Information Sciences 605 (2022) 86–98

Contents lists available at ScienceDirect

Information Sciences

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

Deep Residual Surrogate Model

Tianxin Huang [a], Yong Liu [a,b,c,*], Zaisheng Pan [a]

[a] Institute of Cyber-Systems and Control, Zhejiang University, Hangzhou, China
[b] State Key Laboratory of Industrial Control Technology, Zhejiang University, Hangzhou, China
[c] Huzhou Institute of Zhejiang University, China

ARTICLE INFO

Article history:
Received 11 April 2021
Received in revised form 15 April 2022
Accepted 23 April 2022
Available online 11 May 2022

Keywords:
First derivate validation
Multi-layer structure
Deep residual model

ABSTRACT

Surrogate models are widely used to model the high computational cost problems such as industrial simulation or engineering optimization when the size of sampled data for modeling is greatly limited. They can significantly improve the efficiency of complex calculations by modeling original expensive problems with simpler computation-saving functions. However, a single surrogate model cannot always perform well for various problems. On this occasion, hybrid surrogate models are created to improve the final performances on different problems by combining advantages of multiple single models. Nevertheless, existing hybrid methods work by estimating weights for all alternative single models, which limits the efficiency when more single models are adopted. In this paper, we propose a novel hybrid surrogate model quite different from former methods, named the Deep Residual Surrogate model (DRS). DRS does not merge all alternative single surrogate models directly by weights, but by assembling selected ones in a multiple layers structure. We propose first derivate validation (FDV) to recurrently select the single surrogate model adopted in each layer from all candidates. Experimental results on multiple benchmark problems demonstrate that DRS has better performances than existing single and hybrid surrogate models in both prediction accuracy and stability with higher efficiency.

1. Introduction

With the rapid development of the computation ability of devices, many actual problems can be modeled and solved directly. However, some computation expensive problems such as the engineering optimizations [39,38,50,31,5,36], require for multiple complicated calculations to get the optimal solution, which suffer from great computational cost. Under this circumstance, surrogate models are proposed to model actual engineering problems with a small amount of sampled data. Computation expensive problems can be approximated with low resource-consumed surrogate models. Then an optimal solution for the original complex problem can be solved with an optimization strategy such as Evolutionary Algorithm [40,3,44,7,30] based on corresponding surrogate models, which can greatly improve the optimization efficiency.

Since modeling engineering problems more accurately can improve the optimal results, Many works have been proposed to improve the prediction accuracy of surrogate models under the limited sampled data. Single surrogate models have been proposed such as Kriging [15,34,11] and RBF [17,20,18]. As every single surrogate model is only suitable for a few kinds of problems, later works combine multiple models together in order to improve the overall performances on multiple different

[*] Corresponding author at: State Key Laboratory of Industrial Control and Technology, Zhejiang University, Room 221, Institute of Cyber-System and Control, Zhejiang University, Hangzhou 310027, China.

E-mail address: yongliu@iipc.zju.edu.cn (Y. Liu).

https://doi.org/10.1016/j.ins.2022.04.041
0020-0255/© 2022 Elsevier Inc. All rights reserved.

problems, known as the hybrid surrogate models [46,2,43,1,48,37]. Most Hybrid surrogate models estimate weights for provided alternative models and use all of them during prediction, which may take much longer time than single surrogate models.

In this work, we propose a new hybrid algorithm named the Deep Residual Surrogate model (DRS), which is composed of multiple single models organized in a multi-layer structure. In DRS, we propose an algorithm named First Derivate Validation (FDV) to evaluate and choose a single surrogate model for each layer from all provided candidates. The subsequent layer is responsible for predicting the residual of the previous layer. Finally, the chosen models will be overlaid together to construct the complete DRS model. Experimental results confirm that DRS can outperform former single surrogate models, also much faster than existing hybrid surrogate models. The remainder of the paper is organized as follows: Section 2 reviews existing surrogate modeling techniques; Section 3 presents our methodology of FDV and DRS; Section 4 shows all relevant experiments; Section 5 summarizes the whole work and discuss the future works where DRS may be further applied.

Our contribution can be summarized as:

1. We propose a new algorithm named First Derivate Validation (FDV) to measure the error of single surrogate models by considering the error gain after removing vital points with lower gradients;
2. We propose a new hybrid algorithm named Deep Residual Surrogate model (DRS), which chooses and assembles single models in a multi-layer structure based on FDV instead of combining them by weights like other methods;
3. Our experiments on multiple benchmark problems confirm that our proposed FDV can evaluate the model error more accurately than other validation methods, while DRS can outperform existing surrogate models in both accuracy and stability.

2. Related Works

2.1. Single Surrogate Model

Many surrogate models have been developed to solve computation expensive problems. Common methods include: Polynomial Response Surface Model (PRSM) [29], Kriging [15,11], Radial Basis Function (RBF) [17,18], Extended Radial Basis Function(E-RBF) [27,28,49], Artificial Neural Network (ANN) [13,45], and Support Vector Regression (SVR) [10,41,42,4]. PRSM is one of the most widely used surrogate models, suitable to capture the global trend of training data. Parameters of PRSM are determined by the data dimension and polynomial order, which will not increase as the data size grows. Generally speaking, the quadratic polynomial model (QP), also known as the second order polynomial model is the most popular among PRSM models of different orders. However, PRSM ignores local shapes of target curves, which dramatically limits its adoption. Kriging models data by the combination of response model and stochastic process, which has shown great performance in many conditions. RBF presents data by the linear combination of radial basis kernel functions based on Euclidean distance between points. E-RBF combines RBF with N-RBF [33], which adopts kernel functions not based on radial distances but based on distances between coordinates dimensions. ANN and SVR are typical machine learning methods which model data with sufficient parameters and get optimized with multiple iterations. Machine learning methods can achieve good performances with complex settings of hyper-parameters and relatively large data. But it may be challenging to find the correct settings for them. To select the single surrogate model for a specific optimized problem, COSMOS [25] and AutoSM [19] introduce the Evolutionary Algorithm (EA) to search for the most appropriate one from all candidates, while they have relatively high computational costs due to the adoption of EA.

2.2. Hybrid Surrogate Model

Except the commonly used single surrogate models, hybrid surrogate models are proposed to combine the advantages of different single models, such as [46,2,43,1]. They usually work by estimating weights for multiple single surrogate models with designed algorithms. In related researches [43,2], the cross-validation error is often adopted to estimate the accuracy of single models and calculate corresponding weights. Some works [2,43] give constant weights for single models and optimize them by minimizing validation errors over the whole design space, known as global measure methods. These works can perform better than single surrogate models. However, they are often doubted for the local accuracy of each single model. On this condition, some local measure methods [35,1,23] distribute weights for single models point-by-point, which means that weights for different points on the same single model could be quite different. However, they may instead ignore the global trend of models and suffer from high computational costs. To integrate advantages of global and local measure methods, some recent works [48,8,47] hybrid both measures to estimate the weights.

AHF [48] proposes a Crowding Distance-based Trust Region (CD-TR) algorithm to estimate the errors of single models. This algorithm estimates upper and lower boundary functions by a base model to measure the global trend and sampling density around each training point to measure local distributions. Boundary functions are used together to estimate model errors and weights. The performance of AHF is greatly influenced by the chosen base model. ES-HGL [8] combines the global and local measure methods by dividing design space into two regions, one for the region far from sampled points, the other for the region near sampled points. Global constant and local weights are calculated and adopted in the outer and inner regions, respectively. UES [47] adopts a similar idea with ES-HGL to hold global constant weights and local weights in different regions. But UES has a more fuzzy boundary because it designs a function to further estimate factors for global and local weights according to the distance from sampled points, instead of natural selection as ES-HGL.

3. Methodology

3.1. First Derivative Validation

In most applications, we compare the robustness and performances of different surrogate models with cross-validation methods such as the commonly used Holdout cross-validation and K-fold cross-validation. Holdout cross-validation calculates validation errors on the validation data randomly divided from the training data, which has apparent uncertainty due to the randomness. K-fold cross-validation reduces the uncertainty by evaluating the validation errors through the average value of multiple non-coincident training and validation data groups. However, K-fold cross-validation has high time and memory costs due to multiple circulations to acquire the validation error. Besides, the final validation error is the average result of multiple experiments. More valuable results may be covered by less valuable ones in this process, which may mislead the evaluation.

First derivative validation (FDV) aims to estimate the performances of different single surrogate models and choose the best one for subsequent processing. For FDV, we assume that an appropriate surrogate model should be suitable for the data no matter how the sampled points change. In other words, an appropriate model should suffer from a smaller error gain after removing some of the training points, where the error gain denotes how much the prediction errors on all training points increase.

Points with smaller first derivatives are often close to vertexes of curves or surfaces, which play more critical roles in describing shapes. In this condition, by removing some points with smaller first derivatives and training models on the remained points, more general validation errors can be evaluated by the error gain on all training points. However, evaluating performances purely by error gains after removing points with smaller first derivatives from training points may be misleading sometimes. Simple models, e.g., quadratic polynomial model, often enjoy lower error gains when data are changed. But they often have higher training errors on training points. So, adding a constraint for training error is necessary. The final validation error is the combination of FDV error gain and training error. The algorithm is presented as Algorithm 1, where the points removing ratio β and balance ratio α denote the percentage of removed training points and the weight for error gain.

```
Algorithm 1: First Derivative Validation
Input: Training points $x = (x_1, x_2, x_3, \cdots, x_N)$,
Target values $y = (y_1, y_2, y_3, \cdots, y_N)$,
Target model $f$, points removing ratio $\beta$, balance ratio $\alpha$
Output: Evaluated error of target model $evaerr$
Train target model with training points and target values:
$f = solve(x, y, f)$
Evaluate training error on training points:
$err = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(f(x_i) - y_i)^2}$
Get $k$ points with smaller first derivates:
$ids = topk(-\nabla_x f(x)), k = \beta \cdot len(x)$
Remove $k$ points with smaller first derivates:
$x\prime = x_{id \notin ids}, y\prime = y_{id \notin ids}$
Train model with $x\prime$ and $y\prime$:
$f\prime = solve(x\prime, y\prime, f)$
Evaluate the training error of $f\prime$ on all training points:
$err\prime = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(f\prime(x_i) - y_i)^2}$
The final error:
$evaerr = \alpha \cdot (err\prime - err) + err = \alpha \cdot err\prime + (1 - \alpha) \cdot err$
```

3.2. Deep Residual Surrogate Model

Though many hybrid surrogate models have been proposed to overcome the limitations of single surrogate models, they often calculate the final output by the weighted sum of all alternative models. In our work, we proposed a framework named the Deep Residual Surrogate model (DRS) which is quite different from former hybrid methods. Single surrogate models are recurrently chosen and overlaid together to make up the more effective model. In this way, the final hybrid model is composed of several chosen models in multiple layers instead of all models, greatly improving inference efficiency. In this work, we choose 5 single surrogate models as candidates, including quadratic polynomial model (QP) [29], Kriging model [11], SVR [42], RBF and RBF_MQ model [17]. For convenience, we name the set including all alternative single surrogate models as the model pool in follow-up discussions.

DRS is organized in a multi-layer structure. In each layer, a single surrogate model is chosen from the model pool with FDV presented in Section 1. Corresponding model parameters are saved, and models in subsequent layers will refine the current prediction. The final result of DRS would be the sum of models in multiple layers. To ensure the training process can pass through multiple layers, we define an equalization operation for the prediction of single-layer model:

$$\begin{aligned} pred &= \frac{pred}{layernum}, \\ y &= y - pred, \end{aligned} \tag{1}$$

where *pred* and *layernum* are the prediction of surrogate model in the former layer and the number of layers in DRS. As illustrated in Eq. 1, the target value for the next layer will be replaced by the residual between the current target value and the predicted result. The equalization operation is used to scale the predicted result to prevent the residual from being too small to train the model in the next layer. In this way, the model in the next layer will learn to make up for weaknesses of the model in the former layer and improve the final performance.

Note that the multi-layer structure in DRS is adopted to find the best combination of single models, where the selected model in each layer could be quite different. The latter model is determined according to the performance of the former model, which is different from direct distributed weights of $1/layernum$. For instance, if we choose a relatively smooth single model in the first layer which can fit the global shapes well, then a varied model is tended to be chosen in the second layer to describe the local shapes. More intuitive confirmation by experiments could be found in Section 4.5. The details of solving and prediction with DRS are shown in Algorithm 2 and Algorithm 3, where β is 1/7, α is 1, and layernum of DRS is set as 2. In other words, two single models would be chosen from the model pool to calculate the final output.

Algorithm 2: The Solving process of Deep Residual Surrogate Model

```
Input: Training points $x = (x_1, x_2, x_3, \cdots, x_N)$,
Target values $y = (y_1, y_2, y_3, \cdots, y_N)$, model pool $P$,
Target model $f$, points removing ratio $\beta$, balance ratio $\alpha$,
The number of layers $layernum$,
Output: Trained model list $Model\_list$
Initialize intermediate residual value:
$y_{inter} = y$
while $len(Model\_list) < layernum$ do
    $Errlist \rightarrow []$
    for each model p in P do
        $evaerr = FDV(x, y_{inter}, p, \alpha, \beta)$
        $Errlist.insert(evaerr)$
    end for
    $Model = solve(x, y, P[min(Errlist)])$
    $y_{inter} = y_{inter} - \frac{Model.predict(x)}{layernum}$
    $Model\_list.insert(Model)$
end while
```

Algorithm 3: The Prediction process of Deep Residual Surrogate Model

```
Input: Target points $\hat{x} = (\hat{x_1}, \hat{x_2}, \hat{x_3}, \cdots, \hat{x_N})$,
Trained model list $Model\_list$
Output: Predicted values $\hat{y} = (\hat{y_1}, \hat{y_2}, \hat{y_3}, \cdots, \hat{y_N})$
Initialize predicted value $\hat{y} : \hat{y} = 0$
for each model m in $Model\_list$ do
    if m is not the last in $Model\_list$ then
        $\hat{y} = \hat{y} + \frac{m.predict(\hat{x})}{len(Model\_list)}$
    else
        $\hat{y} = \hat{y} + m.predict(\hat{x})$
    end if
end for
```

Experiments in Section 4.6 show that DRS outperforms existing surrogate models in both accuracy and stability. Besides, the time efficiency of DRS is also higher other hybrid algorithms, which confirms that DRS is an effective and efficient algorithm.

4. Experiments and Discussion

4.1. Benchmark Problems

To evaluate the performance of DRS on multiple conditions, we choose 27 commonly used benchmark problems [16,26,12,27,6,26,14,37] from 1 to 8 dimensions. The specific definitions of them are demonstrated in Fig. 5. Latin hypercube sampling (LHS) [24] is used to sample both training and testing points on benchmark problems. The process of LHS is presented in Fig. 1. In each dimension, intervals are uniformly divided between the lower and upper bounds of benchmark problems according to the number of sampled points. The intervals are then shuffled separately in each dimension. Data points are randomly sampled among the intervals. We sample 36 training points and 100 test points for each benchmark problem in this work.

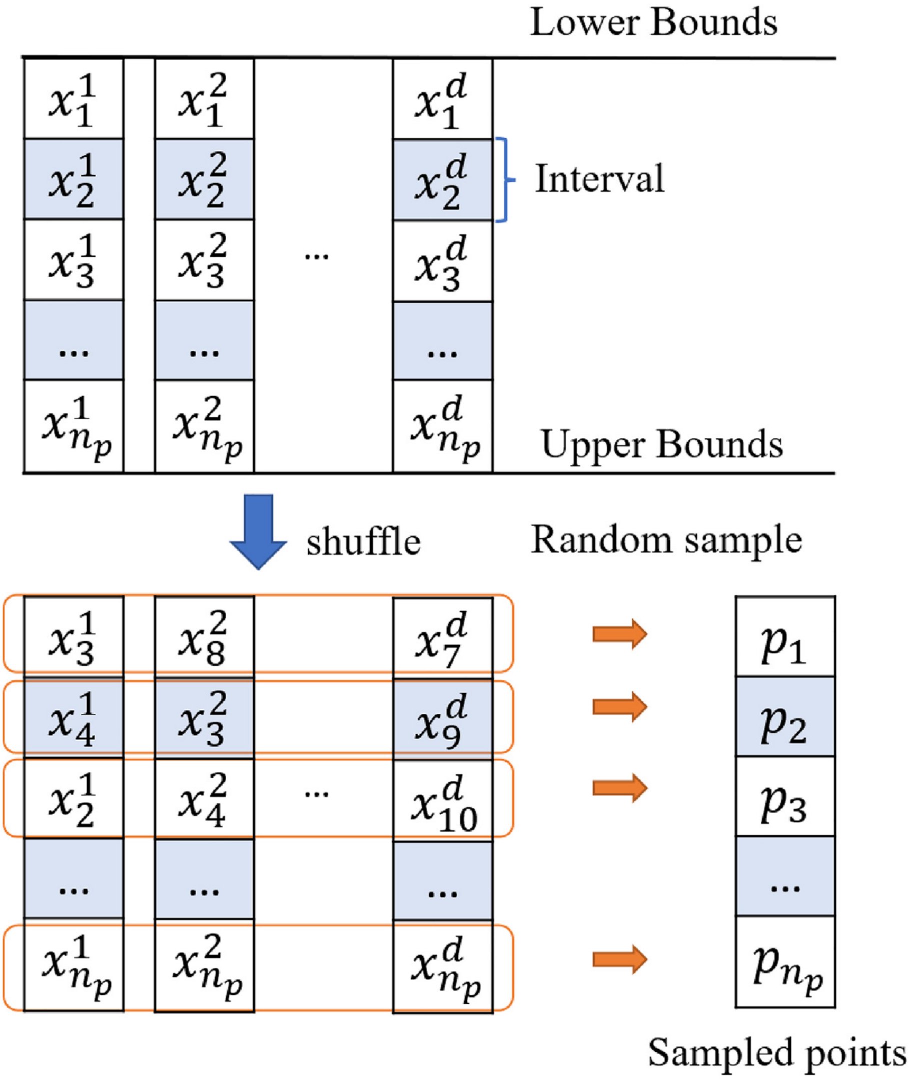


Fig. 1. The process of Latin hypercube sampling(LHS). n_p and d are the number of sampled points and data dimensions, respectively. $x_{n_p}^d$ means the $n_p th$ interval in dth dimension.

4.2. Metrics

1. **RMSE** (Root Mean Squared Error) is the most common metric for regression problems, which evaluates the average distance between the predicted outputs and target values. It can be presented as

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - f(x_i)^2)}, \tag{2}$$

   where n denotes the number of testing points chosen for the evaluation, y_i and $f(x_i)$ denote the true value and predicted value for testing point x_i, respectively. Lower RMSE means the algorithm is more accurate.
   To avoid the influence of various output ranges in different benchmark problems, we also introduce the **NRMSE** (Normalized Root Mean Squared Error) presented as

$$NRMSE = \sqrt{\frac{\sum_{i=1}^{n}(y_i - f(x_i)^2}{\sum_{i=1}^{n}y_i^2}}, \tag{3}$$

   where corresponding definitions of n, y_i and $f(x_i)$ are the same as Eq. 2. In this work, we adopt both RMSE and NRMSE to evaluate the errors.
2. **Hit Rate** is a metric proposed to better compare the performances between different surrogate models. It evaluates the probability when a surrogate model can work best, which can be defined as

$$Hit\ Rate = \frac{1}{N}\sum_{i=1}^{N}F(E_f(p_i), \min_f E_f(p_i)), \tag{4}$$

$$F(x,y) = \begin{cases} 0 & \|x-y\| \geqslant \delta \\ 1 & \|x-y\| \leqslant \delta, \end{cases} \tag{5}$$

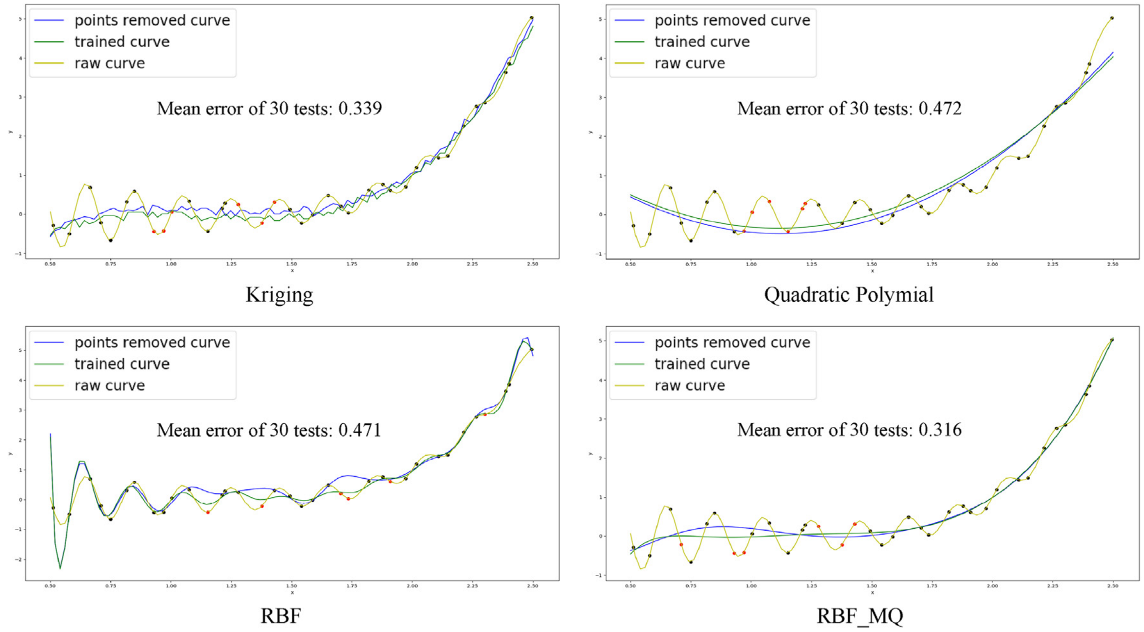
   where N is the number of benchmark problems. δ is a tiny value as threshold to judge if two values are equal. $F(x,y)$ is the count function used to count the number of times that $E_f(p_i)$ achieves the best, while $E_f(p_i)$ denotes the metric value E of surrogate model f on *ith* benchmark problem p_i.
   Note that Hit Rate is a general concept, which means that an arbitrary metric can be evaluated under Hit Rate. For example, RMSE can be applied to calculate RMSE Hit Rate, which can evaluate the probability of getting minimum RMSE. Hit Rate pays more attention to the relative performances between models. In other words, It is more concerned about which model tends to perform better than other models.
3. **Time cost** is adopted to evaluate the efficiency as a supplementary metric in this work, which is defined as the execution time for a specific operation. For the evaluation of FDV, it is defined as time taken by different validation operations. As for the evaluation of DRS, it is defined as time consumed by predicting the final output. Less time cost means that the algorithm can work faster.
4. **RMSE Variance** is used to describe the model stability during multiple experiments, also reflecting the robustness against the changing of initial sampled points. It is defined as the error variance during multiple experiments, which can be presented as

$$RMSE\ Variance = \frac{1}{N}\sum_{i=1}^{N}(RMSE_i - \frac{1}{N}\sum_{i=1}^{N}RMSE_i)^2, \tag{6}$$

   where N is the number of experiments, $RMSE_i$ denotes $RMSE$ of *ith* experiment. Smaller RMSE Variance means the algorithm gets relatively close errors during multiple experiments, which confirms it works steadier.

4.3. Visualization of the Points Selection Rule in FDV

In this section, We take GRAMACY & LEE function [16] to show the regularity of FDV in Fig. 2. The red points are points with lower first derivatives to be removed, while the black ones are those that remained. We can see that Kriging and RBF tend to change a lot after removing FDV points, which will lead to large error gains. Though QP has relatively small changing and error gain after removing points, it has a too big deviation from the raw curve. RBF_MQ is more robust to the removement of points with also slight deviation, which is finally chosen as the best model by FDV. According to the errors evaluated on testing points, the chosen RBF_MQ has the most minor error of 0.316 among all candidates, which means FDV can really help choose the most accurate model.



Kriging

Quadratic Polymial

RBF

RBF_MQ

Fig. 2. The visualization of FDV regularity.

4.4. Comparisons between FDV and other Validation Methods

In this section, we compare FDV with two commonly used validation methods: K-fold cross-validation and Holdout cross-validation. K-fold cross-validation with 1,3,5,7 folds and Holdout cross-validation with 1/3, 1/5, 1/7, 1/9 selected validation points are adopted for comparisons in this work.

Validation methods are often adopted to compare the performances of different surrogate models and choose the best one. To observe the effectiveness of our validation strategy, we compare the effects of single models chosen through different validation strategies from the model pool including QP [29], Kriging [11], SVR [42], RBF and RBF_MQ [17]. We use (1) Time cost, (2) RMSE, (3) NRMSE, (4) RMSE Variance to evaluate the efficiency and performances. Besides, to further confirm if FDV has effectively measured the actual error, we calculate the predicted RMSE and NRMSE between the validation error calculated on training points and the actual error acquired on testing points.

The experimental results are presented in Table 1. From the result, we can see that models chosen by FDV have both the smallest errors and variances. It confirms that FDV can choose the best surrogate model more accurately and steadier than other methods. Besides, the time cost of FDV is close to Holdout cross-validation, which is much faster than K-fold cross-validation. We can also see that FDV gets the smallest predicted RMSE and NRMSE. It means the validation error calculated with FDV is much closer to the actual error than other validation methods, which confirms that FDV is an effective validation method.

To verify that FDV gets the most effective model for every benchmark problem from a given model pool instead of choosing a certain model which always performs well in all conditions, we examine our method on different model pools. We gradually remove the best performance model from the model pool to get four different model pools.

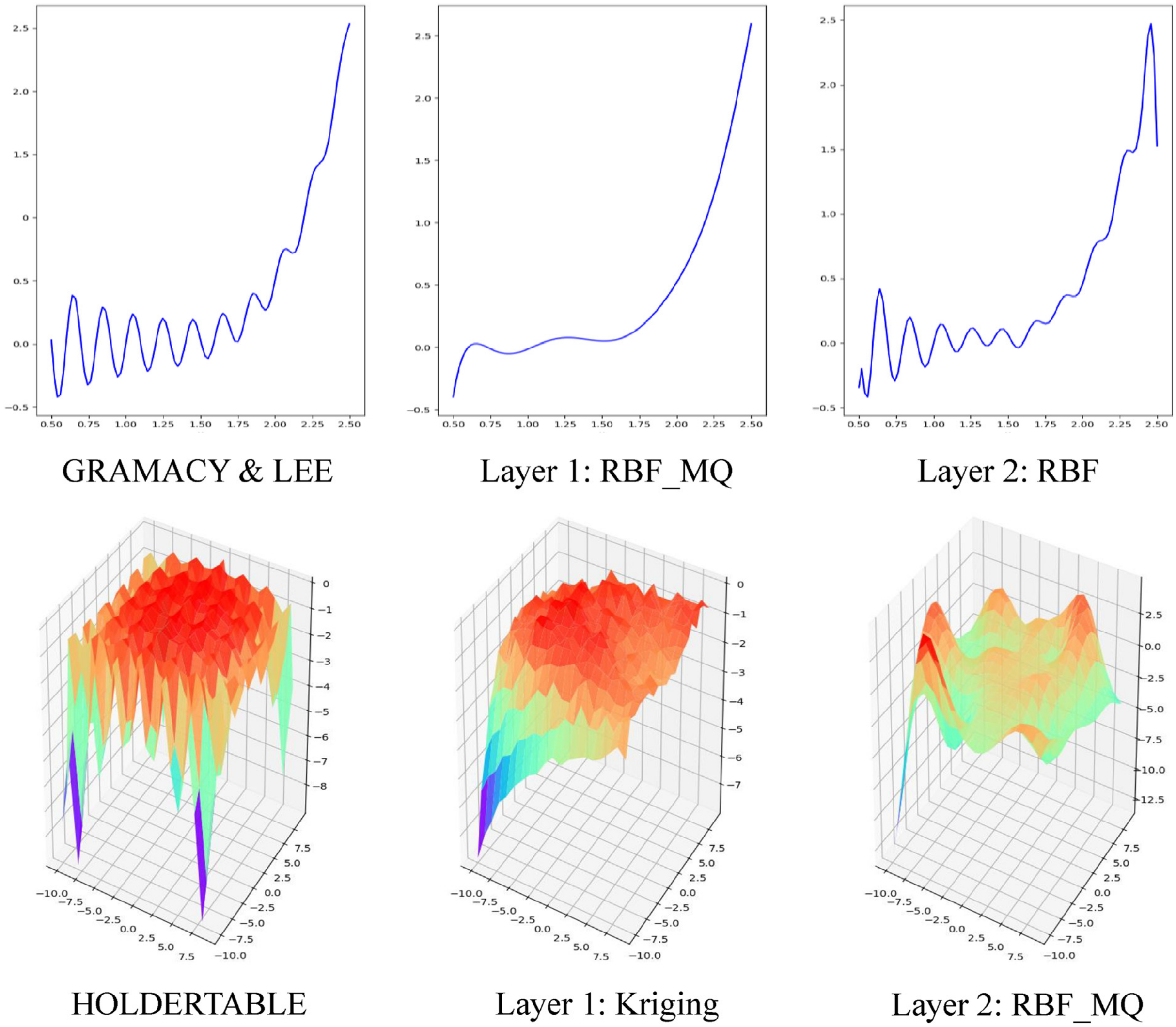
Table 1
The comparison of FDV with other validation methods. K-fold and Holdout denote K-fold cross-validation and Holdout cross-validation, respectively. Rate denotes the number of folds for K-fold cross-validation and the percentage of validation points for Holdout cross-validation.

	Rate	Time(ms)	NRMSE(10^{-2})	RMSE(10^3)	RMSE Variance(10^8)	Predicted RMSE(10^3)	Predicted NRMSE
K-fold	3	5.172	8.834	10.515	5.683	10.816	0.299
	5	8.959	9.353	8.662	2.835	8.232	0.473
	7	12.889	9.126	9.084	2.213	9.570	0.477
	9	17.506	10.815	11.155	5.192	7.512	0.567
Holdout	1/3	1.733	8.528	15.018	9.856	11.208	0.290
	1/5	1.823	9.667	11.411	3.831	11.019	0.187
	1/7	1.888	7.459	12.861	10.002	11.723	0.365
	1/9	1.874	7.525	12.474	8.540	13.473	0.230
FDV		**1.892**	**6.953**	**7.781**	**1.576**	**7.045**	**0.106**

The results are presented in Table 2. We can see that models chosen by FDV can always reach smaller errors and variances than other methods, even when the model pool changes. FDV always works much faster than K-fold cross-validation, though slightly slower than Holdout cross-validation, which divides data randomly. Chosen models under all combinations of candidate models in the model pool have relatively big variances because the training points sampled by LHS [24] would be

Table 2
The Comparison of single surrogate models chosen with validation methods among different alternative model pools.

Model Pool	Validation	Time(ms)	NRMSE(10^{-2})	RMSE(10^3)	RMSE Variance (10^8)
Kriging + RBF_MQ + RBF + QP + SVR	K-fold	12.889	9.126	9.084	2.213
	Holdout	**1.888**	7.459	12.861	10.002
	FDV	1.892	**6.953**	**7.781**	**1.576**
RBF_MQ + RBF + QP + SVR	K-fold	4.455	11.385	8.985	2.4963
	Holdout	**0.666**	10.098	15.643	11.657
	FDV	0.672	**9.515**	**7.202**	**0.546**
RBF + QP + SVR	K-fold	4.033	30.165	19.004	4.863
	Holdout	**0.604**	**27.045**	20.743	5.536
	FDV	0.608	27.515	**18.007**	**3.873**
QP + SVR	K-fold	2.889	31.748	24.652	4.97
	Holdout	0.440	27.829	29.059	8.295
	FDV	**0.438**	**24.694**	**24.071**	**3.669**



GRAMACY & LEE

Layer 1: RBF_MQ

Layer 2: RBF

HOLDERTABLE

Layer 1: Kriging

Layer 2: RBF_MQ

Fig. 3. The Visualization of multiple layers in DRS.

quite different in multiple experiments. FDV has the smallest variances for all model pools, which confirms that it is the most robust to sampled points. It can choose the best single surrogate model steadier when the training points change in multiple experiments.

4.5. Visualization of Chosen Models in DRS

As we present in Section 3.2, DRS recurrently selects and overlies different single surrogate models in multiple layers to achieve good performances. In this section, we choose a 1-dimensional test function GRAMACY & LEE [16] and a 2-

Table 3
The Comparison of hybrid surrogate methods under different model pools.

Model Pool	Algorithm	Time(ms)	NRMSE(10^{-2})	RMSE(10^3)	RMSE Variance (10^8)
Kriging + RBF_MQ + RBF + QP + SVR	AHF	0.845	16.485	16.294	3.206
	ES-HGL	1.063	21.334	11.869	1.695
	UES	17.677	22.174	11.951	1.814
	DRS	**0.275**	**8.530**	**8.058**	**1.579**
RBF_MQ + RBF + QP + SVR	AHF	0.566	19.612	17.255	5.241
	ES-HGL	0.786	26.420	13.667	3.265
	UES	26.826	26.994	15.110	4.581
	DRS	**0.125**	**10.447**	**7.915**	**2.243**
RBF + QP + SVR	AHF	0.478	**24.468**	23.668	**6.441**
	ES-HGL	0.747	36.529	23.292	8.140
	UES	39.419	36.222	23.328	8.137
	DRS	**0.126**	27.777	**22.929**	8.491
QP + SVR	AHF	0.402	24.975	24.524	4.462
	ES-HGL	0.695	25.109	23.986	4.115
	UES	8.685	**23.816**	23.318	4.190
	DRS	**0.124**	24.361	**23.183**	**3.176**

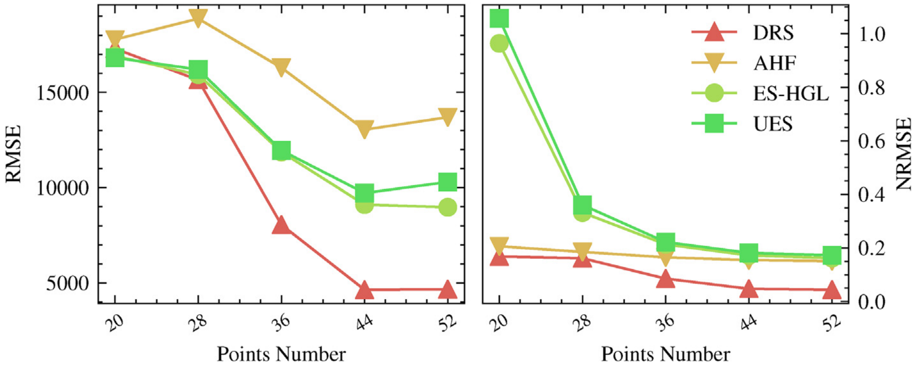
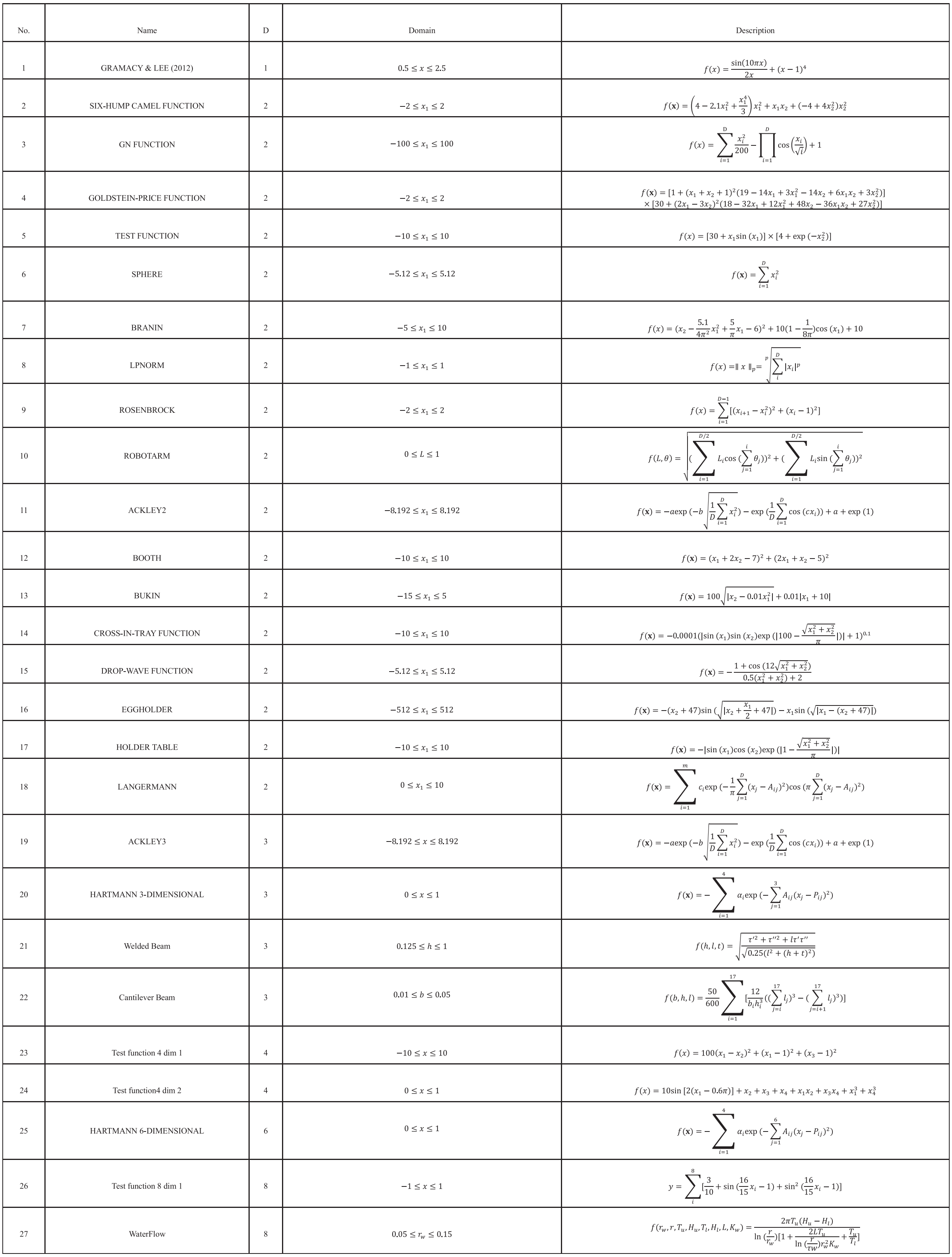


Fig. 4. The Comparison of hybrid models under different points numbers.

Table 4
The Comparison of all surrogate methods. ANN-1 $\sim$ ANN-7 denote the Artificial Neural Networks with 1 $\sim$ 7 hidden layers.

Methods	Time(ms)	NRMSE	RMSE(10^3)	RMSE Variance(10^7)	NRMSE Hit rate	RMSE Hit rate	Variance Hit Rate
AHF	0.930	0.228	9.417	15.043	0.185	0.148	0.111
ES-HGL	1.161	0.358	9.602	7.053	0.037	0.037	0
UES	16.458	0.375	9.86	7.525	0	0	0
QP	**0.068**	0.271	12.524	10.661	0.074	0.074	0.111
Kriging	0.272	0.227	4.317	8.108	0.148	0.148	0.222
RBF_MQ	0.074	0.465	9.752	6.648	0.111	0.111	0.111
RBF	0.071	0.457	17.146	18.592	0.037	0.037	0.037
SVR	0.106	0.544	21.794	19.017	0.037	0.074	0.111
ANN-1	0.858	2.055	80.965	395.849	0	0	0
ANN-3	2.094	2.081	83.519	539.706	0	0	0
ANN-5	3.176	2.051	83.185	689.056	0	0	0
ANN-7	4.279	2.099	82.702	802.780	0	0	0
DRS	0.312	**0.211**	**3.316**	**5.911**	**0.37**	**0.37**	**0.296**



No.	Name	D	Domain	Description
1	GRAMACY & LEE (2012)	1	$0.5 \le x \le 2.5$	$f(x) = \frac{\sin(10\pi x)}{2x} + (x-1)^4$
2	SIX-HUMP CAMEL FUNCTION	2	$-2 \le x_1 \le 2$	$f(\mathbf{x}) = \left(4 - 2.1x_1^2 + \frac{x_1^4}{3}\right)x_1^2 + x_1x_2 + (-4 + 4x_2^2)x_2^2$
3	GN FUNCTION	2	$-100 \le x_1 \le 100$	$f(x) = \sum_{i=1}^{\mathrm{D}} \frac{x_i^2}{200} - \prod_{i=1}^{D} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$
4	GOLDSTEIN-PRICE FUNCTION	2	$-2 \le x_1 \le 2$	$f(\mathbf{x}) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)]$ $\times [30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$
5	TEST FUNCTION	2	$-10 \le x_1 \le 10$	$f(x) = [30 + x_1\sin(x_1)] \times [4 + \exp(-x_2^2)]$
6	SPHERE	2	$-5.12 \le x_1 \le 5.12$	$f(\mathbf{x}) = \sum_{i=1}^{D} x_i^2$
7	BRANIN	2	$-5 \le x_1 \le 10$	$f(x) = (x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6)^2 + 10(1 - \frac{1}{8\pi})\cos(x_1) + 10$
8	LPNORM	2	$-1 \le x_1 \le 1$	$f(x) = \lVert x \rVert_p = \sqrt[p]{\sum_{i}^{D} \lvert x_i \rvert^p}$
9	ROSENBROCK	2	$-2 \le x_1 \le 2$	$f(x) = \sum_{i=1}^{D-1} [(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$
10	ROBOTARM	2	$0 \le L \le 1$	$f(L, \theta) = \sqrt{(\sum_{i=1}^{D/2} L_i\cos(\sum_{j=1}^{i} \theta_j))^2 + (\sum_{i=1}^{D/2} L_i\sin(\sum_{j=1}^{i} \theta_j))^2}$
11	ACKLEY2	2	$-8.192 \le x_1 \le 8.192$	$f(\mathbf{x}) = -a\exp(-b\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}) - \exp(\frac{1}{D}\sum_{i=1}^{D} \cos(cx_i)) + a + \exp(1)$
12	BOOTH	2	$-10 \le x_1 \le 10$	$f(\mathbf{x}) = (x_1 + 2x_2 - 7)^2 + (2x_1 + x_2 - 5)^2$
13	BUKIN	2	$-15 \le x_1 \le 5$	$f(\mathbf{x}) = 100\sqrt{\lvert x_2 - 0.01x_1^2 \rvert} + 0.01\lvert x_1 + 10 \rvert$
14	CROSS-IN-TRAY FUNCTION	2	$-10 \le x_1 \le 10$	$f(\mathbf{x}) = -0.0001(\lvert\sin(x_1)\sin(x_2)\exp(\lvert 100 - \frac{\sqrt{x_1^2 + x_2^2}}{\pi}\rvert)\rvert + 1)^{0.1}$
15	DROP-WAVE FUNCTION	2	$-5.12 \le x_1 \le 5.12$	$f(\mathbf{x}) = -\frac{1 + \cos(12\sqrt{x_1^2 + x_2^2})}{0.5(x_1^2 + x_2^2) + 2}$
16	EGGHOLDER	2	$-512 \le x_1 \le 512$	$f(\mathbf{x}) = -(x_2 + 47)\sin(\sqrt{\lvert x_2 + \frac{x_1}{2} + 47 \rvert}) - x_1\sin(\sqrt{\lvert x_1 - (x_2 + 47) \rvert})$
17	HOLDER TABLE	2	$-10 \le x_1 \le 10$	$f(\mathbf{x}) = -\lvert\sin(x_1)\cos(x_2)\exp(\lvert 1 - \frac{\sqrt{x_1^2 + x_2^2}}{\pi}\rvert)\rvert$
18	LANGERMANN	2	$0 \le x_1 \le 10$	$f(\mathbf{x}) = \sum_{i=1}^{m} c_i\exp(-\frac{1}{\pi}\sum_{j=1}^{D}(x_j - A_{ij})^2)\cos(\pi\sum_{j=1}^{D}(x_j - A_{ij})^2)$
19	ACKLEY3	3	$-8.192 \le x \le 8.192$	$f(\mathbf{x}) = -a\exp(-b\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}) - \exp(\frac{1}{D}\sum_{i=1}^{D} \cos(cx_i)) + a + \exp(1)$
20	HARTMANN 3-DIMENSIONAL	3	$0 \le x \le 1$	$f(\mathbf{x}) = -\sum_{i=1}^{4} \alpha_i\exp(-\sum_{j=1}^{3} A_{ij}(x_j - P_{ij})^2)$
21	Welded Beam	3	$0.125 \le h \le 1$	$f(h, l, t) = \sqrt{\frac{\tau'^2 + \tau''^2 + l\tau'\tau''}{\sqrt{0.25(l^2 + (h + t)^2)}}}$
22	Cantilever Beam	3	$0.01 \le b \le 0.05$	$f(b, h, l) = \frac{50}{600}\sum_{i=1}^{17} [\frac{12}{b_ih_i^3}((\sum_{j=i}^{17} l_j)^3 - (\sum_{j=i+1}^{17} l_j)^3)]$
23	Test function 4 dim 1	4	$-10 \le x \le 10$	$f(x) = 100(x_1 - x_2)^2 + (x_1 - 1)^2 + (x_3 - 1)^2$
24	Test function4 dim 2	4	$0 \le x \le 1$	$f(x) = 10\sin[2(x_1 - 0.6\pi)] + x_2 + x_3 + x_4 + x_1x_2 + x_3x_4 + x_1^3 + x_4^3$
25	HARTMANN 6-DIMENSIONAL	6	$0 \le x \le 1$	$f(\mathbf{x}) = -\sum_{i=1}^{4} \alpha_i\exp(-\sum_{j=1}^{6} A_{ij}(x_j - P_{ij})^2)$
26	Test function 8 dim 1	8	$-1 \le x \le 1$	$y = \sum_{i}^{8} [\frac{3}{10} + \sin(\frac{16}{15}x_i - 1) + \sin^2(\frac{16}{15}x_i - 1)]$
27	WaterFlow	8	$0.05 \le r_w \le 0.15$	$f(r_w, r, T_u, H_u, T_l, H_l, L, K_w) = \frac{2\pi T_u(H_u - H_l)}{\ln(\frac{r}{r_w})[1 + \frac{2LT_u}{\ln(\frac{r}{\tau w})r_w^2K_w} + \frac{T_u}{T_l}]}$

Fig. 5. All adopted benchmark problems.

dimensional test function HOLDERTABLE [14] as examples to observe how DRS chooses and combines single surrogate models. We visualize the curves of single surrogate models chosen in different layers of DRS in Fig. 3. The first column shows the curves of benchmark problems, while the second and third columns show the curves predicted by models chosen in the first and second layers, respectively.

We can see that DRS actually works by breaking down the original complicated function into multiple simpler functions. It tends to choose a relatively smooth model in the first layer to fit the overall shape and a varied model in the second layer to make up local details. Note that the model pool including multiple alternative single surrogate models is not changed during the iteration of DRS. It means choosing the same model repeatedly is allowed here, in which condition DRS actually chooses only one single model because it is judged to work better than the combination of multiple models under this group of sampled points.

4.6. Comparisons between DRS and other Surrogate Models

To prove that DRS is robust to different combinations of candidate models, we compare DRS with other hybrid surrogate models under changing model pools on test functions 1 to 10. The results are presented in Table 3, all of them are the average values of 30 repeated experiments. We can see that DRS outperforms other hybrid algorithms on Time cost, RMSE, NRMSE and variance in most conditions, which demonstrates that it is faster, steadier and more accurate than other hybrid methods.

Considering that the number of sampled points may have a great impact on the final performances of models, we change the number of sampled training points between $20 \sim 52$. All single surrogate models, including QP [29], Kriging [11], SVR [42], RBF and RBF_MQ model [17], are involved in the model pool. The results are given in Fig. 4. We can see that errors of all models tend to increase when the number of sampled points decreases. DRS still outperforms other hybrid models under different point numbers. Finally, to give a more comprehensive comparison evaluation for DRS, we compare it with all single and hybrid surrogate models on benchmark problems. Results are shown in Table 4. The Artificial Neural Networks with 1,3,5,7 hidden layers are evaluated, while we adopt 64 neurons in each hidden layer. We can see that DRS can achieve smaller errors and variances, while getting the highest Hit rates. It means DRS outperforms other surrogate models in accuracy and stability. Though DRS is slower than single models due to the combination process of multiple single models, it is still much faster than other hybrid surrogate algorithms.

5. Conclusion and Future Work

In this paper, we propose a novel hybrid surrogate model named the Deep Residual Surrogate model (DRS). In DRS, single surrogate models are not combined directly by weights like other hybrid algorithms. They are evaluated by proposed First Derivate Validation (FDV) and chosen from the model pool according to acquired validation errors, which are recurrently trained and overlaid together to predict the final output. In our work, we include QP, Kriging, RBF, RBF_MQ and SVR in the model pool of DRS. 27 commonly used benchmark problems from 1 dimension to 8 dimensions are adopted to test our algorithm. The results confirm that DRS has better performances in both accuracy and stability than former surrogate algorithms, also much faster than existing hybrid models. As an effective and efficient algorithm to model highly computational cost problems, DRS can help accelerate their optimization process. We will further explore to adopt DRS in more diverse areas such as 3D mesh simplification [22], image registration [9], object detection [32] and feature selection [21] in our future work.

CRediT authorship contribution statement

Tianxin Huang: Conceptualization, Methodology. **Yong Liu:** Supervision. **Zaisheng Pan:** Writing - review & editing.

Declaration of Competing Interest

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

Acknowledgments

The authors declare that their work has not received any funding.

References

[1] Erdem Acar, Various approaches for constructing an ensemble of metamodels using local measures, Structural and Multidisciplinary Optimization 42 (6) (2010) 879–896.
[2] Erdem Acar, Masoud Rais-Rohani, Ensemble of metamodels with optimized weight factors, Structural and Multidisciplinary Optimization 37 (3) (2009) 279–294.
[3] Noor H Awad, Mostafa Z Ali, Rammohan Mallipeddi, Ponnuthurai N Suganthan, An improved differential evolution algorithm using efficient adapted surrogate model for numerical optimization, Information Sciences 451 (2018) 326–347.

[4] Anirban Basudhar, Samy Missoum, Adaptive explicit decision functions for probabilistic design and optimization using support vector machines, Computers & Structures 86 (19–20) (2008) 1904–1917.

[5] Malthe K Bisbo and Bjørk Hammer. Efficient global structure optimization with a machine-learned surrogate model. Physical review letters, 124 (8):086102, 2020..

[6] Mohamed Amine Bouhlel, John T Hwang, Nathalie Bartoli, Rémi Lafage, Joseph Morlier, and Joaquim RRA Martins. A python surrogate modeling framework with derivatives. Advances in Engineering Software, 135:102662, 2019..

[7] Guodong Chen, Yong Li, Kai Zhang, Xiaoming Xue, Jian Wang, Qin Luo, Chuanjin Yao, Jun Yao, Efficient hierarchical surrogate-assisted differential evolution for high-dimensional expensive optimization, Information Sciences 542 (2021) 228–246.

[8] Liming Chen, Haobo Qiu, Chen Jiang, Xiwen Cai, Liang Gao, Ensemble of surrogates with hybrid method using global and local measures for engineering design, Structural and Multidisciplinary Optimization 57 (4) (2018) 1711–1729.

[9] Yilin Chen, Fazhi He, Haoran Li, Dejun Zhang, Wu. Yiqi, A full migration bbo algorithm with enhanced population quality bounds for multimodal biomedical image registration, Applied Soft Computing 93 (2020) 106335.

[10] Stella M Clarke, Jan H Griebsch, and Timothy W Simpson. Analysis of support vector regression for approximation of complex engineering analyses. 2005..

[11] Noel Cressie, Statistics for spatial data, John Wiley & Sons, 2015.

[12] Laurence Charles Ward Dixon, The global optimization problem. an introduction, Toward global optimization 2 (1978) 1–15.

[13] Richard O Duda, Peter E Hart, et al, Pattern classification, John Wiley & Sons (2006).

[14] Andrea Gavana. Global optimization benchmarks and ampgo. Accessed Apr, 2016..

[15] Anthony Giunta, Layne Watson, A comparison of approximation modeling techniques-polynomial versus interpolating models, in: 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, 1998, p. 4758.

[16] Robert B Gramacy, Herbert KH Lee, Cases for the nugget in modeling computer experiments, Statistics and Computing 22 (3) (2012) 713–722.

[17] Rolland L Hardy, Multiquadric equations of topography and other irregular surfaces, Journal of geophysical research 76 (8) (1971) 1905–1915.

[18] Mohammed F Hussain, Russel R Barton, Sanjay B Joshi, Metamodeling: radial basis functions, versus polynomials, European Journal of Operational Research 138 (1) (2002) 142–154.

[19] Liangyue Jia, Reza Alizadeh, Jia Hao, Guoxin Wang, Janet K Allen, and Farrokh Mistree. A rule-based method for automated surrogate model selection. Advanced Engineering Informatics, 45:101123, 2020..

[20] Ruichen Jin, Wei Chen, Timothy W Simpson, Comparative studies of metamodelling techniques under multiple modelling criteria, Structural and multidisciplinary optimization 23 (1) (2001) 1–13.

[21] Haoran Li, Fazhi He, Yilin Chen, Yiteng Pan, Mlfs-ccde: multi-objective large-scale feature selection by cooperative coevolutionary differential evolution, Memetic Computing 13 (1) (2021) 1–18.

[22] Yaqian Liang, Fazhi He, and Xiantao Zeng. 3d mesh simplification with feature preservation based on whale optimization algorithm and differential evolution. Integrated Computer-Aided Engineering, (Preprint):1–19, 2020..

[23] Haitao Liu, Xu. Shengli, Xiaofang Wang, Jigang Meng, Sh.uhua. Yang, Optimal weighted pointwise ensemble of radial basis functions with different basis functions, AIAA Journal 54 (10) (2016) 3117–3133.

[24] Michael D McKay, Richard J Beckman, William J Conover, A comparison of three methods for selecting values of input variables in the analysis of output from a computer code, Technometrics 42 (1) (2000) 55–61.

[25] Ali Mehmani, Souma Chowdhury, Christoph Meinrenken, Achille Messac, Concurrent surrogate model selection (cosmos): optimizing model type, kernel function, and hyper-parameters, Structural and Multidisciplinary Optimization 57 (3) (2018) 1093–1114.

[26] Marcin Molga, Czesław Smutnicki, Test functions for optimization needs, Test functions for optimization needs 101 (2005) 48.

[27] Anoop A Mullur, Achille Messac, Extended radial basis functions: more flexible and effective metamodeling, AIAA journal 43 (6) (2005) 1306–1315.

[28] Anoop A Mullur, Achille Messac, Metamodeling using extended radial basis functions: a comparative approach, Engineering with Computers 21 (3) (2006) 203.

[29] Raymond H. Myers, Douglas C. Montgomery, Christine M. Anderson-Cook, Response surface methodology: process and product optimization using designed experiments, John Wiley & Sons, 2016.

[30] Jeng-Shyang Pan, Nengxian Liu, Shu-Chuan Chu, Taotao Lai, An efficient surrogate-assisted hybrid optimization algorithm for expensive optimization problems, Information Sciences 561 (2021) 304–325.

[31] Jiachang Qian, Jiaxiang Yi, Yuansheng Cheng, Jun Liu, Qi Zhou, A sequential constraints updating approach for kriging surrogate model-assisted engineering optimization design problem, Engineering with Computers 36 (3) (2020) 993–1009.

[32] Quan Quan, Fazhi He, Haoran Li, A multi-phase blending method with incremental intensity for training detection networks, The Visual Computer 37 (2) (2021) 245–259.

[33] Avijit Saha, Jim Christian, Dun-Sung Tang, and Wu Chuan-Lin. Oriented non-radial basis functions for image coding and analysis. In Advances in neural information processing systems, pages 728–734, 1991..

[34] S. Sakata, F. Ashida, M. Zako, Structural optimization using kriging approximation, Computer methods in applied mechanics and engineering 192 (7–8) (2003) 923–939.

[35] Egar Sanchez, Salvador Pintos, Nestor V Queipo, Toward an optimal ensemble of kernel-based approximations with engineering applications, Structural and multidisciplinary optimization 36 (3) (2008) 247–261.

[36] René M.M. Slot, John D. Sørensen, Bruno Sudret, Lasse Svenningsen, Morten L. Thøgersen, Surrogate model uncertainty in wind turbine reliability assessment, Renewable Energy 151 (2020) 1150–1162.

[37] Xueguan Song, Liye Lv, Jieling Li, Wei Sun, Jie Zhang, An advanced and robust ensemble surrogate model: extended adaptive hybrid functions, Journal of Mechanical Design 140 (4) (2018).

[38] Meng Tang, Yimin Liu, Louis J Durlofsky, A deep-learning-based surrogate model for data assimilation in dynamic subsurface flow problems, Journal of Computational Physics 413 (2020) 109456.

[39] Jun Tao, Gang Sun, Application of deep learning based multi-fidelity surrogate model to robust aerodynamic design optimization, Aerospace Science and Technology 92 (2019) 722–737.

[40] Hao Tong, Changwu Huang, Leandro L Minku, Xin Yao, Surrogate models in evolutionary single-objective optimization: A new taxonomy and experimental study, Information Sciences 562 (2021) 414–437.

[41] V Vapnik. The nature of statistical learning theory springer new york google scholar. New York, 1995..

[42] Vladimir Vapnik, Vlamimir Vapnik, Statistical learning theory wiley. New York 1 (624) (1998) 2.

[43] Felipe AC Viana, Raphael T Haftka, Valder Steffen, Multiple surrogates: how cross-validation errors can help us to obtain the best predictor, Structural and Multidisciplinary Optimization 39 (4) (2009) 439–457.

[44] Xinjing Wang, Liang Gao, Xinyu Li, Multiple surrogates and offspring-assisted differential evolution for high-dimensional expensive problems, Information Sciences 592 (2022) 174–191.

[45] Bayya Yegnanarayana, Artificial neural networks, PHI Learning Pvt. Ltd., 2009.

[46] Luis E. Zerpa, Nestor V. Queipo, Salvador Pintos, Jean-Louis Salager, An optimization methodology of alkaline–surfactant–polymer flooding processes using field scale numerical simulation and multiple surrogates, Journal of Petroleum Science and Engineering 47 (3–4) (2005) 197–208.

[47] Jian Zhang, Xinxin Yue, Jiajia Qiu, Muyu Zhang, Xiaomei Wang, A unified ensemble of surrogates with global and local measures for global metamodelling, Engineering Optimization 53 (3) (2021) 474–495.

[48] Jie Zhang, Souma Chowdhury, Achille Messac, An adaptive hybrid surrogate model, Structural and Multidisciplinary Optimization 46 (2) (2012) 223–238.

[49] Junqiang Zhang, Achille Messac, Jie Zhang, and Souma Chowdhury. Comparison of surrogate models used for adaptive optimal control of active thermoelectric windows. In 13th AIAA/ISSMO multidisciplinary analysis optimization conference, page 9279, 2010..

[50] Xinshuai Zhang, Fangfang Xie, Tingwei Ji, Zaoxu Zhu, Yao Zheng, Multi-fidelity deep neural network surrogate model for aerodynamic shape optimization, Computer Methods in Applied Mechanics and Engineering 373 (2021) 113485.