



# A new adaptive multi-fidelity metamodel method using meta-learning and Bayesian deep learning

Fenfen Xiong<sup>1</sup> · Chengkun Ren<sup>2</sup> · Bo Mo<sup>1</sup> · Chao Li · Xiao Hu<sup>3</sup>

Received: 9 August 2022 / Revised: 20 January 2023 / Accepted: 23 January 2023 / Published online: 28 February 2023  
© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2023

## Abstract

To reduce the computational cost, multi-fidelity (MF) metamodel methods have been widely used in engineering optimization. Most of these methods are based on the standard Gaussian random process theory; thus, the time cost required for hyperparameter estimation increases significantly with an increase in the dimension and nonlinearity of the problems especially for high-dimensional problems. To address these issues, by exploiting the great potential of deep neural networks in high-dimensional information extraction and approximation, a meta-learning-based multi-fidelity Bayesian neural network (ML-MFBNN) method is developed in this study. Based on this, to further reduce the computational cost, an adaptive multi-fidelity sampling strategy is proposed in combination with Bayesian deep learning to sequentially select the highly cost-effective samples. The effectiveness and advantages of the proposed MF-MFBNN and adaptive multi-fidelity sampling strategy are verified through eight mathematical examples, and the application to model validation of computational fluid dynamics and robust shape optimization of the ONERA M6 wing.

**Keywords** Multi-fidelity modeling · Meta-learning · Bayesian deep learning · Sequential sampling · Cost-effectiveness

## Abbreviations

$\nabla$	Gradient
$\mathcal{L}$	Loss function
$\theta$	Network parameters
$\varphi$	Network initial parameters
$\theta^-$	Parameters of the network other than the output layer
$\mathbf{w}^L$	The weights of the $L$ th layer of the network
$\mathbf{b}^L$	The deviations of the $L$ th layer of the network
$\mathbf{x}$	Input vector
$\mathbf{y}$	Response vector
$D$	Sample dataset
$f(\cdot)$	Bayesian neural networks
$f_{\theta}(\mathbf{x})$	Output of Bayesian neural networks

HF	High-Fidelity
LF	Low-Fidelity
ML	Meta-learning
BNN	Bayesian Neural Network
MF	Multi-fidelity

## 1 Introduction

High-fidelity (HF) simulation models, such as computational fluid dynamics (CFD) for aerodynamic analysis and finite element analysis (FEA) for structural analysis, have been widely used in engineering design optimization to improve design accuracy, which however significantly increases the computational cost. Meanwhile, uncertainties exist widely in engineering systems, manufacturing, and environments, which seriously affect the system performance. To reduce the impact of uncertainties, design using uncertainty methods such as robust optimization (Park et al. 2006) and reliability-based optimization (Paiva et al. 2014) has become an inevitable trend in engineering design. The consideration of uncertainties further increases the computational cost especially for multidisciplinary system with strong coupling among disciplines. Therefore, the metamodel technique (Ellis et al. 2020) was developed to reduce the computational

Responsible Editor: Ramin Bostanabad

Fenfen Xiong  
fenfenx@bit.edu.cn

<sup>1</sup> School of Aerospace Engineering, Beijing Institute of Technology, Beijing, China

<sup>2</sup> Southwest Technology and Engineering Research Institute, Chongqing, China

<sup>3</sup> Department of Mechanical, Imperial College London, London, England

cost by constructing an approximate model to replace the time-consuming HF simulation models. To reduce the number of HF simulations, a multi-fidelity modeling technique (Forrester et al. 2007), also known as model fusion, has been developed where a metamodel is built by fusing data of different fidelities and computational costs using only a small number of HF samples. As a large amount of lower-fidelity data is fused, the computational cost is significantly reduced while preserving the prediction accuracy of the metamodel.

As a commonly used multi-fidelity modeling technique, the Co-Kriging method (Parussini et al. 2017) establishes a recursive mapping relationship from low-fidelity (LF) to HF models by exploiting their correlation using the Gaussian stochastic process (GRP) theory. Bailly applied the Co-Kriging method to the design of a helicopter rotor blade to improve its performance in forward flight, and a factor of six reduction in the CPU time is observed compared with a single-fidelity Kriging model (Bailly and Bailly 2019). Furthermore, Liu et al. proposed exploiting the full correlation between models of all fidelities to improve the accuracy of model hyperparameter estimation (Liu et al. 2018). To reduce the complexity of hyperparameter estimation for Co-Kriging, Han et al. proposed a hierarchical Kriging method that directly treated the Kriging model constructed based on LF samples as the mean term of the multi-fidelity metamodel, and applied it to aircraft aerodynamic modeling (Han and Görtz 2012). Zhou et al. developed a multi-fidelity metamodel method based on the GRP, in which the metamodel was constructed by summing the two GRP models, respectively, based on the LF data and difference between the HF and LF data (Zhou et al. 2017). Wang et al. developed a multi-fidelity PC-Kriging approach for uncertainty propagation using the GRP and demonstrated that it was more accurate and robust than Co-Kriging, especially for problems with non-normally distributed inputs (Wang et al. 2019). Song et al. proposed a multi-fidelity metamodel method based on feasible intervals to improve prediction accuracy, and applied it to the structural performance evaluation of a bucket wheel reclaimer (Zhang et al. 2022). To address the issue that LF models can potentially return wrong trends and erroneous predictions to their HF counterparts for specific ranges of input parameters, Perdikaris et al. developed a probabilistic framework based on Gaussian process regression and nonlinear autoregressive schemes (Perdikaris et al. 2017). To address the non-hierarchical fidelity, i.e., the fidelity levels of lower-fidelity data source are not available, or the fidelity of lower-fidelity models might change over the input space, Chen et al. developed a nonhierarchical multi-model fusion method (Chen et al. 2017). The latent-map Gaussian processes (LMGPs) (Eweis-Labolle et al. 2022) were proposed for multi-fidelity modeling that enables efficient and accurate data fusion for both hierarchical and non-hierarchical cases (Zhang et al. 2020; Zanjani Foumani et al.

2022). In these works, the data fusion was converted into a latent space learning problem where the relations among different data sources were automatically learned. Most of these existing multi-fidelity modeling methods are based on the standard GRP modeling theory that constructs the mapping relationship between inputs and output of a physical process with only the GRP regression method (Liu et al. 2020). The most prominent weakness of standard GRP is that it suffers from a cubic time complexity  $O(n^3)$  because of the inversion and determinant of the  $n \times n$  kernel matrix. And thus, most existing multi-fidelity modeling methods generally require a large number of samples and a long time for hyperparameter estimation, especially for high-dimensional problems (Liu et al. 2020).

In recent years, deep learning techniques (Tripathy and Bilionis Dec. 2018) have been rapidly developed. In particular, for deep neural networks (DNNs), as a deeper model structure is employed compared with traditional metamodeling methods, their inductive ability has been greatly improved. Therefore, traditional machine learning in engineering analysis and optimization can be replaced with deep learning to enhance the prediction accuracy. Li et al. developed a metamodel to quickly detect the geometric abnormality of airfoils or wing sections based on a deep convolutional generative adversarial network and convolutional neural networks (Li et al. 2020). Halder et al. employed a long short-term memory based deep learning approach to model the transonic airfoil-gust interaction and the gust response on transonic aileron-buzz problems in replacement of CFD to reduce the computational cost (Halder et al. 2020). Bouhlel et al. used DNNs to model the aerodynamic force coefficients of airfoils in both the subsonic and transonic regimes, which drastically reduced the computational time of both analysis and design optimization (Bouhlel et al. 2020).

To reduce the computational cost, a combination of multi-fidelity modeling techniques and DNNs has been proposed to construct a multi-fidelity DNN (Meng and Karniadakis 2020). An effective optimization framework for aerodynamic shape design was established based on a multi-fidelity DNN by employing a generalized autoregressive scheme for model fusion similar to that of Co-Kriging (Zhang et al. 2021a). Tao trained a deep belief network for each LF model, based on which a linear regression model was trained to replace the HF model using a small amount of HF data; this method was subsequently used for robust aerodynamic design optimization (Tao and Sun 2019). Meng proposed a multi-fidelity Bayesian neural network (BNN) metamodel consisting of three neural networks in series: a fully connected neural network for the LF model, a BNN for capturing the cross-correlation between the LF and HF models, and a physics-informed neural network (Meng et al. 2021). Generally, a functional relationship between the LF and HF data is required to be established in these works using autoregressive

formulae or DNNs (Zhang et al. 2021b). With an increase in the number of multi-fidelity models to be fused, the complexity and number of unknown hyperparameters are greatly increased, which increases the difficulty and cost of network training. The transfer learning technique has been used to blend LF physics informed and data-driven deep learning techniques (Chakraborty 2021). Although additional network construction is avoided, multi-fidelity modeling using transfer learning yields slow convergence of network training and the generalization ability is poor (Finn et al. 2017).

Compared to transfer learning, the newly proposed meta-learning (ML) theory in the field of deep learning has a stronger generalization ability and is capable of small sample learning (Finn et al. 2017; Xu and Du Jul. 2020). Traditional DNNs aim to find the optimal network hyperparameters to achieve an accurate prediction for a certain learning task. However, the ML technique aims to find the optimal initial hyperparameters suitable for one category of learning tasks by exploring the similarities between different tasks of this category. Based on the obtained optimal initial hyperparameters, the network can quickly converge for a new learning task of this category, thus reducing the number of new samples required. It has been reported that only one picture sample is required for a new task of target classification using meta-learning, and a prediction accuracy of up to 77% can be achieved for a five-classification task (Rodríguez et al. 2020).

To explore the advantages of multi-fidelity deep learning for high-dimensional problem, and reduce the complexity and cost of network training, we propose a meta-learning-based multi-fidelity metamodel (ML-MFM) approach, inspired by the meta-learning theory. With the developed method, an implicit mapping relationship from the LF to HF models is captured using meta-learning theory, in which no additional networks or parameters are introduced. Thus, multi-fidelity ( $> 2$ ) models can be fused quickly and easily. The proposed method starts with the construction of a DNN based on LF data, wherein the current optimal initial network hyperparameters suitable for the learning of input–output mapping are obtained. Through a meta-feature learning of the input–output mapping with increasing fidelity of data, the optimal initial hyperparameters are gradually optimized and passed to the training of the higher-fidelity data. Finally, the network quickly converges with a small amount of HF data. Meanwhile, the proposed ML-MFM approach can also be applied to multi-fidelity modeling with non-hierarchical fidelity. In this case, all data of lower-fidelity models are collected to train a DNN, based on which the network parameters are updated by HF data using the meta-learning technique.

On the other hand, it is well known that the sampling strategy is critical for multi-fidelity metamodels (Jivani et al. 2021). The sampling strategy determines the numbers and

locations of the HF and LF samples. For the existing multi-fidelity DNN methods, the samples of different data sources are basically selected according to personal experience. If the accuracy of the DNN is not sufficient, it is preferable to add HF data, which is time-consuming. There is a lack of scientific sampling method for multi-fidelity deep learning in the literature. In fact, in some design regions, the LF and HF models may have comparable prediction accuracy. For example, in the CFD research field, the flow prediction accuracies of the HF and LF CFD simulations in the region where the flow situation is fully developed are similar. In this case, the computationally cheaper LF model should be evaluated to obtain the response. This idea was proposed by Huang et al. (Huang et al. 2006) for design optimization and has been applied to many problems, such as aircraft aerodynamic design and turbine design. As opposed to being generated in one stage or by only adding HF samples, this method sequentially generates the samples simultaneously considering the sample cost and ability to improve the metamodel accuracy (Ren et al. 2021). Most of these sampling methods employ the GRP theory to quantify the prediction uncertainty, based on which new samples are selected using objective-oriented sampling criteria, such as expectation improvement (EI) (Jones et al. 1998). The practical problem is often highly nonlinear with high-dimensional inputs, rendering it difficult to quantify and distinguish the capability of accuracy improvement for multi-fidelity simulation models. Deep learning employs a completely different model structure from the GRP; thus, it is difficult to adapt existing adaptive sampling algorithms to deep learning. The Bayesian deep learning method can provide uncertainty quantification of the response prediction (Wang and Yeung 2021), which provides a premise for sequential sampling. Therefore, the proposed ML-MFM method is further combined with Bayesian deep learning to quantify the prediction uncertainty of metamodel. Based on it, an adaptive multi-fidelity sampling strategy is developed to sequentially generate samples that can reduce the computational cost as much as possible, while with satisfactory metamodel accuracy. With the proposed sampling scheme, a cost-effectiveness index that evaluates the potential to improve the prediction accuracy and cost for the multi-fidelity models is established and optimized.

The remainder of this paper is organized as follows. A brief review of the existing meta-learning and BNN techniques is presented in Sect. 2. In Sect. 3, the proposed ML-MFM approach and sampling scheme are described in detail. In Sect. 4, the proposed methods are applied to several numerical examples to evaluate their effectiveness and advantages. In Sect. 5, an aerodynamic robust optimization problem is evaluated to further verify the effectiveness and applicability of the proposed methods in addressing practical problems. Finally, the conclusions are presented in Sect. 6.

## 2 Technical background

### 2.1 Meta-learning method

For a certain application scenario, deep learning requires a large number of samples for network training to obtain the input–output mapping relationship. Once the application scenario is changed, a new network must be trained again using a large number of samples, which is undoubtedly time consuming and cumbersome. Moreover, the number of samples available for the new application scenario is often small, and thus the network accuracy cannot be guaranteed. To address this problem, the meta-learning theory has been developed to address small-sample learning problems (Zhang et al. 2021b; Chakraborty 2021; Finn et al. 2017; Xu and Du Jul. 2020). Common deep learning models aims to learn a mathematical model for response prediction, whereas meta-learning is oriented to the process of learning, rather than the result of learning. Meta-learning learns “how to learn a mathematical model faster and better,” that is, learning to learn. As the most effective meta-learning framework, the model-agnostic meta-learning (MAML) model (Zhang et al. 2021b) trains a set of optimal initialization network parameters, based on which one or more gradient adjustment steps are performed to achieve rapid adaptation to new tasks with only a small number of samples. In this study, the proposed multi-fidelity metamodel method employs MAML as the meta-learning framework.

Figure 1 illustrates the basic learning architecture of the MAML framework, where  $\nabla$  and  $L$  denote the gradient and loss functions in the network training, respectively. At the beginning of meta-learning, the number of iterations  $K$  of training must be tested or given empirically, and then the network parameters are initialized as  $\phi_0$ . To improve the generalization ability of the network, the sampling with replacement method is employed on dataset D to generate several training tasks,  $T_i(i = 1, \dots, n)$ , and a certain number of validation sets. For each training task  $T_i$ , a meta-learner ( $M_i$ ) is constructed

based on the initial parameters  $\phi_0$ , and the network parameters  $\theta_i$  are obtained by updating  $M_i$  with the gradients calculated for  $T_i$ . Based on  $\theta_i(i = 1, \dots, n)$ , the corresponding gradients are calculated again on the validation datasets, which are averaged to update the initial parameters of the network, that is,  $\phi_0 = \phi_0^1$ . To date, the first iteration of meta-learning training has been completed.

Then, based on the updated initial network parameters,  $\phi_0$ , the above process is repeated  $K-1$  times. In this process, through the continuous extraction of the meta-features of this category of learning tasks,  $\phi_0^i(i = 1, \dots, K)$  is gradually optimized and eventually converges, to capture the global features of this category of learning tasks to the extent possible. The final set of optimal initial network parameters obtained,  $\phi_0^* = \phi_0^K$ , can greatly facilitate the fast convergence of new training tasks, thus significantly reducing the number of training samples. In fact,  $\phi_0^*$  also represents the optimal network parameters for Bayesian networks with the current dataset D (Zhang et al. 2021b). It is worth noting that meta-learning requires two gradient calculations in one iteration of training, which is an important feature distinguishing it from common deep learning techniques.

### 2.2 Bayesian neural network

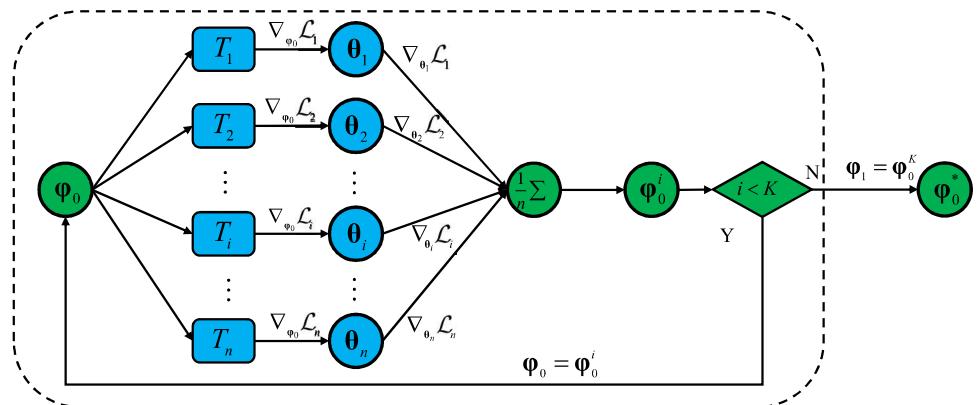
As a branch of DNNs, the BNN method, has the advantage of being highly robust and less prone to overfitting (Wang and Yeung 2021). Most importantly, it can quantify the uncertainty of the response prediction, which provides a premise for the implementation of sequential sampling in this study.

For a specific problem, when given a  $d$ -dimensional input vector  $\mathbf{x} \in \mathbf{R}^d$ , a DNN with  $L$  layers can be defined as follows:

$$f_{\theta}(\mathbf{x}) = \mathbf{w}^L \phi_{\theta^-}(\mathbf{x}) + \mathbf{b}^L \quad (1)$$

where  $\mathbf{w}^L$  and  $\mathbf{b}^L$  are the weights and biases of the  $L$ th layer of the network (i.e., the output layer),  $\phi_{\theta^-}(\mathbf{x})$  is the iterative value of the input variable  $\mathbf{x}$  after the previous  $L-1$  layers of network operations,  $\theta^-$  denotes all the network parameters

**Fig. 1** Basic learning architecture of MAML meta learning



of the previous  $L-1$  layers, and  $f_{\theta}$  denotes the DNN with network parameters  $\theta$ . For general deep-learning methods, the weights  $w$  and biases  $b$  are deterministic. To improve the robustness of the network,  $w$  and  $b$  are considered to obey certain probabilistic distributions and their posterior distributions are estimated using Bayesian theory. Figure 2 shows the structure of a fully connected BNN, where the weight  $w_i$  and biases  $b$  of any neuron are considered to follow certain distributions rather than deterministic values.

We assume that the parameters of the BNN network are  $\theta$ , and  $p(\theta)$  is the prior distribution of  $\theta$ , which is generally assumed to be a standard normal distribution, as is commonly considered in practice (Jones et al. 1998; Wang and Yeung 2021). The training set is  $D = \{X, Y\}$ , where  $X$  is the input data and  $Y$  is the response data. Based on the Bayesian formula, the posterior probability of  $\theta$  is obtained as follows:

$$p(\theta|X, Y) = \frac{p(Y|X, \theta)p(\theta)}{\int p(Y|X, \theta)p(\theta)d\theta} \quad (2)$$

The integration of the denominator in the above equation must be performed over the entire space of  $\theta$ , which is very complicated and difficult. In this study, we adopt the variational inference method proposed in (Blundell et al. 2015) to calculate this integral by establishing a simple distribution following a normal distribution to approximate the posterior distribution  $p(\theta|X, Y)$ . By adjusting the mean and variance of the simple distribution, the Kullback–Leibler (KL) divergence between this simple distribution and the real posterior distribution is minimized to obtain the optimal parameter estimate  $\theta^*$  of the BNN network (Wang and Yeung 2021).

For BNNs, because the parameters are all random variables, the training difficulty and required number of samples are significantly increased. To tradeoff between the evaluation time and model capacity, based on the fully connected neural network, only the output layer is set as the Bayesian layer, and no bias exists in the output layer in this study, as in Ref. (Jones et al. 1998). The aforementioned network is adopted as the basic structure of the BNN in our work, based on which a multi-fidelity DNN metamodel is constructed. For an established BNN,  $W$  denotes the weights of the

output layer, and are considered to be normally distributed to simplify the network parameter estimation and reduce the computational time of training (Snoek et al. 2015), that is,  $W \sim N(\mu_w, \sigma_w^2)$ .  $\theta^-$  denotes the parameters of the other layers, and the BNN can then be expressed as follows:

$$f_{\theta}(x) = w\phi_{\theta^-}(x) \quad (3)$$

where  $\phi_{\theta^-}(x)$  denotes the output vector of the penultimate layer. As the weights and biases of the neurons in all the previous layers before the output layer of the BNN are deterministic,  $\phi_{\theta^-}(x)$  is also a deterministic value at a given input  $x$ . Meanwhile, as the output layer weight  $w$  is considered to follow a normal distribution, the BNN response prediction at input  $x$  can be viewed as a superposition of multiple normal distributions. Therefore, the response prediction of  $f_{\theta}(x)$  for any input can also be considered to follow a normal distribution.

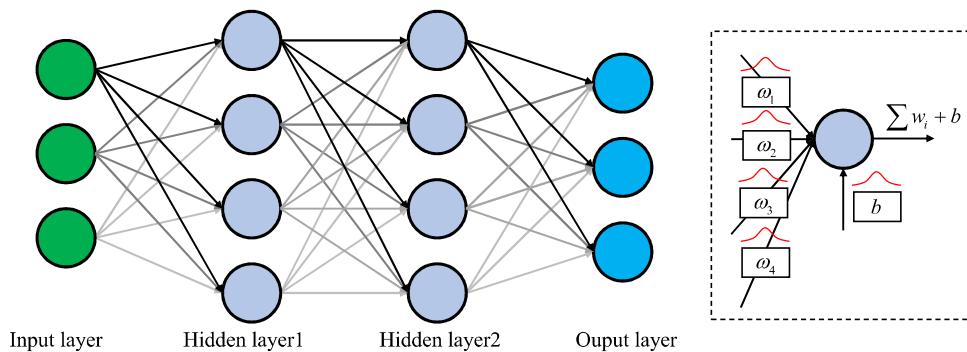
It should be noticed that the response predictions of BNN in this work and GRP both follow normal distributions, which however are caused by different reasons. For GRP, the output response to be approximated has to be assumed to be Gaussian distributed. However, for BNN, as the weights of BNN are considered to be normally distributed and no bias exists in the output layer, the response predictions follows normal distribution.

### 3 The proposed multi-fidelity metamodeling method

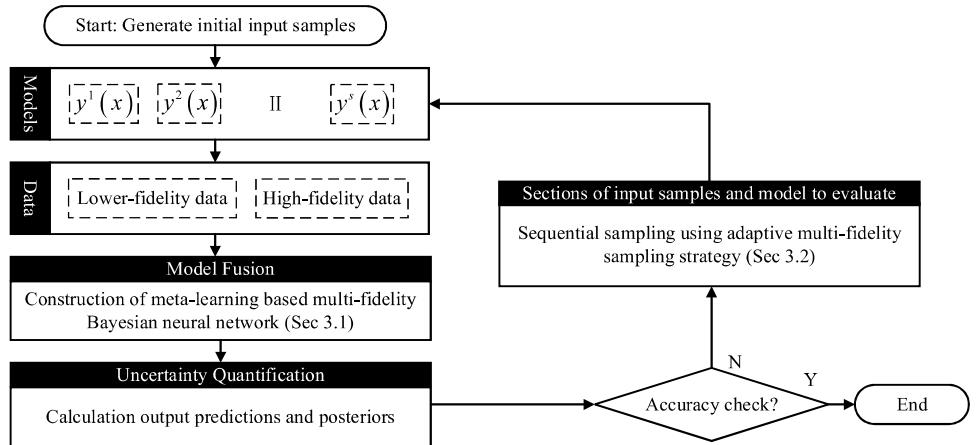
Figure 3 illustrates the flowchart of the proposed multi-fidelity metamodel approach employed in this study. It mainly involves two key components: the construction of a meta-learning-based multi-fidelity Bayesian neural network (ML-MFBNN) metamodel and sequential sampling to update the metamodel using the adaptive multi-fidelity sampling strategy.

*Step 1* Initial input samples are generated for each multi-fidelity model, for example, using the Latin hypercube sampling (LHS) method, and each model is evaluated to

**Fig. 2** Structure of Bayesian neural network



**Fig. 3** Flowchart of the proposed multi-fidelity metamodelling method



calculate the corresponding response data to generate the initial dataset. Simultaneously, a certain number of HF data are generated as the test dataset, and the number of iterations is set to  $I = 0$ .

*Step 2* Based on the current sample dataset, the proposed ML-MFBNN method (Section 3.1) is used for multi-model fusion, and a multi-fidelity BNN metamodel is constructed for response prediction.

*Step 3* The prediction accuracy of the constructed metamodel is checked using the test dataset. The root mean squared error (RMSE) of the metamodel is calculated as

$$\text{RMSE} = \sqrt{\frac{1}{m} \sum_{i=1}^m (y_i - \hat{y}_i)^2} \quad (4)$$

where  $m$  denotes the number of samples in the test set,  $y$  denotes the true response value, and  $\hat{y}$  denotes the metamodel prediction value. If the  $\text{RMSE} < \Delta$  ( $\Delta$  is a small constant specified by the users), the metamodel construction is completed; otherwise, proceed to Step 4.

*Step 4* The proposed adaptive multi-fidelity sampling strategy (Section 3.2) is used to generate a certain number of input samples and determine the corresponding analysis models to be evaluated. The newly generated samples are added to the existing dataset, set  $I = I+1$ , and proceed to Step 2.

### 3.1 Meta-learning-based multi-fidelity BNN

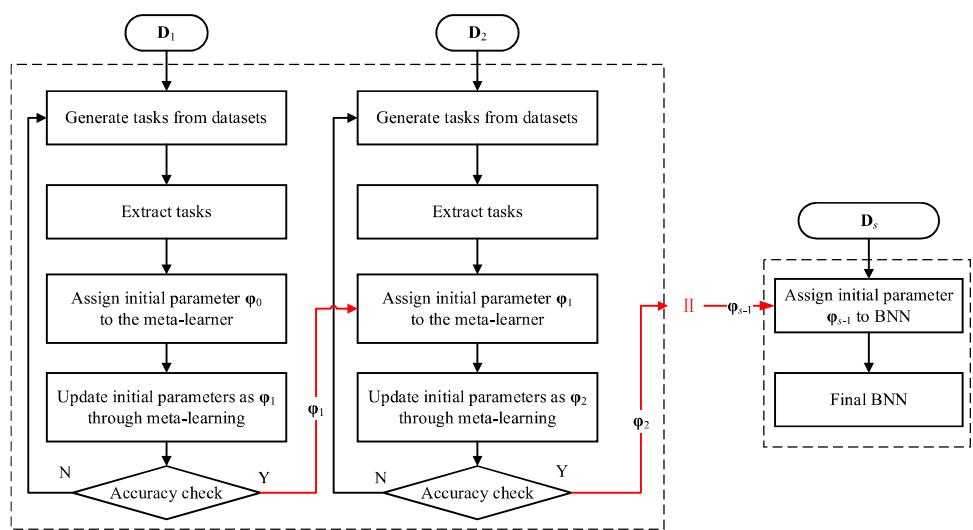
Considering that the multi-fidelity analysis models all describe the same physical process, there must be a correlation among them, i.e. the low-fidelity model can provide correct trends of high-fidelity one in most locations. This is very similar to the application scenario of meta-learning. In this study, we introduce meta-learning into the fusion of multi-fidelity simulation models, in which the BNN is employed as the basic network structure considering its

strong robustness and capability of uncertainty quantification of the response prediction. Through meta-learning, the initial network parameters are considered as the link between the LF and HF models to establish an implicit bridge mapping relationship.

The function  $y = g(x)(x = [x_1, \dots, x_d])$  is taken as an example to introduce the proposed multi-fidelity BNN metamodel method. It is assumed that there exist  $s$  accuracy levels of the multi-fidelity analysis model  $y^t(x)|_{t=1,\dots,s}$ , and its corresponding input and output sample dataset is  $\mathbf{D}_t = \{\mathbf{x}_i^t, \mathbf{y}_i^t\}|_{i=1,\dots,n_t}$  ( $i = 1, \dots, n_t$ ). The larger  $t$  is, the higher the accuracy level of the corresponding model  $y^t$ , and the less sample data it can provide, that is, the smaller the value of  $n_t$  is. For convenience, in this study, we refer to  $y^s(\mathbf{x})$  and  $\mathbf{D}_s$  as the HF model and HF data, respectively;  $y^t(\mathbf{x})$  and  $\mathbf{D}_t$  ( $t = 1, \dots, s-1$ ) are referred to as the lower-fidelity models and lower-fidelity data, respectively. The proposed multi-fidelity metamodel method aims to construct a BNN to replace the HF model  $y^s(\mathbf{x})$  by fusing a large amount of lower-fidelity data and a small amount of HF data. As  $\mathbf{D}_1, \dots, \mathbf{D}_s$  are analysis models with varying fidelities that attempt to describe the same input–output relationship, the training of  $\mathbf{D}_1, \dots, \mathbf{D}_s$  can be considered as one category of learning tasks.

Figure 4 shows the flowchart of the proposed ML-MFBNN method, which consists of a multi-fidelity meta-learning network and a final model training network. Using meta-learning theory, the data  $\mathbf{D}_t$  ( $t = 1, \dots, s-1$ ) are trained from the lowest fidelity to higher fidelity step by step to construct a BNN for  $y^s(\mathbf{x})$ , to obtain the best initial network parameters  $\boldsymbol{\varphi}_t$  that can be easily adapted to this category of learning tasks. During the training of  $\mathbf{D}_t$  ( $t = 1, \dots, s-1$ ), the meta-features of this category of learning tasks can be extracted gradually. After the training of  $\mathbf{D}_{s-1}$  is completed, the initial network parameters  $\boldsymbol{\varphi}_{s-1}$  are updated using all the historical data information, which can fully capture the global features of this category

**Fig. 4** Flowchart of the proposed ML-MFBNN



of learning tasks. Therefore, when the HF samples  $\mathbf{D}_s$  are introduced, the network can achieve rapid convergence with only a small amount of HF data.

As shown in Fig. 4, for the training of  $\mathbf{D}_t$  ( $t = 1, \dots, s-1$ ) of the model  $y^t(\mathbf{x})$ , the sampling with replacement method is employed to generate multiple sets of training tasks  $\{T_{t1}, \dots, T_{ti}, \dots, T_{tm}\}$  and validation sets from  $\mathbf{D}_t$ . Several ( $k, k < n$ ) training tasks are extracted from  $\{T_{t1}, \dots, T_{ti}, \dots, T_{tm}\}$ , which will be used for network training of model  $y^t(\mathbf{x})$ . This process will be repeated many times and the initial network parameters  $\boldsymbol{\varphi}_t$  are continuously updated accordingly till the accuracy criterion is satisfied. Then, the optimal initial network parameters  $\boldsymbol{\varphi}_t$  are calculated by averaging the optimization directions of the current  $k$  training tasks and passed to the task training of the higher fidelity dataset  $\mathbf{D}_{t+1}$  as the initial network parameters, which greatly speeds up the convergence of the network training of  $\mathbf{D}_{t+1}$ . Using the meta-learning theory, the optimal initial parameters of the network are passed to the training of higher-fidelity data step by step, which ensures that the network prediction accuracy can be improved to the extent possible for a given amount of data. Furthermore, with the proposed ML-MFBNN method, in addition to the BNN metamodel  $f_{\boldsymbol{\theta}_s}$  for the HF model  $y^s(\mathbf{x})$ , the BNN metamodel  $f_{\boldsymbol{\theta}_t}(\boldsymbol{\theta}_t = \boldsymbol{\varphi}_t)$  for all the lower-fidelity models  $y^t(\mathbf{x})(t = 1, \dots, s-1)$  can also be obtained. This provides the premise for the following multi-fidelity sequential sampling.

It is worth noting that in addition to hierarchical cases, the proposed ML-MFBNN method is also applicable to non-hierarchical problems. In this case, samples from all lower-fidelity models are collected into one dataset, i.e.  $\mathbf{D}_L = \{\mathbf{D}_1, \dots, \mathbf{D}_{s-1}\}$ . BNN is constructed with  $\mathbf{D}_L$  using the meta-learning theory, based on which the network parameters are re-trained and optimized with  $\mathbf{D}_s$ .

### 3.2 Adaptive multi-fidelity sampling strategy

The selection of samples has a significant impact on the accuracy and efficiency of the metamodels. To further reduce the computational cost, based on the multi-fidelity BNN metamodel, an adaptive multi-fidelity sampling strategy is established, which sequentially selects highly cost-effectiveness samples to ensure the accuracy of the metamodel while minimizing the computational cost. The sampling process consists of the selection of the input samples and the identification of the analysis models to be evaluated, which are executed sequentially.

### 3.2.1 Selection of the input samples

For the BNN metamodel, it is very convenient to provide the posterior variance of the response prediction at any input location, which is a type of prediction uncertainty (Snoek et al. 2015). Similar to the GRP, the magnitude of the prediction uncertainty characterizes the lack of samples at the input location and can therefore be used as a rule for the input sample selection.

Based on the method introduced in Sect. 3.1 and Eq. (3), the BNN metamodel for the HF model  $y^s(\mathbf{x})$  constructed by multi-fidelity model fusion can be expressed as follows:

$$f_{\theta_s}(\mathbf{x}) = \mathbf{w}_s \phi_{\theta_c}(\mathbf{x}) \quad (5)$$

From Sect. 2.2, it can be concluded that the response prediction of the BNN follows a normal distribution. Therefore, the posterior distribution of the response prediction of  $f_{\theta_s}(x)$  at the input location  $x$  is

$$p_s(\mathbf{x}) \sim N(\alpha_s(\mathbf{x}), \eta_s(\mathbf{x})) \quad (6)$$

where

$$\begin{aligned}\alpha_s(\mathbf{x}) &= \mathbb{E}[f_{\theta_s}(\mathbf{x})|D_s] \\ \eta_s(\mathbf{x}) &= \mathbb{E}[f_{\theta_s}^2(\mathbf{x})|D_s] - \mathbb{E}[f_{\theta_s}(\mathbf{x})|D_s]^2\end{aligned}\quad (7)$$

with  $E[\cdot]$  denoting the expectation.

The above equations defined in Eq. (7) can be calculated using numerical integration, such as the Gaussian integration method.

$$\alpha_s(\mathbf{x}) = \sum_m \mathbf{g}_s^m \cdot \mathbf{t}_s^m \phi_{\theta_s}(\mathbf{x}) \quad (8)$$

$$\eta_s(\mathbf{x}) = \sum_m \mathbf{g}_s^m \left( \mathbf{t}_s^m \phi_{\theta_s}(\mathbf{x}) - \alpha_s(\mathbf{x}) \right)^2 \quad (9)$$

where  $\mathbf{t}_s^m$  and  $\mathbf{g}_s^m$  denote the integration nodes and weights corresponding to the weights  $\mathbf{w}_s$  of the output layer of the BNN, respectively.  $\mathbf{t}_s^m$  and  $\mathbf{g}_s^m$  are determined by the mean  $\mu_{w_s}$  and standard deviation  $\sigma_{w_s}$  of the distribution of  $\mathbf{w}_s$  (Lee and Chen 2009); generally, three integration nodes can achieve satisfactory integration (that is,  $m=1,2,3$ ).

In deep learning, if too few samples are generated to update the network each time, the accuracy of the BNN will not greatly improve and the sampling process will be tediously long. Therefore, a multipoint sampling method is established by maximizing the prediction posterior variance obtained by the BNN. Meanwhile, the number of samples taken at each time should not be too large; otherwise, the advantage of sequential sampling cannot be exploited. Therefore, the upper limit of the number of samples selected at each time is set as  $n^U$ . Based on Eq. (9), the input position with the maximum prediction posterior variance ( $\eta_s^{\max}$ ) is selected as the key input sample. In this work, the key input sample is approximately selected from a large number of randomly generated samples using Latin Hypercube sampling (LHS). A number of points with the posterior variance close to  $\eta_s^{\max}$  (i.e.  $\alpha * \eta_s^{\max}$ ,  $0.9 < \alpha < 0.95$ ) are selected from the aforementioned LHS samples as the candidate input samples. Meanwhile, to avoid a sample cluster around a certain input position that limits the overall improvement of the network accuracy, a distance-based sample selection principle is developed. Based on the Euclidean distance among the key input sample and candidate input samples, a set of candidate input samples that are not too close to each other ( $d \geq d_t$ ) are selected as the effective input samples. The distance threshold  $d_t$  is calculated as

$$d_t = \beta \frac{d_{\max} + d_{\max}^c}{2} \quad (10)$$

where  $d_{\max}$  denotes the maximum value of the ranges for all the dimensions,  $d_{\max}^c$  denotes the maximum Euclidean distance between any candidate sample and key input sample, and  $\beta$  is the correction parameter. As the magnitudes of the ranges for different dimensions may be significantly

different, the ranges of all the dimensions are normalized before calculating the distances.

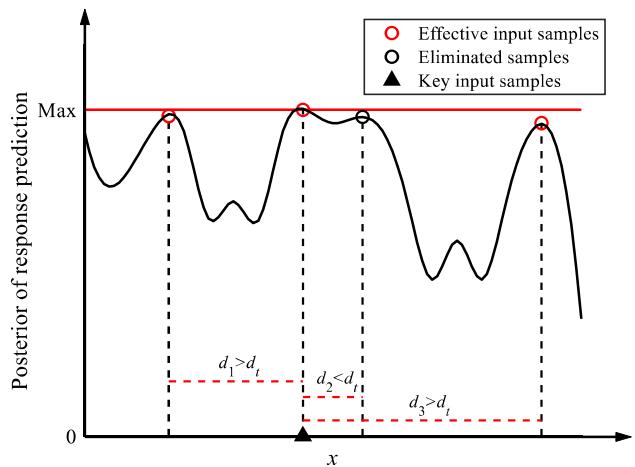
For the above user-defined parameters, there is no theoretically way to determine their values. Based on our experience from large number of simulation tests, it can be approximately set according to the problem dimension  $d$ . To make a good tradeoff between the magnitude and speed of accuracy increase of network, when  $d < 10$ , it can be approximately set  $n^U = 5$ ,  $\alpha = 0.9$  and  $\beta = 0.1$ ; when  $10 \leq d \leq 20$ ,  $n^U = 10$ ,  $\alpha = 0.9$  and  $\beta = 0.1$ ; when  $d > 20$ ,  $n^U \in [15, 20]$ ,  $\alpha = 0.95$ , and  $\beta \in [0.15, 0.2]$ . Figure 5 illustrates the proposed input sample selection scheme. It can be seen that a trade-off between prediction uncertainty reduction and the avoidance of sample cluster is obtained.

### 3.2.2 Selection of the analysis models

By extending the sampling strategy based on the GRP in Ref. (Ren et al. 2021) to the BNNs, a method to quantify the accuracy improvement capability of the multi-fidelity model  $y^t(\mathbf{x})$  with an added new sample is derived without evaluating the real model. A cost effectiveness index (CEI) for each multi-fidelity model is established.

$$\text{CEI}^t = \text{Corr}^t(\mathbf{x}) \times \left( \frac{C^s}{C^t} \right) t = 1, \dots, s \quad (11)$$

where the superscripts  $s$  and  $t$  denote the HF model  $y^s(\mathbf{x})$  and the  $t$ th LF model  $y^t(\mathbf{x})$ , respectively,  $\text{Corr}^t$  denotes the posterior correlation between  $y^t(\mathbf{x})$  and  $y^s(\mathbf{x})$  at some input location  $(\mathbf{x})$ ; and the closer its value is to 1, the closer is the prediction accuracy of  $y^t(\mathbf{x})$  and  $y^s(\mathbf{x})$  at  $(\mathbf{x})$ .  $C^s$  and  $C^t$  denote the costs of  $y^s(\mathbf{x})$  and  $y^t(\mathbf{x})$ , respectively, which can be the modeling time cost or labor cost, the computational cost of one model evaluation, or their combined consideration.



**Fig. 5** Illustration of input sample selection

For one of the obtained input samples, the analysis model with the largest CEI (denoted by  $M^*$ ) is selected for evaluation, which can be determined by:

$$M^* = \arg \max_t CEI^t \quad t = 1, \dots, s \quad (12)$$

Equation (12) indicates that for one of the obtained input samples, the models selected for evaluation can improve the accuracy of the metamodel with as less computational cost as possible. The key procedure involves the calculation of  $Corr^t(\mathbf{x})$ , which is introduced in detail.

According to Sect. 3.1, a BNN metamodel  $f_{\theta_t}$  for the lower-fidelity model  $y^t(\mathbf{x})(t = 1, \dots, s - 1)$  can be obtained in the process of multi-fidelity model fusion. Based on Eqs. (6–9), the posterior distribution  $p_t(x) \sim N(\alpha_t(x), \eta_t(x))$  of the response prediction for model  $y^t(x)$  at the input location  $\mathbf{x}$  can be obtained based on the trained BNN metamodel  $f_{\theta_t}$  without calling the real analysis model.  $Corr^t(x)$  can be characterized by the distance between the posterior distributions ( $p_t(x)$  and  $p_s(x)$ ), based on the BNNs for  $y^t(x)$  and  $y^s(x)$ . At the early stage of metamodel construction, the number of samples is generally small, and thus, the prediction accuracy of the BNN is poor. In this case,  $p_t(x)$  and  $p_s(x)$  may be far away from each other. Therefore, the Wasserstein distance criterion is introduced to better quantify the distance between the two distributions (Panaretos and Zemel Mar. 2019).

It is worth noting that for non-hierarchical fidelity, as all lower-fidelity data sources are collected to train one Bayesian neural network, the network accuracy improvement of lower-fidelity models cannot be distinguished. Therefore, the proposed adaptive multi-fidelity sampling strategy is not applicable in this case.

## 4 Numerical simulation tests

Two tests were conducted to investigate the effectiveness of the proposed ML-MFBNN and adaptive multi-fidelity sampling strategy. For Test 1, the proposed ML-MFBNN metamodel method is tested using eight mathematical examples adopted from literature to verify its accuracy and efficiency in Sect. 4.1. Examples 1, 3–8 are hierarchical fidelity cases, while Example 2 is a non-hierarchical one. For Test 2, the effectiveness of the adaptive multi-fidelity sampling strategy is verified using three mathematical examples in Sect. 4.2. To investigate the adaptability of the proposed methods to different levels of correlation between various data sources, Examples 1, 2, 4–8 of which the multi-fidelity models provide the similar trend at most input locations (denoted as “high correlation” between various data sources) as is commonly considered in literature are tested. In some particular cases,

the HF and LF functions may behave in opposite directions in many input regions (denoted as “low correlation” between various data sources), and thus Example 3 (Perdikari et al. 2017) is tested.

For meta-learning, the number of training tasks is  $k = 10$ , the iteration number of meta-learning training is  $K = 120$ , and the dropout of the BNN is set to 0.5. The computer configuration used for the simulation is 16G RAM, NVIDIA 1660 graphics card, and Intel(R) Core(TM) i5-10400F CPU.

### 4.1 Test 1

The ML-MFBNN method proposed in this paper is compared with the commonly used multi-fidelity modeling method (Co-Kriging) that adopts the standard GRP. The method that directly trains a BNN metamodel using only HF data (denoted as H-BNN) is also compared. The RMSE of the metamodel accuracy is calculated for comparison. Meanwhile, considering that it is often the mean and variance of the output response that are considered in the uncertainty quantification and robust design optimization, the mean and variance of the response obtained based on the three methods (ML-MFBNN, Co-Kriging, and H-BNN) are further compared with the results obtained with Monte Carlo simulation (MCS) directly on the HF data (denoted as Direct MCS, DMCS). Table 1 lists eight mathematical examples adopted from literature with different input dimensions, functional nonlinearities, and the input distribution information.

Table 2 lists the numbers of samples empirically determined for each multi-fidelity model. The proposed ML-MFBNN and Co-Kriging methods use exactly the same multi-fidelity samples to construct the multi-fidelity metamodel, whereas the H-BNN uses only the HF samples as those used by ML-MFBNN and Co-Kriging. For Co-kriging, similar to the proposed MF-MLBNN method, all the lower-fidelity data are collected as the LF dataset for Example 2 (non-hierarchical), and then model fusion is implemented using HF and LF datasets. The number of runs for the DMCS is set to  $10^5$ . To test the accuracy of the constructed metamodels, 30 sets of HF samples are randomly generated, based on which the RMSEs of the three methods are calculated.

It is well known that the network architecture design is very important to the accuracy and efficiency of BNN, which has been explored in many works (Yin and Zhu 2020). For practical applications, the network training can be conducted many times with different combinations of the number of hidden layers and the number of layer neurons to determine the best network structure. Figure 6 shows the RMSE of multi-fidelity BNN of Example 5 with different network structures. In this way, for all the mathematical examples, the network structure with 3 hidden layers and 25 neurons

**Table 1** Test functions

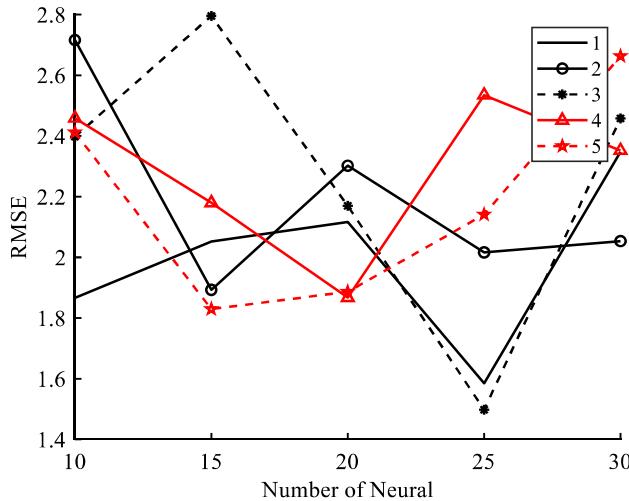
No	Functions	Input distribution
1	$y^3 = \sin(x) + 0.2x + (x - 0.5)^2/16 + 0.5$ $y^2 = \sin(x) + 0.8x + (x - 0.5)^2/45 + 0.5$ $y^1 = \sin(x) + 0.2x + 0.5$	$x \sim N(1.5, 1.33^2)$
2	$y^3 = -\sin(x) - \exp\left(\frac{x}{100}\right) + 10,$ $y^2 = y^3 + 0.3 - 0.03(x - 3)^2,$ $y^1 = y^3 + 0.3 - 0.03(x - 7)^2$	$x \sim N(5, 0.83^2)$
3	$y^2 = (x - \sqrt{2}) \sin^2(8\pi x)$ $y^1 = \sin(8\pi x)$	$x \sim U(0, 1)$
4	$y^3 = \left(1 - e^{\frac{-1}{2x_2}}\right) \left( \frac{1000t_f x_1^3 + 1900x_1^2 + 2092x_1 + 60}{1000t_f x_1^3 + 500x_1^2 + 4x_1 + 20} \right)$ $y^2 = \left(1 - e^{\frac{-1}{2x_2}}\right) \left( \frac{1000t_f x_1^3 + 1900x_1^2 + 2092x_1 + 60}{1000t_f x_1^3 + 500x_1^2 + 4x_1 + 20} \right) + \frac{5e^{-t_f} x_1^{t_h/2}}{x_2^{2+t_h} + 1}$ $y^1 = \left(1 - e^{\frac{-1}{2x_2}}\right) \left( \frac{1000t_f x_1^3 + 1900x_1^2 + 2092x_1 + 60}{1000t_f x_1^3 + 500x_1^2 + 4x_1 + 20} \right) + \frac{5e^{-t_f} x_1^{t_h/2}}{x_2^{2+t_h} + 1} + \frac{10x_1^2 + 4x_2^2}{50x_1 x_2 + 10}$ $t_f = 0.2; t_h = 0.3; t_l = 0.1;$	$x_{1,2} \sim N(0.5, 0.1^2)$
5	$y^3 = 25(x_1 - 2)^2 + (x_2 - 2)^2 + (x_3 - 1)^2 + (x_4 - 4)^2 + (x_5 - 1)^2 + (x_6 - 4)^2$ $y^2 = 20(x_1 - 2)^2 + 0.95(x_2 - 2)^2 + 0.8(x_3 - 1)^2 + 1.05(x_4 - 4)^2 + 0.8(x_5 - 1)^2 + 0.7(x_6 - 4)^2$ $y^1 = 15(x_1 - 2)^2 + 0.85(x_2 - 2)^2 + 0.6(x_3 - 1)^2 + 1.35(x_4 - 4)^2 + 0.6(x_5 - 1)^2 + 0.6(x_6 - 4)^2$	$x_{1,\dots,6} \sim N(5, 0.5^2)$
6	$y^3 = x_1^2 + x_2^2 + x_1 x_2 - 4x_1 - 6x_2 + (x_3 - 2)^2 + 4(x_4 - 5)^2 + (x_5 - 3)^2 + 2(x_6 - 1)^2 + 5x_7^2 + 7(x_8 - 3)^2 + 2(x_9 - 2)^2 + (x_{10} - 1)^2 + 11$ $y^2 = 0.8x_1^2 + 0.7x_2^2 + 0.5x_1 x_2 - 4x_1 - 6x_2 + (x_3 - 2)^2 + 4(x_4 - 5)^2 + 1.1(x_5 - 3)^2 + 2(x_6 - 1)^2 + 4.5x_7^2 + 7(x_8 - 3)^2 + 2(x_9 - 2)^2 + (x_{10} - 1)^2 + 10$ $y^1 = 0.5x_1^2 + 0.6x_2^2 + 0.3x_1 x_2 - 3x_1 - 5x_2 + (x_3 - 2)^2 + 4.5(x_4 - 5)^2 + 1.2(x_5 - 3)^2 + 2(x_6 - 1)^2 + 3x_7^2 + 7(x_8 - 3)^2 + 2(x_9 - 2)^2 + (x_{10} - 1)^2 + 10$	$x_{1,\dots,10} \sim N(2.5, 0.2^2)$
7	$y^3 = (x_1 - 1)^2 + \sum_{i=2}^{15} i(2x_i^2 - x_{i-1})^2$ $y^2 = 0.9(x_1 - 1)^2 + 0.9 \sum_{i=2}^{15} i(2x_i^2 - x_{i-1})^2 - \sum_{i=1}^{15} 0.1x_i x_{i+1}$ $y^1 = 0.8(x_1 - 1)^2 + 0.7 \sum_{i=2}^{15} i(2x_i^2 - x_{i-1})^2 - \sum_{i=1}^{15} 0.2x_i x_{i+1}$	$x_{1,\dots,15} \sim N(1, 0.15^2)$
8	$y^3 = (x_1 - 1)^2 + \sum_{i=2}^{20} (x_i - x_{i-1})^2$ $y^2 = 0.8(x_1 - 1)^2 + 0.8 \sum_{i=2}^{20} (x_i - x_{i-1})^2 - \sum_{i=1}^{20} 0.2x_i x_{i+1}$ $y^1 = 0.5(x_1 - 1)^2 + 0.6 \sum_{i=2}^{20} (x_i - x_{i-1})^2 - \sum_{i=1}^{20} 0.5x_i x_{i+1}$	$x_{1,\dots,20} \sim N(1, 0.15^2)$

per layer of the BNN is used; the dropout is set to 0.5, and a batch normal layer is added after each layer of neural network to suppress overfitting.

Table 3 presents the RMSEs of the metamodel constructed using the proposed ML-MFBNN method, Co-Kriging, and H-BNN, and the relative errors of the mean

**Table 2** Sample size

Examples	1	2	3	4	5	6	7	8
$N_1$	10	10	60	20	25	60	100	120
$N_2$	6	6	14	12	15	25	40	60
$N_3$	2	2	/	4	5	10	15	20

**Fig. 6** RMSE with different number of hidden layers and layer neurons (Example 5)

( $e_m$ ) and standard deviation ( $e_s$ ) with respect to the DMCS. It can be seen that the H-BNN method yields large prediction errors because it directly builds a BNN using only HF data. The proposed ML-MFBNN and Co-Kriging can produce results that are much more accurate than the H-BNN because of the fusion of many lower-fidelity data. At the same time, compared with Co-Kriging, the accuracy of the ML-MFBNN is clearly increased, which can be ascribed to the use of BNN as the basic architecture of the metamodel, and the idea of small sample training using meta-learning is adopted. Meanwhile, it is worth noting that for Example 3 whose HF and LF functions behave in opposite directions in four regions ([0 0.13], [0.25 0.38], [0.5 0.63], [0.75 0.88]),

although it is a one-dimensional problem ( $d=1$ ), the accuracy of Co-Kriging is poor; while the proposed MF-MLBNN still exhibits extremely high accuracy, as DNN has strong feature extraction ability for complex problems.

It should be noted that for the proposed ML-MFBNN method, although the number of HF samples ( $n_s$ ) can be reduced, the prediction accuracy of the metamodel cannot be guaranteed if  $n_s$  is too small. Therefore, for network training, especially for high-dimensional ( $d \geq 10$ ) problems, it is required that  $n_s \geq d$ , according to our experience.

Figure 7 shows the training time of the metamodel required by the three methods. For Examples 1–4, the input variable dimension is low ( $d=1$  or 2), and the training time cost of Co-Kriging is smaller than that of the proposed ML-MFBNN method. For Examples 6–8, the input dimension ( $d \geq 10$ ) as well as the functional nonlinearity increase significantly, and the training time cost of the ML-MFBNN is clearly smaller than that of Co-Kriging. For the eight examples tested here, the training time cost of Co-Kriging increases significantly with an increase in the problem dimension, whereas it increases much more slowly for the ML-MFBNN. Therefore, for high-dimensional and nonlinear problems, the advantages of the ML-MFBNN in terms of the accuracy and efficiency are much more evident than those of Co-Kriging.

## 4.2 Test 2

Considering the space limit of the article, only Examples 3, 6 and 8 in Sect. 4.1 are selected for testing to verify the effectiveness of the proposed sequential sampling strategy. As these examples are both mathematical problems, the cost of each multi-fidelity model ( $y_1, y_2, y_3$ ) is artificially set as

**Table 3** RSMEs and relative errors of mean and standard deviation

Examples	ML-MFBNN			H-BNN			Co-Kriging		
	RMSE	$e_m(\%)$	$e_s(\%)$	RMSE	$e_m(\%)$	$e_s(\%)$	RMSE	$e_m(\%)$	$e_s(\%)$
1	0.018	0.54	1.66	0.682	36.41	45.77	0.017	5.26	6.03
2	0.062	0.44	1.27	0.214	62.71	68.32	0.096	4.37	5.83
3	0.0739	1.49	3.75	0.3809	8.26	31.47	0.4399	20.40	37.75
4	0.006	0.28	1.49	1.392	34.03	57.92	0.136	7.24	9.22
5	1.498	0.32	1.03	37.05	53.16	91.49	3.323	8.20	11.07
6	0.952	1.01	3.41	13.89	73.71	103.25	1.671	7.61	6.92
7	0.038	1.95	3.40	1.032	116.83	173.17	0.060	9.37	13.63
8	0.089	2.26	4.22	2.076	103.90	177.26	0.176	8.82	12.47

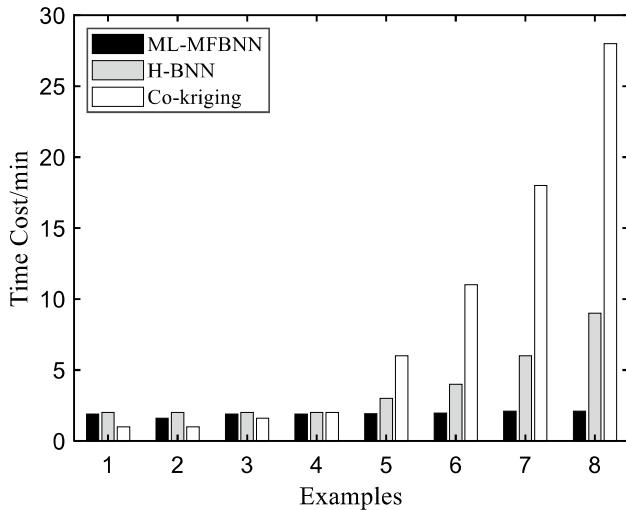


Fig. 7 Training time cost of different methods

$C^1=2$ ,  $C^2=4$ , and  $C^3=12$ . The initial samples are generated using the LHS method, and some HF test samples ( $m=20$ ) are also generated to verify the accuracy of the metamodels by calculating the  $e_m$  and  $e_s$ . The accuracy tolerance is set as  $\Delta=0.03$ , according to the amplitude range of the output response, respectively. Example 3 exhibits “low correlation” between various data sources, and thus the number of initial HF samples is set as  $n_H^0=5d$ , and it is set as  $n_H^0=0.5d$  for the other two examples with “high correlation”. The initial number of samples for each multi-fidelity model, the new samples added in each iteration by the proposed MCS-MFSS, and the final total number of samples are shown in Table 4.

To verify the effectiveness and advantages of the proposed sequential sampling strategy, the samples of each multi-fidelity model are generated by LHS with the same number as the final total number of samples taken by the MCS-MFSS. Based on these samples, the method introduced in Sect. 3.1 is used to construct a multi-fidelity BNN

metamodel (this method is denoted as M2). The errors of the mean ( $e_m$ ) and standard deviation ( $e_s$ ) of the output response obtained using the proposed method (denoted as M1), M2, and the way empirically determines the number of samples for each multi-fidelity model to construct the metamodel using ML-MFBNN as that in Sect. 4.1 (denoted as M3) relative to the DMCS, are also shown in Table 4. In addition, to further verify the advantages of the proposed sampling strategy, the total costs required by the three methods are also compared and shown in Table 4, where C1 ( $C_2=C_1$ ) and C3 denote the total costs required by methods M1 and M3, respectively.

Table 4 shows that, compared to M2 and M3, the metamodel constructed with the proposed adaptive sampling strategy (M1) can yield much more accurate results. Meanwhile, compared to M3, the total cost of M1 is greatly reduced (168 vs. 176, 308 vs. 340 and 676 vs. 720). Our interpretation is that the proposed sampling method can optimize the allocation of the numbers and locations of the multi-fidelity samples, and generate highly cost-effective samples that can improve the accuracy of metamodel with the least possible computational cost. However, for M2 and M3, the samples are selected randomly and empirically.

For the sequential sampling algorithm, the set of initial samples is very critical. However, there is no very scientific and general method at present. According to the recommendations given in Refs. (Zhou et al. 2020; Lv et al. 2020; Aute et al. 2013) and our experience, for general application of the proposed sequential sampling method, the initial number of HF samples should be at least  $n_H^0=0.5d$ , and the number of initial lower-fidelity samples for  $y^t(\mathbf{x})$  can be set as around  $n_t^0=2^{s-t}n_H^0$  ( $t=1, \dots, s-1$ ). In addition, the total number of HF samples should be  $n_H \geq d$  when  $d \geq 10$ . As the meta-learning deep-learning technique is applied in our proposed method, the number of initial and total samples for each fidelity can be reduced compared to the recommendation numbers in the literature (Zhou et al. 2020; Lv et al. 2020; Aute et al. 2013).

Table 4 Results of sampling strategy

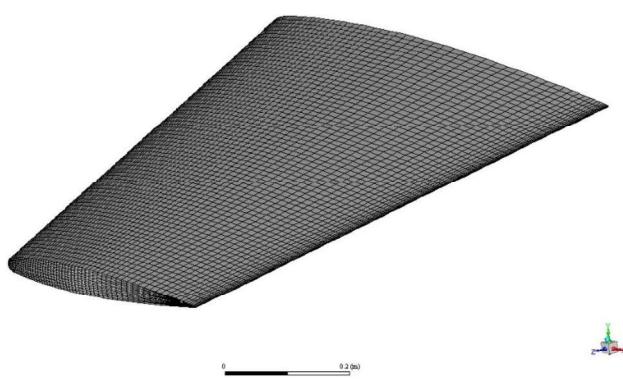
No		Initial	Iteration				Total	C1	C3	$e_m(\%)$			$e_s(\%)$		
			1	2	3	4				M1	M2	M3	M1	M2	M3
3	$N_1$	20	6	8	7	5	46	168	176	0.21	0.92	1.49	0.75	1.97	3.75
	$N_2$	5	4	2	3	5	19								
6	$N_1$	30	3	4	9	10	56	308	340	0.22	3.52	1.01	2.89	6.35	3.41
	$N_2$	10	7	2	0	0	19								
	$N_3$	5	0	4	1	0	10								
8	$N_1$	80	4	3	1	2	90	676	720	1.97	4.37	2.26	2.36	5.74	4.22
	$N_2$	40	6	3	2	7	58								
	$N_3$	10	0	4	7	1	22								

## 5 Engineering application to the ONERA M6 Wing

Aerodynamic optimization is an important component for aerocraft involving multiple disciplines. In this section, the proposed multi-fidelity metamodel method with the adaptive sequential sampling strategy is applied to the robust aerodynamic optimization of the ONERA M6 wing. The geometry of the M6 wing is shown in Fig. 8. The CFD simulation is employed to calculate the lift and drag coefficients of the wing. To ensure the accuracy of the CFD simulation, model validation is first performed CFD at the baseline wing using public wind tunnel experimental data (Schmitt and Charpin 1979). Then, robust optimization is performed based on the validated CFD. Fluent 19.0 is used as the CFD solver, and the computer configuration is 16G RAM, NVIDIA 1660 graphics card, and Intel(R) Core(TM) i5-10400F CPU. For meta-learning training, the number of iterations is  $K = 120$ , the number of training tasks is  $k = 10$ , and the dropout of the BNN is 0.5.

### 5.1 Model validation

In the CFD numerical simulation, there are many uncertainties that affect the CFD output response. For model validation under deterministic conditions, the output response is



**Fig. 8** Geometry of M6 wing

**Table 5** Mesh convergence test

	Number of grids		Lift-to-drag ratio	Computational time	Fidelity
	Far-field boundary	Wing section and boundary			
G1	800	250	14.739	61 min	–
G2	800	275	15.077	82 min	Low-fidelity
G3	1000	300	15.253	95 min	–
G4	1000	325	15.640	116 min	–
G5	1000	350	15.641	142 min	High-fidelity

likely very sensitive to some uncertainties. Therefore, it is necessary to perform model validation by considering the uncertainties (Han and Görtz 2012). As the CFD simulation is often time-consuming and model validation requires a large number of CFD evaluations, a multi-fidelity BNN metamodel of the CFD output response is constructed to reduce the computational cost. An incoming flow with Mach number  $Ma = 0.8359$  and angle of attack  $\alpha = 3.06^\circ$  are considered for this problem. The CFD preprocessing software, Gambit, is used to generate the mesh.

First, the mesh convergence test is performed on the baseline airfoil model, in which five different grid configurations (G1,...,G5) are considered. The test results are shown in Table 5, where it can be seen that the lift-to-drag ratio for G5 is very close to that of G4, indicating that the grid configuration G5 converges. Here only two levels of fidelity are considered as is commonly considered in literature (Zhou et al. 2020). Therefore, a CFD simulation with 1000 grids in the far-field mesh boundaries and 350 grids between the wing section and the boundary is selected as the HF model. In addition, a CFD simulation with 800 grids in the far-field boundaries and 275 grids between the wing section and boundary is selected as the LF model.

For the aerodynamic problem studied here, the SA turbulence model is employed considering its good performance and fast convergence speed in simulating the current flow (Shah et al. 2020). Generally, uncertainties in the turbulence model are one of the most important categories of uncertainty, and the CFD results with default closure coefficients of the turbulence model may not be in good agreement with the experimental data (He et al. 2020). Here, the closure coefficients are considered to be uncertain and must be calibrated during model validation.

Public data from wind tunnel experiments (Schmitt and Charpin 1979) provided the pressure coefficients of five sections ( $y/b = 20\%, 44\%, 65\%, 80\%$ , and  $90\%$  of the wing along the wingspan starting from the wing root) on the wing. Clearly, a correlation exists among the pressure coefficients of these sections, which can be used to improve the accuracy of the aerodynamic prediction. Therefore, to reduce the complexity of the metamodel building and improve the prediction accuracy, a multi-branch BNN with the closure

coefficients of the SA turbulence model as the inputs and the pressure coefficients of the five sections as the outputs is constructed using the proposed ML-MFBNN, as opposed to constructing a BNN metamodel for each section. The network is composed of three hidden layers with 20 neurons in each layer, in which the input layer and the first two hidden layers are backbones and the last hidden layer and output layer are branches.

$N_1$  and  $N_2$  represent the numbers of HF and LF samples, respectively. The cost of the multi-fidelity model is set as the computational time of the CFD simulation, that is,  $C^1 = 82$  min and  $C^2 = 142$  min. Twenty HF CFD samples are generated to verify the accuracy of the metamodel, and the RMSE is calculated. The accuracy tolerance is set to  $\Delta = 0.005$ . Table 6 shows the numbers of the initial HF and LF CFD samples, the samples added by sequential sampling, and the total samples used. As the network relationship is much more complicated here,  $n_H^0 = d$ .

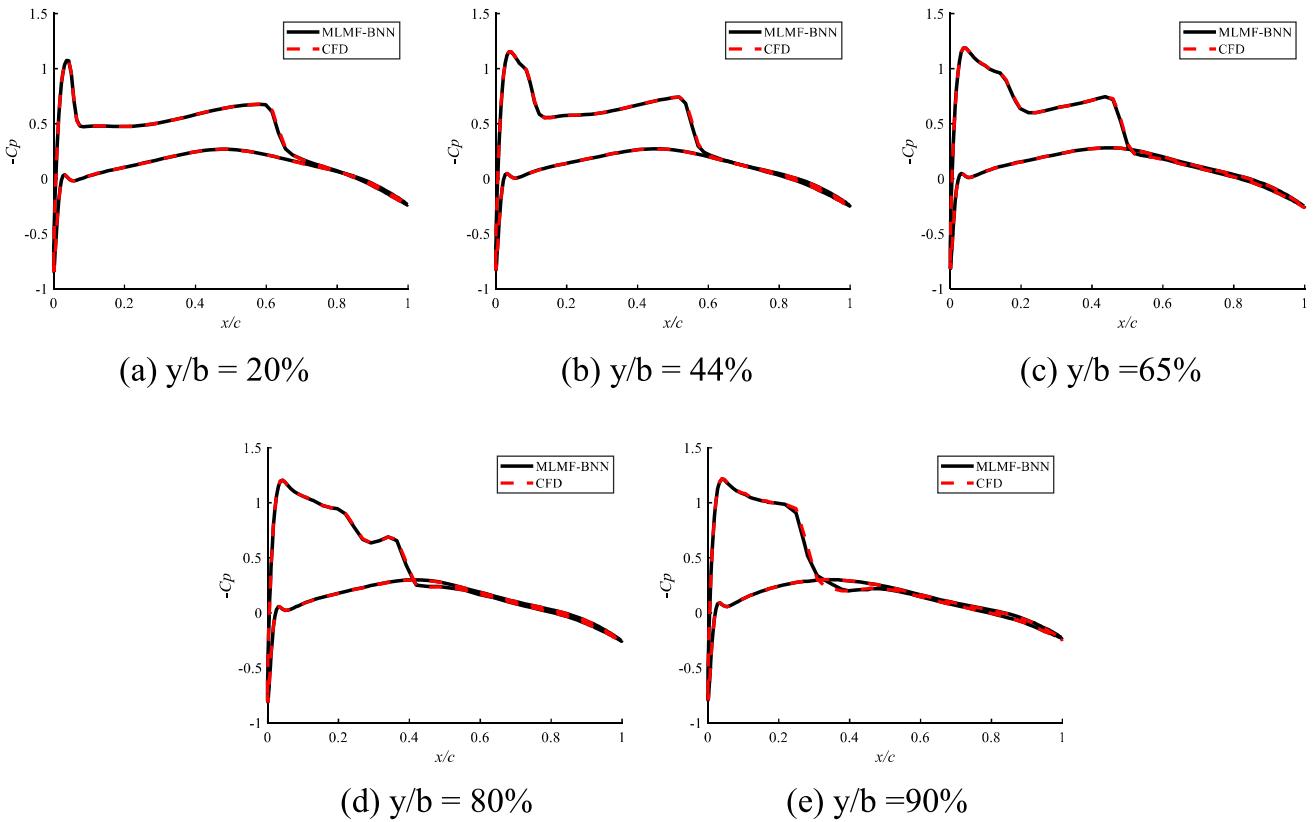
Figure 9 shows the predicted pressure coefficient curves obtained using the proposed method (M1). Meanwhile, to further verify the effectiveness and advantages of the sequential sampling method, M2 defined in Sect. 4.2 is also employed to construct a multi-branch BNN metamodel. The RSMEs of M1 and M2 are also shown in Table 6.

As shown in Fig. 9, the predicted pressure curves of the five sections obtained using M1 show good agreement with those obtained with the HF CFD, indicating the high accuracy of the constructed metamodel. Table 6 shows that the proposed adaptive sampling strategy can significantly improve the prediction accuracy of the metamodel compared to sampling with the LHS.

Based on the constructed metamodel of the HF CFD, uncertainty propagation and parameter calibration are performed to validate the model. The initial ranges of the closure coefficients of the turbulence model are presented in Table 7. To reduce the difficulty of parameter calibration

**Table 6** Numbers of samples and RSMEs

	Initial	Iteration						Total	RMSE	
		1	2	3	4	5	6		M1	M2
$N_1$	18	7	10	9	8	9	8	69	0.00428	0.01386
$N_2$	6	3	0	1	2	1	2	15		



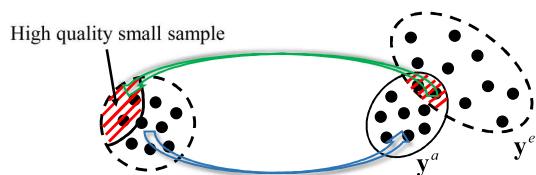
**Fig. 9** Pressure coefficients predicted by multi-fidelity multi-branch BNN

**Table 7** Initial ranges and default values of closure coefficients

	$C_{b1}$	$C_{b2}$	$C_{v1}$	$C_{w2}$	$C_{w3}$	$\sigma$
Minimum value	0.0542	0.2488	2.84	0.12	0.8	0.2668
Maximum value	0.2168	0.9952	11.36	0.48	3.2	1.0672
Default Value	0.1355	0.622	7.1	0.3	2	0.667

**Table 8** Sobol's Index of closure coefficients

Section(y/b)	$C_{b1}$	$C_{b2}$	$C_{v1}$	$C_{w2}$	$C_{w3}$	$\sigma$
20%	0.00%	0.19%	98.34%	0.04%	1.34%	0.09%
44%	0.01%	0.14%	99.02%	0.03%	0.73%	0.07%
65%	0.01%	0.16%	98.61%	0.04%	1.09%	0.09%
80%	0.01%	0.19%	98.15%	0.04%	1.53%	0.08%
90%	0.02%	0.58%	90.51%	0.04%	8.77%	0.08%

**Fig. 10** Schematic of parameter calibration using approximate Bayesian inference method

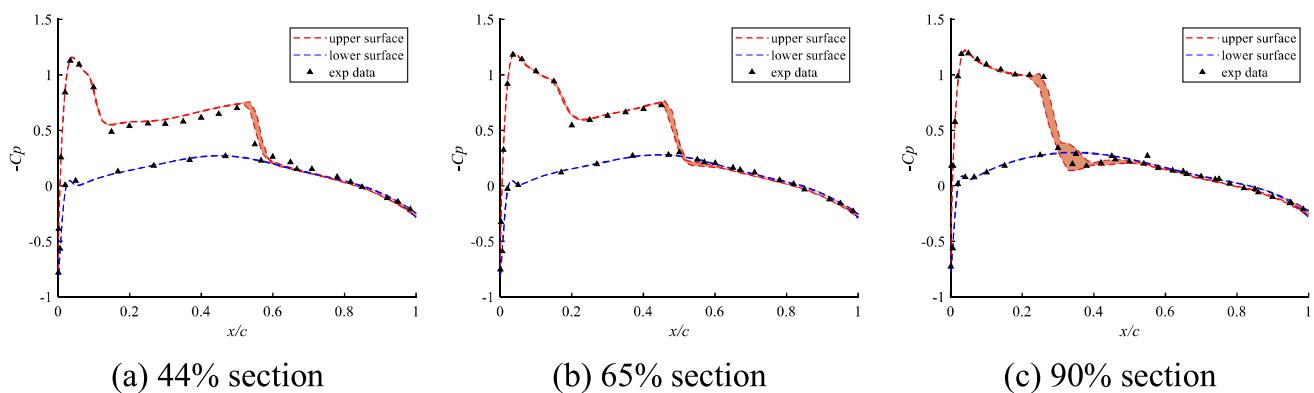
and improve the convergence speed, a variance-based sensitivity analysis of the pressure coefficients based on the constructed metamodel is first conducted to determine the most important closure coefficients. The Sobol's indices of the closure coefficients corresponding to the five sections are presented in Table 8.

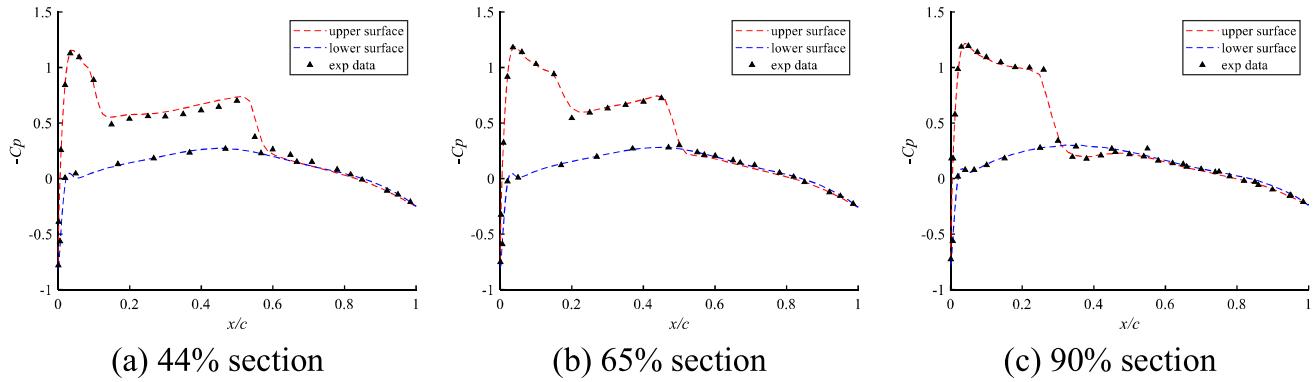
As shown in Table 8, for all the five sections,  $C_{v1}$  has the largest impact on the pressure coefficients, which is much greater than the effect of the other coefficients. Therefore, parameter calibration is performed only on  $C_{v1}$ , and the other coefficients are fixed at the default values.

In this study, the approximate Bayesian inference method (Deng et al. 2014) is adopted for parameter calibration, the basic principle of which is illustrated in Fig. 10.  $y^a$  denotes output samples generated by the constructed multi-fidelity metamodel of HF CFD above, given the range of input ( $C_{v1}$ ); and  $y^e$  means the output samples from experiment at the same input sets as those of  $y^a$ .  $y^a$  that are closer to  $y^e$  are selected, based on which the values of  $C_{v1}$  can be derived (denoted as high-quality small sample) using input–output one-to-one correspondence. Subsequently, the range of  $C_{v1}$  is updated based on the high-quality small sample.

After four iterations, the calibration process is stopped and  $C_{v1}$  is corrected to the interval [6.9141, 6.9161]. Based on the validated CFD model, the pressure coefficients are shown in Figs. 11 and 12, which are significantly improved compared with the results before validation and exhibit good agreement with the experimental data. These results demonstrate the effectiveness and advantages of the proposed ML-MFBNN and adaptive multi-fidelity sampling methods.

Based on the validation results, the closure coefficient  $C_{v1}$  is taken as the median of its calibrated interval, and the

**Fig. 11** Pressure coefficients before CFD validation



**Fig. 12** Pressure coefficients after CFD validation

rest are fixed at the default value shown in Table 7 for the subsequent CFD simulations.

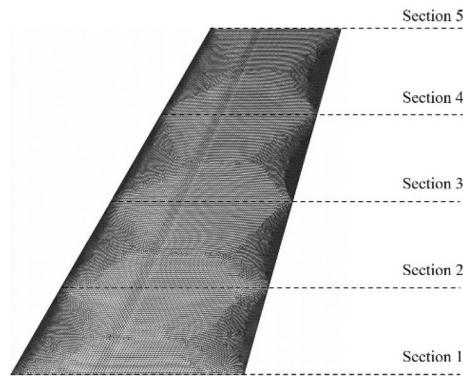
## 5.2 Robust shape optimization

The shape optimization problem considered here aims to minimize the drag coefficient ( $C_d$ ) subjected to constraints on the lift coefficient ( $C_l$ ) and maximum thickness of the wing by optimizing the wing geometry. The same incoming flow condition as that in subSect. 5.1 is considered here. It is considered that uncertainties exist in the incoming flow, and  $Ma$  and  $\alpha$  follow uniform distributions that vary within  $\pm 0.02$  and  $\pm 0.2^\circ$ , respectively, with respect to their mean values, that is,  $Ma \sim U(0.8159, 0.8559)$   $\alpha \sim U(2.86^\circ, 3.26^\circ)$ . The robust shape optimization problem can be formulated as follows:

$$\begin{aligned} \text{Min } F &= -\mu_{Cd} + 3\sigma_{Cd} \\ \text{s.t. } &\left\{ \begin{array}{l} \mu_{Cl} - 3\sigma_{Cl} \geq Cl^0 \\ t_{\max} \geq t_{\max}^0 \end{array} \right. \end{aligned} \quad (13)$$

where  $\mu$  and  $\sigma$  denote the mean and standard deviation, respectively, and  $Cl^0$  ( $Cl^0 = 0.2909$ ) and  $t_{\max}^0$  ( $t_{\max}^0 = 0.09785$ ) are the lift coefficient and maximum thickness of the baseline M6 wing, respectively.

Five sections with equal intervals along the wingspan are selected (see Fig. 13), which lie at  $y/b = 0\%$ ,  $25\%$ ,  $50\%$ ,  $75\%$ , and  $100\%$  of the wing along the wingspan starting from the wing root. In each section, ten control points (five at the top and five at the bottom) are selected, resulting in 50 control points to control the wing geometry. The horizontal positions of the 50 control points are fixed at the chord length (10%, 30%, 50%, 70%, and 90% along the chord of the wing, starting from the leading edge), and their vertical positions are changeable. For simplicity, the vertical positions of the 50 points are denoted by  $y_1, \dots, y_{50}$ . The design space of  $y_1, \dots, y_{50}$  is specified by expanding and shrinking it by 25% with

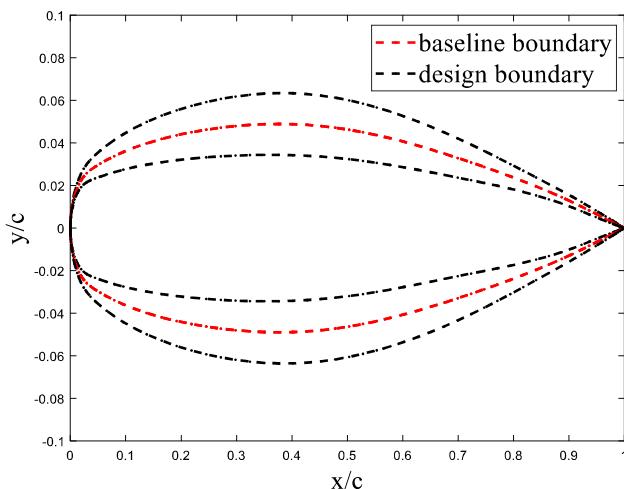


**Fig. 13** Five sections on the Wing

respect to the baseline. Figure 14 shows the design variables and baseline boundaries of the wing root surface ( $y/b = 0\%$ ).

Similarly, to reduce the computational cost of robust optimization, a multi-output BNN is constructed for the lift and drag coefficients using the ML-MFBNN and adaptive sampling strategy. The network inputs are  $y_1, \dots, y_{50}$ ,  $Ma$ , and  $\alpha$ . The network is composed of three hidden layers with 30 neurons in each layer. Forty HF CFD samples are generated based on the LHS to test the accuracy of the metamodels with  $\Delta = 0.3$ . The grid configurations for the HF and LF CFD models are the same as those described in Sect. 5.1. The number of initial samples and final total number of samples after 12 iterations of sequential sampling are shown in Table 9. The RMSEs for M1 and M2 are also listed in Table 9. The results in Table 9 show that the prediction accuracy achieved with the proposed adaptive sampling strategy can be significantly improved compared to sampling with the LHS, which is in good agreement with the HF CFD data. These results further demonstrate the effectiveness of the proposed methods.

Then, aerodynamic robust shape optimization is performed on the above-constructed metamodels. The optimal results are shown in Table 10, where  $\mu_{Cd}, \sigma_{Cd}, \mu_{Cl}, \sigma_{Cl}$ ,



**Fig. 14** Design and baseline boundaries ( $y/b=0\%$ )

**Table 9** Numbers of samples and RSMEs

	Initial	Total	RMSE	
			M1	M2
$N_1$	100	168	0.0115	0.0043
$N_2$	30	82		

**Table 10** Optimal results

	Baseline	DO	RO
$\mu_{cd}$	0.01886	0.01637	0.01748
$\sigma_{cd}$	0.0018	0.00092	0.00089
$\mu_{cl}$	0.2894	0.2825	0.2951
$\sigma_{cl}$	0.0112	0.0119	0.0085
$P(Cl \geq Cl^0)$	52%	38%	85%

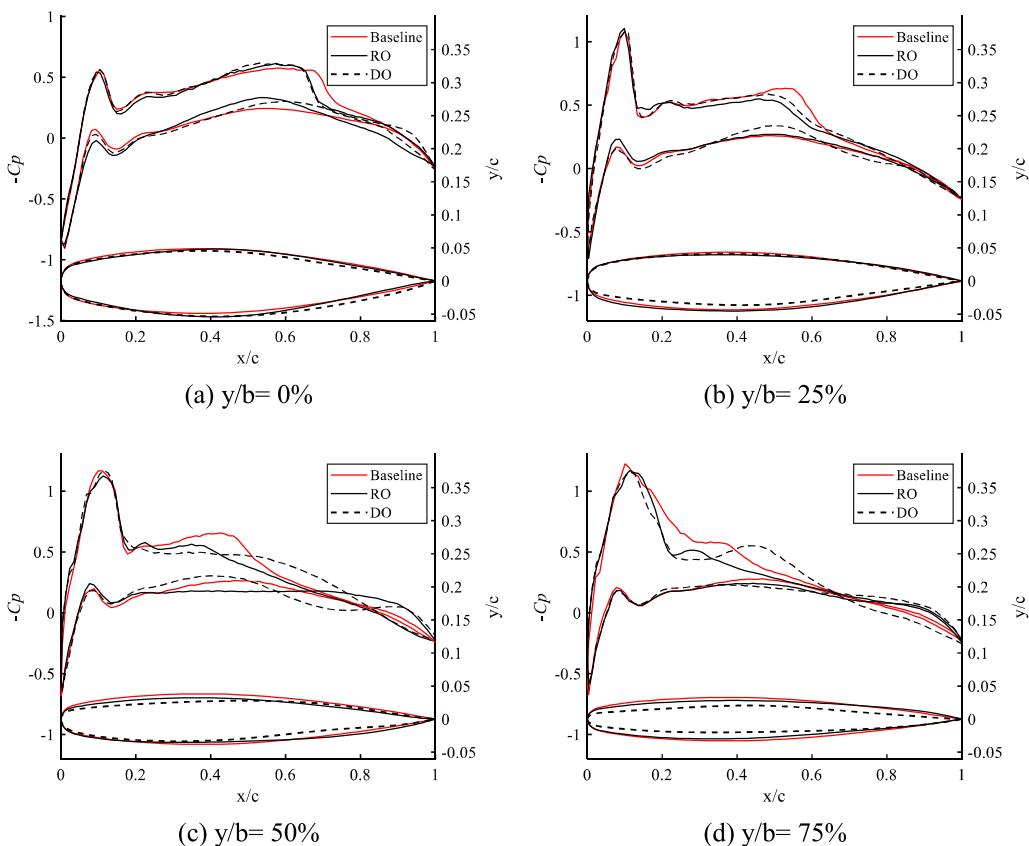
and  $P(Cl \geq Cl^0)$  are calculated using MCS (500 runs) by substituting the obtained optimal design variables into the CFD simulation, considering the same uncertainties from  $Ma$  and  $\alpha$ .  $P(Cl \geq Cl^0)$  represents the probability of  $Cl \geq Cl^0$ , where a larger value corresponds to a more reliable design solution. Deterministic optimization (denoted by DO) based on the two constructed metamodels without considering any uncertainties is also conducted for comparison. The mean values of the drag coefficients obtained with both the optimizations are all reduced compared with that of the baseline wing with the maximum decrease observed for DO. However, robust optimization (denoted by RO) yields smaller variations in both drag ( $\sigma_{cd}$ ) and lift ( $\sigma_{cl}$ ) coefficient than those of DO and the baseline. As

uncertainties are not considered during DO, the variations in the lift coefficients are the largest (see  $\sigma_{cl}$ ), and the lift coefficients have difficulty satisfying the constraints (see  $P(Cl \geq Cl^0) = 38\%$ ), exhibiting weak robustness and reliability to uncertainties.

Figure 15 shows the pressure coefficients and wing section geometry at four sections corresponding to the RO, baseline, and DO. The pressure coefficient curves for both optimizations do not vary as drastically as those of the baseline, especially for the regions close to the leading edge ( $x/c = 0$ ). This indicates that the pressure variations for the four sections decrease via optimization. Therefore, the shock wave is weakened, which reduces the drag. Compared to DO, the fluctuations in the pressure coefficients and gaps between the upper and lower curves of the pressure coefficients are clearly reduced, indicating that the lift and drag variation observed with RO are smaller than those of DO. These results show good agreement with the results presented in Table 10, further demonstrating the effectiveness and advantages of the proposed ML-MFBNN and adaptive sampling methods.

## 6 Conclusions

To reduce the computational cost of multi-fidelity metamodel methods for high-dimensional and high-nonlinear approximation, we proposed a multi-fidelity Bayesian neural network (BNN) metamodel method based on meta-learning (ML-MFBNN) in conjunction with the adaptive multi-fidelity sampling strategy. With the proposed method, a BNN is constructed for each multi-fidelity model, which are fused using meta-learning theory without establishing any link models. Moreover, the cost-effective samples for the multi-fidelity models are selected sequentially to improve the metamodel accuracy with the least achievable computational cost. The results from the mathematical example tests showed that the proposed ML-MFBNN method can reduce the prediction error by up to 75% and the training time by up to 92% compared with the Co-Kriging method, especially for high-dimensional problems. Meanwhile, using the proposed multi-fidelity sampling strategy, the computational cost can be reduced by 9%, while the accuracy is improved by 70% compared with empirical sampling. Model validation of the CFD for the M6 wing and subsequent aerodynamic robust optimization further demonstrated the effectiveness of the proposed multi-fidelity metamodel and sequential sampling methods.



**Fig. 15** Pressure coefficients and wing section geometry

**Acknowledgements** The work was supported by National Natural Science Foundation of China [grant number 52175214] and Basic Research Program of Equipment Development Department [grant number 514010103-302].

**Funding** Funding National Natural Science Foundation of China, 52175214; The basic research program, 514010103-302.

## **Declarations**

**Conflict of interest** The authors declare that they have no conflict of interest.

**Replication of results** The results shown in the manuscript can be reproduced. Example in Sect. 4.1 was uploaded to as supplementary material. The remaining examples are easy to implement by changing the response functions and samples based on the codes provided to obtain the results shown in the manuscript.

**Ethical approval** This article does not contain any studies with human participants or animals performed by any of the authors.

## References

Aute V, Saleh K, Abdelaziz O, Azarm S, Radermacher R (2013) Cross-validation based single response adaptive design of experiments

for Kriging metamodeling of deterministic computer simulations. Struct Multidisc Optim 48(3):581–605. <https://doi.org/10.1007/s00158-013-0918-5>

Bailly J, Bailly D (2019) Multifidelity aerodynamic optimization of a helicopter rotor blade. AIAA J 57(8):3132–3144. <https://doi.org/10.2514/1.J056513>

Blundell C, Cornebise J, Kavukcuoglu K, Wierstra D (2015) Weight uncertainty in neural network. Int Conf Mach Learning 37:1613–1622. <https://doi.org/10.5555/3045118.3045290>

Bouhlel MA, He S, Martins JRRA (2020) Scalable gradient-enhanced artificial neural networks for airfoil shape design in the subsonic and transonic regimes. Struct Multidisc Optim 61(4):1363–1376. <https://doi.org/10.1007/s00158-020-02488-5>

Chakraborty S (2021) Transfer learning based multi-fidelity physics informed deep neural network". J Comput Phys 426:109942. <https://doi.org/10.1016/j.jcp.2020.109942>

Chen S, Jiang Z, Yang S, Chen W (2017) Multi-model fusion based sequential optimization. AIAA J 55(1):241–254. <https://doi.org/10.2514/1.J054729>

Deng Z, Bi S, Atamturktur S (2014) Stochastic model updating using distance discrimination analysis. *Chin J Aeronaut* 27(5):1188–1198. <https://doi.org/10.1016/j.cja.2014.08.008>

Ellis AG, Iskandar R, Schmid CH, Wong JB, Trikalinos TA (2020) Active learning for efficiently training emulators of computationally expensive mathematical models. *Stats Med* 39(25):3521–3548. <https://doi.org/10.1002/sim.8679>

Eweis-Labolle JT, Oune N, Bostanabad R (2022) Data fusion with latent map Gaussian processes. J Mech Des 144(9):091703. <https://doi.org/10.1115/1.4054520>

- Finn, C., Abbeel, P., and Levine, S. 2017. "Model-agnostic meta-learning for fast adaptation of deep networks," Proceedings of the 34th International Conference on Machine Learning. 70: 1126–1135. <https://doi.org/10.5555/3305381.3305498>
- Forrester, A. I. J., Sobester, A., and Keane, A. J. 2007. Multi-fidelity optimization via surrogate modelling. Proceedings of the royal society a: mathematical, physical and engineering sciences. 463(2088): 3251–3269. <https://doi.org/10.1098/rspa.2007.1900>
- Halder R, Damodaran M, Khoo BC (2020) Deep learning based reduced order model for airfoil-gust and aeroelastic interaction. AIAA J 58(10):4304–4321. <https://doi.org/10.2514/1.J059027>
- Han Z, Götz S (2012) Hierarchical kriging model for variable-fidelity surrogate modeling. AIAA J 50(9):1885–1896. <https://doi.org/10.2514/1.J051354>
- He X, Zhao F, Vahdati M (2020) Uncertainty quantification of Spalart-Allmaras turbulence model coefficients for simplified compressor flow features. J Fluids Eng 142(9):091501. <https://doi.org/10.1115/1.4047026>
- Huang D, Allen TT, Notz WI, Miller RA (2006) Sequential kriging optimization using multiple-fidelity evaluations. Struct Multidiscip Optim 32(5):369–382. <https://doi.org/10.1007/s00158-005-0587-0>
- Jivani A, Huan X, Safta C, Zhou BY, Gauger NR (2021) Uncertainty quantification for a turbulent round jet using multifidelity karhunen-loeve expansions. AIAA Scitech. <https://doi.org/10.2514/6.2021-1367>
- Jones DR, Schonlau M, Welch WJ (1998) Efficient global optimization of expensive black-box functions. J Global Optim 13(4):455–492. <https://doi.org/10.1023/A:1008306431147>
- Lee SH, Chen W (2009) A comparative study of uncertainty propagation methods for black-box type problems. Struct Multidiscip Optim 37(3):239–253. <https://doi.org/10.1007/s00158-008-0234-7>
- Li J, Zhang M, Martins JR, Shu C (2020) Efficient aerodynamic shape optimization with deep-learning-based geometric filtering. AIAA J 58(10):4243–4259. <https://doi.org/10.2514/1.J059254>
- Liu Y, Chen S, Wang F, Xiong F (2018) Sequential optimization using multi-level cokriging and extended expected improvement criterion. Struct Multidiscip Optim 58(3):1155–1173. <https://doi.org/10.1007/s00158-018-1959-6>
- Liu H, Ong YS, Shen X, Cai J (2020) When Gaussian process meets big data: a review of scalable GPs. IEEE Trans Neural Networks Learning Syst 31(11):4405–4423. <https://doi.org/10.1109/TNNLS.2019.2957109>
- Lv L, Shi M, Song X, Sun W, Zhang J (2020) A fast-converging ensemble infilling approach balancing global exploration and local exploitation: the go-inspired hybrid infilling strategy. J Mech Des 142(2):021403. <https://doi.org/10.1115/1.4044112>
- Meng XH, Karniadakis GEA (2020) A composite neural network that learns from multi-fidelity data: application to function approximation and inverse PDE problems. J Comput Phys 401:109020. <https://doi.org/10.1016/j.jcp.2019.109020>
- Meng X, Babaei H, Karniadakis GE (2021) Multi-fidelity Bayesian neural networks: algorithms and applications. J Comput Phys 438:110361. <https://doi.org/10.1016/j.jcp.2021.110361>
- Paiva RM, Crawford C, Suleman A (2014) Robust and reliability-based design optimization framework for wing design. AIAA J 52(4):711–724. <https://doi.org/10.2514/1.J052161>
- Panaretos VM, Zemel Y (2019) Statistical aspects of wasserstein distances. Ann Rev Stat App 6:405–431. <https://doi.org/10.1146/annurev-statistics-030718-104938>
- Park GJ, Lee TH, Lee KH, Hwang KH (2006) Robust design: an overview. AIAA J 44(1):181–191. <https://doi.org/10.2514/1.13639>
- Parussini L, Venturi D, Perdikaris P, Karniadakis GE (2017) Multi-fidelity Gaussian process regression for prediction of random fields. J Comput Phys 336:36–50. <https://doi.org/10.1016/j.jcp.2017.01.047>
- Perdikaris, P., Raissi, M., Damianou, A., Lawrence, N. D., and Karniadakis, G. E. 2017. Nonlinear information fusion algorithms for data-efficient multi-fidelity modelling," Proceedings of the Royal Society A: Mathematical, Physical and Engineering Science. <http://dx.doi.org/https://doi.org/10.1098/rspa.2016.0751>
- Ren C, Xiong F, Wang F, Mo B, Hu Z (2021) A maximum cost-performance sampling strategy for multi-fidelity PC-Kriging. Struct Multidiscip Optim 64(6):3381–3399. <https://doi.org/10.1007/s00158-021-02994-0>
- Rodríguez P, Laradji I, Drouin A, Lacoste A (2020) "Embedding propagation: smoother manifold for few-shot classification. European Conf Comput vis. [https://doi.org/10.1007/978-3-030-58574-7\\_8](https://doi.org/10.1007/978-3-030-58574-7_8)
- Schmitt, V., and F. Charpin, "Pressure distributions on the ONERA-M6-Wing at transonic mach numbers," Experimental Data Base for Computer Program Assessment. AGARD AR 138, May 1979. <https://www.grc.nasa.gov/www/wind/valid/m6wing/m6wing.html>
- Shah MZM, Basuno B, Abdullah A (2020) "Comparative study on several type of turbulence model available in ansys-fluent software for ONERA m6 wing aerodynamic analysis. J Adv Mech Eng App 1(1):9–19. <https://doi.org/10.30880/jamea.2020.01.01.002>
- Snoek J, Rippel O, Swersky K, Kiros R, Satish N (2015) Scalable bayesian optimization using deep neural networks. Int Conf Mach Learning, PMLR. <https://doi.org/10.48550/arXiv.1502.05700>
- Tao J, Sun G (2019) Application of deep learning based multi-fidelity surrogate model to robust aerodynamic design optimization. Aerosp Sci Technol 92:722–737. <https://doi.org/10.1016/j.ast.2019.07.002>
- Tripathy RK, Bilionis I (2018) Deep UQ: learning deep neural network surrogate models for high dimensional uncertainty quantification. J Comput Phys 375:565–588. <https://doi.org/10.1016/j.jcp.2018.08.036>
- Wang H, Yeung DY (2021) A survey on Bayesian deep learning. ACM Comput Surv (CSUR) 53(5):1–37. <https://doi.org/10.1145/3409383>
- Wang F, Xiong F, Chen S, Song J (2019) Multi-fidelity uncertainty propagation using polynomial chaos and Gaussian process modeling. Struct Multidiscip Optim 60(4):1583–1604. <https://doi.org/10.1007/s00158-019-02287-7>
- Xu J, Du Q (2020) Learning transferable features in meta-learning for few-shot text classification. Pattern Recogn Lett 135:271–278. <https://doi.org/10.1016/j.patrec.2020.05.007>
- Yin T, Zhu HP (2020) An efficient algorithm for architecture design of Bayesian neural network in structural model updating. Comput-Aided Civil Infra Eng 35(4):354–372. <https://doi.org/10.1111/mice.12492>
- Zanjani Foumani S, Shishehbor M, Yousefpour A, Bostanabad R., "Multi-Fidelity Cost-Aware Bayesian Optimization", 2022. <https://doi.org/10.48550/arXiv.2211.02732>
- Zhang Y, Tao S, Chen W, Daniel W (2020) A latent variable approach to Gaussian process modeling with qualitative and quantitative factors. Technometrics 62(3):291–302. <https://doi.org/10.1080/00401706.2019.1638834>
- Zhang X, Xie F, Ji T, Zhu Z, Zheng Y (2021a) Multi-fidelity deep neural network surrogate model for aerodynamic shape optimization. Comput Method Appl Mech Eng 373:113485. <https://doi.org/10.1016/j.cma.2020.113485>
- Zhang X, Xie F, Ji T, Zhu Z, Zheng Y (2021b) Multi-fidelity deep neural network surrogate model for aerodynamic shape optimization. Comput Methods App Mech Eng 373:113485. <https://doi.org/10.1016/j.cma.2020.113485>
- Zhang S, Liang P, Pang Y, Li J, Song X (2022) Multi-fidelity surrogate model ensemble based on feasible intervals. Struct Multidiscip Optim 60(4):1583–2160. <https://doi.org/10.1007/s00158-022-03329-3>
- Zhou Q, Wang Y, Choi SK, Jiang P, Shao X, Hu J (2017) A sequential multi-fidelity metamodeling approach for data regression.

Knowl-Based Syst 134(15):199–212. <https://doi.org/10.1016/j.knosys.2017.07.033>

Zhou Q, Yang Y, Song X, Han Z, Cheng Y, Hu J, Shu L (2020) Survey of Multi-fidelity surrogate models and their applications in the design and optimization of engineering equipment. *J Mech Eng* 56(24):219–245. <https://doi.org/10.3901/JME.2020.24.219>

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.