



# Accelerating Gaussian Process surrogate modeling using Compositional Kernel Learning and multi-stage sampling framework

Seung-Seop Jin

Sustainable Infrastructure Research Center, KICT, Goyang-Si 10223, Republic of Korea



## ARTICLE INFO

### Article history:

Received 20 March 2020

Received in revised form 6 November 2020

Accepted 11 November 2020

Available online 17 November 2020

### Keywords:

Global surrogate modeling

Gaussian Process

Compositional Kernel Learning

Progressive Latin Hypercube Sampling

Diagnostics

## ABSTRACT

Surrogate modeling is becoming a popular tool to approximate computationally-expensive simulations for complex engineering problems. In practice, there are still difficulties in surrogate modeling as follows: (1) efficient learning for functional relationship of simulation models and (2) diagnostics for the surrogate model. In order to address these difficulties simultaneously, this paper proposes a new sequential surrogate modeling by integrating a Compositional Kernel Learning (CKL) method for Gaussian process into a sequential sampling strategy termed the Progressive Latin Hypercube Sampling (PLHS). The CKL enables efficient learning capability for complex response surfaces based on richly structured kernels, while the PLHS sequentially generates nested samples by maintaining desired properties for distribution. Furthermore, this sequential sampling framework allows users to monitor the diagnostics of the surrogate model and assess the stopping criteria for further sampling. In order to demonstrate useful features of the proposed method, nine test functions were assembled for numerical experiments to cover different types of problems (i.e., scale and complexity). The proposed method was evaluated with a set of surrogate modeling techniques and sampling methods in terms of performance, diagnostics and computational cost. The results show that (1) the proposed method can learn various response surfaces with fewer training samples than other methods; and (2) the proposed method only provides a reliable diagnostic measure for global accuracy over different types of problems.

© 2020 Elsevier B.V. All rights reserved.

## 1. Introduction

Simulation model such as Finite Element Analysis (FEA) and Computational Fluid Dynamics (CFD) is a mathematical representation of a real-world physical problem implemented in a computer code. Nowadays, the simulation models have been extensively used in various types of engineering problems (e.g., domain exploration, design optimization, sensitivity/uncertainty analysis and inverse analysis) [1–4], because physical experiments are either highly expensive or technically impossible.

As physical knowledge and computing power become more advanced, more sophisticated simulation models are gaining widespread use to tackle various complex engineering problems. These simulation models typically have non-linear and complex response surfaces with large input spaces [5,6]. In addition, these simulation models often require huge computational resources (i.e., long run-time with huge computing power). If the analysis of the simulation model is iterated many times, the computational process would be highly challenging under limited resources.

To mitigate the computational burden, surrogate models have been gaining a considerable attention as a cost-effective substitute for the simulation model [7–10]. Since the deterministic simulation model produces identical outputs with identical inputs [3], the response surface of the simulation model can be represented by a mathematical/statistical representation [11]. This representation is referred to as a surrogate model, also known as response surface model, emulator and meta-model. Once the surrogate model is constructed, the surrogate model is implemented without running additional simulations for design optimization, design space exploration and sensitivity/uncertainty analysis.

Based on the purpose of the engineering problems, surrogate modeling can be categorized into (1) global surrogate modeling and (2) black-box optimization. The global surrogate modeling aims at mimicking the response surface over the input space (for sensitivity or uncertainty analysis) [12–14], while the black-box optimization utilizes a sequential design strategy for global optimization of the black-box functions [15–17]. The scope of this study is confined to the global surrogate modeling. The global surrogate modeling consists of two stages: (1) sampling stage, wherein a set of simulation runs (known as training samples) is performed over the input space based on sampling strategies; and (2) model-fitting stage, wherein the surrogate model is fitted

E-mail address: [seungsab@kict.re.kr](mailto:seungsab@kict.re.kr).

using the training samples. Among various types of the surrogate modeling techniques and sampling methods, selecting robust methods is still a challenging for practical problems [14].

For the successful global surrogate modeling, the learning capability of the surrogate model is important. There are various types of the surrogate modeling such as polynomial model (POLY) [18–20], radial basis function (RBF) [19–25] and Gaussian Process (GP) [19,20,22,23,25–31]. Chronologically, Table 1 shows the recent works on comparative study of surrogate modeling. The improvements in computational power significantly increase the research interest for the development of more advanced surrogate modeling to improve learning capability. As a result, the non-parametric models such as GP (also known as Kriging) and RBF are prevalent for the global surrogate modeling, since they can approximate the response surface more flexibly than parametric models (e.g., polynomial model) [9]. Recently, the enhanced-GPs (such as Blind Kriging and Gradient-enhanced Kriging) have been gaining a considerable attention for the engineering problems [13,24,28–33]. Comprehensive works for the surrogate modeling [14,34] show that the optimal model is case-dependent to the problem types and modeling setting (e.g., kernel function in GP).

Sampling method (also known as design of experiment, DOE) generates the training samples to gather informative experiments (simulations) for the surrogate modeling. The accuracy of the surrogate model heavily depends on training samples, so that the sampling method is crucial to the predictive quality of the surrogate model. Classical sampling methods in the DOE (e.g., central composite design) can be utilized to generate training samples. However, they usually generate more samples around the boundary regions. For a computational DOE, it is preferred to fill the entire input space (i.e., space-filling) rather than filling the boundary regions [35]. In this context, space-filling sampling (SFS) method has gained much popularity for the surrogate modeling. Latin hypercube sampling (LHS) [36] and Low-discrepancy sequence (e.g., Sobol's sequence) [37] are the most popular SFS method in various fields.

Conventional SFS method is a single-stage sampling strategy of generating the entire samples at once. The optimal LHS has been developed to improve space-filling properties. The optimal LHS optimizes some space-filling criteria (such as *maximin distance* criterion [38,39], *orthogonal arrays* criterion [40,41] and so on [42–44]) to generate the training samples. In the conventional SFS method, the size of the training samples should be pre-determined. However, it is difficult even for experts to determine an appropriate size of the training samples in advance. Hence, this difficulty actuates the development of the sequential SFS method [5,45–49]. Table 2 shows the development of the SFS method in chronological order. The sequential SFS method has developed based on the conventional SFS to augment training samples. To ensure the desired space-filling property, the sequential SFS method treats the sampling process as a set of optimization problems by optimizing some space-filling criteria. It is worth noting that the sequential SFS method with a nested design has recently gained a considerable attention. The nested design sequentially generates successive sets of samples by making former samples as a subset of the latter samples [5,45,46,48,49].

The accuracy of the surrogate model is strongly dependent to (1) learning capability of the surrogate model and (2) training samples. The accuracy of the surrogate model is often unsatisfactory to represent the response surface of the simulation model. The training samples should be sufficient to capture the response surface, while the learning capability of the surrogate model should be maximized to learn the response surface effectively. In general, they interact with each other and have the influence

on the accuracy of the surrogate model. For example, a large size of training samples is required to get reasonable accuracy of the surrogate model under inefficient learning capability. In this context, it is important to validate the accuracy of the surrogate model before its implementation. However, there is little research for diagnostics of the surrogate model [3,13,51].

To address the issues simultaneously, this paper proposes a new surrogate modeling based on the GP by incorporating a Compositional Kernel Learning (CKL) method [52–54] into a sequential SFS strategy termed the Progressive Latin Hypercube Sampling (PLHS) [5]. The CKL is developed by Duvenaud, et al. [52] in the machine learning community. The covariance kernels of the GP are known to be closed under compositional rules (i.e., sum and product) [52]. Thus, the CKL automatically discovers a compositional kernel for a richly structured kernel to represent complex properties of the function. Although the CKL is outstanding to learn both simple and complex functions, the CKL is somewhat new for surrogate modeling. For the diagnostics of the GP with appropriate size of the training samples, the proposed method introduces the PLHS. The PLHS successively generates a series of the sub-samples (i.e., smaller slices) by maintaining desired properties for the distribution (space-filling and projective properties). Sheikholeslami, et al. [5] demonstrated that the PLHS shows the outstanding performance that scales effectively with the dimensionality of the problem. A series of the sub-samples in the PLHS are Latin hypercube as shown in Sheikholeslami, et al. [5], so that the sub-samples preserves projective properties (i.e., Latin hypercube) along with space-filling properties (i.e., *maximin* distance criterion). For the diagnostics of the surrogate model, the proposed method utilizes two consecutive sub-samples in the PLHS as training and validation samples, respectively. By virtue of using the nested samples in the PLHS, the proposed method allows users to monitor the diagnostics of the GP and assess the stopping criteria for further sampling. Numerical experiments reveal that (1) the proposed method generally outperforms or performs similarly to the best one among a set of surrogate models, so that the proposed method can learn the various types of response surfaces (i.e., scale and complexity) flexibly and efficiently; and (2) the proposed method only provides robust correlations between accuracies from validation samples (generated by the PLHS) and test samples (not available in real applications). These results indicate that the only proposed method can ensure a diagnostic measure for the global surrogate modeling via the proposed framework.

The remaining of this study is organized as follows. Section 2 firstly introduces a Gaussian process model with the CKL. Then, the PLHS is presented for diagnostics of the GP to find an appropriate size of the training samples. In Section 3, the proposed method for surrogate modeling is introduced. Section 4 shows descriptions of numerical experiments. Then, the results of numerical experiments are provided in Section 5. The discussion of the proposed method is given in Section 6. Lastly, Section 7 summarizes the conclusion. Hereafter, the boldface letters indicate vectors or matrices.

## 2. Gaussian Process and Compositional Kernel Learning

### 2.1. Gaussian Processes

A Gaussian Process (GP) is a Bayesian non-parametric model to provide an analytically tractable way of learning a complex function from input to output [26]. GP is a distribution over functions such that any finite set of function values has a joint multivariate Gaussian distribution. In this context, GP is completely defined by a mean function ( $m(\mathbf{X})$ ) and covariance kernel ( $\mathbf{K}(\mathbf{X}, \mathbf{X}'|\psi)$ ). The response surface ( $\eta(\mathbf{X})$ ) is assumed to be a finite set of function values ( $\mathbf{Y}(\mathbf{X})$ ) with input represented by  $\mathbf{X} \subset \mathbb{R}^p$ ; where  $p$  is

**Table 1**  
Summary of literatures on surrogate modeling technique.

Year	Literature	Surrogate modeling technique					
		POLY	RBF	GP	Blind-KRG	GEK	Others
2007	Ben-Ari, et al. [27]			✓			✓
2009	Kim, et al. [22]		✓	✓			✓
2009	Zhu, et al. [23]			✓	✓		✓
2011	Zhao, et al. [19]	✓		✓			
2012	Couckuyt, et al. [28]				✓	✓	
2014	Van Gelder, et al. [20]	✓		✓			
2016	Ulaganathan, et al. [24]		✓			✓	✓
2018	Østergård, et al. [34]		✓				✓
2018	Bhatrai, et al. [25]			✓		✓	
2019	Zhang, et al. [29]			✓	✓		
2020	Zhou, et al. [31]			✓			✓
2020	Zhang, et al. [30]			✓	✓		

**Table 2**  
Summary of literatures on space-filling sampling.

Year	Literature	Method	Sampling-stage	Space-filling criteria	
1979	McKay, et al. [36]	LHS	Single	-	
1979	Sobol [37]	Low-discrepancy sequence	Sequential	-	
1990	Johnson, et al. [38]			<i>Distance</i>	
1993	Tang [40]			<i>Orthogonality</i>	
2000	Ye, et al. [43]			<i>Symmetry</i>	
2007	Dam, et al. [39]	Latin Hypercube Sampling		<i>Distance</i>	
2007	Cioppa, et al. [42]			<i>Distance &amp; Orthogonality</i>	
2009	Xiong, et al. [45]		Sequential	<i>Distance</i>	
2010	van Dam, et al. [46]				
2012	Loeppky, et al. [41]		Single	<i>Orthogonality</i>	
2012	Schretter, et al. [50]	Low-discrepancy sequence		<i>By leaping and scrambling</i>	
2016	Yang, et al. [47]		Sequential	<i>Orthogonality</i>	
2017	Wu, et al. [48]	Latin Hypercube Sampling		<i>Distance &amp; Orthogonality</i>	
2017	Sheikholeslami, et al. [5]			<i>Distance &amp; Projective property</i>	
2019	Zhou, et al. [49]				

the input dimensionality. To build the response surface using GP, the prior distribution about  $\eta(\mathbf{X})$  is defined as a joint Gaussian distribution ( $\mathbf{GP}(\cdot, \cdot)$ ) with  $\mathbf{X}$  and  $\mathbf{Y}(\mathbf{X})$ , as given in Eq. (1).

$$\eta(\mathbf{X}) | \mathbf{X}, \mathbf{Y}(\mathbf{X}), \psi \sim \mathbf{GP}(m(\mathbf{X}), \mathbf{K}(\mathbf{X}, \mathbf{X}'|\psi)), \quad (1)$$

where  $\mathbf{K}(\mathbf{X}, \mathbf{X}'|\psi)$  must be a semi-positive definite matrix. It is common practice to assume the mean function to be zero for numerical efficiency and simplicity ( $m(\mathbf{X}) = 0$ ), since marginalizing over an unknown mean function can be equivalently represented as a zero-mean GP with a new covariance kernel (constant or trend kernels) [53]. As a result, the covariance kernel is chosen to incorporate any expert belief about the response surface. Stated differently, the covariance kernel encodes our inductive bias about the response surface.

Since the simulation model is deterministic, there is no uncertainty in the output. Thus, suppose that  $\mathbf{Y}(\mathbf{X}_{tr}) = [y(\mathbf{x}_{tr,1}) = \eta(\mathbf{x}_{tr,1}), \dots, y(\mathbf{x}_{tr,n}) = \eta(\mathbf{x}_{tr,n})]$  contains  $n$  outputs of the simulation model at the corresponding inputs ( $\mathbf{X}_{tr}$ ) in the training samples ( $\mathbf{D}_{tr} = [\mathbf{X}_{tr}, \mathbf{Y}(\mathbf{X}_{tr})]$ ). According to Eq. (1) with zero mean, the prior distribution of GP about  $\eta(\mathbf{X})$  is reformulated to Eq. (2).

$$\begin{bmatrix} \mathbf{Y}(\mathbf{X}_{tr}) \\ \mathbf{Y}(\mathbf{X}_*) \end{bmatrix} \sim \mathbf{GP} \left( 0, \begin{bmatrix} \mathbf{K}(\mathbf{X}_{tr}, \mathbf{X}_{tr}|\psi) & \dots & \mathbf{K}(\mathbf{X}_{tr}, \mathbf{X}_*|\psi) \\ \vdots & \ddots & \vdots \\ \mathbf{K}(\mathbf{X}_*, \mathbf{X}_{tr}|\psi) & \dots & \mathbf{K}(\mathbf{X}_*, \mathbf{X}_*|\psi) \end{bmatrix} \right), \quad (2)$$

where  $\mathbf{X}_{tr}$  and  $\mathbf{Y}(\mathbf{X}_{tr})$  denote observed  $n \times p$  input matrix and corresponding  $n \times 1$  output vector in the training samples, respectively;  $n$  is the size of the training samples;  $\mathbf{X}_*$  and  $\mathbf{Y}(\mathbf{X}_*)$  denote

non-observed input and corresponding output, respectively;  $0$  denotes the  $n \times 1$  vector having zero for mean function; and  $\mathbf{K}$  is the  $n \times n$  symmetric and positive semi-definite covariance kernel matrix at each input ( $\mathbf{X}_{tr}$  and  $\mathbf{X}_*$ ). The covariance kernel has its own hyper-parameter ( $\psi$ ) to represent characteristics of the response surface.

When the GP is specified (Eq. (2)), Negative Log-Marginal Likelihood (NLML) of given training samples ( $\mathbf{D}_{tr}$ ) in the right side of Eq. (3) can be calculated using standard techniques for conditioning in multivariate normal distributions. The optimal hyper-parameter ( $\hat{\psi}$ ) can be estimated by minimizing the NLML in Eq. (3).

$$\hat{\psi} = \arg \min_{\psi} \left( -\frac{1}{2} \left[ n \log(2\pi) - |\mathbf{K}(\mathbf{X}_{tr}, \mathbf{X}_{tr}|\psi)| - \mathbf{Y}(\mathbf{X}_{tr})^T \mathbf{K}(\mathbf{X}_{tr}, \mathbf{X}_{tr}|\psi)^{-1} \mathbf{Y}(\mathbf{X}_{tr}) \right] \right). \quad (3)$$

Considering that the NLML is typically multi-modal, the optimizer can converge to a local minimum. As a result, the optimal hyper-parameter is estimated using a multi-start gradient based optimization [55]. Using the formula for Gaussian conditionals with optimal hyper-parameters ( $\hat{\psi}$ ), the posterior predictive distribution of a function value ( $\mathbf{Y}_*(\mathbf{X}_*)$ ) at new input ( $\mathbf{X}_*$ ) is analytically defined as the conditional distribution in Eq. (4).

$$\begin{aligned} p(\mathbf{Y}_*(\mathbf{X}_*) | \mathbf{X}_{tr}, \mathbf{Y}(\mathbf{X}_{tr}), \mathbf{X}_*, \hat{\psi}) \\ = \mathbf{GP} \left( \hat{\mathbf{Y}}(\mathbf{X}_* | \mathbf{X}_{tr}, \mathbf{Y}(\mathbf{X}_{tr}), \hat{\psi}), s^2(\mathbf{X}_* | \mathbf{X}_{tr}, \mathbf{Y}(\mathbf{X}_{tr}), \hat{\psi}) \right). \end{aligned} \quad (4)$$

The posterior mean ( $\hat{Y}(\mathbf{X}_*|\mathbf{X}_{tr}, \mathbf{Y}(\mathbf{X}_{tr}), \hat{\psi})$ ) and variance ( $s^2(\mathbf{X}_*|\mathbf{X}_{tr}, \mathbf{Y}(\mathbf{X}_{tr}), \hat{\psi})$ ) are obtained as

$$\begin{aligned}\hat{Y}(\mathbf{X}_*|\mathbf{X}_{tr}, \mathbf{Y}(\mathbf{X}_{tr}), \hat{\psi}) \\ = \mathbf{K}(\mathbf{X}_*, \mathbf{X}_{tr}|\hat{\psi}) \mathbf{K}(\mathbf{X}_{tr}, \mathbf{X}_{tr}|\hat{\psi})^{-1} \mathbf{Y}(\mathbf{X}_{tr}),\end{aligned}\quad (5)$$

$$\begin{aligned}s^2(\mathbf{X}_*|\mathbf{X}_{tr}, \mathbf{Y}(\mathbf{X}_{tr}), \hat{\psi}) = \mathbf{K}(\mathbf{X}_*, \mathbf{X}_*|\hat{\psi}) \\ - \mathbf{K}(\mathbf{X}_*, \mathbf{X}_{tr}|\hat{\psi}) \mathbf{K}(\mathbf{X}_{tr}, \mathbf{X}_{tr}|\hat{\psi})^{-1} \mathbf{K}(\mathbf{X}_{tr}, \mathbf{X}_*|\hat{\psi}),\end{aligned}\quad (6)$$

where  $\mathbf{K}(\cdot, \cdot|\hat{\psi})$  is the covariance kernel as a function of the optimal hyper-parameters ( $\hat{\psi}$ ).

## 2.2. Compositional Kernel Learning (CKL)

GP has been widely used as the stochastic approximation of the simulation model within a Bayesian perspective [3,11]. For surrogate modeling, the GP has widely employed with a squared-exponential kernel (also known as Gaussian kernel) in the most previous studies [2,3,6,56,57]. The prior belief of using the squared-exponential kernel is to assume that the response surfaces are smooth and stationary. However, Bastos, et al. [3] mentioned that the failure of the stationary assumption may result in an inaccurate surrogate model. Although choosing a proper covariance kernel is crucial for successful surrogate modeling, there is little research for surrogate modeling to specify the optimal covariance kernel.

**Fig. 1** shows various types of the covariance kernels. Hereafter, these covariance kernels are referred to as “base-kernel”. The base-kernels include Linear kernel (LIN), constant kernel (CON), Periodic kernel (PER), Squared-Exponential kernel (SE), Rational-Quadratic kernel (RQ), Matérn-3/2 kernel (ME3), and Matérn-5/2 kernel (ME5). **Appendix A** describes the hyper-parameters in the covariance kernels and their roles to encode our prior belief on the response surface. Since these base-kernels can represent only one characteristic of the function, it is not efficient to learn the complex response surface (e.g., polynomial trend with periodic pattern).

A compositional Kernel (CK) is a useful representation to express richer covariance kernels [26] for GPs. The CK can be constructed using compositional rules with a set of the base-kernels that should be positive-semidefinite. If two GPs with zero mean are independent, the covariance kernels of the GPs are known to be closed under composition rules [52]. The first operation of the compositional rule is the sum of the multiple covariance kernels. This is based on an additive assumption to define the GP using the CK as a superposition of independent GPs. A new CK from the compositional rule of the sum is defined as

$$\mathbf{K}_s(\mathbf{X}, \mathbf{X}'|\psi_g, \psi_h) = \mathbf{K}_g(\mathbf{X}, \mathbf{X}'|\psi_g) + \mathbf{K}_h(\mathbf{X}, \mathbf{X}'|\psi_h). \quad (7)$$

The sum of the covariance kernel is equivalent to the sum of the GPs. Then, this can be represented by

$$\mathbf{f}(\mathbf{X}) = \mathbf{g}(\mathbf{X}) + \mathbf{h}(\mathbf{X}) \sim \mathbf{GP}(0, \mathbf{K}_s(\mathbf{X}, \mathbf{X}'|\psi_g, \psi_h)), \quad (8)$$

where  $\mathbf{g}(\mathbf{X}) \sim \mathbf{GP}(0, \mathbf{K}_g(\mathbf{X}, \mathbf{X}'|\psi_g))$  and  $\mathbf{h}(\mathbf{X}) \sim \mathbf{GP}(0, \mathbf{K}_h(\mathbf{X}, \mathbf{X}'|\psi_h))$ . The other operation in the compositional rule is the product of the multiple covariance kernels. The compositional rule of the product accounts for interactions between different inputs or different notions of similarity. A new CK from the compositional rule of the product is defined as

$$\mathbf{K}_p(\mathbf{X}, \mathbf{X}'|\psi_g, \psi_h) = \mathbf{K}_g(\mathbf{X}, \mathbf{X}'|\psi_g) \times \mathbf{K}_h(\mathbf{X}, \mathbf{X}'|\psi_h). \quad (9)$$

Similarly to Eq. (8), Eq. (9) is equivalent to the product of the GPs and this can be represented by the GP in Eq. (10).

$$\mathbf{f}(\mathbf{X}) = \mathbf{g}(\mathbf{X}) \times \mathbf{h}(\mathbf{X}) \sim \mathbf{GP}(0, \mathbf{K}_p(\mathbf{X}, \mathbf{X}'|\psi_g, \psi_h)). \quad (10)$$

Since the compositional rules for the CK construct richer covariance kernels, the CK can represent the complex properties of the response surface (which is not captured by a base-kernel). For example, multiplying LIN kernels can yield a new covariance kernel encoding for cubic trend in the input space. **Fig. 2** shows two examples of the CK.

Since no one knows the underlying structure of the response surface, choosing a proper kernel is still challenging even for experts. One simple way to choose a proper covariance kernel is based on trial-and-error method by trying out a set of the different covariance kernels and selecting the best one among them. However, this iterative process is both tedious and sub-optimal. To discover the optimal structure of the covariance kernel, Duvenaud, et al. [52] developed a Compositional Kernel Learning (CKL) termed the structure discovery algorithm through a compositional kernel search for time-series problems. Variants of this algorithm were proposed in [12,53,54]. In this method, base-kernels include the white noise, constant, linear, squared-exponential, and periodic kernels. To deal with abrupt changes in time series, the compositional rule for the CK includes not only the sum and product but also the change-point operation. A tree-search algorithm over possible CKs is performed based on a greedy search. The tree-search algorithm expands the current CK using compositional rules with base-kernels. Once a new CK is constructed, the hyper-parameters for the CK are optimized by minimizing Eq. (3). As the model selection criterion, Bayesian Information Criterion (BIC) is computed based on the marginal likelihood to score the new CKs [58]. Among the new CKs, the best CK is selected and then it is expanded using all compositional operations to the base-kernels. This tree-search procedure is repeated, until no more better BIC is obtained or a predetermined maximal search depth is reached [54]. This search strategy is referred to as a best-first-search to explore a tree by expanding the most promising node (i.e., smaller BIC) with pruning the other branches, as shown in **Fig. 3**.

$$BIC = -2 \log(p(D|M)) + p \log N, \quad (11)$$

where  $p(D|M)$  denotes optimized marginal likelihood;  $p$  and  $N$  denote the size of hyper-parameters in the CK and sample size, respectively.

## 2.3. Progressive Latin Hypercube sampling (PLHS)

Single-stage sampling strategy such as Latin Hypercube Sampling (LHS) [36] generates entire samples at once, so that it is effective only if the proper size of the samples is fairly confident to be pre-specified. However, the proper size of the samples is not generally known a priori. On the other hand, multi-stage sampling strategies enable sequential generation of the samples by preserving the distribution properties. Progressive Latin Hypercube Sampling (PLHS) [5] is a novel multi-stage version of the sequential LHS. The PLHS successively generates a series of smaller sub-samples such that both entire samples and sub-samples at each stage in the PLHS remain Latin hypercube along with space-filling properties.

The quality of the Latin hypercube property can be evaluated using Eq. (13). Suppose that the  $p$ -dimensional input space ( $\mathbf{X} \in [0, 1]^p$ ) is divided into  $n$  disjoint intervals ( $[(0, 1/n], [1/n, 2/n], \dots, [(n-1)/n, 1]]$ ) indexed by  $q$ . The sample matrix ( $\mathbf{S}(n, p)$ ) consists of  $n \times p$  elements ( $x_{i,j} \in [0, 1]$ , where  $i = 1, \dots, n$  and  $j = 1, \dots, p$ ). As given in Eq. (12), the auxiliary

Base kernel	Kernel function <sup>a)</sup>	Kernel density	Generated samples	Encoding function
Linear kernel (LIN)	$\sigma_b^2 + \sigma_0^2 x \cdot x'$			Linear trend
Constant kernel (CON)	$\sigma^2$			Constant
Periodic kernel (PER)	$\sigma^2 \exp\left(-\frac{2 \sin^2(\pi x-x' /p)}{l^2}\right)$			Repeating pattern
Squared-exponential kernel (SE)	$\sigma^2 \exp\left(-\frac{(x-x')^2}{2l^2}\right)$			Smoothness with local variation
Rational-quadratic kernel (RQ)	$\sigma^2 \left(1 + \frac{(x-x')^2}{2al^2}\right)^{-\alpha}$			Smoothness with multi-scale variation
Matérn-3/2 kernel (ME3)	$\sigma^2 \left(1 + \frac{r\sqrt{3}}{l}\right) \exp\left(-\frac{r\sqrt{3}}{l}\right)$ where $r = (x - x')$			Smoothness With different level of roughness variation
Matérn-5/2 kernel (ME5)	$\sigma^2 \left(1 + \frac{r\sqrt{5}}{l} + \frac{5r^2}{3l^2}\right) \exp\left(-\frac{r\sqrt{5}}{l}\right)$ where $r = (x - x')$			Smoothness With different level of roughness variation

a) the red indicates hyper-parameters

Fig. 1. Base-kernels and their encoding functions.

Operator	Kernel density	Generated samples	Encoding function
Sum (Addition)			Linear trend with repeating pattern
			Polynomial trend

Fig. 2. Compositional kernels and their encoding functions.

binary variables ( $u_{q,j}$ ) are introduced to the projective properties in the one-dimensional marginal distribution (i.e., projected sub-space in Fig. 4).

$$u_{q,j} = \begin{cases} 1, & \text{If any } i \text{ exists for which } x_{i,j} \text{ lies in the interval } q \\ 0, & \text{Otherwise} \end{cases} \quad (12)$$

If the condition in Eq. (13) is satisfied,  $\mathbf{S}(n, p)$  is Latin hypercube.

$$\frac{\sum_{j=1}^p \sum_{q=1}^n u_{q,j}}{n \cdot p} = 1. \quad (13)$$

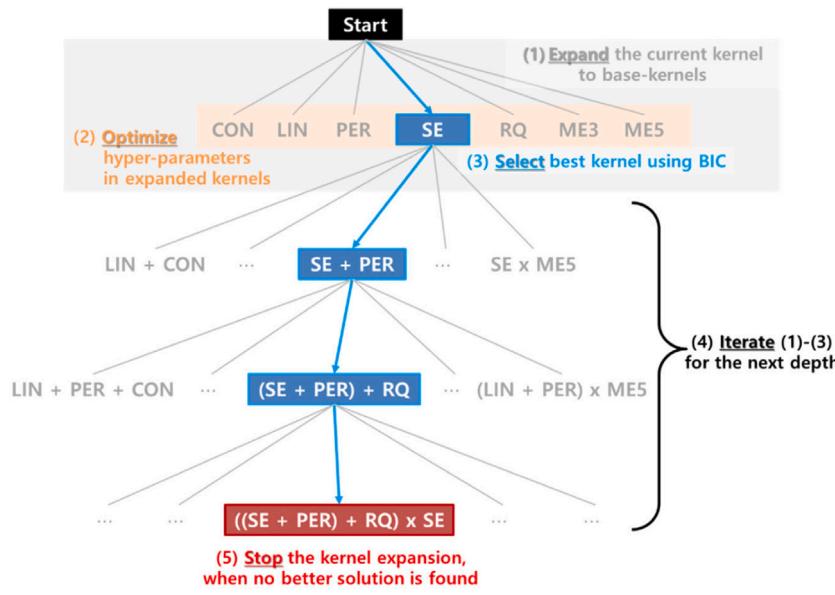
To extend Eq. (13) to a series of sub-samples in the PLHS ( $T$ ), a set of the auxiliary binary variables ( $u_{q,j}^t$ ) is used with the

corresponding sample matrix ( $\mathbf{S}^t = \cup_{k=1}^t \mathbf{S}_k(n_k, p)$ ) in Eq. (14).

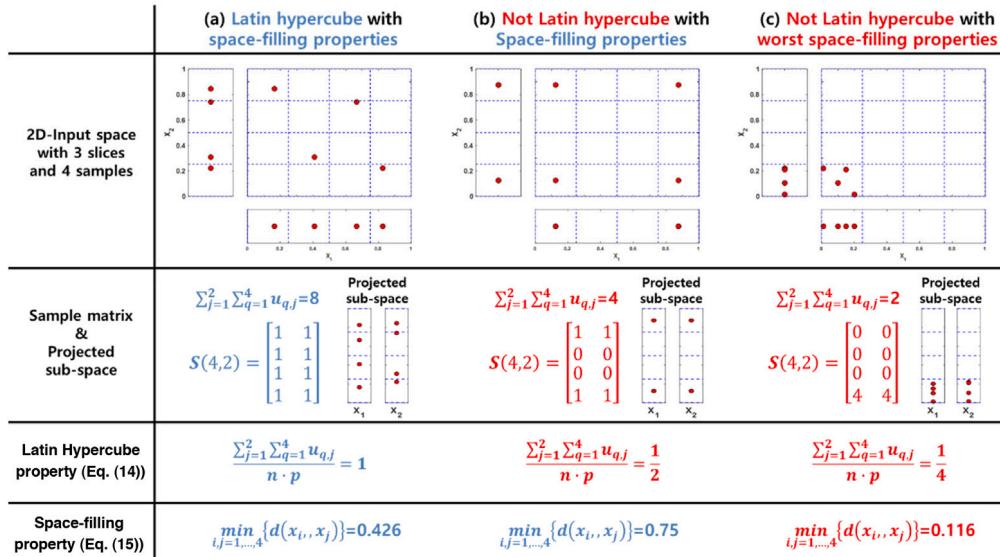
$$\sum_{t=1}^T \frac{\sum_{j=1}^p \sum_{q=1}^{n_t} u_{q,j}^t}{n_t \cdot p} = T, \quad (14)$$

where  $n_k$  and  $T$  denotes the number of the sub-samples and total stage in the PLHS, respectively;  $n_t$  is the total size of the sub-samples up to  $t$ th stage ( $n_t = \sum_{k=1}^t n_k$  with  $t = 1, \dots, T$ ); and the input space ( $\mathbf{X} \in [0, 1]^p$ ) is divided into  $n_t$  disjoint intervals ( $[[0, 1/n_t], [1/n_t, 2/n_t], \dots, [(n_t-1)/n_t, 1]]$ ) indexed by  $q$  ( $q = 1, \dots, n_t$ ); The sample matrix ( $\mathbf{S}_k(n_t, p)$ ) consists of  $n_t \times p$  elements ( $x_{i,j} \in [0, 1]$ , where  $i = 1, \dots, n_t$  and  $j = 1, \dots, p$ ).

The sub-samples are generated based on Sliced Latin Hypercube Sampling (SLHS) [59]. The SLHS is a special Latin hypercube design that can be partitioned into slices of smaller Latin hypercube designs [60]. To improve the space-filling property, the *maximin* distance criterion in Eq. (15) [38] is introduced when



**Fig. 3.** Compositional kernel learning (CKL) method via best-first-search algorithm.



**Fig. 4.** Different configurations of samples with  $n = 4$  and  $p = 2$ .

generating the sub-samples ( $\mathbf{S}_k(n_k, p)$ ) from the SLHS.

$$\mathbf{S}_k(n_k, p) = \operatorname{argmax} \left( \min_{i,j=1,\dots,n_k} \{d(\mathbf{x}_i, \mathbf{x}_j)\} \right), \text{ where } i \neq j, \quad (15)$$

where  $d(\cdot, \cdot)$  denotes Euclidean distance measure. Once the candidates of the sub-samples are generated by the SLHS with the *maximin* distance criterion, a series of the sub-samples ( $\mathbf{S}_k(n_k, p)$ ) are evaluated by maximizing the following function in Eq. (16).

$$\begin{aligned} \mathbf{PLHS}(n_t, p, T) &= \bigcup_{k=1}^T \mathbf{S}_k(n_k, p) \\ &= \operatorname{argmax} \left( \sum_{t=1}^T \frac{\sum_{j=1}^p \sum_{q=1}^{n_t} u_{q,j}^t}{n_t \cdot p} \right). \end{aligned} \quad (16)$$

Maximizing Eq. (16) is equivalent to maximizing PLHS property (i.e., progressively preserving the projective properties) with the space-filling property. In this context, the greedy search algorithm is employed to find a near optimal solution for optimizing Eq. (16). As a result, entire samples ( $\mathbf{PLHS}(n_t, p, T)$ ) have the desired properties for the distribution.

The algorithm for the PLHS consists of two steps. Firstly, the SLHS is used to generate a set of  $T$  sub-samples by maximizing Eq. (15). Then, the arrangement of the sub-samples is permuted to maximize Eq. (16) by combining the sub-samples sequentially. These steps are repeated with different initial random seeds for the SLHS until the best samples are found.

Fig. 5 shows the PLHS on a two-dimensional input space with three stages using the four sub-samples. Initial four samples (blue circle) are generated based on the grid of strata (slice #1 plotted by blue line) to locate each sample on every one-fourth of space in each input. Then, the four samples (red circle) are generated and added to initial four samples (blue circle). In the slice #2 (red line), every one-eighth of space at each input contains one sample. Lastly, the additional four samples (green circle) are generated based on the slice #3 (green line) to ensure that one sample is assigned to every one-twelfth of space in each input. This illustration shows that the PLHS sequentially generates both initial samples, sub-samples, and entire samples by maintaining the nested design.

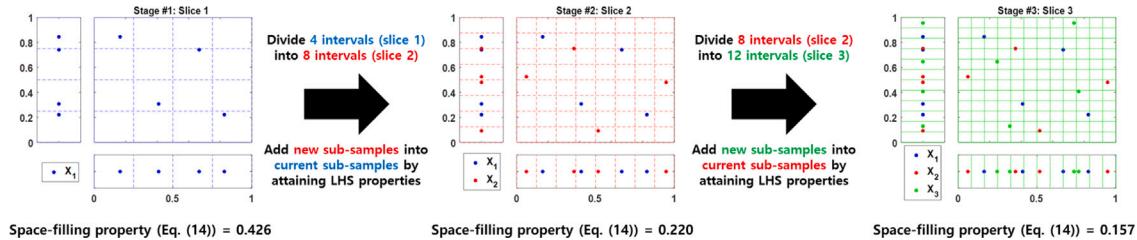


Fig. 5. Illustration of Progressive Latin hypercube sampling.

### 3. Proposed method using CKL and PLHS for surrogate modeling

Although the GP has been widely used for the surrogate modeling due to its advantages (e.g., prediction uncertainty), there still remains three difficulties as follows: (1) choice of the proper covariance kernel; (2) appropriate size of the training samples; and (3) diagnostics for accuracy. To address these difficulties simultaneously, this study proposed an efficient surrogate modeling method based on GP by integrating the CKL with the PLHS. Fig. 6 shows the flowchart of the proposed method. The proposed method consists of the two-stage as: (1) sampling-stage and (2) model-fitting stage.

In the sampling stage, the PLHS is employed to augment the training samples for improving the global accuracy of the GP. In order to improve the GP with its diagnostics, the proposed method utilizes two consecutive sub-samples in the series of the PLHS at each stage. The GP is constructed using the 1st sub-samples as training samples (blue circle in Fig. 7(a)). The fitted GP using the 1st sub-samples is used to predict the outputs of the 2nd sub-samples (red circle in Fig. 7(b)), which is not used for the fitting (validation samples) to evaluate the accuracy of the GP (model-diagnostic). According to Eq. (17), the Normalized Mean Squared Error (NMSE) is computed using validation samples. The smaller NRSME indicates the prediction model is more accurate. The size of the validation samples ( $n_{val}$ ) is equivalent to the size of the sub-samples ( $n_k$ ). Once the NMSE for the validation samples is computed, the 1st and 2nd sub-samples are combined to the training samples, as shown in Fig. 7(c). At this stage, the 3rd sub-samples (black circle in Fig. 7(c)) are used as the validation samples. The GP is fitted using the augmented training samples (i.e., 1st and 2nd sub-samples), and the fitted GP is used to predict the outputs of the 3rd sub-samples (new validation samples). The procedure is repeated by augmenting the training samples with new validation samples. By progressing the stage in the PLHS, the NMSE for the validation samples can be plotted against the size of the training samples (trace-plot in Fig. 7(e)). In Section 5.2, it is empirically shown that the NMSE from validation samples can be used as a diagnostic measure for the global accuracy of the GP. If the desired accuracy of the GP is obtained, the values of the NMSE do not change or change to limit degrees (i.e., small fluctuation to certain values of the NMSE). Therefore, the trace-plot in Fig. 7(e) allows users to monitor the diagnostics of the global accuracy and assess the stopping criteria for further sampling.

In the model-fitting stage, the CKL is employed to discover the optimal covariance kernel automatically to maximize the learning capability for response surfaces. The CKL does not require any explicit representation of the characteristics of the response surface (i.e., inductive bias). As mentioned in Section 2.2, the CK is defined as sums and products of the base-kernels. Any positive semi-definite kernels can be used as the base-kernels and these kernels can be composed to make richly structured kernels (i.e., CK). Considering there is no uncertainty in the outputs of the training samples (i.e., deterministic simulation), seven base-kernels are considered for the proposed method as follows: (1)

Linear kernel (LIN), (2) Constant kernel (CON), (3) Periodic kernel (PER), (4) Squared-exponential kernel (SE), (5) Rational-quadratic, (6) Matérn-3/2 kernel (ME3), and (7) Matérn-5/2 kernel (ME5). More details for the CKL can be found in [52–54]. By integrating the CKL with the PLHS, the proposed method can maximize the efficiencies for the learning capability and sampling. The CKL can discover the optimal covariance kernel under given training samples, while the framework of the PLHS (nested samples) can allow the monitoring for the diagnostics of the GP.

## 4. Research methodology

### 4.1. Test function and their characteristics

To compare the proposed method with other methods, nine test functions were selected from literatures to cover the range of dimensionality and complexity. The mathematical representations of the test functions are summarized in Appendix B. These test functions are well-known benchmark problems in surrogate modeling and optimization problems. Table 3 summarizes the characteristics of the test functions. In terms of the dimensionality ( $d$ ), the test functions can be categorized into three levels: (1) Small-scale ( $d \leq 4$ ): 3 test functions; (2) Medium-scale ( $5 \leq d \leq 9$ ): 3 test functions; and (3) Large-scale ( $d \geq 10$ ): 3 test functions.

In order to classify functional complexity, a full second-order polynomial model was fitted using the test samples (as tabulated in Table 6). Based on the values of the adjusted  $R^2$  ( $R_{test}^2$ ), the functional complexity was divided into three levels:

- Low-order complexity ( $R_{test}^2 \geq 0.99$ ): 3 test functions
- Intermediate-order complexity ( $0.92 < R_{test}^2 < 0.99$ ): 3 test functions
- High-order complexity ( $R_{test}^2 \leq 0.92$ ): 3 test functions

### 4.2. Selection for surrogate modeling techniques

Based on Table 1, a set of surrogate modeling was selected to compare with the proposed methods in terms of accuracy and computational costs. Although neural networks are also commonly used for the surrogate modeling, the neural networks limit with sample size severely [14]. For this reason, the neural networks were not considered in this study. The GEK only requires additional function evaluations to obtain gradient information via the finite difference method in practice. To set the identical number of the function evaluation, the GEK was not included. Therefore, ten surrogate models were considered as tabulated in Table 4. These models can be categorized into two classes as: (1) fixed basis and (2) adaptive basis. The class of the fixed basis requires only a single-fitting, while the class of the adaptive basis requires iterative-fitting to search the optimal structures of the adaptive basis (i.e., mean or covariance function in GP).

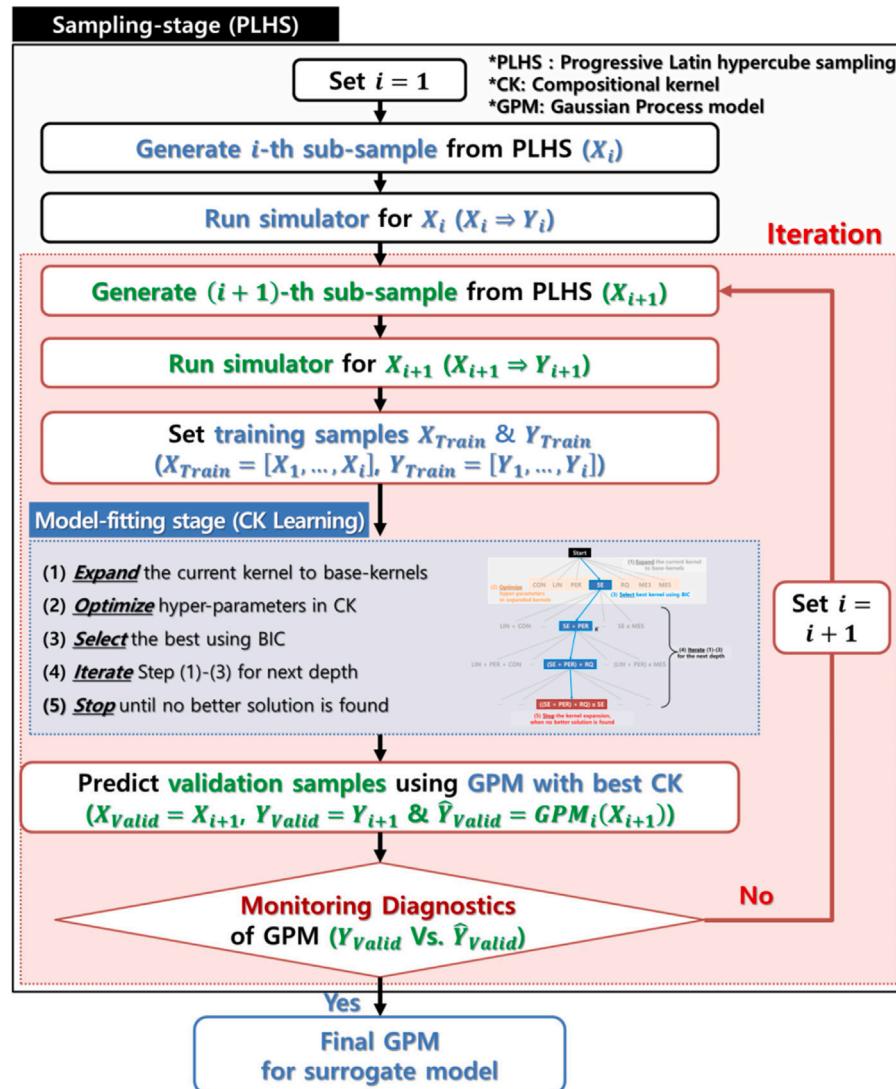


Fig. 6. Flowchart of proposed method using CKL and PLHS for surrogate modeling..

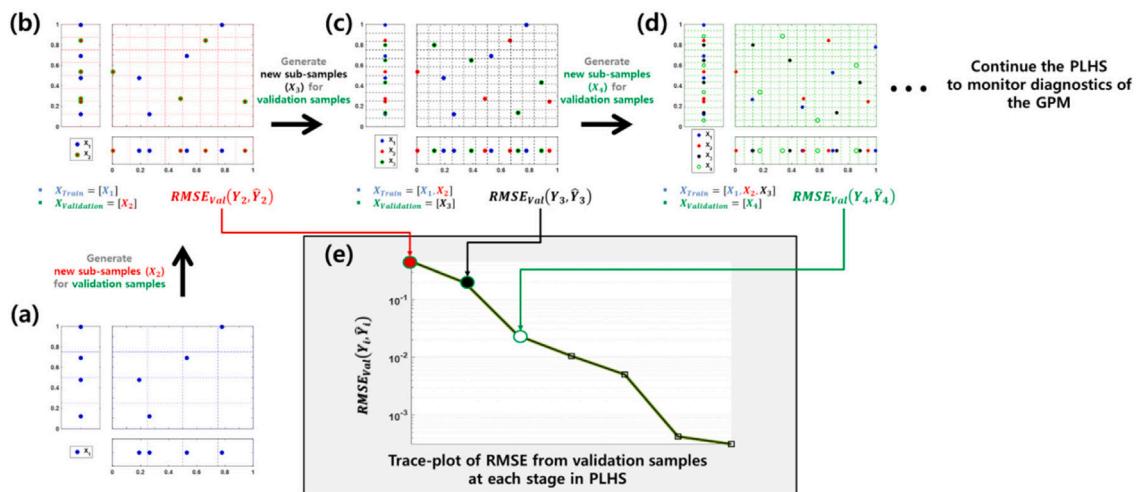


Fig. 7. Diagnostics based on PLHS for GPs.

**Table 3**  
Test functions and their characteristics.

Test function	Problem type			$R^2_{test}^a$	Reference
	Scale	# dimension ( $d$ )	Complexity		
1	Small	2	Low	0.99	[61]
2		3	Intermediate	0.93	[62]
3		2	High	0.75	[63]
4	Medium	6	Low	0.99	[27]
5		8	Intermediate	0.94	[62]
6		5	High	0.92	[64]
7	Large	10	Low	1.00	[65]
8		10	Intermediate	0.95	[66]
9		20	High	0.87	[67]

<sup>a</sup>Adjusted R-squared value from full-second order polynomial models fitted by test samples (10,000 EA).

**Table 4**  
Selection of surrogate modeling technique.

# Method	Type	Abbreviation	Model basis	Fitting stage
1	Polynomial model	POLY <sup>a</sup>	2nd-order polynomial function with full interactions	Single (fixed basis)
2		rbfG <sup>a</sup>	Gaussian kernel	
3		rbfMQ <sup>a</sup>	Multiquadric kernel	
4		rbfTP <sup>a</sup>	Thin-plate kernel	
5		SE <sup>b</sup>	Constant mean with squared-exponential kernel	
6		MA3 <sup>b</sup>	Constant mean with Matérn-3/2 kernel	Single (fixed basis)
7	Gaussian process regression (also known as Kriging)	RQ <sup>b</sup>	Constant mean with rational quadratic kernel	
8		MAS <sup>b</sup>	Constant mean with Matérn-5/2 kernel	
9		BlindKRG <sup>c</sup>	Adaptive polynomial model as mean function with squared-exponential kernel	Iterative (adaptive basis for mean)
10		CKL <sup>d</sup>	Constant mean with adaptive compositional kernel	Iterative (adaptive basis for kernel)

<sup>a</sup>Source code is available from <https://optimizationcodes.wordpress.com/>.

<sup>b</sup>Source code is available from <http://www.gaussianprocess.org/gpml/code/matlab/doc/index.html>.

<sup>c</sup>Source code is available from <http://www.sumo.intec.ugent.be/ooDACE/>.

<sup>d</sup>Source code is developed based on GPML toolbox.

#### 4.3. Selection for sampling method

The proposed method (CKL) was examined by different sampling methods. Table 5 shows four sampling methods with their characteristics. These methods are commonly used for space-filling design to generate samples uniformly over an input space. The OLHS and SLHS generate samples (non-nested design) by optimizing space-filling criteria. On the contrary, the SOBOL and PLHS generate new samples based on existing samples by keeping their own properties for the desired distribution.

#### 4.4. Performance metrics

To evaluate the global accuracy of the surrogate model, test samples ( $\mathbf{X}_{test}$ ) were generated using optimal LHS (*maximin* distance criterion) with sufficiently large sample size ( $n_{test} = 10,000$ ). To assess the global accuracy of the surrogate model, Normalized Mean Squared Error (NMSE) was computed using test samples at each stage of the sampling. The NMSE aggregates the normalized discrepancy between predictions ( $\hat{\mathbf{Y}}(\mathbf{X}_{test})$ ) and true values ( $\mathbf{Y}(\mathbf{X}_{test})$ ) into a single measure. The NMSE is computed using Eq. (17).

$$NMSE_i = \frac{\sum (\mathbf{Y}(\mathbf{X}_i) - \hat{\mathbf{Y}}(\mathbf{X}_i))^2}{\max(\mathbf{Y}(\mathbf{X}_i)) - \min(\mathbf{Y}(\mathbf{X}_i))}. \quad (17)$$

## 5. Results and analysis

In order to account for sampling variability, different random seeds were used to generate ten different replicates of training samples. Based on the ten replicates, numerical experiments were performed to evaluate the robustness against the random components in the proposed methods. The proper size of the training samples depends on the types of the problems and computational budgets. Since there is no optimal way to determine the size of the training samples, the empirical formula (Eq. (18)) was introduced from Razavi, et al. [69] to determine the size of the sub-samples ( $n_k$ ).

$$n_k = \max(2(p+1), 0.1N_{max}), \quad (18)$$

where  $N_{max}$  is the maximal size of the training samples. Table 6 describes the test setup of the sampling in each test function. In Section 5.4, the influence of the hyper-parameters in the PLHS ( $n_k$ ) was investigated to evaluate its sensitivity.

#### 5.1. Performance evaluation of surrogate modeling techniques

In this section, the set of the surrogate modeling was evaluated with the PLHS using the test setup in Table 6. The boxplot is used to show the accuracy and robustness of the surrogate modeling simultaneously. The smaller value for the mean (red line in each box) indicates the better accuracy ( $NMSE_{Test}$ ), while the smaller

**Table 5**

Selection of space-filling sampling method.

Sampling method (abbreviation)	Space-filling criteria	Sampling- Stage	Nested design	Reference		
	<i>Distance</i> <sup>a</sup>	<i>L</i> <sub>2</sub> -discrepancy <sup>b</sup>	Entropy <sup>c</sup>	1D-Projection <sup>d</sup>		
Optimal LHS (OLHS)	✓			Single	Non-nested	[39]
Symmetric LHS (SLHS)	✓		✓	Single	Non-nested	[43]
Sobol's Sequence by leaping and scrambling (SOBOL)		✓		Sequential	Nested	[37,68]
Progressive LHS (PLHS)	✓		✓	Sequential	Nested	[5]

<sup>a</sup>Maximize the inter-point distance criterion.<sup>b</sup>Minimize *L*<sub>2</sub>-discrepancy.<sup>c</sup>Maximize the entropy criterion.<sup>d</sup>Maximize stratification in any one-dimensional projection.**Table 6**

Test setup for sampling methods.

Test function	# Sub-samples ( $n_k$ )	# Stage ( $T$ )	# Total sub-samples ( $N_{\max} = n_k \cdot T$ )	# Test samples ( $n_{\text{test}}$ )
1				
2	10		100	10,000
3				
4	20		200	
5	40		400	10,000
6	20		200	
7				
8	50		500	10,000
9				

variance (range of the box) represents higher robustness of the surrogate modeling. For simplicity, the abbreviations are adopted from Table 4. Because of space limitation, the results of the three stages are only presented in this section. The full results are given in Appendix C.

Fig. 8 summarizes the prediction performance ( $\text{NMSE}_{\text{Test}}$ ) in the small-scale problems. The POLY2 does not improve the  $\text{NMSE}_{\text{Test}}$  using more samples, since the POLY2 has the explicit form of the quadratic trend and it does not have enough flexibility to learn the response surfaces. For all test functions, the GPs using fixed basis (except for Matern kernels) work slightly better than those of the RBF models (rbfG, rbfTP and rbfMQ). The rbfMQ provides the best performance among the RBF models, while the SE and RQ perform best among four GP using fixed basis (SE, RQ, MA3 and MA5). The BlindKRG provides better performance over the SE and RQ except for Test function #2. It is worth noting that the CKL generally outperforms or performs similarly to the best one among the others (except for Test function #3).

Fig. 9 shows the results of the medium-scale problems. Similar to the small-scale problems, the POLY2 does not improve the global accuracy with more samples. It is worth noting that the RBF models provide the similar performance of the POLY2 in Test functions #4 and 5. For Test function #6, they perform slightly better than the POLY2. However, they are not as good as the GPs using fixed basis (except for MA3 in Test functions # 5 and 6). Although the BlindKRG utilizes adaptive basis for the mean function, their prediction performances were not good as those of the SE and RQ. On the contrary, the CKL outperforms the others for all test functions.

The prediction performances of the large-scale problems are illustrated in Fig. 10. In Test function #7, the POLY2 and BlindKRG outperform than other methods. Test function #7 is expressed by the full second-order polynomial model (See Appendix B), so that the POLY2 and BlindKRG can represent the response surface

exactly. As a result, they provide much lower  $\text{NMSE}_{\text{Test}}$  than other methods, even for smaller samples. The RBF models and GPs using fixed basis provide similar prediction performances for Test functions #7 and 8. The SE and RQ perform slightly better than the RBF models. Overall, the CKL outperforms the SE and RQ across the test functions.

The observations from the results in Figs. 9–11 can be summarized as:

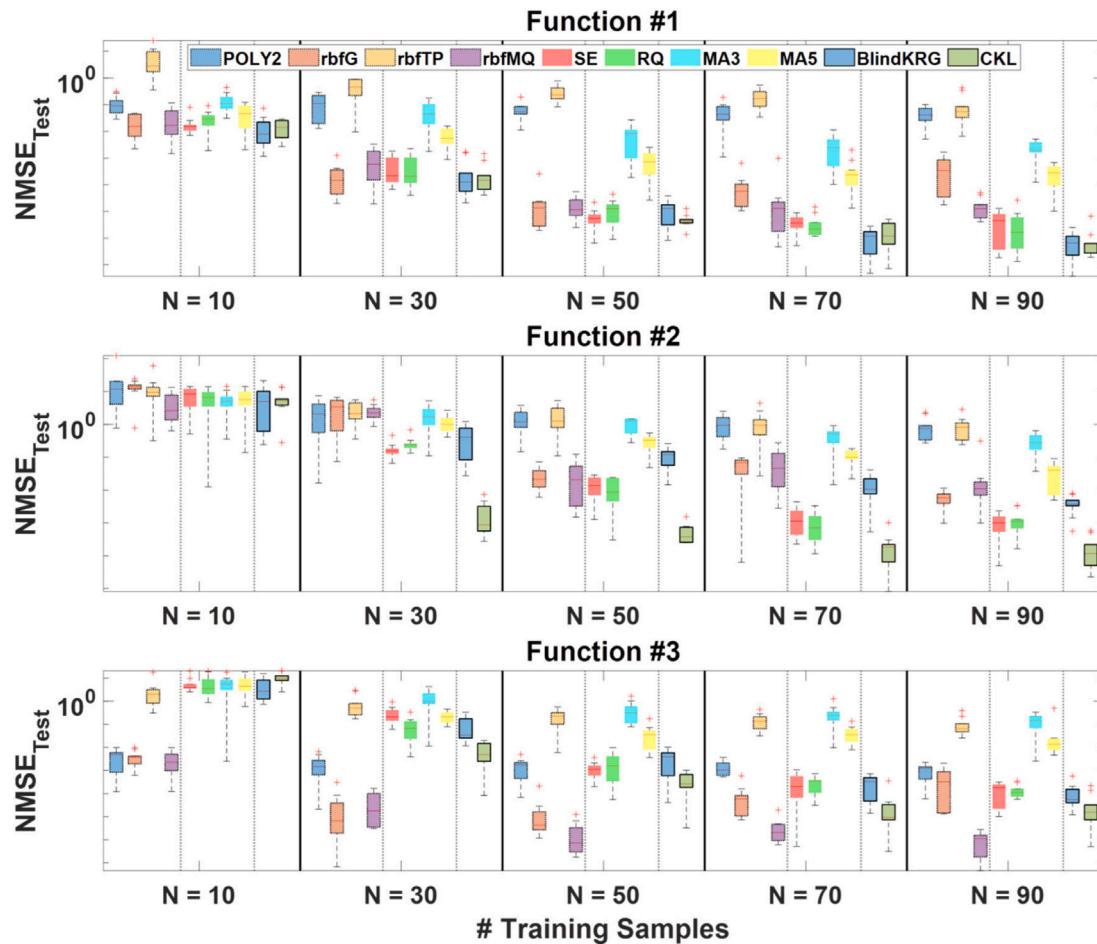
- The polynomial model cannot construct good surrogate models regardless of the types of problems, except for Test function #7. To learn more complex response surfaces, the polynomial model with higher-order is necessary. However, this exponentially increases the number of the parameters to be inferred from the samples especially in the large-scale problems.

- The RBF models can provide more flexible learning than the polynomial model. However, their kernel functions are isotropic, so that the relative importance of the input cannot be reflected to construct the surrogate models. Therefore, their learning capability is still limited.

- The GPs using the fixed basis generally provide similar or better prediction performances than the RBF models. Since their kernel functions assign different weighting factors (correlation length-scale for each input), they flexibly represent complex response surfaces than the RBF models. However, the MA3 and MA5 perform worse than the RBF models in this numerical experiment. This indicates that the GPs using fixed basis are very sensitive to its basis depending on the problem types.

- The BlindKRG generally performs similar to the GP using the fixed basis in this numerical experiment. When the distinct trend in response surfaces (e.g., the quadratic trend in Test function #7) exists, it significantly improves the prediction performance with fewer samples. Otherwise, the adaptive basis for mean function cannot improve the prediction performance significantly.

- The CKL performs best regardless of the problem types. It is worth noting that the CKL can learn various response surfaces



**Fig. 8.** Prediction performances of surrogate models for small-scale problems.

with fewer samples than other methods. Thus, it can be concluded that the CKL exhibits more flexible and efficient learning capability over the existing methods.

### 5.2. Performance evaluation for diagnostic measure

This section investigates empirical evidence on the correlation between  $\text{NMSE}_{\text{Validation}}$  and  $\text{NMSE}_{\text{Test}}$  in the surrogate models with the PLHS. Based on the results in Section 5.1, the  $\text{NMSE}_{\text{Validation}}$  and  $\text{NMSE}_{\text{Test}}$  were computed at each stage. The linear regression was performed using pairs of the  $\text{NMSE}_{\text{Validation}}$  and  $\text{NMSE}_{\text{Test}}$  over the stages. To analyze their correlation of the surrogate models, adjusted R-squared values ( $R^2$ ) were computed across the test functions. The  $R^2$  value of one indicates that the  $\text{NMSE}_{\text{Validation}}$  perfectly fits the  $\text{NMSE}_{\text{Test}}$ . In this context, the  $R^2$  value closer to one supports empirical evidence of the linear correlation between  $\text{NMSE}_{\text{Validation}}$  and  $\text{NMSE}_{\text{Test}}$ . Stated differently, the  $\text{NMSE}_{\text{Validation}}$  can be used as the diagnostic measure for the  $\text{NMSE}_{\text{Test}}$ .

Fig. 11 shows two realizations of the correlation between  $\text{NMSE}_{\text{Validation}}$  and  $\text{NMSE}_{\text{Test}}$  in Test function #6. The color of the circle denotes the size of the training samples. The top panel of Fig. 11 (rbfTP) presents poor correlations between  $\text{NMSE}_{\text{Validation}}$  and  $\text{NMSE}_{\text{Test}}$ . The points were far away from the fitted linear regression model (red dotted line), so that the values of  $R^2$  are observed from 0.004 to 0.787. This indicates that the  $\text{NMSE}_{\text{Validation}}$  is less correlated or uncorrelated to the  $\text{NMSE}_{\text{Test}}$ . On the other hand, the bottom panel of Fig. 11 (CKL) presents good correlations between the  $\text{NMSE}_{\text{Validation}}$  and  $\text{NMSE}_{\text{Test}}$ . The higher values of  $R^2$  are observed (0.860–0.964), so that the  $\text{NMSE}_{\text{Validation}}$  is well

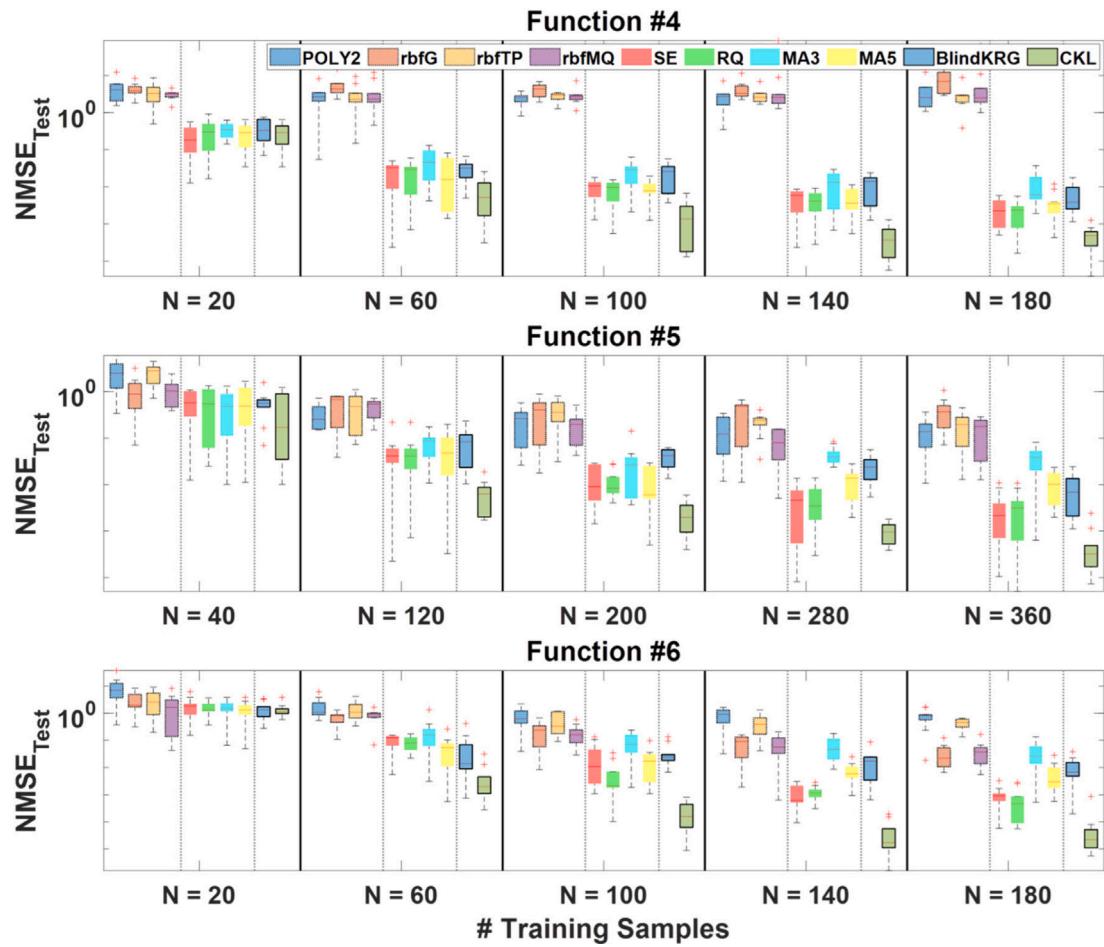
correlated to the  $\text{NMSE}_{\text{Test}}$ . Therefore, the  $\text{NMSE}_{\text{Validation}}$  from the CKL can be used as the diagnostic measure for  $\text{NMSE}_{\text{Test}}$ .

Fig. 12 shows  $R^2$  of the surrogate models across the test functions. It is observed that the CKL only provides robust correlation between  $\text{NMSE}_{\text{Validation}}$  and  $\text{NMSE}_{\text{Test}}$  across the test functions. The values of  $R^2$  in other surrogate models vary depending on the problem types. This indicates that the flexible learning capability (observed in the CKL) is the essential prerequisite to utilize the  $\text{NMSE}_{\text{Validation}}$  as the diagnostics measure for the global accuracy of the surrogate model.

### 5.3. Performance evaluation of sampling method

The performance of the sampling method directly related to the efficiency and robustness of the proposed method. In this context, this sub-section investigates the prediction performance by comparing the PLHS with other sampling methods in Table 5. The OLHS and SLHS generate the entire samples at once, while the SOBOL and PLHS generate nested samples sequentially. As observed in Section 5.2, flexible and efficient learning capability is one of the key factors to utilize the diagnostic measure for the global accuracy of the surrogate model. In this context, the CKL was implemented in this study. Similar to Section 5.1, the results of the three stages are only presented (full results are given in Appendix D).

Fig. 13 shows the prediction performances of the sampling methods across the test functions. In the small-scale problem (Fig. 13(a)), all methods perform similarly in Test functions #1 and #2. In Test function #3, the PLHS provides the relatively



**Fig. 9.** Prediction performances of surrogate models for medium-scale problems.

worse prediction performance than the others. However, the prediction performance of the PLHS is still reasonable to ensure the global accuracy (i.e.,  $\text{NMSE}_{\text{Test}} = 4.62e-5$  at  $N = 90$ ). In the medium and large-scale problems (Fig. 13(b) and (c)), all methods generally improve the prediction performance by adding more samples. The OLHS and SLHS slightly improve the prediction performance except for Test function #4, while the SOBOL fails to improve the prediction performance even with more samples (Test functions #5 and #7). On the other hand, the PLHS provides much better prediction performance than other methods with the faster convergence rate.

On the contrary to the single-stage SFS (OLHS and SLHS), the sequential SFS with the nested sampling is beneficial for diagnostics and computational efficiency. However, the SOBOL does not perform well in high-dimensional problems due to the poor projective properties [70]. On the other hand, the PLHS outperforms the others, regardless of the problem types. This empirical result indicates that the sequential SFS with both space-filling and other criteria (e.g., projective properties in the PLHS) is preferred.

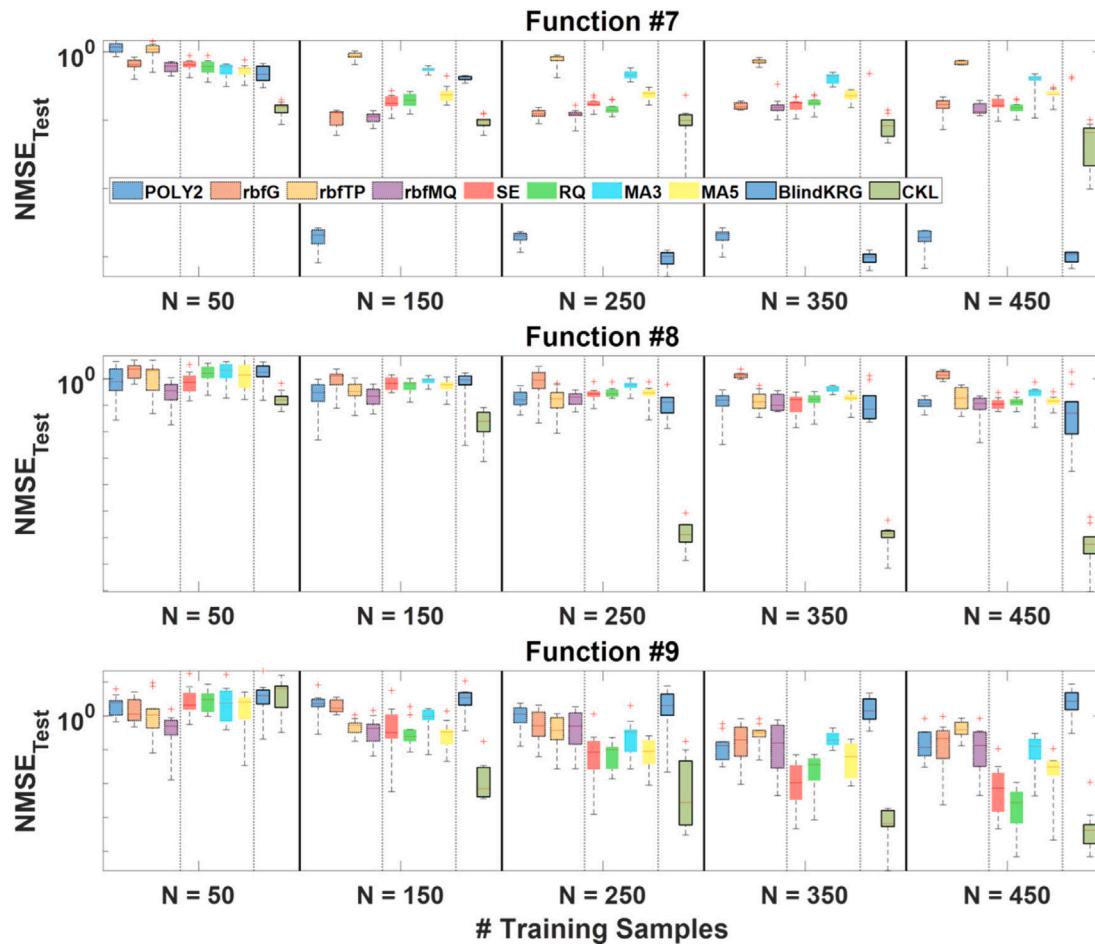
#### 5.4. Influence of hyper-parameter in sampling method

This sub-section investigates the influence of the hyper-parameters (size of sub-samples,  $n_k$ ) in the PLHS. Three case studies were conducted with the five test functions, as tabulated in Table 7. Fig. 14 shows the prediction performances for all case studies. For better visualization, the results are plotted at the nine stages to compare their prediction performances together. The

full results are available in Appendix E. Although the different hyper-parameters of the PLHS were used, the results from the case studies are similar across the test functions. In addition, the results show the similar convergence trend in each case study. Similar to Section 5.2, the  $R^2$  value between  $\text{NMSE}_{\text{Validation}}$  and  $\text{NMSE}_{\text{Test}}$  were computed for further investigation of their influence.

Fig. 15 shows the  $R^2$  values of the case studies. It is observed that the  $R^2$  values from all case studies are larger than 0.78, so that it can be concluded that the  $\text{NMSE}_{\text{Validation}}$  is reasonably correlated with the  $\text{NMSE}_{\text{Test}}$ . Therefore, this observation empirically indicates that the hyper-parameters in the PLHS does not affect the prediction performance and diagnostics performance for the global accuracy significantly.

It is worth noting that using more sub-samples generally produces the higher correlation (Case 3 in Fig. 15). However, this requires more function evaluations of the computationally expensive simulation. Therefore, it is reasonable to determine the size of the sub-samples based on the maximal number of the function evaluation ( $N_{\max}$ ). Since the computational budget is typically limited in engineering problems,  $N_{\max}$  can be determined based on allowable budgets. At this moment, this study empirically recommends that the trace-plot of the  $\text{NMSE}_{\text{Validation}}$  should be constructed with at least 10 stages to monitor convergence during the stage in PLHS. The optimal size of the sub-samples will be investigated as further study.



**Fig. 10.** Prediction performances of surrogate models for large-scale problems.

**Table 7**  
Test setups for different hyper-parameters in PLHS.

Case study #1			
# Test function	# Sub-samples ( $n_k$ )	# Stage ( $T$ )	# Total samples ( $N_{max} = n_k \cdot T$ )
2	5		100
3	5		100
4	10	20	200
6	10		200
9	25		500

Case study #2			
# Test function	# Sub-samples ( $n_k$ )	# Stage ( $T$ )	# Total samples ( $N_{max} = n_k \cdot T$ )
2	10		100
3	10		100
4	20	10	200
6	20		200
9	50		500

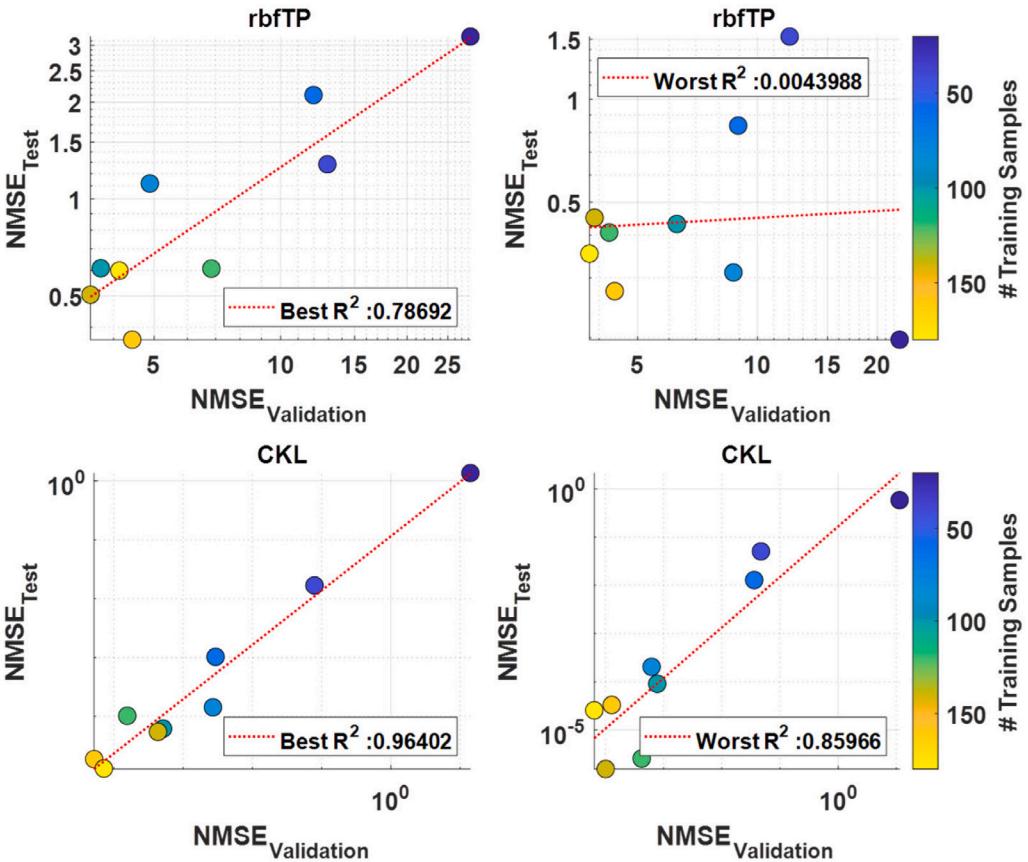
Case study #3			
# Test function	# Sub-samples ( $n_k$ )	# Stage ( $T$ )	# Total samples ( $N_{max} = n_k \cdot T$ )
2	20		100
3	20		100
4	40	5	200
6	40		200
9	100		500

### 5.5. Computational costs

To evaluate the computational cost, total execution time was measured at each stage under an identical environment for computation. Table 8 illustrates summary statistics of the computational time. Due to space limitation, the computation times on

the three stages are summarized. The full results are available in Appendix F.

The computational time for the POLY2 is quite fast regardless of the problem types (i.e., dimensionality and complexity) and sample size. As the sample size and dimensionality increases, the RBF models and GPs using fixed basis increase the computational



**Fig. 11.** Correlation between NMSE<sub>Validation</sub> and NMSE<sub>Test</sub> in Test function #6.

time. The complexity of the test function is not the main factor in computational complexity, since their computational times are similar regardless of the functional complexity. The GPs using adaptive basis (BlindKRG and CKL) are time-consuming, since they require the iterative-fitting to estimate their adaptive basis (i.e., mean or covariance function in GP). As a result, their computational times are significantly influenced by both dimensionality and sample size. It is observed that their computation time also increases in accordance with the functional complexity. To find optimal structures under the complex response surfaces, they take more iterative-fitting and this increases the computational time significantly.

The computational time of the CKL is much expensive than the others. Despite its computational complexity, the CKL provides flexible and efficient learning capability as observed in Section 5.1. Considering that it takes the time from several hours to weeks for simulating the computationally expensive model, the computational time of the CKL (i.e., 0.01–5200s) is negligible to the simulation time. Section 6 describes the technique to improve the computational efficiency of the CKL.

## 6. Discussion on proposed method

This section provides the following two issues related to the proposed method: (1) computational complexity and (2) limitation to discontinuous response surfaces.

### – Computational complexity

The computational complexity of the GP exponentially increases, according to the sample size ( $O(n^3)$ ). Although the CKL provides the superior learning capability as seen in Section 5.1, the iterative-fitting of the CKL aggravates the computational

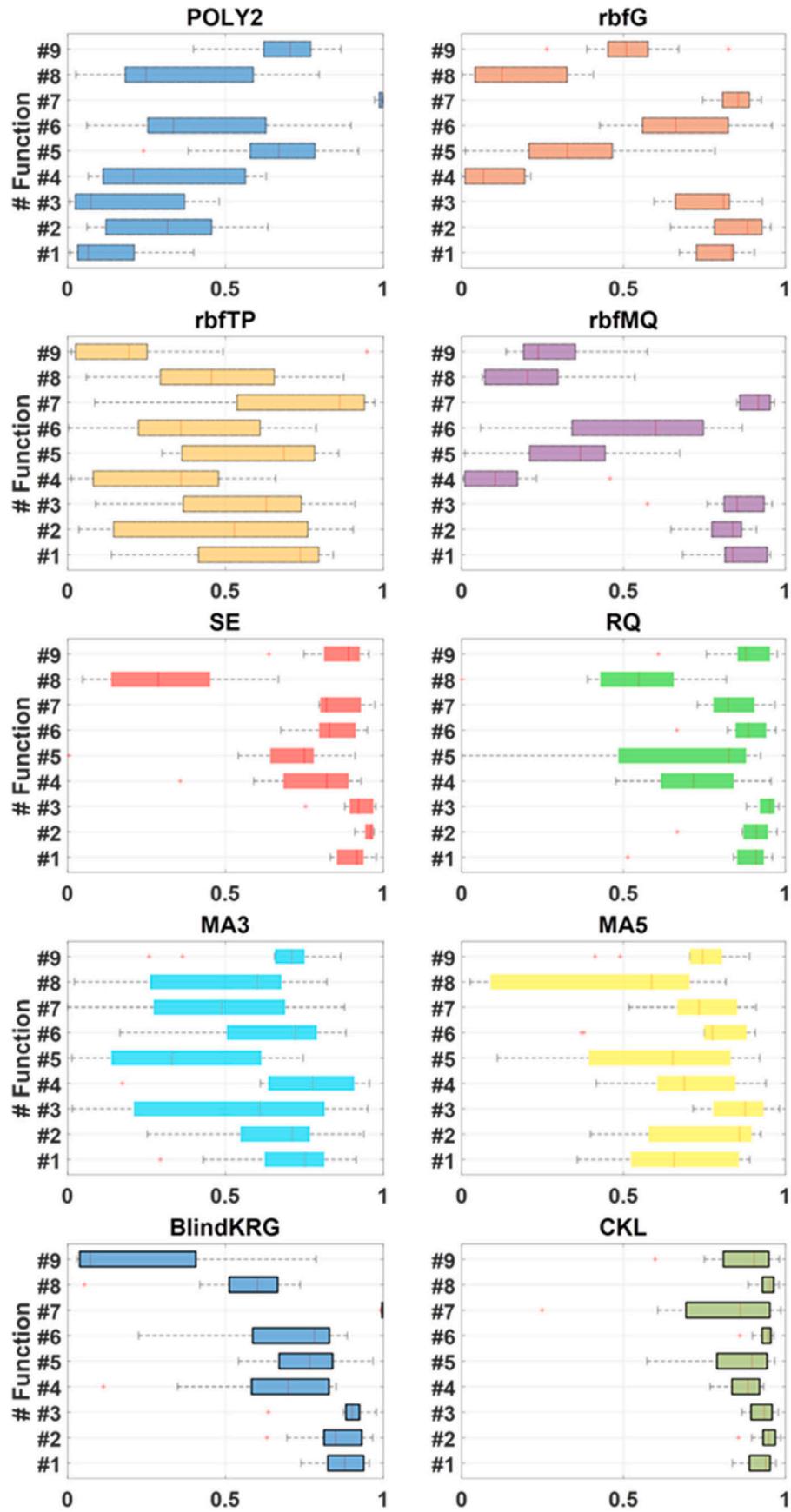
complexity. Therefore, the proposed method is computationally-intensive than others as shown in Table 8. To improve the computational efficiency, parallel processing using multi-core processors can be used for the tree-search algorithm in the CKL. The candidates of the compositional kernel are evaluated independently at each depth (step (1)–(3) in Fig. 3), so that the GP-modeling for the candidates can be employed simultaneously based on the parallel processing. By parallelizing the GP-modeling at each depth of the CKL, the computational complexity can be significantly improved.

### – Limitation to discontinuous response surfaces

Simulation models sometimes generate a discontinuous response surface due to unstable behaviors in physical systems. For example, buckling modes in structural systems induce unstable behaviors of the response due to material plasticity. As a result, the response surface has a switching condition as a function of the inputs. Based on the switching condition (i.e., discontinuities), two different states of the response surface can be realized. Although the CKL can learn the various types of response surfaces flexibly, it cannot capture the local variation due to discontinuities in response surfaces. Although the diagnostic measure from the proposed method (NMSE<sub>Validation</sub>) seems to be converged even for the discontinuous response surface, the proposed method has a risk of overfitting the regions near discontinuities. Therefore, the proposed method is not suitable for the simulation model with the discontinuous response surface. One solution of learning a discontinuous response surface is to incorporate a change point kernel into the base-kernel in the CKL.

### – Potential extension of proposed method

The CKL in the proposed method can be extended to adaptive sampling strategies of the global surrogate modeling [7,56,71–73]. Considering that the CKL is more efficient and flexible to learn response surfaces, adaptive sampling strategies can select

Fig. 12. R-squared values between NMSE<sub>Validation</sub> and NMSE<sub>Test</sub>.

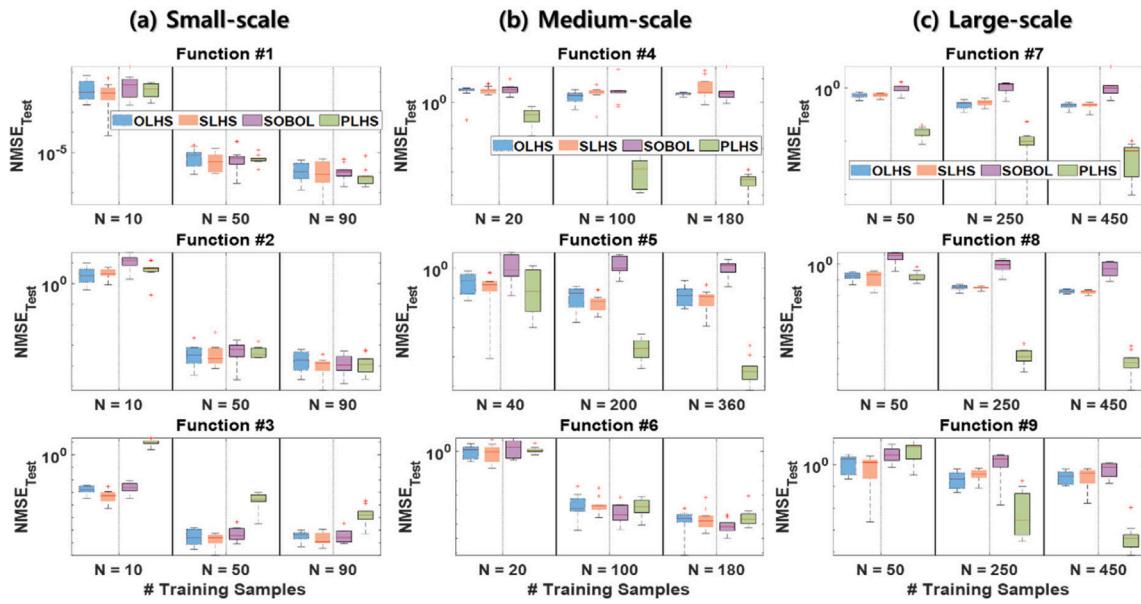


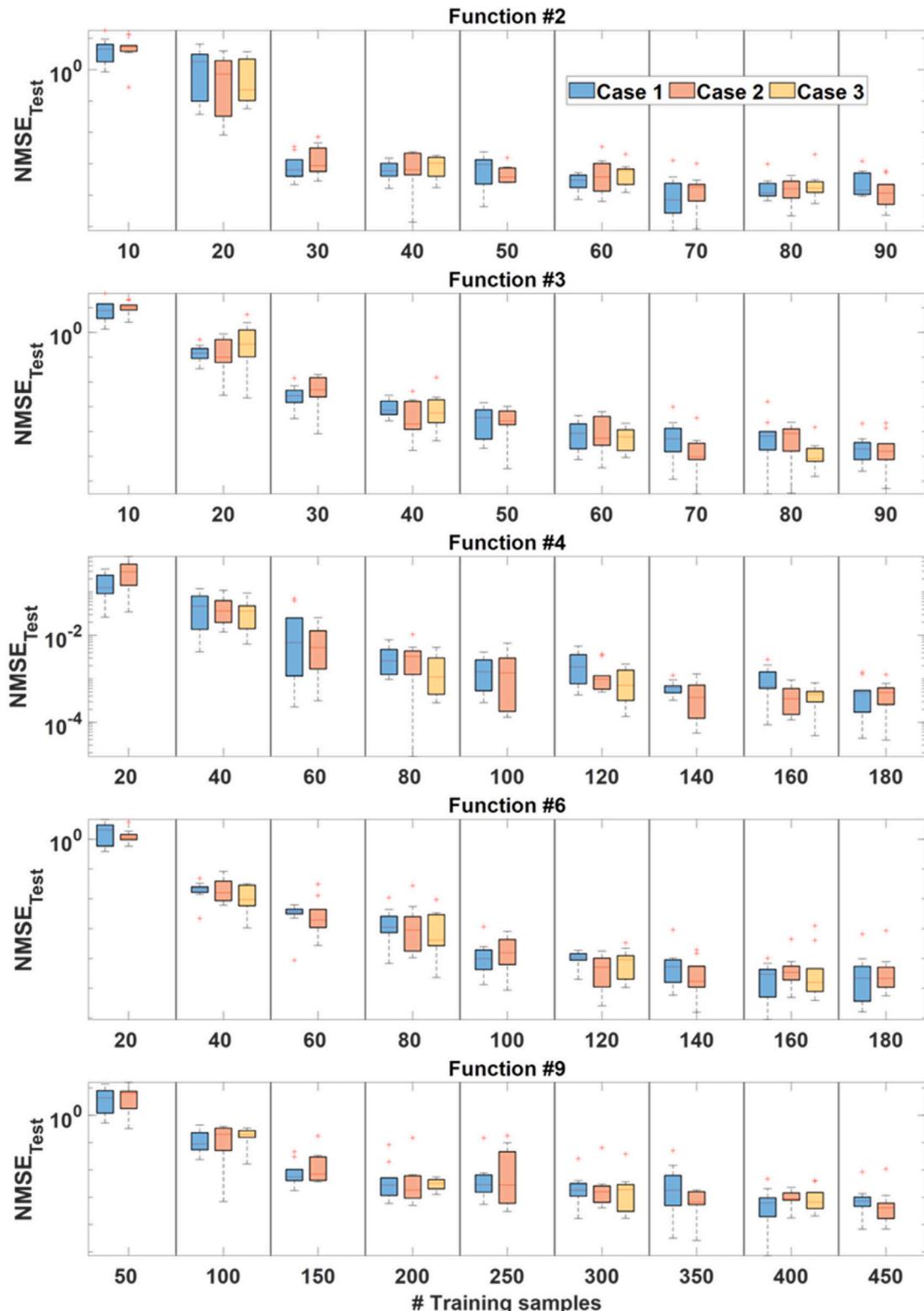
Fig. 13. Prediction performances of sampling methods.

**Table 8**

Comparison of computational costs.

Small-scale problem						Medium-scale problem						Large-scale problem						
Function #1						Function #4						Function #7						
Stage #1		Stage #5		Stage #9		Stage #1		Stage #5		Stage #9		Stage #1		Stage #5		Stage #9		
Mean	CV	Mean	CV	Mean	CV	Mean	CV	Mean	CV	Mean	CV	Mean	CV	Mean	CV	Mean	CV	
POLY2	0.01	0.20	0.01	0.19	0.01	0.16	0.01	0.18	0.01	0.20	0.01	0.37	0.02	0.19	0.02	0.12	0.02	0.09
rbfG	0.03	0.20	1.17	0.08	3.38	0.05	0.17	0.16	6.71	0.05	20.65	0.02	1.93	0.09	45.43	0.08	132.96	0.08
rbfTP	0.01	0.20	0.01	0.26	0.02	0.28	0.01	0.22	0.02	0.20	0.07	0.17	0.01	0.10	0.15	0.24	0.44	0.14
rbfMQ	0.03	0.23	1.90	0.07	6.14	0.04	0.17	0.14	7.42	0.04	25.09	0.01	2.11	0.35	46.26	0.11	145.48	0.08
SE	0.35	0.06	0.45	0.15	0.52	0.15	0.54	0.17	0.86	0.02	1.23	0.03	0.64	0.28	1.13	0.10	6.03	0.11
RQ	0.37	0.04	0.49	0.11	0.59	0.12	0.59	0.02	1.02	0.01	1.75	0.01	0.68	0.05	1.66	0.06	8.96	0.07
MA3	0.36	0.02	0.64	0.03	0.72	0.04	0.58	0.05	0.95	0.02	1.43	0.01	0.68	0.07	1.67	0.01	10.70	0.03
MA5	0.37	0.04	0.58	0.05	0.59	0.09	0.57	0.05	0.96	0.02	1.46	0.02	0.68	0.06	1.42	0.05	8.53	0.07
BlindKRG	0.19	0.22	0.36	0.31	0.82	0.15	0.87	0.53	6.17	0.42	25.21	0.48	6.09	0.45	151.81	0.04	536.47	0.32
CKL	36.03	0.46	121.73	0.26	60.42	0.69	23.07	0.29	165.36	0.29	301.22	0.34	163.13	0.20	707.76	0.54	2963.6	0.31
Function #2						Function #5						Function #8						
Stage #1		Stage #5		Stage #9		Stage #1		Stage #5		Stage #9		Stage #1		Stage #5		Stage #9		
Mean	CV	Mean	CV	Mean	CV	Mean	CV	Mean	CV	Mean	CV	Mean	CV	Mean	CV	Mean	CV	
POLY2	0.01	0.34	0.01	0.16	0.01	0.16	0.01	0.14	0.01	0.13	0.01	0.04	0.03	0.65	0.02	0.49	0.02	0.29
rbfG	0.03	0.19	1.52	0.09	4.32	0.04	0.61	0.08	24.39	0.02	75.08	0.01	1.75	0.08	42.42	0.04	122.86	0.03
rbfTP	0.00	0.20	0.01	0.30	0.02	0.24	0.01	0.44	0.09	0.13	0.30	0.12	0.01	0.24	0.14	0.14	0.43	0.08
rbfMQ	0.03	0.30	1.91	0.08	6.27	0.05	0.62	0.16	28.40	0.06	91.24	0.08	1.74	0.07	42.52	0.01	136.75	0.01
SE	0.49	0.03	0.61	0.05	0.60	0.05	0.43	0.10	1.22	0.06	2.71	0.03	0.61	0.27	1.47	0.05	8.90	0.03
RQ	0.52	0.02	0.66	0.06	0.71	0.08	0.53	0.08	2.15	0.02	5.04	0.03	0.67	0.05	1.97	0.01	11.63	0.01
MA3	0.53	0.02	0.71	0.02	0.83	0.01	0.47	0.09	1.45	0.04	3.15	0.05	0.66	0.07	1.62	0.03	10.26	0.04
MA5	0.52	0.02	0.72	0.02	0.84	0.02	0.48	0.07	1.47	0.04	3.27	0.03	0.65	0.04	1.65	0.03	10.39	0.03
BlindKRG	0.38	0.33	0.48	0.19	1.30	0.30	2.35	0.58	67.75	0.31	353.04	0.37	3.59	0.46	353.76	0.15	1435.7	0.31
CKL	15.49	0.08	164.37	0.51	192.47	0.38	27.96	0.54	308.83	0.38	765.68	0.30	22.08	0.02	208.08	0.23	1046.0	0.31
Function #3						Function #6						Function #9						
Stage #1		Stage #5		Stage #9		Stage #1		Stage #5		Stage #9		Stage #1		Stage #5		Stage #9		
Mean	CV	Mean	CV	Mean	CV	Mean	CV	Mean	CV	Mean	CV	Mean	CV	Mean	CV	Mean	CV	
POLY2	0.01	0.27	0.01	0.31	0.01	0.79	0.01	0.11	0.01	0.12	0.01	0.25	0.11	1.53	0.26	2.27	0.26	2.32
rbfG	0.05	1.06	1.19	0.11	3.32	0.06	0.16	0.13	6.52	0.05	19.40	0.04	1.76	0.07	42.69	0.03	138.37	0.04
rbfTP	0.01	2.99	0.03	2.40	0.02	0.28	0.00	0.21	0.02	0.23	0.08	0.19	0.01	0.29	0.14	0.17	0.45	0.10
rbfMQ	0.04	0.44	1.89	0.04	6.40	0.03	0.18	0.13	7.60	0.07	25.23	0.03	1.76	0.10	44.00	0.04	142.37	0.03
SE	0.48	0.20	0.51	0.09	0.61	0.12	0.53	0.04	0.88	0.02	1.09	0.12	0.59	0.04	1.74	0.05	8.29	0.09
RQ	0.52	0.03	0.58	0.07	0.70	0.08	0.59	0.02	1.04	0.03	1.46	0.05	0.72	0.02	2.57	0.02	11.79	0.02
MA3	0.48	0.05	0.71	0.03	0.84	0.01	0.59	0.02	0.99	0.02	1.56	0.03	0.69	0.05	2.03	0.05	10.37	0.05
MA5	0.48	0.08	0.68	0.03	0.63	0.11	0.58	0.03	0.87	0.05	1.27	0.05	0.69	0.04	2.08	0.06	11.07	0.06
BlindKRG	0.29	1.08	0.39	0.15	0.81	0.30	0.50	0.20	2.78	0.59	16.54	0.33	0.76	0.04	171.27	0.01	1483.0	0.28
CKL	28.98	0.60	127.06	0.96	184.68	0.66	22.14	0.13	207.12	0.33	442.38	0.51	45.98	0.50	1453.9	0.41	5244.4	0.35

CV: coefficient of variation.

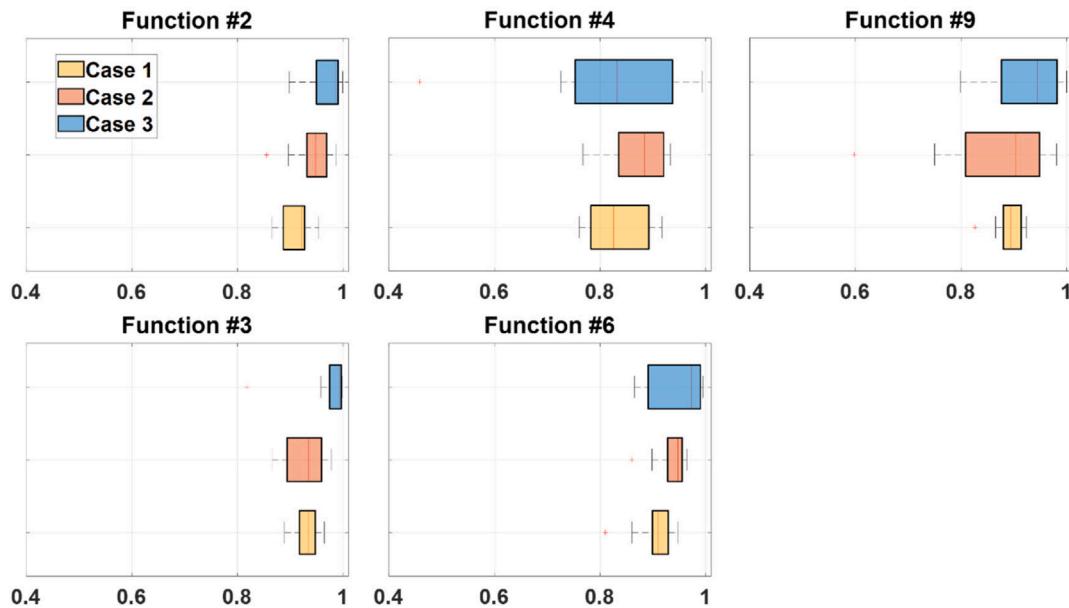


**Fig. 14.** Influence of hyper-parameters of PLHS on prediction performance.

more informative samples based on infill criteria. As a result, it is promising to build an accurate global surrogate modeling automatically and flexibly for a variety of the response-surfaces with minimal user intervention and a tremendous computational efficiency.

## 7. Conclusions

This study proposed the sequential surrogate modeling using the Compositional Kernel Learning (CKL) with the Progressive Latin Hypercube Sampling (PLHS). The proposed method can improve learning capability for response surfaces with the diagnostic measure for the global accuracy of the Gaussian Process



**Fig. 15.** R-squared value between NMSE<sub>Validation</sub> and NMSE<sub>Test</sub> from each case study.

**Table A.1**  
Base-kernels and their hyper-parameters.

Base-kernel	Kernel function	Hyper-parameter
LIN	$\sigma_b^2 + \sigma_v^2 \mathbf{x} \cdot \mathbf{x}'$	$\sigma_b^2$ $\sigma_v^2$
CON	$\sigma^2$	$\sigma$
PER	$\sigma^2 \exp\left(-\frac{2 \sin 2(\pi  \mathbf{x}-\mathbf{x}' /p)}{l^2}\right)$	$\sigma$ $l$ $p$
SE	$\sigma^2 \exp\left(-\frac{(\mathbf{x}-\mathbf{x}')^2}{2l^2}\right)$	$\sigma$ $l$
RQ	$\sigma^2 \left(1 + \frac{(\mathbf{x}-\mathbf{x}')^2}{2\alpha l^2}\right)^{-\alpha}$	$\sigma$ $l$ $\alpha$
ME3	$\sigma^2 \left(1 + \frac{(\mathbf{x}-\mathbf{x}')\sqrt{3}}{l}\right) \exp\left(-\frac{(\mathbf{x}-\mathbf{x}')\sqrt{3}}{l}\right)$	$\sigma$ $l$
ME5	$\sigma^2 \left(1 + \frac{(\mathbf{x}-\mathbf{x}')\sqrt{5}}{l} + \frac{5(\mathbf{x}-\mathbf{x}')^2}{3l^2}\right) \exp\left(-\frac{(\mathbf{x}-\mathbf{x}')\sqrt{5}}{l}\right)$	$\sigma$ $l$

(GP). The proposed method employs the CKL automatically to discover the proper covariance kernel under observed samples. Until the desired accuracy of the GP is achieved, the PLHS is implemented progressively to augment the samples by preserving the desired properties of the distribution (Latin hypercube and space-filling properties). Based on the consecutive sub-samples in the PLHS, the sub-samples except for last one are used as the training samples to construct the GP and the last sub-samples are used as the validation samples to evaluate the global accuracy of the GP.

The numerical experiments show that (1) the CKL is superior to the existing surrogate models with popularity; (2) the sequential SFS with both space-filling criteria and projective properties is preferred with the CKL to generate synergy; and (3) the proposed method provides the diagnostic measure for the global accuracy (NMSE<sub>Validation</sub>). In the proposed method, the diagnostic measure can be utilized to plot the trace against the size of the training samples. Based on the trace-plot of the diagnostic measure, the stopping criteria for further sampling can be assessed. By virtue of the CKL and PLHS, the proposed method can help avoid over- or under-sampling for the global surrogate modeling by achieving

the desired accuracy of the GP. The proposed method allows non-experts to construct the GP for the global surrogate modeling with the minimal user-intervention.

#### CRediT authorship contribution statement

**Seung-Seop Jin:** Conceptualization, Methodology, Software, Writing, Revision, Supervision, Project administration, Funding acquisition.

#### Declaration of competing interest

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

#### Acknowledgment

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2020R1C1C1009236).

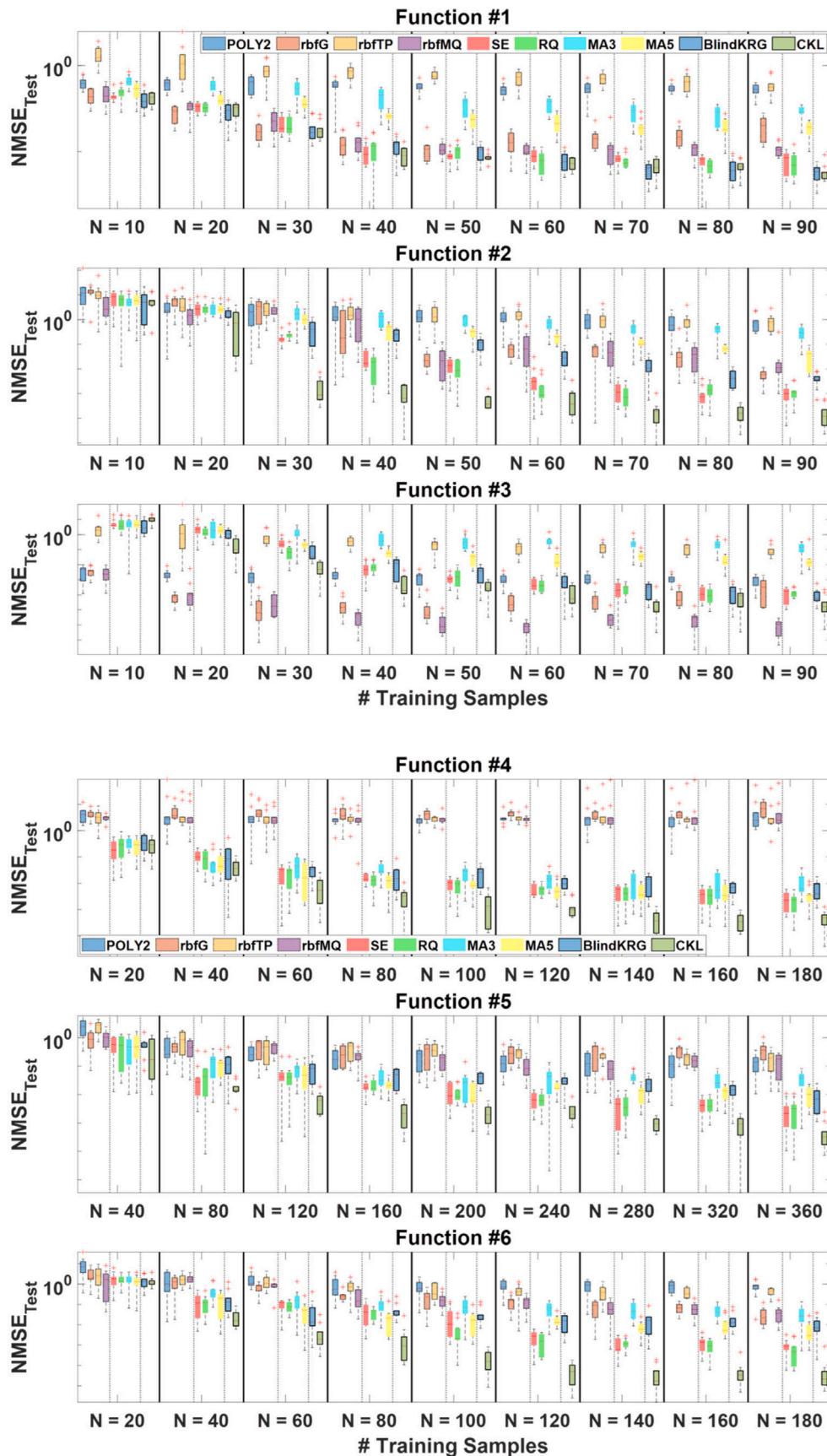


Fig. C.1.

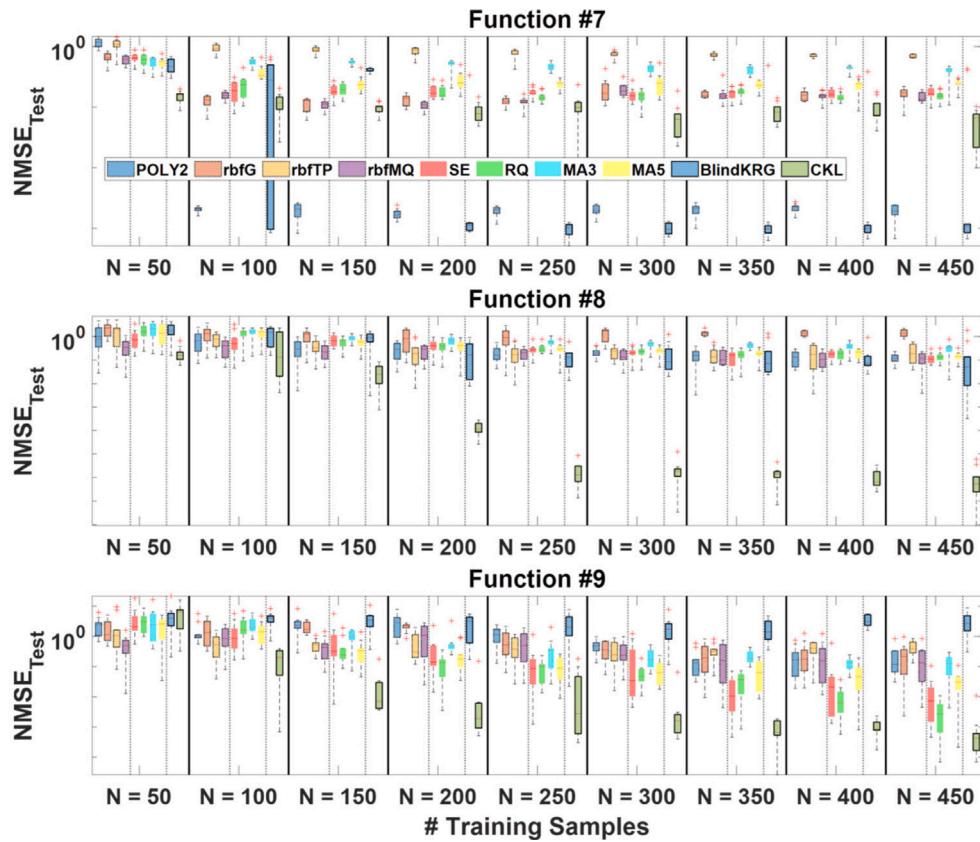


Fig. C.1. (continued).

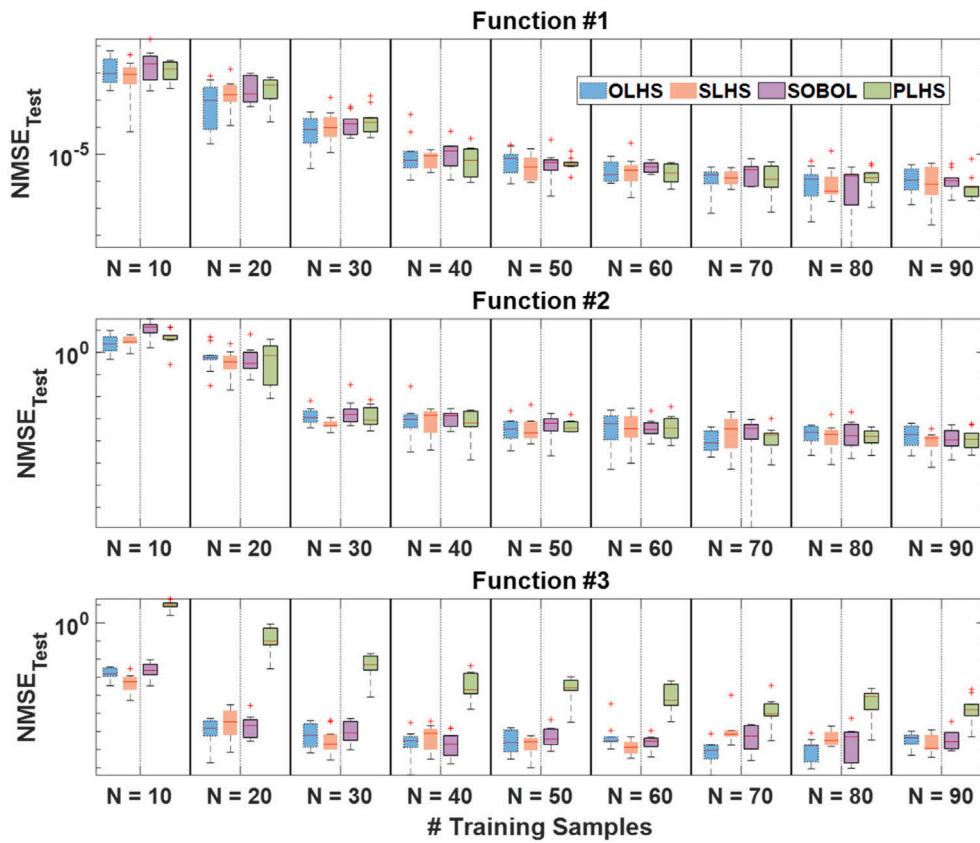


Fig. D.1.

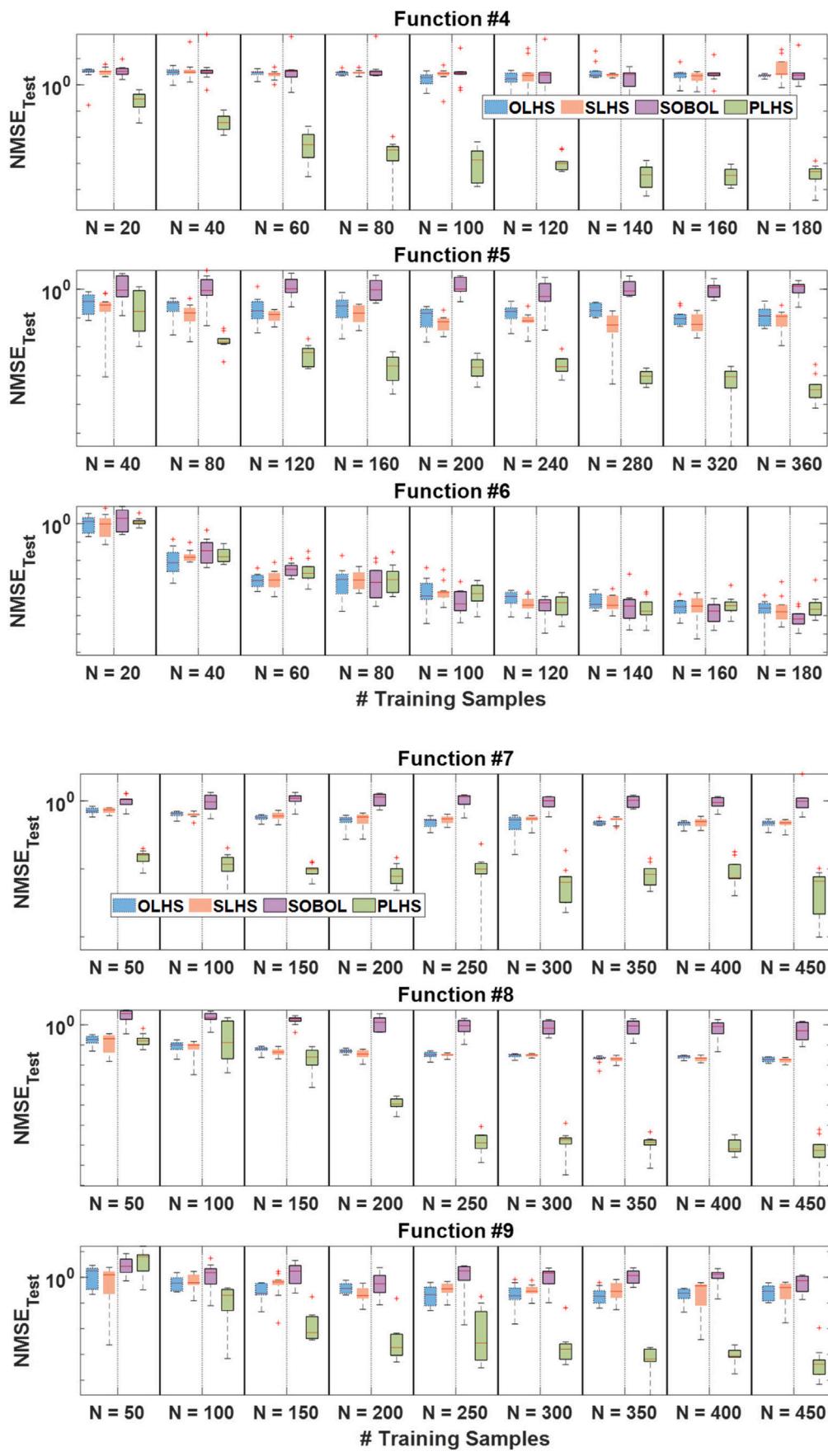


Fig. D.1. (continued).

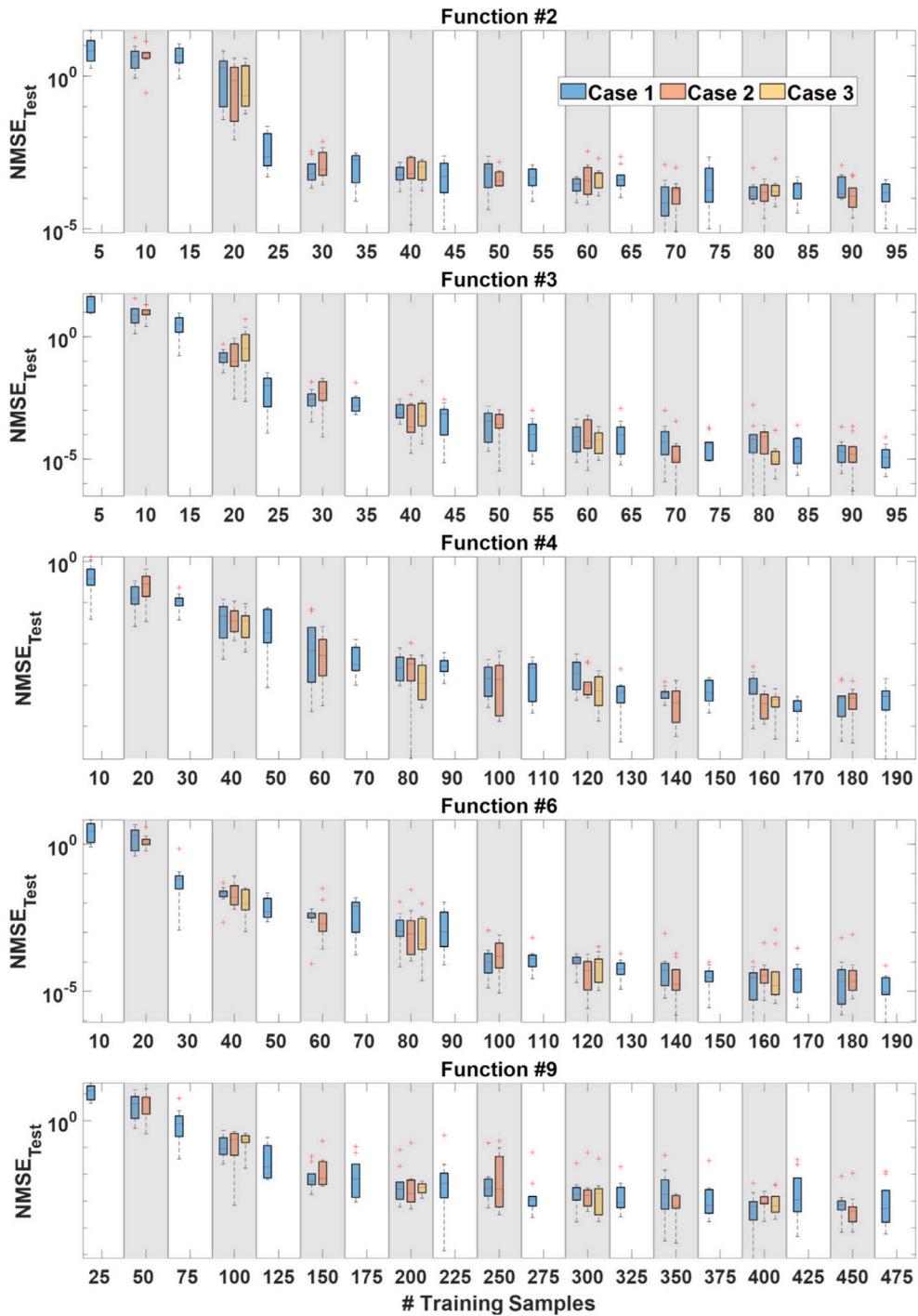


Fig. E.1.

## Appendix A

See Table A.1.

## Appendix B. Test functions

### Function 1 ( $d = 2$ ) [61]

$$f(\mathbf{x}) = 2 + 0.01(x_2 - x_1^2)^2 + (1 - x_1) + 2(2 - x_2)^2 + 7\sin(0.5x_1)\sin(0.7x_1x_2),$$

where  $x_1, x_2 \in [0, 1]$

### Function 2 ( $d = 3$ ) [62]

$$f(\mathbf{x}) = 100 \left[ \exp\left(-\frac{2}{x_1^{1.75}}\right) + \exp\left(-\frac{2}{x_2^{1.5}}\right) + \exp\left(-\frac{2}{x_3^{1.25}}\right) \right], \quad (\text{B.2})$$

where  $x_i \in [0, 1]$  with  $i = 1:3$

### Function 3 ( $d = 2$ ) [63]

$$f(\mathbf{x}) = \left( x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6 \right)^2 + 10 \left( 1 - \frac{1}{8\pi} \right) \cos(x_1) + 10, \quad (\text{B.3})$$

**Table F.1**

Computational costs (unit: second).

**- Small-scale problem**

Function #1	Stage #1		Stage #2		Stage #3		Stage #4		Stage #5		Stage #6		Stage #7		Stage #8		Stage #9	
	Mean	CV																
POLY2	0.01	0.20	0.01	0.13	0.01	0.12	0.01	0.15	0.01	0.19	0.01	0.14	0.01	0.13	0.01	0.11	0.01	0.16
rbfG	0.03	0.20	0.15	0.17	0.29	0.15	0.46	0.11	1.17	0.08	1.62	0.07	2.14	0.09	2.67	0.06	3.38	0.05
rbfTP	0.00	0.20	0.00	0.29	0.00	0.26	0.00	0.28	0.01	0.26	0.01	0.19	0.01	0.23	0.02	0.24	0.02	0.28
rbfMQ	0.03	0.23	0.19	0.15	0.41	0.14	0.63	0.06	1.90	0.07	3.10	0.02	3.77	0.04	5.39	0.03	6.14	0.04
SE	0.35	0.06	0.45	0.05	0.49	0.04	0.42	0.14	0.45	0.15	0.42	0.16	0.48	0.14	0.53	0.15	0.52	0.15
RQ	0.37	0.04	0.53	0.04	0.58	0.06	0.53	0.10	0.49	0.11	0.52	0.06	0.53	0.16	0.58	0.11	0.59	0.12
MA3	0.36	0.02	0.51	0.02	0.53	0.01	0.61	0.01	0.64	0.03	0.67	0.01	0.69	0.05	0.69	0.03	0.72	0.04
MA5	0.37	0.04	0.52	0.04	0.52	0.04	0.59	0.04	0.58	0.05	0.60	0.05	0.63	0.06	0.62	0.14	0.59	0.09
BlindKRG	0.19	0.22	0.23	0.28	0.29	0.23	0.31	0.30	0.36	0.31	0.44	0.28	0.58	0.12	0.68	0.13	0.82	0.15
CKL	36.03	0.46	55.91	0.38	62.08	0.35	122.54	0.24	121.73	0.26	58.36	0.56	88.71	0.58	51.29	0.47	60.42	0.69
Function #2	Mean	CV																
POLY2	0.01	0.34	0.01	0.05	0.01	0.14	0.01	0.09	0.01	0.16	0.01	0.17	0.01	0.11	0.01	0.17	0.01	0.16
rbfG	0.03	0.19	0.18	0.13	0.35	0.15	0.57	0.09	1.52	0.09	2.06	0.08	2.71	0.04	3.58	0.06	4.32	0.04
rbfTP	0.00	0.20	0.00	0.27	0.00	0.23	0.00	0.59	0.01	0.30	0.01	0.26	0.01	0.25	0.02	0.57	0.02	0.24
rbfMQ	0.03	0.30	0.18	0.11	0.39	0.12	0.65	0.10	1.91	0.08	2.89	0.09	3.94	0.03	4.92	0.07	6.27	0.05
SE	0.49	0.03	0.53	0.02	0.55	0.04	0.60	0.02	0.61	0.05	0.58	0.07	0.60	0.09	0.57	0.07	0.60	0.05
RQ	0.52	0.02	0.59	0.02	0.67	0.03	0.67	0.02	0.66	0.06	0.65	0.08	0.65	0.11	0.63	0.09	0.71	0.08
MA3	0.53	0.02	0.58	0.02	0.61	0.03	0.67	0.02	0.71	0.02	0.74	0.02	0.79	0.02	0.80	0.02	0.83	0.01
MA5	0.52	0.02	0.58	0.01	0.60	0.02	0.67	0.03	0.72	0.02	0.75	0.03	0.80	0.03	0.81	0.02	0.84	0.02
BlindKRG	0.38	0.33	0.36	0.29	0.41	0.25	0.45	0.25	0.48	0.19	0.59	0.30	0.80	0.14	0.98	0.23	1.30	0.30
CKL	15.49	0.08	39.21	0.35	59.53	0.26	94.09	0.30	164.37	0.51	142.00	0.42	178.40	0.40	194.70	0.46	192.47	0.38
Function #3	Mean	CV																
POLY2	0.01	0.27	0.01	0.53	0.01	0.18	0.01	0.34	0.01	0.31	0.01	0.22	0.01	0.32	0.01	0.37	0.01	0.79
rbfG	0.05	1.06	0.15	0.16	0.30	0.14	0.47	0.09	1.19	0.11	1.59	0.09	2.08	0.06	2.66	0.06	3.32	0.06
rbfTP	0.01	2.99	0.00	0.32	0.00	0.44	0.00	0.19	0.03	2.40	0.01	0.29	0.01	0.24	0.02	0.23	0.02	0.28
rbfMQ	0.04	0.44	0.19	0.13	0.44	0.16	0.71	0.04	1.89	0.04	2.96	0.09	4.06	0.03	5.01	0.04	6.40	0.03
SE	0.48	0.20	0.53	0.04	0.56	0.09	0.60	0.08	0.51	0.09	0.58	0.07	0.63	0.11	0.62	0.07	0.61	0.12
RQ	0.52	0.03	0.59	0.02	0.68	0.04	0.59	0.08	0.58	0.07	0.60	0.08	0.64	0.12	0.66	0.06	0.70	0.08
MA3	0.48	0.05	0.58	0.02	0.61	0.04	0.68	0.03	0.71	0.03	0.75	0.03	0.79	0.02	0.80	0.01	0.84	0.01
MA5	0.48	0.08	0.58	0.02	0.60	0.02	0.67	0.03	0.68	0.03	0.69	0.05	0.66	0.13	0.66	0.11	0.63	0.11
BlindKRG	0.29	1.08	0.26	0.18	0.38	0.21	0.34	0.24	0.39	0.15	0.54	0.13	0.72	0.13	0.72	0.31	0.81	0.30
CKL	28.98	0.60	40.06	0.02	46.54	0.26	68.80	0.60	127.06	0.96	159.55	0.60	172.71	0.53	95.58	0.69	184.68	0.66

**- Medium-scale problem**

Function #4	Stage #1		Stage #2		Stage #3		Stage #4		Stage #5		Stage #6		Stage #7		Stage #8		Stage #9	
	Mean	CV																
POLY2	0.01	0.18	0.02	0.90	0.01	0.60	0.01	0.80	0.01	0.20	0.01	0.34	0.01	0.29	0.01	0.66	0.01	0.37
rbfG	0.17	0.16	0.63	0.10	2.67	0.05	4.78	0.05	6.71	0.05	9.55	0.03	13.06	0.02	16.59	0.02	20.65	0.02
rbfTP	0.00	0.22	0.00	0.25	0.01	0.25	0.01	0.21	0.02	0.20	0.03	0.18	0.04	0.10	0.06	0.24	0.07	0.17
rbfMQ	0.17	0.14	0.64	0.10	2.64	0.07	5.03	0.05	7.42	0.04	11.28	0.02	15.22	0.01	20.14	0.02	25.09	0.01
SE	0.54	0.17	0.58	0.06	0.65	0.11	0.69	0.06	0.86	0.02	1.05	0.04	1.12	0.05	1.14	0.04	1.23	0.03
RQ	0.59	0.02	0.68	0.02	0.73	0.02	0.82	0.01	1.02	0.01	1.24	0.03	1.36	0.02	1.54	0.01	1.75	0.01
MA3	0.58	0.05	0.65	0.03	0.72	0.03	0.77	0.01	0.95	0.02	1.15	0.03	1.24	0.02	1.33	0.01	1.43	0.01
MA5	0.57	0.05	0.65	0.05	0.72	0.03	0.77	0.01	0.96	0.02	1.15	0.02	1.25	0.02	1.36	0.02	1.46	0.02
BlindKRG	0.87	0.53	1.14	0.41	2.40	0.42	4.64	0.19	6.17	0.42	9.06	0.35	13.11	0.40	17.37	0.60	25.21	0.48
CKL	23.07	0.29	35.96	0.65	91.44	0.32	114.58	0.33	165.36	0.29	232.48	0.61	324.98	0.46	322.32	0.39	301.22	0.34
Function #5	Mean	CV																
POLY2	0.01	0.14	0.01	0.13	0.01	0.09	0.01	0.13	0.01	0.13	0.01	0.07	0.01	0.11	0.01	0.14	0.01	0.04
rbfG	0.61	0.08	4.43	0.05	10.02	0.03	17.15	0.02	24.39	0.02	34.95	0.02	45.73	0.01	59.50	0.01	75.08	0.01
rbfTP	0.00	0.44	0.01	0.23	0.03	0.14	0.06	0.15	0.09	0.13	0.14	0.18	0.19	0.14	0.24	0.12	0.30	0.12
rbfMQ	0.62	0.16	4.51	0.12	10.49	0.06	18.21	0.05	28.40	0.06	40.67	0.05	54.84	0.05	72.71	0.08	91.24	0.08
SE	0.43	0.10	0.52	0.08	0.79	0.03	0.99	0.06	1.22	0.06	1.51	0.09	1.91	0.02	2.34	0.02	2.71	0.03
RQ	0.53	0.08	0.79	0.07	1.20	0.03	1.66	0.02	2.15	0.02	2.72	0.03	3.38	0.03	4.13	0.05	5.04	0.03
MA3	0.47	0.09	0.63	0.05	0.87	0.05	1.13	0.04	1.45	0.04	1.80	0.04	2.27	0.03	2.70	0.03	3.15	0.05
MA5	0.48	0.07	0.65	0.03	0.90	0.04	1.16	0.05	1.47	0.04	1.84	0.03	2.28	0.03	2.77	0.03	3.27	0.03
BlindKRG	2.35	0.58	6.32	0.29	16.45	0.32	34.25	0.33	67.75	0.31	126.47	0.24	191.13	0.22	274.34	0.31	353.04	0.37
CKL	27.96	0.54	94.48	0.20	116.76	0.16	166.84	0.09	308.83	0.38	356.47	0.22	557.44	0.28	575.51	0.47	765.68	0.30
Function #6	Mean	CV																
POLY2	0.01	0.11	0.01	0.11	0.01	0.20	0.01	0.23	0.01	0.12	0.01	0.20	0.01	0.25	0.01	0.23	0.01	0.25
rbfG	0.16	0.13	0.65	0.11	2.64	0.09	4.23	0.05	6.52									

**Table F.1** (continued).**- Large-scale problem**

Function #7	Stage #1		Stage #2		Stage #3		Stage #4		Stage #5		Stage #6		Stage #7		Stage #8		Stage #9	
	Mean	CV																
POLY2	0.02	0.19	0.02	0.12	0.02	0.06	0.02	0.10	0.02	0.12	0.02	0.13	0.02	0.21	0.02	0.34	0.02	0.09
rbfG	1.93	0.09	7.50	0.10	17.29	0.10	30.00	0.09	45.43	0.08	62.89	0.07	82.36	0.08	103.67	0.07	132.96	0.08
rbfTP	0.01	0.10	0.02	0.21	0.05	0.21	0.08	0.07	0.15	0.24	0.20	0.13	0.25	0.08	0.36	0.12	0.44	0.14
rbfMQ	2.11	0.35	7.39	0.11	16.94	0.07	29.85	0.10	46.26	0.11	66.21	0.12	87.84	0.08	114.24	0.07	145.48	0.08
SE	0.64	0.28	0.76	0.09	0.88	0.15	0.96	0.15	1.13	0.10	1.22	0.10	1.53	0.15	5.08	0.09	6.03	0.11
RQ	0.68	0.05	0.95	0.02	1.22	0.04	1.39	0.06	1.66	0.06	1.93	0.05	2.34	0.06	7.18	0.10	8.96	0.07
MA3	0.68	0.07	0.90	0.04	1.18	0.01	1.39	0.01	1.67	0.01	1.94	0.01	2.31	0.03	8.87	0.00	10.70	0.03
MA5	0.68	0.06	0.92	0.04	1.09	0.04	1.20	0.05	1.42	0.05	1.68	0.05	1.93	0.05	7.11	0.07	8.53	0.07
BlindKRG	6.09	0.45	19.26	0.31	42.84	0.06	100.55	0.07	151.81	0.04	226.26	0.07	300.29	0.19	450.30	0.27	536.47	0.32
CKL	163.13	0.20	224.60	0.40	299.01	0.40	433.00	0.38	707.76	0.54	773.62	0.35	935.87	0.45	2825.5	0.31	2963.6	0.31
Function #8	Mean	CV																
POLY2	0.03	0.65	0.02	0.17	0.03	0.53	0.02	0.43	0.02	0.49	0.04	1.68	0.03	0.63	0.03	0.72	0.02	0.29
rbfG	1.75	0.08	7.02	0.08	15.77	0.04	27.81	0.03	42.42	0.04	58.85	0.04	74.52	0.02	97.35	0.04	122.86	0.03
rbfTP	0.01	0.24	0.02	0.22	0.05	0.19	0.09	0.14	0.14	0.22	0.10	0.26	0.17	0.34	0.14	0.43	0.08	
rbfMQ	1.74	0.07	6.80	0.04	15.62	0.01	27.27	0.01	42.52	0.01	61.31	0.01	82.93	0.01	107.94	0.01	136.75	0.01
SE	0.61	0.27	0.82	0.05	1.05	0.05	1.22	0.04	1.47	0.05	1.63	0.05	2.08	0.10	7.42	0.03	8.90	0.03
RQ	0.67	0.05	0.94	0.01	1.30	0.01	1.57	0.01	1.97	0.00	2.41	0.00	2.94	0.00	9.70	0.00	11.63	0.00
MA3	0.66	0.07	0.89	0.03	1.15	0.02	1.33	0.04	1.62	0.03	1.90	0.00	2.27	0.03	8.76	0.00	10.26	0.04
MA5	0.65	0.04	0.89	0.05	1.16	0.01	1.33	0.06	1.65	0.03	1.96	0.00	2.29	0.04	8.90	0.01	10.39	0.03
BlindKRG	3.59	0.46	21.36	0.58	64.14	0.38	161.37	0.29	353.76	0.15	469.12	0.30	648.57	0.43	1152.1	0.28	1435.7	0.31
CKL	22.08	0.02	102.45	0.62	181.50	0.47	331.73	1.19	208.08	0.23	267.58	0.37	386.52	0.41	920.88	0.29	1046.0	0.31
Function #9	Mean	CV																
POLY2	0.11	1.53	0.08	0.49	0.08	0.28	0.15	1.66	0.26	2.27	0.17	1.54	0.19	1.15	0.19	1.74	0.26	2.32
rbfG	1.76	0.07	6.78	0.05	15.61	0.03	27.35	0.03	42.69	0.03	60.94	0.03	82.97	0.02	108.92	0.03	138.37	0.04
rbfTP	0.01	0.29	0.03	0.20	0.05	0.20	0.09	0.20	0.14	0.17	0.22	0.05	0.28	0.09	0.34	0.12	0.45	0.10
rbfMQ	1.76	0.10	7.13	0.04	16.07	0.05	28.21	0.05	44.00	0.04	62.26	0.03	85.04	0.03	111.87	0.04	142.37	0.03
SE	0.59	0.04	0.94	0.05	1.17	0.04	1.39	0.06	1.74	0.05	2.06	0.04	2.53	0.13	6.56	0.07	8.29	0.09
RQ	0.72	0.02	1.22	0.03	1.62	0.02	2.02	0.01	2.57	0.02	3.16	0.01	3.83	0.01	9.52	0.01	11.79	0.02
MA3	0.69	0.05	1.11	0.04	1.38	0.05	1.73	0.04	2.03	0.05	2.55	0.02	3.03	0.04	8.25	0.05	10.37	0.05
MA5	0.69	0.04	1.11	0.05	1.40	0.07	1.71	0.04	2.08	0.06	2.58	0.04	3.10	0.04	8.90	0.05	11.07	0.06
BlindKRG	0.76	0.04	6.49	0.02	26.72	0.01	75.92	0.02	171.27	0.01	347.68	0.01	631.19	0.02	1186.9	0.01	1483.0	0.28
CKL	45.98	0.50	188.83	0.88	754.25	0.72	1236.9	0.42	1453.9	0.41	1634.4	0.40	1493.4	0.44	4448.0	0.23	5244.4	0.35

CV: coefficient of variation.

where  $x_1 \in [-5, 10]$ ,  $x_2 \in [0, 15]$ **Function 4** ( $d = 6$ ) [27]

$$f(\mathbf{x}) = \frac{\left(\frac{12x_2}{x_1+x_2} + 0.74\right)x_6(x_5 + 9)}{x_6(x_5 + 9) + x_3} + \frac{11.35x_3}{x_6(x_5 + 9) + x_3} + \frac{0.74x_3x_6(x_5 + 9)}{(x_6(x_5 + 9) + x_3)x_4}, \quad (\text{B.4})$$

where  $x_1 \in [50, 150]$ ,  $x_2 \in [25, 70]$ ,  $x_3 \in [0.5, 3]$ ,  $x_4 \in [1.2, 2.5]$ ,  $x_5 \in [0.25, 1.2]$ ,  $x_6 \in [50, 300]$ **Function 5** ( $d = 8$ ) [62]

$$f(\mathbf{x}) = 4(x_1 - 2 + 8x_2 - 8x_2^2)^2 + (3 - 4x_2)^2 + 16\sqrt{x_3 + 1}(2x_3 - 1)^2 + \sum_{i=4}^8 i \cdot \ln \left(1 + \sum_{j=3}^i x_j\right), \quad (\text{B.5})$$

where  $x_i \in [0, 1]$  with  $i = 1: 8$ **Function 6** ( $d = 5$ ) [64]

$$f(\mathbf{x}) = 10\sin(\pi x_1 x_2) + 20\left(x_3 - \frac{1}{2}\right)^2 + 10x_4 + 5x_5, \quad (\text{B.6})$$

where  $x_i \in [0, 1]$  with  $i = 1: 5$ **Function 7** ( $d = 10$ ) [65]

$$f(\mathbf{x}) = x_1^2 + x_2^2 + x_1x_2 - 14x_1 - 16x_2 + (x_3 - 10)^2 + 4(x_4 - 5)^2 + (x_5 - 3)^2 + 2(x_6 - 1)^2 + 5x_7^2 + 7(x_8 - 11)^2 + 2(x_9 - 10)^2 + (x_{10} - 7)^2 + 45, \quad (\text{B.7})$$

where  $x_i \in [0, 1]$  with  $i = 1: 8$ **Function 8** ( $d = 10$ ) [66]

$$f(\mathbf{x}) = -0.1 \sum_{i=1}^d \cos(5\pi x_i) - \sum_{i=1}^d x_i^2, \quad (\text{B.8})$$

where  $x_i \in [-1, 1]$  with  $i = 1: 10$ **Function 9** ( $d = 20$ ) [67]

$$\begin{aligned} f(\mathbf{x}) = & \frac{5x_{12}}{1+x_1} + 5(x_4 - x_{20})^2 + x_5 + 40x_{19}^3 - 5x_{19} \\ & + 0.05x_2 + 0.08x_3 \\ & - 0.03x_6 + 0.03x_7 - 0.09x_9 - 0.01x_{10} \\ & - 0.07x_{11} + 0.25x_{13}^3 \\ & - 0.04x_{14} + 0.06x_{15} - 0.01x_{17} - 0.03x_{18}, \end{aligned} \quad (\text{B.9})$$

where  $x_i \in [-0.5, 0.5]$  with  $i = 1: 20$ **Appendix C. Performance evaluation of surrogate modeling**

See Fig. C.1.

**Appendix D. Performance evaluation of sampling method**

See Fig. D.1.

**Appendix E. Case studies with different hyper-parameters of PLHS**

See Fig. E.1.

## Appendix F

See Table F.1.

## References

- [1] N.V. Queipo, et al., Surrogate-based analysis and optimization, *Prog. Aerosp. Sci.* 41 (2005) 1–28.
- [2] M.C. Kennedy, et al., Bayesian calibration of computer models, *J. R. Stat. Soc. Ser. B Stat. Methodol.* 63 (2001) 425–464.
- [3] L.S. Bastos, et al., Diagnostics for Gaussian process emulators, *Technometrics* 51 (2009) 425–438.
- [4] W.L. Oberkampf, et al., Verification and Validation in Scientific Computing, Cambridge University Press, Cambridge, 2010.
- [5] R. Sheikholeslami, et al., Progressive Latin Hypercube Sampling: An efficient approach for robust sampling-based analysis of environmental models, *Environ. Model. Softw.* 93 (2017) 109–126.
- [6] S.-S. Jin, et al., Sequential surrogate modeling for efficient finite element model updating, *Comput. Struct.* 168 (2016) 30–45.
- [7] H. Liu, et al., A survey of adaptive sampling for global metamodeling in support of simulation-based complex engineering design, *Struct. Multidiscip. Optim.* 57 (2018) 393–416.
- [8] M.D. Morris, et al., Bayesian design and analysis of computer experiments: Use of derivatives in surface prediction, *Technometrics* 35 (1993) 243–255.
- [9] D.R. Jones, A taxonomy of global optimization methods based on response surfaces, *J. Global Optim.* 21 (2001) 345–383.
- [10] F. Passos, et al., Radio-frequency inductor synthesis using evolutionary computation and Gaussian-process surrogate modeling, *Appl. Soft Comput.* 60 (2017) 495–507.
- [11] A. O'Hagan, Bayesian Analysis of computer code outputs: A tutorial, *Reliab. Eng. Syst. Saf.* 91 (2006) 1290–1300.
- [12] L.S. Canas, et al., Gaussian Processes with Optimal Kernel Construction for Neuro-Degenerative Clinical Onset Prediction, SPIE, 2018.
- [13] A.A. Kajbaf, et al., Application of surrogate models in estimation of storm surge: A comparative assessment, *Appl. Soft Comput.* (2020) 106184.
- [14] M.R. Kianifar, et al., Performance evaluation of metamodeling methods for engineering problems: towards a practitioner guide, *Struct. Multidiscip. Optim.* 61 (2020) 159–186.
- [15] D.R. Jones, et al., Efficient global optimization of expensive black-box functions, *J. Global Optim.* 13 (1998) 455–492.
- [16] J.M. Henández-Lobato, et al., Predictive entropy search for efficient global optimization of black-box functions, in: Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 1, MIT Press, Montreal, Canada, 2014, pp. 918–926.
- [17] J. Müller, et al., Surrogate optimization of computationally expensive black-box problems with hidden constraints, *INFORMS J. Comput.* 31 (2019) 689–702.
- [18] G.E.P. Box, et al., Empirical Model-Building and Response Surface, John Wiley & Sons, Inc., 1986.
- [19] L. Zhao, et al., Metamodeling method using dynamic kriging for design optimization, *AIAA J.* 49 (2011) 2034–2046.
- [20] L. Van Gelder, et al., Comparative study of metamodeling techniques in building energy simulation: Guidelines for practitioners, *Simul. Model. Pract. Theory* 49 (2014) 245–257.
- [21] M.D. Buhmann, Radial Basis Functions : Theory and Implementations, Cambridge University Press, Cambridge, New York, 2003.
- [22] B.-S. Kim, et al., Comparison study on the accuracy of metamodeling technique for non-convex functions, *J. Mech. Sci. Technol.* 23 (2009) 1175–1181.
- [23] P. Zhu, et al., Metamodel-based lightweight design of an automotive front-body structure using robust optimization, *Proc. Inst. Mech. Eng. D* 223 (2009) 1133–1147.
- [24] S. Ulaganathan, et al., High dimensional kriging metamodeling utilising gradient information, *Appl. Math. Model.* 40 (2016) 5256–5270.
- [25] S. Bhattrai, et al., Efficient uncertainty quantification for a hypersonic trailing-edge flap, using gradient-enhanced kriging, *Aerospace Sci. Technol.* 80 (2018) 261–268.
- [26] C.E. Rasmussen, et al., Gaussian Processes for Machine Learning, MIT Press, Cambridge, Mass, 2006.
- [27] E.N. Ben-Ari, et al., Modeling data from computer experiments: An empirical comparison of Kriging with MARS and projection pursuit regression, *Qual. Eng.* 19 (2007) 327–338.
- [28] I. Couckuyt, et al., Blind Kriging: Implementation and performance analysis, *Adv. Eng. Softw.* 49 (2012) 1–13.
- [29] J. Zhang, et al., Bayesian Model averaging for kriging regression structure selection, *Probab. Eng. Mech.* 56 (2019) 58–70.
- [30] Y. Zhang, et al., A penalized blind likelihood Kriging method for surrogate modeling, *Struct. Multidiscip. Optim.* 61 (2020) 457–474.
- [31] Y. Zhou, et al., An enhanced Kriging surrogate modeling technique for high-dimensional problems, *Mech. Syst. Signal Process.* 140 (2020) 106687.
- [32] H. Liu, et al., Generalized robust Bayesian committee machine for large-scale Gaussian process regression, in: 35th International Conference on Machine Learning, ICML 2018, 2018, pp. 4898–4910.
- [33] D. Rullière, et al., Nested kriging predictions for datasets with a large number of observations, *Statist. Comput.* 28 (2018) 849–867.
- [34] T. Østergård, et al., A comparison of six metamodeling techniques applied to building performance simulations, *Appl. Energy* 211 (2018) 89–103.
- [35] J. Sacks, et al., Design and analysis of computer experiments, *Statist. Sci.* 4 (1989) 409–423.
- [36] M.D. McKay, et al., A comparison of three methods for selecting values of input variables in the analysis of output from a computer code, *Technometrics* 21 (1979) 239–245.
- [37] I.M. Sobol, On the systematic search in a hypercube, *SIAM J. Numer. Anal.* 16 (1979) 790–793.
- [38] M.E. Johnson, et al., Minimax and maximin distance designs, *J. Statist. Plann. Inference* 26 (1990) 131–148.
- [39] E.R.V. Dam, et al., Maximin latin hypercube designs in two dimensions, *Oper. Res.* 55 (2007) 158–169.
- [40] B. Tang, Orthogonal array-based latin hypercubes, *J. Am. Stat. Assoc.* 88 (1993) 1392–1397.
- [41] J.L. Loeppky, et al., Projection array based designs for computer experiments, *J. Statist. Plann. Inference* 142 (2012) 1493–1505.
- [42] T.M. Cioppa, et al., Efficient nearly orthogonal and space-filling Latin hypercubes, *Technometrics* 49 (2007) 45–55.
- [43] K.Q. Ye, et al., Algorithmic construction of optimal symmetric Latin hypercube designs, *J. Statist. Plann. Inference* 90 (2000) 145–159.
- [44] V.R. Joseph, et al., Designing computer experiments with multiple types of factors: The MaxPro approach, *J. Qual. Technol.* (2019) 1–12.
- [45] F. Xiong, et al., Optimizing latin hypercube design for sequential sampling of computer experiments, *Eng. Optim.* 41 (2009) 793–810.
- [46] E.R. van Dam, et al., One-dimensional nested maximin designs, *J. Global Optim.* 46 (2010) 287–306.
- [47] X. Yang, et al., A new class of nested (nearly) orthogonal latin hypercube designs, *Statist. Sinica* 26 (2016) 1249–1267.
- [48] Z. Wu, et al., Efficient space-filling and near-orthogonality sequential Latin hypercube for computer experiments, *Comput. Methods Appl. Math.* 324 (2017) 348–365.
- [49] X. Zhou, et al., Sequential latin hypercube design with both space-filling and projective properties, *Qual. Reliab. Eng. Int.* 35 (2019) 1941–1951.
- [50] C. Schretter, et al., Golden ratio sequences for low-discrepancy sampling, *J. Graph. Tools* 16 (2012) 95–104.
- [51] A. Mehmani, et al., Predictive quantification of surrogate model fidelity based on modal variations with sample density, *Struct. Multidiscip. Optim.* 52 (2015) 353–373.
- [52] D. Duvenaud, et al., Structure discovery in nonparametric regression through compositional kernel search, in: D. Sanjoy, M. David (Eds.), Proceedings of the 30th International Conference on Machine Learning, PMLR, Proceedings of Machine Learning Research, 2013, pp. 1166–1174.
- [53] J.R. Lloyd, et al., Automatic construction and natural-language description of nonparametric regression models, in: Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, AAAI Press, Quebec City, Quebec, Canada, 2014, pp. 1242–1250.
- [54] Y. Hwang, et al., Automatic construction of nonparametric relational regression models for multiple time series, in: B. Maria Florina, Q.W. Kilian (Eds.), Proceedings of the 33rd International Conference on Machine Learning, PMLR, Proceedings of Machine Learning Research, 2016, pp. 3030–3039.
- [55] Z. Ugray, et al., Scatter search and local NLP solvers: A multistart framework for global optimization, *INFORMS J. Comput.* 19 (2007) 313–484.
- [56] S.S. Jin, et al., Self-adaptive sampling for sequential surrogate modeling of time-consuming finite element analysis, *Smart Struct. Syst.* 17 (2016) 611–629.
- [57] R. Mu, et al., Sequential design for response surface model fit in computer experiments using derivative information, *Comm. Statist. Simulation Comput.* 46 (2017) 1148–1155.
- [58] E.J. Ward, A review and comparison of four commonly used Bayesian and maximum likelihood model selection tools, *Ecol. Model.* 211 (2008) 1–10.
- [59] S. Ba, et al., Optimal sliced latin hypercube designs, *Technometrics* 57 (2015) 479–487.
- [60] P.Z.G. Qian, Sliced latin hypercube designs, *J. Am. Stat. Assoc.* 107 (2012) 393–399.
- [61] M.J. Sasena, Flexibility and Efficiency Enhancements for Constrained Global Design Optimization with Kriging Approximations, University of Michigan, Ann Arbor, 2002, p. 221.
- [62] H. Dette, et al., Generalized latin hypercube design for computer experiments, *Technometrics* 52 (2010) 421–429.

- [63] A.I.J. Forrester, et al., Engineering Design Via Surrogate Modelling: A Practical Guide, J. Wiley, Chichester, West Sussex, England; Hoboken, NJ, 2008.
- [64] J.H. Friedman, Multivariate adaptive regression splines, *Ann. Statist.* 19 (1991) 1–67.
- [65] R. Jin, et al., Comparative studies of metamodelling techniques under multiple modelling criteria, *Struct. Multidiscip. Optim.* 23 (2001) 1–13.
- [66] Y.V. Pehlivanoglu, A new particle swarm optimization method enhanced with a periodic mutation strategy and neural networks, *IEEE Trans. Evol. Comput.* 17 (2013) 436–452.
- [67] W.J. Welch, et al., Screening, predicting, and computer experiments, *Technometrics* 34 (1992) 15–25.
- [68] H.S. Hong, et al., Algorithm 823: implementing scrambled digital sequences, *ACM Trans. Math. Software* 29 (2003) 95–109.
- [69] S. Razavi, et al., Numerical assessment of metamodelling strategies in computationally intensive optimization, *Environ. Model. Softw.* 34 (2012) 67–86.
- [70] D.R. Broad, et al., A systematic approach to determining metamodel scope for risk-based optimization and its application to water distribution system design, *Environ. Model. Softw.* 69 (2015) 382–395.
- [71] Y.Q. Gao, Self-adaptive enhanced sampling in the energy and trajectory spaces: Accelerated thermodynamics and kinetic calculations, *J. Chem. Phys.* 128 (2008).
- [72] P. Rakshit, et al., Realization of learning induced self-adaptive sampling in noisy optimization, *Appl. Soft Comput.* 69 (2018) 288–315.
- [73] K. Redouane, et al., Adaptive surrogate modeling with evolutionary algorithm for well placement optimization in fractured reservoirs, *Appl. Soft Comput.* 80 (2019) 177–191.