IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 10 January 2023, accepted 23 January 2023, date of publication 30 January 2023, date of current version 8 February 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3240602

RESEARCH ARTICLE

Multi-Objective Surrogate Modeling Through Transfer Learning for Telescopic Boom Forklift

JINGLIANG LIN[1], HAIYAN LI[2], YUNBAO HUANG[2], JUNJIE LIANG[2], SHENG ZHOU[2], ZEYING HUANG[2], AND GUIMING LIANG[2]

[1]School of Mechanical Engineering, Guangdong Ocean University, Zhanjiang 524088, China
[2]College of Mechanical and Electrical Engineering, Guangdong University of Technology, Guangzhou 510006, China

Corresponding author: Haiyan Li (cathylhy@gdut.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 51775116 and Grant 51975125, and in part by the Program for Scientific Research Start-Up Funds of Guangdong Ocean University under Grant 060302062108.

ABSTRACT Simulation and optimization methods have been widely used in forklift design due to their cost-effectiveness. However, this type of method involves challenges such as the accuracy of the simulation model and the simulation solution time. These challenges reduce the stability and precision of the surrogate model and hence generate further optimization errors. In this paper, a multi-objective surrogate modeling (MSM) method for telescopic boom forklifts based on closed-loop transfer learning is proposed in order to solve these challenges. The MSM consists of the following two steps: to pre-train an initial deep neural network model (deep model) with a large amount of existing simulation data from the same type of forklift and to transfer the model with a small amount of measurement data collected on the current forklift. A general framework for deep neural network (DNN) training is introduced to improve the approximation ability of the initial model. Moreover, a novel uncertainty-analysis-based sampling method is suggested for measurement data development, and combined with transfer learning to form a closed-loop mode to improve the stability of the final model. The superiority of MSM is demonstrated through comparative studies with the fine-tuning method on a telescopic boom forklift with two objectives. The experimental results show that the Correlation coefficient (R) of the deep model can reach 0.9971 by using only 80 sets of training data. In addition, it can also achieve an improvement of at least a 13.25% reduction in Root Mean Squared Error (RMSE) and a 9.19% reduction on average in Maximum Absolute Error (MAE), as well as stronger robustness compared to the benchmarks. Furthermore, it will provide a valuable reference for the simulation optimization of complex electromechanical products.

INDEX TERMS Telescopic boom forklift, surrogate model, transfer learning, uncertainty analysis, hyper-parameter optimization.

I. INTRODUCTION

The forklift is a complex modern engineering vehicle. It has been widely used in factories, stations, docks, gardens, warehouses, and other places due to the advantages of small size, flexible work, and large load. According to the operation mode, it is classified into pallet forklift, stack forklift, balance heavy forklift, heavy-duty forklift, telescopic boom forklift, etc. Among them, the telescopic boom forklift is popular since its telescopic boom can lift to 17-30 meters, making it ideal for entering narrow spaces [1]. To improve the performance of forklifts and reduce the cost of their physical prototype trial production, computer simulation-based methods are often used in its design processes, such as mechanical properties design [2], [3], safety improvement [4], [5], and comfort improvement [6], [7]. A variety of performance requirements need to consider more factors in forklift design. At present, the forklift has become a multidisciplinary coupling system integrating mechanical, control, electronics, hydraulic, etc., which makes

The associate editor coordinating the review of this manuscript and approving it for publication was Giambattista Gruosso.

it challenging to build a simulation model that can accurately reflect its real response characteristics. In particular, the solution time of the simulation model will be very long.

To improve the efficiency of simulation optimization, the surrogate model (or response surface) method based on computer experimental design is usually used in industry and academia, such as Sparse response surface [8], [9], Support vector machine [10], [11], Kriging [12], [13], and Radial basis function [14], [15]. This type of approach regards the simulation model as a black-box function, and a surrogate model is then constructed for the black-box function using a small amount of simulation data. On this basis, an optimization algorithm is employed to search globally in the design space to reduce the number of simulation model calls as many as possible [16]. However, it is difficult to construct a precise surrogate model for the high-dimensional nonlinear simulation model based on a small amount of data, which may lead to deviations between the optimization results and the actual situation.

The surrogate model can be regarded as a predictive model in nature [17]. In the field of machine learning, using neural networks with sufficient depth and width, as well as a large amount of training data, a predictive model (i.e., deep model) can be constructed through a deep learning algorithm to accurately characterize the high-order nonlinear relationship between multiple inputs and multiple outputs implied in the data [18], [19]. However, in many application scenarios, acquiring sufficient labeled data is often prohibitive, such as in computer vision [20] and some complex systems [21], [22]. From another perspective, the above studies mostly focused on developing new surrogate models and/or applying existing surrogate models to practical problems. The surrogate for a new task is often built from scratch and the knowledge gained from previous surrogate modeling for similar tasks is neglected [23].

Similar ideas for knowledge transfer can be found in transfer learning [24], [25]. Among them, domain adaptation [26], [27] is an important branch that has been developed. It reduces the need for expensive data in the target domain by reusing models or data from similar domains. Existing models are usually completely updated [28], [29] or retrained in the last few layers [30], [31]. Their successful applications can be seen in many areas [32]. Especially for image recognition and classification tasks, there are already many large marked image databases [33], [34] and some pre-trained models available for downloads, such as AlexNet [35], VGGNet [36], and Inception [37].

In the engineering optimization community, the above research does not apply to the surrogate modeling of neural networks as function approximators. To the best of our knowledge, there are no pre-trained models or general databases available. Although a few works have explored transfer learning in an engineering context, little has been done on the problem of how to obtain data for pre-training the initial model [38], [39], [40], [41], [42]. Some attempts to plan data directly from simulation models may keep modeling still limited to the efficiency problem [43], [44]. As for serially developed complex electromechanical products, planning a large amount of simulation data seems unnecessary due to the high similarity between new products and existing products [45]. Taking the telescopic boom forklift as an example, the new product in design is usually a variant of the existing product, both of which the product structure, operation principle, and performance response characteristics are similar, resulting in their measurement data isomorphism or close to isomorphism. Based on the above analysis, this paper focuses on: (1) how to train the initial model to accurately approximate the characteristics of the existing simulation data, and (2) how to plan the data for transfer learning so that the existing deep model is better adapted to the new domain.

The training algorithm of deep neural networks has been widely studied in recent years. From different perspectives, the gradient descent algorithm [46], [47], the multi-task learning algorithm [48], [49], and the hyper-parameter optimization algorithm [50], [51], [52] are often involved, but there is a lack of a general framework to apply them to engineering practice. Compared with training algorithms, little attention is paid to data planning for transfer learning, and the commonly used random sampling has been considered insufficient to effectively promote the adaptability of the deep model [41], [43].

In the current context, transfer learning is a kind of small-sample learning. In this case, it is difficult to ensure the efficient filling of design space by design points even using a uniform sampling algorithm [53], [54]. In addition, due to the existence of noise, the perturbation of measurement data will lead to the uncertainty of model transfer, making its prediction accuracy may be lower than the expected value [55], [56]. Although the alphabet design [57], [58], [59] can reduce the impact of data perturbation by exchanging different combinations of design points, ensuring that the prediction accuracy of the deep model is close to the theoretical mean. However, the design points obtained by this type of method are mainly concentrated in the large curvature region or edge region of the model, resulting in a large blank in the filling of the design space [59], [60]. Since the insufficient learning of the blank area, there may be a large error between the transferred model and the reality.

To solve the above problems, an uncertainty-analysis-based closed-loop transfer learning method is proposed in this paper. The main contributions are as follows:

(1) Based on the simulation model of a telescopic boom forklift, the influencing factors of its target performance are analyzed, and a multi-layer perceptron (MLP) deep model is designed.

(2) A general training framework of the deep model is suggested, and an algorithm application example is given. The framework can be applied to single-output and multi-output problems, and it is easy to combine existing

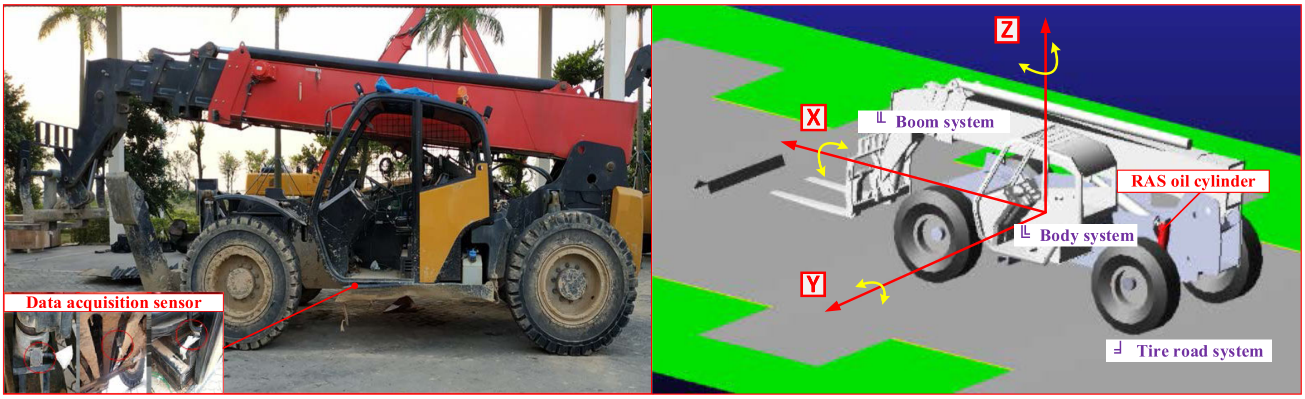


FIGURE 1. The forklift and its simulation model.

training algorithms to improve modeling efficiency and model accuracy.

(3) A new uncertainty-analysis-based sampling algorithm is proposed, based on which the data for transfer learning is developed in a closed-loop iterative mode.

(4) The effectiveness of the modeling method is verified by an engineering experiment of a telescopic boom forklift.

The rest of this paper is organized as follows. Section II briefly introduces the simulation model and the deep model of the forklift. Section III gives the general framework for deep model training. Section IV introduces the uncertainty-analysis-based closed-loop transfer method in detail. Section V presents experimental results and discussions. Concluding remarks are shared in Section VI.

II. THE MODEL OF TELESCOPIC BOOM FORKLIFT

Fig. 1 shows a forklift and its simulation model, which the simulation model consists of three parts: body system, boom system, and tire road system. The body system is mainly composed of the frame and the cab. To improve driver comfort, the frame and the cab are connected by rubber shock absorbers, but the cab is allowed to jump in a small range in the Z direction, and bolts are used to restrict the movement of the cab in other directions. Two oil cylinders are arranged at the front and rear of the frame. The front cylinder is a leveling oil cylinder, which is used to level the entire forklift before work. The rear cylinder is a rear axle stabilization system (RAS) oil cylinder, which is used to absorb shocks and improve driving comfort. The boom system is connected to the frame through a rotating pair and is driven by a luffing cylinder to realize lifting. The boom system includes four-section telescopic arms, of which the first and second sections are driven by a telescopic cylinder, and the third and fourth sections are driven by a chain and pulley. The tire road system is a rigid-flexible coupling model, including a tire model and a road model with trapezoidal obstacles, which aims to realistically restore the working state of the forklift and simulate driving conditions.

The construction of the simulation model of the forklift is the preliminary work of this paper. In the simulation model, the structural design parts are completed in Pro/E software and imported into ADAMS software for configuration and kinetic analysis. The circuit and hydraulic control parts are designed in AMESim software. To better reproduce the functions of actual components, the spool and flow area of the balance valve are approximated by the hydraulic design and hydraulic component design modules, and the variable amplitude multi-way valve is also designed using this method. It should be noted that when configuring the structural model in ADAMS, the type of material needs to be changed simultaneously if a part with large deformation is transformed into a flexible body, in order to avoid the loss of elastic modulus and Poisson's ratio. After the simulation model is constructed, the response value is obtained through simulation and compared with the measurement value. Then, according to the error between the response value and the measurement value, the parameters of the three systems are further adjusted to obtain a sufficiently accurate simulation model.

Based on the simulation model, various performance responses of the forklift can be calculated. Fig. 2 shows an example of the maximum roll angle of the frame and the Z-axis amplitude of the cab base from the simulation model under driving conditions. These two outputs usually represent the safety and comfort of forklifts and are also the focus of this paper. In Fig. 2, the sharp wave position indicates that the forklift passes the obstacles at about 2.5 seconds and 6.8 seconds. The first obstacle is for the right tire to pass, and the second obstacle is for the left tire to pass. Since the cab is installed on the left side of the frame, the center-of-mass of the forklift is not on the center of the frame, causing the frame to tilt slightly to the left when reaches the stable state. As can be seen from the picture below, the base of the cab is about 376 mm above the ground (excluding the dimensions of the tires).

To construct the surrogate of the simulation model, we analyzed the design parameters related to the above two outputs. In engineering practice, design parameters related to the two outputs are coupled with each other. For example, the amplitude of the cab base is related to the stiffness and damping of the rubber shock absorber, while the roll angle is related to the mass and center of mass of the forklift.

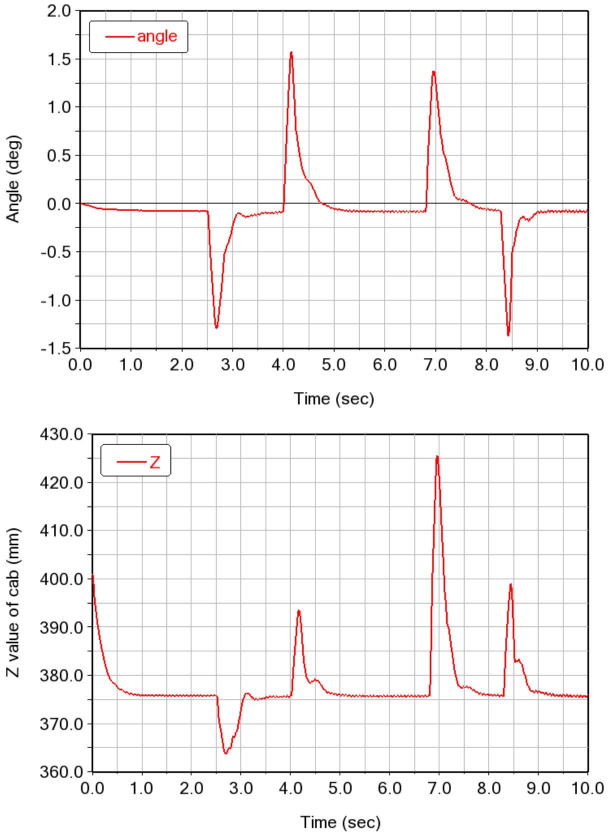


FIGURE 2. Roll angle and Z-value of the forklift when it passes obstacles.

Meanwhile, these two outputs are related to the parameters of the hydraulic control system of the RAS cylinder and the parameters of the tires. Accordingly, the mathematical expression of the objective of surrogate modeling is written as

$$f(x) = [f_1(x), f_2(x)]^T, \tag{1}$$

where $f_1(x)$ and $f_2(x)$ are the outputs, which respectively represent the roll angle of the frame and the Z-axis amplitude of the cab base; x is the input, that is, the design parameters related to the outputs.

In the engineering practice of forklift design, some design parameters have been fixed or are difficult to optimize (or the optimization cost is very high). In general, these parameters will be treated as constants, resulting in the following design variables for Eq. (1): (a) the stiffness and the damping of the rubber shock absorber, (b) the mass and center-of-mass of the forklift, and (c) the current of the hydraulic control system. The above variables will serve as x, where $x = [x_1, x_2, x_3, x_4, x_5]^T$, and its definition space Θ is listed in Table 1, $x \in \Theta$.

Eq. (1) is a black box function, whose mapping relationship between inputs and outputs is contained in a large number of simulation data and can be approximated by a surrogate model. Based on the powerful learning ability of the deep neural network to data features, an MLP deep model is designed for Eq. (1), as shown in Fig. 3. The framework of MLP has three types of layers: the input layer, k hidden layers, and the output layer. The 5 design parameters to be input constitute the input layer, and the 2 fitted results are exported by the output layer. The middle hidden layers establish the nonlinear model of the forklift under driving conditions, in which k is usually set to $k \geq 2$. The different layers are connected by weights and biases.

TABLE 1. Design variable and its definition space.

Design variable	Variable name (unit)	Lower limits	Upper limits
x_1	Stiffness of the rubber shock absorber (N/mm)	1269	3269
x_2	Damping of the rubber shock absorber (N.s/mm)	9	9.3
x_3	Center of mass of the forklift (mm)	850	950
x_4	Mass of the forklift (kg)	11478	13478
x_5	Control current of the hydraulic system (mA)	100	750

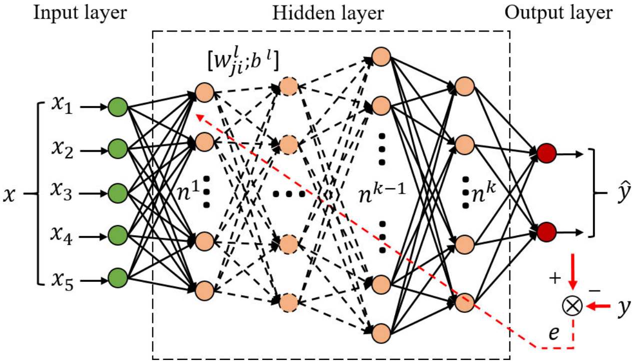


FIGURE 3. The framework of the MLP deep model.

In Fig. 3, n^l, w_{ij}^l and b^l represent the number of neurons, the connection weights, and biases of neurons, in the lth hidden layer, respectively, $l = 1, \cdots, k$; e represents the error between the actual output $\hat{y} = f(x)$ and the desired output y; the dotted line represents the error back-propagation.

To improve the training efficiency, the input and output data are uniformly normalized using the following equation as

$$x' = \frac{2 \times (x - x_{\min})}{x_{\max} - x_{\min}} - 1. \tag{2}$$

where x, x' represent the data before and after normalization, respectively, and $x_{\max}$ and $x_{\min}$ represent the maximum and minimum values of the data.

The neurons of the input layer of MLP are mainly used to cache data. The neurons in the hidden layer and output layer perform data transformation through activation functions. In this work, ReLU is chosen as the activation function, and its mathematical description is

$$f\left(O_j^l\right) = \max\left(0, O_j^l\right), \tag{3}$$

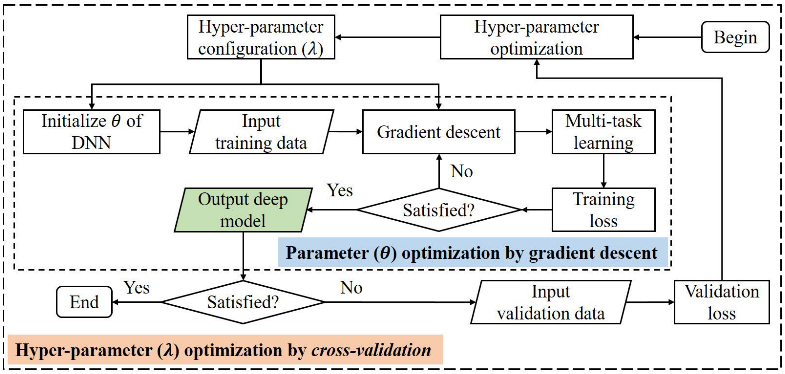

Hyper-parameter configuration (λ)
Hyper-parameter optimization
Begin
Initialize θ of DNN
Input training data
Gradient descent
Multi-task learning
No
Yes
Output deep model
Satisfied?
Training loss
Parameter (θ) optimization by gradient descent
End
Yes
Satisfied?
No
Input validation data
Validation loss
Hyper-parameter (λ) optimization by *cross-validation*

FIGURE 4. The general framework for deep model training.

where O_j^l represents the output of the jth neuron of the lth hidden layer, and

$$O_j^l = \sum_{i=1}^{n^{l-1}} w_{ji}^l O_i^{l-1} + b^l, \tag{4}$$

O_i^{l-1} is the ith output of the $(l-1)$th layer, O_j^l is the jth output of the lth layer, w_{ji}^l is the weight used to connect the jth output of the lth layer and the ith output of the $(l-1)$th layer, and b^l is the bias of the lth layer.

III. GENERAL FRAMEWORK FOR DEEP MODEL TRAINING

Let $\mathcal{A}^{\theta,\lambda}(\cdot)$ be the deep model of the given hyper-parameter configuration λ, $\theta = \{w; b\}$ are the unknown parameters to be estimated, then the training of the model can be regarded as minimizing a loss function, which is defined as

$$\mathcal{L}^{\theta,\lambda} = \frac{1}{N_{out}} \sum_{i=1}^{N_{out}} \gamma_i \left(y_i - \mathcal{A}_i^{\theta,\lambda}(x)\right)^2, \tag{5}$$

where N_{out} is the number of outputs, γ_i is the weight coefficient, $0 \le \gamma_i \le 1$, $\sum_{i=1}^{N_{out}} \gamma_i = 1$, y_i and $\mathcal{A}_i^{\theta,\lambda}(x)$ are the expected and predicted values of the deep model, respectively.

As mentioned earlier, Eq. (5) can be solved by the back-propagation method [18]. Specifically, θ are first randomly initialized, and then optimized by gradient descent algorithm to asymptotically approximate the expected θ^* over $\theta \in \Theta$, which is mathematically described as

$$\mathcal{A}^{\theta^*,\lambda}(x) = \mathrm{Arg}\min_{\theta\in\Theta} \operatorname*{mean}_{x\in X^{\text{train}}} \mathcal{L}_t^{\theta,\lambda}, \tag{6}$$

where Θ is the neural networks parameter space, X^{train} is the training set, $\mathcal{L}_t^{\theta,\lambda}$ is the training error calculated by Eq. (5). Obviously, for any given λ, we can get a deep model $\mathcal{A}^{\theta^*,\lambda}(x)$, and the best model $\mathcal{A}^{\theta^*,\lambda^*}(x)$ depends on an optimal hyper-parameter configuration λ^* defined by

$$\lambda^* = \mathrm{Arg}\min_{\lambda\in\Lambda} \operatorname*{mean}_{x\in X^{\text{valid}}} \mathcal{L}_v^{\theta^*,\lambda}, \tag{7}$$

where Λ is the hyper-parameter space, X^{valid} is the validation set, and $\mathcal{L}_v^{\theta,\lambda}$ is the validation error calculated by Eq. (5). In this work, Λ usually includes the framework of the neural network, parameters of the gradient descent algorithm, weight coefficients of outputs, etc.

Note that we cannot evaluate λ over the unknown function $\mathcal{A}^{\theta^*,\lambda}(x)$, since θ^* still needs to be optimized. Fortunately, the technique of *cross-validation* can be used to estimate it. Therefore, Eq. (7) is rewritten as

$$\lambda^* \approx \mathrm{Arg}\min_{\lambda\in\{\lambda_1\cdots\lambda_S\}} \Psi(\lambda) = \hat{\lambda}, \tag{8}$$

where $\Psi(\lambda) = \operatorname*{mean}_{x\in X^{\text{valid}}} \mathcal{L}_v^{\theta^*,\lambda}$, S is the number of trials $\{\lambda_1 \cdots \lambda_S\}$, and $\hat{\lambda}$ is the best trial from the set of $\{\lambda_1 \cdots \lambda_S\}$ with limited resources.

The problem of Eq. (8), also known as Hyper-parameter optimization of deep learning algorithm [50], [51], [61], has been well studied in the machine learning community, but may not be well known in the engineering optimization community. To facilitate engineering application, a general framework for deep model training is given in this paper, as shown in Fig. 4.

The framework reveals a two-level nested iterative optimization process, which involves: (1) an inner optimization that minimizes the training error $\mathcal{L}_t^{\theta,\lambda}$ over $\theta \in \Theta$; and (2) an outer optimization that minimizes the validation error $\mathcal{L}_v^{\theta,\lambda}$ over $\lambda \in \Lambda$, where Λ is usually determined by the type of gradient descent algorithm and DNN. Take the MLP model as an example. Before the training starts, the following hyper-parameters need to be given. (1) Parameters of gradient descent algorithm Adam [46]: the initial learning rate δ, the learning rate step size s, the decay factor τ, and the attenuation coefficient β_j, $j = 1, 2$; (2) Structural parameters of neural network MLP: the number of hidden layer k and the number of neurons in the lth hidden layer n^l; (3) Batch size of the training data: $\mathcal{B}$. The combination of these hyper-parameters is the so-called hyper-parameter configuration $\lambda = \left\{k, n^l, \mathcal{B}, \delta, s, \tau, \beta_1, \beta_2\right\}$, and their value ranges constitute the hyper-parameter space Λ mentioned above.

It is worth noting that in hyper-parameter optimization studies, the weight coefficient γ is usually not taken into account. Although evenly distributing the weight coefficients can avoid tedious debugging, it may lead to unbalanced training for some scenarios. To solve this problem, the Gaussian likelihood estimation method [48], namely Multi-task learning, is introduced into the general training framework. Taking a model with two outputs as an example, the likelihood estimate can be constructed as follows:

$$\begin{aligned} p\left(y_1, y_2|\mathcal{A}^{\theta,\lambda}(x)\right) &= p\left(y_1|\mathcal{A}^{\theta,\lambda}(x)\right) \cdot p\left(y_2|\mathcal{A}^{\theta,\lambda}(x)\right) \\ &= \mathcal{N}\left(y_1; \mathcal{A}^{\theta,\lambda}(x), \sigma_1^2\right) \\ &\quad \cdot \mathcal{N}\left(y_2; \mathcal{A}^{\theta,\lambda}(x), \sigma_2^2\right), \end{aligned} \tag{9}$$

where σ is the uncertainty parameter of the output, which is inversely proportional to the weight coefficient γ. Accordingly, $\theta = \{\theta;\sigma\}$ can be optimized by the gradient descent algorithm to maximize the likelihood estimation to obtain similar prediction performance for the two outputs. The likelihood estimation construction of the model with more output can be referred to [48].

Assuming that the dimension of λ is a and the partition of each dimension is b, then the number of trials $S = b^a$. When the dimension is high or each dimension needs to be carefully divided, S is a very large number, and some brute-force search algorithms [62] such as Grid search and Random search will be difficult to cope with. In this case, heuristic hyper-parameter optimization algorithms such as Hyperband [52], Bayesian optimization [51], [63], or hybrid methods [50], [64] should be used. Whichever algorithm is chosen, it can be easily incorporated into the framework of Fig. 4, and the termination condition of the nested iteration can be set to such that the evaluation index of the model does not improve after multiple iterations (for example, 10 iterations). Then, with the given iteration resources, the approximately optimal hyper-parameter configuration λ^* is selected from the evaluated trials of $\{\lambda_1 \cdots \lambda_S\}$, and the surrogate model $\mathcal{A}^{\theta^*,\lambda^*}(x)$ is finally obtained.

IV. UNCERTAINTY-ANALYSIS-BASED CLOSED-LOOP TRANSFER

Due to the existence of noise, transfer learning with a small amount of measurement data will lead to uncertainty in the model, thus affecting its prediction accuracy [55], [56]. Unfortunately, many existing studies fail to take this into consideration in the surrogate modeling process. To address this issue, an uncertainty-analysis-based sampling algorithm is introduced in this section to obtain the data for model transfer and further form a closed-loop transfer learning approach.

A. UNCERTAINTY-ANALYSIS-BASED SAMPLING

Data sampling based on uncertainty analysis is also known as active data development (ADD) [59], [65]. Assuming that θ^* is the expected parameter matrix of deep model, its parameter evaluation error $(\theta - \theta^*)$ follows zero mean value, and the variance is $C = \sigma^2 M(\Omega, \theta)^{-1}$. Further, suppose p_i is the probability that the design point x_i placed at different positions in the design space, ADD firstly uses $m \times m$ Fischer information matrix (FIM) to associate a design Ω and p_i as follows:

$$M(\Omega, \theta) = \sum_{i=1}^{N} p_i V(\Omega, \theta)_i V(\Omega, \theta)_i^T = M(p), \tag{10}$$

where $x_i \in \Omega$, N is the size of Ω, and $V(\Omega, \theta)$ is the $N \times m$ Jacobian matrix, in which the jth column is $v_j = \partial \mathcal{A}^{\theta}(x_i) / \partial \theta_j, j = 1, \cdots, m$.

In Eq. (10), the design Ω is a group of support points from the design space [65]. The combination of different points will result in different $M(\Omega, \theta)$. In general, the larger $M(\Omega, \theta)$, the larger the uncertainty of the model, the lower the prediction ability of the model. Since $M(p)$ is a matrix, the mathematical expression of the optimization objective can be defined as

$$\Omega = \underset{x \in \Theta}{max} \left[\text{logdet}(M(p))\right] \tag{11}$$

where logdet $(\cdot)$ represents the log-determinant value of the matrix.

Eq. (11) can be solved by the Federov exchange algorithm [66], [67] or the Multiplication algorithm [68], [69], and the obtained design Ω is a set of design points with $p_i = 1$. As mentioned earlier, these design points are usually concentrated in the large curvature region or edge region of the model [59], [60]. As a result, large portions of the design space are not covered, which may lead to a large deviation between the deep model and the expected model. To solve this problem, p_i should not be directly used as the sampling index, or the value of p_i should not be considered only.

We suggested a new data development method that combined a Monte Carlo sampling [70] with the multiplication iteration, where p_i is relaxed to 0∼1 and converted to a cumulative probability g_c. Further, g_c is taken as the sampling mode, so that x_i with large g_c value has a high probability of being sampled, and vice versa. Since $g_c > 0$, the sampling probability of all design points is greater than zero, which makes design points of Ω statistically cover the whole design space.

The detailed process of the new method is summarized as follows:

(1) Given the size of the design Ω: N;

(2) Given a monotone increasing function $f(d, \delta)$ with $\delta > 0$ for multiplication iteration:

$$f(d, \delta) = \begin{cases} \delta e^d, & d < 0 \\ \delta(d+1), & d \geq 0. \end{cases} \tag{12}$$

Different from the functions commonly used in the existing study as [68]: $f(d, \delta) = e^{\delta d}$, $f(d, \delta) = e^{\delta(d-1)}$, and $f(d, \delta) = e^{d\delta} / (1+e^{d\delta})$, a piecewise function, that is Eq. (12), is used in this paper for multiplication iteration, so as to avoid the situation where the function value converges too slowly or increases too fast when d is a large positive number.

(3) Use the uniform design (UD) algorithm [53] to discretize the design space Θ to N_c candidate design points, where $N_c \gg N$;

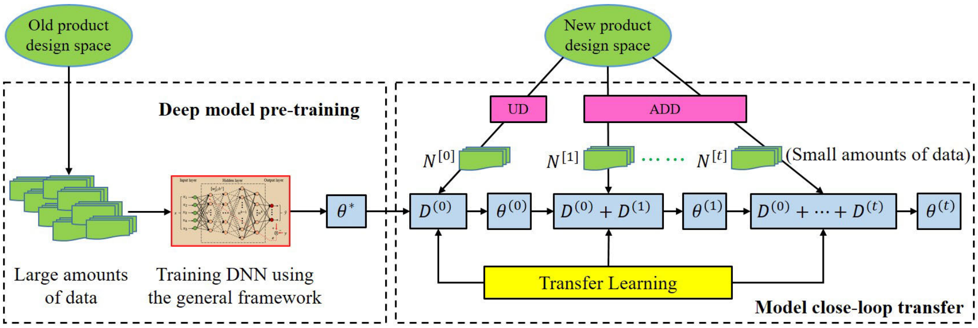


FIGURE 5. The process of the close-loop transfer.

(4) Use the multiplication iteration to calculate the p_i of the N_c candidate design points as

$$p_i^{(r+1)} = \frac{p_i^{(r)} f\left(d_i^{(r)}, \delta\right)}{\sum_{j=1}^{N_c} p_j^{(r)} f\left(d_j^{(r)}, \delta\right)}, \tag{13}$$

where

$$d_i^{(r)} = \frac{\partial \emptyset\,(p)}{\partial p_i^{(r)}} = \frac{1}{m} V_i^T M\left(p^{(r)}\right)^{-1} V_i, \tag{14}$$

and

$$\emptyset\,(p) = \mathrm{logdet}\,(M\,(p))\,, \tag{15}$$

V is the $V\,(\Omega, \theta)$ that ignores the (Ω, θ) for simplification, r is the iteration step and the initial value of p_i is set to $p_i = 1/N_c$. After each iteration, p_i is normalized to between 0 and 1, and $\sum_{i=1}^{N_c} p_i = 1$.

(5) Sort the N_c design points in descending order according to p_i and divide them into K contours as follows:

$$E_c = \{x_i\colon (c-1)\,u < i \le cu\}\,, \quad c = 1, \cdots, K, \tag{16}$$

where $u = N_c/K$ is assumed to be an integer, the first contour E_1 contains u design points with the largest value in p_i, and the last contour E_K contains u design points with the smallest value in p_i.

(6) Calculate the average value of p_i in each contour:

$$a_c = \frac{1}{u} \sum\nolimits_{i=(c-1)u+1}^{cu} p_i, \tag{17}$$

(7) Calculate the cumulative distribution probability of the contours:

$$g_c = \sum_{j=1}^{c} a_c, \tag{18}$$

(8) Finally, sample from candidate design points based on g_c. At first, generate N random numbers s_t, where $s_t \in [0, 1]$, and $t = 1, \cdots, N$. If $s_t \le g_c$, the cth contour is drawn, and then a design point will be randomly selected from the contour. If the contour E_c occurs $n_c > 0$ times in these draws, n_c design points will be generated from E_c.

After obtaining the design Ω, the response value of the design points should be obtained by the real forklift test, so as to obtain the data D for model transfer.

B. MODEL CLOSED-LOOP TRANSFER

Eq. (13) reveals that the calculation of p_i of candidate design points is affected by θ^* of the deep model, thus affecting the design Ω and the data D. Since the θ^* of the deep model cannot precisely reflect the personality characteristics of the new product, it is unreliable if planning the N design points of Ω at once. Therefore, to obtain a good distribution of design points in Ω, it is recommended to update θ^* with a small amount of data developed by UD, and then implement a closed-loop transfer, as shown in Fig. 5. For ease of description, we use $\theta^{(t)}$ represents the parameter matrix of the model obtained after transferring, where t represents the number of closed-loop transfer, $t \ge 1$, and therefore $N = \sum N^{[t]}$. The $\theta^{(0)}$ represents the parameter matrix of the deep model before the closed-loop transfer, and $N^{[0]}$ is the number of data (or design points) developed by UD.

At each iteration of the closed-loop transfer, it maintains:

- Set θ^* to be the *optimal*$(\theta^{(t)})$;
- Generate $N^{[t]}$ design points by the proposed sampling method based on θ^* and collect their observations;
- Merge the previous data as $D_t = \sum D^{(t)}$;
- Transfer the model with D_t to obtain $\theta^{(t)}$.

It is worth noting that the data used to transfer the model is D_t, not $D^{(t)}$. The purpose of merging data is to avoid overfitting. To improve the modeling efficiency, design points can be generated in batches, that is, $N^{[t]} \ge 2$, since the same area on the model should have close gradients in two adjacent samplings. The maximum number T of closed-loop iterations is a user-defined parameter, which can also be set to correlate with the prediction accuracy of the model.

V. EXPERIMENTAL RESULTS AND DISCUSSIONS

A. EXPERIMENT SETTINGS

1) EVALUATION CRITERIA

The following three metrics [8], namely Root Mean Square Error (RMSE), Maximum Absolute Error (MAE), and Correlation coefficient (R) are used to assess the modeling performance, where RMSE is defined by

$$\mathrm{RMSE} = \left\{\frac{1}{N_v} \sum_{i=1}^{N_v} \left(y_i - \hat{y}_i\right)^2\right\}^{1/2}, \tag{19}$$

TABLE 2. Settings of the closed-loop transfer algorithm.

Parameter	Settings
δ for monotone increasing function $f(d,\delta)$	$\delta = 1.3$
No. of contours	$K = 100$
No. of design points planned by UD	$N^{[0]} = 20$
No. of design points sampled for each closed-loop	$N^{[t]} = 5$, $t \geq 1$
No. of candidate design points	$N_c = 1024$
Termination condition of closed-loop iteration	$T = 12$

TABLE 3. Definition space of hyper-parameter.

Hyper-parameter	Scale	Min	Max
Structure hyper-parameters			
No. of hidden layers（k）	linear	2	5
No. of neurons in each hidden layer（n^l）	power	2^3	2^6
Learning algorithm hyper-parameters			
Learning rate（δ）	linear	1e-6	1e-2
Batch size（B）	power	2^4	2^8
Attenuation coefficient（β_1）	linear	0.8	0.99
Attenuation coefficient（β_2）	linear	0.8	0.99

MAE is defined by

$$\mathrm{MAE} = max\left|y_i - \hat{y}_i\right|, \quad i = 1, \cdots, N_v, \tag{20}$$

R is defined by

$$\mathrm{R} = \frac{\left(\sum_{i=1}^{N_v}(y_i - \bar{y})\left(\hat{y}_i - \bar{\hat{y}}\right)\right)}{\left(\left\{\sum_{i=1}^{N_v}(y_i - \bar{y})^2\right\}^{\frac{1}{2}}\left\{\sum_{i=1}^{N_v}\left(\hat{y}_i - \bar{\hat{y}}\right)^2\right\}^{\frac{1}{2}}\right)}, \tag{21}$$

y_i is the expected response and $\hat{y}_i$ is the predicted response at the ith test point, N_v is the size of the validation set, $\bar{y}$ is the mean of the expected response, and $\bar{\hat{y}}$ is the mean of the predicted response.

2) ALGORITHM SETTING

At the beginning of the closed-loop transfer, some algorithm parameters need to be set first, as shown in Table 2.

In the general framework, the combination of Hyperband and multi-task learning is used to pre-train the DNN by default. Random search (RS) [62] is the benchmark for comparison in this study because it is a representative hyper-parameter optimization method. For Hyperband [52], the maximum iteration resources for each hyper-parameter configuration is $r_{max} = 100$ and the downsampling factor is $\eta = 3$. The total number of iteration resources is set to $50 \times r_{max} = 5000$ as a constraint. To demonstrate the possible imbalance of gradient descent-based surrogate modeling, the multi-task learning method is not affiliated with RS for model training, and the weight parameter of the loss function is set to $\gamma_1 = \gamma_2 = 0.5$. The hyper-parameter definition space is shown in Table 3.

The existing product in this paper is a forklift with a rated load of 6klb (6K), while the newly developed product is a forklift with a rated load of 10klb (10K). They belong to the same type of sequence-developed products. In the model pre-training stage, a total of 1725 sets of simulation data are available. For the convenience of the experiment, the simulation data is also used in the model transfer part, that is, the response of the developed design points is obtained from the simulation model instead of the real forklift test. Not surprisingly, this data needs to be added with noise. The validation data consisted of 100 groups of mixed data including real forklift test data and simulation data of the 10K forklift. The program is written based on PyTorch framework and runs on Intel i5-9600K-3.7GHz CPU and GeForce RTX-2080 GPU. The operating system is Windows 10 64-bit. After 10 repetitions, the optimal model is selected as the pre-trained deep model, that is, the initial model for transfer learning.

B. EXPERIMENTAL RESULTS

1) VERIFICATION RESULTS OF THE PRE-TRAINED MODEL

It takes about 2 minutes on average to pre-train the surrogate model under the constraint of total iteration resources. The experimental results are listed in Table 4 and plotted in Fig. 6 (average best results), where α represents the roll angle of the frame, Z represents the Z-axis amplitude of the cab base, RS-NML represents RS with non-multi-task learning, HB-ML represents Hyperband with multi-task learning, *Improvement* represents the comparison results, and the Network framework is the optimal model framework defined by (k, n^l).

Table 4 indicates that as for the R indicator of the deep model, RS-NML gets 0.9066 and 0.9184, while HB-ML gets 0.9324 and 0.9326. These results of the two outputs reveal that there should be a training imbalance in the construction of the deep model, and multi-task learning can improve the accuracy of the model. In other words, after using multi-task learning, the accuracy difference between the two outputs is reduced from 1.27% to 0.02% compared with the same weight parameter. The R indicator also shows that the characteristics of the two forklifts have a great similarity since the training data of the deep model comes from the 6K forklift and the validation data comes from the 10K forklift. This discovery further inspires us that it is feasible to reuse existing data when constructing deep models of complex electromechanical products of the same type. The experimental results also show that, compared with RS-NML, an improvement of at least a 17.655% reduction in RMSE and a 24.475% reduction on average in MAE, as well as a 2.145% increment on average in R, are achieved after using HB-ML.

The main reason for the different modeling results may be that Hyperband uses adaptive resource allocation and early stopping strategies [52]. Compared with RS, it can evaluate more hyper-parameter configurations under the same resources, thus finding the best configuration faster. As shown in the right-most column of Table 4, among the two optimal hyper-parameter configurations, they are consistent only in the batch size of training data. From the multiple of r_{max} marked on the horizontal axis of Fig. 6,

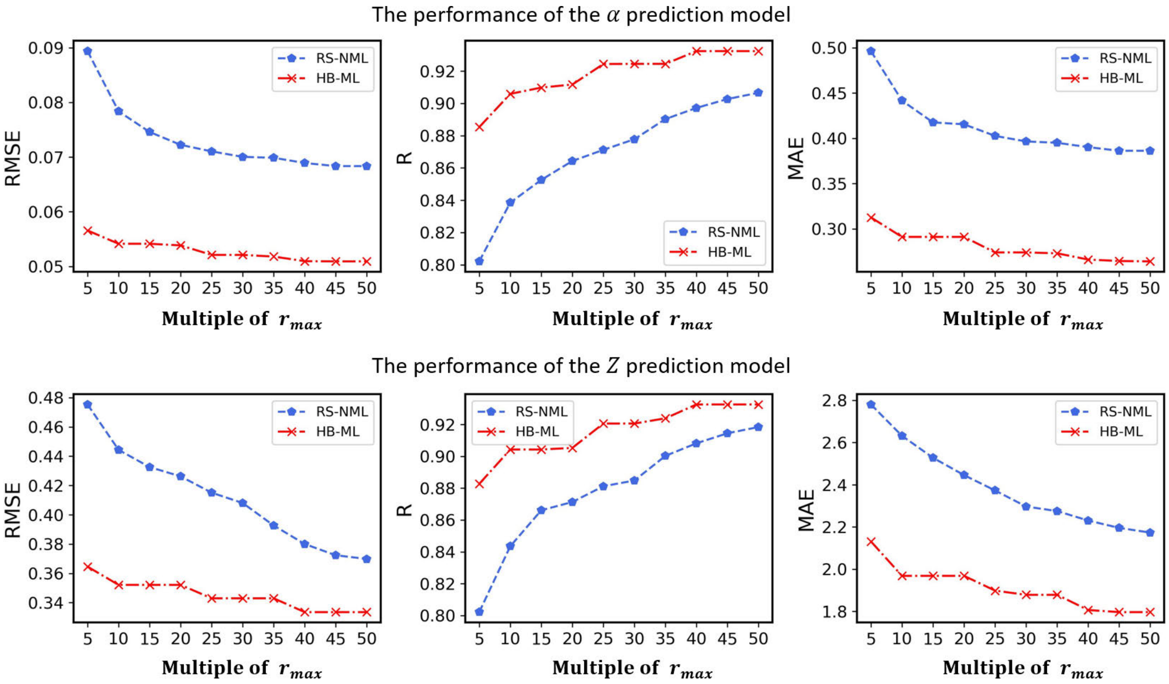


FIGURE 6. The results of pre-training models using different algorithms.

TABLE 4. Verification results of the pre-training model.

Method	MAE		RMSE		R		Network framework and $(\mathcal{B}, \delta, \beta_1, \beta_2)$
	α (deg)	Z（mm）	α (deg)	Z（mm）	α (deg)	Z（mm）	
RS-NML	0.3863	2.1732	0.06835	0.3698	0.9066	0.9184	5-2^3-2^5-2^3-2^3-2 (5,0.0078,0.8,0.8)
HB-ML	0.2642	1.7963	0.05093	0.3335	0.9324	0.9326	5-2^5-2^3-2^3-2^5-2 (5,0.0086,0.9,0.9)
Improvement	-31.61%	-17.34%	-25.49%	-9.82%	+2.85%	+1.55%	--

TABLE 5. Verification results of modeling on 80 groups of data.

Method	MAE		RMSE		R		Network framework and $(\mathcal{B}, \delta, \beta_1, \beta_2)$
	α (deg)	Z（mm）	α (deg)	Z（mm）	α (deg)	Z（mm）	
RT	0.4057	2.2515	0.0682	0.3896	0.8973	0.8924	5-2^5-2^3-2 (5,0.0064,0.9,0.8)
FT	0.2365	1.3222	0.0492	0.2959	0.9955	0.9964	5-2^3-2^5-2^3-2^3-2 (5,0.0078,0.8,0.8)
	0.2025	1.2692	0.0408	0.2567	0.9971	0.9971	5-2^5-2^3-2^3-2^5-2 (5,0.0086,0.9,0.9)
CLFT	0.2104	--	0.0433	--	0.9966	--	5-2^5-2^3-2^5-1 (5,0.0071,0.9,0.9)
	--	1.2875	--	0.2723	--	0.9968	5-2^5-2^3-2^3-2^3-1 (5,0.0068,0.9,0.9)

it also can be seen that after consuming $25r_{max}$, the modeling performance of HB-ML is better than those of RS-NML exhausted $50r_{max}$.

2) VERIFICATION RESULTS OF MODEL TRANSFER

In this work, the classical fine-tuning (FT) method is used as a benchmark to illustrate the effectiveness of the proposed uncertainty-analysis-based closed-loop transfer method (i.e., closed-loop fine-tuning, CLFT). As for FT, the 80 design points are planned by LHD [54], and on this basis, training data is further collected from the 10K forklift. Moreover, these training data are also used to retrain a deep model from scratch and serve as another benchmark for comparison, namely Retraining (RT). The results of the three comparison modeling methods are listed in Table 5 and plotted in Fig. 7. As mentioned above, the real forklift test is replaced by model simulation for convenience.

Fig. 7 reveals that CLFT achieved the best performance after modeling with 80 groups of data. Compared with the RT, both fine-tuning methods have a higher performance starting point and faster convergence due to the use of the pre-training model. By comparing Fig. 6, it can be found that the performance of the pre-training model is even better than that of the mode training from scratch with 80 groups of 10K forklift data. The main reason may be that the response performance of the two forklifts is very similar. In addition, the curves of the three indicators remind us that the CLFT has a faster convergence rate than the FT. For example, after using 55 groups of data, the model performance obtained by CLFT is similar to those obtained by the FT using 80 groups of data. According to the trend of the three curves, we can also predict that with the increase of training data, the modeling performance of the three methods will gradually approach.

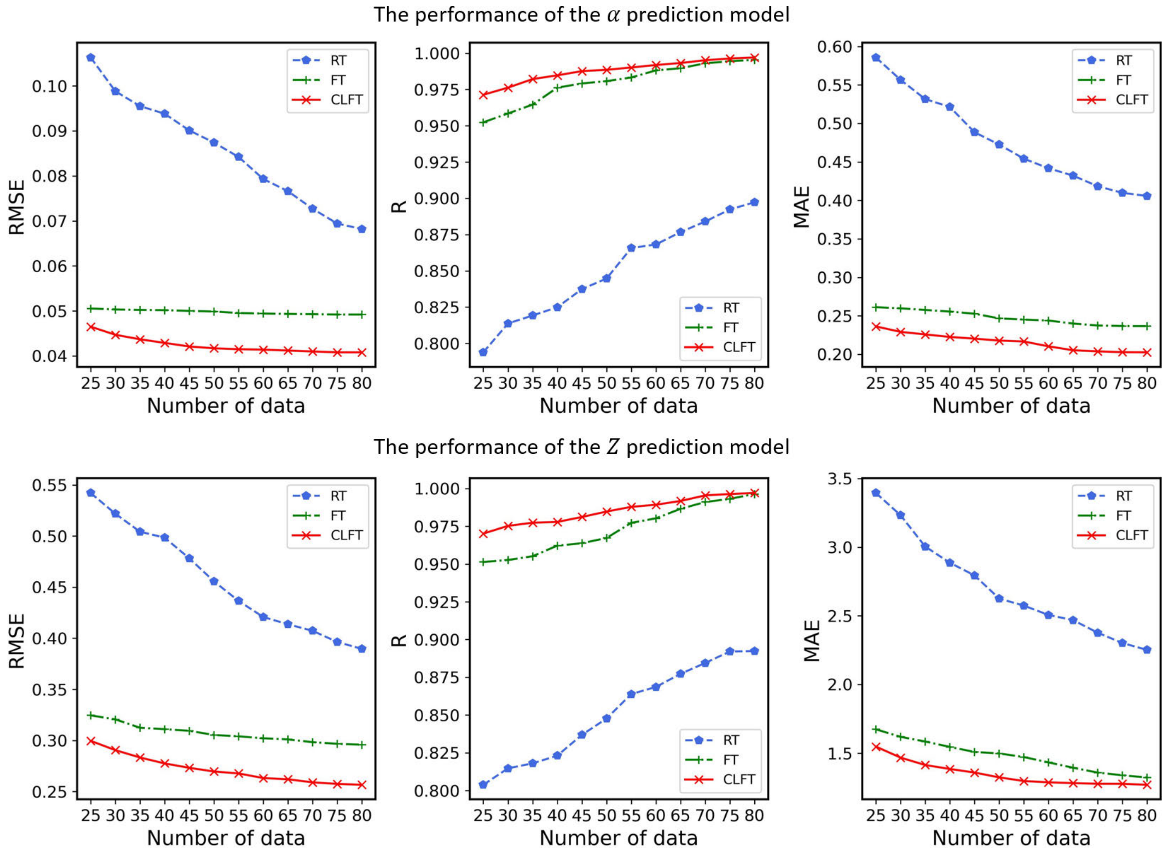


FIGURE 7. The results of modeling 80 groups of data using different algorithms.

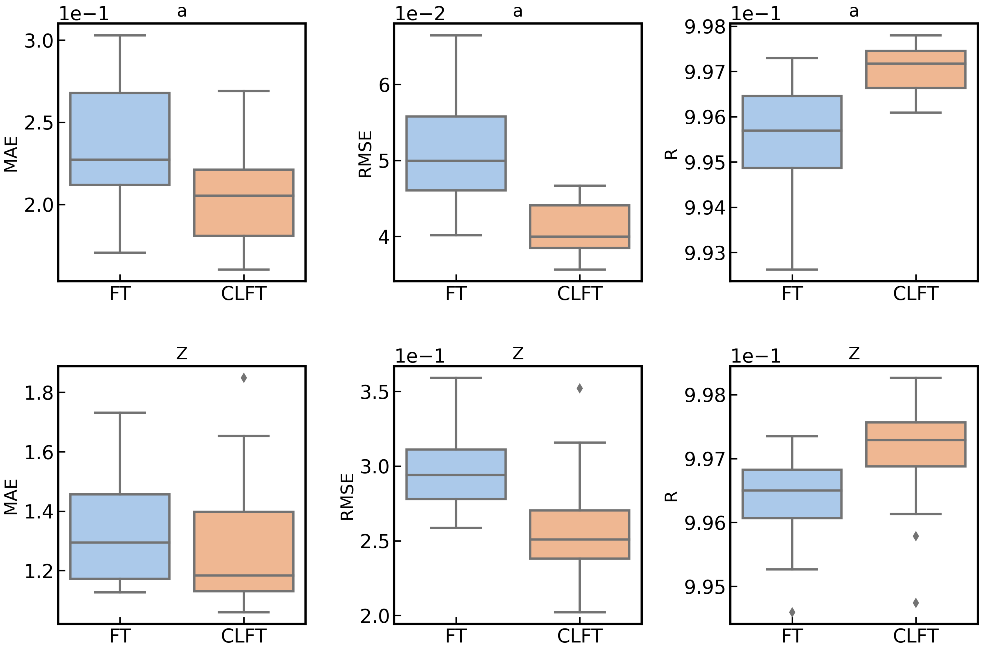


FIGURE 8. The box plots of modeling results of different methods.

As can be seen from Table 5, the R indicator obtained by the two fine-tuning methods is at least 0.9955, while the best R indicator obtained by the RT method is 0.8973. Moreover, compared with the RT method, the two fine-tuning methods also achieved about 40% improvement in MAE and RMSE. When only comparing the two fine-tuning methods, as for

RMSE and MAE, after using CLFT, average improvements of 16.78% and 9.19% are achieved, respectively. From the R indicators obtained by CLFT, the two outputs both reach 0.9971, which proves that it is feasible in surrogate modeling of the telescopic boom forklift. It should be noted that the training data in RT and FT are obtained through a single sampling. The sampling time is less than 1 second, and the training time is about 2s and 1s, respectively. In CLFT, design point planning is performed 13 times, and each planning takes about 4 seconds. As a result, the modeling time reached about 1 minute. To improve sampling efficiency, CLFT can be accelerated with the Pronzato method [65] in the part of multiplication iteration.

The results of modeling the two outputs separately using CLFT are recorded in the bottom two rows of Table 5. It can be seen that the modeling results are slightly worse than those based on multi-task learning. The main reason may be that associated tasks enhance the performance of each other by sharing information and complementing each other. In addition, training a multi-output model is clearly more efficient than training multiple models separately.

To evaluate the robustness of the modeling results using CLFT and FT, 30 independent experiments are repeated. The box plot of the results is plotted in Fig. 8, in which, the three MAE/RMSE/R indicators of the roll angle of the frame α are drawn in the upper row, while the three MAE/RMSE/R indicators of the Z-axis amplitude of the cab base Z are drawn in the lower row. The results in Fig. 8 reveal that CLFT is more robust than FT since the distance between the upper quartile and the lower quartile of those boxes is smaller.

C. DISCUSSION

The $M(\Omega, \theta)$ describes the stable distribution range of the surrogate model $\mathcal{A}^{\theta}(x)$ on its mean value, which is related to the structure of the parameter θ, the number of experimental measurement data D, and the distribution of the design Ω. Generally, the more measurement data we get, the stronger the regularity of its distribution will be, but the acquisition of large amounts of data is very time-consuming. In this paper, a good distribution of design Ω is obtained by closed-loop sampling based on uncertainty analysis, and the structure of the parameter θ is searched by Hyperband. Consequently, after only 80 groups of data are used for fine-tuning, the R indicators of the surrogate model both reach 0.9971. However, the results in Fig. 8 also show that it is a challenge to improve the prediction accuracy and robustness of the surrogate model with multiple outputs at the same time, for example, the box plots of Z have a few singularities, and the box plots of MAE are still large.

Theoretically, the deep model can only represent the reality as well as the simulation model at most, even if the cost of simulation data acquisition for the 10K forklift is not considered. In this case, the transfer learning of the model using the developed real forklift test data is also beneficial, because it helps to correct the expression deviation of the deep model to the real system.

VI. CONCLUSION

This paper mainly studies the application of transfer learning to surrogate modeling of complex electromechanical products, specifically, constructing a deep model for a telescopic boom forklift, and focuses on the following two points: (1) how to pre-train the initial deep model to accurately approximate the mapping features of the existing simulation data, and (2) how to plan the data for transfer learning to improve the adaptability of the deep model. To solve the above problems, a general framework for deep model training is given, a new sampling algorithm based on uncertainty analysis is proposed, and a closed-loop transfer learning method is further developed. The effectiveness of the proposed method has been verified by a comparative experimental study. The results show that the R indicators of the surrogate model with two outputs, namely the roll angle of the frame and the Z-axis amplitude of the cab base, can both reach 0.9971 by only using 80 simulation data of the currently designed forklift, which meets the requirements of engineering application. The results of a variety of evaluation indicators also show the advantages of the proposed method compared with the benchmark. However, the method still needs to be tested in more cases and compared with the state-of-the-art approaches (especially different types of parameter-based transfer learning approaches), which is one of the future works. In addition, multi-objective performance optimization of the 10K forklift based on the existing deep model will also be implemented in the future.

ACKNOWLEDGMENT

Jingliang Lin would like to thank Sun Weiqi for her support and help in the process of scientific research.

REFERENCES

[1] J. H. Lee and K. S. Kim, "Optimal design of boom for telescopic boom type forklift truck," *J. Korean Soc. Precis. Eng.*, vol. 37, no. 6, pp. 457–464, Jun. 2020.

[2] H. K. Celik, A. E. W. Rennie, and I. Akinci, "Design and structural optimisation of a tractor mounted telescopic boom crane," *J. Brazilian Soc. Mech. Sci. Eng.*, vol. 39, no. 3, pp. 909–924, Mar. 2017.

[3] L. X. Meng, Z. J. Gui, K. Zhang, J. R. Liu, and S. M. Liu, "Analytical method for the out-of-plane buckling of the telescopic boom with guy cables," *Sci. Prog.*, vol. 104, pp. 1–19, Jan./Mar. 2021.

[4] A. Ji, C. Chen, L. Peng, P. Lv, and X. He, "Collaborative optimization of NURBS curve cross-section in a telescopic boom," *J. Mech. Sci. Technol.*, vol. 31, no. 8, pp. 3861–3873, Aug. 2017.

[5] Y. Zhang, G. Xia, X. W. Tang, L. F. Zhao, and B. Q. Sun, "Variable universe fuzzy control of the lateral stability of forklift trucks based on roll grading," *Proc. Inst. Mech. Eng. D, J. Automobile Eng.*, vol. 236, pp. 1–15, Jul. 2021.

[6] E. Zhang, Q. Zhang, J. Xiao, L. Hou, and T. Guo, "Acoustic comfort evaluation modeling and improvement test of a forklift based on rank score comparison and multiple linear regression," *Appl. Acoust.*, vol. 135, pp. 29–36, Jun. 2018.

[7] R. Verschoore, J. G. Pieters, and I. V. Pollet, "Measurements and simulation on the comfort of forklifts," *J. Sound Vibrat.*, vol. 266, no. 3, pp. 585–599, Sep. 2003.

[8] P. Li, H. Li, Y. Huang, K. Wang, and N. Xia, "Quasi-sparse response surface constructing accurately and robustly for efficient simulation based optimization," *Adv. Eng. Softw.*, vol. 114, pp. 325–336, Dec. 2017.

[9] P. Li, H. Li, Y. Huang, S. Yang, H. Yang, and Y. Liu, "A high sparse response surface method based on combined bases for complex products optimization," *Adv. Eng. Softw.*, vol. 129, pp. 1–12, Mar. 2019.

[10] S. Haoyuan, M. Yizhong, L. Chenglong, Z. Jian, and L. Lijun, "Hierarchical Bayesian support vector regression with model parameter calibration for reliability modeling and prediction," *Rel. Eng. Syst. Saf.*, vol. 229, Jan. 2023, Art. no. 108842.

[11] K. Cheng and Z. Lu, "Active learning Bayesian support vector regression model for global approximation," *Inf. Sci.*, vol. 544, pp. 549–563, Jan. 2021.

[12] Y. Z. Ma, J. J. Royer, H. Wang, Y. Wang, and T. Zhang, "Factorial Kriging for multiscale modelling," *J. Southern Afr. Inst. Mining Metall.*, vol. 114, pp. 651–657, Aug. 2014.

[13] D. Meng, S. Yang, A. M. P. D. Jesus, and S.-P. Zhu, "A novel Kriging-model-assisted reliability-based multidisciplinary design optimization strategy and its application in the offshore wind turbine tower," *Renew. Energy*, vol. 203, pp. 407–420, Feb. 2023.

[14] H. Chen, H. Qiao, L. Xu, Q. Feng, and K. Cai, "A fuzzy optimization strategy for the implementation of RBF LSSVR model in Vis–NIR analysis of pomelo maturity," *IEEE Trans. Ind. Informat.*, vol. 15, no. 11, pp. 5971–5979, Nov. 2019.

[15] N. Namura, K. Shimoyama, S. Jeong, and S. Obayashi, "Kriging/RBF-hybrid response surface methodology for highly nonlinear functions," *J. Comput. Sci. Technol.*, vol. 6, no. 3, pp. 81–96, 2012.

[16] P. Li, Y. Huang, H. Li, K. Wang, N. Xia, and H. Yang, "Efficient modelling and optimization for double wishbone suspensions based on a non-adaptive sampling sparse response surface," *Eng. Optim.*, vol. 51, no. 2, pp. 286–300, Feb. 2019.

[17] X. Ma, P. K. Wong, and J. Zhao, "Practical multi-objective control for automotive semi-active suspension system with nonlinear hydraulic adjustable damper," *Mech. Syst. Signal Process.*, vol. 117, pp. 667–688, Feb. 2019.

[18] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2015.

[19] P. Li, H. Y. Li, Y. B. Huang, S. Q. Yang, H. T. Yang, and Y. S. Liu, "A state-of-the-art survey on deep learning theory and architectures," *Electron.*, vol. 8, pp. 1–66, Mar. 2019.

[20] W. Mei and D. Weihong, "Deep visual domain adaptation: A survey," *Neurocomputing*, vol. 312, pp. 135–153, Jul. 2018.

[21] K. D. Humbird, J. L. Peterson, B. K. Spears, and R. G. McClarren, "Transfer learning to model inertial confinement fusion experiments," *IEEE Trans. Plasma Sci.*, vol. 48, no. 1, pp. 61–70, Jan. 2020.

[22] X. Zhao, Z. Gong, J. Zhang, W. Yao, and X. Chen, "A surrogate model with data augmentation and deep transfer learning for temperature field prediction of heat source layout," *Structural Multidisciplinary Optim.*, vol. 64, no. 4, pp. 2287–2306, Oct. 2021.

[23] M. Cheng, C. Dang, D. M. Frangopol, M. Beer, and X.-X. Yuan, "Transfer prior knowledge from surrogate modelling: A meta-learning approach," *Comput. Struct.*, vol. 260, Feb. 2022, Art. no. 106719.

[24] S. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, pp. 1345–1359, 2010.

[25] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu, "A survey on deep transfer learning," presented at the Int. Conf. Artif. Neural Netw., Rhodes, Greece, 2018.

[26] M. Long, Y. Cao, Z. Cao, J. Wang, and M. I. Jordan, "Transferable representation learning with deep adaptation networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 12, pp. 3071–3085, Dec. 2019.

[27] Q. Sun, C. Zhao, Y. Tang, and F. Qian, "A survey on unsupervised domain adaptation in computer vision tasks," *Scientia Sinica Technologica*, vol. 52, no. 1, pp. 26–54, Jan. 2022.

[28] X. Han et al., "Pre-trained models: Past, present and future," *AI Open*, vol. 2, pp. 1–44, Jan. 2021.

[29] Y. Gu, X. Han, Z. Liu, and M. Huang, "PPT: Pre-trained prompt tuning for few-shot learning," 2021, *arXiv:2109.04332*.

[30] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks," in *Proc. Adv. Neural Inf. Process. Syst. Conf.*, Montreal, QC, Canada, 2014, pp. 1–14.

[31] K. Tian, Z. Li, J. Zhang, L. Huang, and B. Wang, "Transfer learning based variable-fidelity surrogate model for shell buckling prediction," *Compos. Struct.*, vol. 273, Oct. 2021, Art. no. 114285.

[32] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, "A comprehensive survey on transfer learning," *Proc. IEEE*, vol. 109, no. 1, pp. 43–76, Jan. 2021.

[33] M. Everingham, S. M. A. Eslami, L. J. V. Gool, C. K. I. Williams, J. M. Winn, and A. Zisserman, "The Pascal visual object classes challenge: A retrospective," *Int. J. Comput. Vis.*, vol. 111, no. 1, pp. 98–136, 2015.

[34] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, and A. C. Berg, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.

[35] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 2, pp. 1097–1105, 2012.

[36] Y. Gao and K. M. Mosalam, "Deep transfer learning for image-based structural damage recognition," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 33, no. 9, pp. 748–768, Sep. 2018.

[37] C. Feng, H. Zhang, S. Wang, Y. Li, H. Wang, and F. Yan, "Structural damage detection using deep convolutional neural network and transfer learning," *KSCE J. Civil Eng.*, vol. 23, no. 10, pp. 4493–4502, Oct. 2019.

[38] A. T. W. Min, R. Sagarna, A. Gupta, Y.-S. Ong, and C. K. Goh, "Knowledge transfer through machine learning in aircraft design," *IEEE Comput. Intell. Mag.*, vol. 12, no. 4, pp. 48–60, Nov. 2017.

[39] Z. Yang, X. Li, L. C. Brinson, A. N. Choudhary, W. Chen, and A. Agrawal, "Microstructural materials design via deep adversarial learning methodology," *J. Mech. Des.*, vol. 140, no. 11, pp. 1–34, Nov. 2018.

[40] M. Kaya and S. Hajimirza, "Using a novel transfer learning method for designing thin film solar cells with enhanced quantum efficiencies," *Sci. Rep.*, vol. 9, no. 1, p. 5034, Mar. 2019.

[41] Y. Ma, J. Wang, Y. Xiao, L. Zhou, and H. Kang, "Transfer learning-based surrogate-assisted design optimisation of a five-phase magnet-shaping PMSM," *IET Electric Power Appl.*, vol. 15, no. 10, pp. 1281–1299, Oct. 2021.

[42] E. Whalen and C. Mueller, "Toward reusable surrogate models: Graph-based transfer learning on trusses," *J. Mech. Des.*, vol. 144, no. 2, pp. 1–12, Feb. 2022.

[43] B. Kustowski, J. A. Gaffney, B. K. Spears, G. J. Anderson, J. J. Thiagarajan, and R. Anirudh, "Transfer learning as a tool for reducing simulation bias: Application to inertial confinement fusion," *IEEE Trans. Plasma Sci.*, vol. 48, no. 1, pp. 46–53, Jan. 2020.

[44] Z. Zhu and H. Guo, "Design of an RBF surrogate model for low Reynolds number airfoil based on transfer learning," in *Proc. Chin. Control Decis. Conf. (CCDC)*, Jun. 2019, pp. 4555–4559.

[45] G. Pahl, W. Beitz, J. Feldhusen, and K. H. Grote, *Engineering Design, A Systematic Approach*. Berlin, Germany: Springer, 2007.

[46] D. P. Kingma and L. J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Represent.*, San Diego, CA, USA, 2015, pp. 1–15.

[47] K. O. Stanley, J. Clune, J. Lehman, and R. Miikkulainen, "Designing neural networks through neuroevolution," *Nature Mach. Intell.*, vol. 1, pp. 24–35, Jan. 2019.

[48] R. Cipolla, Y. Gal, and A. Kendall, "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7482–7491.

[49] Y. He, X. Feng, C. Cheng, G. Ji, Y. Guo, and J. Caverlee, "MetaBalance: Improving multi-task recommendations via adapting gradient magnitudes of auxiliary tasks," presented at the WWW, Lyon, France, 2022.

[50] J. L. Lin, H. Y. Li, Y. B. Huang, J. H. Chen, P. C. Huang, and Z. Y. Huang, "An efficient modified Hyperband and trust-region-based mode-pursuing sampling hybrid method for hyperparameter optimization," *Eng. Optim.*, vol. 54, no. 2, pp. 1–17, Jan. 2021.

[51] A. Klein, S. Falkner, S. Bartels, P. Hennig, and F. Hutter, "Fast Bayesian optimization of machine learning hyperparameters on large datasets," in *Proc. 20th Int. Conf. Artif. Intell. Statist.*, Fort Lauderdale, FL, USA, 2017, pp. 528–536.

[52] L. Li, K. Jamieson, G. DeSalvo, A. Rostamizadeh, and A. Talwalkar, "Hyperband: A novel bandit-based approach to hyperparameter optimization," *J. Mach. Learn. Res.*, vol. 18, no. 185, pp. 1–52, 2018.

[53] K. T. Fang, "The uniform design: Application of number-theoretic methods in experimental design," *Acta Mathematicae Applicatae Sinica*, vol. 3, no. 4, pp. 363–372, 1980.

[54] M. D. McKay, R. J. Beckman, and W. J. Conover, "A comparison of three methods for selecting values of input variables in the analysis of output from a computer code," *Technometrics*, vol. 21, no. 2, pp. 239–245, 1979.

[55] Y. F. Li and Z. G. Liu, "Method for determining the probing points for efficient measurement and reconstruction of freeform surfaces," *Meas. Sci. Technol.*, vol. 14, no. 8, pp. 1280–1288, Aug. 2003.

[56] Y. Huang and X. Qian, "A dynamic sensing-and-modeling approach to three-dimensional point- and area-sensor integration," *J. Manuf. Sci. Eng.*, vol. 129, no. 3, pp. 623–635, Jun. 2007.

[57] J. Chen, "A method of D optimum design for artificial neural network," *J. Syst. Simul.*, vol. 15, pp. 1586–1588, Nov. 2003.

[58] M. H. Choueiki and C. A. Mount-Campbell, "Training data development with the D-optimality criterion," *IEEE Trans. Neural Netw.*, vol. 10, no. 1, pp. 56–63, Jan. 1999.

[59] L. M. Haines, "Optimal design for neural networks," *New Develop. Appl. Exp. Des.*, vol. 34, pp. 152–162, Jan. 1998.

[60] G. Wang, "Adaptive response surface method using inherited Latin hypercube design points," *J. Mech. Des.*, vol. 125, no. 2, pp. 210–220, Jun. 2003.

[61] L. Hertel, J. Collado, P. Sadowski, J. Ott, and P. Baldi, "Sherpa: Robust hyperparameter optimization for machine learning," *SoftwareX*, vol. 12, pp. 1–23, Jul. 2020.

[62] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *J. Mach. Learn. Res.*, vol. 13, pp. 281–305, Feb. 2012.

[63] K. Swersky, J. Snoek, and R. P. Adams, "Freeze-thaw Bayesian optimization," 2014, *arXiv:1406.3896*.

[64] S. Falkner, A. Klein, and F. Hutter, "BOHB: Robust and efficient hyperparameter optimization at scale," in *Proc. 35th Int. Conf. Mach. Learn.*, Stockholm, Sweden, 2018, pp. 1–10.

[65] S. Issanchou and J. P. Gauchi, "Computer-aided optimal designs for improving neural network generalization," *Neural Netw.*, vol. 21, pp. 945–950, Sep. 2008.

[66] F. Triefenbach, "Design of experiments: The D-optimal approach and its implementation as a computer algorithm," Dept. Comput. Sci., Umea Univ., Umea, Sweden, Tech. Rep. SE-90187, 2008.

[67] A. Miller and N. Nguyen, "A Fedorov exchange algorithm for D-optimal design," *Appl. Stat.*, vol. 43, no. 4, pp. 669–678, 1994.

[68] B. Torsney and S. Mandal, "Two classes of multiplicative algorithms for constructing optimizing distributions," *Comput. Statist. Data Anal.*, vol. 51, no. 3, pp. 1591–1601, Dec. 2006.

[69] B. Torsney and S. Mandal, "Multiplicative algorithms for constructing optimizing distributions: Further developments," in *mODa 7—Advances in Model-Oriented Design and Analysis*. Berlin, Germany: Springer, 2004, pp. 163–171.

[70] J. C. Fu and L. Wang, "A random-discretization based Monte Carlo sampling method and its applications," *Methodol. Comput. Appl. Probab.*, vol. 4, pp. 5–25, Mar. 2002.

JINGLIANG LIN received the B.E. degree in mechanical engineering from the Guangdong University of Technology, Guangzhou, China, in 2008, the M.E. degree in mechanical and electrical engineering from the Guilin University of Electronic Technology, Guilin, China, in 2011, and the Ph.D. degree in mechanical engineering from the Guangdong University of Technology, in 2021.

He is currently a Lecturer at the School of Mechanical Engineering, Guangdong Ocean University. He has published more than a dozen research papers. His research interests include digital design, simulation optimization, and machine learning.

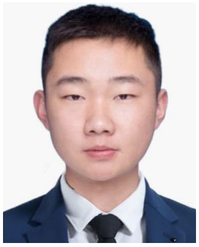
HAIYAN LI received the B.E. and M.E. degrees in mechanical engineering and the Ph.D. degree in marine engineering from the Huazhong University of Science and Technology, Wuhan, China, in 1997, 2006, and 2010, respectively.

She is currently an Associate Professor with the Department of Mechanical and Electrical Engineering, Guangdong University of Technology. She has presided over one National Natural Science Foundation Project, from January 2018 to December 2021, and one Youth Fund Project, from January 2015 to December 2017. She has participated in three projects of the National Natural Science Foundation of China. She has published more than a dozen research papers. Her research interests include artificial intelligence, machine vision, visual inspection, mechanical and electrical product design and development, design optimization, and deep learning.

YUNBAO HUANG received the B.E., M.E., and Ph.D. degrees in mechanical engineering from the Huazhong University of Science and Technology, Wuhan, China, in 1998, 2001, and 2004, respectively.

From May 2005 to September 2007, he conducted postdoctoral research at the Department of Mechanical, Materials, and Aeronautical Engineering, Illinois Institute of Technology. He is currently a Professor with the Department of Mechanical and Electrical Engineering, Guangdong University of Technology. He has presided over three National Natural Science Fund Projects, two National 863 Projects, more than ten Provincial Science and Technology Projects, and has published more than 40 papers. His research interests include big data-driven product design methods, complex product multidisciplinary integrated design optimization, artificial intelligence deep learning, and machine vision inspection.

JUNJIE LIANG is currently pursuing the Ph.D. (Engineering) degree with the School of Mechanical and Electrical Engineering, Guangdong University of Technology, China. His research interests include multi-objective optimization, robot kinematics, dynamics, and product line simulation software development.



SHENG ZHOU received the master's degree in mechanical engineering from the Guangdong University of Technology, Guangzhou, China, in 2021.

He is currently a Research and Development Technician at China Railway Construction Heavy Industry Group Company Ltd. His research interests include multi-field co-simulation and optimization.

ZEYING HUANG received the B.E. and M.E. degrees in mechanical design and manufacturing and automation from the Guangdong University of Technology, Guangzhou, China, in 2019 and 2022, respectively. His research interests include the application of deep learning and transfer learning in the field of mechanical engineering.

GUIMING LIANG is currently pursuing the Ph.D. degree with the School of Mechatronic Engineering, Guangdong University of Technology, China. His research interests include multi-body dynamics, sparse theory, and uncertainty analysis.