applied sciences

MDPI

Article

KNN-Based Maximization of Weighted Expected Prediction Error for Adaptive Kriging Modeling

Jingfang Shen [1], Yu Xia [1], Yaohui Li [2,*], Wenwei Liu [3,*] and Zebin Zhang [4]

[1] College of Informatics, Huazhong Agricultural University, Wuhan 430070, China; sjf_712@mail.hzau.edu.cn (J.S.); itsmyworld@webmail.hzau.edu.cn (Y.X.)
[2] College of Mechanical and Electrical Engineering, Xuchang University, Xuchang 461000, China
[3] Guangdong Provincial Key Laboratory of Electronic Information Products Reliability Technology, Guangzhou 511370, China
[4] School of Mechanical and Power Engineering, Zhengzhou University, Kexue Avenue, Zhengzhou 450001, China; zebin.zhang@zzu.edu.cn
* Correspondence: lyh@xcu.edu.cn (Y.L.); wenwei2009@163.com (W.L.)

Abstract

The adaptive Kriging method is widely used in the engineering design for complex black-box problems, yet its accuracy is limited by imbalanced exploitation–exploration. This paper proposes a KNN-based maximization of the weighted expected prediction error (KMWEPE) method to address this challenge. For each iteration, the most sensitive region is identified by the leave-one-out cross-validation error (LOOCVE) and the distance between sample points. Two different sets of candidate points are generated, respectively, in the most sensitive region and the design space, in order to dynamically balance the local exploitation and global exploration. Then, the bias–variance decomposition method is used to convert the expected prediction error of each candidate point into the sum of the bias and the Kriging prediction variance. And the bias is replaced by the weighted sum of the LOOCVE of the *K*-nearest neighbors sample points based on KNN. Furthermore, the arithmetic sum of the bias and the Kriging prediction variance above is used to construct a new function. Finally, the candidate with the maximum weighted expected prediction error is selected as the new sample point for the next iteration. Six benchmark test functions, two publicly available datasets, and two engineering examples are tested to demonstrate the effectiveness of the proposed KMWEPE method in improving the model accuracy. The test results show that compared to the LHD and MEPE methods, the RMSE mean and standard deviation of the KMWEPE method decreased by an average of 31.6% and 28.8%, respectively.

Keywords: adaptive Kriging method; *K*-nearest neighbors; bias–variance decomposition; local exploitation; global exploration

check for updates

Academic Editor: Francesca Pelosi

Received: 11 March 2025
Revised: 12 April 2025
Accepted: 16 April 2025
Published: 15 December 2025

Citation: Shen, J.; Xia, Y.; Li, Y.; Liu, W.; Zhang, Z. KNN-Based Maximization of Weighted Expected Prediction Error for Adaptive Kriging Modeling. *Appl. Sci.* **2025**, *15*, 13149. https://doi.org/10.3390/app152413149

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

In recent years, surrogate models [1,2] (also known as meta-models or response surface models) have gained increasing popularity in solving black-box problems due to their convenience and efficiency. Black-box problems [3,4] refer to a class of problems where the explicit expression of the objective function is unavailable. Surrogate models, however, can simulate and approximate the true objective function using known data, thereby facilitating the solution of black-box problems. Common surrogate models, such as Kriging [5–8], radial basis functions [9,10], neural networks [11,12], support vector machines [13,14], and

polynomial chaos expansion [15,16], have been extensively employed in engineering design problems. Among these, the Kriging model is widely used due to its high computational efficiency, suitability for low-dimensional problems, and ability to approximate highly nonlinear models.

The initial Design of Experiments (DoE) in the construction of a Kriging model acquires samples for modeling. The initial DoE involves collecting all the samples needed for modeling in a single instance, such as Latin Hypercube Sampling (LHS) [17,18], Optimal Latin Hypercube Sampling (OLHS) [19,20], Monte Carlo Random Sampling [21,22], etc. Since the true objective function is unknown, there is no guarantee that all the sample points generated by the initial DoE are of high quality for high-precision modeling [23]. Furthermore, a greater number of sample points means higher computational costs, so it is necessary to reduce the number of samples as much as possible in order to minimize the costs. Consequently, the key problem of modeling is to construct surrogate models with the highest possible accuracy at the lowest possible computational cost, which necessitates that the sample points used for modeling are of a higher quality, i.e., the sample points contain richer information [24–27]. To address this challenge, adaptive Kriging methodologies have been developed to iteratively refine sampling distributions through intelligent infill criteria [28].

In the field of complex engineering system design, Kriging-based engineering design methods have become a core technology for handling black-box problems [26]. However, as the complexity of various engineering fields increasingly demands a higher model accuracy, the existing methods have revealed significant shortcomings in model fitting, spatial correlation, and candidate set generation. This paper addresses three key issues in Kriging modeling and proposes solutions.

First, the current adaptive sampling methods based on Kriging overly rely on variance criteria, which limits the exploration capability of the parameter space and significantly affects the global prediction accuracy of the surrogate model. Forrester et al. [29] found in multi-peak function modeling experiments that if Kriging variance is solely used as the sampling criterion, over 90% of new samples would densely cluster around the existing samples rather than in regions of high nonlinearity or high gradient changes in the true response surface. Similarly, Schöbi et al. [30] found in knee cartilage constitutive model calibration that variance-guided sampling reduced the prediction interval coverage of a fiber direction modulus from 95% to 67%, indicating that the model misjudged the physical mechanisms in unsampled regions. While bootstrap resampling methods [31] can improve the variance estimation robustness through ensemble Kriging models, they require repeated model fitting (typically hundreds of iterations) and still fail to address the fundamental bias-variance decoupling issue. For instance, in [31]'s geotechnical case study, bootstrap-enhanced variance estimates showed limited improvement in the discontinuous regions despite a significant computational overhead.

The deeper contradiction stems from the mismatch between the variance criterion and the modeling objective: Kriging variance is essentially a geometric measure of interpolation uncertainty in blank regions, while the modeling task requires statistical control of the global prediction error. For example, in climate model simplification surrogate modeling, Lataniotis et al. [32] pointed out that variance-guided sampling would cause the prediction confidence interval of tropical cyclone generation thresholds to miss the key phase transition points because it did not consider the nonlinear sensitivity differences of the model near the threshold. In this case, a hybrid sampling criterion based on prediction error gradients can more effectively identify model structural defect regions [33]. To address this issue, this paper integrates prediction bias as a compensation term, introducing a bias-variance synergy mechanism, considering both the uncertainty represented by variance and the

prediction bias caused by the model's inaccuracy during the adaptive sampling process of the Kriging model.

Secondly, the traditional LOOCV method overlooks the spatial correlation of sample points. How to quantify the prediction bias of the model is one of the key issues in improving the accuracy of adaptive sampling. Leave-one-out cross-validation (LOOCV) is a method for evaluating the model bias. Although LOOCV can reflect the stability of the model in single-point prediction, its core defect lies in ignoring the synergistic effect of spatial correlation on the prediction bias. The existence of spatial autocorrelation directly violates the independence assumption of LOOCV. Le et al. [34] pointed out that when data exhibit spatial clustering or gradient distribution, LOOCV systematically overestimates the model prediction performance due to the spatial dependence between the validation set and the training set, which allows the model to "anticipate" the characteristics of the validation samples, thereby weakening the objectivity of bias assessment. Similarly, Juergen and Marcelo [35] found in tropical ecosystem data research that traditional LOOCV could introduce an error estimation bias of up to 28% for rare species distribution models, while introducing spatial block partitioning in cross-validation reduced the bias to below 5%, further verifying the interference of spatial association with the reliability of LOOCV. Maximum likelihood estimation (MLE) methods [36] attempt to address the spatial dependence through explicit covariance modeling, but their effectiveness heavily relies on assumptions of data distribution (e.g., Gaussianity). As showed in [36], MLE-based spatial models exhibited significant performance degradation when handling non-stationary cloud-cover patterns. This highlights the challenge of balancing theoretical rigor with adaptability in spatial bias quantification.

The deeper reason lies in the fundamental conflict between the statistical basis of LOOCV and the essential characteristics of spatial data. Zhang et al. [37] theoretically proved that the independent and identically distributed assumption of LOOCV is incompatible with the spatial dependence network of geographical data—after removing a sampling point, its unsampled spatial neighborhood still influences the prediction residual of the current point through spatial autocorrelation mechanisms such as Moran's I. For example, Hanna and Edzer [38] demonstrated that when the spatial autocorrelation coefficient exceeds 0.6, LOOCV overestimates the coefficient of determination (R^2) of the regression model by more than 0.15, and the error distribution shows significant spatial aggregation. This conclusion is consistent with the empirical research of Hijmans [39] in the field of ecology: in species distribution modeling, LOOCV ignoring spatial dependence mistakenly identifies overfitting as model robustness. To address this defect, this paper considers the spatial neighborhood set of sample points and proposes a spatially corrected weighted bias estimator based on KNN by integrating LOOCV and the KNN algorithm.

Finally, the static candidate set strategy significantly constrains the model convergence efficiency. The inherent conflict between its fixed parameter configuration and dynamic modeling requirements has been widely demonstrated across multidisciplinary modeling studies. The conventional approaches typically predefine the parameter distributions or sampling densities in candidate sets (e.g., uniform grids, fixed confidence intervals), resulting in the model's inability to capture the dynamic evolution of sensitive regions during later iterations. Kennedy [40] demonstrated that fixed candidate sets generate approximately 35% invalid sampling points during the model calibration stages—these parameter combinations are filtered out via computationally intensive high-fidelity model evaluation processes due to their deviation from the high-probability parameter regions.

This phenomenon fundamentally stems from the phase-dependent characteristics of modeling processes, which demand dynamic adaptability in candidate sets: the early stages require the comprehensive coverage of the global possibilities in the parameter space, while

the later stages necessitate focused fine-tuning in the critical response regions. Nevertheless, static strategies fail to satisfy such progressive modeling demands. While Bayesian optimization methods [41] dynamically adapt the sampling regions through posterior updates, they require computationally expensive posterior recalculations, making them impractical for high-dimensional or time-sensitive applications. To address this challenge, this paper proposes a dual candidate set dynamic balancing strategy that intelligently explores the design space to capture the most sensitive regions.

Probabilistic machine learning methods—ranging from simple bootstrapping to maximum likelihood estimation—have demonstrated significant value in enhancing the interpretability across disciplines [31,36,41]. While these probabilistic methods enhance the interpretability, their computational and distributional constraints limit the effectiveness in adaptive sampling. To overcome these limitations while addressing the above three issues, this paper proposes a Kriging modeling adaptive sampling method based on KNN through maximizing the weighted expected prediction error (KMWEPE), with the following contributions:

(1) This paper introduces a bias–variance decomposition strategy, decomposing the expected prediction error of candidate points into the sum of bias and Kriging variance, considering both the prediction bias caused by the model's inaccuracy and the uncertainty represented by the variance, thereby solving the problem of limited spatial exploration capability due to the over-reliance on the variance in the sampling methods. At the same time, the KNN algorithm based on the Euclidean distance between sample points is introduced, and *K*-nearest neighbor samples are determined based on the spatial correlation. The weighted sum of LOOCVE of *K*-nearest neighbor samples is used to quantify the bias, reducing the error caused by traditional LOOCV methods ignoring the spatial correlation.

(2) A dual candidate set dynamic balance strategy is proposed to achieve a balance between global exploration and local exploitation. This paper defines the local most sensitive region with the highest uncertainty based on LOOCVE and the distance between the sample points, and dynamically updates the local most sensitive region in each iteration based on prior information. Two sets of candidate points are generated in this region and the global design space, respectively, improving the convergence efficiency of modeling.

The remainder of this paper is structured as follows. Section 2 provides a brief overview of Kriging and the general process of Kriging modeling and adaptive sampling, and introduces the basic content of the KNN algorithm and bias–variance decomposition. In Section 3, the KMWEPE method proposed in this paper is introduced, and the flowchart and specific steps of the method for modeling are provided. Section 4 demonstrates the effectiveness of the method with six benchmark test functions and two engineering application examples. Finally, the main conclusions of this work are given in Section 5.

2. Technical Background

2.1. Kriging

The Kriging model was proposed by the South African geologist, D.G. Krige, in 1951 [42] for predicting the mineral resource content of unknown areas. It can now be used for engineering design and optimization in various fields.

The Kriging model consists of a regression trend model with a stochastic process and predicts the function value of unknown data through a weighted approximation of the known data [43]. The formulation can be expressed as follows:

$$y(x) = f(x)^T \beta + z(x), \tag{1}$$

where $f(x) = [f_1(x), f_2(x), \ldots, f_p(x)]^T$ is a regression vector composed of p selected regression functions, $\beta = [\beta_1, \beta_2, \ldots, \beta_p]^T$ is the parameter vector of the regression function to be estimated, and $z(x)$ denotes a stochastic process with mean zero and a covariance of:

$$\mathrm{cov}[Z(x_i, x_j)] = \sigma^2 R(\theta; x_i, x_j), \tag{2}$$

where σ^2 is the process variance to be estimated, $R(\theta; x_i, x_j)$ denotes the correlation function between the sample points x_i and x_j, θ is the hyperparameter vector to be estimated, and the matrix R denotes the correlations between all the sample points. The correlation function used in this paper is the Gaussian correlation function, and it is expressed as follows:

$$R(\theta; x_i, x_j) = exp\left[\sum_{k=1}^{n} -\theta_k (x_k^i - x_k^j)^2\right]. \tag{3}$$

As mentioned above, the parameters to be estimated in the Kriging model are as follows: β, σ^2, and θ. Given a sample set of m design sites $X = [x_1, x_2, \ldots, x_m]^T$ and the response set $Y = [y_1, y_2, \ldots, y_m]^T$, the essence of Kriging is to solve for unknown parameters based on the input data X and output data Y by using generalized least squares or maximum likelihood estimation. Under the assumption of an unbiased estimation (i.e., the value of the stochastic process is zero), there is $F\beta \approx Y$, where $F = [f(x_1), f(x_2), \ldots, f(x_m)]^T$. The estimated values of β and σ^2 can be obtained by generalized least squares and maximum likelihood estimation as:

$$\hat{\beta} = \left(F^T R^{-1} F\right)^{-1} F^T R^{-1} Y, \tag{4}$$

$$\hat{\sigma}^2 = \frac{1}{m}(Y - F\hat{\beta})^T R^{-1} (Y - F\hat{\beta}). \tag{5}$$

Based on Equations (4) and (5), the estimated value of the hyperparameter θ can be obtained via maximum likelihood estimation as:

$$\theta = argmax\left(-\frac{m}{2} ln\hat{\sigma}^2 - \frac{1}{2} ln|R|\right). \tag{6}$$

After solving the hyperparameters of the Kriging model, the prediction value and prediction variance of any unknown point x can be obtained:

$$\hat{y} = f(x)^T \hat{\beta} + r(x)^T \gamma, \tag{7}$$

$$s^2 = \hat{\sigma}^2 \left[1 - r^T R^{-1} r + \frac{(1 - 1^T R^{-1} r)^2}{1^T R^{-1} 1}\right], \tag{8}$$

where the correlation vector $r(x) = [R(x, x_1), R(x, x_2), \ldots, R(x, x_m)]$ denotes the correlation of the unknown point x with all the other sample points, and 1 denotes the m-dimension vector of ones. Note that the Kriging model not only predicts the function value of the unknown point, but also obtains the predicted variance, which is one of the characteristics of Kriging.

2.2. Adaptive Kriging Method

Since the true objective function is unknown, there is no guarantee that the initial samples obtained in the initial DoE are of a high quality for high-precision modeling, so it is inadvisable to obtain sufficient initial samples in the initial DoE [23]. Therefore, the adaptive sampling method is required to add informative sample points to improve the model accuracy. The general process of the adaptive Kriging method is as follows:

- Input data are obtained via the initial DoE.

- Output data are obtained via the expensive function evaluation based on the input data.
- Based on the input and output data, the Kriging model is obtained through the Kriging interpolation.
- New sample points are filtered based on the set-filling criterion.
- Expensive function evaluations are performed on the new sample points, and the Kriging model is updated.
- Determine whether the Kriging model satisfies the stopping criterion, and if so, the iteration ends; if not, continue to filter samples based on the filling criterion and update the Kriging model.

As shown in Figure 1, the process outside the dashed box represents the modeling process of the Kriging model, while the dashed box indicates the adaptive sampling process. It can be observed that the primary distinction between the different adaptive sampling methods lies in the filling criteria, which also play a decisive role in improving the model accuracy.

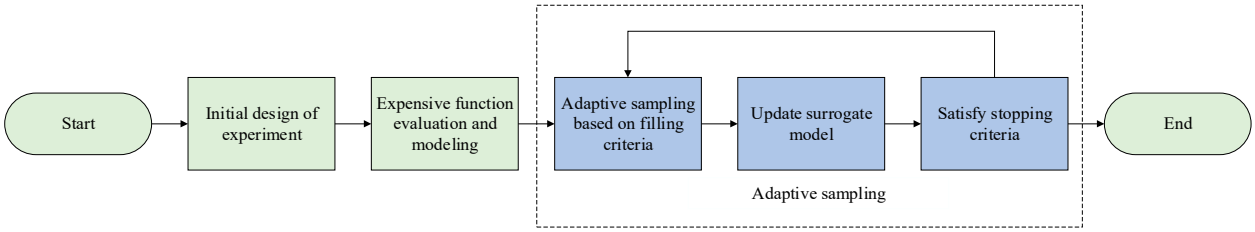

Figure 1. Adaptive sampling process for Kriging modeling.

2.3. K-Nearest Neighbors (KNN) Algorithm

According to whether the training dataset is labeled, machine learning algorithms can be divided into unsupervised learning algorithms and supervised learning algorithms [44]. Unsupervised learning algorithms receive unlabeled data with the goal of discovering hidden structures or patterns within the data, which are used for clustering and dimensionality reduction. In supervised learning algorithms, each sample in the training dataset has a corresponding label. The algorithms predict the labels of the unknown input data based on the labels of the known data, which are usually used for classification and regression. The *K*-Nearest Neighbors (KNN) algorithm is one type of supervised learning algorithm.

The basic idea of KNN is to make predictions based on the correlation between data points. The correlation between data points is quantified through some appropriate measure, and then data points with a higher correlation are considered to have similar characteristics. Therefore, KNN can be used to predict the category or value of the unknown data points.

The basic steps of KNN are as follows:

- Determination of correlation: For each unknown data point, the correlation between that point and all data points in the training set is calculated based on a selected metric. For example, one of the most commonly used metrics is the Euclidean distance. Calculate the Euclidean distance between the unknown data point and all data points, and a closer distance between the two points generally means a higher correlation [45].
- Selection of the *K*-value: A suitable *K*-value indicates the number of nearest neighbor data points to be considered in the prediction. The *K*-value is selected manually, so it will affect the performance of the algorithm to some extent.
- Prediction: For each unknown data point, predict its label based on the labels of the *K*-nearest neighbor data points of the unknown data point. For classification problems, the categories to which the *K*-nearest neighbor data points belong are identified. And

then the category with the most data points among these categories is selected as the category of the unknown data point, while for regression problems, the output value of the unknown data point is calculated via an arithmetic or weighted average of the values of the K-nearest neighbor data points [46].

KNN requires no training process and is insensitive to outliers. However, it also has a high computational complexity and is sensitive to the parameter K. In practical applications, suitable K-values and metrics can be selected according to the characteristics of specific problems to obtain better prediction performance.

2.4. Bias–Variance Decomposition

Bias–variance decomposition is an important method for explaining the model performance [47]. The bias–variance decomposition is used to analyze whether the model error is mainly due to high bias caused by underfitting on the dataset, or high variance caused by overfitting on the current dataset but underfitting on other datasets. Therefore, this method enables one to optimize the model in a more targeted way and improve the model performance.

Let $\hat{f}(x)$ denote the predicted value of the unknown point x, $f(x)$ denote the true value of point x, and y denote the response value of point x.

$$y = f(x) + \varepsilon, \tag{9}$$

where ε is the loss caused by the sample distribution and noise, and ε follows the distribution $N(0, \sigma_\varepsilon^2)$. Let the loss function be denoted as:

$$L(x) = \left(y - \hat{f}(x)\right)^2, \tag{10}$$

then,

$$\begin{aligned} E(L(x)) &= E\left[\left(y - \hat{f}(x)\right)^2\right] = E\left[\left(f(x) + \varepsilon - \hat{f}(x)\right)^2\right] \\ &= E\left[\left(f(x) - \hat{f}(x)\right)^2 + 2\varepsilon\left(f(x) - \hat{f}(x)\right) + \varepsilon^2\right] \\ &= E\left[\left(f(x) - \hat{f}(x)\right)^2\right] + 2E\left[\varepsilon\left(f(x) - \hat{f}(x)\right)\right] + E(\varepsilon)^2, \end{aligned} \tag{11}$$

where,

$$E\left[\varepsilon\left(f(x) - \hat{f}(x)\right)\right] = E[\varepsilon]E\left[\left(f(x) - \hat{f}(x)\right)\right]. \tag{12}$$

The mean value of the loss ε is zero, i.e., $E[\varepsilon]$ is zero, so the value of Equation (12) is zero. The loss is caused by the sample distribution and noise, which cannot be reduced artificially, so the item, $E(\varepsilon)^2$, is ignored and not considered in this paper. The prediction values of point x on the Kriging model established on different datasets are different, so let $E\left(\hat{f}(x)\right)$ denote the expectation of the prediction values of point x on different Kriging models. Then:

$$\begin{aligned} E(L(x)) &= E\left[\left(f(x) - \hat{f}(x)\right)^2\right] = E\left[\left(f(x) - E\left(\hat{f}(x)\right) + E\left(\hat{f}(x)\right) - \hat{f}(x)\right)^2\right] \\ &= E\left[\left(f(x) - E\left(\hat{f}(x)\right)\right)^2\right] + 2E\left[\left(f(x) - E\left(\hat{f}(x)\right)\right)\left(E\left(\hat{f}(x)\right) - \hat{f}(x)\right)\right] \\ &\quad + E\left[\left(E\left(\hat{f}(x)\right) - \hat{f}(x)\right)^2\right], \end{aligned} \tag{13}$$

where,

$$E\left[\left(f(x)-E\left(\hat{f}(x)\right)\right)\left(E\left(\hat{f}(x)\right)-\hat{f}(x)\right)\right]=E\left[f(x)-E\left(\hat{f}(x)\right)\right]E\left[\left(E\left(\hat{f}(x)\right)-\hat{f}(x)\right)\right]$$
$$=E\left[f(x)-E\left(\hat{f}(x)\right)\right]\left(E\left(\hat{f}(x)\right)-E\left(\hat{f}(x)\right)\right)=0, \quad (14)$$

then,

$$E(L(x))=E\left[\left(f(x)-E\left(\hat{f}(x)\right)\right)^2\right]+E\left[\left(E\left(\hat{f}(x)\right)-\hat{f}(x)\right)^2\right]. \quad (15)$$

Therefore, the loss function at any point can be transformed into the two components of Equation (15) through bias–variance decomposition. The first term is the bias, which refers to the difference between the true value and the expectation of prediction value on different models, measuring the model fitting ability and quantifying the model accuracy. The latter term is the variance, which refers to the difference between the expectation of prediction value on different models and the prediction value on the current model, reflecting the model fluctuation and quantifying the model stability. Equation (15) can be simplified as:

$$E(L(x))=e^2(x)+s^2(x), \quad (16)$$

where $e(x)$ denotes the bias of point x under the current model, and $s^2(x)$ denotes the variance of point x.

3. The Proposed KMWEPE Method

This paper proposes a novel adaptive sampling strategy, namely KMWEPE. In this section, the specific details and flowchart of the KMWEPE method will be provided, including the criterion for balancing local exploitation with global exploration and the method for calculating the weighted expected prediction error.

3.1. Criterion for Balancing Local Exploitation with Global Exploration

In the adaptive modeling process, the static candidate set significantly impedes the convergence efficiency of the model. Consequently, it is imperative to dynamically update the candidate set based on the information obtained during the model iterations. At the same time, it is essential to consider how to achieve the dynamic balance between local exploitation and global exploration. For example, Zhao et al. [48] decided whether subsequent sampling favors local exploitation or global exploration based on the comparison of the cross-validation error between the current iteration and the last iteration.

Local exploitation refers to searching for some interesting regions (such as areas with large errors) in the design space and the sampling within these local regions to obtain informative sample points. For example, Liu et al. [49] divided the design space into several Voronoi cells and quantified the uncertainty of each cell via the LOOCVE of the sample point within the cell. And then, the sample with the highest uncertainty in the cell is filtered. The advantage of local exploitation is that the search space is limited to some local regions where the Kriging model differs significantly from the real model, and the sampling in these local regions can accelerate the construction of high-precision models.

Global exploration refers to sampling in the design space to avoid certain promising regions not being explored. The high accuracy for modeling refers to the global accuracy. Indeed, local exploitation can be considered as another form of global exploration, which involves searching for points with global characteristics within local regions. Both aim at improving the global accuracy of the model.

A novel criterion for balancing local exploitation with global exploration is proposed in this paper. For each iteration, two sets of candidate points are generated in the most sensitive region and the design space, respectively. Then, the WEPEs of all candidate points are calculated, and the point with the largest WEPE is selected from each set of candidate

points. Finally, the WEPEs of the two points are compared, and the point with the larger WEPE is selected as the new sample point (the calculation method of WEPE is shown in Section 3.2). If the point comes from the candidate points in the most sensitive region, the current iteration favors local exploitation; if the point comes from the candidate points in the design space, the current iteration favors global exploration.

For local exploitation, the prior information from previous iterations is used to direct the current iteration. In each iteration, the LOOCVE for each known sample point is calculated. A larger error indicates higher uncertainty in the area around the sample point. Therefore, the sample point with the largest error is selected as the center point x_{center}, and the distance between x_{center} and the nearest sample point to x_{center} is used as the radius r. The hypercube formed by the center point x_{center} and the radius r is the most sensitive region $D_{sensitive}$ (as shown in Equation (19)), implying high uncertainty in this region. In order to identify the region with the highest uncertainty in $D_{sensitive}$, a complete exploration is required in the entire region. Therefore, LHS is used to generate a set of local candidate points S_{local} within $D_{sensitive}$ to ensure that the sample points in S_{local} are uniformly distributed within $D_{sensitive}$. The WEPE for all points in S_{local} is calculated, and the point with the largest WEPE is selected as the iterative point x_{local} for local exploitation.

$$x_{center} = arg \max_{x_i \in X} CV(x_i). \tag{17}$$

$$r = arg \min_{\substack{x_i \in X \\ x_i \neq x_{center}}} d(x_i, x_{center}). \tag{18}$$

$$D_{sensitive} = \{x \in R^n : d(x, x_{center}) \leq r\}. \tag{19}$$

$$x_{local} = arg \max_{x \in S_{local}} WEPE(x). \tag{20}$$

The uniformly distributed initial sample points obtained via LHS ensure that all the areas in the design space are sampled, and the screening of local candidate points $D_{sensitive}$ ensures the exploration of regions with high uncertainty. Therefore, for global exploration, Monte Carlo Random Sampling is used to generate a second set of candidate points S_{global} to ensure that each sample point in the design space has an equal chance of being selected, thus reducing the sampling bias. Similarly, the WEPE of all points in S_{global} is calculated and the point with the largest WEPE is selected as the iterative point S_{global} for global exploration.

$$x_{global} = arg \max_{x \in S_{global}} WEPE(x). \tag{21}$$

Compare the WEPE of x_{local} and x_{global}, and select the point with the larger value as the new sample point x_{new} for the current iteration, i.e.,

$$x_{new} = \begin{cases} x_{local}, & WEPE(x_{local}) \geq WEPE(x_{global}) \\ x_{global}, & WEPE(x_{local}) < WEPE(x_{global}) \end{cases}. \tag{22}$$

If $WEPE(x_{local}) \geq WEPE(x_{global})$, it indicates that the region around x_{local} in $D_{sensitive}$ has higher uncertainty, so then, the current iteration favors local exploration; if $WEPE(x_{local}) < WEPE(x_{global})$, it indicates that some promising regions near x_{global} have not been explored, so then, the current iteration favors global exploration.

3.2. The Method for Maximizing the Weighted Expected Prediction Error

The calculation method for the weighted expected prediction error is introduced in this subsection.

To mitigate the over-reliance of the sampling criterion on the variance, a bias term is introduced to enhance the model's exploratory capability across the design space. As mentioned in Section 2.4, the bias–variance decomposition converts the expected prediction error at any candidate point into the sum of bias and variance. The variance can be predicted directly via Kriging, while the bias is the difference between the true value and the expectation of prediction value. The prediction value can also be predicted directly by using the Kriging model, but the true value is unknown. Therefore, it is necessary to consider other methods to calculate the bias of candidate points.

One of the characteristics of adaptive sampling is that it uses information obtained from the previous iterations to direct the subsequent iterations. Therefore, prior information can be used to calculate the bias of the candidate points in the subsequent iterations. In this paper, the Euclidean distance-based KNN algorithm is introduced to address the spatial correlation neglect in the traditional LOOCV's impact on the model performance. For each candidate point, KNN selects the K-nearest neighbor sample points based on the Euclidean distance.

Theoretically, a closer distance between an unknown candidate point and a known sample point indicates a higher correlation between them, i.e., the error at the known sample point reflects the uncertainty at the unknown candidate point to a greater extent. Therefore, for each candidate point, the weighted average sum of the LOOCVEs of the K sample points is used to replace the bias of the candidate point rather than the arithmetic sum. For a sample point closer to the candidate point, the error of that sample point should be assigned a greater weight, which is obviously a negative correlation. Consequently, the inverse proportion method is used to calculate the weights of each sample point (the weight calculation is detailed in Section 4.3). The weights of all sample points are normalized so that the sum of the weights is 1, as shown in Equation (24),

$$\omega_i = \frac{\frac{1}{d_i}}{\sum_{i=1}^{k} \frac{1}{d_i}}, \tag{23}$$

$$\sum_{i=1}^{k} \omega_i = 1, \tag{24}$$

where ω_i denotes the weight assigned to the $i-th$ nearest neighbor sample point, and d_i denotes the distance between the candidate point x and the $i-th$ nearest neighbor sample point to point x.

The weighted sum can also be calculated to obtain the bias of the candidate point x, as shown in Equation (25).

$$e(x) = \sum_{i=1}^{k} \omega_i \, e_i(x), \tag{25}$$

where $e_i(x)$ denotes the leave-one-out cross-correlation validation error of the $i-th$ nearest sample point to the candidate point x. The variance $s^2(x)$ of the candidate point x is predicted via Kriging. Then, the arithmetic sum of the bias $e(x)$ and the variance $s^2(x)$ is used to construct a new function, $WEPE(x)$, for calculating the expected prediction error of point x. And the expected prediction error of point x is termed as the weighted expected prediction error of point x, which quantifies the uncertainty of the candidate point x, as shown in Equation (26). Note that in "the weighted expected prediction error", the term "weighted" refers to the weights of leave-one-out cross-validation errors when calculating the bias, rather than the weights of the bias and variance.

$$WEPE(x) = \left(\sum_{i=1}^{k} \omega_i \, e_i(x) \right)^2 + s^2(x). \tag{26}$$

Therefore, in the adaptive sampling process, the maximization of the weighted expected prediction error is used as the filling criterion to identify the new sample point x_{new}, as shown in Equation (27).

$$x_{new} = arg \max_{x \in D} WEPE(x) = arg \max_{x \in D} \left(\sum_{i=1}^{k} \omega_i \, e_i(x) \right)^2 + s^2(x). \tag{27}$$

3.3. Implementation of the KMWEPE Method

The KMWEPE method proposed in this paper is shown in Figure 2. The details of the specific steps are as follows:

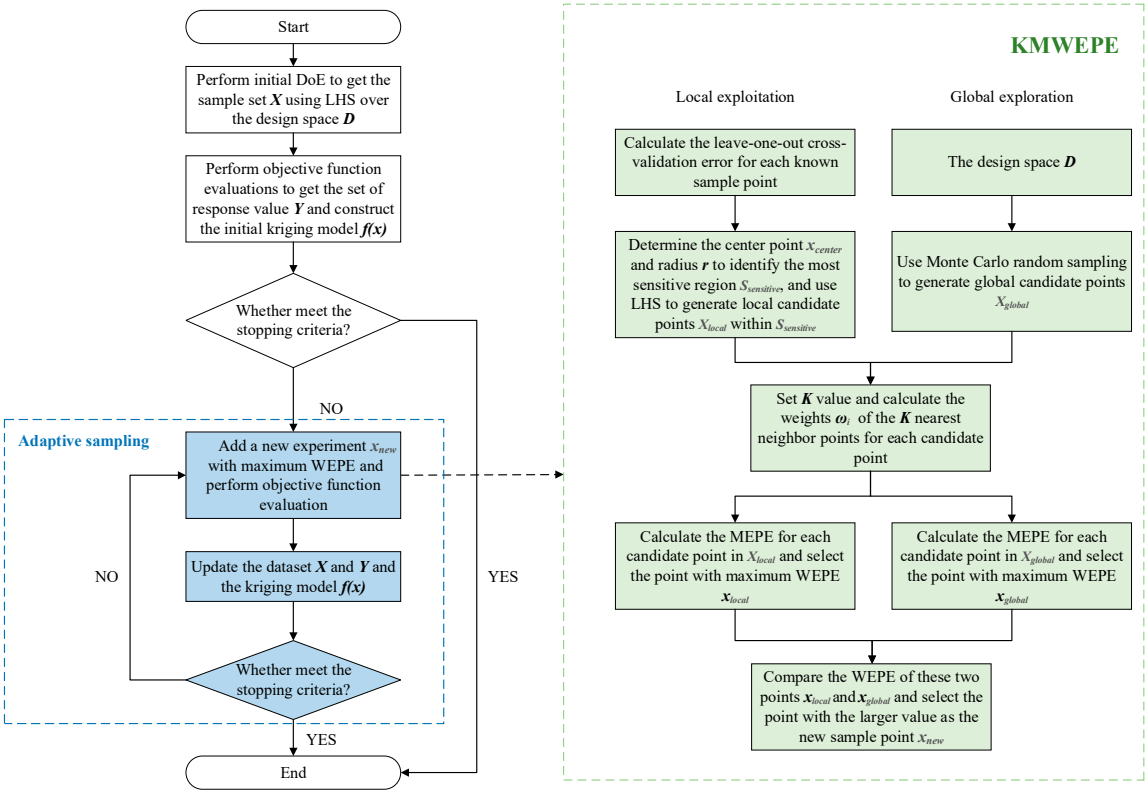

Figure 2. The flowchart of the KMWEPE method.

Step 1: The initial DoE is performed via LHS to obtain the initial sample set X, ensuring that the initial sample points are uniformly distributed in the design space. The number of initial sample points should be enough for constructing an effective initial Kriging model to provide sufficient prior information for the subsequent iterations. Conversely, an excessively large number may lead to overlapping information, which has a limited effect on the model accuracy. Therefore, the number of initial sample points in this paper is 5 times the dimension of the function, i.e., $5dim$ (where dim refers to the dimension of the real function).

Step 2: Expensive function evaluation is conducted on the sample set X to obtain the response set Y. Based on the input data X and output data Y, the initial Kriging model $f(x)$ is constructed by using Kriging interpolation.

Step 3: Determine whether the current Kriging model satisfies the stopping criterion. If so, then the iteration ends; if not, proceed to step 4.

Step 4: Calculate the LOOCVE for each known sample point in X, and select the sample point with the largest error as the center point x_{center}. The distance between x_{center} and the nearest sample point to x_{center} is used as the radius r, and the most sensitive region $D_{sensitive}$ is identified by the center point x_{center} and the radius r. LHS is used in $D_{sensitive}$ to generate a set of local candidate points S_{local}, and Monte Carlo Random Sampling is used in the design space to generate a set of global candidate points S_{global}. Note that S_{local} and S_{global} are resampled in each iteration.

Step 5: Set the value of K. For each candidate point in S_{local} and S_{global}, calculate the Euclidean distance between the candidate point and all the sample points. Select the K-nearest neighbor sample points, and calculate the weights of these K sample points based on the inverse proportion method.

Step 6: Obtain the prediction variance s^2 of each candidate point in S_{local} and S_{global} based on the updated Kriging model (if it is the first iteration based on the initial Kriging model), and calculate the WEPE of each candidate point based on Equation (26). Select the candidate point with the largest WEPE in S_{local} and S_{global}, respectively, denoted as x_{local} and x_{global}.

Step 7: Compare the WEPE of these two points, x_{local} and x_{global}, and select the point with the larger value as the new sample point x_{new}.

Step 8: Perform the expensive function evaluation of point x_{new} to obtain its response value y_{new}, and add x_{new} and y_{new} to the dataset X and Y, respectively. Update the Kriging model $f(x)$ and determine whether the current Kriging model satisfies the stopping criterion; if so, then the iteration ends; if not, proceed to step 4.

4. Discussion

In this chapter, a comprehensive discussion is presented on various aspects of the proposed KMWEPE method. The discussion encompasses the selection process of the K-value, the identification of the local most sensitive regions, the calculation method for the weights in the bias term of KNN prediction, the balance between the modeling accuracy and efficiency, and a series of tests.

To demonstrate the universality and effectiveness of KMWEPE, a series of tests were conducted using six benchmark functions, two publicly available datasets from the UCI repositories, and two engineering examples. The LHD and MEPE methods were also employed for comparative analysis.

Four distinct types of functions were selected for testing in order to test the impact of the proposed KMWEPE method on different types of functions. The first type of test functions is the multi-peak function represented by the Alpine01 function, which usually has multiple local minimum points and multiple peaks, making the modeling process more challenging. The second type is the plate-shaped function represented by the Zakharov5 function, which usually has a flat or plateau-like shape in certain regions. In flat regions, the function values remain constant within a certain range. In contrast, in plateau regions, the function values usually remain relatively constant in a certain region and then change drastically. The third type is the valley-shaped function represented by the Sixhump function, where the image of this type of function presents a downward concave shape, similar to the contour of a valley. The last type is other functions, including Colville, Hartman3, and Hartman6 functions.

4.1. Selection Process of the K-Value

The calculation of the bias of candidate points in KMWEPE is based on the KNN algorithm, and the value of K in KNN has a significant impact on the experimental results, which is often set artificially. In order to make the KMWEPE method more universal, the value of K should be different for the objective functions with different dimensions. Therefore, the value of K set in this study is proportional to the dimension of the objective function. Additionally, as the number of sample points set in the initial DoE is 5D (D here refers to the number of dimensions), i.e., in the first adaptive sampling process, the maximum number of samples that can be utilized is 5D. Consequently, the value of K is set to 1D, 2D, 3D, 4D, and 5D. Furthermore, due to the limitation of the number of initial sample points, and also taking into account the possibility that a greater number of sample points for obtaining information leads to a better modeling effect, this study adds another set of comparison experiments—K = all. In this case, K represents the number of all points in the current dataset (i.e., when applying KNN in each iteration, use all sample points in the current dataset to calculate the bias). The effect of each value of K on the results is compared in the subsequent experiments to further determine the optimal value of K.

In order to reduce the impact of sampling randomness on the experimental results, the KMWEPE approach was conducted with 100 independent repeated experiments of each test function for each of the six values of K. The number of sample points in the initial DoE is 5D, and the number of updated points added in the adaptive sampling is 6D. The mean and standard deviation of the RMSE for 100 independent repeated experiments are used as evaluation indicators. The experimental data are presented in Table 1, and the best results of the experiments for each benchmark function are highlighted in bold.

Table 1. Test results of six benchmark functions under KMWEPE method with different K-values.

Case	Dim	RMSE	KMWEPE					
			K = 1D	K = 2D	K = 3D	K = 4D	K = 5D	K = all
Alpine01	2	Mean	2.6845	2.4136	2.2494	2.0573	**1.9186** [1]	1.9439
		Std	0.6449	0.7267	0.6351	**0.5062**	0.5316	0.5378
Sixhump	2	Mean	5.2418	6.3476	4.8290	3.9682	**3.5054**	5.2389
		Std	**1.6058**	1.6887	1.6754	1.6250	1.8171	2.4976
Hartman3	3	Mean	0.4762	0.4515	0.4506	0.4124	0.3711	**0.3587**
		Std	0.0703	**0.0663**	0.0755	0.0704	0.1507	0.0815
Colville	4	Mean	2.5554×10^5	2.4541×10^5	2.4093×10^5	2.2780×10^5	$\mathbf{2.2553 \times 10^5}$	2.3722×10^5
		Std	4.2025×10^4	4.3255×10^4	5.2382×10^4	4.4335×10^4	4.4201×10^4	$\mathbf{3.9607 \times 10^4}$
Zakharov5	5	Mean	2.27570×10^6	1.85640×10^6	1.39210×10^6	1.21390×10^6	9.3133×10^5	$\mathbf{8.8238 \times 10^5}$
		Std	6.0684×10^5	4.4566×10^5	4.5664×10^5	3.4019×10^5	$\mathbf{2.7458 \times 10^5}$	3.334×10^5
Hartman6	6	Mean	0.4004	0.3361	0.3304	0.3184	**0.3145**	0.3242
		Std	0.0653	0.0616	**0.0573**	0.0577	0.0693	0.0829

[1] The bolded data in the table indicates the best result of the experiments for each benchmark function.

From the test results in Table 1, it can be seen that in the six groups of experiments, when K takes 5D, the RMSE mean value demonstrates a superior performance on the four test functions, including Alpine01, Sixhump, Colville, and Hartman6. For the other two test functions, Hartman3 and Zakharov5, the test results for K = 5D are only slightly worse than K = all. As for the RMSE standard deviation, except for the Hartman3 function where the standard deviation is twice as large when the value of K is taken as 5D compared to the other five sets of experiments, the standard deviations of the other five test function experiments show no significant variation. Therefore, it can be inferred that among the six K-values, the modeling effect is best when the value of K is 5D.

From a statistical perspective, further validation of the optimal value of K is conducted. To eliminate the differences in the magnitude of the RMSE across different functions, the RMSE for each function is independently normalized using max-min normalization. A one-way ANOVA is employed to analyze the impact of dimensionality on the value of K, as shown in Table 2. Initially, a Levene's test is performed, yielding a p-value of 0.2291, which is greater than 0.05, indicating that the homogeneity of variances is confirmed by Levene's test, satisfying the ANOVA assumption. Subsequently, a one-way ANOVA is conducted, resulting in a p-value of 8.39349×10^{-9}, which is less than 0.001, indicating an extremely significant difference between the groups, meaning that the value of K has a significant impact on the RMSE. The effect size calculation yielded $\eta^2 = 0.77$, reflecting the proportion of variance between the groups to the total variance, indicating that the intergroup differences accounted for 77% of the total variance, which is considered a large effect. Compared to other values (1D, 2D, 3D, 4D, and all), the RMSE mean for K = 5D decreased by 14.4%, 30.0%, 12.8%, 6.4%, 31.95%, and 7.4% across the six test functions, with an average decrease of 17.15%. Therefore, from a statistical standpoint, it is also demonstrated that the optimal value of K is 5D.

Table 2. One-way ANOVA results for RMSE across different K-values.

Source	SS	df	MS	F	p-Value	η^2
Between Groups	3.55506	5	0.71101	20.24	8.39349×10^{-9}	0.77
Within Groups	1.05403	30	0.03513			
Total	4.60909	35				

4.2. Identification of the Local Most Sensitive Regions

During the adaptive sampling process, it is more likely to search for informative sample points in areas with higher uncertainty. The KMWEPE method identifies the local most sensitive regions using the sample point with the largest LOOCVE and its distance to the nearest neighbor. However, this approach may simplify the complex uncertainty of the black-box function. Therefore, this subsection explores the identification of the local most sensitive regions through three metrics: the KMWEPE method, variance gradient, and gradient-based sensitivity.

All three metrics establish the local most sensitive regions by determining the center point and the radius, with the radius being determined by the distance between the center point and its nearest neighbor sample. The difference lies in the method of determining the center point. In each iteration, the KMWEPE method calculates the LOOCVE for all the sample points and uses the sample point with the largest LOOCVE as the center point. The variance gradient (VG) method calculates the partial derivatives of the predicted variance of the sample points in each dimension to obtain the gradient vector, and selects the point with the largest variance gradient magnitude as the center point. Gradient-based sensitivity (GBS), on the other hand, calculates the gradient of the function values of the sample points, and the sample point with the largest gradient magnitude is selected as the center point.

During a complete experiment involving multiple iterations, for each iteration, the local sensitive regions are identified using the aforementioned three methods. Within the local most sensitive region determined in the current iteration, 100D sample points are generated using LHS, and the RMSE value of the current model is calculated. The arithmetic mean of the RMSE values obtained throughout all the iterations of a complete experiment is calculated, as shown in Table 3.

Table 3. Test results of the local most sensitive regions under the KMWEPE, VG, and GBS methods.

	KMWEPE	VG	GBS
Alpine01	**4.1877** [1]	2.3489	2.6114
Sixhump	**19.9494**	10.8496	12.6722
Hartman3	**0.5209**	0.4096	0.4522
Colville	$\mathbf{1.7697 \times 10^5}$	1.5234×10^5	1.5541×10^5
Zakharov5	$\mathbf{1.7238 \times 10^6}$	6.6922×10^5	6.9398×10^5
Hartman6	**0.2739**	0.2245	0.2241

[1] The bolded data in the table indicates the best result of the experiments for each benchmark function.

As can be observed from Table 3, among the six test functions, the KMWEPE method yields the highest RMSE mean for the candidate points within the identified local most sensitive regions. This indicates that, compared to the VG and GBS methods, the local most sensitive regions determined by using KMWEPE exhibit the highest level of uncertainty. The sampling within these regions can potentially reduce the model error more rapidly.

The variance gradient method relies on the accuracy of the surrogate model; in sparse sample regions, the variance gradient may be falsely amplified. Gradient-based sensitivity is more applicable to smooth regions and is unstable under discrete data or noise interference. Large gradients may indicate local oscillations (such as high-frequency noise) and do not necessarily correspond to critical regions. Gradient-based methods only contain first-order information, reflecting instantaneous rates of change. In contrast, LOOCV directly quantifies the model's generalization ability in local regions by assessing the reconstruction error after sample point exclusion. It incorporates the influence of function values, gradients, and curvature, and reduces the dependence on individual samples through multiple resampling, thereby demonstrating more reliable performance in identifying sensitive regions.

4.3. Calculation Method for the Weights

The KMWEPE method employs an inverse distance weighting scheme to compute the bias from KNN-selected LOOCVE values, assuming a uniform decay of spatial influence. However, the design space of practical problems may not be uniform, and using an inverse distance weighting scheme may lead to estimation biases in regions with heterogeneous correlation structures. Therefore, this subsection discusses the weighting calculation methods for different spatial correlation functions.

Four methods are used to calculate the weights of candidate points: Inverse Distance (ID), Exponential Decay (ED), Gaussian Decay (GD), and Inverse-Square Distance (ISD). The ID method exhibits the uniform decay of weights with increasing distance; the ED method features strong correlation at close distances and weak correlation at far distances, with weights decaying exponentially—rapidly at first and then leveling off; the GD method's weights decay exponentially with the square of the distance, starting gently and then dropping sharply, providing a more balanced weight distribution for intermediate distances; and the ISD method's weights decay inversely with the square of the distance, more steeply than when using the ID method. Based on these four methods, the weights of candidate points are calculated to compute the bias. Ten independent repeated experiments were conducted on six test functions. The RMSE mean was used as the evaluation metric, and the test results are shown in Table 4.

Table 4. Test results of different weights under the ID, ED, GD, and ISD methods.

	ID	ED	GD	IISD
Alpine01	1.9488	**1.7352** [1]	2.4061	1.8364
Sixhump	3.4996	**3.4932**	3.6573	3.7598
Hartman3	0.4071	**0.4037**	0.4450	0.4042
Colville	2.6705×10^5	$\mathbf{2.0555 \times 10^5}$	2.1905×10^5	3.0149×10^5
Zakharov5	$\mathbf{8.7563 \times 10^5}$	1.5586×10^5	1.4430×10^5	1.1853×10^5
Hartman6	**0.2733**	0.3777	0.3188	0.3657

[1] The bolded data in the table indicates the best result of the experiments for each benchmark function.

The Alpine01 function exhibits a highly heterogeneous space, featuring multiple periodic sharp peaks with significant variations in the rate of change across different regions, and a nonlinear correlation decay. The Sixhump function presents moderate heterogeneity, with six local extrema, alternating flat and steep regions, and regionally dependent correlation structures. The Hartman3 function is locally homogeneous, consisting of the superposition of four Gaussian peaks, with uniform correlation decay within each peak but abrupt changes in the correlation between peaks. The Colville function exhibits complex heterogeneity, with multiple interacting terms and anisotropic correlation structures in different directions. The Zakharov5 function is globally homogeneous, with its overall correlation decaying uniformly. The Hartman6 function is highly localized, comprising a combination of six-dimensional Gaussian peaks, with strong internal correlations within each peak and rapid decay to zero outside the peaks.

As can be observed from Table 4, among the six sets of experiments, the ED method demonstrates the optimal performance on the Alpine01, Sixhump, Hartman3, and Colville functions, while the ID method exhibits the optimal performance on the Zakharov5 and Hartman6 functions. For the globally homogeneous Zakharov5 function, the ID method significantly outperforms the other three methods, and it also shows a good performance on the locally homogeneous Hartman3 and Hartman6 functions. It can be inferred that the ID method is suitable for cases with uniform spatial correlation. When calculating the bias of KNN predictions, a *K*-value of 5D is used, meaning that the local region encompasses the 5D nearest neighbor samples of the candidate point. This approach excludes the consideration of distant regions in the design space, thereby focusing exclusively on the local spatial correlations around the candidate point. Consequently, the ED method, which emphasizes strong near-distance correlations, demonstrates a superior performance across all the four test functions.

In practical problems, the characteristics of the design space vary, and when calculating the weights of neighboring samples for candidate points, it is crucial to choose different weighting methods based on the characteristics of the design space. For functions with uniform spatial correlation, the inverse distance method should be selected to calculate weights, whereas for functions with strong spatial correlation decay, exponential decay will be a better choice.

4.4. Balance Between Modeling Accuracy and Efficiency

In the modeling process, there is typically a trade-off between the modeling accuracy and modeling efficiency. An improved modeling accuracy is often achieved at the expense of reduced modeling efficiency, and vice versa. Striking an optimal balance between these two factors is a critical consideration in the modeling process.

To demonstrate how the KMWEPE method balances the accuracy and efficiency, another adaptive sampling method—MEPE (Maximization of the Expected Prediction Error)—

is selected for comparative testing. The experimental results of MEPE and KMWEPE are analyzed to evaluate their respective performance in terms of accuracy–efficiency trade-offs.

The LHD, MEPE, and KMWEP methods are evaluated across six test functions. To ensure statistical robustness, both the mean value and standard deviation of the RMSE from 100 independent repeated experiments are also employed as evaluation metrics, with the comprehensive experimental results detailed in Table 5. Since LHD is a non-adaptive sampling method, the comparative analysis focuses exclusively on the accuracy–efficiency trade-off between the MEPE and KMWEPE methods.

Table 5. Test results of six benchmark functions under LHD, MEPE, and KMWEPE methods.

Case	Dimension	RMSE	Approach		
			LHD	MEPE	KMWEPE
Alpine01	2	Mean	2.5398	2.8711	**1.9186** [1]
		Std	0.7186	**0.5246**	0.5316
Sixhump	2	Mean	8.0013	6.897	**3.5054**
		Std	2.7967	2.0964	**1.8171**
Hartman3	3	Mean	0.6057	0.4798	**0.4056**
		Std	0.1423	**0.0755**	0.0773
Colville	4	Mean	2.8503×10^5	2.6981×10^5	$\mathbf{2.2553 \times 10^5}$
		Std	5.5368×10^4	$\mathbf{4.1837 \times 10^4}$	4.4201×10^4
Zakharov5	5	Mean	1.4862×10^6	2.0460×10^6	$\mathbf{1.0280 \times 10^6}$
		Std	5.9755×10^5	4.9094×10^5	$\mathbf{3.3854 \times 10^5}$
Hartman6	6	Mean	0.3693	0.3356	**0.3164**
		Std	0.1103	0.065	**0.0631**

[1] The bolded data in the table indicates the best result of the experiments for each benchmark function.

According to the experimental data in Table 5, compared to the MEPE method, KMWEPE shows the best performance in terms of the RMSE mean value across the six test functions, and has a better RMSE standard deviation across the three test functions, including Sixhump, Zakharov5, and Hartman6. The RMSE standard deviation across the other three test functions, including Alpine01, Hartman3, and Colville is only slightly inferior to MEPE, indicating that KMWEPE has a better performance in regard to the modeling accuracy.

However, MEPE is a single-candidate-set method that dynamically generates one candidate set per iteration, whereas KMWEPE employs a dual-candidate-set strategy by simultaneously generating candidate sets in both the global design space and local most sensitive regions during each iteration. The size of each candidate set is consistently set to 100D. Consequently, while KMWEPE achieves an improved modeling accuracy compared to MEPE, this enhancement comes at the expense of increased computational costs—a typical trade-off in engineering applications where a higher model fidelity justifies additional computational expenditure.

To quantitatively demonstrate the rationality of KMWEPE's accuracy–computation trade-off, the normalized Precision per Time Efficiency (nPTE) metric is introduced. Since different test functions exhibit varying RMSE magnitudes and computational time requirements, both the relative RMSE improvement and relative time increment are calculated to eliminate scaling effects and hardware-dependent variations. The nPTE calculation formula for comparing the MEPE and KMWEPE methods is presented in Equation (28), with the computed nPTE values for all six test functions summarized in Table 6.

$$nPTE = \frac{\frac{(RMSE_{MEPE} - RMSE_{KMWEPE})}{RMSE_{MEPE}}}{\frac{(Time_{KMWEPE} - RMSE_{MEPE})}{RMSE_{MEPE}}} \times 100\% \quad (28)$$

Table 6. nPTE evaluation results across six test functions.

	Alpine01	Sixhump	Hartman3	Colville	Zakharov5	Hartman6
nPTE	23.13%	24.53%	16.59%	13.57%	25.96%	6.63%

The physical meaning of nPTE represents the percentage reduction in the RMSE per unit increase in the computational time. For instance, in the case of the Alpine01 function, the KMWEPE method achieves a 23.13% decrease in the RMSE value per additional unit of computation time compared to the MEPE method. As evidenced by Table 6, all the six test functions exhibit positive nPTE values, demonstrating that the KMWEPE method significantly enhances the modeling accuracy of the Kriging surrogate while incurring only a controlled increase in the computational cost. This systematic improvement confirms that the additional computational resources required via KMWEPE are effectively converted into measurable performance gains, as quantitatively reflected by the reduced RMSE values across all the benchmark functions.

Meanwhile, the nPTE values generally exhibit a decreasing trend with increasing dimensionality, indicating diminishing returns in the precision improvement per unit computational time (with the notable exception of the Zakharov5 function, where the KMWEPE method significantly improves the RMSE mean). This demonstrates that while KMWEPE maintains superior performance over MEPE in higher dimensional scenarios, its marginal benefits decrease progressively with dimension growth. This phenomenon fundamentally reflects the impact of the "curse of dimensionality" on modeling—as the dimensions increase, the computational costs grow exponentially, and while some accuracy improvement is achieved, the rate of the computational cost escalation outpaces the precision gains. Consequently, the KMWEPE method proves particularly effective for low-dimensional problems.

The curse of dimensionality presents significant challenges in high-dimensional modeling. Previous work has addressed the dimension reduction techniques; for instance, principal component analysis (PCA) has been implemented in Kriging modeling to achieve faster modeling efficiency under the premise of meeting certain accuracy requirements [50]. In the future research, the integration of multidimensional scaling, principal component analysis, and other dimensionality reduction algorithms can be considered within the KMWEPE method. On the one hand, the adaptive sampling process after dimensionality reduction can greatly improve the modeling efficiency and reduce the computational cost. On the other hand, the efficiency of the KMWEPE method in low-dimensional modeling can be fully utilized. Additionally, exploring parallel or distributed computing techniques could further reduce the computational costs and broaden the KMWEPE method's applicability.

4.5. Publicly Available Dataset Testing

The aforementioned experiments are all based on benchmark test functions. To validate the practicality of the KMWEPE method in real-world engineering problems, this subsection selects two real datasets from the UCI repositories (Auto MPG dataset and Forest Fire dataset) for verification.

The Auto MPG dataset is used for predicting the fuel efficiency, consisting of 398 samples, including seven features and one target variable. The seven features are as follows: cylinders (number of engine cylinders, with common values of 4, 6, and 8); displacement (engine displacement in cubic inches, reflecting the engine size); horsepower (engine horsepower (HP), with some missing values filled with the mean); weight (vehicle weight in pounds, which directly affects the fuel efficiency); acceleration (acceleration performance, i.e., the time required to accelerate from 0 to 60 mph in seconds, where a smaller value indicates faster acceleration); model_year (model year, reflecting technological iteration);

and origin (origin code, a categorical variable where 1 indicates made in the USA, 2 indicates Europe, and 3 indicates Japan). The target variable is mpg (miles per gallon), which measures the fuel efficiency, with higher mpg values indicating better fuel economy.

The validation using datasets differs from that using benchmark test functions. When using benchmark test functions, the candidate points are generated from the given design space for selection, allowing for infinite sampling. However, when validating with datasets, the real dataset is finite, and all data points are given. In this case, the essence of the adaptive sampling process is data selection, and the training set and candidate set are continuously updated. For the KMWEPE method, it is not possible to establish the local most sensitive regions when validating with datasets, and thus, high-uncertainty candidate points cannot be generated. Therefore, the adaptive sampling process can only be based on the bias term predicted by KNN, selecting the next iteration point according to the WEPE value, which to some extent limits the improvement of the modeling accuracy by using the KMWEPE method.

Specifically, when validating with the seven-dimensional Auto MPG dataset, 5D samples are randomly selected from the dataset as the initial dataset to establish the initial Kriging model, and the remaining data are used as the candidate set. The WEPE values for all the candidate points are calculated based on the KNN algorithm, the candidate point with the highest WEPE value is selected as the new iteration point, and the Kriging model is updated. The number of sample points added by the adaptive sampling process is 6D.

The Forest Fire dataset is used for predicting the burned area of forest fires, derived from fire records between January 2000 and December 2003 in the Montesinho Natural Park, Portugal. The Forest Fire dataset contains 517 samples, encompassing 12 features and one target variable. Due to the low predictive contribution and high correlation with the core features of some features, this study selects the five most core features from the Forest Fire dataset as input variables: X, Y, month, temp, and RH. X and Y represent the spatial location of the fire area, which is strongly correlated with the fire risk; month represents the time variable, as fires exhibit significant seasonality; temp and RH represent meteorological variables—temperature and relative humidity. The target variable, area, represents the burned area of the fire. Similarly, when validating with the Forest Fire dataset, the initial dataset contains 5D samples, and the adaptive process adds 6D samples. Finally, the remaining data are used as the test set, and the mean and standard deviation of the RMSE from 100 independent repeated experiments are used as the evaluation metrics. The test results for the Auto MPG and Forest Fire datasets are shown in Table 7.

Table 7. Test results for the Auto MPG and Forest Fire datasets.

		MEPE	KMWEPE
Auto MPG	Mean	2.9808	**2.8506** [1]
	Std	1.3798	**1.2199**
	Time(s)	**169.52**	171.00
Forest Fire	Mean	3.1551	**2.9035**
	Std	2.8161	**2.5628**
	Time(s)	**50.94**	55.74

[1] The bolded data in the table indicates the best result of the experiments for each dataset.

As shown in Table 7, compared to the MEPE method, the KMWEPE method still maintains better performance on the five-dimensional Forest Fire dataset and the seven-dimensional Auto MPG dataset without generating a dynamic dual candidate set, which demonstrates the effectiveness of the KNN-based bias term. The bias predicted by KNN explains the true bias of the candidate points to a greater extent, thereby filtering out more informative sample points. On the other hand, without a dynamic dual candidate set, the

improvement of the KMWEPE method over the MEPE method is reduced, which also reflects the effectiveness of the local most sensitive region in the dynamic dual candidate set for generating high-uncertainty candidate points.

In conclusion, whether it is for practical engineering problems (such as vehicle fuel prediction in the Auto MPG dataset) or real-world case applications (such as fire area prediction in the Forest Fire dataset), the KMWEPE method can effectively improve the modeling process in relation to prediction, thereby enhancing the prediction accuracy. It holds certain value and significance for global approximation problems based on the Kriging model.

4.6. Benchmark Functions and Engineering Examples' Testing

This section demonstrates the effectiveness of the KMWEPE method through comparative analysis using both the LHD and MEPE methods.

Table 5 presents the experimental results of the three methods. The experimental data provide an objective explanation of the effectiveness of KMWEPE, while images provide a more intuitive observation. For two-dimensional functions, the DACE toolbox can be used to draw the MSE grid curve, which provides a more intuitive representation of the overall MSE of the function and its overall trend of change. This paper first presents the MSE grid curves of the two 2D functions—Alpine01 and Sixhump—after conducting a complete adaptive modeling experiment using the three testing methods, as shown in Figures 3 and 4. The bottom plane is a contour map that reflects the MSE. The color gradient ranges from blue (indicating lower MSE values) to red (indicating higher MSE values), providing a visual representation of how MSE varies across the parameter space.

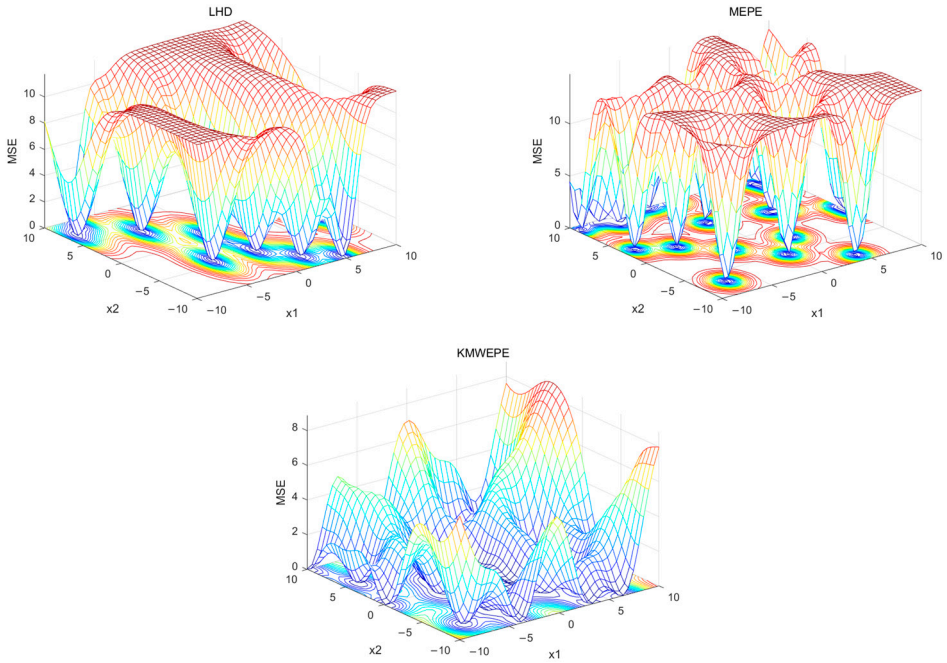

Figure 3. MSE grid curves of Kriging of the Alpine01 function obtained by using three different testing methods.

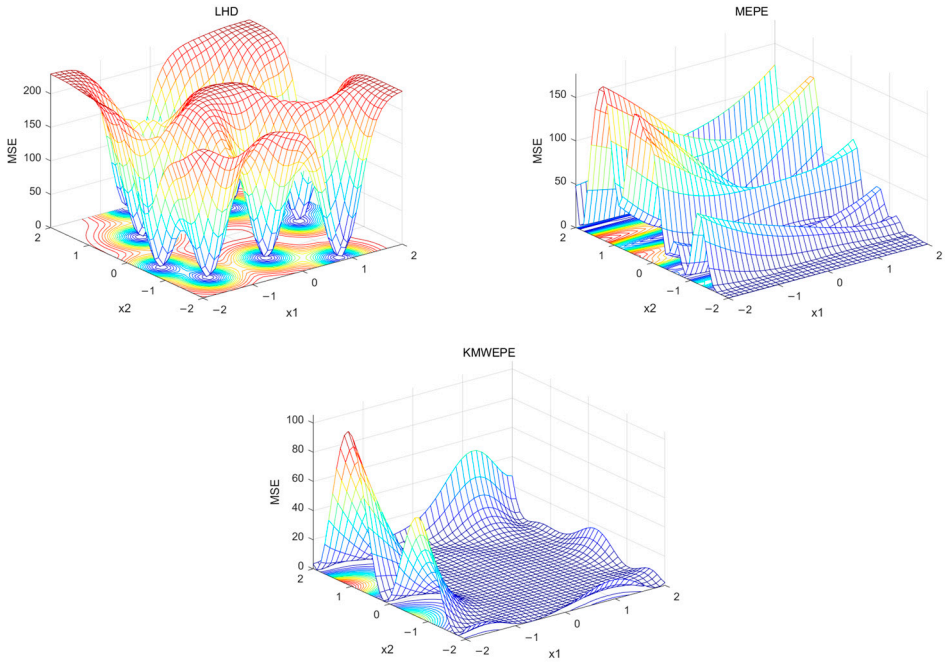

Figure 4. MSE grid curves of Kriging of the Sixhump function obtained by using three different testing methods.

From Figures 3 and 4, it can be observed that for the Alpine01 function, the MSE of the Kriging model obtained via LHD and MEPE are greater than 10 in most areas of the design space, and the changes are relatively drastic. In contrast, the MSE of the Kriging model obtained via KMWEPE are almost all less than 8 in the design space and relatively flat. For the Sixhump function, the MSE of the Kriging model obtained via LHD is above 100 in most areas of the design space, exhibiting a high degree of variability, while the Kriging model obtained via MEPE is extremely non-smooth. In contrast, the MSE of the Kriging model obtained via KMWEPE is close to zero in most regions with only a small portion having a large MSE, and the entire region is also smoother.

For functions with more than two dimensions, it is difficult to use grid curves to represent the overall trend of the MSE. Therefore, the test results are presented in the form of box plots, as shown in Figures 5–7.

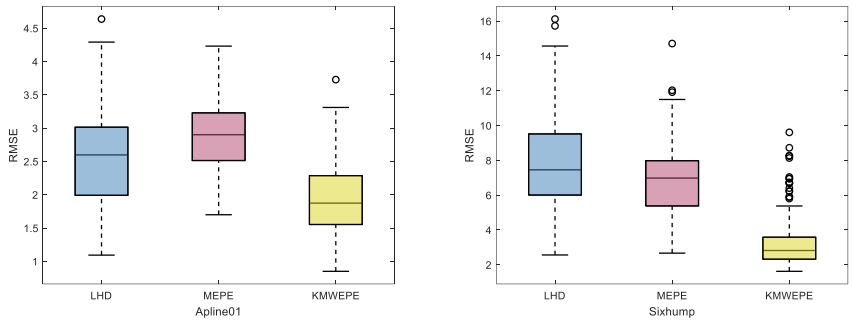

Figure 5. Box plot of Apline01 and Sixhump function test results.

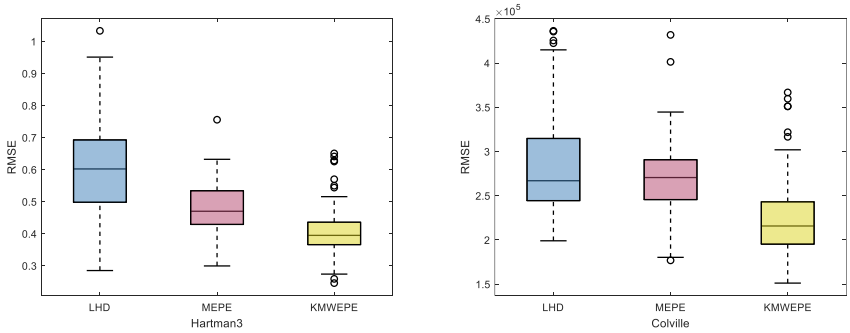

Figure 6. Box plot of Hartman3 and Colville function test results.

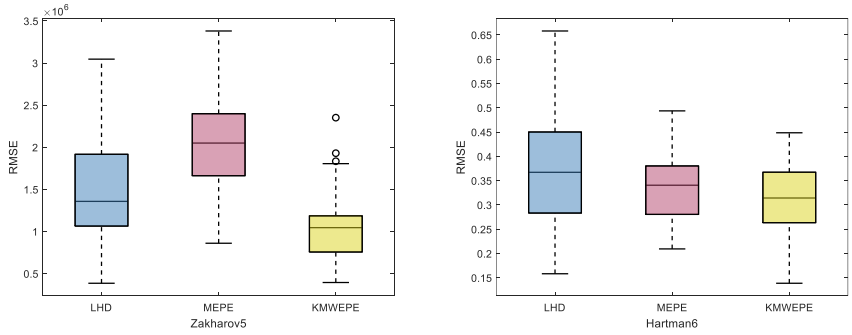

Figure 7. Box plot of Zakharov5 and Hartman6 function test results.

The box plot shows the distribution and dispersion of data from 100 independent repeated experiments, and the lines inside the box represent the median of the data. As shown in Figures 5–7, in the comparison experiments of the three different methods for six test functions, it can be observed that the modeling accuracy of KMWEPE is better than that of LHD and MEPE. Among them, the test results of the Sixhump, Hartman3, and Colville functions via KMWEPE are significantly superior to those of the other two methods, and the other three test functions also demonstrate the effectiveness of KMWEPE to a certain extent. Although some data points of the experimental results via KMWEPE are separated from the box by a certain distance, the data distribution of the test function experimental results via KMWEPE is significantly better than via LHD and MEPE.

This study also focuses on the convergence trend of the RMSE with the increase in the number of update points in an experiment, and the convergence curves of six benchmark test functions under MEPE and KMWEPE are plotted. The number of sample points in the initial DoE is 5D, and the number of updated points is increased to 60D in order to better observe the convergence trend of the RMSE. As shown in Figures 8–10, except for the Alpine01 function, the other five test functions can converge when sufficient updated points are added to the adaptive sampling phase of KMWEPE. Among them, the Colville and Zakharov5 functions showed the best experimental results, achieving convergence in the initial stage of the iteration. The Sixhump, Hartman3, and Hartman6 functions can gradually converge after adding 20D updated points. After achieving convergence, the RMSE obtained via KMWEPE are all less than via MEPE. Even for the Apline01 function that failed to achieve convergence, the modeling accuracy of KMWEPE is better than that of MEPE after adding more than 20D updated points.

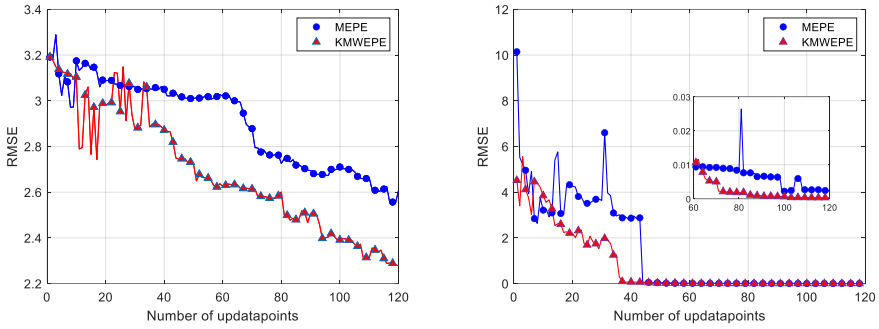

Figure 8. Convergence curve of Apline01 and Sixhump function test results.

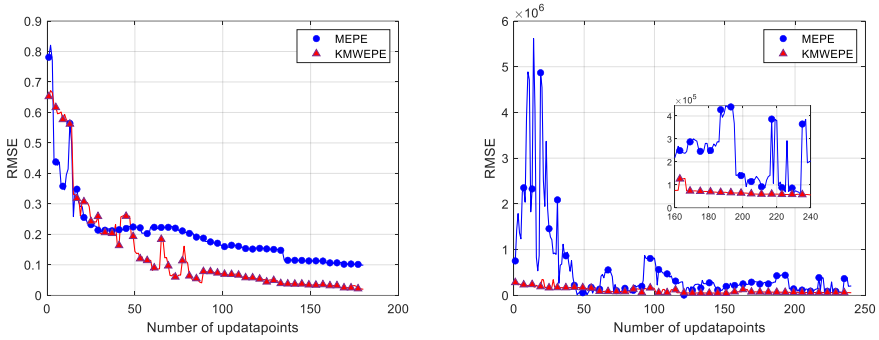

Figure 9. Convergence curve of Hartman3 and Colville function test results.

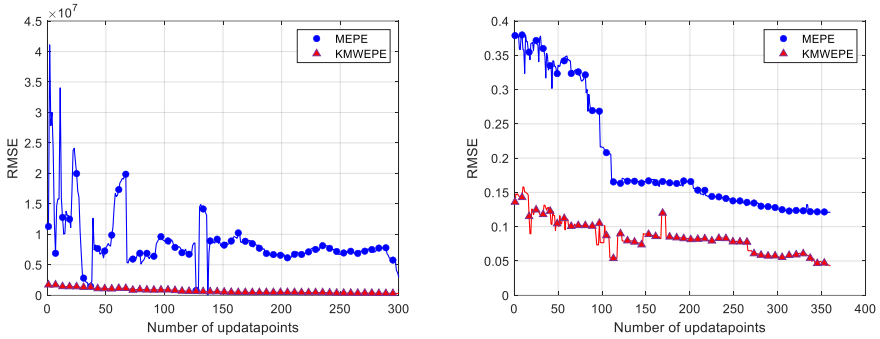

Figure 10. Convergence curve of Zakharov5 and Hartman6 function test results.

From the above analysis, it can be observed that from both the objective experimental data and the intuitive images, KMWEPE shows a better modeling performance compared to LHD and MEPE.

Two engineering examples are also used to demonstrate the effectiveness of KMWEPE. The first engineering example is the design problem of helical tension cylinder springs. The Helical Tension Cylinder Spring (HTCS) is a high-temperature alloy spring with good elasticity and high-temperature resistance, which is suitable for engineering applications in high-temperature environments, such as automotive engines, aerospace, machinery manufacturing, etc. The design schematic of the HTCS is shown in Figure 11.

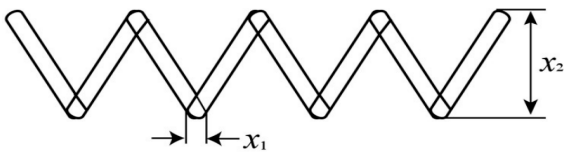

Figure 11. The design schematic of HTCS.

The HTCS design problem is a three-dimensional problem with three design variables: the spring diameter x_1, the average coil diameter x_2, and the number of active coils x_3. The response function of this design problem is shown in Equation (29). The HTCS design problem represents a complex engineering optimization challenge, where the primary objective is to minimize the spring weight while satisfying specified performance constraints. This study selects the HTCS problem as a test case precisely because of its inherent nonlinearity and multimodal characteristics, which make it particularly suitable for validating the effectiveness of the modeling methods.

$$f(x) = (2 + x_3)x_1^2 x_2, x_1 \in [0.05, 2], x_2 \in [0.25, 1.3], x_3 \in [2, 15]. \tag{29}$$

The second engineering example is the Output Transformer Less (OTL) circuit. In traditional power amplifier circuits, the output transformer plays the role of signal transmission and impedance matching, but the output transformer is often expensive and large in size. The OTL circuits are designed to connect directly to the load, eliminating the need for an output transformer and thus reducing the cost and size. OTL circuits are suitable for applications such as audio amplifiers, high-fidelity sound systems, laboratory instruments, and communication equipment.

The OTL circuit design problem is a six-dimensional problem with six design variables of five resistors $R_{b1}, R_{b2}, R_f, R_{c1}, R_{c2}$ and the current gain β. The OTL circuit function models an output transformerless push–pull circuit, and the response is the midpoint voltage V_m with the expression shown in Equation (31).

$$V_m(x) = \frac{(V_{b1} + 0.74)\beta(R_{c2} + 9)}{\beta(R_{c2} + 9) + R_f} + \frac{11.35R_f}{\beta(R_{c2} + 9) + R_f} + \frac{0.74R_f\beta(R_{c2} + 9)}{\Big(\beta(R_{c2} + 9) + R_f\Big)R_{c1}}. \tag{30}$$

$$\begin{gathered} V_{b1} = \frac{12R_{b2}}{R_{b1} + R_{b2}}, R_{b1} \in [50, 150], R_{b2} \in [25, 70], R_f \in [0.5, 3], R_{c1} \in [1.2, 2.5], \\ R_{c2} \in [1.2, 2.5], \beta \in [50, 300]. \end{gathered} \tag{31}$$

As above, the experimental setup involves 100 independent repeated experiments. The number of sample points in the initial DoE is 5D, and the number of updated points in the adaptive sampling phase is 6D. The experimental test results for the HTCS and OTL circuits are shown in Tables 8 and 9.

Table 8. Test results for spring design problems.

		LHD	MEPE	KMWEPE
RMSE	Mean	4.3711	2.2041	**1.1705** [1]
	Std	1.5320	0.6440	**0.4376**

[1] The bolded data in the table indicates the best result of the experiment.

Table 9. Test results for OTL circuit design problems.

		LHD	MEPE	KMWEPE
RMSE	Mean	0.8248	0.6099	**0.5742** [1]
	Std	0.1120	0.0962	**0.0487**

[1] The bolded data in the table indicates the best result of the experiment.

From Tables 8 and 9, for the spring design problem, the RMSE mean value of MEPE is almost half that of LHD, while the RMSE mean value of KMWEPE is further reduced by nearly half compared to that of MEPE. Additionally, the RMSE standard deviation of KMWEPE is also the smallest among the three methods, which shows that KMWEPE significantly improves the modeling accuracy compared to LHD and MEPE for the spring

design problem. For the OTL circuit design problem, the RMSE mean value and standard deviation of KMWEPE are smaller than when using the other two methods. Moreover, as shown in Figure 12, for the two engineering example design problems, not only the RMSE mean value and standard deviation of KMWEPE are better than those of the other two methods, but the dispersion of the test result data for the 100 independent repeated experiments is also superior. Under the KMWEPE method, all the data for the spring design problem are within the box, and for the OTL circuit design problem, the number of experimental data outside the box are also less than for the other two methods.

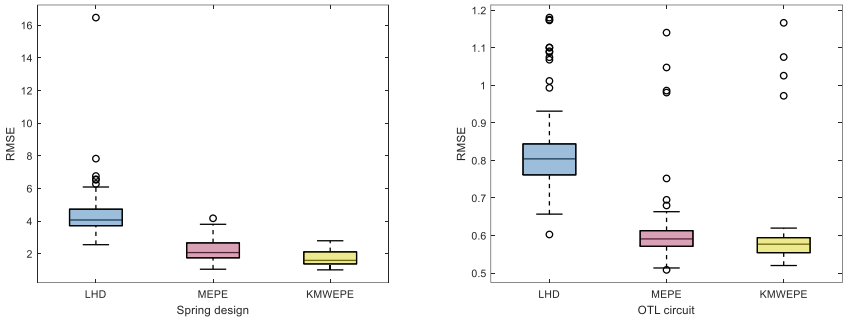

Figure 12. Box plot of spring design and OTL circuit function test results.

Similarly, the convergence curves of the test results for the two engineering examples are also plotted for a more visual representation of the test results, as shown in Figure 13. For the spring design problem, although the RMSE of MEPE decreases rapidly and converges quickly during the iteration process, the RMSE of KMWEPE remains lower than for MEPE when the updated point is greater than 30D. For the OTL circuit problem, the RMSE of KMWEPE decreases rapidly, becomes lower than the RMSE of MEPE at the beginning of the iteration process, and maintains the better performance throughout the subsequent iterations.

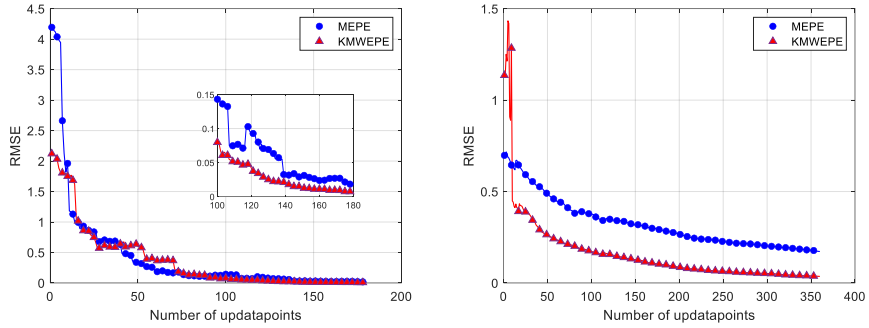

Figure 13. Convergence curves for spring design and OTL circuit function test results.

5. Conclusions

To enhance the modeling accuracy of Kriging, this study proposes a novel adaptive sampling method based on *K*-nearest neighbors (KNN) by maximizing the weighted expected prediction error (KMWEPE). In each iteration, two sets of candidate points are generated in the most sensitive region and the design space, respectively, to achieve a dynamic balance between local exploitation and global exploration. The uncertainty of candidate points is quantified using a new function of weighted expected prediction error, which is constructed based on bias–variance decomposition and the KNN algorithm. A new filling criterion is then applied to select the point with the highest uncertainty from the two sets as the updated sample point.

The experimental results from the six benchmark test functions and two engineering application examples demonstrate the superior performance of the KMWEPE method compared to the LHD and WEPE methods. However, according to the experimental data, the KMWEPE method is more effective in low-dimensional problems.

Therefore, the future research can focus on applying the KMWEPE method to high-dimensional problems and the improvement of the modeling efficiency. Dimensionality reduction algorithms can be considered within the KMWEPE method. The modeling efficiency can be greatly improved, and the efficiency of the KMWEPE method in low-dimensional modeling can be fully utilized.

Author Contributions: Methodology, J.S. and Y.L.; software (MATLAB R2020b), Y.X.; validation, Y.X.; formal analysis, Y.X.; writing—original draft preparation, Y.X.; writing—review and editing, J.S., Y.X. and Y.L.; visualization, Y.X.; supervision, Z.Z. and W.L.; funding acquisition, J.S., Y.L. and W.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was financed by the National Natural Science Foundation of China [Grant number 12272354 and 52375270], the Program for Innovative Research Team in Science and Technology of Henan Province [Grant number 25IRTSTHN023], and the Open Foundation of the Guangdong Provincial Key Laboratory of Electronic Information Products Reliability Technology [Grant number GDDZXX202503]: Research on Reliability Simulation Optimization Based on Reinforcement Learning and High-Dimensional Kriging-AI Modeling.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding authors.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

DoE	Design of Experiments
HTCS	Helical Tension Cylinder Spring
KMWEPE	KNN-based Maximization of Weighted Expected Prediction Error
KNN	*K*-Nearest Neighbors
LHS	Latin Hypercube Sampling
LOOCVE	Leave-One-Out Cross-Validation Error
MEPE	Maximization of Expected Prediction Error
MSE	Mean Square Error
OLHS	Optimal Latin Hypercube Sampling
OTL	Output Transformer less
RMSE	Root Mean Square Error
WEPE	Weighted Expected Prediction Error

References

1. Chen, H.; Zhang, Z.; Li, W.; Liu, Q.; Sun, K.; Fan, D.; Cui, W. Ensemble of surrogates in black-box-type engineering optimization: Recent advances and applications. *Expert Syst. Appl.* **2024**, *248*, 123427. [CrossRef]
2. Li, Z.; Li, X.; Li, C.; Ge, J.; Qiu, Y. A Semi-Parallel Active Learning Method Based on Kriging for Structural Reliability Analysis. *Appl. Sci.* **2023**, *13*, 1036. [CrossRef]
3. Beizer, B.; Wiley, J. Black Box Testing: Techniques for Functional Testing of Software and Systems. *IEEE Softw.* **1996**, *13*, 98–105. [CrossRef]
4. Shahriari, B.; Swersky, K.; Wang, Z.; Adams, R.; de Freitas, N. Taking the Human Out of the Loop: A Review of Bayesian Optimization. *Proc. IEEE.* **2016**, *104*, 148–175. [CrossRef]

5. Ling, C.; Lu, Z.; Feng, K. An efficient method combining adaptive Kriging and fuzzy simulation for estimating failure credibility. *Aerosp. Sci. Technol.* **2019**, *92*, 620–634. [CrossRef]
6. Arcidiacono, G.; Berni, R.; Cantone, L.; Nikiforova, N.; Placidoli, P. A Kriging modeling approach applied to the railways case. *Procedia Struct. Integr.* **2018**, *8*, 163–167. [CrossRef]
7. Giovanni, V.; Ernesto, B.; Łukasz, Ł. A Kriging-assisted multiobjective evolutionary algorithm. *Appl. Soft Comput.* **2017**, *58*, 155–175.
8. Ma, Y.-Z.; Liu, M.; Nan, H.; Li, H.-S.; Zhao, Z.-Z. A novel hybrid adaptive scheme for Kriging-based reliability estimation—A comparative study. *Appl. Math. Model.* **2022**, *108*, 1–26. [CrossRef]
9. Arora, G.; Bala, K.; Emadifar, H.; Khademi, M. A review of radial basis function with applications explored. *J. Egypt. Math. Soc.* **2023**, *31*, 6. [CrossRef]
10. Majdisova, Z.; Skala, V. Radial basis function approximations: Comparison and applications. *Appl. Math. Model.* **2017**, *51*, 728–743. [CrossRef]
11. Tripathy, R.K.; Bilionis, I. Deep UQ: Learning deep neural network surrogate models for high dimensional uncertainty quantification. *J. Comput. Phys.* **2018**, *375*, 565–588. [CrossRef]
12. White, D.A.; Arrighi, W.J.; Kudo, J.; Watts, S.E. Multiscale topology optimization using neural network surrogate models. *Comput. Meth. Appl. Mech. Eng.* **2019**, *346*, 1118–1135. [CrossRef]
13. Pan, Y.; Qin, J.; Hou, Y.; Chen, J. Two-stage support vector machine-enabled deep excavation settlement prediction considering class imbalance and multi-source uncertainties. *Reliab. Eng. Syst. Saf.* **2024**, *241*, 109578. [CrossRef]
14. Hu, Z.; Tang, C.; Liang, Y.; Chang, S.; Ni, X.; Xiao, S.; Meng, X.; He, B.; Liu, W. Feature Detection Based on Imaging and Genetic Data Using Multi-Kernel Support Vector Machine–Apriori Model. *Mathematics* **2024**, *12*, 684. [CrossRef]
15. Novák, L.; Sharma, H.; Shields, M.D. Physics-informed polynomial chaos expansions. *J. Comput. Phys.* **2024**, *506*, 112926. [CrossRef]
16. Modak, A.; Chakraborty, S. An enhanced learning function for bootstrap polynomial chaos expansion-based enhanced active learning algorithm for reliability analysis of structure. *Struct. Saf.* **2024**, *109*, 102467. [CrossRef]
17. Borisut, P.; Nuchitprasittichai, A. Adaptive Latin Hypercube Sampling for a Surrogate-Based Optimization with Artificial Neural Network. *Processes* **2023**, *11*, 3232. [CrossRef]
18. Phromphan, P.; Suvisuthikasame, J.; Kaewmongkol, M.; Chanpichitwanich, W.; Sleesongsom, S. A New Latin Hypercube Sampling with Maximum Diversity Factor for Reliability-Based Design Optimization of HLM. *Symmetry* **2024**, *16*, 901. [CrossRef]
19. Rajabi, M.; Ataie-Ashtiani, B.; Janssen, H. Efficiency enhancement of optimized Latin hypercube sampling strategies: Application to Monte Carlo uncertainty analysis and meta-modeling. *Adv. Water Resour.* **2015**, *76*, 127–139. [CrossRef]
20. Wang, Z.; Zhao, D.; Heidari, A.; Chen, Y.; Chen, H.; Liang, G. Improved Latin hypercube sampling initialization-based whale optimization algorithm for COVID-19 X-ray multi-threshold image segmentation. *Sci. Rep.* **2024**, *14*, 13239. [CrossRef]
21. Mondal, A.; Mandal, A. Stratified random sampling for dependent inputs in Monte Carlo simulations from computer experiments. *J. Stat. Plan. Inference.* **2020**, *205*, 269–282. [CrossRef]
22. Apers, S.; Gribling, S.; Szilagyi, D. Hamiltonian Monte Carlo for efficient Gaussian sampling: Long and random steps. *J. Mach. Learn. Res.* **2024**, *25*, 1–30.
23. Davis, S.; Cremaschi, S.; Eden, M. Efficient Surrogate Model Development: Impact of Sample Size and Underlying Model Dimensions. *Comput. Aid. Chem. Eng.* **2018**, *5*, 412–425.
24. Forrester, A.; Keane, A. Recent advances in surrogate-based optimization. *Prog. Aerosp. Sci.* **2009**, *45*, 50–79. [CrossRef]
25. Gramacy, R.; Lee, H. Bayesian Treed Gaussian Process Models With an Application to Computer Modeling. *J. Am. Stat. Assoc.* **2008**, *103*, 1119–1130. [CrossRef]
26. Jones, D.; Schonlau, M. Efficient Global Optimization of Expensive Black-Box Functions. *J. Glob. Optim.* **1998**, *13*, 455–492. [CrossRef]
27. Regis, R.G.; Shoemaker, C.A. A Stochastic Radial Basis Function Method for the Global Optimization of Expensive Functions. *Inf. J. Comput.* **2007**, *19*, 497–509. [CrossRef]
28. Hao, P.; Feng, S.J.; Li, Y.W.; Wang, B.; Chen, H.H. Adaptive infill sampling criterion for multi-fidelity gradient-enhanced kriging model. *Struct. Multidiscip. Optim.* **2020**, *62*, 353–373. [CrossRef]
29. Forrester, A.I.; Sóbester, A.; Keane, A.J. *Engineering Design via Surrogate Modelling: A Practical Guide*; Wiley: Chichester, UK, 2008; p. 89.
30. Schöbi, R.; Sudret, B.; Wiart, J. Polynomial-chaos-based Kriging for uncertainty quantification. *Int. J. Uncertain. Quantif.* **2015**, *5*, 171–193. [CrossRef]
31. Jamhiri, B.; Xu, Y.; Shadabfar, M.; Costa, S. Probabilistic machine learning for predicting desiccation cracks in clayey soils. *Bull. Eng. Geol. Environ.* **2023**, *82*, 355. [CrossRef]
32. Lataniotis, C.; Marelli, S.; Sudret, B. Extending classical surrogate modeling to high dimensions through supervised dimensionality reduction: A data-driven approach. *Int. J. Uncertain. Quantif.* **2020**, *10*, 55–82. [CrossRef]

33. Picheny, V.; Ginsbourger, D.; Richet, Y.; Caplin, G. Quantile-Based Optimization of Noisy Computer Experiments With Tunable Precision. *Technometrics* **2013**, *55*, 2–13. [CrossRef]
34. Le, R.K.; Pinaud, D.; Monestiez, P.; Chadoeuf, J.; Bretagnolle, V. Spatial leav-one-out cross-validation for variable selection in the presence of spatial autocorrelation. *Glob. Ecol. Biogeogr.* **2014**, *23*, 811–820.
35. Juergen, D.; Marcelo, C. Accounting for Spatial Autocorrelation in Algorithm-Driven Hedonic Models: A Spatial Cross-Validation Approach. *J. Real Estate Financ. Econ.* **2022**, *68*, 235–273.
36. Hugo, T.; Carlos, F.; Mathieu, D.; Philippe, L. Assessment of machine learning techniques for deterministic and probabilistic intra-hour solar forecasts. *Renew. Energy* **2018**, *123*, 191–203.
37. Zhang, Z.; Li, Z.; Song, Y. On ignoring the heterogeneity in spatial autocorrelation: Consequences and solutions. *Int. J. Geogr. Inf. Sci.* **2024**, *38*, 2545–2571. [CrossRef]
38. Hanna, M.; Edzer, P. Machine learning-based global maps of ecological variables and the challenge of assessing them. *Nat. Commun.* **2022**, *13*, 2208.
39. Hijmans, R.J. Cross-validation of species distribution models: Removing spatial sorting bias and calibration with a null model. *Ecology* **2012**, *93*, 679–688. [CrossRef]
40. Kennedy, C.M.; O'Hagan, A. Bayesian Calibration of Computer Models. *J. R. Stat. Soc. Ser. B (Stat. Methodol.)* **2001**, *63*, 425–464. [CrossRef]
41. Zhang, W.; Zhao, M.; Du, X.; Gao, Z.; Ni, P. Probabilistic machine learning approach for structural reliability analysis. *Probabilistic Eng. Mech.* **2023**, *74*, 103502. [CrossRef]
42. Krige, D.G. A statistical approach to some basic mine valuation problems on the Witwatersrand. *J. Chem. Metall. Min. Eng. Soc. S. Afr.* **1951**, *52*, 119–139.
43. Kleijnen, J.P.C. Kriging metamodeling in simulation: A review. *Eur. J. Oper. Res.* **2009**, *192*, 707–716. [CrossRef]
44. Pugliese, R.; Regondi, S.; Marini, R. Machine learning-based approach: Global trends, research directions, and regulatory standpoints. *Data Sci. Manag.* **2021**, *4*, 19–29. [CrossRef]
45. Cover, T.; Hart, P. Nearest neighbor pattern classification. *IEEE Trans. Inf. Theory* **1967**, *13*, 21–27. [CrossRef]
46. Altman, S. An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression. *Am. Stat.* **2012**, *46*, 175–185. [CrossRef]
47. Kane, P.B.; Broomell, S.B. Applications of the bias–variance decomposition to human forecasting. *J. Math. Psychol.* **2020**, *98*, 102417. [CrossRef]
48. Zhao, D.; Ma, M.; You, X. A Kriging-based adaptive parallel sampling approach with threshold value. *Struct. Multidiscip. Optim.* **2022**, *65*, 225. [CrossRef]
49. Liu, H.; Cai, J.; Ong, Y.-S. An adaptive sampling approach for Kriging metamodeling by maximizing expected prediction error. *Comput. Chem. Eng.* **2017**, *106*, 171–182. [CrossRef]
50. Li, Y.; Shi, J.; Yin, Z.; Shen, J.; Wu, Y.; Wang, S. An Improved High-Dimensional Kriging Surrogate Modeling Method through Principal Component Dimension Reduction. *Mathematics* **2021**, *9*, 1985. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.