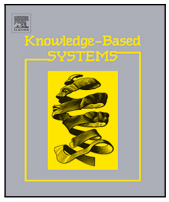
Knowledge-Based Systems 299 (2024) 112113

Contents lists available at ScienceDirect

Knowledge-Based Systems

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



A novel meta-transfer learning approach via convolutional multi-head self-attention network for few-shot fault diagnosis

Lanjun Wan [a,*], Le Huang [a], Jiaen Ning [a], Changyun Li [a], Keqin Li [b]

[a] School of Computer Science, Hunan University of Technology, Zhuzhou 412007, China
[b] Department of Computer Science, State University of New York, New Paltz, NY, 12561, USA

ARTICLE INFO

Keywords:
Fault diagnosis
Few-shot
Meta-transfer learning
Multi-head self-attention mechanism

ABSTRACT

In practical industrial applications, it is crucial to train a robust fault diagnosis (FD) model that can quickly adapt to new working conditions or fault modes using a few labeled fault samples. Therefore, a novel convolutional multi-head self-attention network-based meta-transfer learning approach (CMS-MTL) for few-shot fault diagnosis (FSFD) is proposed. Firstly, a convolutional multi-head self-attention network (CMHSAN) is designed, which ingeniously combines the multi-head self-attention (MHSA) blocks and convolution blocks. The local and global feature information of the input time–frequency images are fully considered through the mutual cooperation of MHSA and convolution, so as to fully extract the discriminative features among various fault classes. Secondly, a three-stage CMHSAN-based meta-transfer learning (MTL) scheme is proposed, which provides a good initialization state for the meta-training of the CMHSAN model through the pre-training stage, updates the pre-trained model with the scaling and shifting parameters in the meta-training stage, and fine-tunes the updated model in the meta-testing stage, so as to quickly adapt to new FSFD tasks from the target domain. Thirdly, aiming at the fault classes that are difficult to be diagnosed during meta-training, a meta-task re-training (MTRT) strategy is designed to learn more valuable transferable knowledge in the meta-training stage, thereby improving the adaptability of the CMHSAN model to hard FSFD tasks. Finally, extensive experiments are conducted under different FSFD scenarios to verify the effectiveness of the proposed approach. The results prove that the approach can quickly adapt to new FSFD tasks through the learned meta-knowledge and achieve high diagnosis accuracies.

1. Introduction

Rolling bearings and gearboxes are the most common components of rotating machinery. Their health conditions directly impact the performance and safe operation of the machines [1]. Therefore, it is of great significance to monitor the running states of rotating machinery and perform fault diagnosis. In past years, the deep learning-based fault diagnosis methods [2] have made significant progress. For example, Ruan et al. [3] designed a convolutional neural network for rotating machinery fault diagnosis (RMFD), where the input sample length and convolution kernel size are determined by the physics-guided rules, which can achieve higher accuracy. Hou et al. [4] proposed a Transformer-based fault diagnosis model to significantly improve the diagnosis accuracy of bearings. Tong et al. [5] effectively realized RMFD under noisy working conditions through the improved deep residual shrinkage network. Han et al. [6] studied a semi-supervised RMFD method using adversarial learning, which achieves superior fault diagnosis performance under limited labeled training samples. Chen et al. [7] successfully realized the fault detection of gearboxes by using the physics knowledge on the fault signature to determine the hyper-parameters of the long-short term memory neural network. Yao et al. [8] proposed a Bayesian deep learning-based intelligent fault diagnosis approach, which not only has higher diagnosis accuracy but also effectively increases the trustworthiness and reliability of diagnosis results. The above research has conducted in-depth and successful exploration of deep learning-based fault diagnosis from different aspects, greatly promoting the development of the fault diagnosis field.

The deep learning-based RMFD methods can automatically extract fault features from the vibration data of rotating machinery and identify different fault modes to achieve efficient and accurate FD, but most of them usually need numerous labeled fault samples and expensive computing resources as support. In industrial production, the cost of acquiring numerous labeled fault samples is extremely high due to the complicated working conditions. Especially for new fault types, the available fault data are extremely limited. In the case of a few

* Corresponding author.
E-mail address: wanlanjun@hut.edu.cn (L. Wan).

https://doi.org/10.1016/j.knosys.2024.112113
Received 20 January 2024; Received in revised form 29 April 2024; Accepted 9 June 2024
Available online 12 June 2024

labeled fault samples, the RMFD models based on deep learning are difficult to learn and generalize to new faults. In recent years, many researchers have adopted data augmentation, transfer learning, meta-learning, and MTL to cope with the problem of few-shot in FD. Data augmentation technology usually uses oversampling methods and generative adversarial network (GAN) to increase the diversity and scale of training data [9]. The oversampling methods enlarge minority class samples by generating synthetic samples. Wei et al. [10] combined the majority weighted minority oversampling method and the cluster algorithm to deal with the within-class imbalances that exist in bearing datasets under complex working conditions. Li et al. [11] studied an improved synthetic minority oversampling method, where the natural neighbors are introduced to generate high-quality synthetic samples. These oversampling methods lack consideration of the distribution of the original data and are prone to produce noise samples. GAN can be adopted for generating samples with a similar distribution to the original data. Zhou et al. [12] designed a GAN with global optimization, where a generator is adopted for generating fault features and a discriminator is adopted for filtering low-quality fault samples. Liu et al. [13] utilized an improved GAN to address the issue of imbalanced fault classes in the bearing dataset. The sample space can be effectively expanded through GAN. However, when the original training data are seriously insufficient, the high-quality fault sample generation ability of GAN will be limited to a certain extent, which may lead to mode collapse and instability during training.

In order to reduce the demand for model training on the labeled fault samples from the target domain, the transfer learning performs the FD tasks through the knowledge learned from the source domains [14]. Li et al. [15] offered an improved domain adaptation approach to fully extract domain invariant features, thereby effectively realizing RMFD under cross-working condition scenarios. Tan et al. [16] designed a joint distributed adaptation network, which can minimize the distribution discrepancies of features learned from different domains. Li et al. [17] exploited a weighted adversarial transfer network to effectively realize cross-domain knowledge transfer. Wan et al. [18] extracted domain invariant features more effectively by reducing joint distribution discrepancies. Chen et al. [19] adopted an effective transfer learning method to perform cross-domain FD, where the parameters of the pre-trained model are used as the initial weights of the target domain FD model, and the limited training data from the target domain are used for fine-tuning the model. Transfer learning can apply the knowledge learned from one source task to solve another related but different target task, whereas less attention is paid to how to effectively select and adjust the transfer strategies to adapt to the changing transfer learning scenarios.

Meta-learning can learn a general strategy that can transfer knowledge between different tasks through learning from multiple FSFD tasks, which enables the model to quickly adapt to the changes and challenges in the face of new FSFD tasks. Therefore, recently some researchers have begun to apply meta-learning to FSFD [20]. Zhang et al. [21] studied a model-agnostic meta-learning (MAML)-based FSFD approach that can well adapt to unknown working conditions, which can effectively enhance the performance of FSFD. Yang et al. [22] put forward an improved MAML method, which enhances the generalization of the FSFD model. Feng et al. [23] explored a meta-learning method that employs the unlabeled samples to optimize the original prototypes generated by using the labeled samples from the support set, which can more accurately reflect the class features. Ma et al. [24] used distance measurement to help the meta-learning model obtain the similarity between samples, which effectively improves the ability of FSFD. Lin et al. [25] presented a generalized MAML method that can increase the performance of heterogeneous signal-driven FSFD. Meta-learning can enhance the ability of the FD model to quickly adapt to new target tasks by exploiting the meta-knowledge learned from several few-shot tasks, whereas the knowledge transfer between different domains is less concerned.

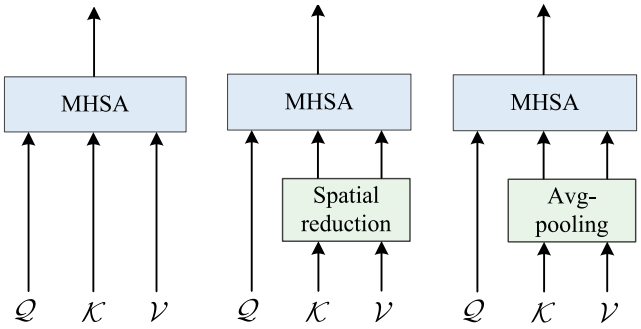
MTL [26] can effectively exploit the meta-knowledge learned from multiple meta-tasks to promote knowledge transfer between different domains, thereby significantly enhancing the adaptability of the model on new tasks and effectively alleviating the limitations of traditional transfer learning and meta-learning methods in dealing with the problem of few-shot classification. Recently, MTL contributes another idea for FSFD. Li et al. [27] presented a MTL approach for RMFD, which effectively realizes fine-grained FSFD through cross-domain knowledge transfer. Ma et al. [28] studied a digital twin-assisted MTL approach, where the distribution discrepancies between simulated and real data are effectively reduced through domain adaptation. Lei et al. [29] developed a prior knowledge-embedded MTL framework, where the prior knowledge is embedded into the meta-learning based on metric, which enhances the model generalization in different FSFD tasks. The existing research shows that MTL has a promising potential in FSFD. However, the collected rotating machinery fault data are complex and limited in actual industrial scenarios, and it is usually hard to extract the discriminative features among different fault classes, which poses a challenge to the design of the pre-training model in MTL. Moreover, when there are large domain discrepancies between the new FSFD tasks and the meta-training tasks, the model used in MTL may not learn enough knowledge for accurate FD. Therefore, a novel CMHSAN-based MTL approach for FSFD is explored.

Compared with the FSFD methods described above, the proposed CMS-MTL approach holds the following advantages. Firstly, the CMHSAN designed in the proposed CMS-MTL approach meticulously combines the MHSA blocks and convolution blocks, which not only can better capture the fault features with higher correlation with fault classes, but also can better enhance the extraction and fusion of the local and global fault features. Secondly, the three-stage CMHSAN-based MTL scheme adopted in the proposed CMS-MTL approach combines the advantages of transfer learning and meta-learning, where the pre-trained CMHSAN model is updated through scaling and shifting parameters and the updated model is fine-tuned, which can train a robust CMHSAN model that can rapidly adapt to the new FSFD tasks. Finally, the proposed CMS-MTL approach provides a meta-task re-training strategy, which can help the CMHSAN model learn more generalized and transferable fault diagnosis knowledge to better adapt to various hard FSFD tasks.

The main contributions of this study are as follows.

(1) A convolutional multi-head self-attention network which ingeniously combines the MHSA blocks and convolution blocks is designed. The local and global feature information of the input time–frequency images are fully considered through the mutual cooperation of MHSA and convolution, which can fully extract the discriminative features among various fault classes.
(2) A three-stage CMHSAN-based MTL scheme is proposed, which provides a good initialization state for the meta-training of the CMHSAN model through the pre-training stage, updates the pre-trained model with scaling and shifting parameters in the meta-training stage, and fine-tunes the model in the meta-testing stage, which can quickly adapt to new FSFD tasks from the target domain.
(3) A meta-task re-training strategy is designed to learn more valuable transferable knowledge for the fault classes that are difficult to be diagnosed in the meta-training stage, thereby improving the adaptability of the CMHSAN model to hard FSFD tasks.

The remainder of this study is organized as follows. Section 2 introduces the basic theory. Section 3 describes the proposed CMS-MTL approach. Section 4 presents the experimental results and analysis. Finally, Section 5 provides the conclusions.



(a) Original MHSA (b) Space-reduction attention (c) Linear space-reduction attention

Fig. 1. Three different MHSA mechanisms.

2. Basic theory

2.1. MHSA mechanism

The MHSA mechanism [30] is a kind of attention mechanism in the Transformer model, which helps the model focus on the information of input sequences in different representation subspaces at the same time, so as to capture the important features and long-distance dependencies in the sequences. In the process of self-attention calculation, firstly, the input sequence X is linearly transformed to obtain the query matrix $\mathcal{Q}$, key matrix $\mathcal{K}$, and value matrix $\mathcal{V}$, which can be described as

$$\mathcal{Q} = XW^{\mathcal{Q}}, \tag{1}$$

$$\mathcal{K} = XW^{\mathcal{K}}, \tag{2}$$

and

$$\mathcal{V} = XW^{\mathcal{V}}, \tag{3}$$

where $W^{\mathcal{Q}}$, $W^{\mathcal{K}}$, and $W^{\mathcal{V}}$ are the linear transformation parameter matrices. Secondly, the self-attention weight matrix A can be obtained by calculating the similarity through the dot-product operation between $\mathcal{Q}$ and $\mathcal{K}$ and normalizing the similarity through the softmax function as follows:

$$A = \text{Softmax}\left(\frac{\mathcal{Q}\mathcal{K}^T}{\sqrt{d_k}}\right), \tag{4}$$

where d_k is the dimensionality of $\mathcal{K}$. Finally, the output representation of self-attention can be obtained by

$$\text{Attention}\left(\mathcal{Q},\ \mathcal{K},\ \mathcal{V}\right) = A\mathcal{V}. \tag{5}$$

The output of MHSA is the concatenation of h self-attention outputs, which is defined as

$$\text{MHSA}\left(X\right) = \text{Concat}\left(SA_1,\ SA_2,\ \ldots,\ SA_h\right)W^O, \tag{6}$$

where

$$SA_i = \text{Attention}\left(XW_i^{\mathcal{Q}},\ XW_i^{\mathcal{K}},\ XW_i^{\mathcal{V}}\right). \tag{7}$$

In Eq. (7), $W_i^{\mathcal{Q}}$, $W_i^{\mathcal{K}}$, and $W_i^{\mathcal{V}}$ represent three different linear transformation matrices of the ith self-attention head respectively, and W^O is a parameter matrix.

Fig. 1 shows three different MHSA mechanisms. Fig. 1(a) shows the original MHSA mechanism [30], which has high computational complexity. Fig. 1(b) shows the space-reduction attention mechanism [31], which reduces the computational complexity of the self-attention layer by reducing the spatial scales of $\mathcal{K}$ and $\mathcal{V}$ before performing the self-attention operation. However, the computational complexity of the space-reduction attention layer is still high when processing high-resolution images. Fig. 1(c) shows the linear space-reduction attention mechanism [31], which uses the avg-pooling operation to reduce the spatial dimension of the input of the self-attention layer, making the space-reduction attention layer have linear complexity and further reducing the computational cost of the self-attention layer. The linear space-reduction attention can be expressed as

$$\text{LSRA}\left(X\right) = \text{Attention}\left(XW^{\mathcal{Q}}, P_s\left(XW^{\mathcal{K}}\right), P_s\left(XW^{\mathcal{V}}\right)\right), \tag{8}$$

where P_s is the avg-pooling operation with the stride of s.

2.2. Meta-learning

The goal of meta-learning [20] is to enable the model to learn general meta-knowledge from multiple related tasks to achieve rapid adaptation to new tasks. The four main concepts in meta-learning are meta-knowledge, meta-task, support set, and query set. Meta-knowledge is a general strategy obtained by the model in the learning process of several tasks, which is employed to guide the model in adapting to new tasks. Meta-task T is often called an "N-class K-shot" task, and each meta-task is composed of a support set T^s and a query set T^q, where N denotes the number of classes and K denotes the number of samples per class in T^s. The support set $T^s = \left\{\left(x_i^s, y_i^s\right)\right\}_{i=1}^{N\times K}$ is used for helping the model adapt to the specific task, where x_i^s and y_i^s represent the ith training sample and corresponding class label, respectively. The query set $T^q = \left\{\left(x_i^q, y_i^q\right)\right\}_{i=1}^{N\times Q}$ is adopted for testing the performance of the model on a specific task, where Q denotes the number of samples per class in T^q.

Meta-learning acquires meta-knowledge through the inner-level learning and outer-level learning. MAML is a common optimization-based meta-learning method, which aims to find the good initialization of model parameters suitable for all FSFD tasks. Specifically, in the inner-level learning stage of MAML, the parameterized model of a specific task is obtained by gradient updating on T^s. In the outer-level learning stage of MAML, the global shared meta-learner is obtained by optimizing the parameters of the meta-learner on T^q of multiple meta-tasks. This bi-level learning mechanism enables the model can quickly converge on new FSFD tasks through fine-tuning with a few samples, thus performing well in FSFD.

3. Proposed method

3.1. Overall process of FSFD

The overall process of FSFD using the proposed CMS-MTL method is depicted in Fig. 2, including the following three main steps.

Step 1: Data acquisition. The rotating machinery vibration signals are collected by the accelerometers. According to different FSFD tasks, the collected rotating machinery vibration signals are split into the source and target domains.

Step 2: Data preprocessing. To better analyze the fault features in vibration signals, the short-time Fourier transform (STFT) is adopted to transform the rotating machinery vibration signals from the source and target domains into two-dimensional time–frequency images. Fig. 3 shows the examples of the raw vibration signals and corresponding time–frequency images of the inner-race faults of the drive-end bearing under different rotating speeds in the Case Western Reserve University (CWRU) dataset [32]. As seen in Fig. 3, the waveforms of the raw vibration signals have certain similarity under different rotating speeds, whereas the time–frequency images generated by STFT demonstrate the differences of fault features under different rotating speeds, which is helpful for the model to extract more discriminative fault features. Therefore, the time–frequency images generated by STFT are used as the input of CMHSAN to provide more accurate information for FSFD using the CMHSAN model.

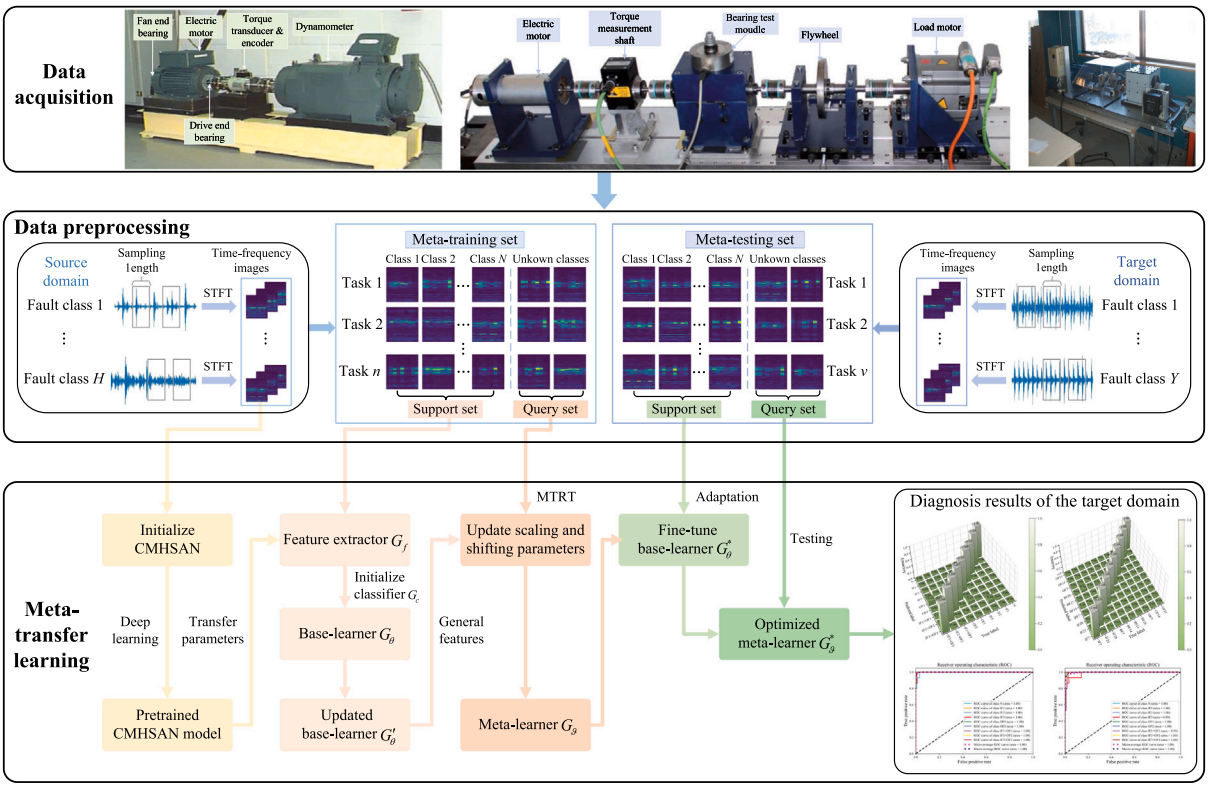


Fig. 2. Overall process of FSFD using the proposed CMS-MTL method.

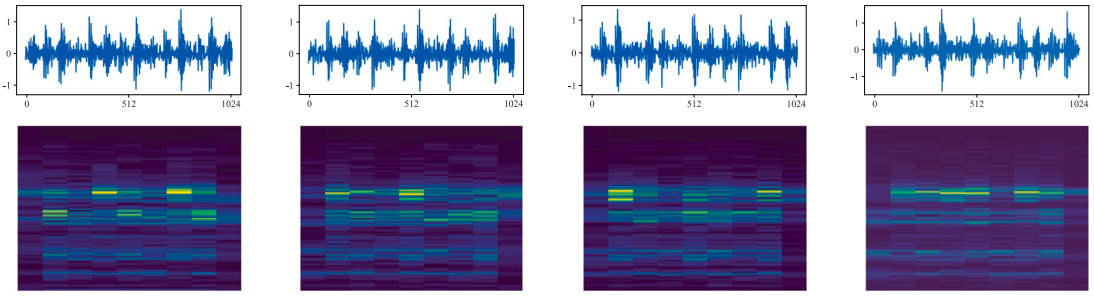


Fig. 3. Examples of the raw vibration signals and corresponding time–frequency images of the inner-race faults of the drive-end bearing under different rotating speeds in CWRU dataset.

Step 3: Meta-transfer learning. The process of MTL based on CMHSAN includes three stages: pre-training, meta-training, and meta-testing. Firstly, during the pre-training stage, the CMHSAN model is pre-trained by using the time–frequency images from the source domain to learn the general fault feature representations. Secondly, during the meta-training stage, the pre-trained CMHSAN model is updated by using the scaling and shifting parameters for each FSFD task, and the meta-task re-training strategy is used to re-train the fault classes that are difficult to identify in each meta-task. Finally, during the meta-testing stage, the trained CMHSAN model is fine-tuned through the time–frequency images in the support sets from the target domain, and the fine-tuned model is tested on the query sets from the target domain to validate the adaptability of the model on the new FSFD tasks.

3.2. Design of CMHSAN

The traditional convolutional networks have the problems of the low computing efficiency and limited global feature representation capabilities. In contrast, Transformer [30] has a strong global feature capture capability due to the design of its MHSA mechanism, but has some deficiencies in local perception capability. To overcome these limitations, the convolution blocks are combined with the MHSA blocks in Transformer. Fig. 4 depicts the overall design of the proposed CMHSAN, which is mainly composed of residual (Res) blocks and Transformer (Trans) blocks. Its unique modular hierarchical structure can fully extract local and global features of time–frequency images, so as to realize efficient and accurate FSFD.

The CMHSAN model uses four 3 × 3 convolution (Conv) layers to extract shallow general features. Firstly, the input time–frequency images pass through a 3 × 3 Conv layer with an output channel of 64 and a stride of 2, which can reduce the size of the feature maps and the calculation amount of the model. Secondly, two 3 × 3 Conv layers with stride 1 are adopted to help the model better extract local features. Finally, a 3 × 3 Conv layer with stride 2 is adopted to further reduce the size of the feature maps. In addition, the ReLU activation function is introduced for improving the nonlinear fitting ability of the model and make it better adapt to the FSFD tasks.

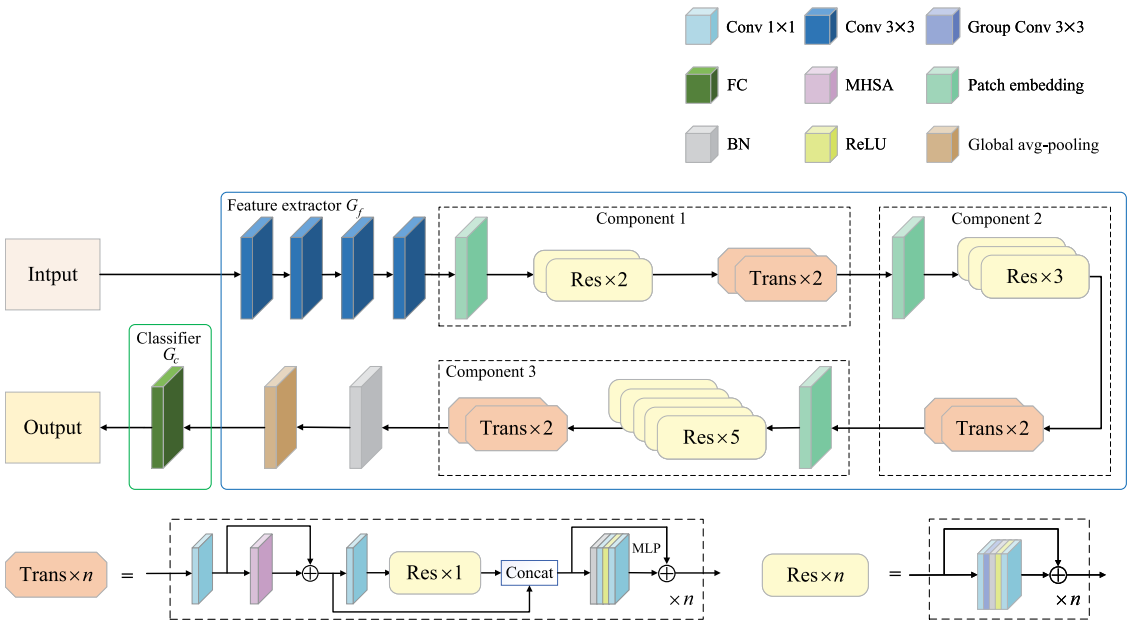


Fig. 4. Overall design of the proposed CMHSAN.

Each sample from the rotating machinery vibration signals is transformed into a two-dimensional time–frequency image, which is input into CMHSAN. The output z_{l+1} after passing through the lth Conv layer is

$$z_{l+1} = \mathrm{ReLU}\left(\sum_{k=1}^{C} w_k * z_l + b_l\right), \tag{9}$$

where z_l represents the input of the lth Conv layer, $*$ represents the convolution operation, C denotes the number of Conv kernels in the lth Conv layer, w_k indicates the weight of the kth Conv kernel in the lth Conv layer, and b_l denotes the bias of the lth Conv layer.

After the four Conv layers, three components are followed. Each component includes a patch embedding layer, multiple Res blocks, and Trans blocks. The patch embedding layer is used to divide the feature maps outputted by its previous Conv layer into patches to reduce the feature dimension and further extract the local features. The split-transform-merge strategy of the Inception module in GoogLeNet [33] is adopted in the Res blocks. Each Res block includes two 1×1 Conv layers, a 3×3 group Conv layer, a batch normalization (BN) layer, and the ReLU activation function, where the BN layer is employed to speed up the convergence of the CMHSAN model. The nonlinear mapping $F : \rho \rightarrow F(\rho)$ is constructed through a Res block. The output ρ_{r+1} after the rth Res block is

$$\rho_{r+1} = \rho_r + F\left(\rho_r\right), \tag{10}$$

where ρ_r represents the input of the rth Res block and $1 \leq r \leq R$. R denotes the number of Res blocks in each component, and R in the three components are 2, 3, and 5, respectively. Although the Res blocks can fully extract the local features of the time–frequency images, the extraction of global features of the time–frequency images is equally very important in FSFD. To comprehensively consider the local and global features, two Trans blocks are introduced after the Res blocks in each component to capture the global features. In each Trans block, firstly, the output of its previous Res block enters the 1×1 Conv layer, and the input channel dimensions are reduced to reduce the amount of calculation of the MHSA layer in the Trans block, thereby accelerating the training and reasoning process of the CMHSAN model. The output $\mathfrak{p}$ after the output ρ of the Res block passes through a 1×1 Conv layer is

$$\mathfrak{p} = \mathrm{Proj}\left(\rho\right), \tag{11}$$

where $\mathrm{Proj}(\cdot)$ denotes the 1×1 convolution operation. Secondly, the output $\mathfrak{p}$ enters the MHSA layer. Inspired by the linear space-reduction attention shown in Fig. 1(c), the input $\mathfrak{p}$ of the MHSA layer is avg-pooled and its spatial dimension is reduced according to Eq. (8) before calculating MHSA. The output z_{MHSA} of the MHSA layer is denoted as

$$z_{\mathrm{MHSA}} = \mathrm{Concat}\left(\mathrm{LSRA}_1\left(\mathfrak{p}\right), \mathrm{LSRA}_2\left(\mathfrak{p}\right), \ \ldots, \right.$$
$$\left.\mathrm{LSRA}_h\left(\mathfrak{p}\right)\right) W^O + \mathfrak{p}, \tag{12}$$

where h denotes the number of self-attention heads in the MHSA layer and h is set to 32 in the CMHSAN model. To capture the fine-grained fault features of rotating machinery, a Res block is introduced into each Trans block to cooperate with MHSA, making the CMHSAN model more expressive, thereby enhancing the FSFD ability of the CMHSAN model. Before entering the Res block, the output $\mathscr{H}$ after reducing the channel dimensions of z_{MHSA} through the 1×1 convolution operation is

$$\mathscr{H} = \mathrm{Proj}\left(z_{\mathrm{MHSA}}\right). \tag{13}$$

The output $\mathscr{P}$ after $\mathscr{H}$ is passed through the Res block within the Trans block is

$$\mathscr{P} = \mathscr{H} + F\left(\mathscr{H}\right). \tag{14}$$

Thirdly, the output of the MHSA layer is concatenated with the output of the Res block to mix the high- and low-frequency fault feature information captured by the model, which is represented as

$$\mathscr{M} = \mathrm{Concat}\left(z_{\mathrm{MHSA}}, \mathscr{P}\right). \tag{15}$$

Finally, the multi-layer perceptron (MLP) layer is adopted for enhancing the extraction of the discriminative features of the time–frequency images of different fault classes. The MLP layer includes a BN layer, two 1×1 Conv layers, and the ReLU activation function. The output $\mathscr{Z}$ of the MLP layer is

$$\mathscr{Z} = \mathrm{MLP}\left(\mathscr{M}\right) + \mathscr{M}. \tag{16}$$

Through the above series of operations, the features of different granularity of the time–frequency images can be fully extracted and the final feature maps can be obtained. The global avg-pooling is performed on the feature maps to obtain one-dimensional feature vectors, which are mapped to the label space of fault classes through the fully connected (FC) layer to achieve fault classification.

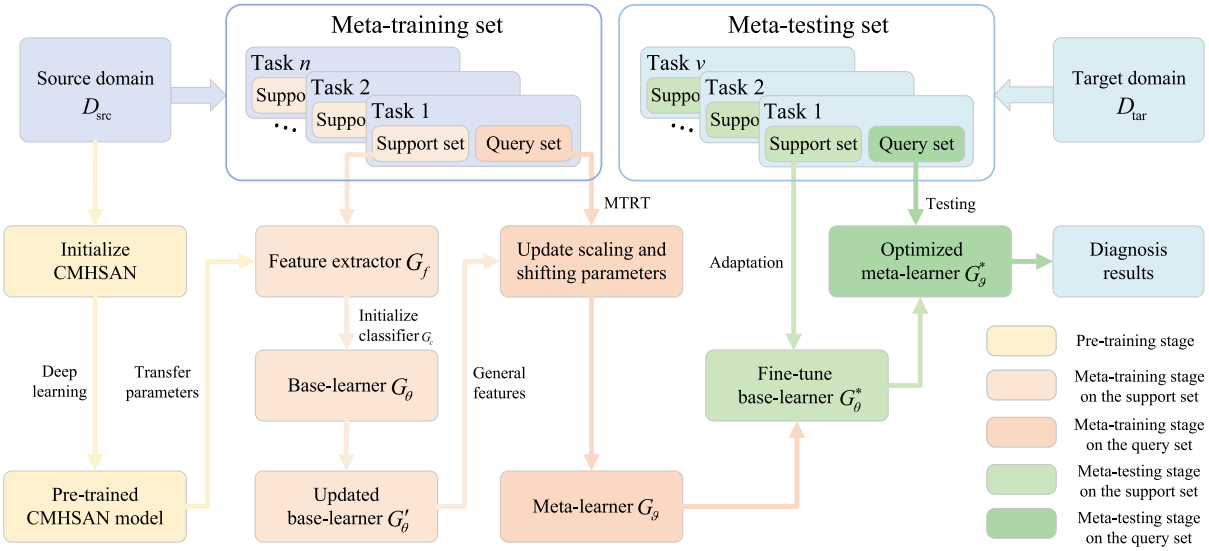


Fig. 5. Process of MTL based on CMHSAN.

3.3. Meta-transfer learning based on CMHSAN

Fig. 5 depicts the process of MTL based on CMHSAN. In Fig. 5, different colors represent different stages of MTL, that is, yellow, orange, and green represent the pre-training, meta-training, and meta-testing stages of MTL, respectively.

3.3.1. Pre-training stage

In the pre-training stage, firstly, the feature extractor G_f and classifier G_c in the CMHSAN model shown in Fig. 4 are initialized randomly, where G_f is composed of the rest of the CMHSAN model except for the FC layer, and G_c is composed of the FC layer. Secondly, the pre-training of the CMHSAN model is performed on the source domain D_{src}, and the pre-training process is described in Algorithm 1. In the pre-training process of the CMHSAN model, stochastic gradient descent is utilized for updating G_f and G_c, which can be expressed as

$$\left[G_f; G_c\right] = \left[G_f; G_c\right] - \alpha \nabla L_{\text{CE}}\left(\left[G_f; G_c\right]\right), \tag{17}$$

where α is the learning rate of the pre-training stage, which is used to control the step size of the parameter update. $L_{\text{CE}}\left(\left[G_f; G_c\right]\right)$ is the cross-entropy (CE) loss on the source domain D_{src}, which can be calculated by

$$L_{\text{CE}}\left(\left[G_f; G_c\right]\right) = \frac{1}{|D_{\text{src}}|} \sum_{(x_i,\, y_i) \in D_{\text{src}}} \mathcal{J}\left(\mathcal{F}_{[G_f;\, G_c]}\left(x_i\right), y_i\right), \tag{18}$$

where $\mathcal{F}_{[G_f;\, G_c]}\left(x_i\right)$ and y_i denote the predicted label and true label of sample x_i respectively, $|D_{\text{src}}|$ denotes the number of samples from D_{src}, and $\mathcal{J}(\cdot)$ represents the CE loss function with label smoothing used to solve the issue of overfitting to the source domain. $\mathcal{J}\left(\mathcal{F}_{[G_f;\, G_c]}\left(x_i\right), y_i\right)$ is used to measure the difference between the predicted and true labels of sample x_i, which can be calculated as

$$\mathcal{J}\left(\mathcal{F}_{[G_f;G_c]}\left(x_i\right), y_i\right) = -\log\left((1-\epsilon)\frac{\exp\left(\mathfrak{z}_{y_i}\right)}{\sum_{j=1}^{\mathfrak{N}} \exp\left(\mathfrak{z}_{j|x_i}\right)} + \frac{\epsilon}{\mathfrak{N}}\right), \tag{19}$$

where ϵ is the label smoothing factor and $0 < \epsilon < 1$, $\mathfrak{N}$ is the number of fault classes, and $\mathfrak{z}_{y_i}$ and $\mathfrak{z}_{j|x_i}$ are the unnormalized log-probabilities of the CMHSAN model for sample x_i belonging to the true label y_i and the jth class, respectively.

Algorithm 1 The pre-training process of the proposed CMS-MTL method

Input: The source domain D_{src}, the learning rate α, the batch size B, the number of iterations I, and the number of epochs E.
Output: The pre-trained G_f and G_c.

```
Randomly initialize G_f and G_c;
for i = 1 to E do
  for j = 1 to I do
    Randomly choose a batch of samples {x_k, y_k}_{k=1}^B from D_src;
    Calculate L_CE([G_f; G_c]) by Eq. (18);
    Update G_f and G_c by Eq. (17);
  end for
end for
```

3.3.2. Meta-training stage

Due to the classification objectives of each meta-task in the meta-training stage are different from those in the pre-training stage, a new CMHSAN model is initialized randomly for each meta-task in the meta-training stage, and the network structure of the model is exactly the same as that of the CMHSAN model obtained in the pre-training stage. During the meta-training process of the CMHSAN model, firstly, a meta-training set consisting of n "N-class K-shot" meta-tasks is formed by random sampling from D_{src}. Secondly, the weight and bias parameters of G_f in the pre-trained CMHSAN model are transferred to the CMHSAN model corresponding to the first meta-task to obtain the base-learner G_θ, which provides a good initialization state for the meta-training of the CMHSAN model, so as to reduce the risk of overfitting when using a few samples for meta-training. Finally, the base-learner and meta-learner are updated on the support set T_{trn}^s and query set T_{trn}^q of each meta-task through gradient descent, respectively. Specifically, the meta-training process of the CMHSAN model on each meta-task includes the following steps.

Step 1: G_θ is iteratively updated on T_{trn}^s of the current meta-task through gradient descent, which can be described as

$$G'_\theta = G_\theta - \beta \nabla_{G_\theta} L_{T_{\text{trn}}^s}\left(G_\theta,\, S^\omega,\, S^\tau\right), \tag{20}$$

where β is the base-learning rate and G'_θ is the updated base-learner. $L_{T_{\text{trn}}^s}\left(G_\theta,\, S^\omega,\, S^\tau\right)$ denotes the CE loss on T_{trn}^s of the current meta-task, which can be calculated by

$$L_{T_{\text{trn}}^s}\left(G_\theta,\, S^\omega,\, S^\tau\right) = \frac{1}{N \times K} \sum_{(x_i,\, y_i) \in T_{\text{trn}}^s} \mathcal{J}\left(\mathcal{F}_{(G_\theta,\, S^\omega,\, S^\tau)}\left(x_i\right), y_i\right), \tag{21}$$

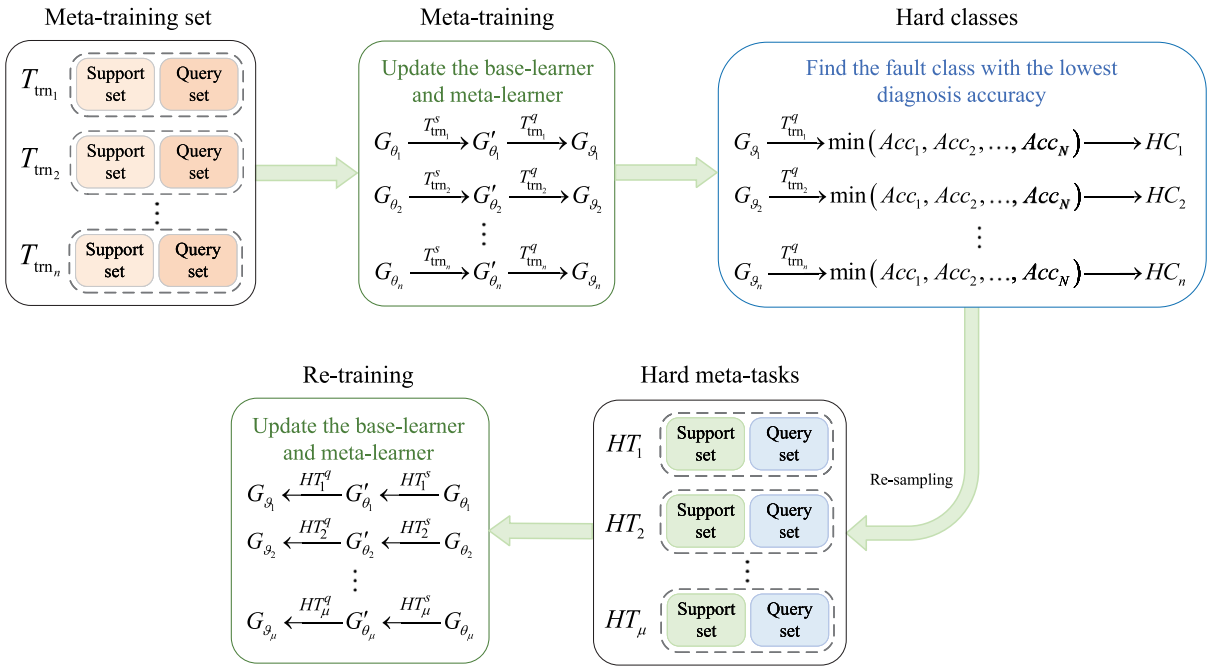


Fig. 6. Proposed meta-task re-training strategy.

where S^{ω} and S^{τ} represent the scaling and shifting parameters and are initialized to ones and zeros, respectively.

Step 2: S^{ω} and S^{τ} are iteratively updated on T^{q}_{trn} of the current meta-task through gradient descent, which can be described as

$$S^{\omega} = S^{\omega} - \gamma \nabla_{S^{\omega}} L_{T^{q}_{\mathrm{trn}}}\left(G'_{\theta},\ S^{\omega},\ S^{\tau}\right) \tag{22}$$

and

$$S^{\tau} = S^{\tau} - \gamma \nabla_{S^{\tau}} L_{T^{q}_{\mathrm{trn}}}\left(G'_{\theta},\ S^{\omega},\ S^{\tau}\right), \tag{23}$$

where γ is the meta-learning rate. $L_{T^{q}_{\mathrm{trn}}}\left(G'_{\theta},\ S^{\omega},\ S^{\tau}\right)$ denotes the CE loss on T^{q}_{trn} of the current meta-task, which can be calculated by

$$L_{T^{q}_{\mathrm{trn}}}\left(G'_{\theta},\ S^{\omega},\ S^{\tau}\right) = \frac{1}{N \times Q} \sum_{(x_i,\, y_i) \in T^{q}_{\mathrm{trn}}} \mathcal{J}\left(\mathcal{F}_{\left(G'_{\theta},\ S^{\omega},\ S^{\tau}\right)}\left(x_i\right), y_i\right). \tag{24}$$

Step 3: The updated scaling and shifting parameters are introduced to adaptively adjust the weight parameters ϕ and bias parameters ζ of G_f in the CMHSAN model corresponding to the current meta-task, enabling the CMHSAN model to rapidly adapt to the new FSFD tasks, which can be expressed as

$$G_f(\phi,\ \zeta) = \phi \odot S^{\omega} + \zeta + S^{\tau}, \tag{25}$$

where $\odot$ represents the element-wise multiplication.

Step 4: G'_{θ} is iteratively updated on T^{q}_{trn} of the current meta-task through gradient descent to obtain the meta-learner G_{ϑ}, which can be described as

$$G_{\vartheta} = G'_{\theta} - \gamma \nabla_{G'_{\theta}} L_{T^{q}_{\mathrm{trn}}}\left(G'_{\theta},\ S^{\omega},\ S^{\tau}\right). \tag{26}$$

Repeating steps 1–4 to complete the model training on each meta-task of the meta-training set in turn, and the meta-learner G_{ϑ} obtained on the current meta-task provides a good initialization state for the base-learner G_{θ} corresponding to the next meta-task. The $\mathscr{E}$ epochs of training are performed repeatedly on the meta-training set, and the G_{ϑ} obtained from the last epoch of training provides a good initialization state for the base-learner G^{*}_{θ} used in the meta-testing stage to learn new FSFD tasks.

3.3.3. Meta-testing stage

The meta-testing stage aims to evaluate the rapid adaptability of the meta-learner obtained in the meta-training stage when facing new FSFD tasks from the target domain. In the meta-testing stage, a meta-testing set consisting of υ "N-class K-shot" meta-tasks is formed by random sampling from the target domain D_{tar}, and each meta-task is composed of a labeled support set T^{s}_{tst} and an unlabeled query set T^{q}_{tst}. For each meta-task, at first the base-learner G^{*}_{θ} is fine-tuned on T^{s}_{tst} of the meta-task through gradient descent to obtain the optimized meta-learner G^{*}_{ϑ}, and then the G^{*}_{ϑ} is tested on T^{q}_{tst} of the meta-task to gain the FSFD accuracy. After obtaining the FSFD accuracies of G^{*}_{ϑ} on all meta-tasks of the meta-testing set, the average accuracy is calculated and employed for evaluating the generalization performance of the meta-learner obtained through meta-training on new FSFD tasks.

3.4. Meta-task re-training strategy

In the traditional meta-learning methods, the generalization ability is improved by learning multiple meta-tasks formed by random sampling in the meta-training stage. Due to the randomness of sampling, the fault classes contained in different meta-tasks also have randomness, which may lead to differences in the difficulty of different meta-tasks used for model training. The traditional meta-learning methods usually treat all meta-tasks equally in the meta-training stage, that is, the meta-tasks with high classification difficulty during the meta-training process are not paid enough attention, which may affect the FSFD performance to a certain extent. Therefore, a meta-task re-training strategy is proposed, as depicted in Fig. 6. The μ "N-class K-shot" meta-tasks are formed by random re-sampling from the source domain according to the fault classes that are difficult to be diagnosed, and the CMHSAN model is re-trained on these meta-tasks to help the model learn more valuable and more general transferable fault diagnosis knowledge, enabling the model to faster adapt to different FSFD tasks. The meta-training process of the CMHSAN model with the MTRT strategy is described in Algorithm 2, mainly including the following steps.

Step 1: Firstly, the base-learner G_{θ_k} is updated on the support set $T^{s}_{\mathrm{trn}_k}$ of the kth meta-task in the meta-training set to obtain G'_{θ_k}

Algorithm 2 The meta-training process of the proposed CMHSAN model with the MTRT strategy

Input: The source domain D_{src}, the meta-training set including n meta-tasks, the pre-trained G_f, the base-learning rate β, the meta-learning rate γ, the meta-batch size m, the number of epochs $\mathscr{E}$, and the empty set $\mathscr{C}$.
Output: The final meta-learner G_ϑ.

Transfer the parameters of G_f and randomly initialize the classifier G_c to obtain G_{θ_1};
for $i = 1$ **to** $\mathscr{E}$ **do**
  for $j = 1$ **to** n/m **do**
    Randomly choose a batch of meta-tasks $\left\{T_{\text{trn}_{(j-1)m+1}}, \ \cdots, \ T_{\text{trn}_{(j-1)m+m}}\right\}$ from the meta-training set;
    for T_{trn_k} **in** $\left\{T_{\text{trn}_{(j-1)m+1}}, \ \cdots, \ T_{\text{trn}_{(j-1)m+m}}\right\}$ **do**
      Initialize S_k^{ω} and S_k^{τ} by ones and zeros, respectively;
      Update G_{θ_k} to obtain G'_{θ_k} by Eq. (20) on $T^s_{\text{trn}_k}$;
      Update S_k^{ω} and S_k^{τ} by Eqs. (22) and (23) on $T^q_{\text{trn}_k}$, respectively;
      Adjust the parameters of G_f corresponding to T_{trn_k} by Eq. (25);
      Update G'_{θ_k} to obtain G_{ϑ_k} by Eq. (26) on $T^q_{\text{trn}_k}$;
      Classify the samples corresponding to N fault classes in $T^q_{\text{trn}_k}$ by G_{ϑ_k};
      Select the fault class with the lowest accuracy as HC_k and add it to the set $\mathscr{C}$;
      $G_{\theta_{k+1}} \leftarrow G_{\vartheta_k}$;
    end for
  end for
  Form μ meta-tasks by random re-sampling from D_{src} according to the hard classes in $\mathscr{C}$;
  $G_{\theta_1} \leftarrow G_{\vartheta_\mu}$;
  for $j = 1$ **to** μ/m **do**
    Randomly choose a batch of hard meta-tasks $\left\{HT_{(j-1)m+1}, \ \cdots, \ HT_{(j-1)m+m}\right\}$;
    for HT_k **in** $\left\{HT_{(j-1)m+1}, \ \cdots, \ HT_{(j-1)m+m}\right\}$ **do**
      Initialize S_k^{ω} and S_k^{τ} by ones and zeros, respectively;
      Update G_{θ_k} to obtain G'_{θ_k} by Eq. (20) on HT^s_k;
      Update S_k^{ω} and S_k^{τ} by Eqs. (22) and (23) on HT^q_k, respectively;
      Adjust the parameters of G_f corresponding to HT_k by Eq. (25);
      Update G'_{θ_k} to obtain G_{ϑ_k} by Eq. (26) on HT^q_k;
      $G_{\theta_{k+1}} \leftarrow G_{\vartheta_k}$;
    end for
  end for
  $G_{\theta_1} \leftarrow G_{\vartheta_\mu}$;
end for

according to Eq. (20). Secondly, the scaling and shifting parameters are updated on the query set $T^q_{\text{trn}_k}$ of the kth meta-task according to Eqs. (22) and (23) respectively, and the parameters of G_f corresponding to the meta-task are adjusted by Eq. (25). Finally, the base-learner G'_{θ_k} is updated on $T^q_{\text{trn}_k}$ to obtain the meta-learner G_{ϑ_k} according to Eq. (26).

Step 2: All the samples corresponding to N fault classes in the query set $T^q_{\text{trn}_k}$ are classified by the meta-learner G_{ϑ_k} corresponding to the kth meta-task to obtain the diagnosis accuracies $\left\{Acc_1, \ Acc_2, \ \cdots, \ Acc_N\right\}$. The fault class with the lowest diagnosis accuracy is called the hard class HC_k of the kth meta-task.

Step 3: Repeat steps 1 and 2 until the hard class corresponding to each meta-task in the meta-training set is obtained, and the μ new "N-class K-shot" meta-tasks are formed by random re-sampling from the source domain D_{src} according to these hard classes, where each task is called a hard meta-task.

Step 4: The CMHSAN model is re-trained on the μ hard meta-tasks.

4. Experimental results and analysis

4.1. Experimental setup

Fig. 7 presents the bearing test bench of CWRU. The detailed description of CWRU dataset is given in [32]. The bearing vibration data of the drive-end collected under the following four different working conditions at 12 kHz sampling frequency are selected: 1797, 1772, 1750, and 1730 rpm, including normal (N), inner-race fault (IF), outer-race fault (OF), and ball fault (BF) data. For CWRU dataset, a cross-working condition FSFD scenario is designed. The selected vibration data and the corresponding task assignments are shown in Table 1, where 36 kinds of vibration data are formed the source domain for pre-training and meta-training, and 12 kinds of vibration data are formed the target domain for meta-testing. The four different types of FSFD tasks of 5-class 1-shot, 5-class 5-shot, 10-class 1-shot, and 10-class 5-shot are performed under this scenario, where the 1-shot and 5-shot respectively denote that there are 1 and 5 training samples under each fault class within the support set of each FSFD task. The source and target domains contain 3960 and 1320 samples, respectively. For different types of FSFD tasks, a few samples are randomly chosen from the source and target domains to form 100 and 110 meta-tasks for meta-training and meta-testing, respectively. In addition, during the pre-training stage, all samples from the source domain are used.

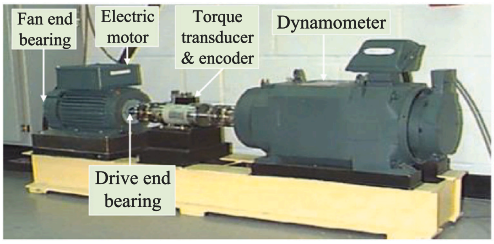


Fig. 7. Bearing test bench of CWRU [32].

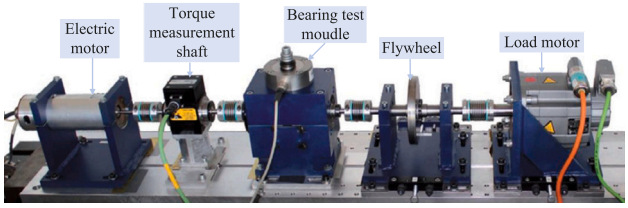


Fig. 8. Bearing test bench of PU [34].

Fig. 8 presents the bearing test bench of Paderborn University (PU). The detailed description of PU dataset is given in [34]. The bearing vibration data collected under the following four different working conditions at 64 kHz sampling frequency are selected: W_1 (0.7 N m/1500 rpm/1000 N), W_2 (0.7 N m/900 rpm/1000 N), W_3 (0.1 N m/1500 rpm/1000 N), and W_4 (0.7 N m/1500 rpm/400 N). The nine kinds of vibration data given in Table 2 are selected under each working condition. For PU dataset, the four cross-working condition FSFD scenarios are designed, as described in Table 3. For example, under scenario 1, the vibration data under W_2 and W_3 are formed the source domain for pre-training and meta-training, and the vibration data under W_1 are formed the target domain for meta-testing. The two different types of FSFD tasks of 9-class 1-shot and 9-class 5-shot are performed under these four different scenarios, respectively. The source and target domains under each scenario include 2250 and 1125 samples, respectively. For different types of FSFD tasks, a few samples are randomly chosen from the source and target domains to form 100 and 110 meta-tasks for meta-training and meta-testing, respectively. In addition, during the pre-training stage, all samples from the source domain are adopted.

Fig. 9 presents the schematic and overview of the gearbox used in PHM 2009 data challenge competition. The detailed description of PHM dataset is given in [35]. In this experiment, five different health status data of the input terminal under low load collected in 50 Hz rotating

Table 1
Selected CWRU vibration data and the corresponding task assignments.

Rotating speed (rpm)	Fault diameter (mils)	Health status	Number of fault classes	Task
1797/1772/1750	0	N	3	
1797/1772/1750	7/14/21	IF/OF/BF	27	Pre-training/Meta-training
1797/1772/1750	28	IF/BF	6	
1730	0	N	1	
1730	7/14/21	IF/OF/BF	9	Meta-testing
1730	28	IF/BF	2	

Table 2
Description of PU dataset.

Bearing code	Health status	Damage level
K004	N	–
KI21	IF1	1
KI18	IF2	2
KI16	IF3	3
KA04	OF1	1
KA16	OF2	2
KB27	IF1+OF1	1
KB23	IF2+OF2	2
KB24	IF3+OF3	3

Table 3
Four different cross-working condition scenarios of PU dataset.

Scenario	Transfer task
Scenario 1	W_2, $W_3 \rightarrow W_1$
Scenario 2	W_1, $W_4 \rightarrow W_2$
Scenario 3	W_1, $W_4 \rightarrow W_3$
Scenario 4	W_2, $W_3 \rightarrow W_4$

speed are selected and labeled as C_1, C_2, C_3, C_4, and C_5, respectively, as shown in Table 4.

In this experiment, the proposed CMS-MTL method is compared with the following six methods: transferable convolutional neural network (TCNN) [19], deep convolution multi-adversarial domain adaptation (DCMADA) [18], prototypical network (ProtoNet) [20], semi-supervised meta-learning network (SSMN) [23], model-agnostic meta-learning-based few-shot classification (MAML-FSC) [21], and an improved variant of the MAML framework (MAML++) [36]. For a fair comparison, all the comparison methods and CMS-MTL perform the same data preprocessing as follows: STFT is used to transform each sample consisting of 1024, 1024, and 2048 consecutive sampling points in the raw vibration signals of CWRU, PU, and PHM datasets into a 84×84 time–frequency image respectively, and 110, 125, and 130 time–frequency images of each fault class in CWRU, PU, and PHM datasets are obtained respectively. Moreover, the other six comparison methods all adopt the same feature extractor (i.e., backbone network) as the proposed CMS-MTL, the other components in their respective models (such as fault classifiers) remain unchanged, and the model training strategies of the other six comparison methods also remain unchanged. For TCNN, DCMADA, and the pre-training stage of CMS-MTL, the batch size B is 64, the learning rate α is initially set to 0.1 and decayed every 30 epochs by a factor of 0.2, the maximum training epoch is 100, the dropout ratio is 0.1, the label smoothing factor ϵ is 0.1, and the stochastic gradient descent optimizer is adopted. For ProtoNet, SSMN, MAML-FSC, MAML++, and the meta-training stage of CMS-MTL, each fault class in the query set of each meta-task contains 15 samples, the base-learning rate β is 0.01, the meta-learning rate γ is 0.001, the meta-batch size m is 2, the number of training epoch $\mathscr{E}$ is 100, and Adam optimizer is adopted.

All the methods are implemented using PyTorch 1.9.0 and Python 3.8, and all the experiments are performed on the NVIDIA RTX 2070 GPU.

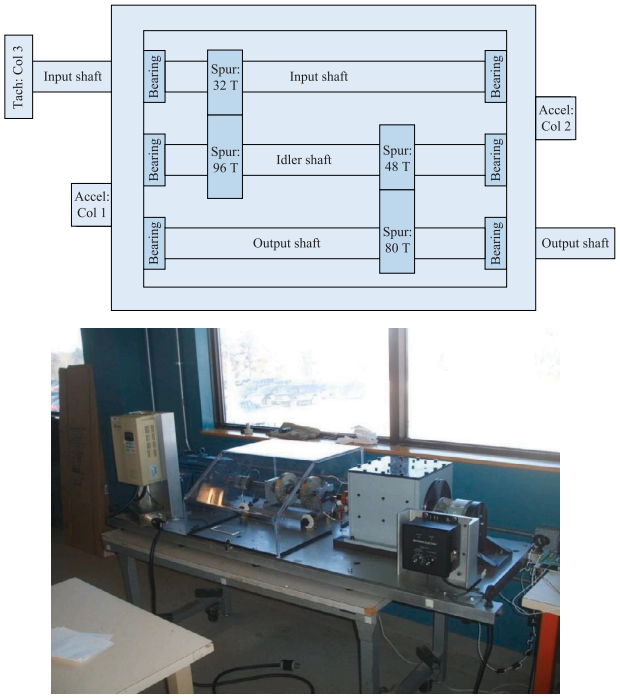


Fig. 9. Schematic and overview of the gearbox [35].

4.2. Comparison with other fault diagnosis methods

4.2.1. Cross-working condition FSFD on CWRU dataset

Table 5 gives the FSFD accuracies obtained with different methods under the cross-working condition scenario of CWRU dataset. The accuracies obtained with seven different methods on the 5-class 5-shot and 10-class 5-shot tasks are higher than those on the 5-class 1-shot and 10-class 1-shot tasks, respectively. This is because the number of training samples for each fault class has increased, which is helpful for these methods to learn more general fault knowledge. Compared with the 5-class FSFD tasks, the accuracies obtained with seven different methods on the 10-class FSFD tasks have decreased. This is because the fault diagnosis models are required to have better feature representation ability and generalization to capture the differences between different fault classes more accurately as the number of fault classes increases.

As seen in Table 5, the average diagnosis accuracies of TCNN, DCMADA, ProtoNet, SSMN, MAML-FSC, MAML++, and CMS-MTL are 91.63%, 94.25%, 95.79%, 97.61%, 96.63%, 96.91%, and 99.21%, respectively, indicating that CMS-MTL is superior to the other methods. Compared with TCNN and DCMADA, the average diagnosis accuracy of CMS-MTL is increased by 7.58% and 4.96%, respectively. This is mainly because CMS-MTL can learn more general learning strategies from different FSFD tasks through meta-learning and can more effectively share knowledge among different meta-tasks, hence the CMHSAN model can more flexibly be adjusted to rapidly adapt to new FSFD tasks. Compared with ProtoNet, SSMN, MAML-FSC, and MAML++, the average diagnosis accuracy of CMS-MTL is improved

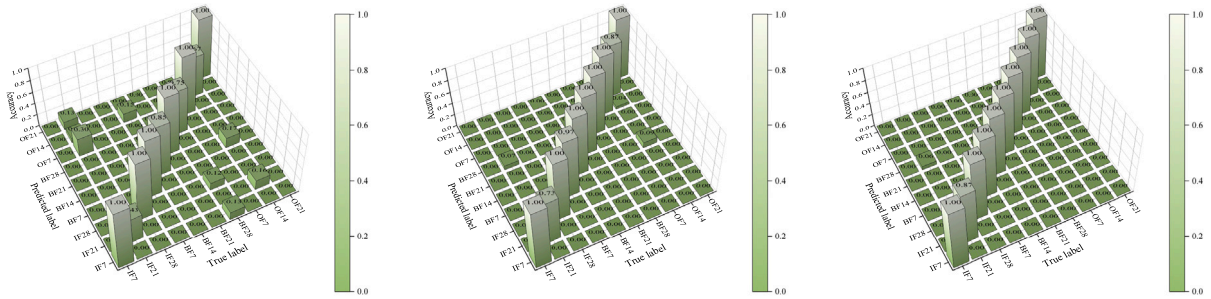
Table 4
Description of PHM dataset.

Class label	Gear				Bearing						Shaft	
	32 T	48 T	80 T	96 T	IS:IS	ID:IS	OS:IS	IS:OS	ID:OS	OS:OS	Input	Output
C_1	N	N	N	N	N	BF	OF	N	N	N	IM	N
C_2	N	N	N	N	IF	N	N	N	N	N	N	KS
C_3	N	EC	N	N	N	N	N	N	N	N	N	N
C_4	CH	EC	N	N	N	N	N	N	N	N	N	N
C_5	N	N	N	N	N	N	N	N	N	N	N	N

CH = Chipped; EC = Eccentric; KS = Keyway sheared; IM = Imbalance; IS = Input shaft; :IS = Input side; ID = Idler shaft; OS = Output shaft; :OS = Output side.

Table 5
FSFD accuracies (%) obtained with different methods under the cross-working condition scenario of CWRU dataset.

Method	1797, 1772, 1750 rpm → 1730 rpm				Average
	5-class 1-shot	5-class 5-shot	10-class 1-shot	10-class 5-shot	
TCNN [19]	93.53 ± 1.16	95.31 ± 0.70	87.00 ± 1.97	90.67 ± 1.77	91.63
DCMADA [18]	95.79 ± 0.38	96.35 ± 0.28	91.56 ± 2.51	93.30 ± 1.64	94.25
ProtoNet [20]	96.10 ± 1.95	97.28 ± 0.98	94.10 ± 1.70	95.69 ± 0.60	95.79
SSMN [23]	97.41 ± 0.48	98.86 ± 0.10	96.45 ± 0.47	97.73 ± 0.21	97.61
MAML-FSC [21]	96.45 ± 0.65	97.81 ± 0.16	95.33 ± 0.40	96.93 ± 0.15	96.63
MAML++ [36]	96.93 ± 0.55	98.26 ± 0.46	95.40 ± 0.69	97.06 ± 0.26	96.91
CMS-MTL	99.47 ± 0.33	99.85 ± 0.05	98.67 ± 0.11	98.86 ± 0.11	99.21



(a) TCNN (b) MAML-FSC (c) CMS-MTL

Fig. 10. Confusion matrices obtained with three different methods on the 10-class 1-shot FSFD tasks of CWRU dataset.

by 3.42%, 1.60%, 2.58%, and 2.30%, respectively. The main reasons are as follows. Firstly, CMS-MTL combines the advantages of meta-learning and transfer learning. CMS-MTL can learn the general fault feature representations through the pre-training of the CMHSAN model, which provides a good initialization state for the meta-training of the model, thereby enabling the model to adapt to the new FSFD tasks more quickly. Secondly, CMS-MTL introduces the scaling and shifting parameters to fine-tune the pre-trained CMHSAN model in the case of a few samples, enabling the model to better adapt to the differences between different FSFD tasks. Thirdly, CMS-MTL can focus on the fault classes that are difficult to be diagnosed during the process of meta-training through the MTRT strategy, which enables the CMHSAN model to learn more general knowledge, thereby improving the FSFD ability of the model.

Fig. 10 shows the confusion matrices obtained with three different methods on the 10-class 1-shot FSFD tasks of CWRU dataset. As depicted in Fig. 10, TCNN, MAML-FSC, and CMS-MTL accurately identify the five fault classes of IF7, IF28, BF21, OF7, and OF21, but they all have classification errors on IF21. As shown in Fig. 10(a), TCNN misclassifies 30%, 13%, 7%, and 7% of samples belonging to IF21 as OF7, OF21, IF28, and OF14, respectively. As seen in Fig. 10(b), MAML-FSC misclassifies 20% and 7% of samples belonging to IF21 as IF28 and BF28, respectively. Compared with TCNN and MAML-FSC, the diagnosis accuracy obtained with CMS-MTL on IF21 is increased by 44% and 14%, respectively, as shown in Fig. 10(c), suggesting that CMS-MTL can more effectively extract the discriminative features of each fault class with a few training samples.

To further analyze the fault classification effect of the proposed CMS-MTL, the t-distributed stochastic neighbor embedding method is adopted to visualize the diagnosis results of CMS-MTL on different FSFD tasks under the cross-working condition scenario of CWRU dataset. Fig. 11 displays the feature visualization of diagnosis results obtained with CMS-MTL on different FSFD tasks of CWRU dataset. As depicted in Figs. 11(a) and 11(b), the features of five different classes of fault samples extracted with CMS-MTL are closely clustered, which indicates that CMS-MTL can accurately distinguish different fault classes on the 5-class 1-shot and 5-class 5-shot tasks. As seen in Figs. 11(c) and 11(d), CMS-MTL has slight misclassifications on the 10-class 1-shot and 10-class 5-shot tasks, but it can still relatively accurately distinguish different fault classes on the whole. The above results further confirm that CMS-MTL can effectively extract and exploit the discriminative features in fault samples on different FSFD tasks, thereby achieving accurate fault classification.

4.2.2. Cross-working condition FSFD on PU dataset

Table 6 gives the FSFD accuracies obtained with seven methods under the four cross-working condition scenarios of PU dataset. The average accuracy obtained with CMS-MTL is 20.17%, 7.28%, 6.60%, 2.70%, 5.63%, and 3.23% higher than that obtained with TCNN, DCMADA, ProtoNet, SSMN, MAML-FSC, and MAML++ under scenarios 1 to 4, respectively. The results indicate that CMS-MTL still performs best on the more complex PU dataset, which is due to its stronger generalization ability. As shown in Table 6, the average accuracies obtained with seven different methods on different FSFD tasks under scenario 2 are

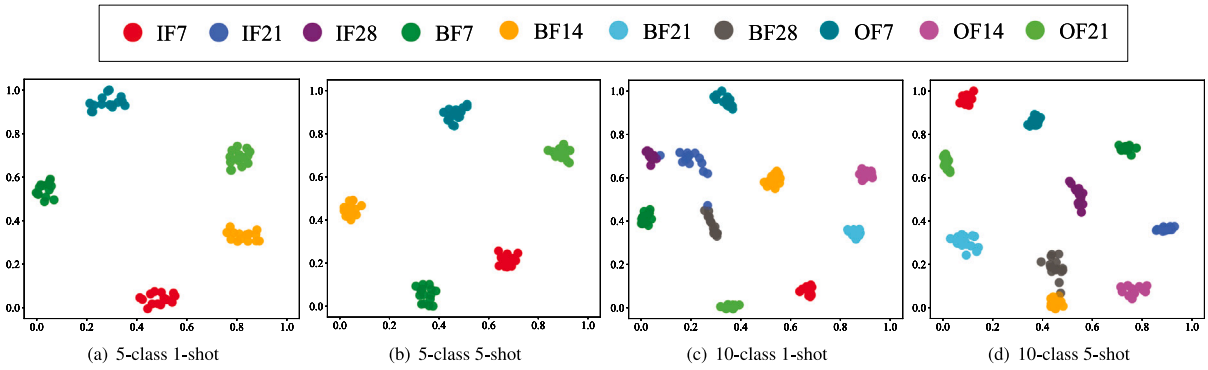


Fig. 11. Feature visualization of diagnosis results obtained with CMS-MTL on different FSFD tasks of CWRU dataset.

Table 6
FSFD accuracies (%) obtained with seven different methods under the four different cross-working condition scenarios of PU dataset.

Method	Scenario 1		Scenario 2		Scenario 3		Scenario 4		Average
	9-class 1-shot	9-class 5-shot	9-class 1-shot	9-class 5-shot	9-class 1-shot	9-class 5-shot	9-class 1-shot	9-class 5-shot	
TCNN [19]	81.67 ± 1.53	87.60 ± 0.75	63.24 ± 1.58	67.55 ± 1.13	73.50 ± 1.87	81.54 ± 1.13	71.14 ± 1.67	73.67 ± 0.87	74.99
DCMADA [18]	91.73 ± 0.76	94.10 ± 0.46	78.45 ± 1.20	84.95 ± 0.95	89.07 ± 0.64	93.39 ± 0.41	83.21 ± 1.23	88.15 ± 0.59	87.88
ProtoNet [20]	91.89 ± 1.82	95.89 ± 1.48	79.65 ± 3.04	86.64 ± 2.47	88.68 ± 3.24	93.10 ± 1.85	83.37 ± 3.59	89.29 ± 1.60	88.56
SSMN [23]	93.40 ± 0.82	96.59 ± 0.48	85.78 ± 2.04	91.06 ± 1.47	92.31 ± 1.24	95.56 ± 0.85	90.72 ± 1.59	94.22 ± 0.60	92.46
MAML-FSC [21]	92.70 ± 1.02	95.93 ± 0.64	80.08 ± 3.12	86.59 ± 2.04	89.60 ± 1.44	93.03 ± 1.03	87.22 ± 2.20	91.06 ± 1.60	89.53
MAML++ [36]	93.20 ± 0.64	96.20 ± 0.36	83.84 ± 2.99	90.59 ± 1.35	91.40 ± 0.88	95.93 ± 0.85	90.48 ± 1.98	93.83 ± 1.36	91.93
CMS-MTL	97.03 ± 0.33	98.05 ± 0.18	91.85 ± 0.62	93.70 ± 0.38	94.78 ± 0.40	97.54 ± 0.15	92.44 ± 0.55	95.89 ± 0.34	95.16

the lowest among the four different cross-working condition scenarios. This is because the distribution discrepancies between the source and target domains under scenario 2 are more significant than those under scenarios 1, 3, and 4, which leads to the FSFD performance of these methods under scenario 2 being limited to a certain extent.

Fig. 12 gives the average accuracies gained using seven different methods on different FSFD tasks under the four cross-working condition scenarios of PU dataset. As depicted in Fig. 12, the average accuracies of these seven different methods obtained on the 9-class 1-shot tasks are lower than those obtained on the 9-class 5-shot tasks. This is because there is only one training sample for each class in each 9-class 1-shot task, and it is more difficult to learn the feature representations with strong generalization from fewer training samples. However, the proposed CMS-MTL still achieves an average diagnosis accuracy of 94.03% on the 9-class 1-shot tasks, meaning that it has better FSFD ability than the other six methods.

Fig. 13 shows the confusion matrices obtained with CMS-MTL on the 9-class 1-shot tasks under the four different cross-working condition scenarios of PU dataset. As shown in Fig. 13, CMS-MTL can relatively accurately identify the samples of different fault classes under different cross-working condition scenarios, but there are also phenomena of misclassifying the inner/outer-race faults as the compound faults and misclassifying the compound faults as the inner/outer-race faults. For instance, the 12% of samples belonging to IF3+OF3 are misclassified as OF1 under scenario 1, the 15% and 8% of samples belonging to OF1 are misclassified as IF1+OF1 and IF2+OF2 under scenario 2 respectively, the 10% and 13% of samples belonging to IF3 are misclassified as IF1+OF1 and IF3+OF3 under scenario 3 respectively, and the 13% and 11% of samples belonging to OF1 are misclassified as IF1+OF1 and IF3+OF3 under scenario 4 respectively. This is because the data distributions between the compound faults and the inner-race faults are similar, and the data distributions between the compound faults and the outer-race faults are also similar, which makes it difficult to distinguish these faults, thus causing some misclassifications.

Fig. 14 presents the receiver operating characteristic (ROC) curves obtained with CMS-MTL on the 9-class 1-shot tasks under the four different cross-working condition scenarios of PU dataset. The classification performance of CMS-MTL on each fault class is evaluated

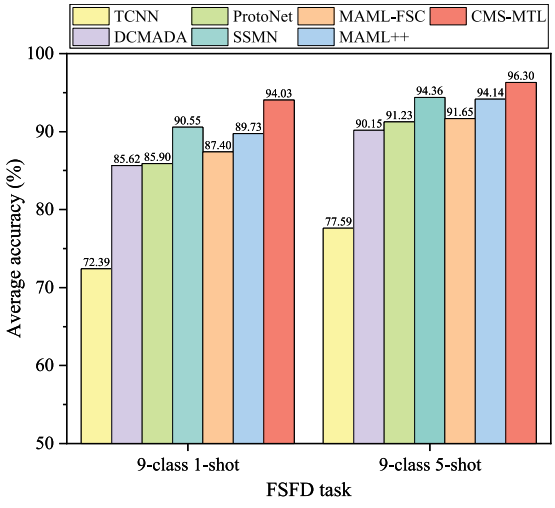


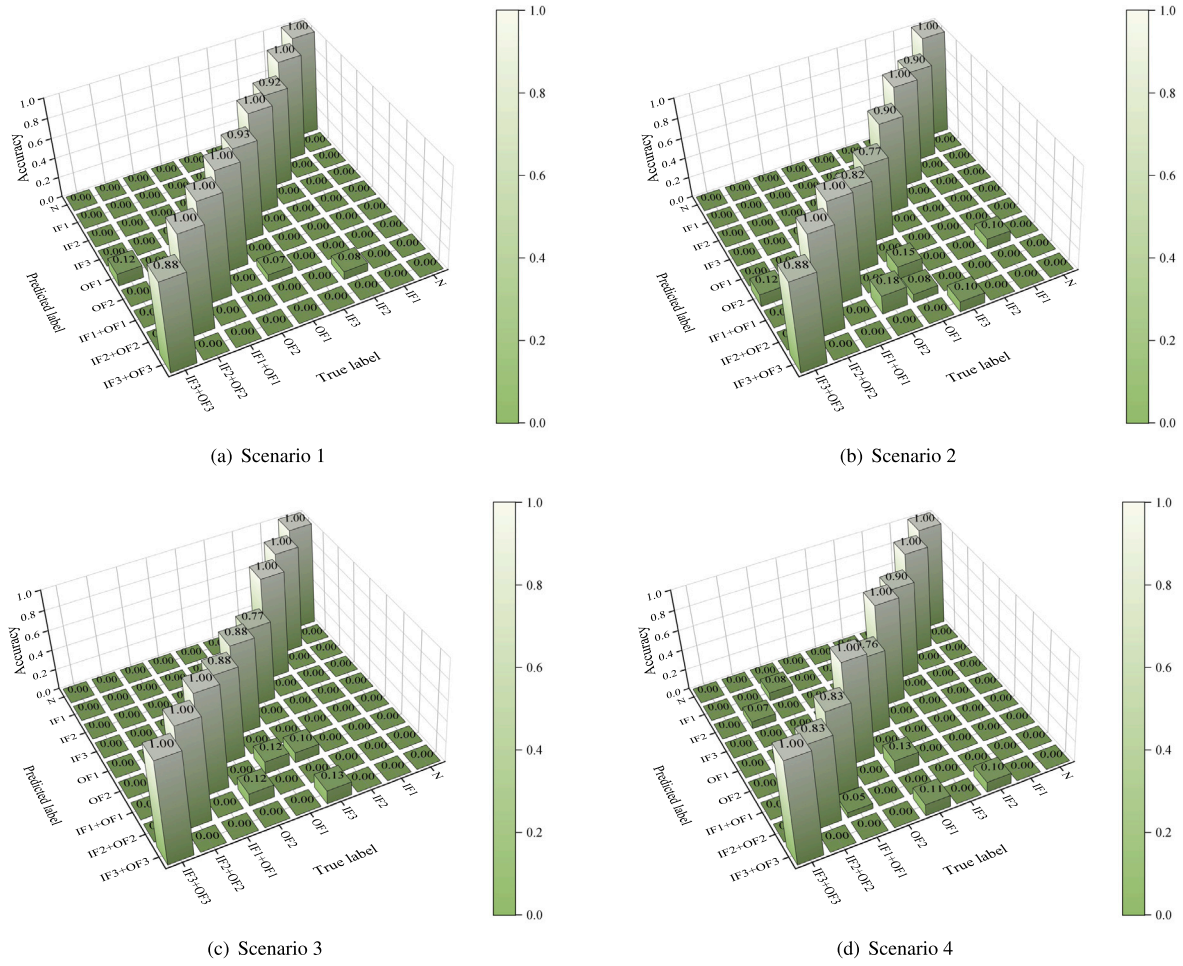
Fig. 12. Average diagnosis accuracies obtained with different methods on different FSFD tasks under the four cross-working condition scenarios of PU dataset.

by plotting the ROC curve for each fault class and calculating the corresponding area under the curve (AUC). The higher the AUC value, the better the performance of CMS-MTL under different classification thresholds. As depicted in Fig. 14, the macro and micro average values obtained with CMS-MTL under the four different cross-working condition scenarios are all above 0.99, which proves the effectiveness of CMS-MTL in FSFD.

4.2.3. FSFD under cross-equipment scenarios

To validate the FSFD ability of CMS-MTL under the cross-equipment scenarios, some FSFD experiments are performed under the following three different cross-equipment scenarios.

- The cross-equipment scenario of CWRU → PU: the time–frequency images of different fault classes under the three different rotating



(a) Scenario 1 (b) Scenario 2

(c) Scenario 3 (d) Scenario 4

Fig. 13. Confusion matrices obtained with CMS-MTL on the 9-class 1-shot tasks under four different cross-working condition scenarios of PU dataset.

speeds of 1772, 1750, and 1730 rpm from CWRU dataset are formed the source domain, and the time–frequency images of different fault classes under the working condition of W_1 from PU dataset are formed the target domain. The seven different methods are used to perform the 9-class 1-shot and 9-class 5-shot FSFD tasks, respectively.

- The cross-equipment scenario of PU → CWRU: the time–frequency images of different fault classes under the three different working conditions of W_1, W_2, and W_3 from PU dataset are formed the source domain, and the time–frequency images of different fault classes under the rotating speed of 1730 rpm from CWRU dataset are formed the target domain. The seven different methods are adopted to conduct the 5-class 1-shot, 5-class 5-shot, 9-class 1-shot, and 9-class 5-shot FSFD tasks, respectively.
- The cross-equipment scenario of PU → PHM: the time–frequency images of different fault classes under the three different working conditions of W_1, W_2, and W_3 from PU dataset are formed the source domain, and the time–frequency images of different fault classes from PHM dataset are formed the target domain. The seven different methods are used to carry out the 5-class 1-shot and 5-class 5-shot FSFD tasks, respectively.

Table 7 provides the FSFD accuracies obtained with seven different methods under three different cross-equipment scenarios. The average accuracies of TCNN, DCMADA, ProtoNet, SSMN, MAML-FSC, MAML++, and CMS-MTL are 62.65%, 69.28%, 70.52%, 74.88%, 73.78%, 75.22%, and 80.02%, respectively, which indicates that CMS-MTL can more accurately identify different fault classes than the other six methods under the cross-equipment scenarios. However, the accuracies of CMS-MTL obtained on the cross-equipment FSFD tasks are lower than those obtained on the cross-working condition FSFD tasks of CWRU and PU datasets, respectively. This is because the distribution discrepancies between the source and target domains are more significant, and there are unknown fault classes in the target domain under the three different cross-equipment scenarios. Therefore, the cross-equipment FSFD between different datasets is more challenging than the cross-working condition FSFD on the same dataset. As seen in Table 7, when the number of samples corresponding to each fault class in the support set of each meta-task increases from 1 to 5, the more generalized and adaptive features can be learned, making the performance of these seven methods has been raised to a certain extent. For example, the FSFD accuracies obtained on the 9-class 5-shot tasks are 8.93% and 8.24% higher than those obtained on the 9-class 1-shot tasks using CMS-MTL under CWRU → PU and PU → CWRU, respectively. The FSFD accuracies obtained on the 5-class 5-shot tasks are 8.94% and 13.19% higher than those obtained on the 5-class 1-shot tasks using CMS-MTL under PU → CWRU and PU → PHM, respectively.

Fig. 15 gives the average accuracies obtained with seven different methods on the 9-class 1-shot and 9-class 5-shot tasks under two different cross-equipment scenarios. As depicted in Fig. 15, the average accuracies of the seven methods obtained under PU → CWRU are higher than those obtained under CWRU → PU. For example, the average accuracy of CMS-MTL obtained under PU → CWRU is 11.54% higher than that obtained under CWRU → PU. The results indicate that the meta-knowledge learned from the meta-tasks on the more

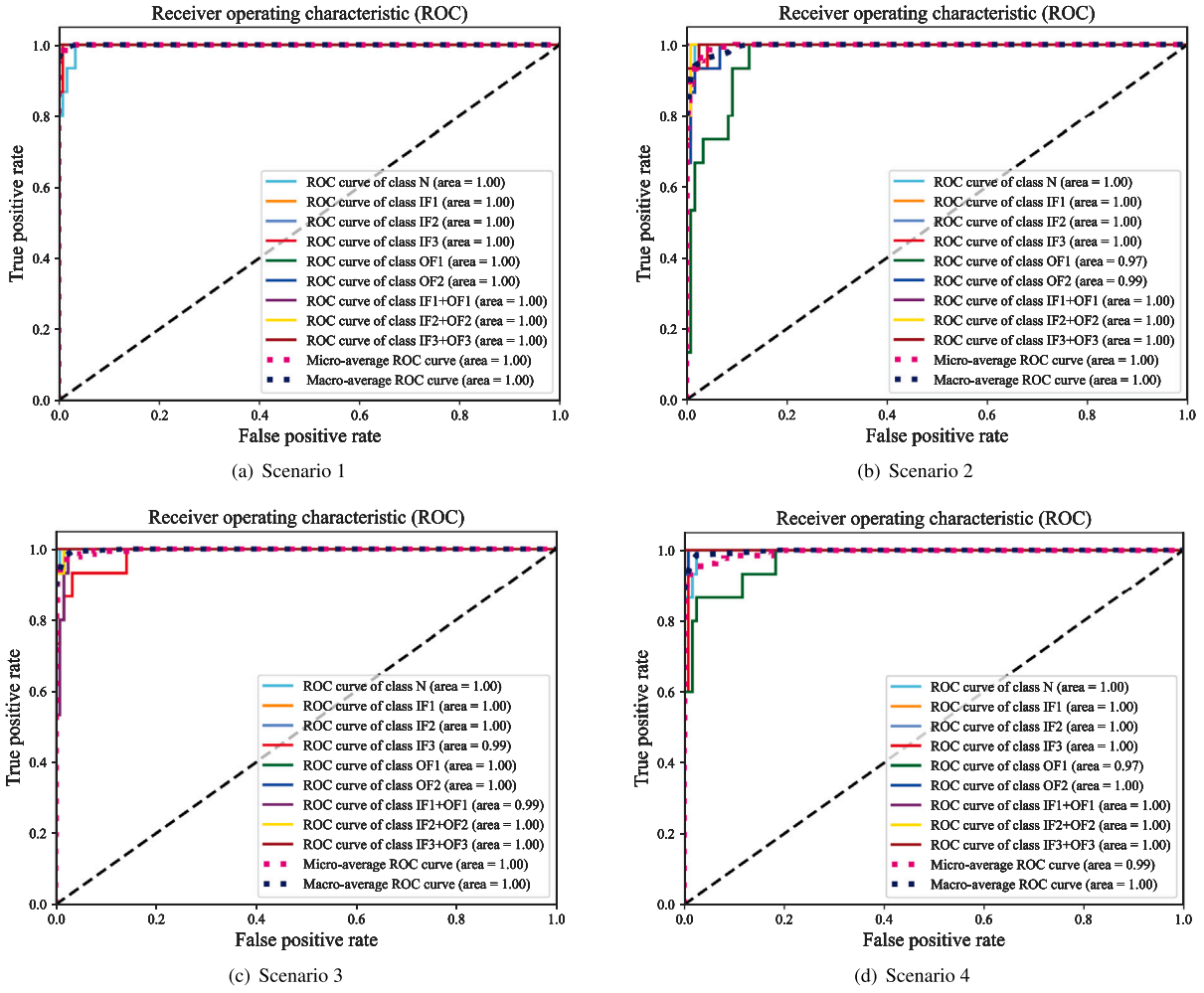


Fig. 14. ROC curves obtained with CMS-MTL on the 9-class 1-shot tasks under the four different cross-working condition scenarios of PU dataset.

Table 7
FSFD accuracies (%) obtained with different methods under different cross-equipment scenarios.

Method	CWRU → PU		PU → CWRU				PU → PHM		Average
	9-class 1-shot	9-class 5-shot	9-class 1-shot	9-class 5-shot	5-class 1-shot	5-class 5-shot	5-class 1-shot	5-class 5-shot	
TCNN [19]	48.12 ± 3.89	59.45 ± 3.34	65.89 ± 2.13	69.56 ± 1.43	66.97 ± 1.76	75.24 ± 1.64	52.45 ± 2.23	63.48 ± 1.59	62.65
DCMADA [18]	55.88 ± 2.16	63.02 ± 1.44	71.36 ± 1.90	77.90 ± 1.13	74.89 ± 1.59	82.39 ± 1.13	60.39 ± 2.05	68.37 ± 1.45	69.28
ProtoNet [20]	56.26 ± 3.54	65.19 ± 2.72	71.48 ± 3.80	78.56 ± 1.47	75.54 ± 2.24	84.83 ± 1.68	61.48 ± 2.15	70.84 ± 1.82	70.52
SSMN [23]	61.05 ± 0.96	73.25 ± 0.41	74.10 ± 0.74	81.21 ± 0.22	78.95 ± 0.85	87.65 ± 0.45	65.56 ± 1.75	77.25 ± 1.13	74.88
MAML-FSC [21]	59.18 ± 3.69	72.33 ± 3.21	73.77 ± 2.51	80.48 ± 1.72	78.67 ± 1.87	86.20 ± 1.04	64.03 ± 1.48	75.56 ± 1.60	73.78
MAML++ [36]	60.89 ± 3.30	73.89 ± 2.51	74.02 ± 2.98	82.61 ± 0.65	79.82 ± 2.14	88.93 ± 1.46	65.24 ± 2.03	76.37 ± 1.36	75.22
CMS-MTL	67.54 ± 0.66	76.47 ± 0.41	77.43 ± 0.66	85.67 ± 0.43	84.92 ± 0.97	93.86 ± 0.32	70.53 ± 1.01	83.72 ± 0.85	80.02

complex PU dataset can be better generalized to the new meta-tasks of CWRU dataset, thus achieving better FSFD accuracies under PU → CWRU. Fig. 16 gives the average accuracies obtained with seven different methods on the 5-class 1-shot and 5-class 5-shot tasks under two different cross-equipment scenarios. As depicted in Fig. 16, the average accuracies of the seven methods obtained under PU → PHM are lower than those obtained under PU → CWRU. This is because the PU and CWRU datasets only contain bearing faults, whereas the PHM dataset contains the combined gear-bearing-shaft faults. Therefore, the data distribution discrepancies between the PU and PHM datasets are greater than those between the PU and CWRU datasets. The significant increase of the distribution discrepancy leads to the performance degradation of FSFD of the seven different methods under PU → PHM. However, the average diagnosis accuracy obtained with CMS-MTL is 77.13% under PU → PHM, indicating that the proposed CMS-MTL has the ability to identify unknown faults with a few training samples.

4.3. Validation of the proposed CMHSAN model

The comparative experiments are performed on CWRU and PU datasets with the CMHSAN model and its two variants. The first variant

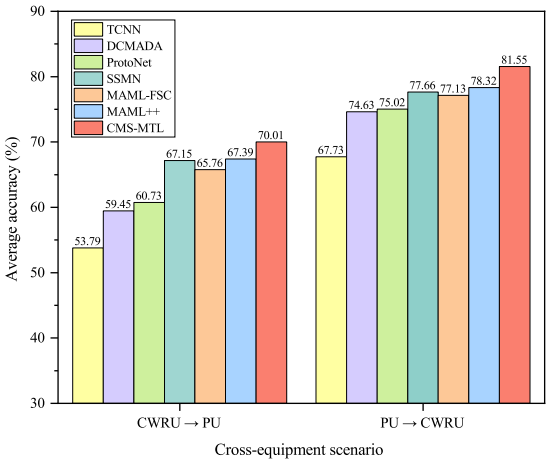


Fig. 15. Average accuracies obtained with seven different methods on the 9-class 1-shot and 9-class 5-shot tasks under two different cross-equipment scenarios.

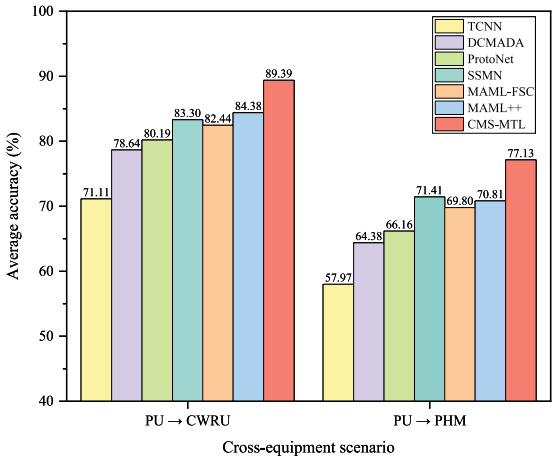


Fig. 16. Average accuracies obtained with seven different methods on the 5-class 1-shot and 5-class 5-shot tasks under two different cross-equipment scenarios.

Table 8
FSFD accuracies (%) obtained with three different models under the cross-working condition scenario of CWRU dataset.

Model	1797, 1772, 1750 rpm → 1730 rpm				Avg.
	5-class 1-shot	5-class 5-shot	10-class 1-shot	10-class 5-shot	
Model 1	70.38 ± 1.20	85.56 ± 0.89	62.00 ± 0.66	71.97 ± 0.52	72.48
Model 2	96.08 ± 0.36	97.52 ± 0.13	93.53 ± 0.31	96.69 ± 0.13	95.96
CMHSAN	99.47 ± 0.33	99.85 ± 0.05	98.67 ± 0.11	98.86 ± 0.11	99.21

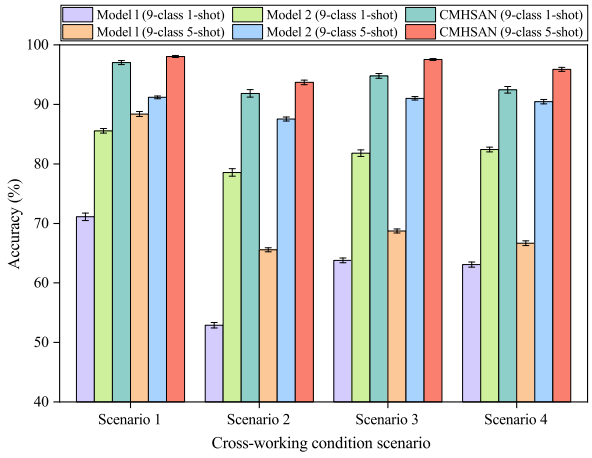


Fig. 17. FSFD accuracies obtained with three different models under the cross-working condition scenarios of PU dataset.

is to replace the Res blocks in components 1 to 3 of the CMHSAN model with Trans blocks, that is, only Trans blocks are used in the three components, which is called model 1. The second variant is to replace the Trans blocks in components 1 to 3 of the CMHSAN model with Res blocks, that is, only Res blocks are used in the three components, which is called model 2. Table 8 and Fig. 17 present the FSFD accuracies obtained with three different models under the cross-working condition scenarios of CWRU and PU datasets, respectively.

As seen in Table 8 and Fig. 17, the accuracies obtained with model 1 on all FSFD tasks are low, mainly because the model has some shortcomings in the local feature extraction in the case of a few samples. Model 2 performs better than model 1, because the use of multiple Res blocks greatly enhances the local feature learning ability of the model in the case of a few samples. However, there is a certain gap between the FSFD accuracies of model 2 and those of the CMHSAN model, mainly because model 2 lacks attention to the global feature extraction. As shown in Table 8, the average FSFD accuracy of the CMHSAN model is 26.73% and 3.25% higher than that of model 1 and model 2 under the cross-working condition scenario of CWRU dataset, respectively. As depicted in Fig. 17, the average FSFD accuracy of the CMHSAN model is 27.64% and 9.09% higher than that of model 1 and model 2 under the four cross-working condition scenarios of PU dataset, respectively. This is mainly because the CMHSAN model, which ingeniously combines Res and Trans blocks, can fully extract the local and global features of bearing faults in the case of a few samples.

The above results reveal that the global and local features play an important role in FSFD. The proposed CMHSAN model fully considers the global and local feature information of the input time–frequency images. The CMHSAN model can better capture the time-frequency features with higher correlation with bearing fault classes through MHSA, and reduce the computational complexity through the weight sharing mechanism of convolution, thus improving the performance of FSFD.

To further analyze the performance differences of the three different models in the pre-training and meta-training stages, the pre-training and meta-training are performed under the cross-working condition scenario of CWRU dataset, where the meta-training is performed on the 10-class 1-shot tasks. Fig. 18 gives the pre-training time and meta-training time of three different models. As seen in Fig. 18, model 1 has the longest total training time, followed by the CMHSAN model, and model 2 has the shortest total training time. Specifically, the pre-training time of the CMHSAN model is 37.20% less than that of model 1 and 22.75% more than that of model 2, and the meta-training time of the CMHSAN model is 33.82% less than that of model 1 and 12.45% more than that of model 2. This is because the computational complexities of model 1, model 2, and the CMHSAN model are different, where the total number of parameters of model 1, model 2, and the CMHSAN model are 14 622 986, 10 147 274, and 11 954 954, respectively. Figs. 19 and 20 display the FSFD accuracies and losses obtained with three different models in the pre-training and meta-training stages, respectively. Compared with model 1 and model 2,

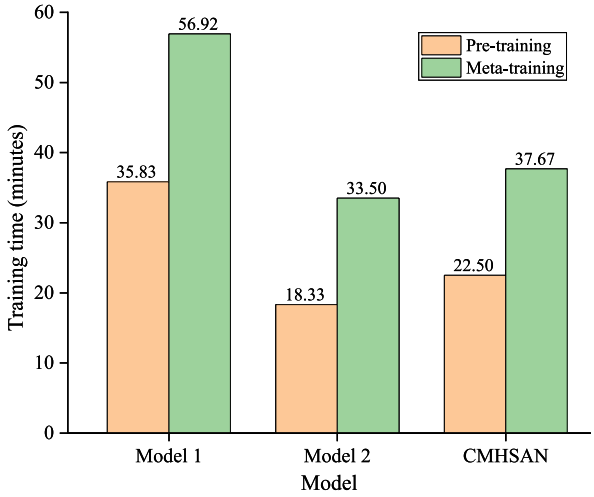


Fig. 18. Pre-training time and meta-training time of three different models.

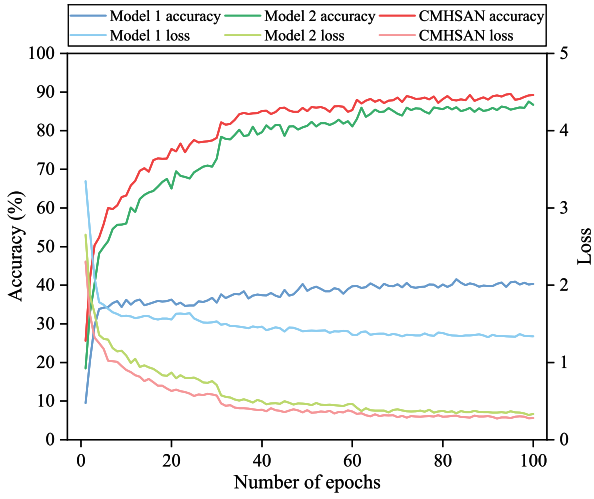


Fig. 19. FSFD accuracies and losses obtained with three different models in the pre-training stage.

the CMHSAN model obtains higher and more stable FSFD accuracies and has better convergence, which indicates that the CMHSAN model can learn the fault feature representations more effectively, and has excellent generalization performance and training efficiency.

4.4. Validation of the meta-task re-training strategy

The comparative experiments between the proposed CMS-MTL method with the MTRT strategy and that without the MTRT strategy are conducted on CWRU and PU datasets. Fig. 21(a) provides the comparison of the accuracies obtained on different FSFD tasks under the cross-working condition scenario of CWRU dataset, where the FSFD accuracy obtained with the MTRT strategy is 1.35% higher than that obtained without the MTRT strategy on the 10-class 1-shot task. Fig. 21(b) gives the comparison of the accuracies obtained on the 9-class 1-shot tasks under the four different cross-working condition scenarios of PU dataset. The FSFD accuracies obtained with the MTRT strategy are 1.47%, 2.07%, 1.93%, and 2.81% higher than those obtained without the MTRT strategy under the four different cross-working condition scenarios, respectively. Fig. 21(c) presents the

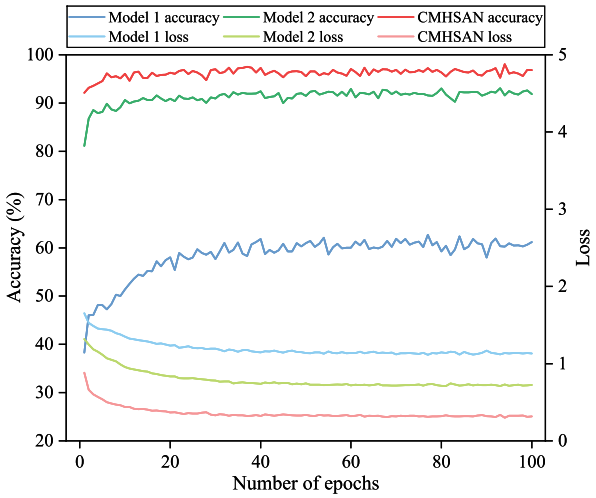


Fig. 20. FSFD accuracies and losses obtained with three different models in the meta-training stage.

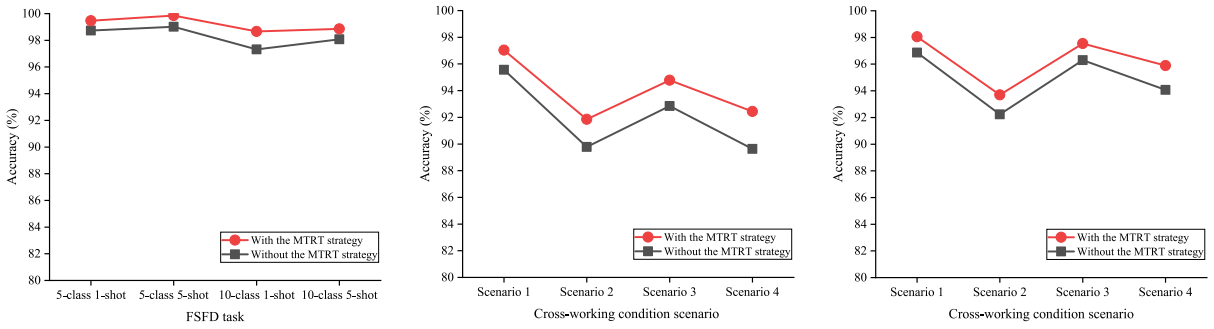
comparison of the accuracies obtained on the 9-class 5-shot tasks under the four different cross-working condition scenarios of PU dataset. The FSFD accuracies obtained with the MTRT strategy are 1.19%, 1.47%, 1.24%, and 1.82% higher than those obtained without the MTRT strategy under scenarios 1 to 4, respectively. The results demonstrate that using the MTRT strategy to re-train the CMHSAN model during the meta-training stage is helpful for increasing the FSFD accuracies.

As depicted in Fig. 21, the improvement of FSFD accuracy brought by using the MTRT strategy on PU dataset is apparent. This is because the distribution discrepancies between the source and target domains are significant under scenarios 1 to 4, which makes it difficult for some fault classes to be accurately diagnosed. The use of the MTRT strategy can re-train the CMHSAN model according to the hard classes during the meta-training stage, thereby enabling the model to more accurately distinguish the hard classes. Therefore, the FSFD accuracies obtained using the CMHSAN model on different FSFD tasks of PU dataset have been effectively improved after adopting the proposed meta-task re-training strategy.

5. Conclusion

In the present study, a novel CMHSAN-based MTL approach for FSFD is proposed, which combines meta-learning and transfer learning, and can use a few labeled training samples to obtain a robust FSFD model that can quickly adapt to new working conditions or fault classes. The meticulously designed CMHSAN integrates the convolution blocks and MHSA blocks, which can effectively enhance the local and global feature extraction ability of the CMHSAN model. The proposed approach enables the pre-trained CMHSAN model to quickly adapt to new FSFD tasks, greatly improving the cross-domain FSFD performance. By using the proposed meta-task re-training strategy, the CMHSAN model can learn more transferable fault diagnosis knowledge during the meta-training stage, which significantly improves the generalization of the model. The effectiveness of the proposed approach has been verified through extensive experiments. This approach achieves the average diagnosis accuracies of 99.21% and 95.16% on different FSFD tasks under the cross-working condition scenarios of CWRU and PU datasets, respectively, which are superior to the other comparison methods, suggesting that the proposed approach has excellent FSFD ability.

Future research will focus on improving the interpretability of the proposed FSFD model to enhance its acceptability and confidence in practical applications.



(a) Different FSFD tasks of CWRU dataset (b) 9-class 1-shot tasks of PU dataset (c) 9-class 5-shot tasks of PU dataset

Fig. 21. FSFD accuracies of CMS-MTL method with and without the MTRT strategy.

CRediT authorship contribution statement

Lanjun Wan: Writing – review & editing, Writing – original draft, Software, Methodology, Investigation, Funding acquisition, Conceptualization. **Le Huang:** Writing – review & editing, Writing – original draft, Validation, Software, Methodology, Data curation. **Jiaen Ning:** Visualization, Validation, Software, Methodology. **Changyun Li:** Supervision, Funding acquisition. **Keqin Li:** Writing – review & editing, Methodology, Formal analysis.

Declaration of competing interest

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

Data availability

Data will be made available on request.

Acknowledgments

This work was supported by the Hunan Provincial Natural Science Foundation of China [grant number 2023JJ30217]; the Scientific Research Foundation of Hunan Provincial Education Department, China [grant number 21B0547]; and the National Natural Science Foundation for Young Scientists of China [grant number 61702177].

References

[1] X. Chen, R. Yang, Y. Xue, M. Huang, R. Ferrero, Z. Wang, Deep transfer learning for bearing fault diagnosis: A systematic review since 2016, IEEE Trans. Instrum. Meas. 72 (2023) 3508221.
[2] B.A. Tama, M. Vania, S. Lee, S. Lim, Recent advances in the application of deep learning for fault diagnosis of rotating machinery using vibration signals, Artif. Intell. Rev. 56 (5) (2023) 4667–4709.
[3] D. Ruan, J. Wang, J. Yan, C. Gühmann, CNN parameter design based on fault signal analysis and its application in bearing fault diagnosis, Adv. Eng. Inform. 55 (2023) 101877.
[4] Y. Hou, J. Wang, Z. Chen, J. Ma, T. Li, Diagnosisformer: An efficient rolling bearing fault diagnosis method based on improved Transformer, Eng. Appl. Artif. Intell. 124 (2023) 106507.
[5] J. Tong, S. Tang, Y. Wu, H. Pan, J. Zheng, A fault diagnosis method of rolling bearing based on improved deep residual shrinkage networks, Measurement 206 (2023) 112282.
[6] T. Han, W. Xie, Z. Pei, Semi-supervised adversarial discriminative learning approach for intelligent fault diagnosis of wind turbine, Inform. Sci. 648 (2023) 119496.
[7] Y. Chen, M. Rao, K. Feng, M.J. Zuo, Physics-Informed LSTM hyperparameters selection for gearbox fault detection, Mech. Syst. Signal Process. 171 (2022) 108907.
[8] Y. Yao, T. Han, J. Yu, M. Xie, Uncertainty-aware deep learning for reliable health monitoring in safety-critical energy systems, Energy 291 (2024) 130419.
[9] T. Zhang, J. Chen, F. Li, K. Zhang, H. Lv, S. He, E. Xu, Intelligent fault diagnosis of machines with small & imbalanced data: A state-of-the-art review and possible extensions, ISA Trans. 119 (2022) 152–171.
[10] J. Wei, H. Huang, L. Yao, Y. Hu, Q. Fan, D. Huang, New imbalanced fault diagnosis framework based on Cluster-MWMOTE and MFO-optimized LS-SVM using limited and complex bearing data, Eng. Appl. Artif. Intell. 96 (2020) 103966.
[11] J. Li, Q. Zhu, Q. Wu, Z. Fan, A novel oversampling technique for class-imbalanced learning based on SMOTE and natural neighbors, Inform. Sci. 565 (2021) 438–455.
[12] F. Zhou, S. Yang, H. Fujita, D. Chen, C. Wen, Deep learning fault diagnosis method based on global optimization GAN for unbalanced data, Knowl.-Based Syst. 187 (2020) 104837.
[13] J. Liu, C. Zhang, X. Jiang, Imbalanced fault diagnosis of rolling bearing using improved MsR-GAN and feature enhancement-driven CapsNet, Mech. Syst. Signal Process. 168 (2022) 108664.
[14] W. Li, R. Huang, J. Li, Y. Liao, Z. Chen, G. He, R. Yan, K. Gryllias, A perspective survey on deep transfer learning for fault diagnosis in industrial scenarios: Theories, applications and challenges, Mech. Syst. Signal Process. 167 (2022) 108487.
[15] Y. Li, Y. Song, L. Jia, S. Gao, Q. Li, M. Qiu, Intelligent fault diagnosis by fusing domain adversarial training and maximum mean discrepancy via ensemble learning, IEEE Trans. Ind. Inform. 17 (4) (2021) 2833–2841.
[16] Y. Tan, L. Guo, H. Gao, L. Zhang, Deep coupled joint distribution adaptation network: A method for intelligent fault diagnosis between artificial and real damages, IEEE Trans. Instrum. Meas. 70 (2021) 3043510.
[17] W. Li, Z. Chen, G. He, A novel weighted adversarial transfer network for partial domain fault diagnosis of machinery, IEEE Trans. Ind. Inform. 17 (3) (2021) 1753–1762.
[18] L. Wan, Y. Li, K. Chen, K. Gong, C. Li, A novel deep convolution multi-adversarial domain adaptation model for rolling bearing fault diagnosis, Measurement 191 (2022) 110752.
[19] Z. Chen, K. Gryllias, W. Li, Intelligent fault diagnosis for rotary machinery using transferable convolutional neural network, IEEE Trans. Ind. Inform. 16 (1) (2020) 339–349.
[20] Y. Feng, J. Chen, J. Xie, T. Zhang, H. Lv, T. Pan, Meta-learning as a promising approach for few-shot cross-domain fault diagnosis: Algorithms, applications, and prospects, Knowl.-Based Syst. 235 (2022) 107646.
[21] S. Zhang, F. Ye, B. Wang, T.G. Habetler, Few-shot bearing fault diagnosis based on model-agnostic meta-learning, IEEE Trans. Ind. Appl. 57 (5) (2021) 4754–4764.
[22] T. Yang, T. Tang, J. Wang, C. Qiu, M. Chen, A novel cross-domain fault diagnosis method based on model agnostic meta-learning, Measurement 199 (2022) 111564.
[23] Y. Feng, J. Chen, T. Zhang, S. He, E. Xu, Z. Zhou, Semi-supervised meta-learning networks with squeeze-and-excitation attention for few-shot fault diagnosis, ISA Trans. 120 (2022) 383–401.
[24] R. Ma, T. Han, W. Lei, Cross-domain meta learning fault diagnosis based on multi-scale dilated convolution and adaptive relation module, Knowl.-Based Syst. 261 (2023) 110175.
[25] J. Lin, H. Shao, X. Zhou, B. Cai, B. Liu, Generalized MAML for few-shot cross-domain fault diagnosis of bearing driven by heterogeneous signals, Expert Syst. Appl. 230 (2023) 120696.
[26] Q. Sun, Y. Liu, Z. Chen, T.-S. Chua, B. Schiele, Meta-transfer learning through hard tasks, IEEE Trans. Pattern Anal. Mach. Intell. 44 (3) (2022) 1443–1456.
[27] C. Li, S. Li, H. Wang, F. Gu, A.D. Ball, Attention-based deep meta-transfer learning for few-shot fine-grained fault diagnosis, Knowl.-Based Syst. 264 (2023) 110345.
[28] L. Ma, B. Jiang, L. Xiao, N. Lu, Digital twin-assisted enhanced meta-transfer learning for rolling bearing fault diagnosis, Mech. Syst. Signal Process. 200 (2023) 110490.

[29] Z. Lei, P. Zhang, Y. Chen, K. Feng, G. Wen, Z. Liu, R. Yan, X. Chen, C. Yang, Prior knowledge-embedded meta-transfer learning for few-shot fault diagnosis under variable operating conditions, Mech. Syst. Signal Process. 200 (2023) 110491.

[30] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, in: Proc. 31st Int. Conf. Neural Inf. Process. Syst., NIPS, Long Beach, CA, USA, 2017, pp. 6000–6010.

[31] W. Wang, E. Xie, X. Li, D.-P. Fan, K. Song, D. Liang, T. Lu, P. Luo, L. Shao, PVT v2: Improved baselines with Pyramid vision transformer, Comput. Vis. Media 8 (3) (2022) 415–424.

[32] W. A. Smith, R. B. Randall, Rolling element bearing diagnostics using the Case Western Reserve University data: A benchmark study, Mech. Syst. Signal Process. 64 (2015) 100–131.

[33] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., CVPR, Boston, MA, USA, 2015, pp. 1–9.

[34] C. Lessmeier, J.K. Kimotho, D. Zimmer, W. Sextro, Condition monitoring of bearing damage in electromechanical drive systems by using motor current signals of electric motors: A benchmark data set for data-driven classification, in: Proc. Eur. Conf. PHM Soc., PHME16, Vol. 3, Bilbao, Bizkaia, Spain, 2016, pp. 1–17.

[35] B. Zhang, W. Li, X.-L. Li, S.-K. Ng, Intelligent fault diagnosis under varying working conditions based on domain adaptive convolutional neural networks, IEEE Access 6 (2018) 66367–66384.

[36] A. Antoniou, H. Edwards, A. Storkey, How to train your MAML, 2018, http://dx.doi.org/10.48550/arXiv.1810.09502, arXiv:1810.09502.