Parallel active learning XGBoost for structural reliability analysis with application to an onshore wind turbine tower

Weiwei Tang[a], Chao Dang[c,*], Jun Xu[a,b,*]

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

Abstract

Active learning methods have emerged as a powerful tool in structural reliability analysis. However, conventional approaches may still fall short in terms of efficiency, accuracy, and applicability when addressing complex real-world problems. To this end, this study develops a novel active learning method called 'parallel active learning XGBoost' (PALX). In this method, the XGBoost model is employed as a surrogate for the true performance function instead of the commonly used Kriging model, with prediction uncertainty quantified through cross-validation. By assuming that the resulting predictions follow a Gaussian process, a convenient failure probability estimator and a robust stopping criterion are introduced, which are adapted from a well-established Bayesian active learning method. The failure probability estimator and stopping criterion are numerically solved using the sequential variance-amplified importance sampling. Furthermore, a new learning function, termed 'prediction variance-weighted epistemic uncertainty contribution', is proposed for identifying the best next evaluation point. To enable parallel computing, a multi-point selection method called 'lower confidence bound believer' is developed. The effectiveness of PALX is demonstrated through three numerical examples and a practical engineering problem involving a onshore wind turbine tower. It is shown that PALX can significantly reduce computational costs without compromising accuracy, demonstrating its potential for real-world engineering challenges.

Keywords: Structural reliability analysis; Active learning; Parallel computing; Learning function; Stopping criterion; Wind turbine tower

*Corresponding authors
Email addresses: chao.dang@tu-dortmund.de (Chao Dang), xujun86@hnu.edu.cn (Jun Xu)

1. Introduction

One of the major tasks in structural reliability analysis is evaluating the complement of reliability, known as the failure probability. This involves solving a multi-dimensional probability integral, which is defined as:

$$P_f = \Pr\{g(\boldsymbol{X}) < 0\} = \int_{\Theta_{\boldsymbol{X}}} I(\boldsymbol{x}) f_{\boldsymbol{X}}(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}, \tag{1}$$

where $\Pr\{\cdot\}$ represents the probability operation; $\boldsymbol{X} = [X_1, X_2, \ldots, X_d] \in \Theta_{\boldsymbol{X}} \subseteq \mathbb{R}^d$ is a vector of d continuous random variables with support $\Theta_{\boldsymbol{X}}$; $f_{\boldsymbol{X}}(\boldsymbol{x})$ is the joint probability density function (PDF) of $\boldsymbol{X}$; $g(\boldsymbol{X})$ denotes the performance function (also known as the limit state function); $I(\boldsymbol{x})$ is the indicator function, which equals 1 if $g(\boldsymbol{x}) < 0$ and 0 otherwise.

In most real-world scenarios, obtaining an analytical solution to Eq. (1) is intractable. To overcome this obstacle, various reliability analysis methods have been developed over the past few decades. Existing methods generally fall into four categories: (i) simulation methods, including Monte Carlo simulation (MCS) [1], importance sampling (IS) [2, 3], subset simulation (SS) [4], line sampling [5] and directional simulation (DS) [6]; (ii) approximate analytical methods, such as first-order reliability method [7] and second-order reliability method [8]; (iii) moment based methods, including integer moments-based methods [9, 10] and fractional moments-based methods [11, 12]; (iv) surrogate-assisted methods, including Kriging [13], response surface [14], support vector machine [15], polynomial chaos expansion (PCE) [16].

Among recent developments, surrogate-assisted methods have received increasing attention in structural reliability analysis, particularly when integrated with active learning. Active learning reliability (ALR) methods, such as efficient global reliability analysis [17] and adaptive Kriging-MCS (AK-MCS) [18], substitute a computationally expensive performance function with a Kriging model. This statistical model, based on Gaussian processes, not only provides predictions but also quantifies uncertainty. In the ALR framework, the Kriging model is iteratively refined through a sequential experimental design, where candidate sample points are progressively added to the training dataset based on the model's prediction uncertainty. This process continues until a predefined convergence criterion is met. Active learning ensures that the surrogate model focuses on the most important regions of the random-variate space, thereby avoiding unnecessary

evaluations of the true performance function. For a comprehensive review of recent developments in ALR methods, interested readers can refer to [19, 20].

The focus of ALR methods primarily centers on the following four aspects [20]: (i) surrogate models; (ii) numerical integrators; (iii) stopping criteria; and (iv) learning functions.

- Surrogate models serve as simplified emulators of the original performance function to alleviate computational burdens. These models are primarily categorized into two types. Probabilistic surrogate models, such as Kriging [18], Gaussian process regression [21], polynomial-chaos Kriging [22] and Bayesian support vector machine (SVM) [23], can provide built-in prediction uncertainty. In contrast, deterministic surrogate models, including PCE [24] and SVM [25], offer deterministic approximations without directly quantifying prediction uncertainty. However, techniques such as cross-validation and bootstrapping can be employed to estimate uncertainty for these models.

- Numerical integrators in ALR are crucial for estimating the failure probability based on the surrogate model, while also providing a candidate pool and/or evaluating the stopping criterion or learning function. Representative methods include MCS [18], IS [26], SS [27, 28], DS [29], variance-amplified importance sampling (VAIS) [30], and hyper-shell simulation [31], among others.

- Learning functions guide the selection of the most informative points at which to evaluate the true performance function. Examples of such learning functions include expected feasibility function [17] and U function [18], expected risk function [32], least improvement function [33], and so on. Recent research has introduced new learning functions based on the posterior statistics of the failure probability from a Bayesian perspective, such as upper-bound posterior variance contribution [21, 34], right-shifted contribution [35], weighted misclassification probability [31] and weighted epistemic uncertainty contribution [36]. To enable parallel computing, several multi-point selection strategies have also been proposed, including clustering-based methods [37–39], Kriging believer method [40], and many others [31, 41–43].

- Stopping criteria determine when to terminate the active learning process and typically fall into two

categories. The first category is based on the extrema of the learning function, such as $\min(U) > 2$ [18], $\max(EFF) < 0.001$ [18] and $\max(ERF) < 0.001$ [32]. However, these criteria are not directly tied to the error of the failure-probability estimate. The second category, by contrast, directly assesses this error, including error-based stopping criterion [44], uncertainty function measure [33] and ε stopping criterion [45]. Besides, other notable stopping criteria have been developed based on the posterior statistics of the failure probability from a Bayesian perspective, such as upper bound posterior coefficient of variation (COV) of failure probability [21], relative difference between posterior mean and right-shifted posterior mean of failure probability [35], and quasi posterior COV of failure probability [42].

Despite the significant progress, there remains room for further development of ALR methods especially when addressing real-world challenges. First, most existing approaches use the Kriging model, which means that other flexible surrogate models have received insufficient attention. Second, powerful numerical integrators are still needed to solve the analytically intractable integrals involved. Third, it is crucial to establish new stopping criteria that can achieve an optimal balance between avoiding premature convergence and preventing unnecessary evaluations. Finally, further advancements in learning functions and multi-point selection strategies are still desired.

To partially address the existing research gap, this paper introduces a novel active learning method for structural reliability analysis based on the extreme gradient boosting (XGBoost) model [46]. The method is called 'parallel active learning XGBoost' (PALX), which can estimate (extremely) small failure probabilities and support parallel computing. The primary contributions of the present study are summarized as follows:

- First, to the best of the authors' knowledge, this work introduces the XGBoost model into the ALR methods for the first time. XGBoost is widely recognized in machine learning for its robust and accurate predictive capabilities, but its application in structural reliability analysis has received limited attention. By integrating with the cross-validation technique, the resulting XGBoost models can not only provide a prediction mean but also the prediction variance, thereby making it a promising tool for developing advanced active learning schemes.

- Second, we introduce a pragmatic assumption that the XGBoost models derived from cross-validation behave as a GP. This assumption enables us to develop the failure probability estimator and stopping criterion by adapting results from a recently developed Bayesian active learning method [21]. The analytically intractable integrals involved are solved using a sequential VAIS approach [30].

- Third, we propose a novel learning function, which simultaneously accounts for both the epistemic uncertainty in the failure probability and prediction variance of the XGBoost models. This enhancement facilitates more effective selection of the evaluation points and accelerates convergence.

- Fourth, a novel multi-point selection strategy called 'lower confidence bound believer' (LCBB) is proposed. The core idea of LCBB is to trust the predictions provided by the lower confidence bound, which allows for the selection of multiple informative evaluation points in each iteration.

- Fifth, unlike most existing ALR methods, which are typically validated only using simple academic examples, the proposed method is applied to a practical engineering problem—specifically, an onshore wind turbine tower. This application demonstrates the method's effectiveness and its potential benefits for real-world engineering challenges.

The remainder of this paper is organized as follows. Section 2 provides a brief review of the XGBoost model and cross-validation. The proposed PALX method is introduced in Section 3. Three numerical examples are examined in Section 4 to demonstrate the performance of the proposed method. Section 5 applies the proposed method to an onshore wind turbine tower. The paper concludes with some final remarks, which are given in Section 6.

2. Brief overview of XGBoost and cross-validation

In this section, we provide an overview of the XGBoost model that underpins our proposed method. We also introduce the cross-validation approach used to evaluate the prediction uncertainty of the XGBoost model.

2.1. XGBoost model

XGBoost is an advanced supervised algorithm proposed by Chen et al. [46] under the tree boosting framework, which constructs an ensemble of decision trees to approximate input-output relationships. By leveraging ensemble learning and gradient boosting, XGBoost achieves high predictive accuracy, capturing non-linear data patterns while mitigating overfitting through built-in regularization mechanisms. In this paper, XGBoost is employed as the surrogate model for structural reliability analysis. Based on the dataset $\mathcal{D}_n = \{(\boldsymbol{x}^{(1)}, g(\boldsymbol{x}^{(1)})), (\boldsymbol{x}^{(2)}, g(\boldsymbol{x}^{(2)})), \ldots, (\boldsymbol{x}^{(n)}, g(\boldsymbol{x}^{(n)}))\}$, the performance function $y = g(\boldsymbol{x})$ is approximated using XGBoost as follows:

$$\hat{g}_n(\boldsymbol{x}) = \sum_{k=1}^{K} f_k(\boldsymbol{x}), \quad f_k \in \mathcal{F}, \tag{2}$$

where $\hat{g}_n(\boldsymbol{x})$ represents the predicted value at input $\boldsymbol{x}$; K denotes the total number of decision trees; $\mathcal{F}$ is an ensemble model comprising a total of K trees; $f_k(\boldsymbol{x})$ corresponds to the prediction from the k-th tree. The overall prediction $\hat{g}_n(\boldsymbol{x})$ is obtained by aggregating the outputs from all individual regression trees. The optimal number of trees and the structure of each tree are determined by optimizing the objective function Γ_{obj}, which is expressed as follows:

$$\Gamma_{obj} = \sum_{i=1}^{n} l(g_n(\boldsymbol{x})_i, \hat{g}_n(\boldsymbol{x})_i) + \sum_{k=1}^{K} \Omega(f_k), \tag{3}$$

where $l(g_n(\boldsymbol{x})_i, \hat{g}_n(\boldsymbol{x})_i)$ represents the loss function that quantifies the discrepancy between predicted and true values; $\Omega(f_k)$ denotes a regularization term that penalizes model complexity to prevent overfitting, which is defined as:

$$\Omega(f_k) = \gamma V + \frac{1}{2}\delta \parallel \omega \parallel^2, \tag{4}$$

where V denotes the number of leaf nodes; ω represents the node weight; γ and δ are two constants that regulate the model's complexity to avoid overfitting. Optimizing Γ_{obj} using traditional optimization methods is generally impractical due to the complexity of the objective function and the large number of parameters involved. Thus, XGBoost employs an additive learning strategy. In this approach, the objective function

Γ_{obj} for the k-th iteration can be described as follows:

$$\begin{aligned}\Gamma_{obj}^{t} &= \sum_{i=1}^{n} l(g_n(\boldsymbol{x})_i, \hat{g}_n(\boldsymbol{x})_i^{t}) + \sum_{k=1}^{t} \Omega(f_k) \\ &= \sum_{i=1}^{n} l(g_n(\boldsymbol{x})_i, \hat{g}_n(\boldsymbol{x})_i^{t-1} + f_t(\boldsymbol{x}_i)) + \Omega(f_t) + C_0,\end{aligned} \tag{5}$$

where C_0 is a constant that represents a fixed offset in the objective function. By performing the second-order Taylor expansions, the $\hat{\Gamma}_{obj}^{t}$ can be approximated by:

$$\hat{\Gamma}_{obj}^{t} = \sum_{i=1}^{n} \left[l(g_n(\boldsymbol{x})_i, \hat{g}_n(\boldsymbol{x})_i^{t-1}) + \mathrm{g}_i f_t(\boldsymbol{x}_i) + \frac{1}{2} h_i f_t^2(\boldsymbol{x}_i) \right] + \Omega(f_t) + C_0, \tag{6}$$

where $\mathrm{g}_i = \partial_{\hat{g}_n(\boldsymbol{x})_i^{(t-1)}} l(g_n(\boldsymbol{x})_i, \hat{g}_n(\boldsymbol{x})_i^{(t-1)})$ and $h_i = \partial^2_{\hat{g}_n(\boldsymbol{x})_i^{(t-1)}} l(g_n(\boldsymbol{x})_i, \hat{g}_n(\boldsymbol{x})_i^{(t-1)})$ represent the first and second-order partial derivatives of the loss function, respectively. Since the constant C_0 has no influence on the optimization process, $\hat{\Gamma}_{obj}^{t}$ can be further reformulated as follows:

$$\hat{\Gamma}_{obj}^{t} = \sum_{i=1}^{n} \left[\mathrm{g}_i f_t(\boldsymbol{x}_i) + \frac{1}{2} h_i f_t^2(\boldsymbol{x}_i) \right] + \Omega(f_t). \tag{7}$$

The optimal model parameters for the k-th tree can be determined by optimizing the objective function $\hat{\Gamma}_{obj}^{t}$ (Eq. (7)). This optimization process continues until the predefined stopping criterion is met, after which the final predictions are obtained. In tree learning, determining the optimal split point is crucial. Both exact and approximate algorithms are used to determine the best split points among potential options, as illustrated in Fig. 1. For a comprehensive explanation of the XGBoost algorithm, please refer to [46].

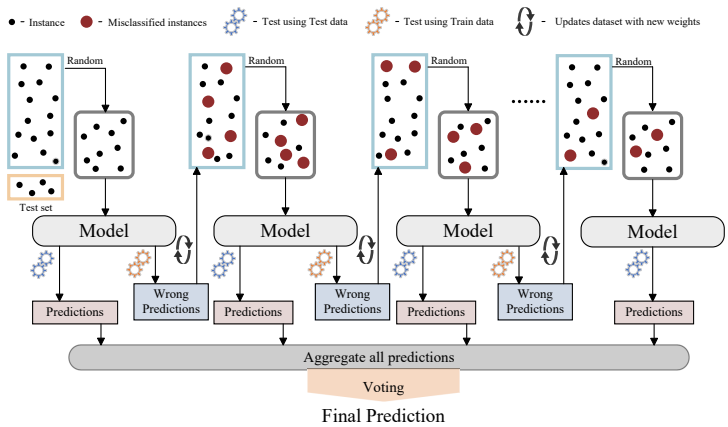


Figure 1: Framework of XGBoost algorithm.

2.2. Cross-validation for uncertainty estimation

XGBoost is effective for making predictions, but it does not directly provide prediction uncertainty. To address this limitation, k-fold cross-validation can be used, which is crucial for informed decision-making during the active learning process. In this procedure, the training dataset is divided into k roughly equal-sized subsets. One subset is used as the validation set, while the remaining $k-1$ are used for training. This process is repeated k times, with each subset serving as the validation set exactly once, ensuring that the model is tested on all points. To fully capture prediction uncertainty and reduce potential biases due to data partitioning, multiple rounds of k-fold cross-validation are employed.

By averaging the results from these iterations, the prediction mean and standard deviation functions can be obtained as follows:

$$m_{\hat{g}_n}(\boldsymbol{x}) = \frac{1}{M}\sum_{m=1}^{M}\left[\frac{1}{k}\sum_{q=1}^{k}\hat{g}_n(\boldsymbol{x})_q^{(m)}\right], \tag{8}$$

$$\sigma_{\hat{g}_n}(\boldsymbol{x}) = \sqrt{\frac{1}{M}\sum_{m=1}^{M}\left[\frac{1}{k}\sum_{p=1}^{k}\left(\hat{g}_n(\boldsymbol{x})_p^{(m)} - \frac{1}{k}\sum_{q=1}^{k}\hat{g}_n(\boldsymbol{x})_q^{(m)}\right)^2\right]}, \tag{9}$$

where $m_{\hat{g}_n}(\boldsymbol{x})$ denotes the predicted mean of $\hat{g}(\boldsymbol{x})$; $\sigma_{\hat{g}_n}(\boldsymbol{x})$ represents the predicted standard deviation; M is the number of cross-validation rounds. To ensure robustness while maintaining computational efficiency, and in line with common practices in the literature [40, 47], we set $k = 6$, $M = 5$. Fig. 2 illustrates the 6-fold cross-validation process. We identified three key parameters in XGBoost that are crucial for this study: the number of trees (K), maximum tree depth, and learning rate [48]. Based on empirical testing and considerations of computational constraints, we defined their search ranges as K ranging from 10 to 100, maximum depth within the range 3 to 10, and learning rate ranging from 0.01 to 0.3. These parameters were optimized by minimizing the objective function, and two widely used evaluation metrics—mean square error (MSE) and coefficient of determination (R^2)—were employed. For parameter selection, an exhaustive grid search was conducted over a subset of possible values. After repeating the training-validation procedure, we found that a consistent set of model parameters delivered strong and reliable performance across different cases. Consequently, the following parameters were selected: $K = 30$, the maximum depth is 5 and the learning rate is 0.05. Other parameters were set to their default values (e.g., Minimum child weight=1 and

L1 regularization=0).

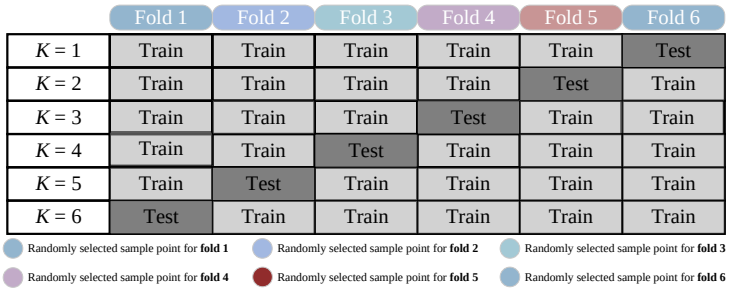


Figure 2: Framework of 6-fold cross-validation.

3. Parallel Active Learning XGBoost

This section introduces the proposed PALX method for structural reliability analysis, which builds upon the XGBoost model using k-fold cross-validation. In Section 3.1, a brief overview of the proposed method is provided, followed by the definition of the failure probability estimator in Section 3.2. Section 3.3 introduces the stopping criterion and its numerical treatment. A new learning function and a multi-point selection method are presented in Section 3.4. Finally, Section 3.5 summarizes the implementation procedure of the proposed method.

3.1. Overview of the PALX method

The core idea of PALX is to accelerate the active learning process by strategically selecting multiple informative evaluation points based on the XGBoost model with k-fold cross-validation. Initially, a small training dataset is generated to construct a preliminary XGBoost surrogate model of the performance function. k-fold cross-validation is then employed to assess the prediction uncertainty. In subsequent iterations, multiple informative evaluation points are selected using a learning function to enrich the training dataset, and the model is updated. This iterative refinement continues until a stopping criterion is satisfied.

It should be mentioned that all the subsequent developments are based on the assumption that aggregating predictions from an XGBoost model using k-fold cross-validation follow a Gaussian process (GP) $\mathcal{GP}(m_{\hat{g}_n}(\boldsymbol{x}), \sigma^2_{\hat{g}_n}(\boldsymbol{x}))$. Although this Gaussian assumption is not theoretically guaranteed, our computational

experiments suggest that it is generally reasonable. Moreover, adopting this assumption facilitates the development of a novel active learning scheme.

3.2. Failure probability estimator

In existing Bayesian active learning reliability methods (e.g., [21, 30, 34]), the posterior mean of the failure probability is used as its estimator. This idea can be adapted to our context by resorting to the assumption that predictions from cross-validated XGBoost models behave as a GP. The resulting failure probability estimator is given by:

$$m_{\hat{P}_{f,n}} = \int_{\Theta_{\boldsymbol{X}}} \Phi\left(-\frac{m_{\hat{g}_n}(\boldsymbol{x})}{\sigma_{\hat{g}_n}(\boldsymbol{x})}\right) f_{\boldsymbol{X}}(\boldsymbol{x})\mathrm{d}\boldsymbol{x}, \tag{10}$$

where Φ is the cumulative distribution function (CDF) of the standard normal distribution. Under the GP assumption, $m_{\hat{P}_{f,n}}$ represents the mean value of the failure probability. Instead of $m_{\hat{P}_{f,n}}$, a more straightforward estimator for the failure probability can be formulated by simply replacing the true performance function with its prediction mean $m_{\hat{g}_n}(\boldsymbol{x})$. However, using $m_{\hat{P}_{f,n}}$ enables the development of a stopping criterion with clear physical interpretation and incurs minimal additional computational cost. Note that the failure probability estimator $m_{\hat{P}_{f,n}}$ entails numerical integration, which will be introduced in the next subsection.

3.3. Stopping criterion and its numerical solution

The stopping criterion determines when to terminate the active learning process by assessing whether the failure probability estimate has reached a desired level of accuracy. In [21], a robust stopping criterion is developed using the upper bound of the posterior COV of the failure probability. Under the GP assumption, this approach can be conveniently adapted to our PALX method. The stopping criterion is expressed as follows:

$$\overline{\delta}_{\hat{P}_{f,n}} = \frac{\sigma_{\hat{P}_{f,n}}}{m_{\hat{P}_{f,n}}} < \epsilon, \tag{11}$$

where $\overline{\delta}_{\hat{P}_{f,n}}$ can be interpreted as the upper-bound of the COV of the failure probability; $\sigma_{\hat{P}_{f,n}}$ represents the upper-bound of the standard deviation of the failure probability; ϵ is a user-specified threshold. The

expression for $\sigma_{\hat{P}_{f,n}}$ is given by the following integral:

$$\sigma_{\hat{P}_{f,n}} = \int_{\Theta_{\boldsymbol{X}}} \sqrt{\Phi\left(-\frac{m_{\hat{g}_n}(\boldsymbol{x})}{\sigma_{\hat{g}_n}(\boldsymbol{x})}\right)\Phi\left(\frac{m_{\hat{g}_n}(\boldsymbol{x})}{\sigma_{\hat{g}_n}(\boldsymbol{x})}\right)} f_{\boldsymbol{X}}(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}, \tag{12}$$

which can be viewed as an uncertainty measure for the failure probability estimator $m_{\hat{P}_{f,n}}$. A straightforward approach to estimate $m_{\hat{P}_{f,n}}$ and $\sigma_{\hat{P}_{f,n}}$ is the crude MCS. However, in problems involving small failure probabilities, obtaining an acceptable level of accuracy would require prohibitively large computational cost, rendering this approach impractical. To balance accuracy and efficiency, this paper employs the VAIS method, originally developed in [30], and implements it in a sequential manner. The VAIS estimators of $m_{\hat{P}_{f,n}}$ and $\sigma_{\hat{P}_{f,n}}$ can be given by:

$$\hat{m}_{\hat{P}_{f,n}} = \frac{1}{N}\sum_{i=1}^{N}\left[\Phi\left(-\frac{m_{\hat{g}_n}(\boldsymbol{x}^{(i)})}{\sigma_{\hat{g}_n}(\boldsymbol{x}^{(i)})}\right)\frac{f_{\boldsymbol{X}}(\boldsymbol{x}^{(i)})}{h(\boldsymbol{x}^{(i)})}\right], \tag{13}$$

$$\hat{\sigma}_{\hat{P}_{f,n}} = \frac{1}{N}\sum_{i=1}^{N}\sqrt{\Phi\left(-\frac{m_{\hat{g}_n}(\boldsymbol{x}^{(i)})}{\sigma_{\hat{g}_n}(\boldsymbol{x}^{(i)})}\right)\Phi\left(\frac{m_{\hat{g}_n}(\boldsymbol{x}^{(i)})}{\sigma_{\hat{g}_n}(\boldsymbol{x}^{(i)})}\right)}\frac{f_{\boldsymbol{X}}(\boldsymbol{x}^{(i)})}{h(\boldsymbol{x}^{(i)})}, \tag{14}$$

where $\{\boldsymbol{x}^{(i)}\}_{i=1}^{N}$ denotes a set of N random samples generated according to $h(\boldsymbol{x})$, which represents the importance sampling density (ISD). The ISD is constructed by amplifying the standard deviations $\boldsymbol{\sigma}_{\boldsymbol{X}}$ of $\boldsymbol{X}$, while keeping the means $\boldsymbol{m}_{\boldsymbol{X}}$ unchanged. Consequently, $h_{\boldsymbol{X}}(\boldsymbol{x})$ is defined as $f_{\boldsymbol{X}}(\boldsymbol{x};\boldsymbol{m}_{\boldsymbol{X}},\alpha\boldsymbol{\sigma}_{\boldsymbol{X}})$, where $\alpha \geq 1$ is the amplification factor for the standard deviations. Notably, if X_i follows a uniform distribution, amplification of the standard deviations is not necessary. The variances of the estimators $\hat{m}_{P_{f,n}}$ and $\hat{\sigma}_{P_{f,n}}$ are expressed as follows:

$$\mathbb{V}\left[\hat{m}_{\hat{P}_{f,n}}\right] = \frac{1}{N-1}\left\{\frac{1}{N}\sum_{i=1}^{N}\left[\Phi\left(-\frac{m_{\hat{g}_n}(\boldsymbol{x}^{(i)})}{\sigma_{\hat{g}_n}(\boldsymbol{x}^{(i)})}\right)\frac{f_{\boldsymbol{X}}(\boldsymbol{x}^{(i)})}{h(\boldsymbol{x}^{(i)})}\right]^2 - \hat{m}^2_{\hat{P}_{f,n}}\right\}, \tag{15}$$

$$\mathbb{V}\left[\hat{\sigma}_{\hat{P}_{f,n}}\right] = \frac{1}{N-1}\left\{\frac{1}{N}\sum_{i=1}^{N}\left[\sqrt{\Phi\left(-\frac{m_{\hat{g}_n}(\boldsymbol{x}^{(i)})}{\sigma_{\hat{g}_n}(\boldsymbol{x}^{(i)})}\right)\Phi\left(\frac{m_{\hat{g}_n}(\boldsymbol{x}^{(i)})}{\sigma_{\hat{g}_n}(\boldsymbol{x}^{(i)})}\right)}\frac{f_{\boldsymbol{X}}(\boldsymbol{x}^{(i)})}{h(\boldsymbol{x}^{(i)})}\right]^2 - \hat{\sigma}^2_{\hat{P}_{f,n}}\right\}, \tag{16}$$

where $\mathbb{V}[\cdot]$ denotes the variance of the argument.

Determining the appropriate sample sizes for $\hat{m}_{\hat{P}_{f,n}}$ and $\hat{\sigma}_{\hat{P}_{f,n}}$ is crucial. To balance accuracy and computational cost, the sample size should be gradually increased. As outlined below, we assume that the same sample size N_0 for each enrichment. At the j step, $\{\boldsymbol{x}^{(i)}\}_{i=1}^{N_0}$ samples are generated from $h(\boldsymbol{x})$. For

each sample $\boldsymbol{x}^{(i)}$, let $\xi^{(i)}$ and $\delta^{(i)}$ be defined as follows:

$$\xi^{(i)} = \Phi\left(-\frac{m_{\hat{g}_n}(\boldsymbol{x}^{(i)})}{\sigma_{\hat{g}_n}(\boldsymbol{x}^{(i)})}\right), \tag{17}$$

$$\delta^{(i)} = \frac{f_{\boldsymbol{X}}(\boldsymbol{x}^{(i)})}{h(\boldsymbol{x}^{(i)})}. \tag{18}$$

Following this, we proceed to evaluate the four quantities listed below:

$$m^{(j)} = \frac{1}{N_0}\sum_{i=1}^{N_0}\xi^{(i)}\delta^{(i)}, \tag{19}$$

$$\sigma^{(j)} = \frac{1}{N_0}\sum_{i=1}^{N_0}\sqrt{\xi^{(i)}(1-\xi^{(i)})}\delta^{(i)}, \tag{20}$$

$$\psi^{(j)} = \frac{1}{N_0}\sum_{i=1}^{N_0}\left[\xi^{(i)}\delta^{(i)}\right]^2, \tag{21}$$

$$\lambda^{(j)} = \frac{1}{N_0}\sum_{i=1}^{N_0}\left[\sqrt{\xi^{(i)}(1-\xi^{(i)})}\delta^{(i)}\right]^2. \tag{22}$$

Subsequently, the estimators for $\hat{m}_{\hat{P}_{f,n}}$ and $\hat{\sigma}_{\hat{P}_{f,n}}$, along with their corresponding variances given in Eqs. (13)-(16), are reformulated as:

$$\hat{m}_{\hat{P}_{f,n}} = \frac{1}{j}\sum_{t=1}^{j} m^{(t)}, \tag{23}$$

$$\hat{\sigma}_{\hat{P}_{f,n}} = \frac{1}{j}\sum_{t=1}^{j} \sigma^{(t)}, \tag{24}$$

$$\mathbb{V}\left[\hat{m}_{\hat{P}_{f,n}}\right] = \frac{1}{jN_0-1}\left[\frac{1}{j}\sum_{t=1}^{j}\psi^{(t)} - \hat{m}^2_{\hat{P}_{f,n}}\right], \tag{25}$$

$$\mathbb{V}\left[\hat{\sigma}_{\hat{P}_{f,n}}\right] = \frac{1}{jN_0-1}\left[\frac{1}{j}\sum_{t=1}^{j}\lambda^{(t)} - \hat{\sigma}^2_{\hat{P}_{f,n}}\right]. \tag{26}$$

The sequential sampling process continues until the conditions $\sqrt{\mathbb{V}\left[\hat{m}_{\hat{P}_{f,n}}\right]}/\hat{m}_{\hat{P}_{f,n}} < \tau_1$ and $\sqrt{\mathbb{V}\left[\hat{\sigma}_{\hat{P}_{f,n}}\right]/\hat{\sigma}_{\hat{P}_{f,n}}} < \tau_2$ are met, where τ_1 and τ_2 are two user-specified tolerances.

In the stopping criterion outlined in Eq. (11), the terms $m_{\hat{P}_{f,n}}$ and $\sigma_{\hat{P}_{f,n}}$ are replaced by the final estimates $\hat{m}_{\hat{P}_{f,n}}$ and $\hat{\sigma}_{\hat{P}_{f,n}}$. Given that $\hat{m}_{\hat{P}_{f,n}}$ and $\hat{\sigma}_{\hat{P}_{f,n}}$ may each have some degree of error depending on τ_1 and τ_2, the stopping criterion must be met twice in a row to prevent fake convergence.

3.4. Learning function and multi-point selection

By evaluating the informativeness of each candidate point, a learning function helps in identifying one or multiple informative points to evaluate the true performance function, thereby accelerating the learning process. To enhance the computational efficiency through parallel computing, an effective multi-point selection strategy is also crucial.

In this study, we propose a new learning function called 'prediction variance-weighted epistemic uncertainty contribution' (PVWEUC), which is expressed as:

$$\mathrm{PVWEUC}_n(\boldsymbol{x}) = \underbrace{\sigma^2_{\hat{g}_n}(\boldsymbol{x})}_{①} \underbrace{\sqrt{\Phi\left(-\frac{m_{\hat{g}_n}(\boldsymbol{x})}{\sigma_{\hat{g}_n}(\boldsymbol{x})}\right)\Phi\left(\frac{m_{\hat{g}_n}(\boldsymbol{x})}{\sigma_{\hat{g}_n}(\boldsymbol{x})}\right)} f_{\boldsymbol{X}}(\boldsymbol{x})}_{②}, \tag{27}$$

where term ② is the integrand of the upper bound of the standard deviation of the failure probability, referred to as the epistemic uncertainty contribution; term ① is the prediction variance of g, serving as the weight of term ②. The learning function is maximized when $m_{\hat{g}_n}(\boldsymbol{x})$ is near to zero, $\sigma_{\hat{g}_n}(\boldsymbol{x})$ is high and $f_{\boldsymbol{X}}(\boldsymbol{x})$ is large. As a result, the larger the values of the PVWEUC function, the more promising are the corresponding candidate points.

In addition to the PVWEUC function, we also propose a new multi-point selection method called 'lower confidence bound believer' (LCBB), which is inspired by the Kriging believer in the context of parallel Bayesian optimization [49]. The key idea behind LCBB is to select a batch of points by applying a lower confidence bound criterion. Suppose that we are at the beginning of a new iteration with training data size n and want to identify another n_a evaluation points. The first point $\boldsymbol{x}^{(n+1)}$ is simply determined by $\boldsymbol{x}^{(n+1)} = \arg\max_{\boldsymbol{x}\in\Theta_{\boldsymbol{X}}} \mathrm{PVWEUC}_n(\boldsymbol{x})$. Instead of directly computing the true g function value, we adopt its lower confidence bound value, i.e., $\hat{y}^{(n+1)} = m_{\hat{g}_n}(\boldsymbol{x}^{(n+1)}) - b\sigma_{\hat{g}_n}(\boldsymbol{x}^{(n+1)})$. Subsequently, $\boldsymbol{x}^{(n+1)}$ and $\hat{y}^{(n+1)}$ are added to the dataset $\mathcal{D}_n$ ($\hat{y}^{(n+1)}$ is temporarily added). XGBoost models are trained on the enriched data with k-fold cross-validation to obtain the updated prediction mean $m_{\hat{g}_{n+1}}(\boldsymbol{x})$ and standard deviation $\sigma_{\hat{g}_{n+1}}(\boldsymbol{x})$ for g, hence also the learning function $\mathrm{PVWEUC}_{n+1}(\boldsymbol{x})$. The second point $\boldsymbol{x}^{(n+2)}$ can be identified using the updated learning function. The process is repeated until the desired n_a points have been selected. Finally, the true g function values at $\left\{\boldsymbol{x}^{(n+l)}\right\}_{l=1}^{n_a}$ are then evaluated in parallel, and the corresponding

entries in the training dataset are replaced accordingly.

Compared to the traditional Kriging believer method [49], the primary difference in LCBB lies in the consideration of prediction uncertainty. Compared to the prediction mean, the lower confidence bound provides a more conservative estimate of the g-function value. Thus, LCBB allows us to avoid overconfidence in the prediction mean, especially when the prediction uncertainty is large, and to select more informative evaluation points.

3.5. Implementation of the proposed PALX method

The implementation procedure for the proposed PALX method is outlined below and is illustrated with a flowchart provided in Fig. 3.

Step 1: Generate the initial training dataset

Generate a small set of uniformly distributed samples $\boldsymbol{\mathcal{X}} = [\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \ldots, \boldsymbol{x}^{(n_0)}]^\top$ within a d-dimensional hyper-rectangle $\Delta_0 = \prod_{i=1}^{d}[L_i, U_i]$ in $\Theta_{\boldsymbol{X}}$ using the Hammersley sequence. The lower and upper bounds L_i and U_i in the i-th dimension can be specified by: $L_i = F_{X_i}^{-1}(v_0)$ and $U_i = F_{X_i}^{-1}(1 - v_0)$, where F_{X_i} represents the marginal CDF of X_i and v_0 is a small truncation probability. These samples are then evaluated on the g-function in parallel to obtain the corresponding output values $\boldsymbol{\mathcal{Y}} = \left[y^{(1)}, y^{(2)}, \cdots, y^{(n_0)}\right]^\top$. Finally, form the initial training dataset as $\boldsymbol{\mathcal{D}}_n = \{\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{Y}}\}$ and set $n = n_0$.

Step 2: Train the XGBoost models

In this step, multiple XGBoost surrogate models are generated through multiple rounds of 6-fold cross-validation based on the training dataset $\boldsymbol{\mathcal{D}}_n$, which provides the prediction mean and standard deviation. This process is implemented using the XGBoost library integrated into MATLAB.

Step 3: Compute the statistics of the failure probability

At this stage, the failure probability estimate $\hat{m}_{\hat{P}_{f,n}}$ and the estimate of an upper bound on the standard deviation $\hat{\sigma}_{\hat{P}_{f,n}}$ need to be computed using the sequential VAIS method. This method iteratively refines the estimates until the convergence criteria $\sqrt{\mathbb{V}\left[\hat{m}_{\hat{P}_{f,n}}\right]}/\hat{m}_{\hat{P}_{f,n}} < \tau_1$ and $\sqrt{\mathbb{V}\left[\hat{\sigma}_{\hat{P}_{f,n}}\right]/\hat{\sigma}_{\hat{P}_{f,n}}} < \tau_2$ are met. Then, proceed to the next step. Refer to Section 3.3 for a detailed description of the sequential VAIS method.

Step 4: Check the stopping criterion

If $\frac{\hat{\sigma}_{\hat{P}_{f,n}}}{\hat{m}_{\hat{P}_{f,n}}} < \epsilon$ is satisfied in two consecutive iterations, proceed to **Step 6**. If not, go to **Step 5**.

Step 5: Enrich the training dataset

Initialize the parameter $l = 1$.

Step 5.1: Generate the first candidate point $\boldsymbol{x}^{(n+l)}$ by optimizing the learning function using the genetic algorithm. This algorithm searches for optimal points within another hyper-rectangle Δ_1, which is specified similarly to Δ_0, but with v_0 being replaced by v_1.

Step 5.2: Obtain the predicted value $\hat{y}^{(n+l)}$ from the trained surrogate models, where $\hat{y}^{(n+l)} = m_{\hat{g}_{(n+l)}}(\boldsymbol{x}^{(n+l)}) - b\sigma_{\hat{g}_{(n+l)}}(\boldsymbol{x}^{(n+l)})$. This predicted value is then used as the temporary g-function value for the point $\boldsymbol{x}^{(n+l)}$.

Step 5.3: Add the pair $(\boldsymbol{x}^{(n+l)}, \hat{y}^{(n+l)})$ to $\boldsymbol{\mathcal{D}}_n$ and then calibrate XGBoost models using 6-fold cross-validation based on the enriched dataset. If $l = n_a$, the multi-point selection process ends. Otherwise, return to Step 5.1 and let $l = l + 1$.

Step 5.4: After obtaining n_a points $\boldsymbol{\mathcal{X}}^+ = \{\boldsymbol{x}^{(n+l)}\}_{l=1}^{n_a}$, evaluate the g-function in parallel at $\boldsymbol{\mathcal{X}}^+$ to obtain the corresponding responses $\boldsymbol{\mathcal{Y}}^+ = \{y^{(n+l)}\}_{l=1}^{n_a}$. let $\boldsymbol{\mathcal{D}}^+ = \{\boldsymbol{\mathcal{X}}^+, \boldsymbol{\mathcal{Y}}^+\}$, and update the dataset $\boldsymbol{\mathcal{D}}_n$ by $\boldsymbol{\mathcal{D}}_n = \boldsymbol{\mathcal{D}}_n \cup \boldsymbol{\mathcal{D}}^+$. Finally, set $n = n + n_a$ and return to **Step 2**.

Step 6: End the method

The proposed method concludes, and the final failure probability estimate $\hat{m}_{\hat{P}_{f,n}}$ is returned.

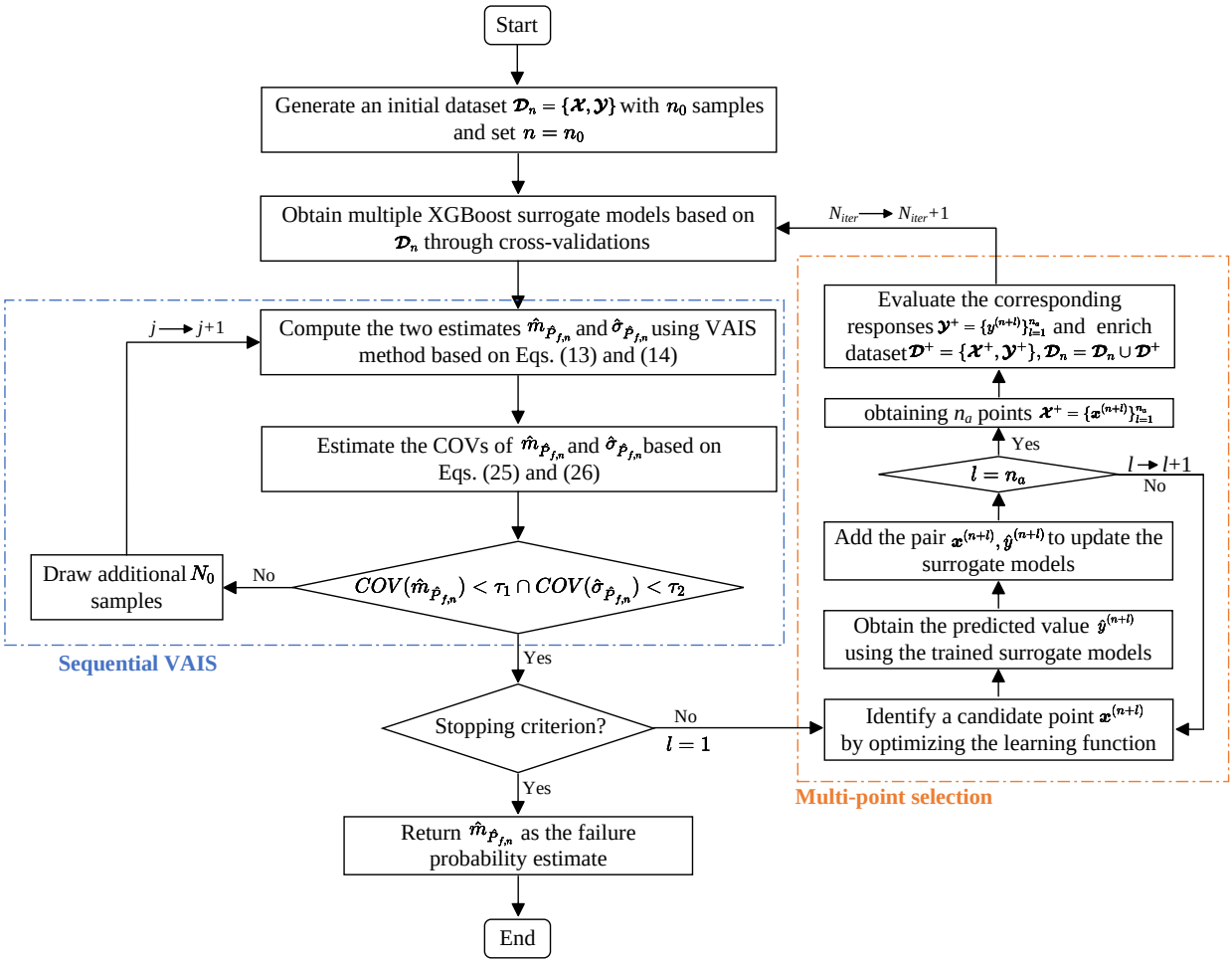


Figure 3: Flowchart of the proposed PALX method.

4. Numerical examples

In this section, the effectiveness of the proposed PALX method is demonstrated through three numerical examples of varying complexity. The parameters for the proposed method are set as follows: $n_0 = 12$, $\alpha = 2$, $b = 1$, $N_0 = 10^6$, $v_0 = 1 \times 10^{-8}$, $v_1 = 1 \times 10^{-10}$, $\tau_1 = 2\%$, $\tau_2 = 5\%$, and $\epsilon = 0.15$. For comparison, several existing non-parallel methods (i.e., AK-MCS [18], PBALC [35], ALK-KDE-IS [50], AK-MCMC [51]) and parallel methods (i.e., QBALC [42], SBALQ [31]) from the literature are also conducted.

In these reliability methods, n_a represents the number of points added in each iteration. Multiple values of n_a are tested for the proposed method, and its effects on the results are systematically analyzed. For each

comparison method, a single value of n_a is used for evaluation. Each method is independently executed 20 times, and all statistical results—including mean values and COVs—are computed based on these repeated runs. The efficiency and accuracy are assessed using four metrics: the average number of iterations N_{iter}, the average number of g-function calls N_{call}, the average failure probability $\hat{P}_f$ and its relative error $\delta_{\hat{P}_f}$ with respect to the reference failure probability. To further evaluate robustness, the means and COVs of N_{call} are calculated. For all methods except MCS and IS, the COVs of $\hat{P}_f$ are also computed from these 20 repetitions. When applicable, a crude MCS with a large sample size is performed to obtain a reference failure probability.

4.1. Example 1: Four-branch series system

The first example involves a series system comprising four branches [18, 31], which has been extensively used in various studies. The performance function is given by:

$$\mathrm{g}\left(\boldsymbol{X}\right) = \min \left\{ \begin{array}{c} a + 0.1(X_1 - X_2)^2 - \frac{(X_1+X_2)}{\sqrt{2}} \\ a + 0.1(X_1 - X_2)^2 + \frac{(X_1+X_2)}{\sqrt{2}} \\ X_1 - X_2 + \frac{b}{\sqrt{2}} \\ X_2 - X_1 + \frac{b}{\sqrt{2}} \end{array} \right\}, \tag{28}$$

where X_1 and X_2 are two independent standard normal variables; a and b are two constant parameters, specified as $a = 6$ and $b = 12$.

The proposed PALX method is compared to several other parallel and non-parallel methods, as listed in Table 1. The reference failure probability, obtained through the MCS method with 10^{12} samples, is 3.01×10^{-9} with a COV of 1.82%. In the non-parallel case ($n_a = 1$), the proposed PALX method requires an average of 24.1 iterations and 35.1 function calls, resulting in a mean failure probability of 3.03×10^{-9} with a COV of 3.07%. Compared to non-parallel methods like ALK-KDE-IS (with an average of 75.1 iterations) and PBALC (with an average of 39.1 iterations), the proposed method demonstrates competitive performance in terms of the number of iterations. In the parallel case (e.g., $n_a = 4$), the PALX method achieves a mean failure probability of 3.01×10^{-9} with a relative error $\delta_{\hat{P}_f}$ of 0.00% and a COV of 1.76%, while requiring only 9.6 iterations and 46.4 function calls on average. In contrast, the SBALQ and QBALC methods produce mean failure probability estimates with errors below 1% and COVs under 5%, but they require slightly more

iterations and function calls, indicating the superior performance of the proposed method. Additionally, it is evident for the proposed method that N_{call} increases with n_a, while the average number of iterations N_{iter} decreases with n_a until $n_a = 7$, after which N_{iter} slightly increases. This observation suggests that choosing an excessive number of points in each iteration might not necessarily reduce the total number of iterations required.

To visually illustrate the proposed method, Fig. 4 shows the points identified at each iteration of the PALX method ($n_a = 4$) along with the true limit-state curve. In the first iteration, the initial points are evenly distributed in the safe domain. Most added points from the active learning phase approach to the four key regions of the true limit state curve.

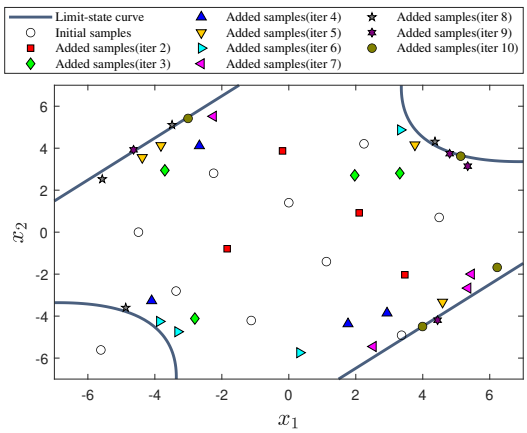


Figure 4: Selected points by the PALX method ($n_a = 4$) for Example 1.

4.2. Example 2: Liquid hydrogen tank

As a second example, we analyze the reliability of liquid hydrogen fuel tanks for space launch vehicles [52]. The tank employs a honeycomb sandwich structure, with the top and bottom plates made from aluminum alloy AL2024 and the core composed of Hexcell 1/8-in.-5052.0015. It is segmented longitudinally into ten sections, each further divided into four panels, as illustrated in Fig. 5. The pressure on the fuel tank results from air pressure, head pressure, axial force due to acceleration, and the bending and shear stress caused by the fuel weight. The tank is vulnerable to failure in three ways: von Mises strength, isotropic strength, or

Table 1: Reliability analysis results of Example 1.

Method		N_{iter}	N_{call}		$\hat{P}_f$		$\delta_{\hat{P}_f}$ (%)	Reference
			Mean	COV (%)	Mean ($\times 10^{-9}$)	COV (%)		
MCS		-	10^{12}	-	3.01	1.82	-	[35]
ALK-KDE-IS	$n_a = 1$	75.1	84.1	-	3.03	0.55	0.67	[35]
PBALC1 ($\epsilon_1 = 2.5\%$)	$n_a = 1$	35.8	44.8	-	3.04	3.82	1.00	[35]
PBALC2 ($\epsilon_2 = 2.5\%$)	$n_a = 1$	41.1	50.1	-	3.04	1.39	1.00	[35]
PBALC3 ($\epsilon_3 = 5\%$)	$n_a = 1$	40.5	49.5	-	3.03	1.99	0.67	[35]
SBALQ($\epsilon = 2\%$)	$n_a = 4$	12.8	57.0	-	3.02	0.80	0.33	[31]
QBALC($\sqrt{\bar{\rho}} = 0.50$)	$n_a = 4$	13.1	58.4	-	3.03	1.50	0.67	[42]
	$n_a = 1$	24.1	35.1	3.5	3.03	3.07	0.67	-
	$n_a = 2$	13.9	37.9	7.1	3.02	2.43	0.33	-
	$n_a = 3$	10.7	41.2	6.2	3.04	1.39	1.00	-
Proposed PALX	$n_a = 4$	9.6	46.4	4.8	3.01	1.76	0.00	-
	$n_a = 5$	8.7	50.6	9.1	2.99	1.69	0.67	-
	$n_a = 6$	8.4	56.3	9.4	3.00	1.90	0.33	-
	$n_a = 7$	8.5	64.6	10.2	3.02	1.58	0.33	-
	$n_a = 8$	8.4	71.2	8.7	3.01	2.15	0.00	-

honeycomb buckling. The limit state function can be formulated as follows:

$$\mathrm{g}(\boldsymbol{X}) = \min \left\{ \begin{array}{c} \dfrac{84000 t_{plate}}{\sqrt{N_x^2 + N_y^2 - N_x N_y + 3N_{xy}^2}} - 1 \\ \dfrac{84000 t_{plate}}{|N_y|} - 1 \\ 0.847 + 0.96X_1 + 0.986X_2 - 0.216X_3 + 0.077X_1^2 + 0.11X_2^2 \\ +0.007X_3^2 + 0.378X_1X_2 - 0.106X_1X_3 - 0.11X_2X_3 \end{array} \right\}, \tag{29}$$

where X_1, X_2, X_3 are defined as: $X_1 = 4(t_{plate} - 0.075), X_2 = 20(t_{nc} - 0.1), X_3 = -6000\left(\frac{1}{N_{xy}} + 0.003\right)$. The random variables, including the thickness of the plate (t_{plate}), the thickness of the honeycomb (t_{nc}), and

the loads (N_x, N_y, N_{xy}) on the tank, are listed in Table 2.

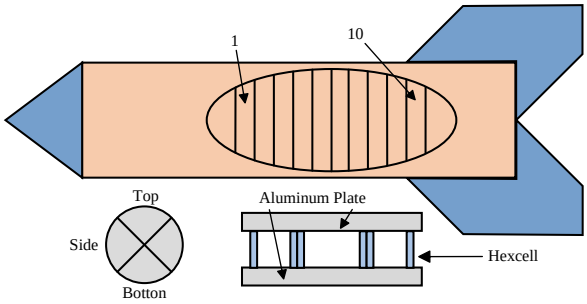


Figure 5: Schematic of liquid hydrogen tank.

Table 2: Distribution types and parameters of the random variables in Example 2.

Random variables	Distribution	Mean	COV
t_{plate}	Normal	0.07	0.10
t_{nc}	Normal	0.10	0.10
N_x	Lognormal	13	0.10
N_y	Lognormal	3051	0.10
N_{xy}	Lognormal	404	0.10

Table 3 presents a comparison of the reliability analysis results obtained by various methods. The failure probability of 4.07×10^{-5} with a COV of 0.54%, provided by MCS, is adopted as the reference result. In the sequential case ($n_a = 1$), the proposed PALX method, on average, requires 39.2 iterations and 50.2 function calls, achieving a failure probability mean estimate with a relative error of 0.74% and a COV of 2.72%. In contrast, AK-MCS demands over 240 iterations on average and exhibits a significantly larger error of approximately 4%, regardless of whether the U or EFF learning function is used. Similarly, although the PBALC2 method achieves a low relative error of 0.25%, it requires an average of 63.9 iterations—substantially more than that of PALX. For the parallel case $n_a = 4$, the proposed PALX method completes in an average of 17.4 iterations and 77.4 function calls, while maintaining a relative error of 0.25% and a COV of 2.57%. By comparison, the QBALC method requires 25.6 iterations on average. As n_a increases from 1 to 8, PALX

consistently maintains an error below 1.0% and a low COV, while notably reducing the average number of iterations required.

Table 3: Reliability analysis results of Example 2.

Method		N_{iter}	N_{call}		$\hat{P}_f$		$\delta_{\hat{P}_f}$ (%)
			Mean	COV (%)	Mean ($\times 10^{-5}$)	COV (%)	
MCS		-	10^9	-	4.07	0.54	-
AK-MCS-U	$n_a = 1$	242.1	253.1	24.9	4.26	4.14	4.67
AK-MCS-EFF	$n_a = 1$	269.8	280.8	23.6	4.21	3.74	3.44
PBALC2 ($\epsilon_2 = 5\%$)	$n_a = 1$	63.9	72.6	10.3	4.08	2.01	0.25
QBALC($\sqrt{\bar{\rho}} = 0.50$)	$n_a = 4$	25.6	108.4	14.6	4.04	3.15	0.74
	$n_a = 1$	39.2	50.2	8.2	4.10	2.72	0.74
	$n_a = 2$	23.3	56.6	13.3	4.06	1.79	0.25
	$n_a = 3$	18.7	65.1	9.9	4.03	3.12	0.98
Proposed PALX	$n_a = 4$	17.4	77.4	10.3	4.06	2.57	0.25
	$n_a = 5$	16.3	88.5	12.1	4.09	2.16	0.49
	$n_a = 6$	15.0	96.0	11.8	4.08	1.98	0.25
	$n_a = 7$	14.6	107.2	13.1	4.04	2.82	0.74
	$n_a = 8$	14.1	116.8	10.5	4.08	2.20	0.25

4.3. Example 3: Two-bay four-storey spatial concrete frame

The third example examines a two-bay, four-storey spatial concrete frame structure subjected to concentrated loads [53]. This structure accounts for the complex nonlinear behaviors inherent to both concrete and rebar materials. To accurately simulate the system's behavior, each structural member is modeled as a nonlinear beam-column finite element using the OpenSees software. In this example, node 8 is identified as the most critical point, with its horizontal displacement influenced by 15 independent random variables. As

shown in Fig. 6, the control index is defined by the horizontal displacement at node 8. The performance function is given by:

$$\mathrm{g}(\boldsymbol{X}) = \bar{D} - D_8(f_c, \varepsilon_c, f_u, \varepsilon_u, f_y, E_s, b, F_6, F_8, F_5, F_7, F_{11}, F_{12}, F_{19}, F_{20}), \tag{30}$$

where D_8 denotes the horizontal displacement of node 8; $\bar{D}$ represents the allowable displacement, specified as $\bar{D} = 60$ mm. The physical meanings and statistical characteristics of the involved random variables are detailed in Table 4.

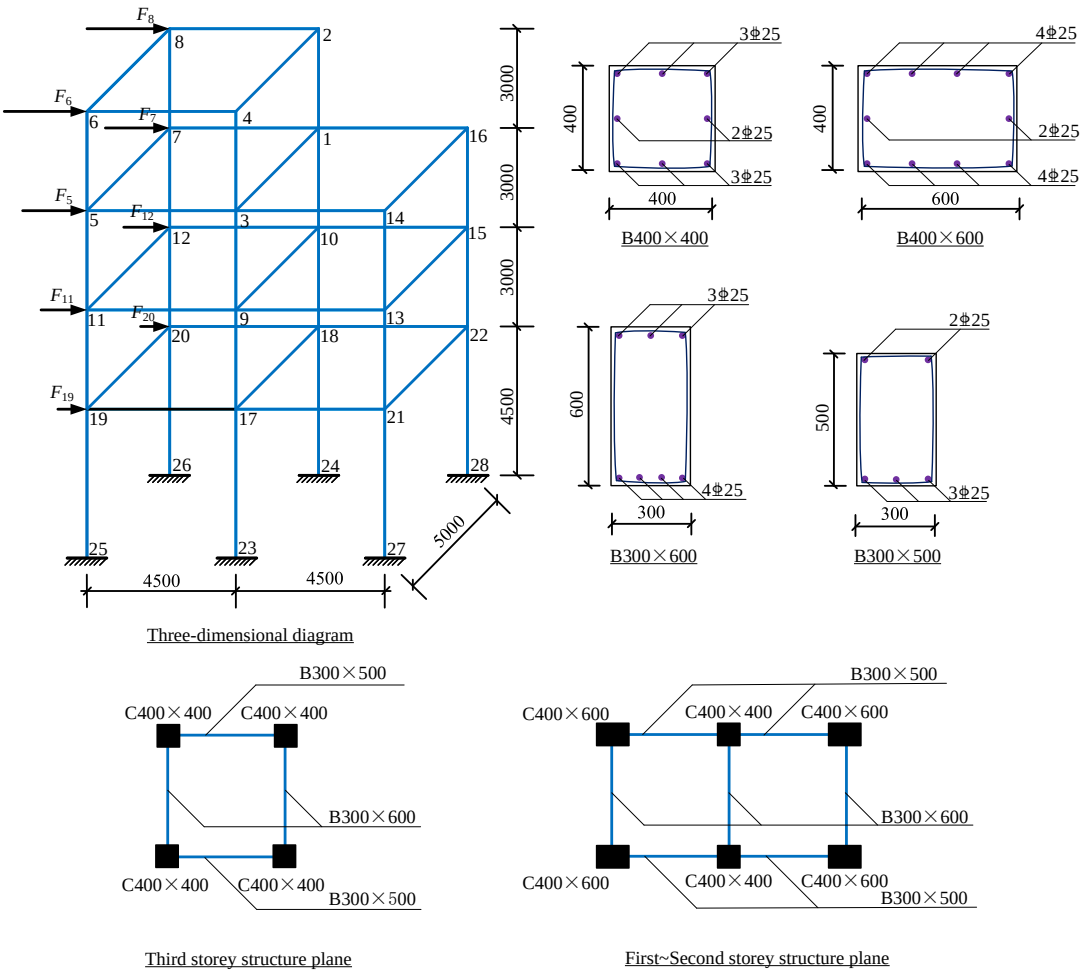


Figure 6: Two-bay four-story spatial concrete frame in Example 3.

Table 5 presents the reliability analysis results obtained from several methods. MCS with 10^{10} samples

Table 4: Distribution type and parameters of the random variables in Example 3.

Random variables	Unit	Description	Distribution	Mean	COV
f_c	MPa	Concrete compressive strength	Lognormal	26.8	0.1
ε_c	-	Concrete strain at maximum strength	Lognormal	0.0001	0.05
f_u	MPa	Concrete crushing strength	Lognormal	10.0	0.1
ε_u	-	Concrete strain at crushing strength	Lognormal	0.0035	0.05
f_y	MPa	Yield strength of rebar	Lognormal	355	0.1
E_s	GPa	Initial elastic modulus of rebar	Lognormal	200	0.1
b	-	Strain-hardening ratio of rebar	Lognormal	0.001	0.05
F_6	kN	External load	Lognormal	54	0.2
F_8	kN	External load	Lognormal	54	0.2
F_5	kN	External load	Lognormal	42	0.2
F_7	kN	External load	Lognormal	42	0.2
F_{11}	kN	External load	Lognormal	30	0.2
F_{12}	kN	External load	Lognormal	30	0.2
F_{19}	kN	External load	Lognormal	18	0.2
F_{20}	kN	External load	Lognormal	18	0.2

produces a reference failure probability of $P_f = 3.23 \times 10^{-5}$, with a COV of 0.39%. In the case of $n_a = 1$, the AK-MCMC method [51] requires an average of 266.1 function calls and results in a substantial relative error of 10.22%. In contrast, the PBALC2 method achieves a significantly lower error of 1.24%, with an average of 59.5 function calls. However, its failure probability COV reaches 7.31%, which remains higher than that of the proposed PALX method (5.03%), while the latter requires only 43.1 function calls on average. In the case of $n_a = 4$, the QBALC method yields a relative error of 0.62%, with a COV of 8.03%, which is notably higher than the 3.64% COV obtained by the proposed PALX method. Furthermore, PALX reduces the average number of function calls by approximately 22.1 calls compared to QBALC. Across all tested

configurations, the proposed method consistently provides nearly unbiased estimates of the failure probability, with COVs maintained below 7%. Overall, PALX substantially lowers computational cost while maintaining high levels of accuracy and robustness, offering a well-balanced and efficient alternative to conventional structural reliability analysis methods.

Table 5: Reliability analysis results of Example 3.

Method		N_{iter}	N_{call}		$\hat{P}_f$		$\delta_{\hat{P}_f}$ (%)
			Mean	COV (%)	Mean ($\times 10^{-5}$)	COV (%)	
MCS		-	10^{10}	-	3.23	0.39	-
AK-MCMC	$n_a = 1$	257.1	266.1	27.1	3.56	6.31	10.22
PBALC2 ($\epsilon_2 = 5\%$)	$n_a = 1$	50.5	59.5	13.4	3.19	7.31	1.24
QBALC($\sqrt{\bar{\rho}} = 0.50$)	$n_a = 4$	16.2	72.9	14.6	3.21	8.03	0.62
	$n_a = 1$	32.1	43.1	8.7	3.16	5.03	0.93
	$n_a = 2$	18.1	46.1	11.2	3.22	4.26	0.31
	$n_a = 3$	13.2	48.5	7.52	3.28	6.42	1.55
Proposed PALX	$n_a = 4$	10.7	50.8	12.0	3.21	3.64	0.62
	$n_a = 5$	11.4	64.2	11.3	3.19	4.30	1.24
	$n_a = 6$	9.3	61.9	8.7	3.22	3.91	0.31
	$n_a = 7$	8.2	62.4	11.5	3.29	3.85	1.86
	$n_a = 8$	7.5	64.0	12.3	3.18	2.66	1.55

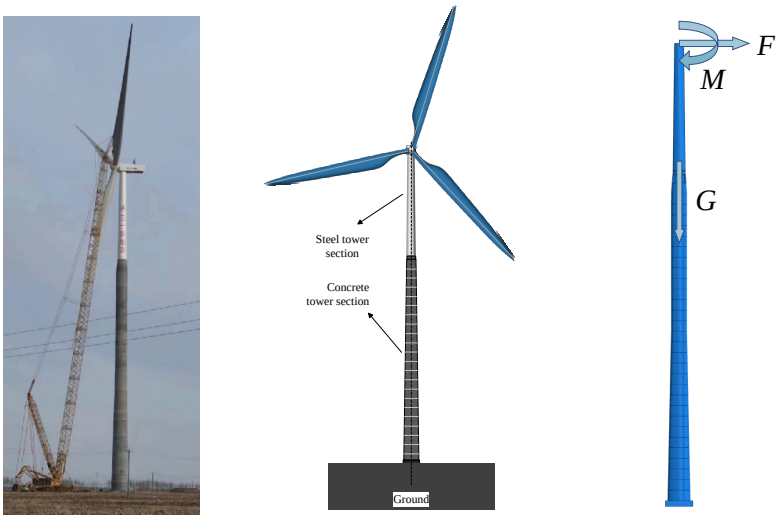
5. Application of the proposed PALX method to an onshore wind turbine tower

With the rapid expansion of wind energy in recent years, ensuring the safety and operational reliability of wind turbines has become increasingly vital [54]. The reliability of wind turbine towers, particularly steel-concrete hybrid towers (SCHTs), is significantly influenced by various uncertain factors, such as material properties and external loads, with wind loads being a significant source of potential damage [55]. Therefore,

analyzing the reliability of this concrete section is crucial to ensure the structural integrity and safety of the wind turbine. Traditional methods for calculating reliability, particularly when applied to complex FEMs of wind turbine towers, can be extremely time-consuming due to the extensive computational resources required for simulations. Given the complexity of these models, minimizing the number of g-function calls and computational effort is essential. To address these challenges, we propose using the PALX method to analyze the reliability of the SCHT under wind loads. This approach aims to showcase the practical engineering applicability of PALX, as an efficient and effective solution for reliability assessment in real-world wind turbine systems.

5.1. Description of the wind turbine tower

A 5-MW SCHT wind turbine, located in Tongyu, China, is chosen as the research object due to its representative characteristics and relevance to practical engineering applications. The wind turbine model is composed of two primary sections: the upper section, which includes the blades and nacelle, and the lower section, known as the SCHT. Detailed structural parameters of the model are provided in Table 6. The actual model is depicted in Fig. 7(a), and a schematic diagram is shown in Fig. 7(b).



(a) Actual model (b) Schematic model (c) FEM

Figure 7: The onshore wind turbine model.

Table 6: Design parameters and material properties of the wind turbine.

Part	Property	Value
Gross	Rating	5 MW
	Rotor orientation, configuration	Upwind, 3 blades
	Rotor diameter (m)	193
	Hub height (m)	160
	Rated rotor speed (rpm)	8
Impeller, hub and nacelle	Mass (kg)	250,000
Concrete tower section	Height (m)	112.05
	Segment number (pcs)	31
	Top diameter (m)	4.74
	Bottom diameter (m)	8.25
	Material	C65
Transition section	Height (m)	1.53
	Material	Q355
Steel tower section	Height (m)	43.80
	Top diameter (m)	3.38
	Bottom diameter (m)	4.49
PT tendons	Number	36
	Nominal cross-sectional area (mm^2)	140
	Material	Steel strand wire
Bolts and nuts	Thread specification	M56

For simplification, we assume the concrete foundation is a fixed support, and the impeller and nacelle are represented as a lumped mass concentrated at the top of the tower [56]. This approach reduces computational

complexity while still capturing the essential forces acting on the wind turbine. Additionally, the wind loads transmitted by the blades is represented by the thrust force F, which can be expressed as follows [57]:

$$F = \left(\frac{1}{2}\rho V^2\right) C_F(\pi R^2), \tag{31}$$

where ρ is the air density; V is the wind speed; C_F is the thrust coefficient; R is the rotor radius. As noted in [57], the bending moment at the tower base caused by blade wind loads is an order of magnitude larger than that induced by direct wind loads on the tower. Therefore, the direct wind loads on the tower are neglected in this analysis.

Based on the detailed description above, the simplified FEM constructed using Abaqus software is shown in Fig. 7(c). In this model, the hexahedral solid element C3D8R is employed to simulate the concrete, bolts, and nuts, ensuring detailed and accurate simulations. The steel reinforcement and pre-stressing reinforcement are represented by the 3D truss element T3D2, effectively modeling the reinforcement's structural behavior. The steel tower section is modeled with the shell element S4R, which is suitable for capturing the thin-walled nature of the tower. These modeling choices help achieve a balance between computational efficiency and accuracy. The FEM serves as the foundation for the subsequent reliability analysis.

5.2. Reliability analysis of the wind turbine tower

5.2.1. Description of the performance function

In the real world, failure of wind turbines frequently occurs in the concrete tower section of the tower. Therefore, analyzing the reliability of this concrete tower section is crucial. The tensile and compressive stresses experienced by the concrete tower section result from the combined effects of wind loads, the top mass, and the tower's self-weight. According to the GB 50010-2010 and GB 50135-2019 specifications [58, 59], concrete tower sections can fail through three distinct modes: compressive stress failure, tensile stress failure, and excessive top displacement. Consequently, the performance function for this problem is formulated as

follows:

$$\mathrm{g}(\boldsymbol{X}) = \min \begin{cases} \Delta F_{\text{press}} - F_{\text{press}}^{\max}, \\ \Delta U_{\text{top\_concrete}} - U_{\text{top\_concrete}}^{\max}, \\ \Delta F_{\text{tension}} - F_{\text{tension}}^{\max} \end{cases} \tag{32}$$

where the parameters are defined as follows according to the GB 50010-2010 and GB 50135-2019 specifications:

- $F_{\text{press}}^{\max} = 29.7\,\text{MPa}$: the maximum permissible compressive stress in the concrete tower section,
- $U_{\text{top\_concrete}}^{\max} = 1.12\,\text{m}$: the maximum permissible displacement at the top of the concrete tower section,
- $F_{\text{tension}}^{\max} = 2.04\,\text{MPa}$: the maximum permissible tensile stress in the concrete tower section.

Additionally, ΔF_{press}, $\Delta U_{\text{top\_concrete}}$, and $\Delta F_{\text{tension}}$ represent the actual maximum compressive stress, displacement at the top, and tensile stress in the FEM of SCHT.

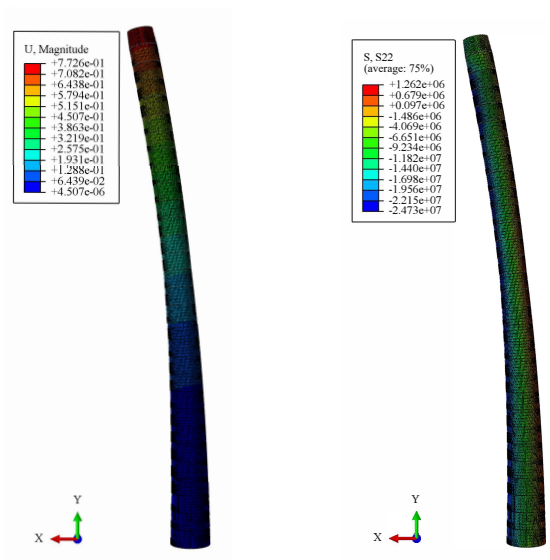
The random variables of the wind turbine are listed in Table 7. Fig. 8(a) and Fig. 8(b) show the displacement and force diagrams in the FEM of SCHT under wind loads, respectively. By evaluating the maximum tensile and compressive stresses, along with the top displacement, we can determine whether the concrete tower section is at risk of failure by comparing the calculated values to the permissible limits set by the specifications.

5.2.2. Analysis of reliability results

Reliability analysis was conducted using a computer equipped with an AMD Ryzen 7 5800X 8-Core processor running at 3.80 GHz, 32 GB of RAM, and MATLAB® 2022b. Due to the excessively long computation time required for a crude MCS to obtain a reference solution, the IS method available in UQLab [63] was employed instead. The failure probability determined by the IS method is 8.07×10^{-4}, with a COV of 3.56%. The reliability analysis results of the proposed PALX method, along with comparisons to other approaches, are summarized in Table 7. Among the non-parallel methods, AK-MCS-U and PBALC2 either exhibit excessive computational demands or insufficient robustness. Specifically, AK-MCS-U is extremely computationally intensive, requiring an average of 1396.2 minutes. PBALC2, although more efficient, still lacks robustness in terms of accuracy and consistency. In contrast, the proposed PALX method achieves

Table 7: Distribution type and parameters of the random variables in the wind turbine

Random variables	Unit	Distribution	Mean	COV	Reference
Thrust force	N	Normal	7.75×10^5	0.1	[60]
Bending moment	N·mm	Normal	1.2×10^7	0.1	[60]
Rotor and nacelle assembly mass	N	Normal	2.5×10^5	0.02	[60]
Young's modulus of the Q355C	Pa	Lognormal	2×10^{11}	0.03	[61]
Young's modulus of the concrete C65	Pa	Normal	3.65×10^{10}	0.06	[61]
Young's modulus of the concrete C70	Pa	Normal	3.7×10^{10}	0.06	[61]
Young's modulus of the rebar HRB400	Pa	Normal	2×10^{11}	0.033	[62]
Young's modulus of the steel strand	Pa	Normal	1.95×10^{11}	0.03	[62]
Initial tension of prestressed strand	N	Normal	1.28×10^9	0.015	[62]
Steel strand diameter	mm	Normal	15.2	0.04	[62]



(a) Displacement (b) Stress

Figure 8: Abaqus simulation results for concrete segments under wind loads.

substantial improvements in both efficiency and robustness, with an average computation time of 404.7 minutes and a COV of 3.11%, representing a clear advantage over both AK-MCS-U and PBALC2. For parallel methods, both PALX and QBALC demonstrate enhanced computational performance. Notably, PALX further outperforms QBALC by requiring only 7.1 iterations on average and saving 153.4 minutes in average computation time, thereby confirming its superior overall effectiveness.

These results highlight the superior performance of the PALX method in terms of both efficiency and accuracy for practical engineering reliability analysis of wind turbine towers. Specifically, the failure probabilities computed using PALX are approximately 8.09×10^{-4} for $n_a = 1$ and 8.06×10^{-4} for $n_a = 4$, demonstrating the method's robustness across different levels of parallelism. However, it is worth noting that both values exceed the specification limit of 6.87×10^{-4} as defined by the GB 50135–2019 standard. This suggests that the current structural design may not fully comply with the required safety criteria. Therefore, further structural optimization or the implementation of reliability enhancement measures may be necessary to ensure compliance with regulatory safety requirements. To address this issue, it is recommended to adopt higher-strength concrete in the tower section to enhance the pre-stressing effect throughout the wind turbine system. Alternatively, other effective reinforcement strategies may also be considered. These modifications are expected to enhance structural reliability and ensure compliance with safety standards. In addition, the findings of this study are significant for the wind energy sector as they can reduce computational costs in reliability analysis, making safety assessments more feasible for large-scale wind turbine systems. These results can also inform regulatory decisions and supporting the development of safety standards and contributing to the sustainable growth of the renewable energy sector.

Table 8: Reliability analysis results of the wind turbine.

Method		N_{iter}	N_{call}		$\hat{P}_f$		$\delta_{\hat{P}_f}$ (%)	Time (min)
			Mean	COV (%)	Mean ($\times 10^{-4}$)	COV (%)		
IS[a]		-	1042	-	8.07	3.56	-	21070.2
AK-MCS-U	$n_a = 1$	59.1	70.1	13.2	8.03	6.52	0.50	1396.2
PBALC2 ($\epsilon_2 = 5\%$)	$n_a = 1$	28.5	37.5	7.6	8.04	8.43	0.37	682.1
QBALC($\sqrt{\bar{\rho}} = 0.50$)	$n_a = 4$	12.3	54.8	8.3	8.08	4.11	0.12	343.6
Proposed PALX	$n_a = 1$	16.7	27.7	4.4	8.09	3.11	0.25	404.7
	$n_a = 4$	7.1	36.5	7.4	8.06	2.28	0.12	190.2

[a] The results of IS are calculated using UQLab [63].

6. Concluding remarks

This paper presents a novel method, called 'parallel active learning XGBoost' (PALX), to address the challenge of computationally expensive structural reliability analysis. The proposed approach integrates XGBoost—a gradient-boosting framework adept at modeling complex nonlinear relationships—whose predictive uncertainty is quantified via cross-validation. By introducing a Gaussian assumption, a convenient failure probability estimator is adapted from a Bayesian active learning method, as well as a stopping criterion to ensure reliable convergence. Furthermore, we propose a novel learning function, termed 'prediction variance-weighted epistemic uncertainty contribution' (PVWEUC), and develop a multi-point selection strategy termed 'lower confidence bound believer' (LCBB), which supports parallel computing and significantly reduces the overall computational cost.

The effectiveness of the proposed PALX method is demonstrated through three numerical examples and one practical engineering application involving a hybrid tower wind turbine. The results lead to the following key conclusions: (1) The proposed method is able to estimate extremely small failure probabilities, with magnitudes down to 10^{-9}; (2) Selecting multiple points in each iteration, typically around $n_a = 8$, minimizes the total number of iterations required; (3) The PALX method performs well in a practical engineering

context, specifically in assessing the failure probability of hybrid tower wind turbines, demonstrating its potential for real-world structural reliability applications, especially in the wind power industry.

Although PALX shows promising results, further improvements are possible. One challenge is the increased computational time resulting from the sequential VAIS technique, particularly as the problem dimensionality grows. Future work could focus on developing more advanced numerical integration methods to address this issue. In addition to these methodological advancements, extending the application of PALX to time-dependent reliability problems, in which performance functions depend on both random variables and time-varying parameters, represents a distinct and promising direction for future research. Such extensions would broaden the applicability of PALX to a wider range of engineering problems involving deterioration, fatigue, and evolving uncertainties.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (Grant No. 52278178).

References

[1] R. Y. Rubinstein, D. P. Kroese, Simulation and the Monte Carlo method, John Wiley & Sons, 2016.

[2] R. Melchers, Importance sampling in structural systems, Structural Safety 6 (1) (1989) 3–10.

[3] A. Tabandeh, G. Jia, P. Gardoni, A review and assessment of importance sampling methods for reliability analysis, Structural Safety 97 (2022) 102216.

[4] S.-K. Au, J. L. Beck, Estimation of small failure probabilities in high dimensions by subset simulation, Probabilistic Engineering Mechanics 16 (4) (2001) 263–277.

[5] P.-S. Koutsourelakis, H. J. Pradlwarter, G. I. Schueller, Reliability of structures in high dimensions, part I: algorithms and applications, Probabilistic Engineering Mechanics 19 (4) (2004) 409–417.

[6] P. Bjerager, Probability integration by directional simulation, Journal of Engineering Mechanics 114 (8) (1988) 1285–1302.

[7] R. Rackwitz, B. Flessler, Structural reliability under combined random load sequences, Computers & Structures 9 (5) (1978) 489–494.

[8] A. Der Kiureghian, H.-Z. Lin, S.-J. Hwang, Second-order reliability approximations, Journal of Engineering Mechanics 113 (8) (1987) 1208–1225.

[9] Y.-G. Zhao, T. Ono, Moment methods for structural reliability, Structural Safety 23 (1) (2001) 47–75.

[10] J. Xu, C. Dang, A new bivariate dimension reduction method for efficient structural reliability analysis, Mechanical Systems and Signal Processing 115 (2019) 281–300.

[11] 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.

[12] C. Dang, J. Xu, A mixture distribution with fractional moments for efficient seismic reliability analysis of nonlinear structures, Engineering Structures 208 (2020) 109912.

[13] I. Kaymaz, Application of Kriging method to structural reliability problems, Structural Safety 27 (2) (2005) 133–151.

[14] C. G. Bucher, U. Bourgund, A fast and efficient response surface approach for structural reliability problems, Structural Safety 7 (1) (1990) 57–66.

[15] C. M. Rocco, J. A. Moreno, Fast monte carlo reliability evaluation using support vector machine, Reliability Engineering & System Safety 76 (3) (2002) 237–243.

[16] 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.

[17] 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.

[18] 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.

[19] R. Teixeira, M. Nogal, A. O'Connor, Adaptive approaches in metamodel-based reliability analysis: A review, Structural Safety 89 (2021) 102019.

[20] M. Moustapha, S. Marelli, B. Sudret, Active learning for structural reliability: Survey, general framework and benchmark, Structural Safety 96 (2022) 102174.

[21] 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.

[22] R. Schöbi, B. Sudret, S. Marelli, Rare event estimation using polynomial-chaos Kriging, ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part A: Civil Engineering 3 (2) (2017) D4016002.

[23] K. Cheng, Z. Lu, Adaptive Bayesian support vector regression model for structural reliability analysis, Reliability Engineering & System Safety 206 (2021) 107286.

[24] S. Marelli, B. Sudret, An active-learning algorithm that combines sparse polynomial chaos expansions and bootstrap for structural reliability analysis, Structural Safety 75 (2018) 67–74.

[25] T. Zhou, Y. Peng, An active-learning reliability method based on support vector regression and cross validation, Computers & Structures 276 (2023) 106943.

[26] B. Echard, N. Gayton, M. Lemaire, N. Relun, A combined importance sampling and Kriging reliability method for small

failure probabilities with time-demanding numerical models, Reliability Engineering & System Safety 111 (2013) 232–240.

[27] X. Huang, J. Chen, H. Zhu, Assessing small failure probabilities by AK–SS: An active learning method combining Kriging and subset simulation, Structural Safety 59 (2016) 86–95.

[28] G. Li, T. Wang, Z. Chen, J. He, X. Wang, X. Du, RBIK-SS: A parallel adaptive structural reliability analysis method for rare failure events, Reliability Engineering & System Safety 239 (2023) 109513.

[29] X. Zhang, Z. Lu, K. Cheng, AK-DS: An adaptive Kriging-based directional sampling method for reliability analysis, Mechanical Systems and Signal Processing 156 (2021) 107610.

[30] C. Dang, M. A. Valdebenito, M. G. Faes, P. Wei, M. Beer, Structural reliability analysis: A Bayesian perspective, Structural Safety 99 (2022) 102259.

[31] C. Dang, M. Beer, Semi-Bayesian active learning quadrature for estimating extremely low failure probabilities, Reliability Engineering & System Safety 246 (2024) 110052.

[32] X. Yang, Y. Liu, Y. Zhang, Z. Yue, Probability and convex set hybrid reliability analysis based on active learning Kriging model, Applied Mathematical Modelling 39 (14) (2015) 3954–3971.

[33] Z. Sun, J. Wang, R. Li, C. Tong, LIF: A new Kriging based learning function and its application to structural reliability analysis, Reliability Engineering & System Safety 157 (2017) 152–165.

[34] 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.

[35] 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.

[36] C. Dang, T. Zhou, M. A. Valdebenito, M. G. Faes, Yet another Bayesian active learning reliability analysis method, Structural Safety 112 (2024) 102539.

[37] V. Dubourg, B. Sudret, F. Deheeger, Metamodel-based importance sampling for structural reliability analysis, Probabilistic Engineering Mechanics 33 (2013) 47–57.

[38] R. Teixeira, M. Nogal, A. O'Connor, B. Martinez-Pastor, Reliability assessment with density scanned adaptive Kriging, Reliability Engineering & System Safety 199 (2020) 106908.

[39] Z. Chen, G. Li, J. He, Z. Yang, J. Wang, A new parallel adaptive structural reliability analysis method based on importance sampling and K-medoids clustering, Reliability Engineering & System Safety 218 (2022) 108124.

[40] Z. Chen, J. He, G. Li, Z. Yang, T. Wang, X. Du, Fast convergence strategy for adaptive structural reliability analysis based on Kriging believer criterion and importance sampling, Reliability Engineering & System Safety 242 (2024) 109730.

[41] Z. Hu, C. Dang, L. Wang, M. Beer, Parallel Bayesian probabilistic integration for structural reliability analysis with small failure probabilities, Structural Safety 106 (2024) 102409.

[42] C. Dang, A. Cicirello, M. A. Valdebenito, M. G. Faes, P. Wei, M. Beer, Structural reliability analysis with extremely small

failure probabilities: A quasi-Bayesian active learning method, Probabilistic Engineering Mechanics 76 (2024) 103613.

[43] T. Zhou, X. Zhu, T. Guo, Y. Dong, M. Beer, Multi-point bayesian active learning reliability analysis, Structural Safety 114 (2025) 102557.

[44] Z. Wang, A. Shafieezadeh, ESC: an efficient error-based stopping criterion for Kriging-based reliability analysis methods, Structural and Multidisciplinary Optimization 59 (2019) 1621–1637.

[45] Y. Zhang, Y. Dong, J. Xu, An accelerated active learning Kriging model with the distance-based subdomain and a new stopping criterion for reliability analysis, Reliability Engineering & System Safety 231 (2023) 109034.

[46] T. Chen, C. Guestrin, XGBoost: A scalable tree boosting system, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 785–794.

[47] 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.

[48] W. Liu, Z. Chen, Y. Hu, Xgboost algorithm-based prediction of safety assessment for pipelines, International Journal of Pressure Vessels and Piping 197 (2022) 104655.

[49] D. Ginsbourger, R. Le Riche, L. Carraro, Kriging is well-suited to parallelize optimization, in: Computational intelligence in expensive optimization problems, Springer, 2010, pp. 131–162.

[50] X. Yang, Y. Liu, C. Mi, X. Wang, Active learning Kriging model combining with kernel-density-estimation-based importance sampling method for the estimation of low failure probability, Journal of Mechanical Design 140 (5) (2018) 051402.

[51] 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.

[52] B. J. Bichon, J. M. McFarland, S. Mahadevan, Efficient surrogate models for reliability analysis of systems with multiple failure modes, Reliability Engineering & System Safety 96 (10) (2011) 1386–1395.

[53] X. Chen, Z. Lin, Structural nonlinear analysis program OpenSEES theory and tutorial, China Architecture & Building Press, Beijing, China (2014) 87–89.

[54] M. Mardfekri, P. Gardoni, Multi-hazard reliability assessment of offshore wind turbines, Wind Energy 18 (8) (2015) 1433–1450.

[55] S.-Z. Li, X.-H. Zhou, Y. Gao, Y.-H. Wang, D. Gan, W. Ren, M.-H. Lai, R.-H. Zhu, Damage detection of concrete segment in steel-concrete hybrid wind turbine tower, Structures 56 (2023) 105009.

[56] D. Zhang, H. B. Bhattarai, F. Wang, X. Zhang, H.-J. Hwang, X. Wu, Y. Tang, S. Kang, Dynamic characteristics of segmental assembled HH120 wind turbine tower, Engineering Structures 303 (2024) 117438.

[57] C. Sun, V. Jahangiri, Fatigue damage mitigation of offshore wind turbines under real wind and wave conditions, Engineering Structures 178 (2019) 472–483.

[58] GB 50135-2006, Code for design of high-rising structures, Standardization Administration of China, Beijing, China, 2019

(in Chinese).

[59] GB 50010-2010, Code for design of concrete structures, China Planning Press, Beijing, China, 2010 (in Chinese).

[60] S. Al-Sanad, L. Wang, J. Parol, A. Kolios, Reliability-based design optimisation framework for wind turbine towers, Renewable Energy 167 (2021) 942–953.

[61] J. Velarde, C. Kramhøft, A. Mankar, J. D. Sørensen, Uncertainty modeling and fatigue reliability assessment of offshore wind turbine concrete structures, International Journal of Offshore and Polar Engineering 29 (02) (2019) 165–171.

[62] T. Qu, B. Zeng, Z. Zhou, L. Huang, D. Chang, Dynamic and uncertainty-based assessment of the progressive collapse probability of prestressed concrete frame structures with infill walls, Structures 61 (2024) 106105.

[63] M. Moustapha, S. Marelli, B. Sudret, Uqlab user manual–active learning reliability, Report UQLab-V2, in Chair of Risk, Safety and Uncertainty Quantification, ETH Zurich (2022) 0–117.