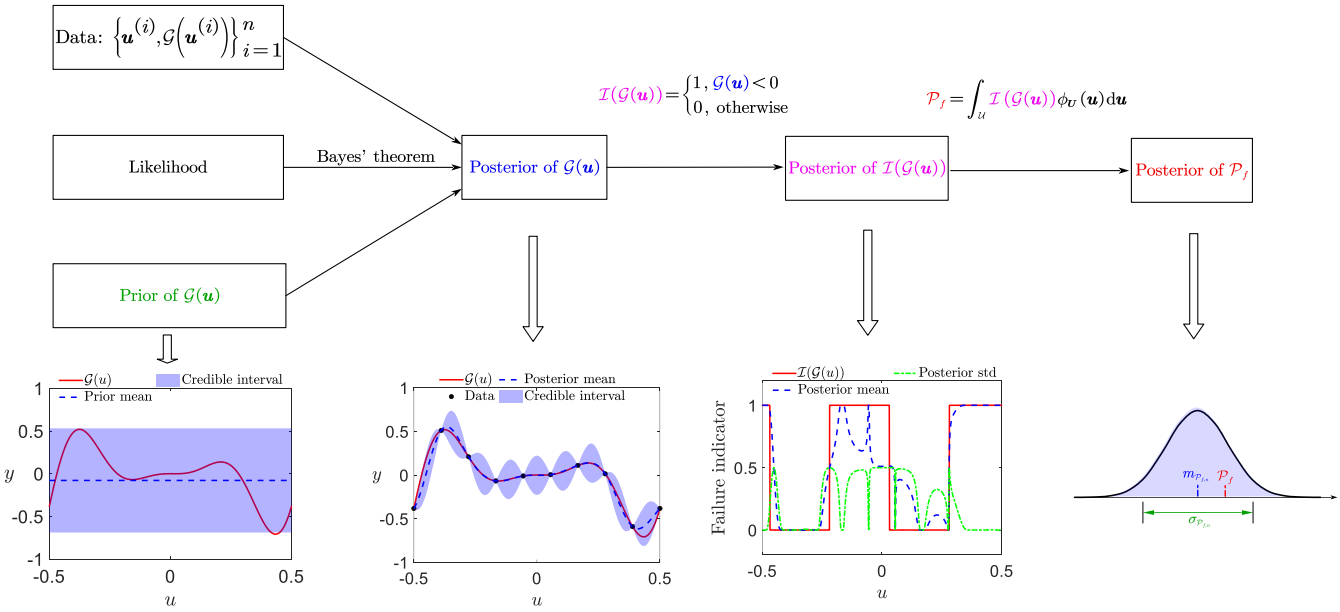


Figure 1: Schematic diagram of the BFPI framework in standard normal space.

2.1. Prior distribution

The BFPI framework begins by placing a Gaussian process prior over the transformed performance function $\mathcal{G}(\boldsymbol{u})$ such that:

$$\mathcal{G}_0(\boldsymbol{u}) \sim \mathcal{GP}(m_{\mathcal{G}_0}(\boldsymbol{u}), k_{\mathcal{G}_0}(\boldsymbol{u}, \boldsymbol{u}')), \tag{2}$$

where $\mathcal{G}_0$ denotes the prior distribution of $\mathcal{G}$; $m_{\mathcal{G}_0}(\boldsymbol{u})$ and $k_{\mathcal{G}_0}(\boldsymbol{u}, \boldsymbol{u}')$ are the prior mean and covariance functions of the GP respectively. It is further assumed that the prior mean function takes a constant value and the prior covariance function takes a squared exponential kernel respectively:

$$m_{\mathcal{G}_0}(\boldsymbol{u}) = \beta, \tag{3}$$

$$k_{\mathcal{G}_0}(\boldsymbol{u}, \boldsymbol{u}') = \sigma_0^2 \exp\left(-\frac{1}{2}(\boldsymbol{u} - \boldsymbol{u}')^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{u} - \boldsymbol{u}')\right), \tag{4}$$

where $\beta \in \mathbb{R}$; $\sigma_0 > 0$ denotes the process standard deviation; $\boldsymbol{\Sigma} = \mathrm{diag}\left(l_1^2, l_2^2, \cdots, l_d^2\right)$ with $l_i > 0$ being the length scale in the i-th dimension. The prior mean and covariance functions are parameterized by $d + 2$ hyperparameters, denoted by $\boldsymbol{\vartheta} = [\beta, \sigma_0, l_1, l_2, \cdots, l_d]^\top$. Note that in most cases these hyperparameters cannot be known a priori.

2.2. Tuning hyperparameters

Suppose that we have a dataset $\boldsymbol{\mathcal{D}} = \{\boldsymbol{\mathcal{U}}, \boldsymbol{\mathcal{Y}}\}$, where $\boldsymbol{\mathcal{U}} = \left[\boldsymbol{u}^{(1)}, \boldsymbol{u}^{(2)}, \cdots, \boldsymbol{u}^{(n)}\right]^\top$ is an $n \times d$ matrix comprising n observation locations and $\boldsymbol{\mathcal{Y}} = \left[y^{(1)}, y^{(2)}, \cdots, y^{(n)}\right]^\top$ is an $n \times 1$ vector with $y^{(j)} = \mathcal{G}(\boldsymbol{u}^{(j)})$. Then, the hyperparameters $\boldsymbol{\vartheta}$ can be learned from the dataset $\boldsymbol{\mathcal{D}}$ by maximizing the log-marginal likelihood:

$$\log p(\boldsymbol{\mathcal{Y}}|\boldsymbol{\mathcal{U}}, \boldsymbol{\vartheta}) = -\frac{1}{2}\left[(\boldsymbol{\mathcal{Y}} - \beta)^\top \boldsymbol{K}_{\mathcal{G}_0}^{-1}(\boldsymbol{\mathcal{Y}} - \beta) + \log|\boldsymbol{K}_{\mathcal{G}_0}| + n\log 2\pi\right], \tag{5}$$

where $\boldsymbol{K}_{\mathcal{G}_0}$ denotes an $n \times n$ covariance matrix with its (i, j)-th entry being $k_{\mathcal{G}_0}(\boldsymbol{u}^{(i)}, \boldsymbol{u}^{(j)})$.

2.3. Posterior statistics

The posterior distribution of $\mathcal{G}$ conditional on the data $\boldsymbol{\mathcal{D}}$ also proves to be a GP:

$$\mathcal{G}_n(\boldsymbol{u}) \sim \mathcal{GP}(m_{\mathcal{G}_n}(\boldsymbol{u}), k_{\mathcal{G}_n}(\boldsymbol{u}, \boldsymbol{u}')), \tag{6}$$

where $\mathcal{G}_n$ stands for the posterior distribution of $\mathcal{G}$; $m_{\mathcal{G}_n}(\boldsymbol{u})$ and $k_{\mathcal{G}_n}(\boldsymbol{u},\boldsymbol{u}')$ are the posterior mean and covariance functions of $\mathcal{G}$ respectively, which have the following analytical expressions:

$$m_{\mathcal{G}_n}(\boldsymbol{u}) = m_{\mathcal{G}_0}(\boldsymbol{u}) + \boldsymbol{k}_{\mathcal{G}_0}(\boldsymbol{u},\boldsymbol{\mathcal{U}})^\top \boldsymbol{K}_{\mathcal{G}_0}^{-1}\left(\boldsymbol{\mathcal{Y}} - \boldsymbol{m}_{\mathcal{G}_0}(\boldsymbol{\mathcal{U}})\right), \tag{7}$$

$$k_{\mathcal{G}_n}(\boldsymbol{u},\boldsymbol{u}') = k_{\mathcal{G}_0}(\boldsymbol{u},\boldsymbol{u}') - \boldsymbol{k}_{\mathcal{G}_0}(\boldsymbol{u},\boldsymbol{\mathcal{U}})^\top \boldsymbol{K}_{\mathcal{G}_0}^{-1}\boldsymbol{k}_{\mathcal{G}_0}(\boldsymbol{\mathcal{U}},\boldsymbol{u}'), \tag{8}$$

where $\boldsymbol{m}_{\mathcal{G}_0}(\boldsymbol{\mathcal{U}})$ is an $n \times 1$ mean vector whose j-th element is $m_{\mathcal{G}_0}(\boldsymbol{u}^{(j)})$; $\boldsymbol{k}_{\mathcal{G}_0}(\boldsymbol{u},\boldsymbol{\mathcal{U}})$ is an $n \times 1$ covariance vector whose j-th element is $k_{\mathcal{G}_0}(\boldsymbol{u},\boldsymbol{u}^{(j)})$; $\boldsymbol{k}_{\mathcal{G}_0}(\boldsymbol{\mathcal{U}},\boldsymbol{u}')$ is an $n \times 1$ covariance vector whose j-th element is $k_{\mathcal{G}_0}(\boldsymbol{u}^{(j)},\boldsymbol{u}')$.

The posterior distribution of the indicator function $\mathcal{I}$ conditional on the data $\boldsymbol{\mathcal{D}}$ follows a generalized Bernoulli process (GBP):

$$\mathcal{I}_n(\boldsymbol{u}) \sim \mathcal{GBP}(m_{\mathcal{I}_n}(\boldsymbol{u}), k_{\mathcal{I}_n}(\boldsymbol{u},\boldsymbol{u}')), \tag{9}$$

where $\mathcal{I}_n$ denotes the posterior distribution of $\mathcal{I}$; $m_{\mathcal{I}_n}(\boldsymbol{u})$ and $k_{\mathcal{I}_n}(\boldsymbol{u},\boldsymbol{u}')$ are the posterior mean and covariance functions of $\mathcal{I}$ receptively, which can be expressed as:

$$m_{\mathcal{I}_n}(\boldsymbol{u}) = \Phi\left(-\frac{m_{\mathcal{G}_n}(\boldsymbol{u})}{\sigma_{\mathcal{G}_n}(\boldsymbol{u})}\right), \tag{10}$$

$$k_{\mathcal{I}_n}(\boldsymbol{u},\boldsymbol{u}') = \Phi_2\left([0,0]^\top; \boldsymbol{m}_{\mathcal{G}_n}(\boldsymbol{u},\boldsymbol{u}'), \boldsymbol{K}_{\mathcal{G}_n}(\boldsymbol{u},\boldsymbol{u}')\right) - \Phi\left(\frac{-m_{\mathcal{G}_n}(\boldsymbol{u})}{\sigma_{\mathcal{G}_n}(\boldsymbol{u})}\right)\Phi\left(\frac{-m_{\mathcal{G}_n}(\boldsymbol{u}')}{\sigma_{\mathcal{G}_n}(\boldsymbol{u}')}\right), \tag{11}$$

where Φ denotes the cumulative distribution function (CDF) of the standard normal variable; $\sigma_{\mathcal{G}_n}(\boldsymbol{u})$ is the posterior standard deviation function of $\mathcal{G}$, i.e., $\sigma_{\mathcal{G}_n}(\boldsymbol{u}) = \sqrt{k_{\mathcal{G}_n}(\boldsymbol{u},\boldsymbol{u})}$; Φ_2 stands for the bi-variate normal CDF, which has no closed form; $\boldsymbol{m}_{\mathcal{G}_n}(\boldsymbol{u},\boldsymbol{u}')$ is the posterior mean vector of $\mathcal{G}$, i.e., $\boldsymbol{m}_{\mathcal{G}_n}(\boldsymbol{u},\boldsymbol{u}') = [m_{\mathcal{G}_n}(\boldsymbol{u}), m_{\mathcal{G}_n}(\boldsymbol{u}')]^\top$; $\boldsymbol{K}_{\mathcal{G}_n}(\boldsymbol{u},\boldsymbol{u}')$ is the posterior covariance matrix of $\mathcal{G}$:

$$\boldsymbol{K}_{\mathcal{G}_n}(\boldsymbol{u},\boldsymbol{u}') = \begin{bmatrix} \sigma_{\mathcal{G}_n}^2(\boldsymbol{u}) & k_{\mathcal{G}_n}(\boldsymbol{u}',\boldsymbol{u}) \\ k_{\mathcal{G}_n}(\boldsymbol{u},\boldsymbol{u}') & \sigma_{\mathcal{G}_n}^2(\boldsymbol{u}') \end{bmatrix}. \tag{12}$$

The posterior mean and variance of the failure probability $\mathcal{P}_f$ read:

$$m_{\mathcal{P}_{f,n}} = \int_{\mathcal{U}} \Phi\left(-\frac{m_{\mathcal{G}_n}(\boldsymbol{u})}{\sigma_{\mathcal{G}_n}(\boldsymbol{u})}\right)\phi_{\boldsymbol{U}}(\boldsymbol{u})\mathrm{d}\boldsymbol{u}, \tag{13}$$

$$\sigma_{\mathcal{P}_{f,n}}^2 = \int_{\mathcal{U}}\int_{\mathcal{U}} \left[\Phi_2\left([0,0]^\top; \boldsymbol{m}_{\mathcal{G}_n}(\boldsymbol{u},\boldsymbol{u}'), \boldsymbol{K}_{\mathcal{G}_n}(\boldsymbol{u},\boldsymbol{u}')\right) - \Phi\left(\frac{-m_{\mathcal{G}_n}(\boldsymbol{u})}{\sigma_{\mathcal{G}_n}(\boldsymbol{u})}\right)\Phi\left(\frac{-m_{\mathcal{G}_n}(\boldsymbol{u}')}{\sigma_{\mathcal{G}_n}(\boldsymbol{u}')}\right)\right]\phi_{\boldsymbol{U}}(\boldsymbol{u})\phi_{\boldsymbol{U}}(\boldsymbol{u}')\mathrm{d}\boldsymbol{u}\mathrm{d}\boldsymbol{u}', \tag{14}$$

where $\mathcal{P}_{f,n}$ denotes the posterior distribution of $\mathcal{P}_f$ conditional on $\boldsymbol{\mathcal{D}}$.

The above BFPI framework treats the problem of failure probability estimation as a Bayesian inference problem, and provides a principled Bayesian approach to inferring the failure probability. As such, it belongs to a class of probabilistic numerics, i.e., probabilistic integration [23, 24]. Two salient features of the BFPI framework are: (1) it allows the numerical uncertainty (i.e., discretization error) to be quantified through a computational pipeline; (2) it permits the incorporation of our prior knowledge about the performance function. Nevertheless, one main drawback is that the posterior mean and variance of the failure probability are not analytically tractable. In particular, it should be noted that the posterior variance involves the evaluating the posterior covariance of $\mathcal{G}$ and integrating with respect to the bivariate normal CDF (which itself usually requires numerical integration). This, of course, poses a significant computational challenge to the development of Bayesian active learning reliability methods.

3. Quasi-Bayesian active learning cubature

This section is devoted to the development of a Bayesian active learning method, QBALC, for structural reliability analysis with extremely small failure probabilities using the BFPI framework. First, a stopping criterion is proposed as one of the main components for Bayesian active learning based on the simplification of the posterior variance of the failure probability. Second, the analytically intractable integrals involved in the stopping criterion are solved with an effective numerical integration technique. Third, a learning function is derived from the simplified posterior variance as another ingredient for Bayesian active learning. Fourth, the step-by-step procedure for implementing the proposed method is summarized.

3.1. Stopping criterion

A well-defined stopping criterion is crucial for a Bayesian active learning method, as it determines when the active learning phase should be stopped. The choice of stopping criterion depends on several factors, such as the specific goals and available computational resources. In this study, we are particularly interested in developing a stopping criterion that can reflect whether the posterior mean of the failure probability (i.e. $m_{P_{f,n}}$) as a predictor of the failure probability reaches a satisfactory level of accuracy. A natural choice would

be to use the posterior coefficient of variation of the failure probability. However, such a stopping criterion can be computationally prohibitive, mainly due to the numerical complexity of the posterior variance of the failure probability. With this in mind, our basic idea is to find a simplified version of the posterior variance defined in Eq. (14) that is computationally tractable without losing too much precision.

Note that the posterior variance of the failure probability is actually an expectation integral with respect to the posterior covariance function of $\mathcal{I}$ such that:

$$\sigma^2_{\mathcal{P}_{f,n}} = \int_{\mathcal{U}}\int_{\mathcal{U}} k_{\mathcal{I}_n}(\boldsymbol{u},\boldsymbol{u}')\phi_{\boldsymbol{U}}(\boldsymbol{u})\phi_{\boldsymbol{U}}(\boldsymbol{u}')\mathrm{d}\boldsymbol{u}\mathrm{d}\boldsymbol{u}'. \tag{15}$$

The above equation can be further written as:

$$\sigma^2_{\mathcal{P}_{f,n}} = \int_{\mathcal{U}}\int_{\mathcal{U}} \rho_{\mathcal{I}_n}(\boldsymbol{u},\boldsymbol{u}')\sigma_{\mathcal{I}_n}(\boldsymbol{u})\sigma_{\mathcal{I}_n}(\boldsymbol{u}')\phi_{\boldsymbol{U}}(\boldsymbol{u})\phi_{\boldsymbol{U}}(\boldsymbol{u}')\mathrm{d}\boldsymbol{u}\mathrm{d}\boldsymbol{u}', \tag{16}$$

where $\rho_{\mathcal{I}_n} \in [-1,1]$ is the posterior correlation coefficient of $\mathcal{I}$; $\sigma_{\mathcal{I}_n}(\boldsymbol{u})$ is the posterior standard deviation function of $\mathcal{I}$, which has the following expression:

$$\sigma_{\mathcal{I}_n}(\boldsymbol{u}) = \sqrt{\Phi\left(-\frac{m_{\mathcal{G}_n}(\boldsymbol{u})}{\sigma_{\mathcal{G}_n}(\boldsymbol{u})}\right)\Phi\left(\frac{m_{\mathcal{G}_n}(\boldsymbol{u})}{\sigma_{\mathcal{G}_n}(\boldsymbol{u})}\right)}. \tag{17}$$

To avoid solving the correlation coefficient $\rho_{\mathcal{I}_n}(\boldsymbol{u},\boldsymbol{u}')$ and also the double integral in Eq. (16), let us replace $\rho_{\mathcal{I}_n}(\boldsymbol{u},\boldsymbol{u}')$ by an equivalent constant $\tilde{\rho}$ such that:

$$\begin{aligned}\tilde{\sigma}^2_{\mathcal{P}_{f,n}} =& \int_{\mathcal{U}}\int_{\mathcal{U}} \tilde{\rho}\sigma_{\mathcal{I}_n}(\boldsymbol{u})\sigma_{\mathcal{I}_n}(\boldsymbol{u}')\phi_{\boldsymbol{U}}(\boldsymbol{u})\phi_{\boldsymbol{U}}(\boldsymbol{u}')\mathrm{d}\boldsymbol{u}\mathrm{d}\boldsymbol{u}' \\ =& \tilde{\rho}\left[\int_{\mathcal{U}} \sigma_{\mathcal{I}_n}(\boldsymbol{u})\phi_{\boldsymbol{U}}(\boldsymbol{u})\mathrm{d}\boldsymbol{u}\right]^2 \\ =& \tilde{\rho}\left[\int_{\mathcal{U}} \sqrt{\Phi\left(-\frac{m_{\mathcal{G}_n}(\boldsymbol{u})}{\sigma_{\mathcal{G}_n}(\boldsymbol{u})}\right)\Phi\left(\frac{m_{\mathcal{G}_n}(\boldsymbol{u})}{\sigma_{\mathcal{G}_n}(\boldsymbol{u})}\right)}\phi_{\boldsymbol{U}}(\boldsymbol{u})\mathrm{d}\boldsymbol{u}\right]^2,\end{aligned} \tag{18}$$

where $\tilde{\sigma}^2_{\mathcal{P}_{f,n}}$ is referred to as the quasi posterior variance of the failure probability; the equivalent correlation coefficient $\tilde{\rho}$ should take a value between 0 and 1, which is defined by:

$$\tilde{\rho} = \frac{\sigma^2_{\mathcal{P}_{f,n}}}{\left[\int_{\mathcal{U}} \sqrt{\Phi\left(-\frac{m_{\mathcal{G}_n}(\boldsymbol{u})}{\sigma_{\mathcal{G}_n}(\boldsymbol{u})}\right)\Phi\left(\frac{m_{\mathcal{G}_n}(\boldsymbol{u})}{\sigma_{\mathcal{G}_n}(\boldsymbol{u})}\right)}\phi_{\boldsymbol{U}}(\boldsymbol{u})\mathrm{d}\boldsymbol{u}\right]^2}. \tag{19}$$

It is worth pointing out that once $\tilde{\rho}$ is given, the quasi posterior variance $\tilde{\sigma}^2_{\mathcal{P}_{f,n}}$ can be much cheaper to compute than the exact one $\sigma^2_{\mathcal{P}_{f,n}}$. When $\tilde{\rho} = 1$, the quasi posterior variance $\tilde{\sigma}^2_{\mathcal{P}_{f,n}}$ reduces to the upper bound of the posterior variance $\sigma^2_{\mathcal{P}_{f,n}}$ given in [14, 15].

In this study, it is suggested that the stopping criterion could be set as follows:

$$\tilde{\delta}_{\mathcal{P}_{f,n}} = \frac{\tilde{\sigma}_{\mathcal{P}_{f,n}}}{m_{\mathcal{P}_{f,n}}} < \epsilon, \tag{20}$$

where $\tilde{\delta}_{\mathcal{P}_{f,n}}$ is referred to as the quasi posterior COV of the failure probability; ϵ is a user-specified threshold. To use this stopping criterion in practice, two problems need to be considered and addressed properly. The first one is related to the choice of $\tilde{\rho}$. An ideal choice is according to Eq. (19). However, this is clearly not feasible as it requires evaluating the original posterior variance $\sigma^2_{\mathcal{P}_{f,n}}$ that we want to avoid. A more pragmatic strategy for choosing $\tilde{\rho}$ might be to use our computational experience. This is most likely feasible because the value of $\tilde{\rho}$ is only in a small interval between 0 and 1. The second problem concerns the evaluation of $m_{\mathcal{P}_{f,n}}$ and $\tilde{\sigma}_{\mathcal{P}_{f,n}}$, due to their analytical intractability. To ensure the computational accuracy and efficiency, a suitable numerical integrator is of vital importance. In this paper, the variance-amplified importance sampling (VAIS) method originally developed in [16] is applied in a sequential manner.

The VAIS estimators of $m_{\mathcal{P}_{f,n}}$ and $\tilde{\sigma}_{\mathcal{P}_{f,n}}$ can be given by:

$$\hat{m}_{\mathcal{P}_{f,n}} = \frac{1}{N}\sum_{i=1}^{N} \Phi\left(-\frac{m_{\mathcal{G}_n}(\boldsymbol{u}^{(i)})}{\sigma_{\mathcal{G}_n}(\boldsymbol{u}^{(i)})}\right)\frac{\phi_{\boldsymbol{U}}(\boldsymbol{u}^{(i)})}{h(\boldsymbol{u}^{(i)})}, \tag{21}$$

$$\hat{\tilde{\sigma}}_{\mathcal{P}_{f,n}} = \frac{\sqrt{\tilde{\rho}}}{N}\sum_{i=1}^{N} \sqrt{\Phi\left(-\frac{m_{\mathcal{G}_n}(\boldsymbol{u}^{(i)})}{\sigma_{\mathcal{G}_n}(\boldsymbol{u}^{(i)})}\right)\Phi\left(\frac{m_{\mathcal{G}_n}(\boldsymbol{u}^{(i)})}{\sigma_{\mathcal{G}_n}(\boldsymbol{u}^{(i)})}\right)}\frac{\phi_{\boldsymbol{U}}(\boldsymbol{u}^{(i)})}{h(\boldsymbol{u}^{(i)})}, \tag{22}$$

where $h(\boldsymbol{u})$ is the sampling density, which equals to the joint PDF of d independent normal variables with a mean of zero and a standard deviation of $\lambda > 1$; $\left\{\boldsymbol{u}^{(i)}\right\}_{i=1}^{N}$ is a set of N random samples drawn from $h(\boldsymbol{u})$. The variances of the two estimators can be formulated as:

$$\mathbb{V}\left[\hat{m}_{\mathcal{P}_{f,n}}\right] = \frac{1}{N-1}\left\{\frac{1}{N}\sum_{i=1}^{N}\left[\Phi\left(-\frac{m_{\mathcal{G}_n}(\boldsymbol{u}^{(i)})}{\sigma_{\mathcal{G}_n}(\boldsymbol{u}^{(i)})}\right)\frac{\phi_{\boldsymbol{U}}(\boldsymbol{u}^{(i)})}{h(\boldsymbol{u}^{(i)})}\right]^2 - \hat{m}^2_{\mathcal{P}_{f,n}}\right\}, \tag{23}$$

$$\mathbb{V}\left[\hat{\tilde{\sigma}}_{\mathcal{P}_{f,n}}\right] = \frac{1}{N-1}\left\{\frac{\tilde{\rho}}{N}\sum_{i=1}^{N}\left[\sqrt{\Phi\left(-\frac{m_{\mathcal{G}_n}(\boldsymbol{u}^{(i)})}{\sigma_{\mathcal{G}_n}(\boldsymbol{u}^{(i)})}\right)\Phi\left(\frac{m_{\mathcal{G}_n}(\boldsymbol{u}^{(i)})}{\sigma_{\mathcal{G}_n}(\boldsymbol{u}^{(i)})}\right)}\frac{\phi_{\boldsymbol{U}}(\boldsymbol{u}^{(i)})}{h(\boldsymbol{u}^{(i)})}\right]^2 - \hat{\tilde{\sigma}}^2_{\mathcal{P}_{f,n}}\right\}, \tag{24}$$

where $\mathbb{V}$ is the variance operator. Given a sample set $\left\{\boldsymbol{u}^{(i)}\right\}_{i=1}^{N}$, we can obtain the estimates of $m_{\mathcal{P}_{f,n}}$ and $\tilde{\sigma}_{\mathcal{P}_{f,n}}$ using Eqs. (21) and (22) and their associated variances using Eqs. (23) and (24). However, it is most likely that the appropriate sample size to ensure that the two estimates reach a desirable level of accuracy is not known a priori. Furthermore, if one tends to choose a sample size that is too large, it may not be

feasible for the GP posterior predictions due to numerical issues. For these reasons, the sample size should be enlarged gradually, as described below.

For convenience, assume that the sample size is the same for each enrichment, denoted as N_0. At the j-th step, a set of N_0 random samples $\left\{\boldsymbol{u}^{(i)}\right\}_{i=1}^{N_0}$ are first generated from $h(\boldsymbol{u})$. Then, the following two quantities are evaluated for each sample $\boldsymbol{u}^{(i)}$:

$$\eta^{(i)} = \Phi\left(-\frac{m_{\mathcal{G}_n}(\boldsymbol{u}^{(i)})}{\sigma_{\mathcal{G}_n}(\boldsymbol{u}^{(i)})}\right), \tag{25}$$

$$\gamma^{(i)} = \frac{\phi_{\boldsymbol{U}}(\boldsymbol{u}^{(i)})}{h(\boldsymbol{u}^{(i)})}. \tag{26}$$

Next, we evaluate the following four quantities:

$$m^{(j)} = \frac{1}{N_0}\sum_{i=1}^{N_0}\eta^{(i)}\gamma^{(i)}, \tag{27}$$

$$\tilde{\sigma}^{(j)} = \frac{\tilde{\rho}}{N_0}\sum_{i=1}^{N_0}\sqrt{\eta^{(i)}(1-\eta^{(i)})}\gamma^{(i)}, \tag{28}$$

$$r^{(j)} = \frac{1}{N_0}\sum_{i=1}^{N_0}\left[\eta^{(i)}\gamma^{(i)}\right]^2, \tag{29}$$

$$s^{(j)} = \frac{\tilde{\rho}}{N_0}\sum_{i=1}^{N_0}\left[\sqrt{\eta^{(i)}(1-\eta^{(i)})}\gamma^{(i)}\right]^2. \tag{30}$$

After that, the estimates and their associated variances of $m_{\mathcal{P}_{f,n}}$ and $\tilde{\sigma}_{\mathcal{P}_{f,n}}$ can be computed as follows:

$$\hat{m}_{\mathcal{P}_{f,n}} = \frac{1}{j}\sum_{t=1}^{j} m^{(t)}, \tag{31}$$

$$\hat{\tilde{\sigma}}_{\mathcal{P}_{f,n}} = \frac{1}{j}\sum_{t=1}^{j} \tilde{\sigma}^{(t)} \tag{32}$$

$$\mathbb{V}\left[\hat{m}_{\mathcal{P}_{f,n}}\right] = \frac{1}{jN_0-1}\left[\frac{1}{j}\sum_{t=1}^{j} r^{(t)} - \hat{m}^2_{\mathcal{P}_{f,n}}\right], \tag{33}$$

$$\mathbb{V}\left[\hat{\tilde{\sigma}}_{\mathcal{P}_{f,n}}\right] = \frac{1}{jN_0-1}\left[\frac{1}{j}\sum_{t=1}^{j} s^{(t)} - \hat{\tilde{\sigma}}^2_{\mathcal{P}_{f,n}}\right]. \tag{34}$$

Repeat the above procedure until a stopping criterion is reached, e.g., $\sqrt{\mathbb{V}\left[\hat{m}_{\mathcal{P}_{f,n}}\right]}/\hat{m}_{\mathcal{P}_{f,n}} < \tau_1$ and $\sqrt{\mathbb{V}\left[\hat{\tilde{\sigma}}_{\mathcal{P}_{f,n}}\right]/\hat{\tilde{\sigma}}_{\mathcal{P}_{f,n}}} < \tau_2$, where τ_1 and τ_2 are two user-specified tolerances. An important advantage of

the above process is that the most time-consuming term $\eta^{(i)}$ is reused in several places, hence reducing the overall computation time.

The terms $m_{\mathcal{P}_{f,n}}$ and $\tilde{\sigma}_{\mathcal{P}_{f,n}}$ in Eq. (20) should thus be replaced by their respective estimates $\hat{m}_{\mathcal{P}_{f,n}}$ and $\hat{\tilde{\sigma}}_{\mathcal{P}_{f,n}}$. Since both $\hat{m}_{\mathcal{P}_{f,n}}$ and $\hat{\tilde{\sigma}}_{\mathcal{P}_{f,n}}$ may process a certain amount of error depending on the values of τ_1 and τ_2, the stopping criterion in Eq. (20) may need to be satisfied several times in a row to avoid fake convergence.

3.2. Learning function

Another essential component of a Bayesian active learning method is the learning function, which comes into play when the stopping criterion is not satisfied. Specifically, a learning function can guide the learning process by suggesting one or multiple informative points at which to observe the $\mathcal{G}$-function next. In general, there are many ways to construct a capable learning function. In our context, we are especially interested in making fullest possible use of the available posterior statistics of the failure probability. In addition, the resulting learning function should facilitate the selection of multiple points at each iteration, and thus enabling parallel distributed processing and reducing the overall computational burden.

The proposed learning function, called 'penalized quasi posterior variance contribution' (PQPVC), has the following form:

$$\mathrm{PQPVC}(\boldsymbol{u}|p) = \sqrt{\varPhi\left(-\frac{m_{\mathcal{G}_n}(\boldsymbol{u})}{p\sigma_{\mathcal{G}_n}(\boldsymbol{u})}\right)\varPhi\left(\frac{m_{\mathcal{G}_n}(\boldsymbol{u})}{p\sigma_{\mathcal{G}_n}(\boldsymbol{u})}\right)}\phi_{\boldsymbol{U}}(\boldsymbol{u}), \tag{35}$$

where $p \in (0, 1]$ is the penalty factor that penalizes the current posterior standard deviation function of $\mathcal{G}$. Obviously $\sqrt{\bar{\rho}}\int_{\mathcal{U}} \mathrm{PQPVC}(\boldsymbol{u}|p=1)\mathrm{d}\boldsymbol{u} = \tilde{\sigma}_{\mathcal{P}_{f,n}}$ holds. Therefore, the PQPVC function given $p = 1$ can be interpreted as a scaled measure of the contribution at point $\boldsymbol{u}$ to the quasi posterior standard deviation (hence also the quasi posterior variance) of the failure probability. Moreover, the learning function called 'upper bound posterior variance contribution' developed in [14, 15] turns out to be a special case of the PQPVC function when $p = 1$. It must be stressed that the introduction of the penalty factor p is quite crucial, as it facilitates the selection of a set of points by simply optimizing the PQPVC function given different p. The reason why we penalize the current posterior standard deviation function $\sigma_{\mathcal{G}_n}(\boldsymbol{u})$ but leave the posterior

mean function $m_{\mathcal{G}_n}(\boldsymbol{u})$ unchanged is because the posterior standard deviation at any unobserved point, which is important for an accurate failure probability estimation, is most likely to decrease in the future, while it is difficult to prejudge whether its posterior mean will increase or decrease.

Suppose that we wish to select n_{add} points, which are denoted as $\left\{\boldsymbol{u}^{+,(i)}\right\}_{i=1}^{n_a}$. The i-th point $\boldsymbol{u}^{+,(i)}$ can be identified by maximizing the proposed PQPVC function such that:

$$\boldsymbol{u}^{+,(i)} = \underset{\boldsymbol{u}\in[-R,R]^d}{\arg\max}\ \mathrm{PQPVC}(\boldsymbol{u}|p=\frac{i}{n_a}), \tag{36}$$

where $[-R,R]^d$ is a hyperrectangle defining a reduced region of in the d-dimensional standard normal space; R is the side length, which can be specified according to $R=\sqrt{\chi_d^{-2}(1-\upsilon)}$, where χ_d^2 is the CDF of a chi-squared distribution of degree d and the parameter υ is set to be 10^{-10}. In Eq. (36), the penalty factor p is given as $\frac{i}{n_a}$ so that its values are equally spaced within $(0,1]$. In order to produce n_{add} points, the PQPVC function must be optimized n_{add} times. Fortunately, the time required for optimization is negligible compared to the time required for evaluating the $\mathcal{G}$ function, which is often computationally expensive in practice. Thus, the optimization problem can be solved by any suitable global optimization algorithm. Usually, if n_a is not too large, a set of diverse points can be identified by our multi-point selection strategy.

3.3. Numerical implementation procedure of the proposed method

The step-by-step procedure for implementing the proposed QBALC method is summarized below and accompanied by the flowchart shown in Fig. 2.

Step 1: Generate an initial observation dateset

The proposed method needs to be initialized with an initial dateset from observing the $\mathcal{G}$-function. This can be achieved by first generating a small number (say n_0) of samples $\boldsymbol{\mathcal{U}}=\left[\boldsymbol{u}^{(1)},\boldsymbol{u}^{(2)},\cdots,\boldsymbol{u}^{(n_0)}\right]^\top$ that are uniformly distributed within a d-ball of radius R_0 using the Hammersley sequence. The radius R_0 can be specified by $R_0=\sqrt{\chi_d^{-2}(1-\upsilon_0)}$ with $\upsilon_0=10^{-8}$. Next, evaluating the $\mathcal{G}$-function at these points $\boldsymbol{\mathcal{U}}$ gives the output values $\boldsymbol{\mathcal{Y}}=\left[y^{(1)},y^{(2)},\cdots,y^{(n_0)}\right]^\top$ with $y^{(i)}=\mathcal{G}(\boldsymbol{u}^{(i)})$. Finally, the initial observation dateset is constructed as $\boldsymbol{\mathcal{D}}=\{\boldsymbol{\mathcal{U}},\boldsymbol{\mathcal{Y}}\}$. Let $n=n_0$.

Step 2: Obtain the GP posterior of the $\mathcal{G}$-function

This step involves obtaining the posterior distribution of the $\mathcal{G}$-function $\mathcal{GP}(m_{\mathcal{G}_n}(\boldsymbol{u}), k_{\mathcal{G}_n}(\boldsymbol{u}, \boldsymbol{u}'))$ conditional on the observation dataset $\boldsymbol{\mathcal{D}}$. In this study, the *fitrgp* function available in the Statistics and Machine Learning Toolbox of Matlab is used, where the prior mean and covariance functions are specified as a constant and an anisotropic squared exponential kernel, respectively.

Step 3: Compute the posterior statistics of the failure probability

At this stage, one needs to compute the posterior mean estimate $\hat{m}_{\mathcal{P}_{f,n}}$ and the quasi posterior standard deviation estimate $\hat{\tilde{\sigma}}_{\mathcal{P}_{f,n}}$ of the failure probability using the sequential VIAS method, as described in subsection 3.1.

Step 4: Check the stopping criterion

If $\frac{\hat{\tilde{\sigma}}_{\mathcal{P}_{f,n}}}{\hat{m}_{\mathcal{P}_{f,n}}} < \epsilon$ is satisfied twice in a row, go to **Step 6**; Otherwise, go to **Step 5**.

Step 5: Enrich the observation dataset

In this step, we need to enrich the currently available observation dataset with some newly identified data. First, the next best points $\boldsymbol{\mathcal{U}}^+ = \left\{\boldsymbol{u}^{+,(i)}\right\}_{i=1}^{n_a}$ where to evaluate the $\mathcal{G}$-function can be selected by optimizing the PQPVC function, where the genetic algorithm is used in this study. After that, the corresponding output values $\boldsymbol{\mathcal{Y}}^+ = \left\{y^{+,(i)}\right\}_{i=1}^{n_a}$ of the $\mathcal{G}$-function at $\boldsymbol{\mathcal{U}}^+$ are obtained using parallel computing, where $y^{+,(i)} = \mathcal{G}(\boldsymbol{u}^{+,(i)})$. At last, the current observation dataset is enriched with $\boldsymbol{\mathcal{D}}^+ = \left\{\boldsymbol{\mathcal{U}}^+, \boldsymbol{\mathcal{Y}}^+\right\}$ such that $\boldsymbol{\mathcal{D}} = \boldsymbol{\mathcal{D}} \cup \boldsymbol{\mathcal{D}}^+$. Let $n = n + n_a$ and go to **Step 2**.

Step 6: Stop the method

Return $\hat{m}_{\mathcal{P}_{f,n}}$ as the failure probability estimate and stop the algorithm.

4. Numerical examples

To illustrate the performance of the proposed QBALC method, four numerical examples are studied in this section. In all the examples, some of the parameters of the proposed method are set to $n_0 = 10$, $\lambda = 2.0$, $\tau_1 = \tau_2 = 2\%$, $\epsilon = 5\%$. Multiple cases of the remaining parameters $\tilde{\rho}$ and n_{add} are considered in order to see their effects. If applicable, the crude MCS with a considerably large sample size is carried out to provide a reference solution for the failure probability. For comparison purposes, several exiting competing methods in

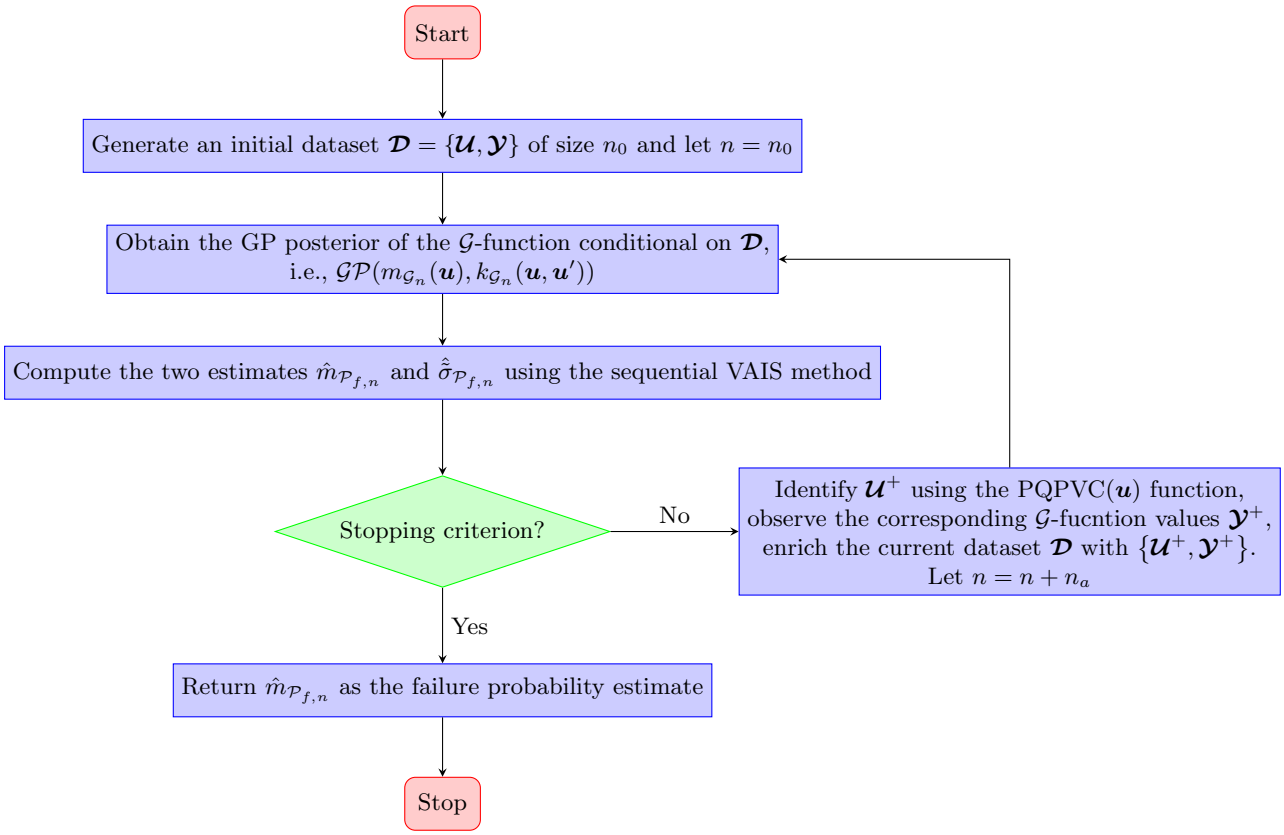


Figure 2: Flowchart of the proposed QBALC method.

the literature, i.e., Active learning Kriging Markov Chain Monte Carlo (AK-MCMC) [25], Bayesian subset simulation (BSS) [26] and extreme AK-MCS (eAK-MCS) [27], are also implemented in each example. The initial sample size is set to 10 for all (Bayesian) active learning methods to make the comparison as fair as possible. To evaluate the robustness of all methods except MCS, 20 independent runs are performed and the corresponding statistical results are reported.

4.1. Example 1: A series system with four branches

The first example considers a series system with two linear branches and two nonlinear branches, which has been used extensively in many studies (e.g., [11, 15, 16]). The performance function is given by:

$$g(\boldsymbol{X}) = \min \begin{cases} a + \frac{(X_1 - X_2)^2}{10} - \frac{(X_1 + X_2)}{\sqrt{2}} \\ a + \frac{(X_1 - X_2)^2}{10} + \frac{(X + X_2)}{\sqrt{2}} \\ (X_1 - X_2) + \frac{b}{\sqrt{2}} \\ (X_2 - X_1) + \frac{b}{\sqrt{2}} \end{cases}, \tag{37}$$

where X_1 and X_2 are two standard normal variables that are independently and identically distributed; a and b are two constant parameters, which are specified as $a = 6$ and $b = 12$ in this study.

Table 1 summarizes the results obtained using several structural reliability analysis methods. The reference value of the failure probability is 3.01×10^{-9} with a COV of 1.82%, provided by MCS with 10^{12} samples. AK-MCMC requires an average of 171.10 iterations (equivalent to an average of 180.10 performance function calls), but it gives a slightly smaller failure probability mean with a very large COV, say 29.22%. BSS can significantly reduce the average number of iterations and $\mathcal{G}$ function calls, and also produce a more unbiased failure probability mean compared to AK-MCMC. Nevertheless, its robustness is not good, as evidenced by the large value of the COV, which is up to 28.58%. Like the proposed QBALC method, eAK-MCS allows us to select multiple points at each iteration. Unfortunately, it encounters non-convergence problem in this example, so its results are missing. Considering different parameter combinations (n_a and $\sqrt{\tilde{\rho}}$), a total of 18 cases of the proposed QBALC method are investigated. Overall, the proposed method performs very well in almost all the studied cases. Besides, it is also found that: (1) For a fixed $\sqrt{\tilde{\rho}}$, the average number of iterations can be reduced by increasing n_a from 1 to 6, though the average number of $\mathcal{G}$-function calls also increases; (2) For a fixed n_a, the average number of iterations and $\mathcal{G}$-function calls can be increased by increasing $\sqrt{\tilde{\rho}}$ from 0.25 to 0.75, while the COV of the failure probability estimates decreases.

To further illustrate how the proposed method works, Fig. 3 shows the points selected at each iteration with an arbitrary run of the proposed method ($n_a = 2$ and $\sqrt{\tilde{\rho}} = 0.50$), together with the true limit state

curve. It can be observed that: (1) the initial 10 points are evenly distributed as we expected; (2) the two points identified by the proposed learning function are far apart in some iterations, and are close but not identical in others; (3) most of the identified points from iterations 2-18 are distributed around the four regions of the true limit state curve that are important for accurate failure probability estimation.

4.2. Example 2: A nonlinear oscillator

As a second example, we consider a nonlinear single-degree-of-freedom oscillator driven by a rectangular pulse load [7], as shown in Fig. 4. The performance function is given as follows:

$$g\left(m, c_1, c_2, r, F_1, t_1\right) = 3r - \left| \frac{2F_1}{c_1 + c_2} \sin\left(\frac{t_1}{2} \sqrt{\frac{c_1 + c_2}{m}} \right) \right|, \tag{38}$$

where m, c_1, c_2, r, F_1 and t_1 are six random variables, as described in Table 2.

The results of several methods, i.e., MCS, AK-MCMC, BSS, eAK-MCS and QBALC, are reported in Table 3. We take the reference failure probability to be 1.52×10^{-8} (with a COV of 2.56%), which is produced by MCS with 10^{11} samples. AK-MCMC gives a fairly good failure probability mean with a very small COV (i.e., 0.88%). However, it requires an average of 176.25 iterations (corresponding to an average of 185.25 $\mathcal{G}$-function evaluations), which is the most of the four competing methods and far more than others. The number of iterations on average can be significantly reduced to 25.10 by BSS, but the variability of its failure probability estimates is quite large, as indicated by the COV. By selecting $n_a = 4$ points at each iteration of the active learning phase, eAK-MCS only needs 7.95 iterations on average (34.10 $\mathcal{G}$-function calls) and gives a failure probability mean of 1.55×10^{-8} with a COV of 6.61%. Under the same setting (i.e. $n_a = 4$), the proposed QBALC method can perform better than eAK-MCS ($n_a = 4$) overall, except for $\sqrt{\tilde{\rho}} = 0.25$. Furthermore, for the proposed method it can be seen that the average number of iterations can be reduced by increasing n_a, but increased by enlarging $\sqrt{\tilde{\rho}}$. It should also be noted that in some cases, when $\sqrt{\tilde{\rho}} = 0.25$, the proposed method can produce a COV significantly greater than 5%.

4.3. Example 3: A reinforced concrete section

The third example involves the bending limit state a reinforced concrete section [28], as shown in Fig. 5. The performance function is formulated as:

$$Z = g(\boldsymbol{X}) = X_1 X_2 X_3 - \frac{X_1^2 X_2^2 X_4}{X_5 X_6} - X_7, \tag{39}$$

where X_1 to X_7 are seven random variables, as listed in Table 4.

In Table 5, we summarize the results obtained from several structural reliability analysis methods. The failure probability estimate by MCS with 5×10^{11} samples is 1.57×10^{-8} with a COV of 1.13%, which is adopted as the reference solution. At cost of an average of 143.65 iterations (152.65 $\mathcal{G}$-function calls), AK-MCMC gives a failure probability mean close to the reference one, with a small COV. BSS requires much less iterations on average, but its COV is quite large, say 34.88%. Note that eAK-MCS ($n_a = 4$) requires a slightly smaller average N_{iter} (or N_{call}) than the proposed QBALC method ($n_a = 4$), while producing a larger variability in the failure probability results (say $\delta_{\hat{P}_f} = 5.02\%$). On the contrary, in all 18 cases studied, the proposed method is able to give an almost unbiased failure probability mean with a COV less than 5%.

4.4. Example 4: A 56-bar space truss structure

The fourth and last example consists of a 56-bar space truss structure that was studied early in [29], as shown in Fig. 6. The structure is modeled as a three-dimensional finite element model using OpenSees with 56 truss elements and 25 nodes. Nine external loads, denoted $P_1, P_2, \cdots, P_9$, are applied to nodes $1, 2, \cdots, 9$ along the negative z-axis. It is assumed that the modulus of elasticity and the cross-sectional area of each member are the same and are denoted as E and A respectively. The structure is considered to failure when the vertical displacement of the top node exceeds a certain threshold, resulting in the following performance function:

$$g(P_1, P_2, \cdots, P_9, E, A) = \Delta - V_1(P_1, P_2, \cdots, P_9, E, A), \tag{40}$$

where V_1 is the vertical displacement of node 1; Δ is the tolerance, which is specified as 50 mm; $P_1, P_2, \cdots, P_9$, E and A are 11 random variables, as listed in Table 6.

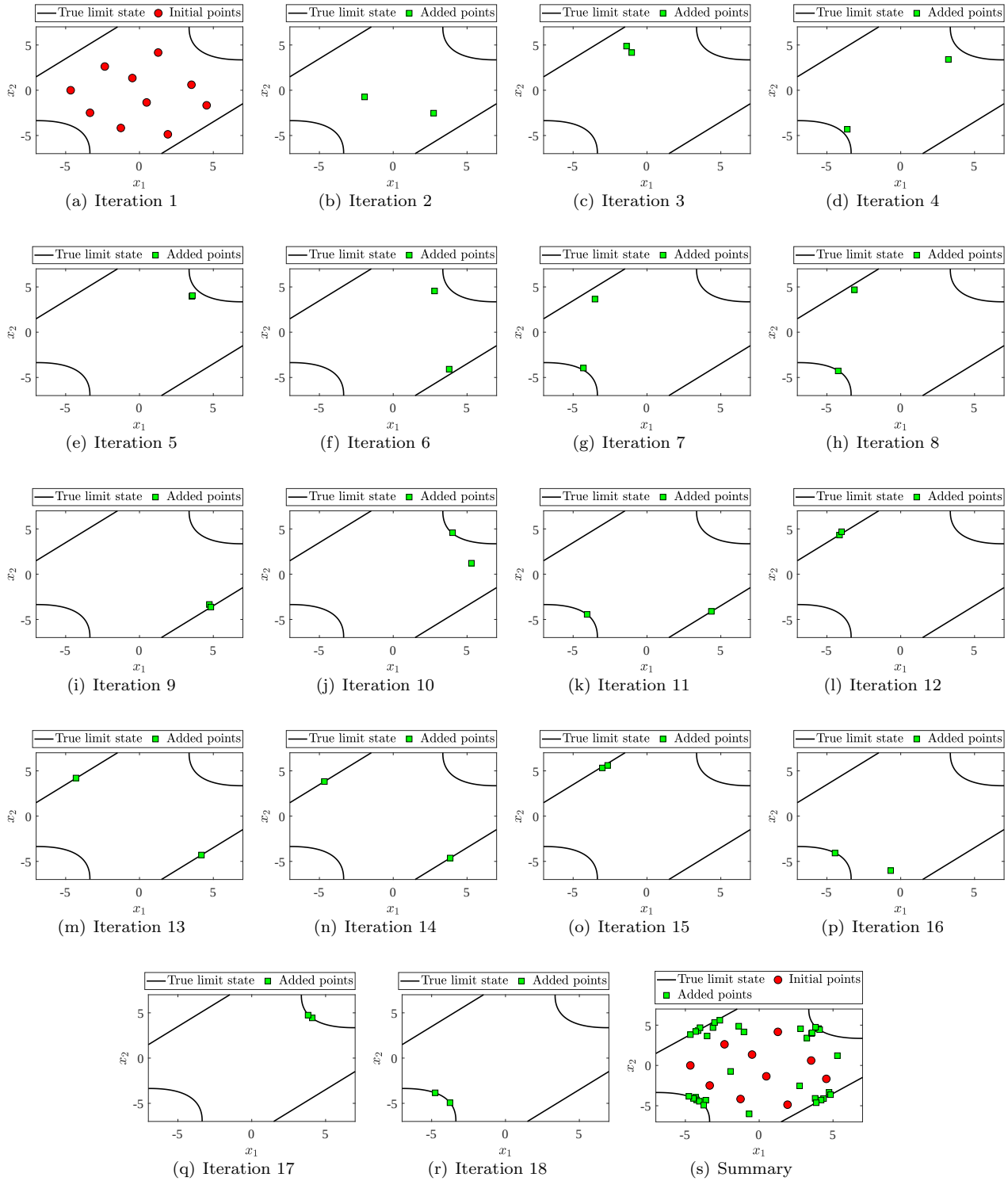
We implement the importance sampling (IS) method available in UQLab [30] as an alternative to providing a reference solution, as MCS is computationally prohibitive in this example. The results of IS and several other methods are listed in Table 7. The failure probability estimate given by IS is 4.94×10^{-8} with a COV of 1.00%, at the cost of 66,107 $\mathcal{G}$-function evaluations. The two non-parallel active learning methods, namely AK-MCMC and BSS, are either too computationally intensive or lack robustness. eAK-MCS as a parallel active learning method fails to converge in some trials, so its results are missing. In contrast, the proposed QBALC method ($n_a = 4$) can produce fairly good results in all three cases $\sqrt{\tilde{\rho}} = 0.25, 0.50, 0.75$ with less than 10 iterations. Note also that as $\sqrt{\tilde{\rho}}$ increases, $\delta_{\hat{P}_f}$ decreases.

4.5. Final remarks

Through the four numerical examples, we have studied the effects of the parameters n_a and $\sqrt{\tilde{\rho}}$ on the performance of the proposed QBALC method. In general, it can be observed that the proposed method: (1) can produce a failure probability mean with a COV less than 5% in all the studied cases, except for $\sqrt{\tilde{\rho}} = 0.25$; (2) does not lead to a significant reduction in the number of iterations on average when n_a is larger than 4. Therefore, $\sqrt{\tilde{\rho}} = 0.50$ and $n_a = 4$ could be a good choice in practice.

Table 1: Reliability analysis results of Example 1 by several methods.

Method			N_{iter}	N_{call}	$\hat{P}_f$	$\delta_{\hat{P}_f}$
MCS	-	-	-	10^{12}	3.01×10^{-9}	1.82%
AK-MCMC	$n_a = 1$	-	171.10	180.10	2.38×10^{-9}	29.22%
BSS	$n_a = 1$	-	57.20	66.20	2.97×10^{-9}	28.58%
eAK-MCS	$n_a = 4$	-	-	-	-	-
Proposed QBALC	$n_a = 1$	$\sqrt{\bar{\rho}} = 0.25$	31.15	40.15	2.94×10^{-9}	4.87%
		$\sqrt{\bar{\rho}} = 0.50$	35.75	44.75	3.03×10^{-9}	2.57%
		$\sqrt{\bar{\rho}} = 0.75$	38.35	47.35	3.03×10^{-9}	1.58%
	$n_a = 2$	$\sqrt{\bar{\rho}} = 0.25$	17.95	43.90	2.93×10^{-9}	5.35%
		$\sqrt{\bar{\rho}} = 0.50$	20.10	48.20	3.03×10^{-9}	1.70%
		$\sqrt{\bar{\rho}} = 0.75$	20.70	49.40	3.04×10^{-9}	1.06%
	$n_a = 3$	$\sqrt{\bar{\rho}} = 0.25$	13.65	47.95	3.00×10^{-9}	4.52%
		$\sqrt{\bar{\rho}} = 0.50$	15.30	52.90	3.02×10^{-9}	1.51%
		$\sqrt{\bar{\rho}} = 0.75$	15.85	54.55	3.03×0^{-9}	1.15%
	$n_a = 4$	$\sqrt{\bar{\rho}} = 0.25$	12.05	54.20	2.99×10^{-9}	3.30%
		$\sqrt{\bar{\rho}} = 0.50$	13.10	58.40	3.03×10^{-9}	1.50%
		$\sqrt{\bar{\rho}} = 0.75$	13.45	59.80	3.01×10^{-9}	0.97%
	$n_a = 5$	$\sqrt{\bar{\rho}} = 0.25$	11.10	60.50	2.96×10^{-9}	4.11%
		$\sqrt{\bar{\rho}} = 0.50$	12.45	67.25	3.02×10^{-9}	1.12%
		$\sqrt{\bar{\rho}} = 0.75$	12.25	66.25	3.03×10^{-9}	1.03%
	$n_a = 6$	$\sqrt{\bar{\rho}} = 0.25$	10.04	66.40	3.02×10^{-9}	1.51%
		$\sqrt{\bar{\rho}} = 0.50$	11.40	72.40	3.02×10^{-9}	0.80%
		$\sqrt{\bar{\rho}} = 0.75$	11.70	74.20	3.02×10^{-9}	0.73%



(a) Iteration 1 (b) Iteration 2 (c) Iteration 3 (d) Iteration 4

(e) Iteration 5 (f) Iteration 6 (g) Iteration 7 (h) Iteration 8

(i) Iteration 9 (j) Iteration 10 (k) Iteration 11 (l) Iteration 12

(m) Iteration 13 (n) Iteration 14 (o) Iteration 15 (p) Iteration 16

(q) Iteration 17 (r) Iteration 18 (s) Summary

Figure 3: Illustration of the proposed QBALC method ($n_a = 2$ and $\sqrt{\bar{\rho}} = 0.50$) for Example 1.

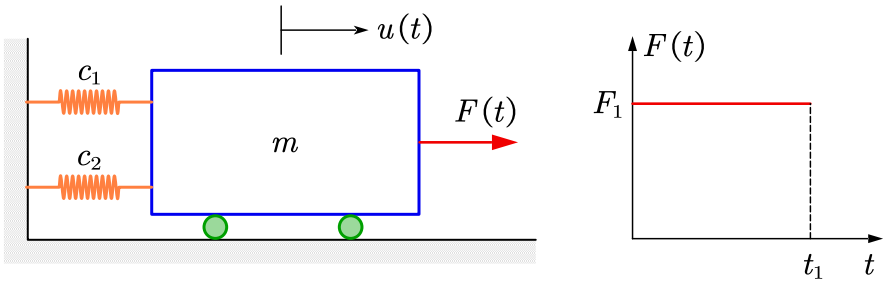
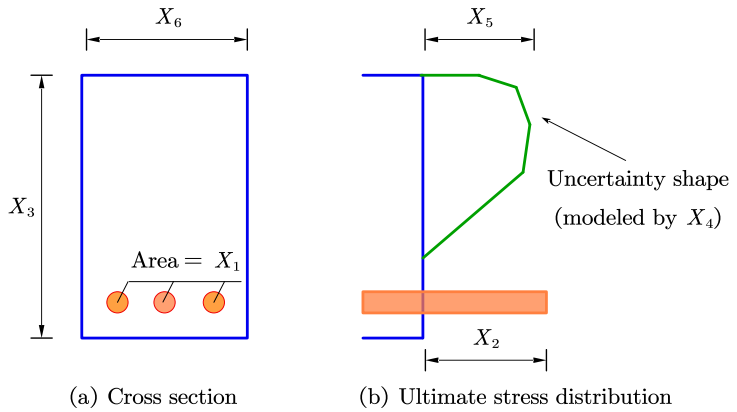


Figure 4: A nonlinear single-degree-of-freedom oscillator under a rectangular pulse load.

Table 2: Random variables for Example 2.

Variable	Description	Distribution	Mean	Standard deviation
m	Mass	Normal	1.0	0.05
k_1	Stiffness	Normal	1.0	0.10
k_2	Stiffness	Normal	0.2	0.01
r	Yield displacement	Normal	0.5	0.05
F_1	Load amplitude	Normal	0.45	0.075
t_1	Load duration	Normal	1.0	0.20



(a) Cross section (b) Ultimate stress distribution

Figure 5: Ultimate stress state of the reinforced concrete section.

Table 3: Reliability analysis results of Example 2 by several methods.

Method			N_{iter}	N_{call}	$\hat{P}_f$	$\delta_{\hat{P}_f}$
MCS	-	-	-	10^{11}	1.52×10^{-8}	2.56%
AK-MCMC	$n_a = 1$	-	176.25	185.25	1.51×10^{-8}	0.88%
BSS	$n_a = 1$	-	25.10	34.10	1.72×10^{-8}	45.63%
eAK-MCS	$n_a = 4$	-	7.95	37.80	1.55×10^{-8}	6.61%
Proposed QBALC	$n_a = 1$	$\sqrt{\bar{\rho}} = 0.25$	10.00	19.00	1.50×10^{-8}	12.63%
		$\sqrt{\bar{\rho}} = 0.50$	15.70	24.70	1.51×10^{-8}	4.13%
		$\sqrt{\bar{\rho}} = 0.75$	18.45	27.45	1.49×10^{-8}	2.57%
	$n_a = 2$	$\sqrt{\bar{\rho}} = 0.25$	6.65	21.30	1.46×10^{-8}	7.87%
		$\sqrt{\bar{\rho}} = 0.50$	9.35	26.70	1.47×10^{-8}	3.15%
		$\sqrt{\bar{\rho}} = 0.75$	11.10	30.20	1.49×10^{-8}	2.90%
	$n_a = 3$	$\sqrt{\bar{\rho}} = 0.25$	5.10	22.30	1.46×10^{-8}	8.08%
		$\sqrt{\bar{\rho}} = 0.50$	7.30	28.90	1.48×10^{-8}	3.21%
		$\sqrt{\bar{\rho}} = 0.75$	8.20	31.60	1.50×10^{-8}	1.78%
	$n_a = 4$	$\sqrt{\bar{\rho}} = 0.25$	4.45	23.80	1.51×10^{-8}	10.42%
		$\sqrt{\bar{\rho}} = 0.50$	6.30	31.20	1.50×10^{-8}	1.75%
		$\sqrt{\bar{\rho}} = 0.75$	6.95	33.80	1.50×10^{-8}	2.48%
	$n_a = 5$	$\sqrt{\bar{\rho}} = 0.25$	4.10	25.50	1.49×10^{-8}	5.07%
		$\sqrt{\bar{\rho}} = 0.50$	5.50	32.50	1.49×10^{-8}	2.15%
		$\sqrt{\bar{\rho}} = 0.75$	6.15	35.75	1.51×10^{-8}	1.59%
	$n_a = 6$	$\sqrt{\bar{\rho}} = 0.25$	4.10	28.60	1.48×10^{-8}	3.78%
		$\sqrt{\bar{\rho}} = 0.50$	4.90	33.40	1.50×10^{-8}	1.99%
		$\sqrt{\bar{\rho}} = 0.75$	5.70	38.20	1.51×10^{-8}	1.63%

Table 4: Basic random variables for Example 3.

Variable	Description	Distribution	Mean	COV
X_1	Area of reinforcement	Normal	1260 mm^2	0.05
X_2	Yield stress of reinforcement	Lognormal	300 N/mm^2	0.10
X_3	Effective depth of reinforcement	Normal	770 mm	0.05
X_4	Stress–strain factor of concrete	Lognormal	0.35	0.10
X_5	Compressive strength of concrete	Lognormal	30 N/mm^2	0.15
X_6	Width of section	Normal	400 mm	0.05
X_7	Applied bending moment	Lognormal	80 kN·m	0.20

Table 5: Reliability analysis results of Example 3 by several methods.

Method			N_{iter}	N_{call}	$\hat{P}_f$	$\delta_{\hat{P}_f}$
MCS	-	-	-	5×10^{11}	1.57×10^{-8}	1.13%
AK-MCMC	$n_a = 1$	-	143.65	152.65	1.58×10^{-8}	0.98%
BSS	$n_a = 1$	-	25.85	34.85	1.46×10^{-8}	34.88%
eAK-MCS	$n_a = 4$	-	5.60	28.40	1.56×10^{-8}	5.02%
Proposed QBALC	$n_a = 1$	$\sqrt{\bar{\rho}} = 0.25$	11.30	20.30	1.58×10^{-8}	3.86%
		$\sqrt{\bar{\rho}} = 0.50$	14.55	23.55	1.59×10^{-8}	2.79%
		$\sqrt{\bar{\rho}} = 0.75$	16.25	25.25	1.59×10^{-8}	3.30%
	$n_a = 2$	$\sqrt{\bar{\rho}} = 0.25$	7.65	23.30	1.59×10^{-8}	4.26%
		$\sqrt{\bar{\rho}} = 0.50$	8.35	24.70	1.61×10^{-8}	2.71%
		$\sqrt{\bar{\rho}} = 0.75$	9.95	27.90	1.59×10^{-8}	2.10%
	$n_a = 3$	$\sqrt{\bar{\rho}} = 0.25$	7.05	28.15	1.61×10^{-8}	2.30%
		$\sqrt{\bar{\rho}} = 0.50$	7.85	30.55	1.57×10^{-8}	2.33%
		$\sqrt{\bar{\rho}} = 0.75$	8.50	32.50	1.58×10^{-8}	1.79%
	$n_a = 4$	$\sqrt{\bar{\rho}} = 0.25$	6.15	30.60	1.58×10^{-8}	2.13%
		$\sqrt{\bar{\rho}} = 0.50$	6.55	32.20	1.57×10^{-8}	2.65%
		$\sqrt{\bar{\rho}} = 0.75$	7.25	35.00	1.55×10^{-8}	1.98%
	$n_a = 5$	$\sqrt{\bar{\rho}} = 0.25$	5.65	33.25	1.57×10^{-8}	3.18%
		$\sqrt{\bar{\rho}} = 0.50$	6.20	36.00	1.57×10^{-8}	1.77%
		$\sqrt{\bar{\rho}} = 0.75$	6.75	38.75	1.57×10^{-8}	1.88%
	$n_a = 6$	$\sqrt{\bar{\rho}} = 0.25$	5.45	36.70	1.57×10^{-8}	2.84%
		$\sqrt{\bar{\rho}} = 0.50$	5.80	38.80	1.56×10^{-8}	2.58%
		$\sqrt{\bar{\rho}} = 0.75$	6.70	44.20	1.56×10^{-8}	1.90%

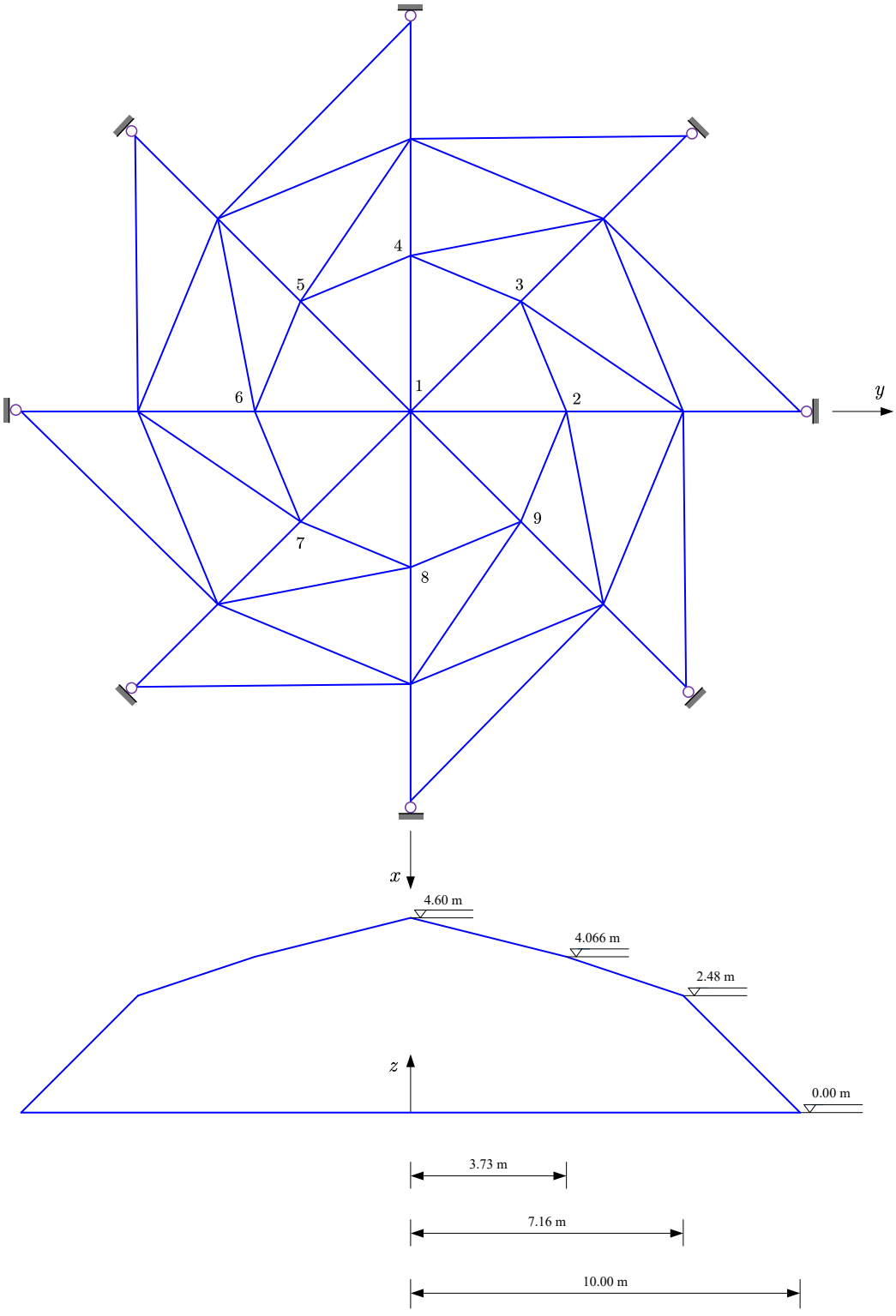


Figure 6: Schematic of a 56-bar space truss structure.

Table 6: Random variables for Example 4.

Variable	Distribution	Mean	COV
P_1	Lognormal	150 kN	0.20
$P_2, P_3, \cdots, P_9$	Lognormal	100 kN	0.20
E	Normal	2.06 GPa	0.10
A	Normal	2,000 mm^2	0.05

Table 7: Reliability analysis results of Example 4 by several methods.

Method			N_{iter}	N_{call}	$\hat{P}_f$	$\delta_{\hat{P}_f}$
IS	-	-	-	66,107	4.94×10^{-8}	1.00%
AK-MCMC	$n_a = 1$	-	456.00	465.00	4.97×10^{-8}	2.92%
BSS	$n_a = 1$	-	27.60	36.60	5.06×10^{-8}	33.49%
eAK-MCS	$n_a = 4$	-	-	-	-	-
		$\sqrt{\bar{\rho}} = 0.25$	7.15	34.60	4.86×10^{-8}	5.37%
	$n_a = 4$	$\sqrt{\bar{\rho}} = 0.50$	8.40	39.60	4.92×10^{-8}	4.77%
		$\sqrt{\bar{\rho}} = 0.75$	9.75	45.00	4.98×10^{-8}	3.21%

5. Concluding remarks

This article presents a new Bayesian active learning method, called 'Quasi-Bayesian Active Learning Cubature' (QBALC), for structural reliability analysis with extremely small failure probabilities. The method leverages the previously developed Bayesian failure probability inference framework. To avoid solving the costly exact posterior variance of the failure probability, we propose a quasi posterior variance which is cheaper to evaluate. Two critical ingredients for a Bayesian active learning method, i.e. the stopping criterion and the learning function, are then derived based on the use of the posterior mean and quasi posterior variance of the failure probability. Specifically, a stopping criterion based on the quasi posterior coefficient of variation of the failure probability is proposed and its numerical solution is developed. Furthermore, a learning function motivated by the quasi posterior variance is proposed, which itself allows multi-point selection and thus parallel distributed processing. By means of studying four numerical examples, it is empirically shown that: (1) the proposed method is able to estimate extremely small failure probabilities (in the order of 10^{-8}-10^{-9}) with a satisfactory degree of accuracy; (2) selecting multiple points at each iteration can reduce the number of iterations, and may improve the computational efficiency for expensive structural reliability analysis if parallel computing is available; (3) $\sqrt{\tilde{\rho}} = 0.50$ and $n_a = 4$ may be a good choice in practice.

The authors believe that the proposed QBALC method can be extended in many ways. First, one possible way is to incorporate some dimension techniques, making the proposed method applicable to higher dimensions. Second, the proposed method can be extended to system reliability analysis by assigning a Gaussian process prior to each component performance function instead of the composite performance function. Other directions include time-variant reliability analysis and reliability analysis under mixed uncertainties, etc.

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 mainly supported by China Scholarship Council (CSC). Alice Cicirello would like to thank the financial support provided by the Alexander von Humboldt Foundation Research Fellowship for experienced researchers. Pengfei Wei is grateful to the support from the National Natural Science Foundation of China (grant no. 51905430 and 72171194).

Data availability

Data will be made available on request.

References

[1] C. Song, R. Kawai, Monte Carlo and variance reduction methods for structural reliability analysis: A comprehensive review, Probabilistic Engineering Mechanics (2023) 103479doi:https://doi.org/10.1016/j.probengmech.2023.103479.

[2] K. W. Breitung, Asymptotic approximations for probability integrals, Springer, 2006.

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

[4] X. Zhang, M. D. Pandey, Structural reliability analysis based on the concepts of entropy, fractional moment and dimensional reduction method, Structural Safety 43 (2013) 28–40. doi:https://doi.org/10.1016/j.strusafe.2013.03.001.

[5] J. Li, J.-b. Chen, W.-l. Fan, The equivalent extreme-value event and evaluation of the structural system reliability, Structural Safety 29 (2) (2007) 112–131. doi:https://doi.org/10.1016/j.strusafe.2006.03.002.

[6] M.-Z. Lyu, J.-B. Chen, 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, Structural Safety 98 (2022) 102233. doi:https://doi.org/10.1016/j.strusafe.2022.102233.

[7] C. G. Bucher, U. Bourgund, A fast and efficient response surface approach for structural reliability problems, Structural Safety 7 (1) (1990) 57–66. doi:ShareCitehttps://doi.org/10.1016/0167-4730(90)90012-E.

[8] G. Blatman, B. Sudret, An adaptive algorithm to build up sparse polynomial chaos expansions for stochastic finite element analysis, Probabilistic Engineering Mechanics 25 (2) (2010) 183–197. doi:https://doi.org/10.1016/j.probengmech.2009.10.003.

[9] I. Kaymaz, Application of kriging method to structural reliability problems, Structural Safety 27 (2) (2005) 133–151. doi:https://doi.org/10.1016/j.strusafe.2004.09.001.

[10] B. J. Bichon, M. S. Eldred, L. P. Swiler, S. Mahadevan, J. M. McFarland, Efficient global reliability analysis for nonlinear implicit performance functions, AIAA Journal 46 (10) (2008) 2459–2468. doi:https://doi.org/10.2514/1.34321.

[11] B. Echard, N. Gayton, M. Lemaire, AK-MCS: an active learning reliability method combining Kriging and Monte Carlo simulation, Structural Safety 33 (2) (2011) 145–154. doi:https://doi.org/10.1016/j.strusafe.2011.01.002.

[12] R. Teixeira, M. Nogal, A. O'Connor, Adaptive approaches in metamodel-based reliability analysis: A review, Structural Safety 89 (2021) 102019. doi:https://doi.org/10.1016/j.strusafe.2020.102019.

[13] M. Moustapha, S. Marelli, B. Sudret, Active learning for structural reliability: Survey, general framework and benchmark, Structural Safety 96 (2022) 102174. doi:https://doi.org/10.1016/j.strusafe.2021.102174.

[14] C. Dang, P. Wei, J. Song, M. Beer, Estimation of failure probability function under imprecise probabilities by active learning–augmented probabilistic integration, ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part A: Civil Engineering 7 (4) (2021) 04021054. doi:https://doi.org/10.1061/AJRUA6.0001179.

[15] C. Dang, P. Wei, M. G. Faes, M. A. Valdebenito, M. Beer, Parallel adaptive Bayesian quadrature for rare event estimation, Reliability Engineering & System Safety 225 (2022) 108621. doi:https://doi.org/10.1016/j.ress.2022.108621.

[16] C. Dang, M. A. Valdebenito, M. G. Faes, P. Wei, M. Beer, Structural reliability analysis: A Bayesian perspective, Structural Safety 99 (2022) 102259. doi:https://doi.org/10.1016/j.strusafe.2022.102259.

[17] C. Dang, M. G. Faes, M. A. Valdebenito, P. Wei, M. Beer, Partially Bayesian active learning cubature for structural reliability analysis with extremely small failure probabilities, Computer Methods in Applied Mechanics and Engineering 422 (2024) 116828. doi:https://doi.org/10.1016/j.cma.2024.116828.

[18] C. Dang, , M. Beer, Semi-Bayesian active learning quadrature for estimating extremely low failure probabilities, Reliability Engineering & System Safety 246 (2023) 110052. doi:https://doi.org/10.1016/j.ress.2024.110052.

[19] Z. Hu, C. Dang, L. Wang, M. Beer, Parallel Bayesian probabilistic integration for structural reliability analysis with small failure probability, Structural Safety (2023). doi:https://doi.org/10.1016/j.strusafe.2023.102409.

[20] C. Dang, M. A. Valdebenito, J. Song, P. Wei, M. Beer, Estimation of small failure probabilities by partially Bayesian active learning line sampling: Theory and algorithm, Computer Methods in Applied Mechanics and Engineering 412 (2023) 116068. doi:https://doi.org/10.1016/j.cma.2023.116068.

[21] C. Dang, M. A. Valdebenito, M. G. Faes, J. Song, P. Wei, M. Beer, Structural reliability analysis by line sampling: A Bayesian active learning treatment, Structural Safety 104 (2023) 102351. doi:https://doi.org/10.1016/j.strusafe.2023.102351.

[22] C. Dang, M. A. Valdebenito, P. Wei, J. Song, M. Beer, Bayesian active learning line sampling with log-normal process for rare-event probability estimation, Reliability Engineering & System Safety 246 (2024) 110053. doi:https://doi.org/10.1016/j.ress.2024.110053.

[23] A. O'Hagan, Bayes–hermite quadrature, Journal of Statistical Planning and Inference 29 (3) (1991) 245–260. doi:https://doi.org/10.1016/0378-3758(91)90002-V.

[24] Z. Ghahramani, C. Rasmussen, Bayesian monte carlo, Advances in Neural Information Processing Systems 15 (2002).

[25] P. Wei, C. Tang, Y. Yang, Structural reliability and reliability sensitivity analysis of extremely rare failure events by combining sampling and surrogate model methods, Proceedings of the Institution of Mechanical Engineers, Part O: Journal of risk and reliability 233 (6) (2019) 943–957. doi:https://doi.org/10.1177/1748006X19844666.

[26] J. Bect, L. Li, E. Vazquez, Bayesian subset simulation, SIAM/ASA Journal on Uncertainty Quantification 5 (1) (2017) 762–786. doi:https://doi.org/10.1137/16M1078276.

[27] N. Razaaly, P. M. Congedo, Extension of AK-MCS for the efficient computation of very small failure probabilities, Reliability Engineering & System Safety 203 (2020) 107084. doi:https://doi.org/10.1016/j.ress.2020.107084.

[28] J. Zhou, A. S. Nowak, Integration formulas to evaluate functions of random variables, Structural Safety 5 (4) (1988) 267–284. doi:https://doi.org/10.1016/0167-4730(88)90028-8.

[29] C. Dang, P. Wei, M. G. Faes, M. Beer, Bayesian probabilistic propagation of hybrid uncertainties: Estimation of response expectation function, its variable importance and bounds, Computers & Structures 270 (2022) 106860. doi:https://doi.org/10.1016/j.compstruc.2022.106860.

[30] M. Moustapha, S. Marelli, B. Sudret, UQLab user manual – Active learning reliability, Tech. rep., Chair of Risk, Safety and Uncertainty Quantification, ETH Zurich, Switzerland, report UQLab-V2.0-117 (2022).