Efficient Cutting Power Modeling of Three-Axis Milling Based on Transfer Learning and Neural Network

JIEJUN XIE, PENGCHENG HU, SHIBO GAO, JIHONG CHEN, HUICHENG ZHOU, AND JIANZHONG YANG
School of Mechanical Science and Engineering, Huazhong University of Science and Technology, Wuhan 430074, China
Corresponding author: Pengcheng Hu (pchu@hust.edu.cn)

ABSTRACT The modeling of machining response like the cutting power has great significance for the simulation and optimization of the machining process before real physical cutting. Current cutting power models are usually constructed from some trial cutting experiments under specific cutting conditions, the model constructed under a cutting condition is difficult to apply in another one. To build another model, extensive re-trial cutting experiments should be conducted, which is time-consuming and costly. In this paper, an efficient cutting power modeling approach is proposed based on transfer learning. An instance & model hybrid transfer method for the domain adaption of data from two cutting conditions is proposed, from which the data of one cutting condition can reuse in the modeling process of the other cutting conditions. After the domain adaption process, a boosting technique is then applied that adaptively adjusts the weight of data from different cutting conditions. With the combination of the domain adaption and boosting technique, the cutting power model can be constructed efficiently. Experimental results from two case studies validate that, the cutting power models of three-axis milling as generated from the proposed approach have good prediction performance, which is much superior to the benchmarking algorithms in terms of improving the prediction accuracy and reducing the amount of data required for building the model.

INDEX TERMS Cutting power prediction, three-axis milling, transfer learning, neural network.

I. INTRODUCTION
As one of the most important indicators, the cutting power of the milling operation can reflect the machining state of the machine tool. Efficient and accurate modeling of the cutting power (and also the other cutting responses like the cutting force) has great significance for the simulation and optimization of the machining process, especially for the applications like the optimization of machining quality [1], [2] improvement of machining efficiency [3], reduction of machining costs [4], and enhancement of the service life of cutting tools [5], [6].

There have been extensive works to build the cutting response of machine tools, and they are classified into two categories in general: theoretical modeling and data-driven-based modeling.

The theoretical modeling approaches toward the two most typical cutting responses, i.e., the cutting force and cutting power, are briefly reviewed first. The most mature cutting force modeling method is based on the geometric analysis of the tool and workpiece engagement during the machining process, in which the cutting process of micro-element of tool edge is regarded as an equivalent oblique cutting process and the total cutting force is obtained by integrating the forces along the cutting edges [7]. In addition to the cutting force, the cutting power, which is related to spindle torque and tangential cutting force, has a great impact on tool life and machining energy consumption of machine tools [8], [9]. Aggarwal et al. [8] presented a cutting power model which considers all mechanical and electrical losses of the spindle motor in high-speed milling; the proposed model is further used to identify the coefficients of tangential cutting force for the accurate prediction of cutting forces and chatter-free regions in the process planning of milling.
Xu et al. [10] established an exponential model of cutting power with considering the two cutting modes of down-milling and up-milling. However, the spindle power is affected by mechanical torques (e.g., bearing and windage friction torque [11], [12], cutting torque [8]), and electrical power losses [13]. Although the spindle power estimation models in the existing work could include all mechanical and electrical power loss, they are still difficult to be explained by theoretical models due to the complexity of their influencing factors.

In addition to the theoretical modeling approach, some research has been conducted on the data-driven modeling of cutting responses. Zuperl et al. [14], [15] employed an artificial neural network to predict cutting force during the machining of multi-layered metal material. Vaishnav et al. [16] established a milling force prediction model of a cylindrical milling cutter by combining a mechanistic model with Neural Network (NN) model, for which the feed rate, rotation angle, radial, and axial cutting depth are taken as input for building the model. Xie et al. [3] proposed an NN-based cutting power prediction model that can accurately predict the power of the spindle and be used for optimizing the feed rate of the three-axis milling. Based on the data of cutting power as retrieved from the external sensor, Drouillet et al. [17] focused on tool condition monitoring and proposed a curve fitting method-based artificial NN to predict the remaining life of a cutting tool. Corne et al. [18] fed cutting power data into a NN to predict wear progress, life, and premature breakage of a cutting tool, revealing that the cutting power data are an essential indicator for real-time tool condition monitoring. Pimenov et al. [19] predicted the roughness of machined surfaces based on artificial intelligence methods by using tool wear, machining time, and cutting power as the input.

For the above-mentioned works, no matter for the theoretical modeling and the data-driven modeling, the coefficients or parameters of the models are normally calibrated and determined according to some physical cutting experiments, which are sensitive to the cutting conditions like the material of parts and cutting tool, the geometrics of cutting tool, and the cooling state; different machining conditions have distinct coefficients and parameters for the model. Therefore, the model under one cutting condition is difficult to apply to another one; for example, the cutting power model for machining aluminum cannot be used to predict the power for cutting steel. Under such circumstances, the only feasible way is to rebuild the model for the new cutting condition. However, this is a time-consuming and tedious process, especially for milling operations that have a great many combinations of cutting conditions. How to efficiently and conveniently establish the model under new machining conditions is an urgent and challenging problem. Inspired by the idea of transfer learning in machine learning [20], we proposed a new cutting power modeling approach, which enables the knowledge and experimental data in the old cutting conditions (referring as the source domain) can be migrated and reused for a new cutting condition (referring as the target domain), while only a small amount of experiments need to be conducted for the target domain and the modeling efficiency can be improved drastically.

Regarding the research on the integration of transfer learning with the Computerized Numerical Control (CNC) machining, there are also some works and they can be classified into instance-transfer method and model-transfer method.

The instance-transfer method [21] is designed to directly apply the source domain data when modeling in the target domain. Chen et al. [21] proposed a data-driven method to predict the pose-dependent tool tip dynamics for different cutting tools, in which the data of both the target and source domain is used to train a regression model of the target domain. Based on [21], Liu et al. [22] further proposed a transfer learning-based multimode tool tip dynamics prediction method and greatly reduced the number of impact tests of some new cutting tools. Zhou et al. [23] proposed a transfer learning-based algorithm for the tool selection of CNC machining, in which both a small amount of target domain data and the extensive and low-quality source domain data are utilized to build a high-quality classification model.

The model-transfer method means to (partially) reuse an already trained model of the source domain to the target domain [20]. Based on a limited amount of the relevant images of cutting tools and the basic structure of some typical Convolutional Neural Networks (CNN), Mohamed et al. [24] developed some modified CNNs that can effectively predict the health state of cutting tools. By directly transferring the shallow layers of offline trained CNNs to an online CNN, Xu et al. [25] constructed a deep CNN framework for the online fault diagnosis of mechanical parts like motor and bearing that has desired diagnostic accuracy but needs only limited training time. By leveraging a pre-trained network from non-manufacturing data, Wang et al. [26] further trained it with a small amount of manufacturing data and successfully applied it in machine fault diagnosis.

Although transfer learning has shown great potential in many applications, there is still scarce research on manufacturing applications, especially for the cutting response modeling of machine tools like the cutting power. Based on NN and transfer learning, this paper presents a new method of cutting power modeling for three-axis milling. The cutting power of one cutting condition (source domain) is firstly modeled as the NN from sufficient experimental data; a hybrid transfer method is then designed which transfers both the model and data of the source domain to a new cutting condition (target domain); after that, the target domain model is generated based on a boosting approach that adaptively adjusts the weights of the data of two domains, so that the target domain model could have good prediction performance. Experiments show that the cutting power models generated from transfer learning, which correspond to different cutting conditions, have good prediction performance and are much better than the benchmarks.
This paper is organized as follows. In Section II, a method by representing the cutting power of the source domain as an NN is presented. After that in Section III, a hybrid transfer approach together with a boosting method is proposed for building the model of new cutting conditions. In Section IV, experiments are conducted to verify the effectiveness and advantage of the proposed transfer learning-based modeling approach. This paper concludes in Section V.

II. NEURAL NETWORK-BASED CUTTING POWER MODEL

Under a cutting condition with a particular combination of cutting tool, workpiece, and the cooling condition, the cutting power is susceptible to the machining parameters of feed rate $F$, cutting depth $a_d$, cutting width $a_r$, and spindle speed $S$. Although the other factors such as cutting speed and feed per tooth are also important indexes affecting the cutting power, there can be derived from the $F$, $S$, and the dimension of the cutting tool, i.e., given a cutting tool, the $F$ and $S$ can also reflect the information of cutting speed and feed per tooth.

In this paper, the cutting power is modeled as a multi-hidden layer NN by taking the four machining parameters as the input and the cutting power as output, with the structure of the proposed NN shown in Fig. 1.

For training and validating the NN model of Fig. 1, some experiments are conducted toward a cutting condition $CC_s$ that a three-flute flat-end milling cutter $T_1$ with a diameter of 6mm is utilized to cut the part made of material AL-7075. Details regarding the cutting condition $CC_s$ are shown in Table 1.

For the $CC_s$, we designed some experiments considering all the parameters to model the cutting power, and the NN can be built based on sufficient experimental data. For choosing the number of experimental data, although some work [30], [31] has been done to find the optimal training size for building a good NN model, it is not the key point of this work. In this work, we simply choose to generate 2400 sets of data based on the combination of the parameters: there are 4 spindle speed ranges from 6000 to 12000 rpm/min, 6 feed rate ranges from 500 to 1500 mm/min, 10 cutting depth ranges from 0.3 to 3.0 mm, and 10 cutting width ranges from 0.6 to 6.0 mm. Details regarding the setting of these parameters are listed in Table 2.

Given the 2400 sets of data for $CC_s$, 80% of them are randomly selected to train the network, 10% of them are selected to validate the network, and the remaining 10% are used as the test set. In the training process, the strategies of dropout and early stopping are utilized to avoid overfitting the proposed model. The training process was deployed on the PC with an Intel i7-9700K CPU and RTX 2060Super GPU that runs in Win10 system, Python3.6.5, TensorFlow-GPU1.15.0 environment. In the training process, the log-sigmoid function is taken as the activation function and the Levenberg-Marquardt optimization algorithm is utilized to iteratively update the parameter of the network.
The prediction result of the trained network is shown in Fig. 2, where for the 100 randomly selected samples in the test set, the predicted Mean Relative Error (MRE) is less than 5%. Prediction results show that the proposed NN as trained for CC₆ has good prediction performance on the cutting power.

That is, modeling the cutting power as the NN is feasible and effective.

The 2400 sets of experimental data for CC₆ are taken as the source domain data and will be reused to build the cutting power model of the other cutting conditions. This objective is achieved by a transfer learning method that will be presented in Section III.

### III. TRANSFER LEARNING-BASED MODELING OF CUTTING POWER

The milling processes toward different cutting conditions essentially obey the same physical rules of cutting mechanics, fluid mechanics, and thermodynamics. There is some inherent connection between the cutting power among different cutting conditions. For example, given three cutting conditions that have different tool dimensions or part materials under the same cutting parameters and tool path, as listed in Table 3 and shown in Fig. 3a, their cutting power is different (as shown in Fig. 3b), yet the trend and pattern for the three results are quite similar. One intuitive question is whether the data and model of one cutting condition can be utilized in the other cutting conditions. The transfer learning [20] that emerged in recent years is an affirmative answer. The main idea behind transfer learning is that it is easier to learn new knowledge (e.g., the models for AL-7075 D12 or Steel-45 D6) if you already have some existing knowledge (e.g., data and model for AL-7075 D6).

In this section, a hybrid transfer learning approach based on instance & model transfer is presented to build the cutting power model, with the overall framework of the proposed approach shown in Fig. 4. The proposed approach aims to efficiently build a valid model of the target domain via both

| No. | Diameter of tool (mm) | Material of tool | Flute No. of tool | Materials of part | F (mm/min) | S (r/min) | Cooling condition |
|-----|----------------------|-----------------|------------------|------------------|------------|-----------|------------------|
| 1   | 6                    | Tungsten steel, HRC 60 | 4                | Steel-45        |            |           |                  |
| 2   | 6                    | Tungsten steel, HRC 50 | 3                | AL-7075         | 600        | 3000      | Coolant on       |
| 3   | 12                   | Tungsten steel, HRC 50 | 3                | AL-7075         |            |           |                  |
ample source domain data and only a handful of target domain data. For defining the cutting power model of the target domain, an instance-transfer method is proposed by mapping the instance of the source domain to the target domain via a network called Domain Adaptive Neural Network (DaNN), while the DaNN is defined by a model-transfer-based approach that depends on the already trained source domain NN as well as the technique of fine-tune [32]. Given the instances of the target domain and these mapped from the source domain, a boosting approach is utilized to adjust the weights of the two types of instances adaptively. In this way, the cutting power model of the new cutting condition can be established efficiently.

A. HYBRID TRANSFER-BASED DOMAIN ADAPTATION

Before the proposed DaNN can be constructed, some notations and definitions are introduced as follows:

$X$ denotes the instance space, namely the possible combination of all the machining parameters. For a $x_i \in X$, it has $x_i = \{a_{i1}, a_{i2}, F_i, S_i\}$, where $a_{i1}$, $a_{i2}$, $F_i$, and $S_i$ are respectively the cutting depth, cutting width, feed rate, and spindle speed. $X_s$ and $X_t$ respectively represent the instance space of the source domain and target domain, as indicated by the subscript ‘s’ and ‘t’.

$Y$ denotes the label space, that is, the cutting power of instances. For a $y_i \in Y$, $y_i = \{s_p\}$, where $s_p$ is the cutting power of the $i$-th instance. $Y_s$ and $Y_t$ are the corresponding label space of $X_s$ and $X_t$.

With the above definition, the data of the source domain and target domain are defined as

$$T_s = \{[x^s_i, y^s_i]\} \in R^{n_s \times 5} \tag{1}$$

$$T_t = \{[x^t_i, y^t_i]\} \in R^{n_t \times 5} \tag{2}$$

where in (1) and (2), $x^s_i \in X_s$, $y^s_i \in Y_s$, $i = 1, \ldots, n$; $x^t_i \in X_t$, $y^t_i \in Y_t$, $i = 1, \ldots, m$; $n$ and $m$ are the number of samples in the source domain and target domain, respectively; for a transfer learning task, $m$ is normally much smaller than $n$.

To make the source domain data $T_s$ can be applied in the cutting power modeling of the target domain, the distribution difference of data in the two domains should be reduced. Domain adaptation can be utilized to achieve this objective [33]. This paper presents a DaNN to address the domain adaption of data from the source domain to the target domain. By taking the structure of the already trained NN (i.e., the NN as shown in Fig. 1) as a reference, a DaNN is firstly trained by considering the distribution difference between the $T_s$ and $T_t$, and then a fine-tune process is conducted to further update the coefficients of the adaption layer of the DaNN. The specific processes for the domain adaption and fine-tune of DaNN are shown in Fig. 5 and Fig. 6, respectively.

The proposed DaNN aims to reduce the distribution difference between $T_s$ and $T_t$. Given the $T_s \in R^{n_s \times 5}$ and the $T_t \in R^{n_t \times 5}$, the DaNN means to generate the transferred data $T_s' \in R^{n_t \times 5}$ from $T_s$, so that $T_s'$ and $T_t$ have a much similar distribution.

In this work, the distribution difference between two domains is measured by the Maximum Mean Discrepancy (MMD) [34], a metric widely used to evaluate the performance of transfer learning. The MMD measures the distribution distance of two data domains in the Reproducing Kernel Hilbert Space (RKHS). Given the data of two domains $D_s$ and $D_t$, their MMD is calculated as

$$\text{MMD} (D_s, D_t) = \left\| \frac{1}{n_s} \sum_{i=1}^{n_s} \phi (d^s_i) - \frac{1}{n_t} \sum_{i=1}^{n_t} \phi (d^t_i) \right\|_H \tag{3}$$

where, $d^s_i$ and $d^t_i$ respectively represent the instance of $D_s$ and $D_t$; $n_s$ and $n_t$ are respectively the number of instances of the two domains; $\phi(\cdot)$ is the kernel function to map the instance of both $D_s$ and $D_t$ to the RKHS; the MMD of (3) measures the distance of the kernel embedding of distributions $\phi(D_s)$ and $\phi(D_t)$.

With the MMD defined in (3), the reduction of distribution difference between $T_s$ and $T_t$ means to find a mapping $\psi(\cdot)$ which minimizes the distance between the $T'_s = \psi(T_s)$ and $T_t$, i.e.,

$$\min (\text{MMD} (\psi (T_s), T_t)) \tag{4}$$

For our transfer learning problem, $m$, the size of target domain data, is often much smaller than source domain size $n$; besides, there is high nonlinearity for the result of the cutting power [3]. Therefore, it is difficult to define an explicit $\psi(\cdot)$ pertaining to the (4). In this work, the function $\psi(\cdot)$ is modeled as a NN (i.e., the DaNN as shown in Fig. 5) since NN can essentially approximate any type of nonlinear function, and the (4) is taken as the (part of) loss function when training the DaNN.

In our work, the structure of the proposed DaNN is chosen as the same as that of the source domain NN (i.e., the NN shown in Fig. 1) which has already been validated for its effectiveness in extracting the feature and modeling cutting power. By taking the four machining parameters of each source domain data as input, the estimated value from the proposed DaNN is taken as a new cutting power with which the four machining parameters are combined to form the corresponding transferred data, as shown in Fig. 5. In this way, the transferred data $T'_s$ can be generated from $T_s$ with the
reason why it contains the other term is because be minimized in the training process of the DaNN. The distribution loss between the transferred data and the target data, and \( l_1(T'_s, T_t) \) represents the distribution difference between the transferred data and the target data, and \( l_1(T'_s, T_t) = \text{MMD}(T'_s, T_t) \). \( \lambda \) is a manual-defined weight parameter, controlling the importance of \( l_1(T'_s, T_t) \) as compared to \( l_c(T_t) \). In our current implementation, we manually set \( \lambda = 1 \).

In (5), the term \( l_1(T'_s, T_t) \) reflects the distribution difference between the \( T'_s \) and \( T_t \), and this is a factor aiming to be minimized in the training process of the DaNN. The reason why it contains the other term is because \( l_c(T_t) \) simultaneously enforces the trained parameters of the DaNN embedding the knowledge and features of the cutting power, therefore, making the transferred data \( T'_s \) have good feasibility for building the target domain model.

By representing the \( l_c(T_t) \) and \( l_1(T'_s, T_t) \) respectively as \( l_c(T_t) = (1/n) \sum (|f(x'_s) - y'_t|/y'_t) \) and \( l_1(T'_s, T_t) = \text{MMD} \left( [x'_s, f(x'_s)] , T_t \right) \), the loss function of (5) is expanded as

\[
l = l_c(T_t) + \lambda l_1(T'_s, T_t) \tag{5}
\]

Given the loss function as defined in (6), the DaNN with the structure of Fig. 5 can be trained with the same method as the NN of the CC.

Because there is both regression loss \( l_c(T_t) \) and domain distribution loss \( l_1(T'_s, T_t) \) in (6), the model will neither completely fit the data of the source domain nor the data of the target domain but find a balance between them. The model reduces the distribution difference between the source domain and the target domain while still learning the source domain’s knowledge. Due to \( l_c(T_t) \) in (5), the distribution of the transferred data \( T'_s = \left\{ [x'_s, f(x'_s)] \right\} \) cannot be completely close to that of the \( T_t \). To address this issue, the fine-tune method [35] is utilized to adjust some parameters of the initial trained DaNN, to further reduce the distribution difference between the output data and the target domain data.

The fine-tuning process is shown in Fig. 6. We firstly freeze the structure of the initial trained DaNN and its coefficients of the first layer (input layer) and the second layer. For the mapping of the third layer, i.e., \( sp = \delta_3(W_3x_3 + b_3) \) (where, \( x_3 \) is the vector of the third layer, \( W_3 \) and \( b_3 \) are the coefficients, and \( \delta_3 \) is the activation function), the coefficients \( W_3 \) and \( b_3 \) are then further trained by target domain data \( T_t \) with the loss function \( L_F = l_c(T_t) \). A lower learning rate of 0.001 was used in the fine-tuning process, which is 1/10 of the learning rate in the initial training process.

After the fine-tune on the DaNN, the distribution error between the transferred data \( T'_s = [x'_s, f(x'_s)] \in \mathbb{R}^{n \times 5} \) and the target domain data \( T_t \) can be further reduced. Then, both the \( T'_s \) and \( T_t \) can be utilized to generate the cutting power model for \( T_t \).

Note that if the performance of the proposed DaNN after fine-tune is not good enough, we can further improve the model by increasing the amount of target domain data or adjusting the parameters of more layers. However, in the current practice of our paper, the fine-tuning process works very well. Details could refer to Section IV. C.

For the proposed DaNN, its general objective is to transfer the data from the source domain to the target domain so that their distribution difference can be reduced, and it essentially belongs to an instance-transfer method. However, when defining the DaNN, its structure is chosen as the same as the NN of the source domain; besides, in the process of fine-tuning, only partial parameters of the network are updated; both strategies for the definition and training the DaNN belong to the model-transfer method. With considering both properties, this is a hybrid method with both properties of instance and model-transfer.

In the next section, we will present how the data of the source domain and target domain can be utilized simultaneously so that the cutting power model of the target domain can be constructed with good prediction performance.

### B. BOOSTING-BASED CUTTING POWER MODELING

Although the distribution difference between the source domain data \( T_s \) and target domain data \( T_t \) is reduced via the proposed DaNN, their difference cannot be eliminated. When defining the cutting power model of the target domain, \( T_t \) should have better credibility than the transferred data \( T'_s \) since \( T_t \) is the real retrieved data. In this section, a boosting approach is utilized to address the different statuses of \( T'_s \) and \( T_t \) so that \( T_t \) combined with \( T'_s \) could be utilized to generate a model that has good prediction performance.

Given \( T'_s \) and \( T_t \), the general idea of TrAdaBoost.R2 [36], [37] is utilized for the modeling of the target domain model, in which the weights of instances in \( T'_s \) and \( T_t \) are adaptively adjusted so that the final model can be generated by combining a set of basic NNs generated in the boosting process.
Fig. 7 shows the general boosting process for building the cutting power model of the target domain. An instance set \( T = T'_t \cup T_t \) is defined based on \( T'_t \in R^{n_t \times 5} \) and \( T_t \in R^{n_t \times 5} \), which is utilized for training the NN of cutting power (with the structure of NN shown in Fig. 1) iteratively. Within each round of iteration, a boosting process is conducted: for any instance \( x \in T \) with large prediction error, if \( x \in T'_t \), its weight will be decreased to reduce its influence in the next round of iteration; otherwise, the weight will be increased to enhance its importance. The process of adjusting the weights of instances and the training of the NN are iteratively conducted until the boosting process is terminated. Finally, a model \( H(x) \) can be generated by aggregating the NNs (i.e., the \( h_1, h_2, \ldots, h_N \) as shown in Fig. 7 and \( N \) is the number of iterations) as generated in the iteration process.

Detailed implementation of the boosting approach is explained below.

For the 1\(^{st} \) iteration, the NN model \( h_1 \) can be trained based on an initial set of weights \( w^1 = (w^1_1, w^1_2, \ldots, w^1_{n+m}) \) assigned to the instances in \( T \), and

\[
w^1_i = \begin{cases} 
1, & i = 1, \ldots, n \\
1/m, & i = n + 1, \ldots, n + m
\end{cases}
\]  

(7)

where for the \( w^j_i \), the superscript ‘\( j \)’ represents the iteration number and the subscript ‘\( i \)’ is the index of the instance in \( T \).

For the \( t \)-th iteration (\( t \in (1, \ldots, N) \)) with its weight \( w^t \), where \( N \) is the manually selected iteration number, a base learner \( h_t : x \rightarrow y \) that is represented as the NN (with its structure shown in Fig. 1) can be trained based on the \( T \) and \( w^t \). To update the \( w^t \), the normalized prediction errors of instance (e.g., the \( i \)-th instance) is calculated as

\[
e^t_i = \frac{|(y_i - h_t(x_i))/|y_i|}{E_t}
\]  

(8)

where the \( E_t \) represents the maximum normalized training error for the instances in \( T_t \) since more attention is paid to the prediction performance of the target domain, and

\[
E_t = \max_{i=n+1}^{n+m} \frac{|(y_i - h_t(x_i))/|y_i|}{E_t}
\]  

(9)

The normalized prediction error per (8) is only for one instance of \( T_t \). To reflect the overall error of the target domain, the weighted sum of \( e^t_i \) for all instances in \( T_t \) is calculated, as denoted by \( \varepsilon_t \) and

\[
\varepsilon_t = \sum_{i=n+1}^{n+m} \frac{w^t_i e^t_i}{\sum_{i=n+1}^{n+m} w^t_i}
\]  

(10)

If \( \varepsilon_t \geq 0.5 \), it means that the performance of the trained NN-\( t \) \( h_t \) is poor; as a result, the iteration is terminated and the training result in the current iteration is abandoned. Otherwise, the base learner \( h_t \) meets the requirement of good accuracy, and the weight vector for the (\( t+1 \))-th iteration can be updated as:

\[
w^{t+1}_i = \begin{cases} 
\frac{1}{\beta^t_i}, & i = 1, \ldots, n \\
\frac{1}{\beta^t_i}, & i = n+1, \ldots, n+m
\end{cases}\]

(11)

where, \( \beta^t_i \) and \( \beta \) are two variables, and \( \beta_t = \varepsilon_t/(1-\varepsilon_t) \) and \( \beta = 1/\left(1 + \sqrt{2 \ln n/N}\right) \).

Based on the updated weight vector \( w^{t+1} \), the NN \( h_{t+1} \) in the (\( t+1 \))-th iteration can be trained similarly as the NN of Fig. 1. After \( N \) rounds of iterations, a set of basic learner \( \{h_1, h_2, \ldots, h_N\} \) can be generated, and the final model for predicting the cutting power is defined by aggregating the NNs of the last \( \lfloor N/2 \rfloor \) iterations,

\[
H(x) = \sum_{t=\lceil N/2 \rceil}^{N} \ln \left( \frac{1}{\beta^t} \right) h_t(x)/ \sum_{t=\lceil N/2 \rceil}^{N} \ln \left( \frac{1}{\beta^t} \right)
\]  

(12)

where the term \( \ln \left( \frac{1}{\beta^t} \right)/ \sum_{t=\lceil N/2 \rceil}^{N} \ln \left( \frac{1}{\beta^t} \right) \) is the normalized factor for aggregating the NNs; \( H(x) \) is the generated target domain model.

In this section, a hybrid transfer-based DaNN is presented to map the data from the source domain to the target domain, based on which the distribution difference between the two domains can be significantly reduced. To further address the remaining difference, a boosting method is designed that adaptively adjusts the weight of the instances of the two domains, with the final cutting power model of the target domain aggregated from the NNs trained in the boosting process. The effectiveness and advantage of the proposed transfer learning-based cutting power modeling method will be presented in Section IV.

IV. CASE STUDY AND DISCUSSIONS

In this section, two cases are implemented to verify the effectiveness and advantage of the proposed transfer learning-based cutting power modeling approach.

A. EXPERIMENTAL SETUP

The 2400 sets of experimental data toward the cutting condition \( CC_s \), as listed in Table 1, are taken as source domain data. Cutting conditions \( CC_{t1} \) and \( CC_{t2} \) are defined.
respectively for two target domains. Given CC<sub>s</sub>, CC<sub>t1</sub>, and CC<sub>t2</sub>, two tasks (i.e., Task-1 and Task-2) are designed which aim to construct the cutting power models for CC<sub>t1</sub> and CC<sub>t2</sub> by transferring the data from source domain CC<sub>s</sub> to them, respectively. Details for CC<sub>t1</sub> and CC<sub>t2</sub> are listed in Table 4. Compared with CC<sub>s</sub> as listed in Table 1, CC<sub>t1</sub> has the cutting tool of a different dimension (12 mm vs. 6 mm), while CC<sub>t2</sub> has different types of cutting tools in terms of materials (tungsten steel, HRC 50 vs. tungsten steel, HRC 60) and flute No. (4 vs. 3) and also different materials of a part (AL-7075 vs. steel-45). CC<sub>t1</sub> and CC<sub>t2</sub> are two typical new cutting conditions because changing the cutting tool and materials of a part are very common in industrial practices of CNC milling.

Given CC<sub>t1</sub> and CC<sub>t2</sub>, experiments are conducted to generate the data of the two target domains, with the setting on the four machining parameters listed in Table 5 and Table 6, respectively. For CC<sub>t1</sub> and CC<sub>t2</sub>, there are 300 and 192 sets of data, respectively, which are much less than the source domain which has 2400 sets of data.

The physical cutting experiments for CC<sub>t1</sub>, CC<sub>t2</sub>, and CC<sub>s</sub> are all conducted on a three-axis machine center equipped with an HNC-9 numerical controller, as shown in Fig. 8. The machining data is retrieved from a data acquisition platform of HNC-9, where the cutting power, actual feed rate, and spindle speed are directly collected from the controller and the cutting width and depth are defined in Table 1, Table 5, and Table 6.

B. EXPERIMENTAL RESULTS ON TRANSFER LEARNING-BASED CUTTING POWER MODELING

Task-1 and Task-2 are conducted according to the transfer learning-based method as presented in Section III. Taking Task-1 (i.e., cutting power modeling for CC<sub>t1</sub>) as an example, a domain adaption method proposed in Section III is utilized to build a DaNN so that the distribution difference between the 2400 sets of source domain data and the 300 sets of target domain data can be reduced. The data distribution difference is measured by MMD as defined in (3), with the value listed in Table 7. After the domain adaption, the MMD is reduced from 0.2097 to 0.0687, which has a 67.24% reduction. That is, the distribution difference between source domain data and target domain data is significantly reduced by the proposed domain adaptation method via the DaNN. After the domain adaption, a total of 2700 sets of data (2400 sets of transferred data and 300 target domain data)
TABLE 8. MRE value of our method and the four benchmarks.

|            | Our method | Benchmark-1 | Benchmark-2 | Benchmark-3 | Benchmark-4 |
|------------|------------|-------------|-------------|-------------|-------------|
| CC₁        | 4.75%      | 36.67%      | 18.41%      | 13.89%      | 10.59%      |
| CC₂        | 3.13%      | 36.95%      | 17.39%      | 12.85%      | 8.09%       |

are utilized to generate the cutting power model for CC₁. As there is still some noticeable distribution difference between the data of the two domains (0.0687 as listed in Table 7), when generating the cutting power model for CC₁, the weights of instances from the source domain and the target domain are adaptively adjusted according to the boosting approach presented in Section III. In this way, the cutting power model for CC₁, as denoted by $H_1(\cdot)$, can be generated.

The model $H_1(\cdot)$ is tested against another 100 sets of data collected for CC₁, with the experimental results shown in Fig. 9 and Fig. 10. The results show that the transfer learning-based cutting power model has good prediction performance, where, 92% of the samples have prediction errors within $\pm10\%$, and 60% of the samples have prediction errors within $\pm5\%$. The MRE of the prediction result (listed in Table 8) is

$$\text{MRE}_1 = \frac{1}{100} \sum_{i=1}^{100} |H_1(x_i) - y_i| / y_i = 4.75\%$$  \hspace{1cm} (13)

where, $x_i$ and $y_i$ are respectively the input (the four machining parameters) and output (cutting power) of the $i$-th testing instance.

For Task-2, the cutting power model $H_2(\cdot)$ is built similarly to that of $H_1(\cdot)$: firstly, the 2400 sets of data from CC₅ is mapped to the domain of CC₂, with the MMD reduced from 0.3064 to 0.1519 (a 50.42% reduction); and then, the boosting technique is applied to a total 2592 sets of data to generate the $H_2(\cdot)$. Similarly, another 100 sets of experiments are conducted to verify the prediction model $H_2(\cdot)$, with the prediction results shown in Fig. 11 and Fig. 12. Compared with the real measured cutting power data, 97% of the data have the prediction error within $\pm10\%$, and 81% of the data have the error within $\pm5\%$; regarding the overall prediction error, the MRE is $\text{MRE}_2 = 3.13\%$.

C. COMPARISONS

To validate the advantage of the proposed approach, two kinds of comparison are conducted: (1) comparison with
benchmarks of the same dataset, and (2) comparison with NN trained with sufficient target domain data.

1) COMPARISON WITH BENCHMARKS OF THE SAME DATASET
In addition to the proposed transfer learning-based approach as presented in Section III, four benchmarking strategies are implemented for building the cutting power models of CC\textsubscript{t1} and CC\textsubscript{t2}, which are based on the same datasets as our approach.

Benchmark-1 (Model Trained With the Source Domain Data Only): The cutting power models of CC\textsubscript{t1} and CC\textsubscript{t2} are represented as the NNs with the same structure as that in Fig. 1, and the parameters of NNs are directly trained from the 2400 sets of source domain data.

Benchmark-2 (Model Trained With a Small Amount of Target Domain Data Only): Two NNs are trained with the 300 and 192 sets of target domain data only for CC\textsubscript{t1} and CC\textsubscript{t2}, respectively.

Benchmark-3 (Model Defined From the Boosting Method Without Domain Adaption): Based on the boosting approach [36], [37], the 2400 sets of data for CC\textsubscript{s} and target domain data for CC\textsubscript{t1} and CC\textsubscript{t2} are utilized to define the target domain models without the pre-process of domain adaption.

Benchmark-4 (Model Defined From Affine Transformation-Based Domain Adaption Method and the Follow-Up Boosting Technique): This implementation is the same as our approach except the domain adaption is realized by an affine transformation-based method [21].

The models generated from the four benchmarks are tested against another 50 sets of data retrieved for CC\textsubscript{t1} and CC\textsubscript{t2}. The prediction results of the models from benchmark-1 are shown in Fig. 13, with the MRE being 36.67% and 36.95% for CC\textsubscript{t1} and CC\textsubscript{t2}. For the models from benchmark-2, the MRE is 18.41% and 17.39%, respectively, as shown in Fig. 14. For the models from benchmark-3 with the results shown in Fig. 15, the MRE is 13.89% and 12.85%, respectively. Finally, the MRE of the models from benchmark-4 is 10.59% and 8.09% for CC\textsubscript{t1} and CC\textsubscript{t2}, as shown in Fig. 16.

The MRE of the prediction results from our method and the benchmarks are listed in Table 8. Compared with our
proposed approach, benchmark-1 and benchmark-2 have the worse prediction results for both cutting conditions. The predicted MRE is still extensive regarding benchmark-3, which relies on the boosting approach and does not involve the domain adaption process. The best result among the four benchmarks is the benchmark-4, in which the model is defined based on the affine transformation-based domain adaption and boosting technique, and the predicted MRE is 10.59% and 8.09% for $CC_{t1}$ and $CC_{t2}$, which, however, is still much worse than that of our approach.

2) COMPARISON WITH THE NN TRAINED WITH SUFFICIENT TARGET DOMAIN DATA
To further verify the advantage of the proposed approach, our models are compared with the NN models that are directly trained from sufficient target domain data.

For $CC_{t1}$ and $CC_{t2}$, more experiments are conducted to generate sufficient experimental data, with the setup of the experiments listed in Appendix, where, there are 2430 sets of data for $CC_{t1}$ and 1536 sets of data for $CC_{t2}$. Given the sufficient target domain data, the cutting power model, which is also a NN with the same structure as Fig. 1, can be trained. In this work, we have designed two strategies to train the NN model:

Strategy-1: NN model trained from the 60% randomly selected target domain data, i.e., 1458 sets of data for $CC_{t1}$ and 922 sets of data for $CC_{t2}$.

Strategy-2: NN model trained from the 80% randomly selected data, i.e., 1944 sets of data for $CC_{t1}$ and 1229 sets of data for $CC_{t2}$.

Given the above two strategies, the NN models for $CC_{t1}$ and $CC_{t2}$ are trained with the same settings as these of the source domain. The prediction error of both strategies is listed in Table 9. For $CC_{t1}$, the predicted MRE of our approach is between these of the two strategies (6.19% and 4.20%, respectively), that is, the prediction result of our model (with only 300 target domain instances) is better than the strategy-1 with 1458 instances and a little worse than the strategy-2 with 1944 instances. That is, for the same prediction error, our method can reduce the number of target domain instances for a ratio between 79.42% and 84.57%. Regarding the $CC_{t2}$, both strategy-1 and strategy-2 have a worse predicted MRE than our approach, i.e., the number of target domain instances...
TABLE 9. MRE value of our method and the two strategies.

|       | Our method | Strategy-1 | Strategy-2 |
|-------|------------|------------|------------|
| CC1   | 4.75%      | 6.19%      | 4.20%      |
| CC2   | 3.13%      | 5.16%      | 3.71%      |

TABLE 10. The model building time of our method and the Strategy-2.

| Model   | Experimental time | Training time | Total time  |
|---------|-------------------|---------------|-------------|
| CC1 (our method) | 37 min 13 s | 11 min 42 s | 48 min 55 s |
| CC1 (Strategy-2) | 4 h 1 min 12 s | 1 min 16 s | 4 h 2 min 28 s |
| CC2 (our method) | 31 min 42 s | 10 min 39 s | 42 min 21 s |
| CC2 (Strategy-2) | 3 h 22 min 54 s | 51 s | 3 h 23 min 45 s |

is reduced by more than 84.38% for the same prediction accuracy.

As for the time for building models, the experimental time and model training time of CC1 and CC2 are shown in Table 10, with the total time for our method being 48 min 55 s and 42 min 21 s, respectively. For the benchmark Strategy-2, the total model building time is 4 h 2 min 28 s and 3 h 23 min 45 s, respectively. That is, for CC1 and CC2, our method can reduce the modeling time by 79.8% and 79.2% when modeling for the same level of prediction accuracy.

D. DISCUSSIONS

Case studies toward two cutting conditions have validated the feasibility of the proposed transfer learning-based cutting power modeling approach, from which the predicted cutting power matches the actual value very well with the predicted MRE being 4.75% for CC1 and 3.13% for CC2. Such prediction results can meet the requirement of many industrial applications like the feed rate optimization of milling [3] and service life optimization of machine tools and cutting tools [5].

The superiority of the proposed approach is testified as compared with four benchmarks. The low prediction accuracy for benchmark-1 and benchmark-2 shows that constructing the models directly from the data of a different cutting condition or a small amount of data is infeasible. Compared with benchmark-1 and benchmark-2, the boosting method as utilized in benchmark-3 can somehow improve the performance of the generated model, yet the prediction accuracy is still not satisfactory due to large distribution deviation differences between different domains. By further adding an affine transformation-based domain adaption process to the benchmark-3, the benchmark-4 can improve the prediction accuracy further; however, it is nonetheless much worse than our approach because the affine transformation is essentially a linear mapping without catering to the nonlinear properties between different domains. By designing a hybrid transfer learning method via the DaNN and utilizing the boosting method, our transfer learning-based cutting power modeling approach has much better prediction results than benchmarks.

When comparing the models from our approach with those constructed from sufficient target domain data, it is proved that our approach can save more than 80% of the target domain data while the prediction accuracy of the two kinds of approaches lies at the same level. This further validates the advantage of the proposed transfer learning-based method in terms of reducing the dataset size in the modeling process.

There are some limitations to our current work. The proposed method has only been verified with limited types of cutting tools and materials of part; besides, the other cutting condition like the cooling condition remains unchanged in the entire experimental process; what is more, the cutting power is chosen as the response of the machine tool, which is relatively simple than the other responses like the cutting force, tool wear, machining stability, etc. However, the proposed hybrid transfer learning-based method combined with the boosting technique has been proven to be feasible and much superior to the benchmarks; it also has good potential applications for modeling the other types of responses of the other cutting conditions.

V. CONCLUSION

In this paper, a transfer learning-based cutting power modeling method is presented based on the integration of domain adaption with a boosting technology. A hybrid transfer learning approach is designed to realize the domain adaption via the DaNN and the follow-up fine-tune, based on which the distribution difference between the source domain the target domain data is significantly reduced. To consider the remaining difference after the domain adaption, a boosting method is designed that adaptively adjusts the weight of the data from both the source domain and target domain, and the cutting power model is then defined by aggregating the models as generated in the boosting process. Given two new cutting conditions, the models constructed from our method have good prediction performance, which validates the effectiveness of our approach. By comparing with the benchmarks, our models are also much superior in terms of improving the prediction performance, reducing the size of the dataset, and decreasing the total time required for building the model.

Our future work will focus on three aspects. In addition to the NN, some other machine learning models like the support vector regression model and random forest model will be utilized to represent the cutting power, and our transfer learning-based method will be implemented toward these models. Besides, some other factors such as cutting force, tool wear, and machining stability will be incorporated into our current hybrid transfer learning framework so that it can provide more practical applications. At last, more experiments will be conducted on the cutting conditions that have more distinct features like cutting tools, materials of part, and cooling conditions.
### APPENDIX

**EXPERIMENTS FOR GENERATING SUFFICIENT TARGET DOMAIN DATA**

For training the NN of the target domain with sufficient target domain data, a total of 2430 and 1536 sets of experiments were designed and conducted for CC$_{11}$ and CC$_{12}$, respectively. Details on the setting of the experiments are listed in Table 11 and Table 12. The physical cutting experiments are conducted in the machining scenario as presented in Section IV.

**TABLE 11. Experiments conducted to generate sufficient data for CC$_{11}$**

| No. | $a_p$ (mm) | $a_v$ (mm) | $F$ (mm/min) | $S$ (t/min) |
|-----|------------|------------|--------------|-------------|
| 1   | 0.6        | 2.4        |              |             |
| 2   | 0.9        | 3.6        |              |             |
| 3   | 1.2        | 4.8        |              |             |
| 4   | 1.5        | 6.0        | 600, 800, 1000, 1200, 1400, 1600 | 5000, 7000, 9000, 11000, 13000 |
| 5   | 1.8        | 7.2        | 700          | 3000, 3500, 4000, 4500 |
| 6   | 2.1        | 8.4        | 800          |             |
| 7   | 2.4        | 9.6        |              |             |
| 8   | 2.7        | 10.8       |              |             |
| 9   | 3.0        | 12.0       |              |             |

**TABLE 12. Experiments conducted to generate sufficient data for CC$_{12}$**

| No. | $a_p$ (mm) | $a_v$ (mm) | $F$ (mm/min) | $S$ (t/min) |
|-----|------------|------------|--------------|-------------|
| 1   | 0.3        | 400        |              |             |
| 2   | 0.6        | 500        |              |             |
| 3   | 0.9        | 600        |              |             |
| 4   | 1.2        | 0.6, 1.2, 1.8, 2.4, 3.0, 3.6 | 700, 800, 1000, 1200, 1400, 1600 | 3000, 3500, 4000, 4500 |
| 5   | 1.5        | 700        |              |             |
| 6   | 1.8        | 900        |              |             |
| 7   | 2.1        | 1000       |              |             |
| 8   | 2.4        | 1100       |              |             |

### REFERENCES

[1] Q. Guo, M. Wang, Y. Xu, and Y. Wang, “Minimization of surface roughness and tangential cutting force in whirlwind milling of a large screw,” *Measurement*, vol. 152, Feb. 2020, Art. no. 107256.

[2] A. T. Abbas, D. Y. Pimenov, I. N. Erdakov, T. Mikolajczyk, M. S. Soliman, and M. M. El Rayes, “Optimization of cutting conditions using artificial neural networks and the edgeworth-pareto method for CNC face-milling operations on high-strength grade-H steel,” *Int. J. Adv. Manuf. Technol.*, vol. 105, nos. 5–6, pp. 1323–1339, May 2021.

[3] H. Ning, H. Zheng, S. Zhang, and X. Yuan, “Milling force prediction model development for CFRP multidirectional laminates and segmented specific cutting energy analysis,” *Int. J. Adv. Manuf. Technol.*, vol. 113, nos. 9–10, pp. 2437–2445, Apr. 2021.

[4] Z. Huang, J. Zhu, J. Lei, X. Li, and F. Tian, “Tool wear predicting based on multi-domain feature fusion by deep convolutional neural network in milling operations,” *J. Intell. Manuf.*, vol. 31, pp. 953–966, Aug. 2019.

[5] D. Y. Pimenov, A. Bustillo, S. Wojciechowski, V. S. Sharma, M. K. Gupta, and M. Kuntoğlu, “Artificial intelligence systems for tool condition monitoring in machining: Analysis and critical review,” *J. Intell. Manuf.*, pp. 1–43, Mar. 2022.

[6] Y. Altintas and A. Ber, “Manufacturing automation: Metal cutting mechanics, machine tool vibrations, and CNC design,” *Appl. Mech. Rev.*, vol. 54, no. 5, pp. B84–B94, Sep. 2001.

[7] A. Hughes and B. Drury, *Electric Motors and Drives: Fundamentals, Types and Applications*. Boston, MA, USA: Newnes, 2019.

[8] U. Zuperl, F. Čuš, and T. Irgolic, “Prediction of cutting forces in ball-end milling of multi-layered metal materials,” *J. Mech. Eng.*, vol. 62, no. 6, pp. 340–350, 2016.

[9] G. Chen, Y. Li, and X. Liu, “Pose-dependent tool tip dynamics prediction based on transfer learning,” *Int. J. Mach. Tools Manuf.*, vol. 15, no. 1, pp. 5–17, Mar. 2020.

[10] T. A. Harris, *Rolling Bearing Analysis*. Hoboken, NJ, USA: Wiley, 2001.

[11] B. Bossmanns and J. F. Tu, “A power flow model for high speed motorized spindles—Heat generation characterization,” *J. Manuf. Sci. Eng.*, vol. 123, no. 3, pp. 494–505, 2001.

[12] U. Zuperl, F. Čuš, A. Zawada-Tomkiewicz, and K. Stepien, “Neuro-mechanistic model for cutting force prediction in helical end milling of metal materials layered in multiple directions,” *Adv. Prod. Eng. Manage.*, vol. 15, no. 1, pp. 5–17, Mar. 2020.

[13] S. Vaishnav, A. Agarwal, and K. A. Desai, “Machine-learning-based instantaneous cutting force model for end milling operation,” *J. Intell. Manuf.*, vol. 31, no. 6, pp. 1353–1366, Aug. 2020.

[14] C. Drouillet, J. Karandikar, C. Nath, A.-C. Journeaux, M. El Mansori, and T. Kurfess, “Tool life predictions in milling using spindle power with the neural network technique,” *J. Manuf. Processes*, vol. 22, pp. 161–168, Apr. 2016.

[15] R. Corne, C. Nath, M. El Mansori, and T. Kurfess, “Study of spindle power data with neural network for predicting real-time tool wear/breakage during inconel drilling,” *J. Manuf. Syst.*, vol. 43, pp. 287–295, Apr. 2017.

[16] S. S. Miriyala, P. Mittal, S. Majumdar, and K. Mitra, “Comparative study of artificial neural networks,” *CIRP Ann. Manuf. Techn.*, vol. 426, Dec. 2021, Art. no. 131221.

[17] S. S. Miriyala, V. R. Subramanian, and K. Mitra, “TRANSFORM-ANN as case study,” *Appl. Mech. Rev.*, vol. 64, no. 1, pp. 294–309, Jan. 2018.
[31] R. K. Inapakurthi, S. S. Miriyala, and K. Mitra, “Recurrent neural networks based modelling of industrial grinding operation,” Chem. Eng. Sci., vol. 219, Jun. 2020, Art. no. 115585.

[32] M. Ghifary, W. B. Kleijn, and M. Zhang, “Domain adaptive neural networks for object recognition,” in Proc. Pacific Rim Int. Conf. Artif. Intell., Cham, Switzerland: Springer, 2014, pp. 898–904.

[33] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, “Domain adaptation via transfer component analysis,” IEEE Trans. Neural Netw., vol. 22, no. 2, pp. 199–210, Feb. 2011.

[34] K. M. Borgwardt, A. Gretton, M. J. Rasch, H.-P. Kriegel, B. Scholkopf, and A. J. Smola, “Integrating structured biological data by kernel maximum mean discrepancy,” Bioinformatics, vol. 22, no. 14, pp. e49–e57, Jul. 2006.

[35] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, “How transferable are features in deep neural networks?” in Proc. Adv. Neural Inf. Process. Syst. (NIPS), Dec. 2014, pp. 3320–3328.

[36] W. Dai, Q. Yang, G.-R. Xue, and Y. Yu, “Boosting for transfer learning,” in Proc. 24th Int. Conf. Mach. Learn. (ICML), Corvalis, OR, USA, Jun. 2007, pp. 193–200, doi: 10.1145/1273496.1273521.

[37] D. Pardoe and P. Stone, “Boosting for regression transfer,” in Proc. ICML, Jun. 2010, pp. 863–870.

JIEJUN XIE was born in Hunan, China, in 1994. He received the B.S. degree in mechanical engineering from the College of Mechanical Science and Engineering, Huazhong University of Science and Technology, Hubei, China, in 2016. He is currently pursuing the Ph.D. degree with the National NC System Engineering Research Center, Huazhong University of Science and Technology. His research interests include CNC machining process modeling, simulation and optimization technology, intelligent numerical control technology, virtual process systems, and application of machine learning technology in NC machining.

PHILIPPE HU received the B.S. and M.S. degrees from Tianjin University, Tianjin, China, in 2008 and 2010, respectively, and the Ph.D. degree from the Hong Kong University of Science and Technology, Hong Kong, China, in 2013. He is currently an Associate Professor with the National NC System Engineering Research Center, Huazhong University of Science and Technology. His research interests include multi-axis NC machining tool path generation and optimization, multi-axis adaptive machining, multi-axis measurement trajectory planning, and industrial robots.

SHIBO GAO was born in Hubei, China, in 1995. He received the B.S. and M.S. degrees in mechanical engineering from the College of Mechanical Science and Engineering, Huazhong University of Science and Technology, Hubei, in 2016 and 2021, respectively. His research interests include CNC machining process modeling, simulation and optimization technology, and application of machine learning technology in NC machining.

JIHONG CHEN received the Ph.D. degree in mechanical engineering from the Huazhong University of Science and Technology, Hubei, China, in 1992. He is currently a Professor with the National NC System Engineering Research Center, Huazhong University of Science and Technology. His research interests include intelligent and networked CNC technology, advanced control theory, intelligent automation systems, and robot technology.

HUICHENG ZHOU received the Ph.D. degree in mechanical engineering from the Huazhong University of Science and Technology, Hubei, China, in 2001. He is currently a Professor with the National NC System Engineering Research Center, Huazhong University of Science and Technology. His research interests include complex systems modeling, control, and optimization, advanced control theory, intelligent automation systems, and numerical control technique.

JIANZHONG YANG received the Ph.D. degree in mechanical engineering from the Huazhong University of Science and Technology, Hubei, China, in 2006. He is currently a Professor with the National NC System Engineering Research Center, Huazhong University of Science and Technology. His research interests include intelligent technology of CNC systems, advanced control theory, intelligent automation systems, and robot technology.

***