Reliability Engineering and System Safety 261 (2025) 111059

Contents lists available at ScienceDirect

Reliability Engineering and System Safety

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

Surrogate modeling for probability distribution estimation: Uniform or adaptive design?

Maijia Su [a,*], Ziqi Wang [b], Oreste Salvatore Bursi [a], Marco Broccardo [a,*]

[a] Department of Civil, Environmental and Mechanical Engineering, University of Trento, Trento, Italy
[b] Department of Civil and Environmental Engineering, University of California, Berkeley, United States

ARTICLE INFO

Keywords:
Uncertainty quantification
Full distribution function
Surrogate models
Active learning
Space-filling design

ABSTRACT

The active learning (AL) technique, one of the state-of-the-art methods for constructing surrogate models, has shown high accuracy and efficiency in forward uncertainty quantification (UQ) analysis. This paper provides a comprehensive study on AL-based global surrogates for computing the full distribution function, i.e., the cumulative distribution function (CDF) and the complementary CDF (CCDF). To this end, we investigate the three essential components for building surrogates, i.e., types of surrogate models, enrichment methods for experimental designs, and stopping criteria. For each component, we choose several representative methods and study their desirable configurations. In addition, we use a uniform design based on maximin-distance criteria as a baseline for measuring the improvement of using AL. Combining all the representative methods, a total of 1920 UQ analyses are carried out to solve 16 benchmark examples. The performance of the selected strategies is evaluated based on accuracy and efficiency. In the context of full distribution estimation, this study concludes that (*i*) The benefit of using AL is lower than expected and varies across different surrogate models, with three reasons for this performance variability analyzed in detail. (*ii*) Detailed recommendations are provided for the three surrogate components, depending on the features of the problems (especially the local nonlinearity), target accuracy, and computational budget.

1. Introduction

Real-world engineering systems inevitably involve uncertainties, which may arise from physical properties (e.g., material strength and geometric dimensions) and ambient conditions (e.g., external load and environmental noises). These uncertainties are often represented by probability distributions that describe the likelihood of certain variables occurring within a given range. Uncertainty Quantification (UQ) is a field of study that aims to understand how these uncertainties affect the overall performance of a system. For this purpose, UQ techniques are used to propagate the randomness of the basic variables and determine the statistical properties of the system responses. The results of UQ can be used to inform safety assessments, optimize designs, and make risk-informed decisions. This study focuses on a key aspect of the UQ analysis: estimating the full probability distribution function, including the tails (up to prescribed quantiles of interest) of both the cumulative distribution function (CDF) and the complementary CDF (CCDF).

In this context, the source of uncertainty is modeled by a random vector denoted by $\boldsymbol{X} = [X_1, X_2, \dots, X_N]$ with a prescribed joint probability density function (PDF) $f_{\boldsymbol{X}}(\boldsymbol{x})$. The system output Y, propagated from $\boldsymbol{X}$, represents the Quantity of Interest (QoI) selected to describe the system performance. In general, the QoI can be a multivariate output, but this work only focuses on the unidimensional case. The connection between $\boldsymbol{X}$ and Y is built by a deterministic simulator $\mathcal{M} : \boldsymbol{x} \in \mathbb{R}^N \mapsto y \in \mathbb{R}$. In practice, evaluating $\mathcal{M}(\cdot)$ is computationally expensive since it may involve complex numerical models, e.g., finite element models (FEMs) and computational fluid dynamics models. Given these definitions, the CDF $F_Y(y)$ and CCDF $\bar{F}_Y(y)$ can be expressed as [1]:

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

and

$$\bar{F}_Y(y) = \mathbb{P}(Y > y) = \mathbb{P}(\mathcal{M}(\boldsymbol{X}) > y) = \int_{\mathcal{M}(\boldsymbol{x}) > y} f_{\boldsymbol{X}}(\boldsymbol{x}) d\boldsymbol{x} = 1 - F_Y(y) \quad . \tag{2}$$

Despite the trivial relationship between Eqs. (1) and (2), these equations need to be evaluated separately. This is because the low proba-

* Corresponding authors.
E-mail addresses: maijia.su@unitn.it (M. Su), marco.broccardo@unitn.it (M. Broccardo).

https://doi.org/10.1016/j.ress.2025.111059

Available online 3 April 2025

bility regions of CDF and CCDF span opposite tails of the PDF $f_Y(y)$. The separate computations ensure the accurate estimation of the full distribution, including the low-probability regions.

To solve Eqs. (1) and (2), the distribution function can be discretized into a set of thresholds $[y_{\min} < \cdots < y_m < \cdots < y_{\max}]$. Then, the full distribution can be approximated at a series of positions y_m, i.e., to compute a sequence of multiple integrals of $f_{\boldsymbol{X}}(\boldsymbol{x})$ on the region $D_m = \{\boldsymbol{x} \in \mathbb{R}^N | \mathcal{M}(\boldsymbol{x}) \leq y_m\}$ for $F_Y(y_m)$ or on the region $\bar{D}_m = \{\boldsymbol{x} \in \mathbb{R}^N | \mathcal{M}(\boldsymbol{x}) > y_m\}$ for $\bar{F}_Y(y_m)$. Therefore, the UQ problems are converted into common integration problems, which can be solved with a wide variety of methods. Recently, non-intrusive methods have received wide attention since $\mathcal{M}(\cdot)$ can be taken as a black-box simulator, allowing the UQ analysis to be separated from solving the deterministic equations governing the problem at hand. According to [2], these approaches can be categorized as: (*i*) deterministic numerical integration methods (e.g., classical numerical integration [3]), (*ii*) stochastic simulation methods (e.g., Monte Carlo simulation [4], importance sampling (IS) [5,6], subset simulation (SS) [7]), (*iii*) local approximation methods (e.g., FORM and SORM [8,9], Taylor series method [10]) and (*iv*) surrogate-model aided methods (see an overview in [11]). In practice, the best approach strongly relies on the target accuracy, computational budget, dimensionality, and the specific features of the problem.

In recent decades, surrogate modeling methods have gained popularity since they can save substantial computational costs while providing relatively high-precision estimations. Surrogate models, a.k.a. metamodels, seek to find an inexpensive approximation model $\hat{\mathcal{M}}(\cdot)$ to substitute the original simulator $\mathcal{M}(\cdot)$. In order to solve Eqs. (1) and (2), we need to construct global surrogate models over the entire design domain $\mathcal{D}_{\boldsymbol{x}}$. *Global* surrogate model differs from *local* surrogate model, which only requires local information of the original simulator to solve the problem. Local surrogates can be applied to optimization problems (e.g., search for the global minima [12]) or reliability analysis (i.e., a performance-based classification problem [13]). The study [14] shows that two essential ingredients are required to build surrogate models: (*i*) a function space with certain properties and (*ii*) a Design of Experiments (DoE). The first ingredient implies specifying a function space $\mathcal{F} = \{\hat{\mathcal{M}}(\cdot) : \mathcal{D}_{\boldsymbol{x}} \mapsto \mathbb{R}\}$, where, based on some loss functions, we identify an optimal function that closely resembles the original simulator. The properties of $\mathcal{F}$ rely on the selected types of surrogate models and their configurations and parameters. In the literature, one can find a wide range of surrogate models; an incomplete list includes polynomial response surfaces [15], artificial neural networks [16], Kriging (also called Gaussian process) [17,18], support vector machines [19,20], polynomial chaos expansions (PCE) [21] and PCE-Kriging (PCK) [22,23]. The configuration for the surrogate model determines the flexibility of the function space $\mathcal{F}$. A function space with limited flexibility may not include a good approximation to the true model $\mathcal{M}(\cdot)$. Conversely, using an excessively flexible function space increases the efforts of finding the optimal approximation and may result in overfitting (i.e., the model performs well on the training data but fails to generalize to unseen data). The other ingredient, the DoE, helps the quantification of the loss functions and thus the identification of the best approximation in $\mathcal{F}(\mathcal{D}_{\boldsymbol{x}})$. A DoE consists of a set of input–output pairs, where the inputs span the design domain, and the outputs are evaluated through simulators at corresponding inputs. The core of generating a DoE is to determine the number and locations of input samples. The well-known examples of generating DoEs can be traced back to designs such as Full Factorial or Fractional Factorial [24], and Central Composite Design [25]. These classical DoEs were initially developed for physical experiments [26]. However, DoEs for computer simulations are very different in nature, as the samples are often noise-free (i.e., the outcomes come from deterministic computer simulators), and the control variables are clear and pre-defined. It is no surprise that studies related to DoEs have surged dramatically in the past several decades due to the popularity of computer experiments. In this study, we investigate two categories of DoEs: uniform design and active-learning (a.k.a. adaptive) design.

The uniform design aims to generate samples that uniformly fill both the design space $\mathcal{D}_{\boldsymbol{x}}$ and any possible subspace of $\mathcal{D}_{\boldsymbol{x}}$. In the literature [27], the uniformity in subspace is called projection or non-collapsingness property. The uniform design is achieved by optimizing some metrics that can quantify the space-filling capability of any given sample set. The study [26] summarizes two broad types of uniformity metrics, i.e., discrepancy-based [28,29] and distance-based [30,31]. The discrepancy-based criteria measure the difference between the CDF of the uniform distribution in design space and the empirical CDF of samples, e.g., Sobol's sequence [32] and Halton sequence [33]. The distance-based criteria study the relative geometric position of samples, e.g., maximin/minimax distance design [30], Voronoi Diagram-based design [34], and minimum potential energy design [35]. The projection property motivates researchers to study the class of so-called Latin hypercube sampling (LHS) [36], which demonstrates perfect uniformity in any one-dimensional projection. However, LHS may present poor space-filling ability in the whole design space. A variety of studies attempt to enhance the LHS by integrating the aforementioned uniformity metrics [26], e.g., maximin-LHS [37]. In practice, the optimized size of DoE is generally unknown and it is more common to increase the size gradually until the surrogates are well-trained. This restriction necessitates a new class of quasi-uniform design that can sequentially enrich a DoE while keeping a nearly-optimized uniform metric; examples include the sequential maximin distance design [38] and sequential LHS [39].

The active-learning (AL) design selects the most informative points sequentially. It follows that AL minimizes the size of the DoE for building accurate surrogates. The key idea of AL was initially brought from the machine learning community [40], but its earlier implementation on surrogate models can be found in Bayesian global optimizations [12]. The method relies on the "score functions" (e.g., the expected improvement (EI) function [12]), which can measure the potential benefits at unexplored positions in improving the current solution. More recently, AL was introduced into reliability analyses in [41], which proposed an efficient global reliability analysis (EGRA) method. In this work, the score function EI was adapted into an expected feasibility function (EFF), which can measure the deviation of the approximated model from the zero-contour of $\mathcal{M}(\boldsymbol{x})$. EGRA inspired the AL-Kriging–Monte Carlo Simulation (AK-MCS) [13], which couples an MCS estimator (that represents the continuous design space as a discrete candidate pool) and training of adaptive Kriging. AK-MCS proposes the learning functions U, with more recent variations and advancements such as H [42], IF [43], LIF [44] and many others [45–49]. Another key component of AL is the stopping criteria, which determine when to terminate the training process. The stopping criteria commonly have two essential aspects: a performance function (that evaluates the accuracy of iterative surrogates) and a threshold to trigger the stop. The performance function is typically selected as the learning functions (e.g., EFF, U, H), the stability or confidence of estimated quantities of interest (e.g., the failure probability) [50–53], or a composition of multiple stop criteria [47,54]. The thresholds for triggering the stop are typically determined through numerical studies. Furthermore, advanced techniques can accelerate the AL training process by reducing the size of the candidate pool. This is achieved by replacing the MCS estimator with advanced variance-reduction techniques, such as IS [55], SS [56] and others [54,57–59].

In principle, an extension of AK-MCS to full distribution estimations is straightforward since calculating the CDF or CCDF at a fixed threshold y_m can be considered as a standard reliability analysis problem. However, a naive implementation based on a sequence of y_m significantly increases the computational cost. This computational problem was tackled in [1] using a mesh-free, AL-based Gaussian process (AL-GP). Specifically, the study introduced a two-step learning function to

enhance efficiency. The first step identifies an optimal threshold y^* by maximizing the localized full distribution function error at each AL iteration. The second step uses a reliability-based learning function (see examples in [13,41]) specified with the predetermined threshold y^* to add a new training point into the DoE. The two-step function leverages more information about the estimated full distribution to speed up the training process of surrogate models.

Outside the UQ community, global adaptive surrogate models have been extensively studied in the contexts of sensitivity analysis [60] and parameter estimations [61]. Interested readers may refer to the review papers [62,63] for more details. These types of global surrogates are generally established on a box-constrained uniform space of interest. In UQ problems, the design space is generally not uniform but is defined by the joint PDF of input variables. In this context, the surrogates are generally built on an important region that contains the bulk of the probability mass. It follows that the parametric region of interest is typically given by an iso-probability contour rather than a box-type domain. Therefore, to conduct comparisons between AL and uniform design, we require an approach that generates uniform samples in design spaces defined by iso-probability boundaries. Observe that this is different from generating samples from the joint PDF.

Most of the scientific literature shows the superiority of AL techniques in constructing global surrogate models compared to uniform designs [14,64–67]. However, a few studies pointed out that AL sampling can be outperformed by the uniform design [68]. In the context of full distribution computations, there is a research gap to answer the critical question of whether AL techniques can consistently outperform uniform design. Practically, it remains to be answered whether AL necessarily results in computational cost savings. Furthermore, a comprehensive study to determine the influence of different components, such as the DoE enrichment and stopping criteria, on surrogate performances is missing. Therefore, this study aims at bridging the gap between theoretical developments and engineering applications, offering strategic recommendations for researchers and practitioners. Specifically, our first goal is to systematically compare AL and uniform design approaches in the context of full distribution computations and determine whether AL consistently outperforms uniform design. Building on the results, our second goal is to provide practical guidance for constructing global surrogate models for full distribution estimations.

This study addresses the two goals as follows: we leverage the method presented in [1] as a foundation and then develop a modular framework, drawing upon the insights provided in [69]. The framework is designed by combining three independent modules: a surrogate module, a DoE enrichment module, and a stopping criteria module. The surrogate module is composed by three classical surrogate techniques: Gaussian Process [70], Polynomial Chaos Expansion (PCE), [71] and PCE-Kriging (PCK) [22]. The DoE enrichment module is composed of three learning functions and a new distance-based uniform design method, which is proposed to serve as the baseline. Specifically, the three learning functions are: the maximum of variance (MoV) learning criterion, the two-step learning function proposed in [1], and a gradient-based learning function proposed in [72].[1] Moreover, the proposed uniform design method combines the maximum–minimum (maximin) distance design proposed in [30] with the pool-based representation as adopted in AK-MCS [13]. This sampling method is named as *pool-based maximin-distance design* and it allows a sequential generation of quasi-uniform samples in non-box-constrained design spaces. Finally, the stopping criteria module is composed by two classes of stopping criteria: the static-based and the variance-based. In this module, we further explore the parameter settings, such as the threshold to trigger the stopping condition and the number of consecutive triggers for the stopping condition. A combination of these selected methods is applied to solve a collection of 16 benchmark problems that involve analytical functions and FEM examples. In total, 1920 UQ analyses are carried out to support this comparative study.

The study is organized as follows. Section 2 presents the general framework for constructing global surrogate models and introduces the selected methods for each module. Section 3 develops the pool-based maximin-distance design, a new sequential uniform design method proposed as the baseline of this comparative study. Section 4 carries out a comparative study across a wide range of UQ problems and selected solutions. Section 5 discusses the results and provides recommendations on constructing surrogates for computing full distributions. Finally, Section 6 concludes this study.

2. Methodology: sequential surrogates for computing full distributions

This section presents a modular framework for computing the full distribution through surrogate models. Specifically, we adapt the modular framework introduced in [69] to our context.

2.1. Procedure of sequential surrogate modeling

Sequential surrogate models build on an iterative approach which entails a sequence of surrogate models. The sequence starts with an initial DoE, i.e. a parsimonious sequence of input–output pairs, $\{\mathcal{X}, \mathcal{Y}\} = \left\{\left(\boldsymbol{x}_i^d, \mathcal{M}(\boldsymbol{x}_i^d)\right) | i = 1, 2, \ldots, N_\mathrm{d}\right\}$. In subsequent iterations, new surrogate models are created based on an expanded DoE. This expansion, or enrichment, is achieved by adding additional input–output pairs to the original DoE. The enrichment strategy used is designed to identify the most effective candidates for improving the training of the surrogate model. The process continues until predetermined stopping conditions are met. Ultimately, the size and position of the DoE are determined automatically, and the final updated surrogate model is used to estimate the full distribution. In summary, the steps for constructing sequential surrogate models are outlined in Algorithm 1.

Algorithm 1: The procedure of constructing sequential surrogates

1. *Initialization*. Generate a large candidate pool $S = \{\boldsymbol{x}_i^S | i = 1, 2, .., N_\mathrm{MCS}\}$ for training from the joint PDF $f_{\boldsymbol{X}}(\boldsymbol{x})$. Define an initial DoE $\{\mathcal{X}, \mathcal{Y}\}$ of size $N_\mathrm{d} << N_\mathrm{MCS}$.
2. *Training*. Train the surrogate model based on the current DoE.
3. *Compute CDF/CCDF*. Estimate the CDF in Eq. (1) and CCDF in Eq. (2) by Monte Carlo simulation with the surrogate model.
4. *Check Convergence*. If a specified stopping condition is satisfied, terminate the algorithm.
5. *Enrich the DoE*. Add new training points into the DoE and proceed to Step 2.

2.2. Surrogate modeling techniques

Surrogate modeling represents the core of the framework. We examine three popular surrogate models: GP, PCE, and PCK and integrate them into a unified form. For a detailed introduction to these surrogate models, readers are referred to [73–75]. We start with GP and later show that the other two surrogate models share a similar formulation.

The homogeneous GP treats the black-box simulator $\mathcal{M}(\boldsymbol{x})$ as a realization of a Gaussian process:

$$\hat{\mathcal{M}}^{\mathrm{GP}}(\boldsymbol{x}) = \mu(\boldsymbol{x}|\boldsymbol{\beta}) + \sigma^2 Z(\boldsymbol{x}|\boldsymbol{\theta}), \tag{3}$$

where the mean/trend function $\mu(\boldsymbol{x}|\boldsymbol{\beta})$ of the Gaussian process is a regression model with parameters $\boldsymbol{\beta} \in \mathbb{R}^q$, σ^2 is the process variance, and $Z(\boldsymbol{x}|\boldsymbol{\theta})$ is a zero-mean, unit-variance Gaussian process uniquely determined by a homogeneous correlation function/kernel $R(\boldsymbol{x}_i, \boldsymbol{x}_j|\boldsymbol{\theta}) \in \mathbb{R}$ parameterized by $\boldsymbol{\theta}$. Commonly used kernels include the Gaussian,

[1] This learning function has been developed for global surrogates but without integrating the information from the distribution function.

cubic, spline, and Matérn (see [76] for details). It is typical to assume a linear regression model for $\mu(\boldsymbol{x}|\boldsymbol{\beta})$, that is $\mu(\boldsymbol{x}) = \boldsymbol{\Psi}^{\mathrm{T}}(\boldsymbol{x})\boldsymbol{\beta}$, where the vectors $\boldsymbol{\Psi}(\boldsymbol{x}) \in \mathbb{R}^q$ contains q predefined basis functions.

Given the prior Gaussian assumption in Eq. (3), the GP model parameters $\boldsymbol{\beta}$ and $\boldsymbol{\theta}$ are tuned to optimally fit the observed data. Here, the optimal fit requires the GP to be a Best Linear Unbiased Prediction (BLUP) [77], i.e., the prediction needs to be linear ($\hat{Y}_{\mathrm{L}}(\boldsymbol{x}) = \boldsymbol{c}^{\mathrm{T}}\mathcal{Y}$ with $\boldsymbol{c} = \boldsymbol{c}(\boldsymbol{x}) \in \mathbb{R}^{N_d}$ and $\mathcal{Y} \in \mathbb{R}^{N_d}$ is the observed output vector), unbiased ($\mathbb{E}[\hat{Y}_L(\boldsymbol{x}) - \hat{\mathcal{M}}^{\mathrm{GP}}(\boldsymbol{x})] = 0$), and best ($\min_{\boldsymbol{c}} \mathrm{MSE}(\boldsymbol{x}) = \mathbb{E}[(\hat{Y}_L(\boldsymbol{x}) - \hat{\mathcal{M}}^{\mathrm{GP}}(\boldsymbol{x}))^2]$).[2] Given these constraints, the GP predictor has the analytical form [77]:

$$\begin{aligned}\hat{\mathcal{M}}(\boldsymbol{x}) &= \boldsymbol{\Psi}^{\mathrm{T}}(\boldsymbol{x})\hat{\boldsymbol{\beta}} + \mathbf{r}^{\mathrm{T}}(\boldsymbol{x}|\boldsymbol{\theta})\mathbf{R}^{-1}(\mathcal{Y} - \mathbf{F}\hat{\boldsymbol{\beta}}) \\ &= \boldsymbol{\Psi}^{\mathrm{T}}(\boldsymbol{x})\hat{\boldsymbol{\beta}} + \mathbf{r}^{\mathrm{T}}(\boldsymbol{x}|\boldsymbol{\theta})\hat{\boldsymbol{\gamma}}\,,\end{aligned} \tag{4}$$

where $\hat{\boldsymbol{\beta}} = \hat{\boldsymbol{\beta}}(\boldsymbol{\theta})$ is the generalized least-squares estimation of $\boldsymbol{\beta}$, and other symbols are defined as: $[\mathbf{r}(\boldsymbol{x}|\boldsymbol{\theta})]_{i,1} = R(\boldsymbol{x}, \boldsymbol{x}_i^d|\boldsymbol{\theta})$, $[\mathbf{R}]_{i,j} = R(\boldsymbol{x}_i^d, \boldsymbol{x}_j^d|\boldsymbol{\theta})$ and $[\mathbf{F}]_{i,:} = \boldsymbol{\Psi}^{\mathrm{T}}(\boldsymbol{x}_i^d)$. The coefficients $\hat{\boldsymbol{\gamma}} \in \mathbb{R}^{N_d}$ are constrained by requiring exact interpolations at each design point, i.e., $\mathbf{R}\hat{\boldsymbol{\gamma}} = \boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon} = \mathcal{Y} - \mathbf{F}\hat{\boldsymbol{\beta}}$ are the regression residuals. The unknown hyper-parameters $\boldsymbol{\theta}$ can be calibrated by maximum likelihood estimation or cross-validation [77].

In Eq. (4), the GP predictor is decomposed as a global predictor $\hat{\mathcal{M}}^{\mathrm{G}}(\boldsymbol{x}) = \boldsymbol{\Psi}^{\mathrm{T}}(\boldsymbol{x})\hat{\boldsymbol{\beta}}$ and a local predictor $\hat{\mathcal{M}}^{\mathrm{L}}(\boldsymbol{x}) = \mathbf{r}^{\mathrm{T}}(\boldsymbol{x}|\boldsymbol{\theta})\hat{\boldsymbol{\gamma}}$. Both individual predictors are expressed as a dot product between a basis function vector ($\boldsymbol{\Psi}(\boldsymbol{x})$ or $\mathbf{r}(\boldsymbol{x}|\boldsymbol{\theta})$) and a coefficient vector ($\hat{\boldsymbol{\beta}}$ or $\hat{\boldsymbol{\gamma}}$). Due to the flexibility of basis functions $\mathbf{r}(\boldsymbol{x}|\boldsymbol{\theta})$, a constant trend is assigned to the GP in this study; thus GP degenerates to a local predictor.

Classical PCE is a global predictor by setting the process variance σ^2 in Eq. (3) to zero and eliminating the local terms in Eq. (4). PCE is motivated by spectral approximations, where the black-box model is approximated using a series of multivariate orthogonal polynomials with respect to the input PDF. A notable advantage of PCE over GP is its ability to handle higher-dimensional input spaces. This capability stems efficient coefficient estimation techniques, such as least-square regression, or least angle regression [75]. In contrast, GP requires hyper-parameter tuning for the covariance kernel, limiting its scalability to higher dimensions. Although most test problems (see Section 4.1.1) in this study involve fewer than 20 dimensions, it is important to acknowledge PCE's capacity for addressing high-dimensional problems.

PCK (PCE-Kriging) is interpreted as a GP/Kriging model with PCE as the trend function $\mu(\boldsymbol{x}|\boldsymbol{\beta})$ in Eq. (3). The construction of PCK involves two steps: (i) select an optimal set of multivariate polynomials for the trend functions, and (ii) use the standard calibration procedure of GP to train PCK. As a result, PCK shares the same formulation as GP in Eq. (4) [22].

In this study, both GP and PCK provide the built-in error/prediction uncertainty measure $\sigma^2_{\hat{\mathcal{M}}}(\boldsymbol{x})$ due to the Gaussian process assumption. PCE requires other techniques to estimate the surrogate errors, such as bootstrap resampling strategy [78], k fold cross-validation [79], semi-variance [80], gradient-based variance [14]. This study uses the bootstrap resampling method proposed in [78] to estimate the local error $\sigma^2_{\hat{\mathcal{M}}}(\boldsymbol{x})$ of PCE.

2.3. Error measure

We adopt an error measure proposed in [1] for measuring the discrepancy distance between two CDFs, denoted as $F_Y(y)$ and $\hat{F}_Y(y)$. This metric has a symmetry property in evaluating CDFs and CCDFs simultaneously. In addition, it emphasizes the contribution from the tails of the full distribution. The measure is formulated as

$$\epsilon_F(F_Y(y), \hat{F}_Y(y)) = \frac{1}{y_{\max} - y_{\min}} \times \int_{y_{\min}}^{y_{\max}} w(y)dy, \tag{5}$$

and

$$w(y) = \frac{\left|F_Y(y) - \hat{F}_Y(y)\right|}{\min\left[F_Y(y), 1 - F_Y(y)\right]}, \tag{6}$$

where the integrand $w(y)$ measures the absolute relative distance between $F_Y(y)$ and $\hat{F}_Y(y)$ at position y. The integral in Eq. (5) accounts for the accumulated errors on the integral range $[y_{\min}, y_{\max}]$. The error measure is finally defined as the integration of $w(y)$ divided by $(y_{\max} - y_{\min})$, i.e., the average of relative absolute errors over the range of interest. This error measure is used to assess the accuracy of surrogates and construct stopping criteria in the next subsection.

2.4. Stopping criteria

The stopping criteria are used to terminate the iterative training of sequential surrogates. Conservative stopping criteria could result in redundant DoE points, while relaxed criteria lead to inaccurate surrogates. Similar to [69], this work investigates two classes of stopping criteria related to the convergence trend of the error measure in Eq. (5).

The first type is named the static-based criteria. The criteria measure the stability of the estimated CDF in successive iterations and it is written as:

$$\epsilon_S = \epsilon_F(\hat{F}_Y^{(i-1)}(y), \hat{F}_Y^{(i)}(y)) \le \epsilon_S^{tol}, \tag{7}$$

where the $F_Y^{(i)}(y)$ is the estimated CDF at ith iteration of sequential surrogates. This metric sums up the relative error over the range of interest $[y_{\min}, y_{\max}]$. This iteration stops when the condition in Eq. (7) is satisfied.

The second type is named as variance-based criteria which were originally investigated in [1]. This criterion stops the training when the confidence bounds of the estimated full distribution is sufficiently narrow. This criterion requires an estimate of the prediction error and is defined as:

$$\epsilon_V(\hat{F}_Y^{+}(y), \hat{F}_Y^{0}(y), \hat{F}_Y^{-}(y)) = \frac{1}{y_{\max} - y_{\min}} \times \int_{y_{\min}}^{y_{\max}} \frac{\left|\hat{F}_Y^{+}(y) - \hat{F}_Y^{-}(y)\right|}{\min\left[\hat{F}_Y^{0}(y), 1 - \hat{F}_Y^{0}(y)\right]} dy \le \epsilon_V^{tol}, \tag{8}$$

where $\hat{F}_Y^{+}(y)$, $\hat{F}_Y^{0}(y)$ and $\hat{F}_Y^{-}(y)$ are the CDFs calculated by the bounded surrogates $\hat{\mathcal{M}}^k(x) = \hat{\mathcal{M}}(x) + k\sigma_{\hat{\mathcal{M}}}(\boldsymbol{x})$ with k setting as 2, 0, -2, respectively. Eq. (8) is still developed based on the error measure in Eq. (5).

For both stopping criteria, we can enhance their robustness by requiring them to be consecutively met for multiple iterations, such as 2 or 3 times. In summary, the thresholds ϵ_S^{tol} in Eq. (7), ϵ_V^{tol} in Eq. (8), and the required number of consecutive met criteria are hyper-parameters in this study. Section 4.4 discusses the tuning of these parameters.

2.5. Active learning for DoE enrichment

AL strategies for DoE enrichment are based on an iterative process. Specifically, a single sample is chosen at each step to maximize the learning function. The surrogate model is then retrained, and the training set is updated. This process is repeated until specific stopping criteria are met. The learning function completely defines the active learning strategy.

This subsection introduces three learning functions used in this paper. For simplicity, here, we report a brief yet self-contained description. More details are provided in the supplementary materials (see Appendix A). The three learning functions are:

[2] In our study, the GP predictor is interchangeable with the Kriging predictor as defined in [76,77]. Some studies (e.g., [1]) refer to the GM predictor as the mean of the posterior distribution of Eq. (3) conditional on observations $\mathcal{X}$ and $\mathcal{Y}$. While both methods offer the same prediction to unexplored points, their prediction errors/uncertainties differ.

- Maximum of Variance (MoV) learning function.
- Two-step learning function [1].
- Gradient-based learning function [14,72].

The MoV learning function relies on the maximum variance criterion. Specifically, the best point is the one with the highest variance, $\sigma^2_{\mathcal{M}}(\boldsymbol{x})$. In GP- and PCK-based surrogate models, the variance is explicitly known, whereas, in classical PCE, it is estimated through bootstrap resampling techniques.

The two-step learning function proposed in [1] includes two steps to select the optimal samples. In the first step, a threshold y^* is selected, which contributes the most errors/uncertainties to the full distribution estimation. In the second step, a candidate point is selected using a given learning function (e.g., a reliability-based AL function [13,41]). Notably, this strategy is "mesh-free" as it does not rely on a fixed discretization of the distribution function.

The gradient-based learning function considers a trade-off between global exploration and local exploitation. Global exploration involves investigating a wide area to discover strongly nonlinear regions. Local exploitation focuses on these specific regions to select the best possible samples. Specifically, the global exploration searches the regions where the DoE in design space is sparse. The local search strategy leverages gradient information to exploit locations characterized by pronounced nonlinearity (which are inherently more unpredictable). To achieve a balance between exploration and exploitation, a weighting function is introduced. The gradient-based learning function was first studied in [14]. In this work, we adopt a variation introduced in [72], where the gradient of the surrogate, rather than that of the original simulator, is used.

3. Uniform design based on maximin distance

This section develops a uniform experimental design method, termed *pool-based maximin distance design*, for constructing global surrogates. First, the existing maximin distance design method is presented. Then, this technique is adapted by incorporating the pool-based representation approach. Finally, the modified method is compared with two classical uniform design methods.

3.1. Original maximin-distance design

Maximum–minimum (maximin) distance design was originally proposed to uniformly allocate samples in the unit hypercube space [30]. Consider a design space $\mathcal{D}_{\boldsymbol{x}} \subset \mathbb{R}^N$ and a set of design input $\mathcal{X} = \{\boldsymbol{x}_i^d \in \mathcal{D}_{\boldsymbol{x}} | i = 1, 2, \dots, N_d\}$. $\mathcal{X}^*$ is the maximin-distance design if

$$\mathcal{X}^* = \arg\max_{\mathcal{X}} \left(\min_{j \neq i} d(\boldsymbol{x}_i^d, \boldsymbol{x}_j^d) \right), \tag{9}$$

where $d(\cdot,\cdot)$ is the Euclidean distance measure. Eq. (9), first, implements the "min" operator to calculate the minimum distance among distinct points in $\mathcal{X}$. Then, the "arg max" operator searches for a design $\mathcal{X}^*$ that maximizes this minimum distance. Here, the optimality implies that the distances from each point $\boldsymbol{x}_i^d \subset \mathcal{X}$ to its nearest point are equal, thus the design space is uniformly filled.

The objective function in Eq. (9) is highly nonlinear and non-convex with $N_d \times N$ variables [38]. When $N_d \times N$ is large, the optimization problem is computationally unfeasible. Moreover, extending the maximin-distance design to design spaces with non-box and no-compact sets is challenging. A feasible solution is given by solving the optimization problem sequentially [38]. The idea lies in decomposing the original problem into sub-optimization problems and solving them sequentially. Each sub-optimization problem focuses on optimizing a small number of points based on the maximin-distance principle. Next, the optimized points from previous sub-problems are fixed for the next sub-problems. In a special case, we can optimize a single point in each subproblem. This strategy is straightforward, and the next subsection introduces the combination with the pool-based representation to further simplify the optimization process.

3.2. Pool-based maximin-distance design

To build surrogate models, the method AK-MCS [13] utilized a combination of the AL and the candidate-pool-based representation. The candidate pool can represent the design space by using MCS to simulate a set of candidate points, among which AL techniques are applied to identify the optimal points. This study introduces a sequential maximin-distance design approach applied to an MCS pool, motivated by the following ideas:

- **Computational efficiency:** Representing the design space as a pool of candidate samples transforms the continuous optimization problem (Eq. (9)) of optimizing positions into a discrete optimization problem of selecting N_d points in the pool, leading to significant computational savings.
- **Integration with input PDFs:** The pool-based method implicitly accounts for the distributions of the input variables, enabling the surrogate model training to concentrate on regions with non-negligible probabilities.
- **Improved projection properties:** While discretization and sequential sampling may reduce the uniformity of the input DoE, they enhance the uniformity of projection onto subspaces when compared to the original maximin-distance design, as documented in [81].
- **Robustness to design space boundaries:** Representing the design space by samples makes the optimization process insensitive to the complexity of the design space boundaries.
- **Compatibility with active learning:** The pool-based method allows for a sequential enrichment of the DoE, where the optimal size of the DoE is not predetermined.

The proposed *pool-based maximin-distance design* is reported in Algorithm 2.

Algorithm 2: Procedure of pool-based maximin-distance design

1. *Generate a candidate pool* $S = \{\boldsymbol{x}_i^S | i = 1, 2, .., N_{\text{MCS}}\}$. Note that $N_{\text{MCS}} >> N_{\text{d}}$.
2. *Normalize the candidate pool*. Apply component-wise Z-score normalization to each sample in S, obtaining $\tilde{S} = \{\tilde{\boldsymbol{x}}_i^S | i = 1, 2, .., N_{\text{MCS}}\}$.
3. *Build an initial DoE*. Choose a random point from $\tilde{S}$ to obtain the current DoE $\tilde{\mathcal{X}} \leftarrow \{\tilde{x}_1^d\}$ and set $N_d \leftarrow 1$.
4. *Enrich the DoE by sequential maximin-distance design*. For each iteration, chooses a point in $\tilde{S}$ with the maximum distance to $\tilde{\mathcal{X}}$, i.e.,

   $$\tilde{x}^* = \arg\max_{\boldsymbol{x}_i \in \tilde{S}} \left(\min_{\boldsymbol{x}_j \in \tilde{\mathcal{X}}} d(\boldsymbol{x}_i, \boldsymbol{x}_j) \right).$$

   Then, $\tilde{\mathcal{X}} \leftarrow \tilde{\mathcal{X}} \cup \{\tilde{x}^*\}$, $N_d \leftarrow N_d + 1$. Repeat this enrichment process until N_d reaches a prescribed number.
5. *Reverse the normalization*. Perform inverse Z-score normalization to obtain $\mathcal{X}$.

Algorithm 2 entails three important aspects:

- **Parallel processing:** The maximin-distance design can add multiple points to a DoE without evaluating the simulators $\mathcal{M}(\cdot)$. This feature can be especially advantageous when parallel computing resources are available to construct surrogates.
- **Markov Chain Monte Carlo-based candidate pool generation:** The candidate pool can be generated using Markov Chain Monte

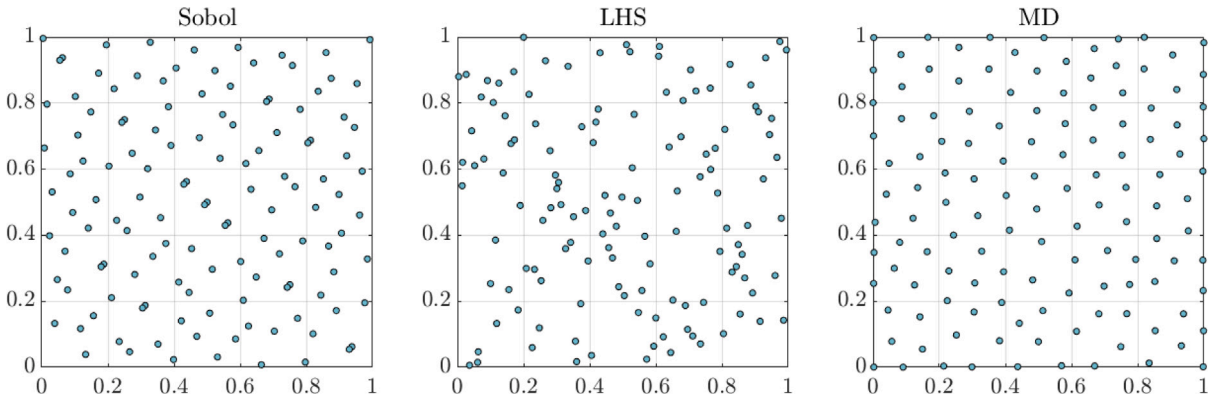


Fig. 1. DoEs in the unit cube using three uniform design methods. *The sample size is 128*.

Carlo techniques, when the input distribution contains an unknown normalizing constant. This is relevant in Bayesian problems.

- **Deterministic design:** If the candidate pool in Step 1 and the initial point in Step 3 are fixed, this method transforms into a deterministic design, ensuring a consistent DoE that can be used to compare the performance of different surrogate models.

3.3. Comparative study of uniform designs

This subsection compares the pool-based maximin-distance design (referred to as MD hereafter) with two other uniform designs: Latin Hypercube Sampling (LHS) [36] and Sobol sequence design (Sobol) [32]. The comparison involves visually inspecting their uniformity in two 2-dimensional design spaces: a unit cube space and a bi-variate standard normal space. Additionally, we compare their performance in assisting surrogate modeling for full distribution estimation. For MD, a pool size of 10^5 is used.

Firstly, we compare the samples generated by the three uniform design methods in the 2-dimensional unit cube. The results are shown in Fig. 1. Sobol sequence demonstrates good global uniformity but exhibits some local clustering along the diagonals. The samples of LHS appear fairly random and lack uniformity. Meanwhile, MD outperforms Sobol sequence and LHS in terms of uniformity and allocates more points along the boundaries.

Next, a comparison is conducted in the standard normal space. In this case, Sobol sequence and LHS generate samples through iso-probability transformations from the unit cube space to the standard normal space. For MD, the samples are selected from a pool of 10^5 random bi-variate standard normal samples. The DoEs and the pool of 10^5 samples are shown in Fig. 2. It is observed that samples from the Sobol sequence and LHS tend to concentrate around the mode. This is expected because they are quasi-Monte Carlo techniques designed to mimic the distribution. In contrast, MD sampling is designed for training surrogate models, thereby distributing points more uniformly. This feature is crucial for the construction of global surrogates to explore both the mode and the tail regions.

Finally, we compare the three uniform design methods in terms of surrogate modeling accuracy using an expanding DoE, as illustrated in Fig. 3. This comparison uses a 6-dimensional analytical model with multiple types of input distributions: the normal, Weibull, and uniform. Details of this computational model are provided in benchmark example #11, as referenced in Appendix A. Since MD and LHS are stochastic methods, the prediction variability is evaluated based on ten independent runs. The surrogate model is a Gaussian process with a constant trend and a Matérn-5/2 kernel. The figure indicates that MD leads to highly accurate and efficient surrogate modeling. Given this superior performance in building surrogates, we leverage MD as a baseline to assess the potential improvements of active learning over uniform designs.

4. Comparative study

4.1. Preliminary work

4.1.1. Benchmark problems set-up

The benchmarks consist of 16 examples drawn from the literature. Most of the examples are formulated in analytical forms, with the exception of three that employ FEM models. The FEM models include a 21-bar truss structure [82], a 3-bays-5-floor frame structure [83], and a 10-story shear structure [84]. The number of random variables varies from 2 to 21, covering the low- and medium-dimensional problems. A detailed list of the benchmarks, including the performance functions, probabilistic models, and the ranges of QoIs $[y_{\min}, y_{\max}]$ are summarized in the supplementary material (see Appendix A). In this study, the range $[y_{\min}, y_{\max}]$ is defined as $[F_Y^{-1}(10^{-k}), F_Y^{-1}(1-10^{-k})]$, where $F_Y^{-1}(\cdot)$ is the inverse function of the target CDF, and we set $k = 3$. This range contains 99.98% of the probability mass, ensuring accurate estimation of lower statistical moments and satisfying the needs of most engineering applications. Notice that these bounds are computed using an empirical CDF obtained from MCS. However, in real-life examples, the target distribution is not available. In this case, [1] provides algorithmic solutions to estimate $[y_{\min}, y_{\max}]$.

4.1.2. Algorithmic settings

This subsection describes in detail the parameters and configurations settings related to Algorithm 1. In *Step 1*, the initial DoE is generated by the pool-based MD. This choice is motivated by [85], which shows that space-filling sampling for initial DoEs is more robust than random sampling methods. In particular, an MCS population of size 10^{k+2} is drawn, from which an initial DoE is extracted using the pool-based MD. The MCS population is later used for computing statistics of interest, with the parameter $k + 2$ chosen to ensure the coefficient of variation of tail estimations at approximately 10%. The size of the initial DoE is set as $\max\{12, 3N\}$, where N denotes the dimension of the problem. We set $3N$ so that there are at least 3 points for each direction to capture possible nonlinearities.[3] In *Step 2*, the settings of the surrogate models, GP, PCE, and PCK, are specified based on [69]. In *Step 3*, the range of interest $[y_{\min}, y_{\max}]$ is equally divided into 100 intervals to compute the full distribution as outlined in Eqs. (1) and (2). It follows that the CDF and CCDF are simultaneously estimated using the surrogate trained in step 2 and the MCS samples generated in step 1. In *Step 4*, the stopping criteria module consists of two ways to terminate the iterative process: the first is introduced in Section 2.4,

[3] If the computational cost is too high, one might opt for $2N$.

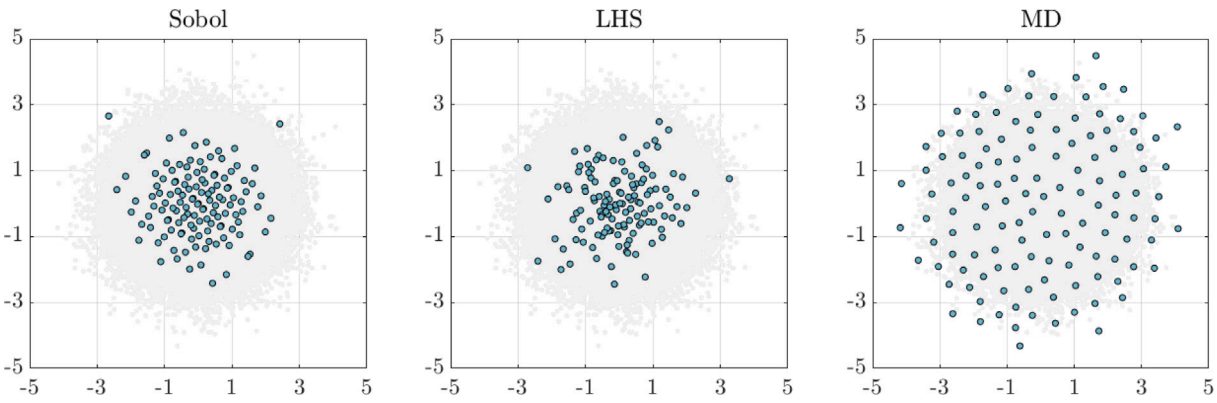


Fig. 2. DoEs in the standard normal space using three uniform design methods. *The sample size is 128.*

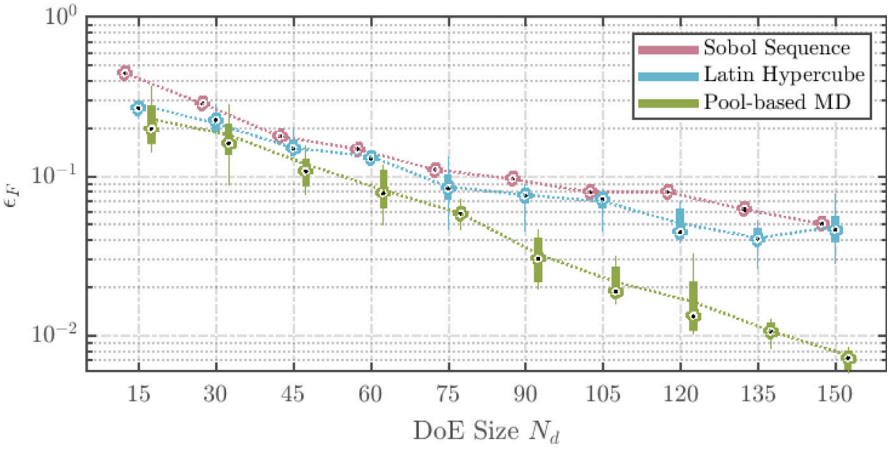


Fig. 3. A comparison of surrogate modeling accuracy for distribution estimation using different uniform designs. *The error measure ϵ_F is described by Eq. (5).*

and the second sets the maximum DoE size to $\min\{100 + N \times 20, 300\}$. The first criterion entails hyper-parameters such as thresholds and the number of consecutive triggers for the stopping criterion. Hence, we employ only the second criterion to stop the algorithm, facilitating the collection of data to analyze the optimal hyper-parameter settings for the first stop criterion. Additionally, we have increased the maximum model evaluations for Examples #3, #5, and #6 to 300 to accommodate their slower convergence rates. In *Step 5*, the DoE enrichment module contains three AL methods introduced in Section 2.5, i.e., the maximum of variance (MoV), the two-step learning function (t-LF), and the gradient-based learning function (g-LF), and the uniform design method introduced in Section 3. Notably, these enrichment methods do not require any hyper-parameter tuning.

To sum up, the 16 benchmark examples are solved by 12 aggregated strategies, which are designed by integrating 3 surrogates with 4 DoE enrichment methods. Each of the 12 strategies is independently run 10 times to solve the benchmarks, resulting in a total of 1920 UQ analyses to support the comparative study.

4.1.3. Criteria for ranking the strategies

In order to identify the "optimal" configurations in each module, this study carries out rankings based on the statistical analysis over the 1920 UQ analyses. The comparative indicators include two metrics regarding the accuracy and efficiency/cost. The accuracy metric relies on the error measure ϵ_F in Eq. (5), and the efficiency/cost metric is formulated based on the number of model evaluations. The adopted comparative indicators vary with different modules. The *surrogate module* and *DoE enrichment module* are evaluated based on the *mean* error measure that averages the results of the 10 repeated runs. The metrics for these two modules include:

- *Accuracy metric*: Given fixed iterations (i.e., the maximum size of the DoE prescribed in Section 4.1.2), compute the percentage of the cases that the average error measure is smaller than a prescribed tolerance ϵ_F^{tol}.
- *Efficiency metric*: Given an error tolerance ϵ_F^{tol}, compute the percentage of the cases that the selected methods use the minimum average iterations to converge (i.e., reach $\epsilon_F \le \epsilon_F^{\text{tol}}$).

The *Stopping criteria module* is used to terminate the iterative process. Each stopping criterion is configured to halt the surrogate training at a specific iteration, returning the required number of model evaluations $N_{\mathcal{M}}$ and the corresponding error measure ϵ_F. The optimal stopping criterion is the one that stops the iteration while satisfying the accuracy demand, i.e., $\epsilon_F \le \epsilon_F^{\text{tol}}$. In other words, the surrogate training achieves convergence using the minimum number of model evaluations $N_{\mathcal{M}}^{\min}$. The comparative indicators for this module are defined as follows:

- *Accuracy metric*: Given an error tolerance ϵ_F^{tol}, compute the percentage of the cases (over the 1920 analyses) that accuracy demand is met, i.e., $\epsilon_F \le \epsilon_F^{\text{tol}}$.
- *Cost metric*: Given an error tolerance ϵ_F^{tol}, the cost is defined as the mean value (over the 1920 analyses) of the normalized number of model evaluations, i.e., $N_{\mathcal{M}}/N_{\mathcal{M}}^{\min}$.

If the cost metric is too small (practically smaller than 1), the stopping criteria likely indicates a premature surrogate that fails to meet the

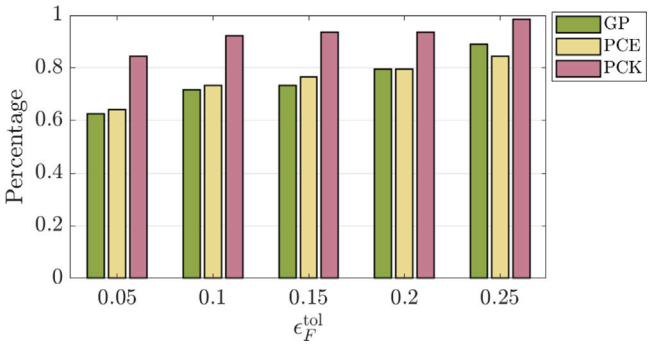


Fig. 4. Comparison of surrogate models in terms of accuracy. *The accuracy metric uses the percentage of cases (out of the 64 cases) that the surrogate model can converge (i.e., $\epsilon_F \le \epsilon_F^{tol}$) within a fixed maximum number of iterations.*

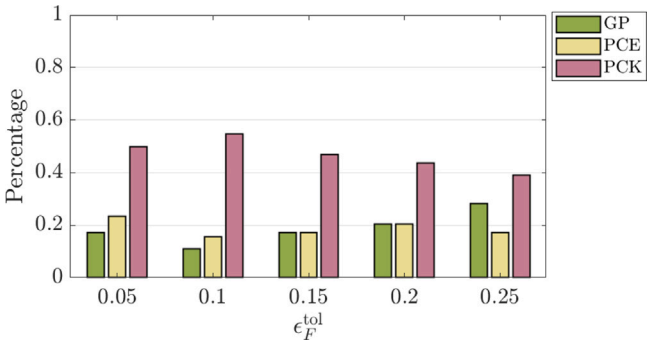


Fig. 5. Comparison of surrogate models in terms of efficiency. *The efficiency metric uses the percentage of cases (out of the 64 cases) that the surrogate model requires the minimum iterations to converge (i.e., $\epsilon_F \le \epsilon_F^{tol}$).*

accuracy requirement. Conversely, if the cost metric is too large, it likely suggests that the stopping criterion is strict, requiring significantly more model evaluations than necessary to achieve the desired accuracy.

4.2. Compare the performance across surrogate models

In this subsection, we compare the performance of the surrogates prescribed with different target accuracy tolerances ϵ_F^{tol}, which accounts for different accuracy demands in practical applications. The set $\epsilon_F^{\text{tol}} \in \{0.05, 0.1, 0.15, 0.20, 0.25\}$ is considered, while 0.05 corresponds to an accurate model and 0.25 a rough estimation.

As a reminder, this work carries out a benchmark study consisting of 16 examples solved by 12 different solution strategies. These strategies are designed by combining the two independent components—3 surrogate models and 4 DoE enrichment methods. The performance of each surrogate is assessed through the 64 cases (16 examples × 4 DoE enrichment methods).

First, the accuracy metric is applied to rank the three surrogates. This metric starts with counting the number of occurrences of $\epsilon_F \le \epsilon_F^{\text{tol}}$ among the 64 cases. This procedure is repeated by changing the thresholds from 0.05 to 0.25. Then the percentage of cases (out of the 64 cases) that the surrogates can converge is summarized. The results are shown in Fig. 4. We observe that, for all surrogates, the accuracy metric increases with the increase of ϵ_F^{tol}. This phenomenon is expected since constructing surrogates with lower accuracy is easier. In Fig. 4, surrogate PCK consistently shows the highest accuracy, which converges in 84% of the tested cases when $\epsilon_F^{\text{tol}} = 0.05$ and this ratio increases to 98% when $\epsilon_F^{\text{tol}} = 0,25$. GP is slightly less effective than the other two surrogates when $\epsilon_F^{\text{tol}} < 0.2$, and its relative performance improves when $\epsilon_F^{\text{tol}} = 0.25$.

Next, the surrogates are compared across the 64 cases regarding the efficiency metric. The surrogate using the least number of model evaluations to converge (i.e., $\epsilon_F \le \epsilon_F^{\text{tol}}$) is labeled as the winner. Then, we compute the proportion of cases in which the given surrogate wins the comparisons. The results are presented in Fig. 5. PCK is again the best choice in terms of efficiency with different ϵ_F^{tol}. GP outperforms PCE when $\epsilon_F^{\text{tol}} = 0.25$, but it becomes comparable to PCE when ϵ_F^{tol} is set as 0.15 or 0.20. Besides, GP performs less accurately than PCE when ϵ_F^{tol} goes down to 0.1 or 0.05.

From the above comparisons, we observe that for most cases PCK outperforms the other two in terms of accuracy and efficiency. To reveal more details regarding the comparison, we decompose the aggregated results in Fig. 5 (with $\epsilon_F^{tol} = 0.1$) into detailed comparisons in Table 1. The table presents the average number of model evaluations required for convergence across the 12 solutions from the 16 examples. The results indicate that the best surrogate configuration varies with the benchmark examples, surrogate types, and enrichment methods of experimental design. To highlight these differences, we specifically visualize the column of "MD" in Table 1, as shown in Fig. 6 It is observed that the best surrogate model varies with the problem. For example, GP is the most efficient option for Examples #1, PCE outperforms the others in Examples #7, #11, #13, #14, and PCK shows the best performance in the remaining examples. It is also observed that the optimal surrogate model sometimes only brings marginal improvement over the alternatives, such as in Examples #1, #8 ∼ #11, and #13, while sometimes it is significantly better, such as in Examples #2, #3, #5 ∼ #7, #15, and #16. In general, without prior knowledge about the problems, PCK would be the most robust choice. Similar observations are found for the three AL design methods (MoV, t-LF, g-LF) in Table 1. For results of setting other thresholds ($\epsilon_F^{tol} = 0.05, 0.15, 0.20, 0.25$), the required model evaluation are additionally attached to the supplementary material (see Appendix A)

4.3. Compare the performance across DoE enrichment methods

This subsection compares the four DoE enrichment methods: MoV, t-LF, g-LF, and MD. We perform a similar statistical analysis as the previous subsection with respect to accuracy and efficiency metrics. Each DoE enrichment method is evaluated based on 48 cases (16 examples × 3 types of surrogates).

As for the accuracy metric, Fig. 7 shows the percentage of times that a method can converge within the pre-specified maximum number of iterations. This percentage increases with ϵ_F^{tol}. However, for a fixed threshold ϵ_F^{tol}, the variation among the 4 enrichment methods is smaller than that of the 3 surrogate models in Fig. 4. Therefore, we conclude that the accuracy metric is dominated by the surrogate module rather than the DoE enrichment module. In addition, the difference between AL strategies (MoV, t-LF, and g-LF) and uniform design (MD) is surprisingly minor, which implies that in most cases, AL approaches may not significantly improve accuracy compared to uniform designs. To quantify the influence of surrogate types and DoE enrichment methods, a two-way ANOVA is performed across various target accuracy thresholds, i.e., $\epsilon_F^{\text{tol}} \in \{0.05, 0.1, 0.15, 0.20, 0.25\}$. The ANOVA uses linear regression to decompose the total variance (variability in the accuracy metric) into main effects (from surrogate type and DoE enrichment method), interaction effects (combined impact of both factors), and residuals (unexplained variation). The results, summarized in Table 2, show the following trends: (1) The influence of surrogate type becomes more pronounced as the accuracy threshold decreases, and (2) The interaction effect becomes more significant at higher tolerance levels (ϵ_F^{tol}). This ANOVA analysis provides quantitative support for our previous findings.

Fig. 8 provides an assessment regarding the efficiency metric, exhibiting irregular fluctuations in contrast to Fig. 5. It is observed that t-LF is generally the most efficient except when $\epsilon_F^{\text{tol}} = 0.2$, while MoV

Table 1
The average number of model evaluations required to converge ($\epsilon_F \leq \epsilon_F^{\text{tol}} = 0.1$) across 12 different solutions (combining 3 surrogate models and 4 DoE enrichment methods) over 16 benchmark examples. The symbol "–" represents that the solution fails to converge.

eg	GP				PCE				PCK			
	MoV	t-LF	g-LF	MD	MoV	t-LF	g-LF	MD	MoV	t-LF	g-LF	MD
1	13	13	13	13	20	14	15	15	20	17	17	17
2	26	25	26	26	–	–	–	–	24	23	25	24
3	–	–	–	–	–	–	–	–	109	–	103	81
4	27	24	28	27	–	53	39	46	13	14	14	13
5	106	94	111	77	–	–	260	281	49	53	49	43
6	300	–	–	–	–	–	–	–	188	161	208	170
7	–	–	–	–	81	57	41	51	–	–	–	–
8	22	16	18	21	23	26	22	22	22	19	20	18
9	54	25	33	47	35	21	26	30	26	22	25	26
10	80	37	51	51	47	33	31	34	28	29	28	30
11	51	24	35	38	28	25	23	25	25	21	24	27
12	153	153	119	121	133	–	78	93	93	113	69	74
13	89	74	68	77	72	70	66	66	56	68	60	68
14	232	77	119	115	91	63	66	65	69	69	66	67
15	–	264	–	–	44	41	49	56	43	48	48	36
16	–	–	–	–	–	200	235	244	186	232	106	149

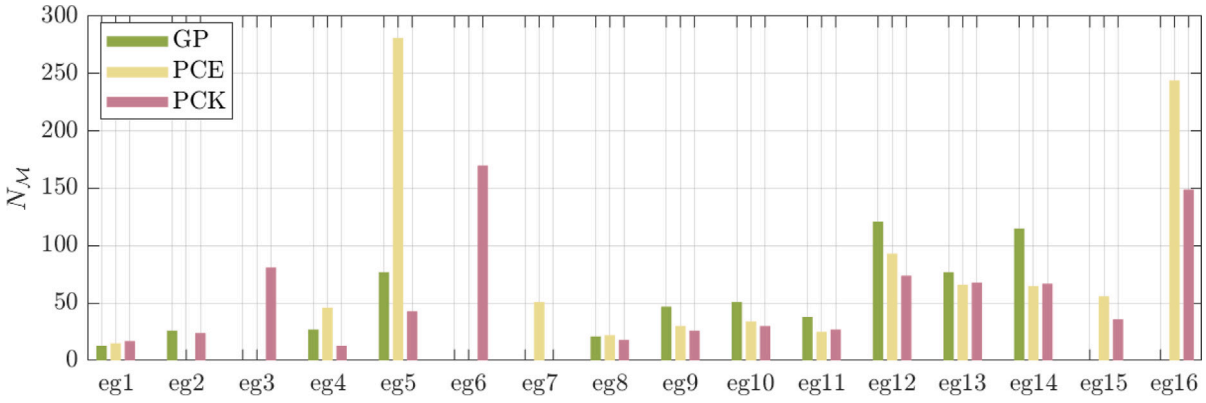


Fig. 6. The average number of model evaluations required to converge ($\epsilon_F \leq \epsilon_F^{\text{tol}} = 0.1$) when using the enrichment method MD. *The bar is not shown if the method fails to converge within the given maximum number of iterations.*

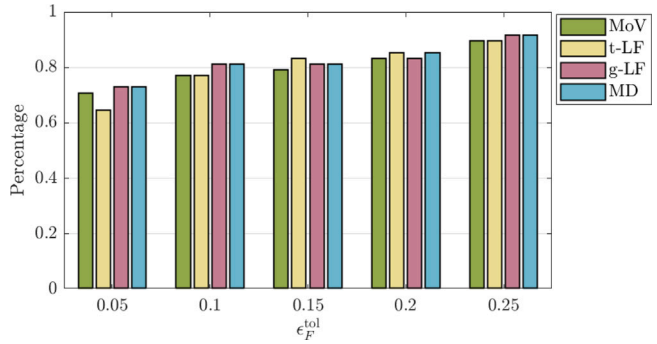


Fig. 7. Comparison of DoE enrichment methods in terms of accuracy. *The accuracy metric uses the percentage of cases (out of the 48 cases) that the DoE enrichment method can converge (i.e., $\epsilon_F \leq \epsilon_F^{tol}$) within a fixed maximum number of iterations.*

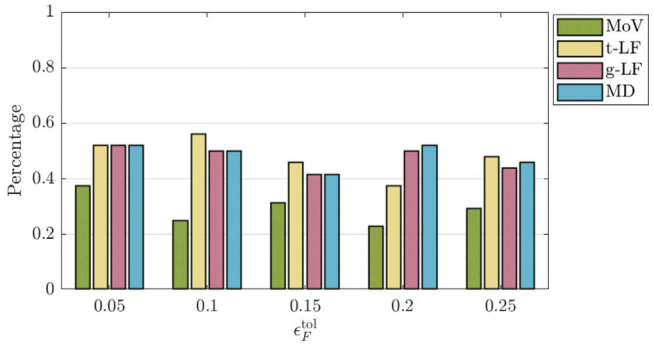


Fig. 8. Comparison of DoE enrichment methods in terms of efficiency. *The efficiency metric uses the percentage of cases (out of the 48 cases) that the DoE enrichment method requires the minimum iterations to converge (i.e., $\epsilon_F \leq \epsilon_F^{tol}$).*

is the least efficient method overall. Besides, the most surprising result is that MD is comparable to AL approaches, it even performs the best when $\epsilon_F^{\text{tol}} = 0.2$. This result again reveals that AL approaches cannot overwhelmingly outperform uniform designs regarding efficiency. Furthermore, both Fig. 7 and Fig. 8 show that the t-LF which integrates information both from the surrogate model and input PDF cannot consistently outperform g-LF which only utilizes the information from the surrogate model. In fact, when ϵ_F^{tol} is set as 0.05, 0.1, or 0.25, t-LF gives a better/comparable performance with respect to efficiency in Fig. 8 but exhibits worse performance with respect to accuracy in Fig. 7. When $\epsilon_F^{\text{tol}} = 0.20$, g-LF outperforms t-LF. Therefore, t-LF does not consistently outperform g-LF.

For a detailed comparison, we further analyze the results in Fig. 8 by focusing on $\epsilon_F^{\text{tol}} = 0.1$. The analysis measures the effect of *AL versus uniform design* for constructing global surrogates. To do this, we split the data into two sets: one set refers to the average number of model evaluations $N_{\mathcal{M}_{i,j}}^{\text{uniform}}$ from the uniform design, and another set refers to $N_{\mathcal{M}_{i,j}}^{\text{AL}}$ from AL approaches, where the subscript denotes the results computed by jth type of surrogate model (GP, PCE and PCK) in the ith example. Since we have 3 AL strategies, $N_{\mathcal{M}_{i,j}}^{\text{AL}}$ refers to the most efficient results among MoV, t-LF, and g-LF. Therefore, a novel metric

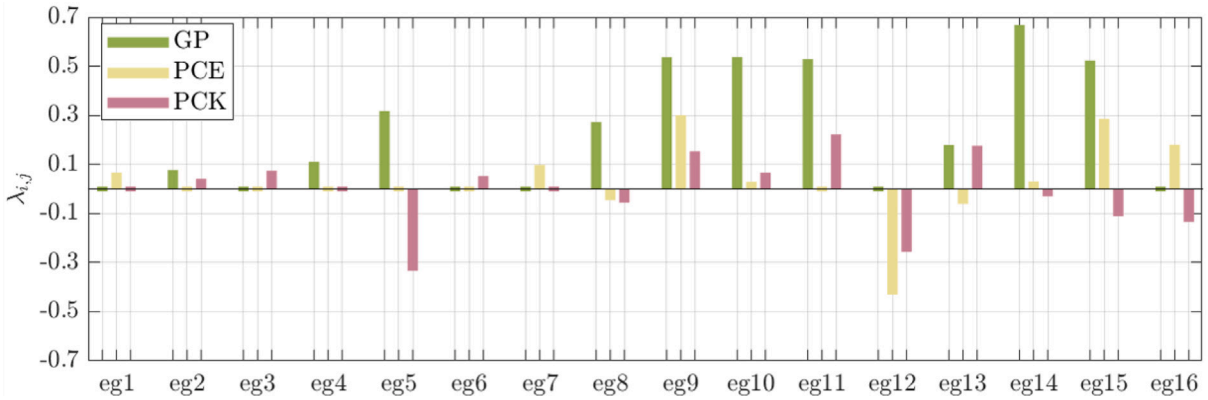


Fig. 9. *AL* versus *uniform* design on constructing global surrogates with $\epsilon_F^{tol} = 0.1$. *Positive* $\lambda_{i,j}$ *denotes AL is effective and negative value represents the opposite.*

Table 2
ANOVA of accuracy metric for surrogate types and DoE enrichment methods.

Factors	Accuracy threshold ϵ_F^{tol}				
	0.25	0.2	0.15	0.1	0.05
Factor A: DoE	20.0%	20.0%	25.0%	21.7%	9.9%
Factor B: surrogate	40.0%	40.0%	43.8%	60.9%	84.1%
Interaction AB	40.0%	40.0%	31.3%	17.4%	6.0%
Residual	0.0%	0.0%	0.0%	0.0%	0.0%

is proposed to measure the benefit of using AL strategies, i.e.,

$$\lambda_{i,j} = \frac{N_{\mathcal{M}_{i,j}}^{\text{uniform}} - N_{\mathcal{M}_{i,j}}^{\text{AL}}}{N_{\mathcal{M}_{i,j}}^{\text{uniform}}}. \tag{10}$$

The positive $\lambda_{i,j}$ implies that the AL approach is more efficient than MD, and the negative represents the opposite. Fig. 9 illustrates the metric $\lambda_{i,j}$ over 16 benchmarks solved by 3 different surrogates. Here the results again illustrate that the AL method is not always superior to the uniform design. The signs of the indicator $\lambda_{i,j}$ vary with the problems and the adopted surrogates. In all 16 examples, GP using AL methods performs the best within 10 cases, but this number drops to 7 for PCE and PCK. Besides, the improved efficiency for GP ranges from 0%~67% in all examples. But this percentage can drop to be negative for PCE (−43%~30%) and PCK (−33%~22%). Hence, GP is more likely to obtain a greater improvement from AL approaches while PCE and PCK are less possible. Furthermore, the benefit from using AL is small compared with the reported results (e.g., [1,14]), where AL techniques have demonstrated the potential to reduce the number of model evaluations by up to 100 times. Meanwhile, the negative effect of adopting AL should not be ignored. To figure out this phenomenon, we will provide an additional investigation in Section 5.

4.4. Stopping criteria settings

This subsection focuses on studying the parameter settings for the two stopping criteria. Specifically, we investigate the best settings of the stopping criteria thresholds and the number of consecutive triggers for stopping conditions. This is achieved by performing a statistical analysis over the 1920 UQ analyses in terms of efficiency and cost metrics. The efficiency metric, similar to that in the previous two subsections, is evaluated on each run. The cost metric is defined as the mean value of the normalized number of model evaluations. The normalized results are obtained by dividing the number of model evaluations by $N_{\mathcal{M}}^{\min}$ (i.e., the minimum number of model evaluations required to reach $\epsilon_F \le \epsilon_F^{\text{tol}}$). Unless otherwise stated, the error tolerance ϵ_F^{tol} is taken as 0.1 in this subsection.

The DoE enrichment process is terminated when the stopping criterion ($\epsilon_S \le \epsilon_S^{\text{tol}}$ or $\epsilon_V \le \epsilon_V^{\text{tol}}$) is triggered within a given number of consecutive iterations. We consider a series of values for the thresholds (i.e., ϵ_S^{tol} and ϵ_V^{tol}) and set the consecutive triggered time(s) as 1, 2, and 3. With different thresholds and the number of consecutive triggers for stop conditions, we respectively compute the accuracy and cost metrics. The results are provided in Fig. 10 for static-based criteria and in Fig. 11 for variance-based criteria. Both stopping criteria show a monotonic pattern in which both accuracy and cost metrics decrease with the increased thresholds. In addition, for a fixed error tolerance, increasing the number of consecutive triggers for stopping conditions leads to more accurate results at the expense of higher cost.

To reveal more details, we combine the Accuracy-axis and Cost-axis of two criteria into one figure, as shown in Fig. 12. Firstly, we compare the effects of threshold settings of two stopping criteria. The same trends can be found in Figs. 12(a) and 12(b). For both stopping criteria, there exists a cost-efficient point beyond which increasing the cost does not significantly improve accuracy. However, some differences between the two stopping criteria can be observed. In the beginning, the accuracy grows almost linearly with respect to the increased cost. The initial slope for the static-based criteria is larger than that of the variance-based criteria. As the cost increases, the slope of the static-based criteria reduces faster than that of the variance-based criteria. Next, we compare the effect of the prescribed number of consecutive triggers for both stopping criteria. It is observed from Fig. 12 that their effects are marginal when the accuracy metric is smaller than 0.6. Beyond this point, setting the number to 2 or 3 is more cost-efficient than setting it to 1. Therefore, we recommend setting the number of consecutive triggers to be larger than 1 if the error threshold is relatively tight.

To balance the cost and accuracy, we introduce the computational details on the *cost-efficient point*, highlighted in Fig. 12, to determine the "best" thresholds for the stopping criteria. First, we define a *cutoff cost* as the position where the initial and final slope of the Cost-Accuracy curve meet. The initial slope refers to the tangent line through the starting point of the curve, while the final slope corresponds to the horizontal line reaching maximum accuracy. Then, the projection of the cutoff cost on the Cost-Accuracy curve is defined as the cost-efficient point. The coordinates of the cost-efficient points are $(1.47, 0.548)$ for ϵ_S in Fig. 12(a) and $(1.86, 0.673)$ for ϵ_V in Fig. 12(b). These two coordinates suggest that using the criterion ϵ_V produces accurate estimations but requires more cost. In contrast, using the criterion ϵ_S needs less computational costs but obtains less precise results.

Based on the cost-efficient points, we can locate the balanced thresholds on the curves in Figs. 10 and 11. This procedure can be generalized to other error tolerance values. In Table 3, we give the recommended thresholds for the two stopping criteria based on the concept of cost-efficient points.

To better understand the convergence trend across different examples, the accuracy curve (triggered time: 3) in Fig. 10 is decomposed

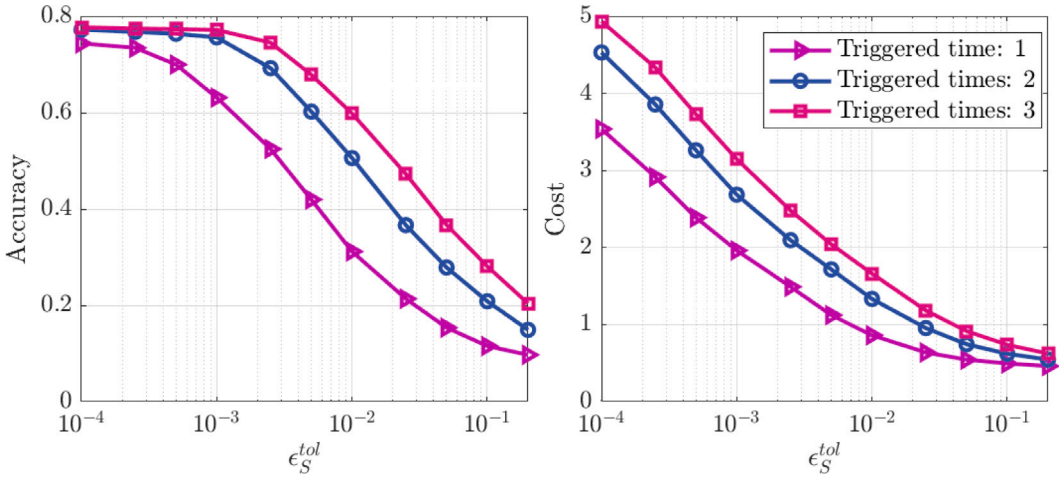


Fig. 10. The influence of static-based stop criterion parameters on the accuracy and efficiency metric.

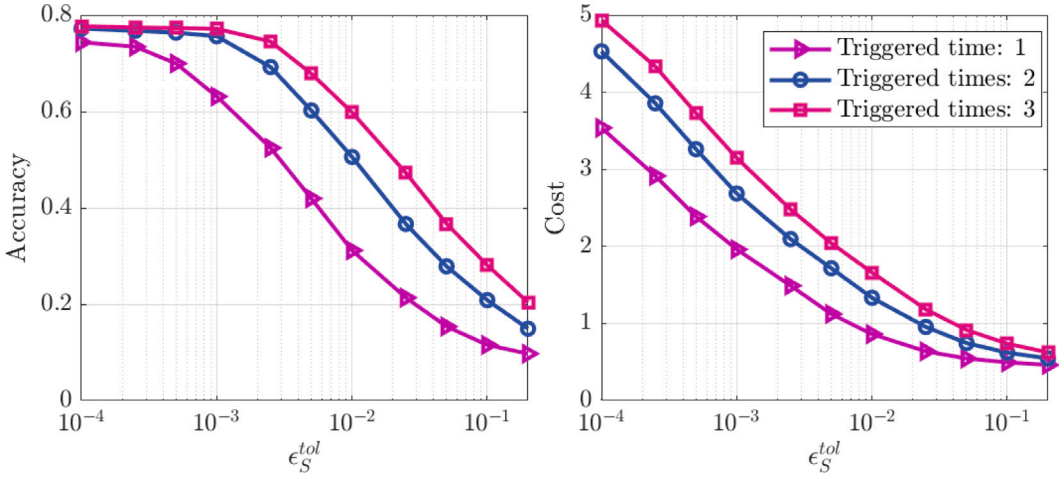
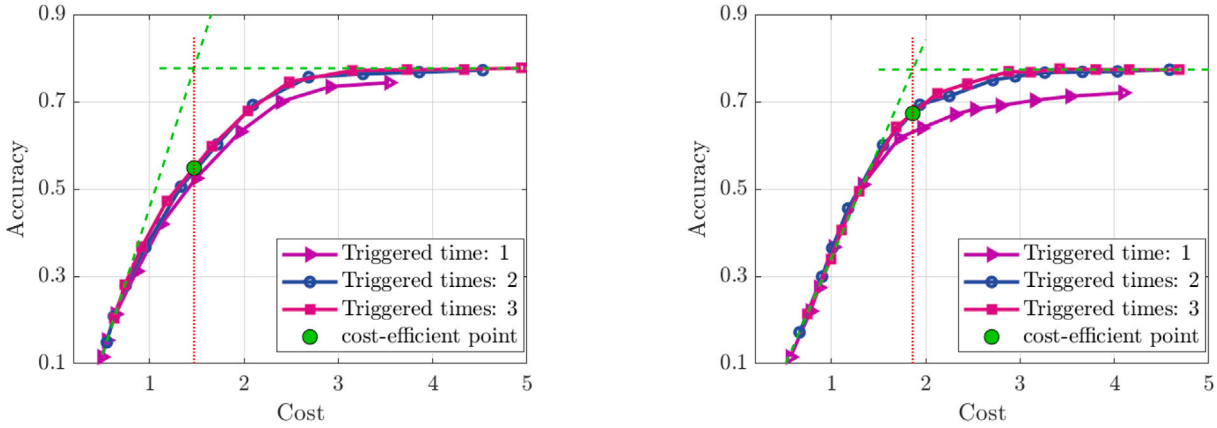


Fig. 11. The influence of variance-based stop criterion parameters on the accuracy and efficiency metric.



(a) Static-based stopping criteria ϵ_S (b) Variance-based stopping criteria ϵ_V

Fig. 12. Cost-Accuracy curve on training sequential surrogates with different settings of two types of stopping criteria.

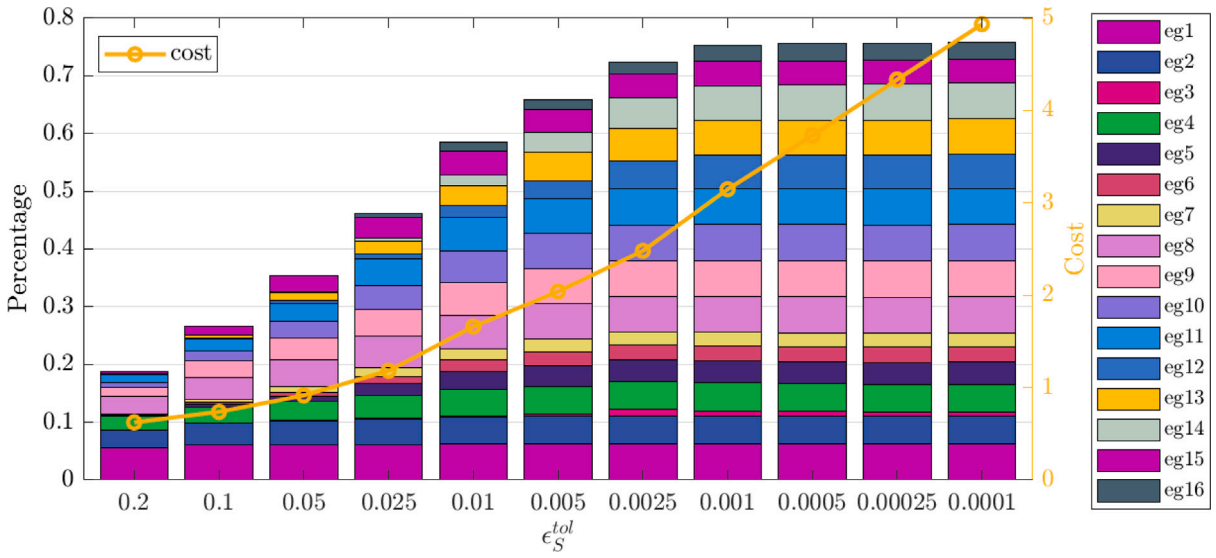


Fig. 13. Decomposition of the aggregated accuracy metric into individual contributions from the 16 examples.

Table 3
Recommended thresholds for different configurations of the stopping criteria.

ϵ_F^{tol}	Consecutive triggers: 2		Consecutive triggers: 3	
	ϵ_S^{tol}	ϵ_V^{tol}	ϵ_S^{tol}	ϵ_V^{tol}
0.05	0.002	0.071	0.004	0.085
0.1	0.008	0.225	0.016	0.274
0.15	0.014	0.320	0.023	0.390
0.2	0.017	0.441	0.028	0.511
0.25	0.020	0.460	0.034	0.533

by detailed contributions from each example. The total accuracy represented in percentage is equal to the sum of the convergence rates (i.e., the proportion of cases that $\epsilon_F \leq \epsilon_F^{tol}$) of the 16 examples. The aggregated accuracy metric together with the total cost metric is shown in Fig. 13. Here, we observe that the optimal stopping threshold varies from example to example. For instance, Example #1 can converge accurately by setting a large threshold as 0.2. Examples #2, #4, and #8 ∼ #11 can achieve high accuracy when the threshold is adjusted to 0.01. Most problems reach their maximum precision when tuning the threshold as 0.0025. After this value, the accuracy rises slowly with increasing the cost. For Examples #3, #6, #7, and #16, reaching high accuracy is hindered by the slow convergence rate. In this case, two solutions can improve the accuracy: (*i*) increase the maximum number of model evaluations, and (*ii*) change the surrogate settings (e.g., other types of surrogates, different kernel functions, higher PCE degrees, etc.). In summary, the optimal setting for the stopping criteria depends on the problem.

5. Discussion and recommendation

5.1. AL versus uniform design

In the context of structural reliability, active learning (AL) techniques are widely recognized for their efficiency. However, structural reliability estimation focuses on only a single point along the full distribution curves. As a result, for full distribution estimation, the surrogate requires more DoE to explore the entire design domain. Given this fundamental difference, the benefits of applying AL are reduced in our context. Regarding the comparison between AL and uniform design, Section 4.3 has showed three findings:

- the profitability of using AL is significantly lower than expected;
- uniform design can even outperform AL;
- the surrogate model GP benefits more from AL than PCE and PCK.

This subsection discusses each finding and offers detailed explanations.

For the first finding, we attribute it to the fact that the maximum profitability of using AL is bounded by the discrepancy of local nonlinearity (DoLN) in the simulators.[4] A low DoLN implies a nearly uniform distribution of local nonlinearity across the design space, regardless of the magnitude of the average nonlinearity; while a high DoLN implies significant variations of local nonlinearity. Therefore, AL techniques are effective in cases of high DoLN, where they allocate the samples over the design space proportional to the local nonlinearity. However, they become inefficient in cases of low DoLN. For example, in our benchmark study, Example #4 involves a simple simulator consisting of two symmetric 2-dimensional linear functions, where an arbitrary DoE (with more than three non-colinear points) suffices for each linear function. Conversely, Example #6, characterized by a strongly nonlinear function with widespread local maxima, requires a densely and uniformly distributed set of samples for an accurate surrogate. These two examples have different levels of nonlinearity, but their DoLNs are both low. Our observation is that all the 16 benchmark examples, despite being collected from various literature, exhibit a low level of DoLN, resulting in the lower-than-expected profitability from using AL.

The second finding can be attributed to the inaccurately estimated local errors from surrogates, which in turn influences the selection of training points and leads to reduced learning efficiency. In fact, local error estimations rely on certain assumptions. For example, the variance-based error of GP assumes that the underlying simulator follows a Gaussian process. Cross-validation error assumes that the surrogate predictions are sensitive to the removal of a proportion of samples from the DoE. As highlighted in [68], if the estimated local errors are misleading, the iterative surrogates might diverge from the real simulators. Although learning functions can, to some extent, correct this divergence, redundant points are inevitably added in comparison to using accurate local errors. Therefore, uniform design can outperform AL, especially when the DoLN is low.

The third finding is attributed to the use of different basis functions for the three surrogates: GP, PCE, and PCK. Eq. (4) indicates that the

[4] Here, the local nonlinearity can be measured by the highest order of Taylor series expansions required to accurately substitute the simulator around a specific local point. Higher orders imply greater local nonlinearity.

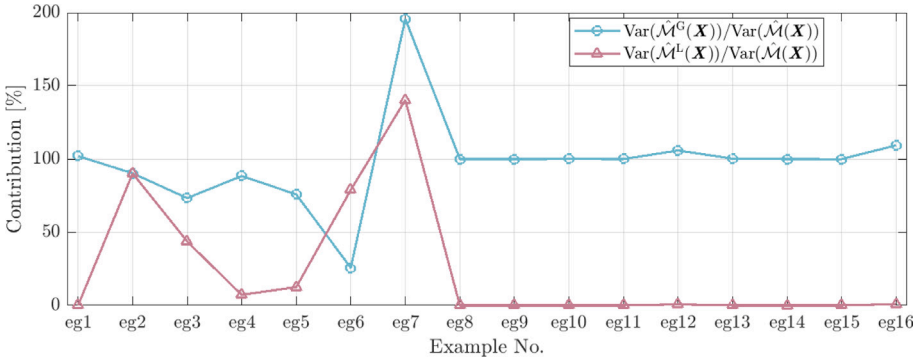


Fig. 14. Contributions of the global and local predictors on the total variance of surrogate model predictions.

three surrogates comprise either a global predictor $\hat{\mathcal{M}}^{\mathrm{G}}(\boldsymbol{x})$ or a local predictor $\hat{\mathcal{M}}^{\mathrm{L}}(\boldsymbol{x})$. Both two predictors share a similar form, i.e., the dot product of basis functions and deterministic coefficients, but their basis functions have distinct features. The basis function $R(\boldsymbol{x}, \boldsymbol{x}_j^d|\boldsymbol{\theta})$ of the local predictor relies on the position of the training points $\boldsymbol{x}_j^d$. Besides, its influence is localized at the neighbor of $\boldsymbol{x}_j^d$, with the correlations decay as $\boldsymbol{x}$ moves away from the training points. This feature makes $\hat{\mathcal{M}}^{\mathrm{L}}(\boldsymbol{x})$ adept at capturing local variations. On the other hand, the polynomial basis functions $\boldsymbol{\Psi}(\boldsymbol{x})$ of the global predictor span over the design space, making $\hat{\mathcal{M}}^{\mathrm{G}}(\boldsymbol{x})$ better suited for capturing global trends. These two distinct features determine that $\hat{\mathcal{M}}^{\mathrm{G}}(\boldsymbol{x})$ and $\hat{\mathcal{M}}^{\mathrm{L}}(\boldsymbol{x})$ require different distribution patterns of design samples to achieve the same level of approximation accuracy. Specifically, $\hat{\mathcal{M}}^{\mathrm{L}}(\boldsymbol{x})$ requires more samples in the nonlinear region but fewer in the linear region. However, $\hat{\mathcal{M}}^{\mathrm{G}}(\boldsymbol{x})$ is globally fitted to capture the underlying patterns in data. For a pure global predictor, a sufficient number of samples (comparable to the number of free regression parameters $\boldsymbol{\beta}$) is more crucial than searching for the optimal allocations of the training samples.

AL techniques specialize in searching regions with large local prediction error/uncertainty, and these regions are generally nonlinear. Therefore, the AL is more effective in accelerating the training of local predictors. As a result, the benefits of using AL are more pronounced for the pure local-prediction-based GP than for the global-prediction-based PCE.

To understand why PCK benefits less from AL than GP, we conduct variance decomposition for PCK to quantify the explained variability separately from its global and local predictors. Specifically, we apply the variance operator to both sides of Eq. (4) with respect to random input $\boldsymbol{X}$, obtaining:

$$\begin{aligned}\mathrm{Var}\left(\hat{\mathcal{M}}(\boldsymbol{X})\right) =& \mathrm{Var}\left(\hat{\mathcal{M}}^{\mathrm{G}}(\boldsymbol{X})\right) + \mathrm{Var}\left(\hat{\mathcal{M}}^{\mathrm{L}}(\boldsymbol{X})\right) \\ &+ 2\mathrm{CoV}\left(\hat{\mathcal{M}}^{\mathrm{G}}(\boldsymbol{X}), \hat{\mathcal{M}}^{\mathrm{L}}(\boldsymbol{X})\right),\end{aligned} \tag{11}$$

where the left-hand side denotes the total variance, and the right-hand side separately denotes the explained variance by $\hat{\mathcal{M}}^{\mathrm{G}}(\boldsymbol{x})$ and $\hat{\mathcal{M}}^{\mathrm{L}}(\boldsymbol{x})$ in addition to their covariance. We compute the sample variance and covariance in Eq. (11) using 10^5 random samples. We illustrate this decomposition for the 16 benchmark examples and train the PCK using the two-step learning function until reaching the maximum DoE size (as prescribed in Section 4.1.2). Fig. 14 illustrates the variance contributed from the global predictor and local predictor, depicting their percentages relative to the total variance. It is observed that only in Examples #2 ~ #7 the local predictors make noticeable contributions to the total variation. For all the other examples, the global predictor dominates. Therefore, the PCK degenerates to PCE for most of the examples studied. This finding can also explain why GP is benefited more from AL than PCK.

5.2. Recommendations to construct surrogates

In this section, we summarize the findings to provide recommendations on constructing surrogates for computing probability distributions. Our suggestions are set up based on the statistical analysis of the results for solving the 16 benchmark problems. The conclusions might become inappropriate when practitioners study problems that are significantly different from the benchmarks. Our general finding is that there is no one strategy that is consistently superior to the others. The best strategy is correlated to features of the problem (e.g., discrepancy of local nonlinearity), the chosen types of surrogates, the expected accuracy requirement, and the computational budget. The recommendations are listed in the following.

1. **Surrogate module**: In general, we recommend practitioners to use PCK, which combines the advantages of global approximation from PCE and the property of exact interpolation from GP. Additionally, we suggest setting the trend of PCK with a low-degree PCE. This recommendation is based on our finding that the simulators in our benchmarks, despite being selected from a diverse literature, exhibit a low-degree polynomial structure. However, it is also important to note that PCK cannot consistently outperform GP and PCE. The best choice between these models ultimately depends on the problem and the required level of accuracy.
2. **DoE enrichment module**: For PCE and PCK, we suggest using uniform design, as the improvement of using AL is generally limited, and AL can sometime perform worse than uniform design. For GP, AL techniques are recommended, although we cannot identify the best AL technique. Our study shows that choosing the proper type of surrogate is more crucial and can result in saving a greater number of training points. When AL techniques are preferable in practice, we recommend consulting the latest review papers about AL techniques for GP [62,63]. However, it is worth noting that the best choice of AL methods for GP often depends on the specific features of the problem [63].
3. **Stopping criteria module**: First, we recommend setting the thresholds provided in Table 3 for the two stopping criteria, i.e., static-based criteria ϵ_S and variance-based criteria ϵ_V. These thresholds are determined by balancing the accuracy and cost. Additionally, we suggest choosing the stopping criterion ϵ_V when the computational budget is more relaxed or higher level of accuracy is required. Conversely, we suggest choosing the static-based criteria ϵ_S in scenarios where computational resources are limited or a lower level of accuracy is acceptable. Finally, we recommend setting the number of consecutive triggers of stopping criteria to a number larger than 1.

6. Conclusions

The paper conducts a comprehensive comparative study on constructing surrogate models for full distribution estimations. It addresses two primary objectives: (*i*) compare uniform design to active learning design, and (*ii*) provide recommendations for practitioners on constructing surrogate models. The study proposes a modular framework that comprises three essential modules: types of surrogate, DoE enrichment methods, and stopping criteria. Various representative methods are investigated for each module to provide insights into their comparative performance.

Our first conclusion is that active learning techniques do not consistently provide a systematic improvement compared to uniform design. We provide three reasons (in Section 5.1) to explain why the benefit of using AL is lower than expected and varies for different surrogate models. Our second conclusion is that no strategy can consistently outperform the others in constructing surrogates, and the recommended configurations (in Section 5.2) depend on the chosen types of surrogate model, features of the problem (especially the discrepancy of the local nonlinearity of the simulators), and the trade-off between target accuracy and affordable cost. It is important to acknowledge that the conclusions drawn are based on the specific methods selected for each module. While these methods are representative, they do not encompass the full range of available techniques. Further investigations are needed to explore a broader variety of surrogate models, active learning strategies, and stopping criteria, as well as tests conducted on a more diverse set of benchmark examples.

Although the focus of the study is on constructing surrogate models for probability distribution estimation, the conclusions are expected to be relevant to other applications such as sensitivity analysis, parameter estimation, and reliability-based optimization.

Finally, this paper proposed a novel pool-based maximin-distance design method that exhibits promising performance in estimating distribution functions. Although this new uniform design method is used only as a baseline for the comparative study, it is expected that the method could be promising for other applications.

CRediT authorship contribution statement

Maijia Su: Writing – original draft, Methodology, Formal analysis, Conceptualization. **Ziqi Wang:** Writing – review & editing, Supervision, Methodology, Conceptualization. **Oreste Salvatore Bursi:** Writing – review & editing, Supervision. **Marco Broccardo:** Writing – review & editing, Supervision, Methodology, Conceptualization.

Declaration of competing interest

The authors declare that they do not possess any identifiable conflicting financial interests or personal relationships that might have potentially influenced the findings presented in this paper.

Acknowledgments

The first and third authors are funded by the Italian Ministry of Education, University and Research (MIUR) in the frame of the "Departments of Excellence 2023–2027" (grant L 232/2016). The last author is by funded the European Union under Next GenerationEU. PRIN 2022 Prot. n 2022MJ82MC_001.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.ress.2025.111059.

Data availability

The code and simulation results involved in this study are available upon reasonable request from the authors.

References

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

[2] Lee SH, Chen W. A comparative study of uncertainty propagation methods for black-box-type problems. Struct Multidiscip Optim 2008;37(3):239–53. http://dx.doi.org/10.1007/s00158-008-0234-7.

[3] Evans DH. An application of numerical integration techniques to statistical tolerancing, III—general distributions. Technometrics 1972;14(1):23–35. http://dx.doi.org/10.1080/00401706.1972.10488880.

[4] Robert CP, Casella G. Monte Carlo statistical methods. Springer New York; 2004, http://dx.doi.org/10.1007/978-1-4757-4145-2.

[5] Melchers R. Importance sampling in structural systems. Struct Saf 1989;6(1):3–10. http://dx.doi.org/10.1016/0167-4730(89)90003-9.

[6] Tabandeh A, Jia G, Gardoni P. A review and assessment of importance sampling methods for reliability analysis. Struct Saf 2022;97:102216. http://dx.doi.org/10.1016/j.strusafe.2022.102216, URL https://www.sciencedirect.com/science/article/pii/S0167473022000297.

[7] Au S-K, Beck JL. Estimation of small failure probabilities in high dimensions by subset simulation. Probab Eng Mech 2001;16(4):263–77. http://dx.doi.org/10.1016/s0266-8920(01)00019-4.

[8] Ditlevsen O, Madsen HO. Structural reliability methods, vol. 178, Wiley New York; 1996.

[9] Der Kiureghian A. Structural and system reliability. Cambridge University Press; 2022.

[10] Madsen HO, Krenk S, Lind NC. Methods of structural safety. Courier Corporation; 2006.

[11] Sudret B, Marelli S, Wiart J. Surrogate models for uncertainty quantification: An overview. In: 2017 11th European conference on antennas and propagation. IEEE; 2017, p. 793–7. http://dx.doi.org/10.23919/EuCAP.2017.7928679.

[12] Jones DR, Schonlau M, Welch WJ. Efficient global optimization of expensive black-box functions. J Global Optim 1998;13(4):455–92. http://dx.doi.org/10.1023/A:1008306431147.

[13] Echard B, Gayton N, Lemaire M. AK-MCS: An active learning reliability method combining kriging and Monte Carlo simulation. Struct Saf 2011;33(2):145–54. http://dx.doi.org/10.1016/j.strusafe.2011.01.002.

[14] Crombecq K, Gorissen D, Deschrijver D, Dhaene T. A novel hybrid sequential design strategy for global surrogate modeling of computer experiments. SIAM J Sci Comput 2011;33(4):1948–74. http://dx.doi.org/10.1137/090761811.

[15] Faravelli L. Response-surface approach for reliability analysis. J Eng Mech 1989;115(12):2763–81. http://dx.doi.org/10.1061/(ASCE)0733-9399(1989)115:12(2763).

[16] Papadrakakis M, Lagaros ND. Reliability-based structural optimization using neural networks and Monte Carlo simulation. Comput Methods Appl Mech Engrg 2002;191(32):3491–507. http://dx.doi.org/10.1016/s0045-7825(02)00287-6.

[17] Kaymaz I. Application of kriging method to structural reliability problems. Struct Saf 2005;27(2):133–51. http://dx.doi.org/10.1016/j.strusafe.2004.09.001.

[18] Guo H, Dong Y, Gardoni P. Adaptive subset simulation for time-dependent small failure probability incorporating first failure time and single-loop surrogate model. Struct Saf 2023;102:102327. http://dx.doi.org/10.1016/j.strusafe.2023.102327, URL https://www.sciencedirect.com/science/article/pii/S0167473023000140.

[19] Basudhar A, Missoum S, Sanchez AH. Limit state function identification using Support Vector Machines for discontinuous responses and disjoint failure domains. Probab Eng Mech 2008;23(1):1–11. http://dx.doi.org/10.1016/j.probengmech.2007.08.004.

[20] Pepper N, Crespo L, Montomoli F. Adaptive learning for reliability analysis using support vector machines. Reliab Eng Syst Saf 2022;226:108635. http://dx.doi.org/10.1016/j.ress.2022.108635.

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

[22] Schobi R, Sudret B, Wiart J. Polynomial-chaos-based Kriging. Int J Uncertain Quantif 2015;5(2). http://dx.doi.org/10.1615/Int.J.UncertaintyQuantification.2015012467.

[23] Guo H, Dong Y, Gardoni P. Efficient subset simulation for rare-event integrating point-evolution kernel density and adaptive polynomial chaos kriging. Mech Syst Signal Process 2022;169:108762. http://dx.doi.org/10.1016/j.ymssp.2021.108762, URL https://www.sciencedirect.com/science/article/pii/S0888327021010785.

[24] Kenett RS, Zacks S. Modern industrial statistics: With applications in R, MINITAB, and JMP. John Wiley & Sons; 2021, http://dx.doi.org/10.1002/9781118763667.ch11.

[25] Plackett RL, Burman JP. The design of optimum multifactorial experiments. Biometrika 1946;33(4):305–25. http://dx.doi.org/10.2307/2332195.

[26] Garud SS, Karimi IA, Kraft M. Design of computer experiments: A review. Comput Chem Eng 2017;106:71–95. http://dx.doi.org/10.1016/j.compchemeng.2017.05.010.

[27] Pronzato L, Müller WG. Design of computer experiments: space filling and beyond. Stat Comput 2012;22:681–701. http://dx.doi.org/10.1007/s11222-011-9242-3.

[28] Hickernell F. A generalized discrepancy and quadrature error bound. Math Comp 1998;67(221):299–322. http://dx.doi.org/10.1090/S0025-5718-98-00894-1.

[29] Caflisch RE. Monte carlo and quasi-monte carlo methods. Acta Numer 1998;7:1–49. http://dx.doi.org/10.1007/978-0-387-78165-5.

[30] Johnson ME, Moore LM, Ylvisaker D. Minimax and maximin distance designs. J Statist Plann Inference 1990;26(2):131–48. http://dx.doi.org/10.1016/0378-3758(90)90122-B.

[31] Morris MD, Mitchell TJ. Exploratory designs for computational experiments. J Statist Plann Inference 1995;43(3):381–402. http://dx.doi.org/10.1016/0378-3758(94)00035-T.

[32] Sobol' IM. On the distribution of points in a cube and the approximate evaluation of integrals. Zh Vychisl'noi Mat i Mat Fiz 1967;7(4):784–802. http://dx.doi.org/10.1016/0167-4730(89)90003-9.

[33] Halton JH. Algorithm 247: Radical-inverse quasi-random point sequence. Commun ACM 1964;7(12):701–2. http://dx.doi.org/10.1145/355588.365104.

[34] Villagran A, Huerta G, Vannucci M, Jackson CS, Nosedal A. Non-parametric sampling approximation via Voronoi tessellations. Comm Statist Simulation Comput 2016;45(2):717–36. http://dx.doi.org/10.1080/03610918.2013.870798.

[35] Eglajs V, Audze P. New approach to the design of multifactor experiments. Probl Dyn Strengths 1977;35(1):104–7.

[36] McKay MD, Beckman RJ, Conover WJ. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. Technometrics 2000;42(1):55–61. http://dx.doi.org/10.2307/1268522.

[37] Grosso A, Jamali A, Locatelli M. Finding maximin latin hypercube designs by iterated local search heuristics. European J Oper Res 2009;197(2):541–7. http://dx.doi.org/10.1016/j.ejor.2008.07.028.

[38] Stinstra E, den Hertog D, Stehouwer P, Vestjens A. Constrained maximin designs for computer experiments. Technometrics 2003;45(4):340–6. http://dx.doi.org/10.1198/004017003000000168.

[39] Wang GG. Adaptive response surface method using inherited latin hypercube design points. J Mech Des 2003;125(2):210–20. http://dx.doi.org/10.1115/1.1561044.

[40] Settles B. Active Learning Literature Survey. Computer sciences technical report 1648, University of Wisconsin–Madison; 2009.

[41] Bichon BJ, Eldred MS, Swiler LP, Mahadevan S, McFarland JM. Efficient global reliability analysis for nonlinear implicit performance functions. AIAA J 2008;46(10):2459–68. http://dx.doi.org/10.2514/1.34321.

[42] Lv Z, Lu Z, Wang P. A new learning function for Kriging and its applications to solve reliability problems in engineering. Comput Math Appl 2015;70(5):1182–97. http://dx.doi.org/10.1016/j.camwa.2015.07.004, URL https://www.sciencedirect.com/science/article/pii/S0898122115003399.

[43] Zhan D, Qian J, Cheng Y. Pseudo expected improvement criterion for parallel EGO algorithm. J Global Optim 2017;68(3):641–62. http://dx.doi.org/10.1007/s10898-016-0484-7.

[44] Sun Z, Wang J, Li R, Tong C. LIF: A new Kriging based learning function and its application to structural reliability analysis. Reliab Eng Syst Saf 2017;157:152–65. http://dx.doi.org/10.1016/j.ress.2016.09.003, URL https://www.sciencedirect.com/science/article/pii/S0951832016304847.

[45] Ma Y-Z, Zhu Y-C, Li H-S, Nan H, Zhao Z-Z, Jin X-X. Adaptive Kriging-based failure probability estimation for multiple responses. Reliab Eng Syst Saf 2022;228:108771. http://dx.doi.org/10.1016/j.ress.2022.108771, URL https://www.sciencedirect.com/science/article/pii/S0951832022003945.

[46] Chauhan MS, Ojeda-Tuz M, Catarelli RA, Gurley KR, Tsapetis D, Shields MD. On active learning for Gaussian process-based global sensitivity analysis. Reliab Eng Syst Saf 2024;245:109945. http://dx.doi.org/10.1016/j.ress.2024.109945, URL https://www.sciencedirect.com/science/article/pii/S0951832024000206.

[47] Zhao Z, Lu Z-H, Zhao Y-G. P-AK-MCS: Parallel AK-MCS method for structural reliability analysis. Probab Eng Mech 2024;75:103573. http://dx.doi.org/10.1016/j.probengmech.2023.103573, URL https://www.sciencedirect.com/science/article/pii/S0266892023001625.

[48] Zhang C, Song C, Shafieezadeh A. GELF: A global error-based learning function for globally optimal adaptive reliability analysis. Struct Saf 2024;109:102464. http://dx.doi.org/10.1016/j.strusafe.2024.102464, URL https://www.sciencedirect.com/science/article/pii/S0167473024000353.

[49] Xian J, Wang Z. A physics and data co-driven surrogate modeling method for high-dimensional rare event simulation. J Comput Phys 2024;510:113069. http://dx.doi.org/10.1016/j.jcp.2024.113069, URL https://www.sciencedirect.com/science/article/pii/S0021999124003188.

[50] Razaaly N, Congedo PM. Extension of AK-MCS for the efficient computation of very small failure probabilities. Reliab Eng Syst Saf 2020;203:107084. http://dx.doi.org/10.1016/j.ress.2020.107084, URL https://www.sciencedirect.com/science/article/pii/S0951832020305858.

[51] Xiong Y, Sampath S. A fast-convergence algorithm for reliability analysis based on the AK-MCS. Reliab Eng Syst Saf 2021;213:107693. http://dx.doi.org/10.1016/j.ress.2021.107693, URL https://www.sciencedirect.com/science/article/pii/S0951832021002301.

[52] Ma Y-Z, Liu M, Nan H, Li H-S, Zhao Z-Z. A novel hybrid adaptive scheme for Kriging-based reliability estimation–A comparative study. Appl Math Model 2022;108:1–26. http://dx.doi.org/10.1016/j.apm.2022.03.015, Publisher: Elsevier.

[53] Zhang Y, Dong Y, Frangopol DM. An error-based stopping criterion for spherical decomposition-based adaptive Kriging model and rare event estimation. Reliab Eng Syst Saf 2024;241:109610. http://dx.doi.org/10.1016/j.ress.2023.109610, URL https://www.sciencedirect.com/science/article/pii/S0951832023005240.

[54] Su M, Xue G, Wang D, Zhang Y, Zhu Y. A novel active learning reliability method combining adaptive Kriging and spherical decomposition-MCS (AK-SDMCS) for small failure probabilities. Struct Multidiscip Optim 2020;62(6):3165–87. http://dx.doi.org/10.1007/s00158-020-02661-w.

[55] Echard B, Gayton N, Lemaire M, Relun N. A combined importance sampling and kriging reliability method for small failure probabilities with time-demanding numerical models. Reliab Eng Syst Saf 2013;111:232–40. http://dx.doi.org/10.1016/j.ress.2012.10.008, Publisher: Elsevier.

[56] Huang X, Chen J, Zhu H. Assessing small failure probabilities by AK–SS: An active learning method combining Kriging and Subset Simulation. Struct Saf 2016;59:86–95. http://dx.doi.org/10.1016/j.strusafe.2015.12.003.

[57] Kim J, Song J. Probability-Adaptive Kriging in n-Ball (PAK-Bn) for reliability analysis. Struct Saf 2020;85:101924. http://dx.doi.org/10.1016/j.strusafe.2020.101924.

[58] Hong F, Song J, Wei P, Huang Z, Beer M. A stratified beta-sphere sampling method combined with important sampling and active learning for rare event analysis. Struct Saf 2024;102546. http://dx.doi.org/10.1016/j.strusafe.2024.102546.

[59] Wang D, Zhang D, Meng Y, Yang M, Meng C, Han X, et al. AK-HRn: An efficient adaptive Kriging-based n-hypersphere rings method for structural reliability analysis. Comput Methods Appl Mech Engrg 2023;414:116146. http://dx.doi.org/10.1016/j.cma.2023.116146.

[60] Jones AC, Wilcox RK. Finite element analysis of the spine: towards a framework of verification, validation and sensitivity analysis. Med Eng Phys 2008;30(10):1287–304. http://dx.doi.org/10.1016/j.medengphy.2008.09.006.

[61] Mosbach S, Braumann A, Man PL, Kastner CA, Brownbridge GP, Kraft M. Iterative improvement of Bayesian parameter estimates for an engine model by means of experimental design. Combust Flame 2012;159(3):1303–13. http://dx.doi.org/10.1016/j.combustflame.2011.10.019.

[62] Liu H, Ong Y-S, Cai J. A survey of adaptive sampling for global metamodeling in support of simulation-based complex engineering design. Struct Multidiscip Optim 2018;57(1):393–416. http://dx.doi.org/10.1007/s00158-017-1739-8.

[63] Fuhg JN, Fau A, Nackenhorst U. State-of-the-art and comparative review of adaptive sampling methods for kriging. Arch Comput Methods Eng 2021;28(4):2689–747. http://dx.doi.org/10.1007/s11831-020-09474-6.

[64] Farhang-Mehr A, Azarm S. Bayesian meta-modelling of engineering design simulations: a sequential approach with adaptation to irregularities in the response behaviour. Internat J Numer Methods Engrg 2005;62(15):2104–26. http://dx.doi.org/10.1002/nme.1261.

[65] Busby D, Farmer CL, Iske A. Hierarchical nonlinear approximation for experimental design and statistical data fitting. SIAM J Sci Comput 2007;29(1):49–69. http://dx.doi.org/10.1137/050639983.

[66] Li G, Aute V, Azarm S. An accumulative error based adaptive design of experiments for offline metamodeling. Struct Multidiscip Optim 2010;40(1):137–55. http://dx.doi.org/10.1007/s00158-009-0395-z.

[67] Aute V, Saleh K, Abdelaziz O, Azarm S, Radermacher R. Cross-validation based single response adaptive design of experiments for Kriging metamodeling of deterministic computer simulations. Struct Multidiscip Optim 2013;48(3):581–605. http://dx.doi.org/10.1007/s00158-013-0918-5.

[68] Jin R, Chen W, Sudjianto A. On sequential sampling for global metamodeling in engineering design. In: International design engineering technical conferences and computers and information in engineering conference, vol. 36223, 2002, p. 539–48. http://dx.doi.org/10.1115/DETC2002/DAC-34092.

[69] Moustapha M, Marelli S, Sudret B. Active learning for structural reliability: Survey, general framework and benchmark. Struct Saf 2022;96:102174. http://dx.doi.org/10.1016/j.strusafe.2021.102174.

[70] Sacks J, Welch WJ, Mitchell TJ, Wynn HP. Design and analysis of computer experiments. Statist Sci 1989;4(4):409–23. http://dx.doi.org/10.1214/ss/1177012413.

[71] Sudret B. Global sensitivity analysis using polynomial chaos expansions. Reliab Eng Syst Saf 2008;93(7):964–79. http://dx.doi.org/10.1016/j.ress.2007.04.002.

[72] Mo S, Lu D, Shi X, Zhang G, Ye M, Wu J, et al. A Taylor expansion-based adaptive design strategy for global surrogate modeling with applications in groundwater modeling. Water Resour Res 2017;53(12):10802–23. http://dx.doi.org/10.1002/2017WR021622.

[73] Schöbi R, Marelli S, Sudret B. UQLab user manual – Polynomial chaos Kriging. In: Chair of risk, safety and uncertainty quantification. Tech. rep., Switzerland: ETH Zurich; 2021, report n UQLab-V1.4-109.
[74] Lataniotis C, Wicaksono D, Marelli S, Sudret B. UQLab user manual – Kriging (Gaussian process modeling). In: Chair of risk, safety and uncertainty quantification. Tech. rep., Switzerland: ETH Zurich; 2022, report UQLab-V2.0-105.
[75] Marelli S, Lüthen N, Sudret B. UQLab user manual – Polynomial chaos expansions. In: Chair of risk, safety and uncertainty quantification. Tech. rep., Switzerland: ETH Zurich; 2022, report UQLab-V2.0-104.
[76] Lophaven S, Nielsen H, Sondergaard J. DACE-a matlab Kriging toolbox, version 2.0. In: Informatics and mathematical modelling. Technical report imm-tr-2002-12, Denmark: Technical University of Denmark DK-2800 Kgs. Lyngby; 2002.
[77] Santner TJ, Williams BJ, Notz WI. The design and analysis of computer experiments. Springer; 2003.
[78] Marelli S, Sudret B. An active-learning algorithm that combines sparse polynomial chaos expansions and bootstrap for structural reliability analysis. Struct Saf 2018;75:67–74. http://dx.doi.org/10.1016/j.strusafe.2018.06.003.
[79] Xiao N-C, Zuo MJ, Zhou C. A new adaptive sequential sampling method to construct surrogate models for efficient reliability analysis. Reliab Eng Syst Saf 2018;169:330–8. http://dx.doi.org/10.1016/j.ress.2017.09.008.
[80] Seung HS, Opper M, Sompolinsky H. Query by committee. In: Proceedings of the fifth annual workshop on computational learning theory. 1992, p. 287–94. http://dx.doi.org/10.1145/130385.130417.
[81] Joseph VR. Space-filling designs for computer experiments: A review. Qual Eng 2016;28(1):28–35. http://dx.doi.org/10.1080/08982112.2015.1100447.
[82] Blatman G, Sudret B. Efficient computation of global sensitivity indices using sparse polynomial chaos expansions. Reliab Eng Syst Saf 2010;95(11):1216–29. http://dx.doi.org/10.1016/j.strusafe.2018.06.003.
[83] Liu P. Der Kiureghian: A. Optimization algorithms for structural reliability. Struct Saf 1991;9(3):161–77. http://dx.doi.org/10.1016/0167-4730(91)90041-7.
[84] Ohtori Y, Christenson R, Spencer Jr B, Dyke S. Benchmark control problems for seismically excited nonlinear buildings. J Eng Mech 2004;130(4):366–85. http://dx.doi.org/10.1061/(ASCE)0733-9399(2004)130:4(366).
[85] Ma Y-Z, Liu M, Nan H, Li H-S, Zhao Z-Z. A novel hybrid adaptive scheme for Kriging-based reliability estimation–A comparative study. Appl Math Model 2022;108:1–26. http://dx.doi.org/10.1016/j.apm.2022.03.015.