Deep Multilayer Sparse Regularization Time-Varying Transfer Learning Networks With Dynamic Kullback–Leibler Divergence Weights for Mechanical Fault Diagnosis

Feiyu Lu, *Graduate Student Member, IEEE*, Qingbin Tong, Xuedong Jiang, Ziwei Feng, *Graduate Student Member, IEEE*, Jianjun Xu, and Jingyi Huo

***Abstract*—Rotating machinery is widely used in industrial production, and its reliable operation is crucial for ensuring production safety and efficiency. Mechanical equipment often faces the challenge of variable speeds. However, existing research pays little attention to domain-adaptive and cross-device diagnostic tasks under time-varying conditions. To fill this research gap and address the serious domain shift problem in cross-device fault diagnosis tasks under time-varying speeds, this article proposes a deep multilayer sparse regularization time-varying transfer learning network (DMsrTTLN) with dynamic Kullback–Leibler divergence weights (DKLDW). The main contributions and innovations of DMsrTTLN are as follows: First, a multilayer sparse regularization module to effectively reduce speed fluctuations; second, an amplitude activation function to enhance the differentiation of data with different labels; third, the kurtosis maximum mean discrepancy, where the Gaussian kernel function adaptively adjusts according to the kurtosis values of the data to enhance domain adaptation capability; and finally, the DKLDW mechanism dynamically balances distance and adversarial metrics to improve model convergence and stability. The DMsrTTLN model with DKLDW exhibits strong generalization performance in cross-device domain shift scenarios. Experimental validation in the same-device and cross-device scenarios is performed on three mechanical machines under time-varying speeds, and the results are compared with those of six state-of-the-art approaches. The results showed that the DMsrTTLN has a better convergence effect and greater diagnostic accuracy.**

***Index Terms*—Cross-device, fault diagnosis, maximum mean diversity (MMD), time-varying, transfer learning.**

Manuscript received 26 April 2024; revised 7 July 2024; accepted 30 July 2024. This work was supported in part by Beijing Natural Science Foundation under Grant L211010, and in part by the Fundamental Research Funds for the Central Universities under Grant 2023JBZY039. Paper no. TII-24-1991. *(Corresponding author: Qingbin Tong.)*

The authors are with the School of Electrical Engineering, Beijing Jiaotong University, Beijing 100044, China (e-mail: 21117039@bjtu.edu.cn; qbtong@bjtu.edu.cn; xdjiang@bjtu.edu.cn; 22110480@bjtu.edu.cn; jjxu@bjtu.edu.cn; jyhuo@bjtu.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TII.2024.3438229.

Digital Object Identifier 10.1109/TII.2024.3438229

I. Introduction

ROTATING mechanical equipment plays an important role in production and daily life. An effective fault diagnosis model can monitor the health status of equipment in real time, thereby improving equipment reliability [1], [2].

The wave of deep learning technology is sweeping across the world [3], [4], [5], [6]. Recently, various deep learning frameworks have been used for fault diagnosis tasks, such as convolutional networks for identifying fault size and type [7], graph neural networks for decoding data structure information, and generative adversarial networks for solving small sample data problems. However, most of the above studies were conducted under the same data distribution. Rotating mechanical equipment is subject to different program settings and task execution requirements and often operates under variable speed conditions. Liu et al. [8] noted that existing research on variable-speed faults can be classified into two categories. In the first category, the machine operates at several different speeds, but the speed is constant. In the second category, the rotational speed, known as the time-varying speed, is nonlinear and varies with time. Depending on the device source of the data, it can also be further subdivided into fault diagnosis tasks under the same device and across devices.

Many scholars have provided numerous solutions for the first category of the variable speed problem [9], [10], [11], [12]. In 2018, Shao et al. [13] achieved high-precision transfer fault diagnosis by pretraining and fine-tuning strategies and conducted feasibility verification in a bearing cross-speed scenario. In 2022, considering the situation where the target domain data are not visible, Yang et al. [14] proposed a multisource transfer learning network, which was validated for its effectiveness in a gear-driven drilling test rig.

To solve the cross-device diagnosis problem at variable speeds. Guo et al. [7] constructed a deep convolutional transfer learning network (DCTLN) using maximum mean diversity (MMD) and domain classification and implemented bearing migration fault diagnosis tasks between three different devices. Yang et al. [15] reported that distance metrics cannot be used to solve for joint distribution differences. They explored a clustering-based conditional distribution to realize cross-device fault diagnosis and proposed an optimal transportation

embedded joint distribution similarity measure (OT-JDSM). Validity verification experiments were conducted on 12 different datasets. Yang et al. [16] also proposed a deep targeted transfer learning method, which also achieved cross-device fault diagnosis. However, this method utilizes the label information in the target domain, and the performance of OT-JDSM and DTTL in scenarios where the target domain data label cannot be used is questionable.

On the other hand, for the second category of time-varying speed, there is relatively less research at present. Chang et al. [17] designed methods involving alternative kernel networks and squeeze-and-excitation attention. These methods maintain accuracy and efficiency in bearing fault diagnosis experiments under speed fluctuations. Liang et al. [18], to address the domain shift caused by speed variations, developed a semisupervised subdomain adaptation graph convolutional network and performed feasibility verification on gear and bearing fault datasets under variable speeds. In the same year, Chen et al. [19], considering the issue of data scarcity under time-varying speed, proposed the hybrid augmented network with a balance domain window. However, those studies were conducted based on issues within the same device, and the training set data had labels. To our knowledge, a cross-device model under a time-varying speed has not been developed thus far. Liu et al. [8] provides a possible explanation: vibration signals under time-varying speeds exhibit strong nonstationary characteristics and feature variability. This results in current intelligent diagnosis models being unable to identify invariant features, thereby compromising their generalization performance. Nevertheless, cross-device fault diagnosis under time-varying speeds has significant research value for the health monitoring of mechanical equipment. In comparison to labeled training set data in the same device issue, the application scenario of this article in cross-device tasks involves unlabeled training data in the target domain.

In fault diagnosis tasks based on transfer learning, a key challenge is how to effectively balance the importance of loss functions. Many scholars have conducted extensive research on this issue. Zhou et al. [20] used dynamic weight factors to adjust the influence of the marginal probability distribution and conditional probability distribution on the model. When the value of the weight factor is close to 1, the weight of the marginal distribution loss function is greater. Conversely, the weight of the conditional probability distribution loss function is greater. Liu et al. [21] constructed weight factors based on exponential functions to balance the importance between distance metric loss and subdomain adaptation loss. Similarly, Yang et al. [16] used exponential functions to construct the importance between divergence loss and triplet loss and applied it to cross-device fault diagnosis tasks. However, current research lacks in-depth exploration of the importance of adversarial metrics and distance metrics. This leads to the possibility that in cross-domain learning tasks, models may not fully utilize the information between distance metrics and adversarial metrics, thereby affecting the performance and generalizability of the model.

In addition, from a technical perspective, the current research still faces several issues and gaps.

1) The lack of a module that can effectively eliminate speed fluctuations.
2) The MMD loss function cannot adaptively narrow the marginal distribution based on the physical information of the vibration signal. This leads to the inability of the distance metric to extract effective fault information from faulty data, restricting the performance of the MMD.
3) The balancing weight factor between the distance and adversarial metric has not been thoroughly studied.

To address these challenges and fill this research gap, this article proposes a deep multilayer sparse regularization time-varying transfer learning networks (DMsrTTLNs) with dynamic Kullback–Leibler divergence weights (DKLDWs). The DMsrTTLN can achieve fault diagnosis under the same device. The DMsrTTLN consists of three parts: a feature extractor; multilayer sparse regularization (MSR); and a classifier. The feature extractor, which is based on convolutional neural networks, directly extracts diverse fault features from the raw vibration signals. MSR eliminates the influence of speed fluctuations from the perspective of feature regularization, significantly improving the diagnostic performance of the model. The classifier, which is based on the amplitude activation function (AAF), uses random sample activation to enhance the discriminability of features. For variable speed conditions, combining the DKLDW and DMsrTTLN methods can achieve high-precision intelligent fault diagnosis tasks across devices. DKLDW adjusts the weight values between distance metrics and adversarial metrics from the perspective of KL divergence, thereby reducing the distance between the source domain and the target domain data. By conducting same-device and cross-device fault diagnosis tasks under variable speed conditions on three different mechanical devices, the effectiveness and superiority of the proposed method are fully verified.

The primary contributions are as follows.

1) To address the impact of time-varying speed on model performance, we design the MSR strategy, which consists of a plug-and-play sparse regularization module. This means that it can be applied to the feature output layer of any deep learning model. Ablation experiments demonstrate that MSR can significantly eliminate the problem of speed fluctuations.
2) Considering that current activation functions cannot adapt to vibration signals under time-varying speeds, leading to inadequate model generalization performance, we propose a new activation function determined by the peak-to-peak value of the original signal for activating the output features. To perform this process flexibly and avoid manual parameter adjustment, we adopt the random sample activation method. The results show that under the proposed activation function, the fault diagnosis accuracy is improved.
3) The current domain adaptive loss function does not consider the performance improvement effect of kurtosis at time-varying speeds, resulting in limited model performance. A kurtosis-based MMD (KMMD) algorithm is proposed, which dynamically selects parameters for the Gaussian kernel function. This approach addresses the shortcomings of MMD in conditional domain adaptation. The results show that the proposed KMMD algorithm can

improve the diagnostic performance of the model in fault diagnosis tasks from the perspective of kurtosis.

4) Different loss functions may have different weights. We propose a weight factor based on KL divergence to address the problem of dynamically balancing distance domain loss functions and adversarial domain loss functions. Comparative experiments show that under the proposed weight factor, the weight values of loss functions can be dynamically adjusted.
5) Based on the above algorithms, we construct DMsrTTLN with DKLDW. Compared to six advanced methods for cross-speed and cross-device fault diagnosis, the proposed method outperforms the other methods in terms of all three performance indicators.

Limitations of existing methods are as follows.

1) There is a lack of modules that effectively eliminate the influence of speed fluctuations, resulting in unstable performance of models under time-varying speed conditions.
2) Traditional activation functions and domain adaptation methods are insufficient for adapting to speed changes and differences in data distribution across devices when dealing with complex industrial data.
3) Existing MMD loss functions cannot adaptively shrink marginal distributions based on the physical information of vibration signals, limiting the effectiveness of fault information extraction.
4) The lack of in-depth research on balancing weight factors between distance metrics and adversarial metrics leads to inadequate convergence performance and stability of the model.

Innovations and necessity of the proposed method are as follows.

1) The MSR module significantly enhances the robustness of feature extraction by reducing the interference of speed fluctuations, addressing the insufficient handling of speed fluctuations in existing methods.
2) The AAF enhances the differentiation of labeled data, improves the generalizability of the model, and meets the vibration signal processing requirements under time-varying speed conditions.
3) The KMMD method adjusts the parameters of the Gaussian kernel function adaptively, enhancing the domain adaptation capability and demonstrating excellent performance in diagnostic needs across different mechanical devices.
4) The DKLDW mechanism dynamically balances distance metrics and adversarial metrics, significantly improving the convergence performance and stability of the model and thereby achieving excellent diagnostic results in different time-varying speed and cross-device scenarios.

II. Problem Formulation

To clearly understand the fault diagnosis problem being addressed, we provide descriptions of fault diagnosis scenarios under time-varying speeds for both same-device and cross-device scenarios. Same-device scenarios: Let D^m represent the data from different devices, where m is the machine type. For fault diagnosis within the same device, $D_{\text{train}}^{m\_\text{train}} = \{(\boldsymbol{x}_{\text{train}}^{(i)}, \boldsymbol{y}_{\text{train}}^{(i)}, \boldsymbol{v}_{\text{train}}^{(i)}\}_{i=1}^{N}$ and $D_{\text{test}}^{m\_\text{test}} = \{(\boldsymbol{x}_{\text{test}}^{(i)}, \boldsymbol{v}_{\text{test}}^{(i)})\}_{i=1}^{M}$ represent the training and testing set data, respectively. N and M are the sample sizes, $\boldsymbol{x}_{\text{train}}^{(i)}$ is the ith training sample, $\boldsymbol{y}_{\text{train}}^{(i)}$ is the corresponding label, and $\boldsymbol{v}_{\text{train}}^{(i)}$ is the corresponding rotational speed. Notably, for the first category of variable speed mentioned earlier, $\boldsymbol{v}_{\text{train}}^{(i)}$ is a constant, while in this article, $\boldsymbol{v}_{\text{train}}^{(i)}$ is a time-varying function. This reflects the dynamic changes during the operation of the machine.

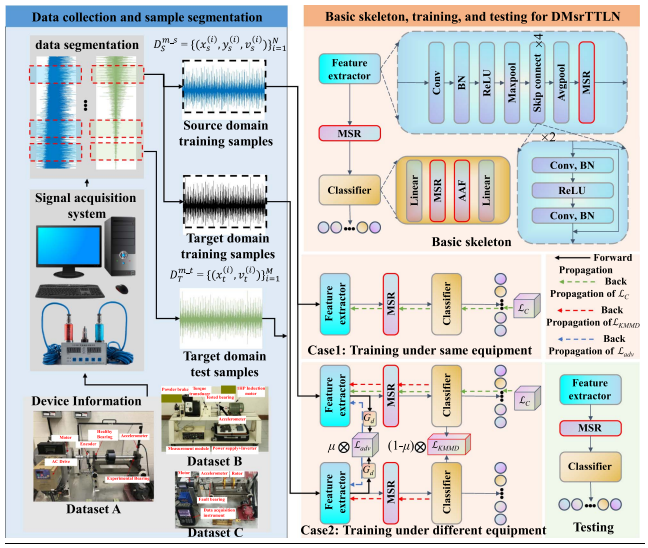

Fig. 1. Structural diagram of DMsrTTLN with DKLDW.

Cross-Device Scenarios: For cross-device fault diagnosis, $D_S^{m\_s} = \{(\boldsymbol{x}_s^{(i)}, \boldsymbol{y}_s^{(i)}, \boldsymbol{v}_s^{(i)}\}_{i=1}^{N}$ and $D_T^{m\_t} = \{(\boldsymbol{x}_t^{(i)}, \boldsymbol{v}_t^{(i)})\}_{i=1}^{M}$ represent the source and target domain datasets, respectively. Unlike the same device, *m_s* and *m_t* are different device types. The task to be accomplished for cross-device diagnosis is, in the case where $\boldsymbol{v}_s^{(i)}$ and $\boldsymbol{v}_t^{(i)}$ are time-varying functions, to achieve the transfer diagnostic task from $D_S^{m\_s}$ to $D_T^{m\_t}$.

III. Proposed Method

The proposed model is shown in Fig. 1. First, vibration signals are collected from equipment using signal acquisition devices to create training and testing sets. Second, a basic framework is established, consisting of three parts: a ResNet18-based feature extractor; an MSR; and a classifier. The detailed specifications of each structure are shown in Fig. 1. Subsequently, the basic framework is applied to perform fault diagnosis on both the same and different devices. Finally, the trained model is utilized for testing.

A. Proposed DMsrTTLN With the DKLDW Model

1) Multilayer Sparse Regularization (MSR): Domain shift is the most significant challenge in cross-device learning. When a model attempts to migrate from one domain to a completely different domain, due to distribution differences, activated features may exhibit instability. This instability makes it difficult for the model to effectively capture the differences between data from different domains.

MSR emphasizes regularization on every column of the data matrix, meaning that each feature is sparsified. Sparsification is achieved by forcing most elements in the matrix to zero. Specifically, it promotes high dispersion of data, preventing the same features from being consistently activated. When features are consistently activated, models tend to overly rely on these features, thereby increasing the difficulty of learning under distributional changes. By enforcing dispersion among features, MSR ensures that the model can learn more generalized features even when significant distribution changes occur without being disrupted by residual features.

We study the problem of data distribution deviation caused by speed fluctuations from the perspective of data sparsity to reduce the number of features needed to reduce distribution deviation. Within the framework of migration learning, we are committed to solving cross-device tasks. Through MSR processing of the input data matrix $\mathbf{X}$, we can obtain the feature matrix $\mathbf{Y}$. The following is the mathematical model of the MSR

$$\mathbf{Y}_1 = \mathbf{X}/\left(f\left(f\left(\mathbf{X}\right)\right)\right) = X_{ij}/\left(f\left(\sqrt{\sum_{k=1}^{m} X_{kj}^2 + \varepsilon}\right)\right) \tag{1}$$

$$\mathbf{Y} = \mathbf{Y}_1/f\left(\mathbf{Y}_1\right) = Y_{ij}/\sqrt{\sum_{k=1}^{m} Y_{ik}^2 + \varepsilon} \tag{2}$$

where $f = \sqrt{\sum_{k=1}^{m} X_{kj}^2 + \varepsilon}$ is the soft absolute value function, with ε taking the value of 1e-8. The value of $\varepsilon$is determined based on the empirical findings in [22]. The activation functions in (1) and (2) are specifically designed for row and column regularization. The activation function plays a crucial role in the row and column regularization of the MSR [22]. MSR effectively reduces speed fluctuation interference through the following mechanisms.

1) Achieving a more uniform distribution in the feature space to alleviate domain shifts caused by speed fluctuations.
2) Reducing the number of active features to diminish the impact of speed fluctuations on feature representation.
3) Ensuring high dispersion of features to minimize the sustained impact of speed changes on feature activation.

2) Amplitude Activation Function (AAF): Under time-varying speeds, the amplitudes of fault signals also exhibit corresponding time-varying characteristics [8], and the amplitude modulation characteristics of fault signals are variable. This poses a challenging problem for existing models based on constant speed with inherent amplitude modulation characteristics. Additionally, most activation functions do not specifically extract invariant features from time-varying speed signals. Considering the potential impact of speed fluctuations on deep learning models based on amplitude, we developed an AAF. This function dynamically and randomly activates the output features of the model, determining which feature values can be scaled, thereby expanding the feature distances under different labels. The mathematical model for AAF is as follows.

Assuming the original data are $\mathbf{X}$, with N samples and m1 data points, after feature extraction by the model, we obtain the data matrix $\mathbf{Y}$with N samples and m2 data features. First, the peak-to-peak value for each sample is calculated

$$A_i = \max\left(X_{ij}\right)_i - \min\left(X_{ij}\right)_i \tag{3}$$

X_{ij} represents the *j*th data point of the *i*th sample. Then, to randomly select the positions of the activated features, *k* smaller A_i corresponding positions are identified

$$K = \operatorname{argmin}(A_i)_i, \quad K = \{k_1, k_2, \ldots, k_k\} \tag{4}$$

$$k \sim \text{Uniform}(a, b) \tag{5}$$

where *k* follows a uniform distribution, which ensures that the number of activated features is randomly selected. In this article, the values of *a* and *b* are 10 and 20, respectively. The data feature set corresponding to k is given as

$$Y_K = \{Y_{k_1}, Y_{k_2}, \ldots, Y_{k_k}\}. \tag{6}$$

The AAF mathematical formula is as follows:

$$Y = \begin{cases} \{Y_{k_1}, Y_{k_2}, \ldots, Y_{k_k}\} \times \zeta, & k \\ Y, & not\ k \end{cases} \tag{7}$$

ζ is a deflation factor taking the value of 5e-2. The dynamic activation function based on magnitude can be realized by (7).

The advantages of the AAF over traditional activation functions are as follows.

1) *Dynamic Adaptation:* The AAF can dynamically adjust the activation strength based on the amplitude characteristics of the input signals, allowing the model to better adapt to amplitude changes under time-varying speeds.
2) *Introduction of Randomness:* By randomly selecting the number of activated features (*k* value), the AAF enhances model robustness and generalizability, reducing the risk of overfitting.
3) *Preservation of Amplitude Information:* The AAF incorporates the amplitude information (peak-to-peak value) of the original signal into the activation process, ensuring that crucial amplitude-related information is preserved during feature extraction.

How AAF promotes the differentiation of data with different labels.

1) *Expansion of Feature Distances:* AAF expands the distance between different labeled data in the feature space by randomly activating subsets of features, thereby enhancing the ability to recognize different fault types.
2) *Increased Feature Dispersion:* By introducing a uniform distribution, the AAF enhances model randomness and dynamics, increasing feature dispersion. This reduces overlap among features of different categories, thereby improving the ability to distinguish labeled data of different fault types.

3) Kurtosis Maximum Mean Discrepancy (KMMD): Deep transfer learning models typically require an essential distance metric. The MMD is a classical metric used to quantify the distance between two random distributions. MMD minimizes the distance within the reproducing kernel Hilbert space $\mathcal{F}$, as shown in

$$\text{MMD}[\mathcal{F}, P, Q] = \sup_{f \in \mathcal{F}} \left(\mathbf{E}_{\mathbf{x}_s}[f(\boldsymbol{x}_s)] - \mathbf{E}_{\mathbf{x}_t}[f(\boldsymbol{x}_t)]\right). \tag{8}$$

In accordance with statistical theory [23], a biased[2] empirical estimate is employed for simplification (10). Consequently, the MMD loss function can be ascertained.

$$\mathcal{L}_{\mathrm{MMD}}[\mathcal{F}, P, Q] = \sup_{f \in \mathcal{F}} \left(\frac{1}{N} \sum_{i=1}^{N} f\left(\boldsymbol{x}_s^{(i)}\right) - \frac{1}{M} \sum_{i=1}^{M} f\left(\boldsymbol{x}_t^{(i)}\right) \right) \tag{9}$$

where $\mathcal{F}$ represents a class of functions, sup $(*)$ is the supremum, and $\mathbf{E}_*$ signifies the expectations derived from the domain distribution.

Subsequently, x^s and x^t are mapped into $\mathcal{F}$ constructed by a Gaussian kernel function, resulting in the mapping functions $\phi : \mathcal{X} \to \mathcal{H}$, and (9) can then be rewritten as

$$\begin{aligned} \mathrm{MMD}\left(\Phi_{x^s}\, \Phi_{x^t}\right) = & \frac{1}{m(m-1)} \sum_{i=1}^{m} \sum_{j \neq 1}^{m} k\left(\phi\left(x_i^s\right), \phi\left(x_i^s\right)\right) \\ & + \frac{1}{n(n-1)} \sum_{i=1}^{n} \sum_{j \neq 1}^{n} k\left(\phi\left(x_j^t\right), \phi\left(x_j^t\right)\right) \\ & - \frac{2}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} k\left(\phi\left(x_i^s\right), \phi\left(x_j^t\right)\right) \end{aligned} \tag{10}$$

where $\Phi_{x^s} = \{\phi(x_i^s)\}_{i=1}^m$, $\Phi_{x^t} = \{\phi(x_i^s)\}_{i=1}^n$.

Considering that the Gaussian kernel function can map data into a space of infinite dimensions, exhibiting strong nonlinear mapping capabilities, most kernel functions opt for the Gaussian kernel. However, the Gaussian kernel possesses a crucial adjustable parameter that influences the probability distribution after data mapping. Current methods often use the Euclidean distance to determine this parameter. However, for vibration data under time-varying speeds, even for data with the same label, the Euclidean distance can vary over time. This may lead to the instability of MMD performance, as confirmed by the ablation experiments below. To address this, we use kurtosis values independent of speed as dynamic parameters for the Gaussian kernel. Specifically, for both the source and target domains, we use the difference in their kurtosis values as the kernel parameter. When the kurtosis values differ significantly, the Gaussian kernel is smaller, indicating a greater distance between the two distributions. Conversely, when the kurtosis values of two distributions are close, the Gaussian kernel value is larger, reflecting a smaller distance for both distributions. The mathematical model is as follows:

$$K\left(x^s, x^t\right) = \exp\left(-\frac{\left\|x^s - x^t\right\|^2}{2\sigma^2}\right) \tag{11}$$

$$\begin{aligned} \sigma &= \sum_{i=1}^{n} \Delta_{\mathrm{kurt}} = \sum_{i=1}^{n} \left|\mathrm{kurt}_x - \mathrm{kurt}_y\right| \\ &= \sum_{i=1}^{n} \left| \frac{E\left[(X-\mu_x)^4\right]}{{\sigma_x}^4} - \frac{E\left[(Y-\mu_y)^4\right]}{{\sigma_y}^4} \right| \\ &= \frac{1}{n} \sum_{i=1}^{n} \left| \sum_{i=1}^{n} \frac{\left(x_{ij}^s - \frac{1}{d}\sum_{j=1}^{d} x_{ij}^s\right)^4}{\left(\frac{1}{d}\sum_{j=1}^{d}\left(x_{ij}^s - \frac{1}{d}\sum_{j=1}^{d} x_{ij}^s\right)^2\right)^2} \right. \\ &\quad \left. - \sum_{i=1}^{n} \frac{\left(x_{ij}^t - \frac{1}{d}\sum_{j=1}^{d} x_{ij}^t\right)^4}{\left(\frac{1}{d}\sum_{j=1}^{d}\left(x_{ij}^t - \frac{1}{d}\sum_{j=1}^{d} x_{ij}^t\right)^2\right)^2} \right| \end{aligned} \tag{12}$$

where n is the number of samples and d is the length of the feature vector. x_{ij} represents the value of the ith sample in the jth dimension. kurt_x represents the kurtosis corresponding to the source domain, E denotes the expectation, and μ_x and σ_x denote the mean and variance of X, respectively. The expression of the KMMD is as follows:

$$\begin{aligned} \mathcal{L}_{\mathrm{KMMD}} &= \mathrm{KMMD}\left(\Phi_{x^s}, \Phi_{x^t}\right) \\ &= \frac{1}{m(m-1)} \sum_{i=1}^{m} \sum_{j \neq i}^{m} K\left(\phi\left(x_i^s\right), \phi\left(x_j^s\right)\right) \\ &\quad + \frac{1}{n(n-1)} \sum_{i=1}^{n} \sum_{j \neq i}^{n} K\left(\phi\left(x_i^t\right), \phi\left(x_j^t\right)\right) \\ &\quad - \frac{2}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} K\left(\phi\left(x_i^s\right), \phi\left(x_j^t\right)\right) \end{aligned} \tag{13}$$

where K is a kurtosis-based Gaussian kernel, as shown in (11).

Combining (11)–(13), it can be observed that the Gaussian kernel function $K(x^s, x^t)$varies with kurtosis kurt_x and kurt_yvariance. As the difference σ in kurtosis increases, the value of the Gaussian kernel function $K(x^s, x^t)$ decreases, indicating an increase in the difference between the distributions of the two datasets. Simultaneously, the difference function $\mathcal{L}_{\mathrm{KMMD}}$ between the data will automatically adjust, thereby reducing the distribution distance between different domain data and achieving domain adaptation functionality.

The advantages of the KMMD compared to the traditional MMD.

1) *Adaptive Adjustment of Gaussian Kernel Parameters:* Traditional MMD methods typically use fixed Gaussian kernel parameters, which cannot adapt to changes in data distribution under time-varying speeds. KMMD, however, dynamically adjusts the Gaussian kernel parameters, enabling better capture of distribution differences between different domains and thereby enhancing the model's domain adaptation capability.
2) *Parameter Selection Based on Physical Information:* The KMMD uses the kurtosis values of vibration signals as kernel parameters. Kurtosis values can reflect the amplitude and shape characteristics of signals, providing stability and robustness. Therefore, the KMMD can more accurately measure distribution differences between the source and target domains, improving the fault diagnosis performance.
3) *Improved Domain Adaptation Performance:* By introducing kurtosis values as dynamic parameters, the KMMD can better handle changes in data distribution at varying speeds, reduce distribution bias during domain migration, and thereby enhance generalizability and stability.
4) *Dynamic KL Divergence Weights (DKLDW):* When there is a significant difference in distribution between two domains,

such as in cross-device fault diagnosis, aligning data distributions often requires adversarial domain metrics to encourage the model to find similar features. It is mathematically modeled as follows:

$$\mathcal{L}_{\text{adv}} = -\frac{1}{n}\sum_{i=1}^{n} \left[d_i \log \frac{1}{G_d\left(G_f\left(x_i\right)\right)} + \left(1-d_i\right) \log \frac{1}{G_d\left(G_f\left(x_i\right)\right)} \right]. \tag{14}$$

Let G_d be the domain classifier, G_f be the feature extractor, x_i be the input data, d_i be the corresponding domain labels, $d_i = 0$ denote the source domain, and $d_i = 1$ denote the target domain. The domain adversarial loss function is a binary cross-entropy loss function. The smaller its value is, the better the classification performance of G_d and the poorer the adversarial performance of G_f. Conversely, the larger its value is, the worse the classification performance of G_d and the better the adversarial performance of G_f.

Combining (13) and (14), we obtain

$$\mathcal{L}_{\text{domain}} = \mathcal{L}_{\text{KMMD}} + \mathcal{L}_{\text{adv}}. \tag{15}$$

Most research does not consider the interactive relationship between $\mathcal{L}_{\text{KMMD}}$ and $\mathcal{L}_{\text{adv}}$. We have observed that there is no effective dynamic weighting term for balancing marginal and adversarial distributions. The widely applied strategy is based on epoch-based dynamic weighting. However, in cross-device tasks under time-varying speeds, where the degree of domain shift changes over time, this strategy may become ineffective. Therefore, we propose a dynamic KL divergence weight update strategy, termed DKLDW.

First, we compute the feature vectors after x^s and x^t through G_f. The corresponding KL divergence is calculated as follows:

$$\text{KL}\left(P \,\|Q\right) = \sum_{x\in X} P\left(G_f\left(x^s\right)\right) \log \frac{P\left(G_f\left(x^s\right)\right)}{Q\left(G_f\left(x^t\right)\right)}. \tag{16}$$

P and Q are the distributions of $G_f(x^s)$and $G_f(x^t)$, respectively. Second, let $\mu = \text{KL}(P \parallel Q)$ be the dynamic weighting factor, which is introduced into (16) to obtain as

$$\mathcal{L}_{\text{domain}} = (1-\mu)\mathcal{L}_{\text{KMMD}} + \mu\mathcal{L}_{\text{adv}}. \tag{17}$$

DKLDW mechanism and its impacts the following.

1) *Dynamic Adjustment Strategy:* According to (17), as the feature vectors correspond to larger KL dispersion, the entropy-based data similarity is lower, and larger weights make the model more focused on the discriminative properties with respect to the output features. Conversely, as μ decreases, the model places more emphasis on aligning the data in the Gaussian kernel space. Through this dynamic linkage strategy, the distance and the adversarial domain metric are jointly used for training, thus improving the training stability and accuracy of the model, as confirmed in the experimental section.
2) *Impact on Model Convergence Performance:* By dynamically adjusting weights μ, DKLDW finds a balance between feature alignment and discriminative preservation. This equilibrium accelerates the model's convergence to effective feature representations, thereby improving training efficiency.
3) *Impact on Model Stability:* The DKLDW adapts to domain shifts induced by varying speeds, mitigating potential instability from fixed weights. This reduction in training oscillations enhances overall stability.
4) *Advantages over Traditional Methods:* Epoch-based dynamic weight strategies struggle with domain shifts at varying speeds. Using real-time computed KL divergence, DKLDW more accurately reflects current domain differences, making more suitable adjustments.

B. Training Process

The details of the proposed DMsrTTLN with DKLDW are shown in Fig. 1, and its overall optimization objective consists of two parts: supervised learning and domain adaptive learning based on DKLDW. The effect of the cross-entropy loss function in supervised learning is well documented

$$\mathcal{L}_C = -\frac{1}{N}\sum_{n=1}^{N}\sum_{c=1}^{C} y_c^{(n)} \log \frac{\exp\left(\hat{y}_c^{(n)}\right)}{\sum_{\tilde{c}=1}^{C}\exp\left(\hat{y}_{\tilde{c}}^{(n)}\right)} \tag{18}$$

where N represents the quantity of samples and C denotes the sample category. $y_c^{(n)}$ is a symbolic function. When the genuine category of sample n corresponds to c, a value of 1 is assigned; otherwise, a value of 0 is assigned. Additionally, $\hat{y}_c^{(n)}$ signifies the characteristic value of the nth sample in FC2 associated with the c label.

Combined with (17), the overall loss is shown as

$$\mathcal{L}_{\text{All}} = \mathcal{L}_C + \mathcal{L}_{\text{domain}} = \mathcal{L}_C + (1-\mu)\mathcal{L}_{\text{KMMD}} + \mu\mathcal{L}_{\text{adv}}. \tag{19}$$

Assuming that θ_f θ_d and θ_c are the parameters of G_f, G_d and classifier G_c, the parameter update formula is as follows:

$$\theta_f = \theta_f - \lambda\frac{\partial\mathcal{L}_{\text{All}}}{\partial\theta_f} \tag{20}$$

$$\theta_d = \theta_d - \lambda\frac{\partial\mathcal{L}_{\text{All}}}{\partial\theta_d} \tag{21}$$

$$\theta_c = \theta_c - \lambda\frac{\partial\mathcal{L}_{\text{All}}}{\partial\theta_c} \tag{22}$$

where ∂ is the partial derivative formula and λ is the learning rate. The training procedure of the proposed DMsrTTLN for cross-device tasks is described in Algorithm 1.

IV. Experimental Study

A. Dataset Descriptions

1) Dataset A: This dataset was collected from the SpectraQuest machinery fault simulator (MFS-PK5M) experimental platform at the University of Ottawa [24]. The ac drive powers the motor rotation, and the acceleration sensor (ICP accelerometer, Model 623C01) collects vibration signals from directly above the tested bearing. There are five bearing health states: normal (NC); outer race fault (OF); inner race fault (IF); rolling element fault (BF); and combined fault (IOBF). The speed varies over time, either increasing or decreasing.

Algorithm 1: The Training Process for DMsrTTLN on Case2.

Input:

-Building the basic skeleton, based on ResNetl8.

-Initial feature extractor G_f , discriminator G_p and a classifier G_c.

-The source domain $D_T^{m\_t} = \{(\boldsymbol{x}_t^{(i)}, \boldsymbol{v}_t^{(i)})\}_{i=1}^{M}$ and the target domain $D_T^{m\_t} = \{(\boldsymbol{x}_t^{(i)}, \boldsymbol{v}_t^{(i)})\}_{i=1}^{M}$.

-learning rate, batchsize, number of iterations, and Radam optimizer.

1: **for** number of iterations **do**

3: Calculate the classification loss using (18):

4: Calculate the domain adaptation loss with (17);

5: Obtain the overall objective with (19);

6: Train and update model parameters with (20)–(22);

7: eud for

Output: The trained diagnosis model.

TABLE I
THREE BEARING DATASETS

Dataset	Equipment	Condition type	Speed (r/min)	Sampling frequency (kHz)
Dataset A	MFS-PK5M ER16 K	NC, OF, IF, RF, IOBF (5)	918~1458 1499~1199 748~448 846~1476 782~1676 0~1451	200 kHz
Dataset B	HUST bearing test bench NSK6206	NC, OF, IF, RF, IOF, IBF, OBF (7)	91~1459 0~1490 0~1449 0~1492 0~1401 0~1451	51.2 kHz
Dataset C	Spectra Quest NSK6203	NC, OF, IF (3)	3000~0 3000~0 3000~0	25.6 kHz

2) Dataset B: This dataset comes from the bearing test platform at Hanoi University of Science and Technology [25]. The vibration signals from the bearings are collected through the NI-9234 analog-to-digital conversion module. The tested bearings have a total of seven states: NC; OF; IF; and BF, inner race outer race fault (IOF), inner race rolling element fault (IBF), and outer race rolling element fault (OBF). The speed increases from 0 to approximately 1500 r/min.

3) Dataset C: This dataset comes from the SpectraQuest (VSQ) test platform at Xi'an Jiaotong University [26]. The CoCo80 device collects vibration signals from the NSK6203 tested bearing under three health states, namely, NC, OF, and IF. The fault sizes are 12 mm^2 and 2 mm. The speed decreases from 3000 r/min to 0. Details of the data are given in Table I. The time-varying speed conditions of the three datasets mentioned above are all achieved by changing the speed.

B. Diagnosis Tasks and Implementation Details

In the three time-varying speed fault datasets mentioned above, two different fault diagnosis cases are conducted under two scenarios: cases 1 and 2. The hyperparameter settings include a batch size of 128, 100 iterations, an L2 weight decay of 5e-1, and the Radam optimizer with a learning rate of 2e-3. The sample lengths, quantities, and test set proportions are given in Table II. The experimental results represent the average of five runs. Three performance metrics, namely, the accuracy (ACC), F1-score (F1), and average area under the receiver operating characteristic curve (AUC) [27], were utilized to assess the testing outcomes of the proposed method and the comparative methods.

TABLE II
DIAGNOSTIC TASKS AND IMPLEMENTATION DETAILS

Tasks	Equipment	Condition type	Detailed description
Case 1: Time-varying speed fault diagnosis with the same equipment	Dataset A	NC, OF, IF, RF, IOBF (5)	500 samples for each health state with 3000 data points; 50% of test samples
	Dataset B	NC, OF, IF, RF, IOF, IBF, OBF (7)	
	Dataset C	NC, OF, IF (3)	
Case 2: Time-varying speed cross-device fault diagnosis with different equipment	A→B, A→C, B→A, B→C, C→A, C→B	NC, OF, IF (3)	500 samples for each health state with 3000 data points; 50% of test samples

TABLE III
PERFORMANCE STATISTICS RESULTS

Method	ACC	F1	AUC	Time (s)
DMsrTTLN (A)	100±0	1±0	1±0	91.03±0.44
DMsrTTLN (B)	98.09±0.45	0.98±0	0.99±0	193.47±100.95
DMsrTTLN (C)	99.81±0.29	1±0	1±0	56.24±1.89
N/O (A)	47.5±5.16	0.4±0.08	0.67±0.03	89.77±0.93
N/O (B)	72.39±13.03	0.72±0.13	0.84±0.08	201.24±75.66
N/O (C)	79.89±14.46	0.76±0.19	0.85±0.11	53.37±0.82
TICNN (A)	68.27±7.13	0.65±0.08	0.8±0.05	46.26±0.49
TICNN (B)	24.65±3.85	0.17±0.06	0.56±0.02	59.87±1.51
TICNN (C)	71.97±12.42	0.68±0.17	0.79±0.09	28.79±0.79
DTCNN (A)	53.23±6.68	0.47±0.06	0.71±0.04	41.9±0.56
DTCNN (B)	20.91±4.47	0.15±0.04	0.54±0.03	54.86±0.74
DTCNN (C)	71.63±9.58	0.69±0.14	0.79±0.07	28.08±0.62

C. Case1: Time-varying Speed Fault Diagnosis With the Same Equipment

Three relevant methods are employed for performance comparison with the proposed method. N/O represents the model without the MSR module, TICNN is the baseline comparison model for domain adaptation under the same device, and DTCNN is the model under variable-speed conditions [27]. Model details can be found here. (https://github.com/John-520/DMsrTTLN). The statistics for the three performance metrics and training times are given in Table III. Clearly, the DMsrTTLN demonstrates a significant advantage in diagnostic effectiveness, while the computation time is constrained by the number of weight parameters, resulting in an average time higher than that of the TICNN and DTCNN. The accuracy of the N/O method is lower than that of DMsrTTLN, directly indicating the effectiveness of the proposed MSR module.

D. Case2: Time-Varying Speed Cross-Device Fault Diagnosis With Different Equipment

1) Compared Approaches: To fully explore the overall performance of the DMsrTTLN in cross-device tasks, it is compared with several state-of-the-art domain adaptive models. The details are given as follows.

a) *Maximum mean square discrepancy (MMSD) [28]:* This is a novel discrepancy metric function proposed by Qian et al., which comprehensively considers the mean and variance information, enhancing domain confusion. The parameters used in this article are consistent with those used in the original paper.

b) *Joint distribution adaptation (JDA) [29]:* The JDA model, proposed by Han et al., is a joint distribution adaptation

model. The difference lies in JDA's use of static joint marginal and conditional distributions. In the following sections, it is applied to cross-device fault diagnosis tasks. The parameters and model configuration remain consistent with those in the original paper.

c) *DCTLN [7]:* The DCTLN was proposed by Guo et al., with high recognition and numerous citations. Similarly, the model and hyperparameters remain consistent with those in the original paper.

d) *Deep adversarial subdomain adaptation network (DASAN) [21]:* DASAN is a model that considers subdomain alignment and has been validated for cross-device diagnosis tasks. The model settings are consistent with those in the original paper.

e) *Deep discriminative transfer learning network (DDTLN) [30]:* This domain adaptation model, which was newly proposed for cross-device tasks, enhances the mechanism for aligning conditional distributions.

f) *Deep dynamic adaptive transfer network (DDATN) [20]:* This model balances the marginal and conditional distributions with a dynamic weighting factor, but requires label information from the target domain. For fairness, the experiments in this article are all conducted in an unsupervised manner in the target domain.

In addition to the latest methods mentioned above, we also employ five ablation versions corresponding to the proposed method. All methods are conducted under the same baseline.

1) *Version_1:* A variant of the proposed method, removing the MSR module while keeping other configurations consistent with DMsrTTLN.
2) *Version_2:* Compared to the proposed DMsrTTLN, the AAF is removed.
3) *Version_3:* Compared to the DMsrTTLN, the KMMD is replaced with the MMD to test the performance of the KMMD.
4) *Version_4:* The dynamic KL divergence weight is removed to test the effect of the proposed dynamic weighting.
5) *Version_5:* The dynamic KL divergence weight is removed, and a popular epoch-based dynamic weighting is used for testing.

2) Results: Table IV gives the statistical results of all methods on three performance indicators.

Among the six advanced comparative methods, JDA has an average diagnostic accuracy of 81.35%, showing the best performance, while the average accuracy for the other five methods is approximately 70%. In addition, version_1 and version_4 represent ablation experiments for MSR and DKLDW, respectively. Comparative analysis revealed that the diagnostic performance improvement of these two modules was most evident, confirming the effectiveness of the proposed modules. This is also the main reason for naming the method. Overall, the proposed DMsrTTLN performs the best in cross-device tasks under time-varying speed conditions, achieving an average diagnostic accuracy of 98.53%.

3) Convergence Curves and Feature Visualization: To test the convergence performance of all methods, we take the A→C task as an example. The ACC on the test set for each method is shown in Fig. 2. The proposed DMsrTTLN model exhibits the best convergence performance. The ACC of JDA fluctuates by approximately 65%, and the other methods also show varying degrees of fluctuation, indicating poor training stability. Fig. 3 presents the feature visualization results for each method on the test set. The feature distribution labels for version_1 do not correspond, indicating a lack of ability of the model to recognize time-varying features, which indirectly reflects the regularization effect of MSR. Overall, the proposed model demonstrates superior clustering performance.

TABLE IV
PERFORMANCE STATISTICS RESULTS

Tasks	Metrics	A→B	A→C	B→A	B→C	C→A	C→B	Average
MMSD	Acc	94.51±0.98	86.05±7.11	40.00±14.91	75.25±11.14	33.33±0.00	72.19±2.96	66.89
	F1	0.94±0.01	0.85±0.08	0.29±0.15	0.72±0.13	0.18±0.02	0.66±0.05	0.61
	AUC	0.96±0.01	0.90±0.05	0.55±0.11	0.81±0.08	0.50±0.00	0.79±0.02	0.75
JDA	Acc	92.51±7.28	89.44±5.25	69.12±5.49	84.99±17.07	59.97±14.89	92.08±12.12	81.35
	F1	0.92±0.08	0.89±0.06	0.60±0.09	0.81±0.23	0.48±0.17	0.91±0.14	0.77
	AUC	0.94±0.05	0.92±0.04	0.77±0.04	0.89±0.13	0.70±0.11	0.94±0.09	0.86
DCTLN	Acc	93.09±5.17	79.79±16.71	46.67±18.26	76.03±3.30	39.15±13.00	66.67±0.00	66.90
	F1	0.93±0.05	0.75±0.22	0.32±0.21	0.72±0.06	0.24±0.16	0.56±0.00	0.59
	AUC	0.95±0.04	0.85±0.13	0.60±0.14	0.82±0.02	0.54±0.10	0.75±0.00	0.75
DASAN	Acc	94.93±1.11	96.11±3.70	46.67±18.26	88.99±7.08	33.33±0.00	66.64±0.06	71.11
	F1	0.95±0.01	0.96±0.04	0.32±0.21	0.88±0.08	0.17±0.00	0.56±0.00	0.64
	AUC	0.96±0.01	0.97±0.03	0.60±0.14	0.92±0.05	0.50±0.00	0.75±0.00	0.78
DDTLN	Acc	95.23±1.85	68.56±14.49	64.85±2.62	64.64±1.33	33.33±0.00	66.72±0.12	65.56
	F1	0.95±0.02	0.64±0.18	0.54±0.02	0.54±0.01	0.18±0.02	0.55±0.00	0.57
	AUC	0.96±0.01	0.76±0.11	0.74±0.02	0.73±0.01	0.50±0.00	0.75±0.00	0.74
DDATN	Acc	81.15±13.59	72.64±10.39	33.33±0.00	87.57±7.83	33.36±0.06	68.56±1.31	62.77
	F1	0.79±0.17	0.67±0.15	0.22±0.00	0.87±0.09	0.17±0.00	0.60±0.03	0.55
	AUC	0.86±0.10	0.79±0.08	0.50±0.00	0.91±0.06	0.50±0.00	0.76±0.01	0.72
Version_1	Acc	68.72±12.35	62.48±16.59	63.44±7.22	64.80±21.87	65.73±2.24	67.20±19.55	65.40
	F1	0.60±0.17	0.53±0.21	0.53±0.06	0.57±0.27	0.54±0.03	0.61±0.24	0.56
	AUC	0.76±0.09	0.72±0.13	0.73±0.05	0.74±0.17	0.74±0.02	0.76±0.14	0.74
Version_2	Acc	99.76±0.22	98.45±2.22	97.41±5.78	100.00±0.00	86.72±13.73	97.17±4.31	96.59
	F1	1.00±0.00	0.98±0.03	0.97±0.06	1.00±0.00	0.85±0.17	0.97±0.04	0.96
	AUC	1.00±0.00	0.99±0.02	0.98±0.04	1.00±0.00	0.90±0.10	0.98±0.03	0.98
Version_3	Acc	98.96±2.18	99.39±1.04	93.87±12.63	99.92±0.18	88.80±13.75	98.77±2.45	96.62
	F1	0.99±0.02	0.99±0.01	0.93±0.14	1.00±0.00	0.86±0.18	0.99±0.02	0.96
	AUC	0.99±0.02	1.00±0.01	0.95±0.10	1.00±0.00	0.91±0.10	0.99±0.02	0.97
Version_4	Acc	98.37±2.95	98.05±2.63	80.96±16.35	98.61±2.74	77.68±12.09	93.09±9.18	91.13
	F1	0.98±0.03	0.98±0.03	0.76±0.22	0.98±0.03	0.72±0.16	0.93±0.09	0.89
	AUC	0.99±0.02	0.99±0.02	0.86±0.12	0.99±0.02	0.83±0.09	0.95±0.06	0.94
Version_5	Acc	99.20±1.79	99.76±0.54	99.01±2.21	100.00±0.00	88.08±14.98	99.01±1.45	97.51
	F1	0.99±0.02	1.00±0.00	0.99±0.02	1.00±0.00	0.87±0.17	0.99±0.02	0.97
	AUC	0.99±0.01	1.00±0.00	0.99±0.01	1.00±0.00	0.91±0.11	0.99±0.01	0.98
DMsrTTLN (Proposed)	Acc	99.89±0.24	98.93±1.79	99.89±0.17	100.00±0.00	92.48±13.45	100.00±0.00	98.53
	F1	1.00±0.00	0.99±0.02	1.00±0.00	1.00±0.00	0.91±0.17	1.00±0.00	0.98
	AUC	1.00±0.00	0.99±0.01	1.00±0.00	1.00±0.00	0.95±0.10	1.00±0.00	0.99

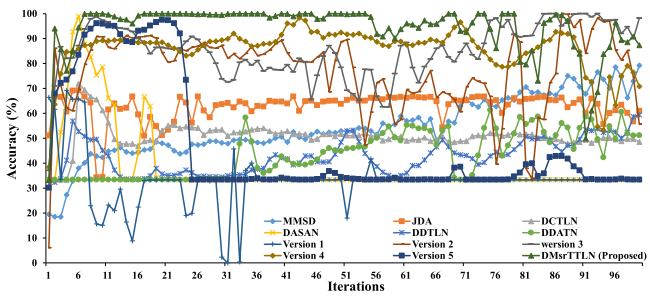

Fig. 2. Accuracy convergence curves for each method.

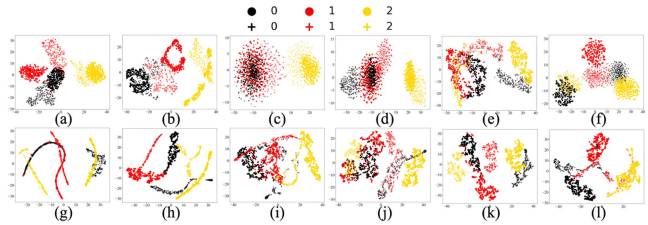

Fig. 3. t-SNE visualization results for task A→C. (a)–(f) Six advanced methods. (g)–(k) Five ablation methods. (l) Proposed DMsrTTLN model.

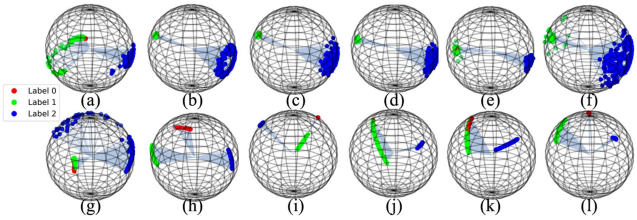

Fig. 4. Visualization results of unit ball surface features. (a)–(f) Six advanced methods. (g)–(k) Five ablation methods. (l) DMsrTTLN model.

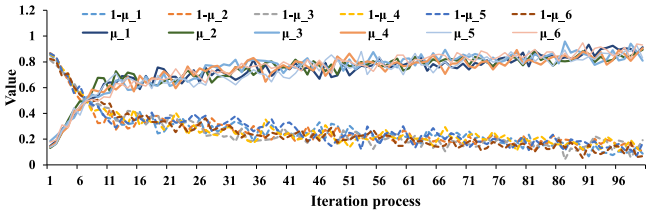

Fig. 5. Dynamic weight distribution of the DMsrTTLN training process.

4) Role of Multilayer Sparse Regularization: To investigate the role played by the proposed MSR, we conducted an analysis of the input features to the last linear layer for the 12 models in the C→B task. The input feature data for all models are uniformly scaled to the surface of a unit sphere, as shown in Fig. 4.

Fig. 4(a)–(f) depicts the results of six advanced methods. These methods exhibit low feature clustering on the sphere, indicating limitations in feature extraction and domain adaptation. Fig. 4(g) shows the results of the version_1 model without using MSR, demonstrating poorer feature clustering due to the absence of MSR. Fig. 4(h)–(l) correspond to methods that utilized MSR. These figures clearly demonstrate the regularization effect of MSR, showing higher feature clustering and indicating that MSR effectively enhances feature discriminability and domain adaptation capability. Fig. 4(l) displays the results of our proposed DMsrTTLN model, which achieves the best clustering effect, highlighting its superior performance in cross-device fault diagnosis tasks under time-varying speeds.

5) Role of Dynamic KL Divergence Weights: To further investigate the role of the DKLDW in model training, the dynamic weight factors and in the six migration tasks are visualized, as shown in Fig. 5. The weight distribution for each task is distinct, displaying dynamic variations. Overall, as the number of iterations increases, $1-\mu$gradually decreases, while μ increases. This phenomenon has significant implications for model training: a decrease in $1-\mu$ indicates that the model gradually reduces its focus on the marginal distribution differences between the source and target domains. An increase in μ signifies that the model increasingly focuses on differentiated features across devices, helping alleviate the impact of domain shifts.

This process can be explained by the fact that a smaller adversarial loss would result in lower feature similarity, leading to an increase in the KL divergence. This positive feedback to μprompts the model to learn in the direction of higher feature similarity, ultimately mitigating the impact of domain shift. This aligns with the theoretical analysis in the foregoing section. DKLDW enables the model to adaptively adjust weights according to task characteristics during training, thereby enhancing the fault diagnosis performance under cross-device and time-varying speed conditions.

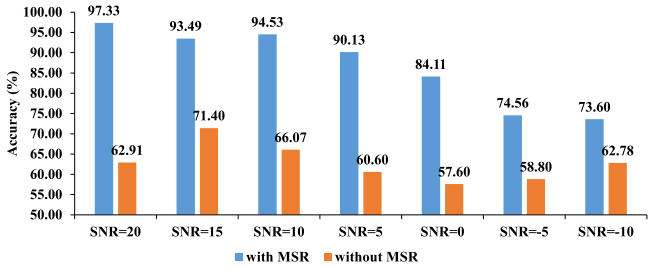

Fig. 6. Results of the noise resistance analysis for DMsrTTLN.

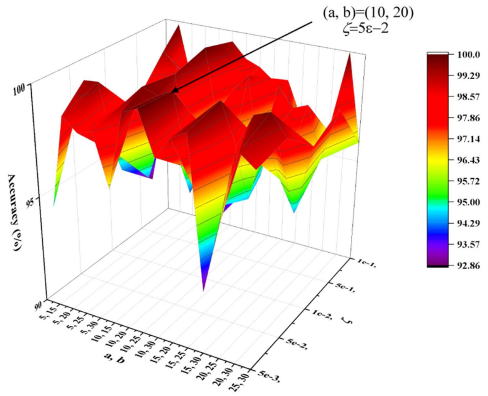

Fig. 7. Results of AAF parameter analysis.

6) Noise Resistance Analysis of DMsrTTLN: Early fault features are often weak and can be easily masked by noise or other nonfault-related signaling components. To investigate whether MSR is prone to neglecting fault features under noise interference, we conducted noise resistance experiments on DMsrTTLN with and without MSR. The experimental results are shown in Fig. 6. Regardless of the presence of MSR, as the noise energy increases, the diagnostic performance of the model decreases. With the MSR, the DMsrTTLN achieves a diagnostic accuracy of less than 90% when the signal-to-noise ratio is less than or equal to 0. This indicates that in the presence of strong noise, the MSR algorithm may also neglect early fault features. However, compared to DMsrTTLN without an MSR, DMsrTTLN with an MSR demonstrates better diagnostic performance, indicating that an MSR exhibits a certain degree of noise resistance.

7) Parameter Analysis of AAF: According to (7), the AAF is a type of data mapping operation algorithm. In this section, we conduct a parameter sensitivity analysis of the AAF. First, we determine the range of the scaling parameter ζ to be [5e-3, 5e-2, 1e-2, 5e-1, 1e-1]. The range of parameters a and b is set to [5, 10, 15, 20, 25, 30], where a should be less than b. Thus, there are 15 possible combinations for (a, b). For case 2, we used the grid search method to determine the parameter combination corresponding to the maximum diagnostic accuracy. The results of the grid search are shown in Fig. 7. It can be observed that different parameter values correspond to different diagnostic accuracies. Overall, the AAF is not very sensitive to parameter values, indicating the robustness of the AAF. Nevertheless, the AAF can achieve maximized model performance through simple

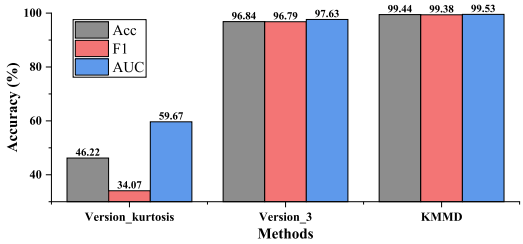

Fig. 8. Comparison of the diagnostic effects of different methods.

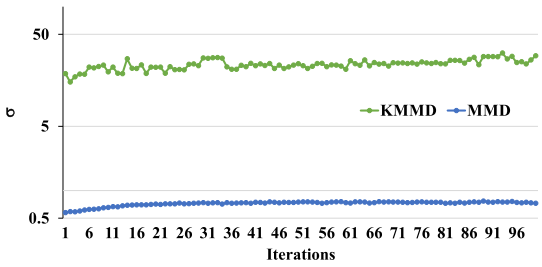

Fig. 9. The Gaussian kernel value σ change graph of the KMMD and MMD.

parameter optimization strategies, demonstrating its practical value.

8) Advantage Analysis of the KMMD: To further explore the internal parameter variations and advantages of the proposed KMMD algorithm, for case 2, we compared the classification models based on kurtosis values and those based on MMD. The results are shown in Fig. 8. The classifier of DMsrTTLN is considered a classification model based on kurtosis values (Version_kurtosis), with the input data being the kurtosis values corresponding to the original data. Version_3 in the comparative methods is regarded as the model based on the MMD. The diagnostic performance of the KMMD is superior. Furthermore, we visualized the variations in the kernel functions inside the KMMD and MMD for the A→B task in case 2, as shown in Fig. 9. To highlight the relative change trend of the data, the ordinate of Fig. 9 is a logarithmic coordinate with a base of 10. Clearly, the parameter σ variations corresponding to the KMMD and MMD are different. A larger value of σ indicates smaller domain differences in the KMMD. This indicates that under the influence of the KMMD, the feature differences between the source domain and the target domain data are reduced.

9) Comparison With Related Work on Time-Varying Speed: To demonstrate the superiority of the proposed method in transfer learning under time-varying speed conditions, we compare it with the current mainstream speed elimination algorithm, order tracking (OT). We constructed two fault diagnosis models based on OT preprocessing, OT-JDA and OT-DASAN, and tested them in the case 2 scenario. Fig. 10 shows that the proposed method achieves the best diagnostic performance, indicating its superiority in cross-device fault diagnosis under time-varying speed conditions.

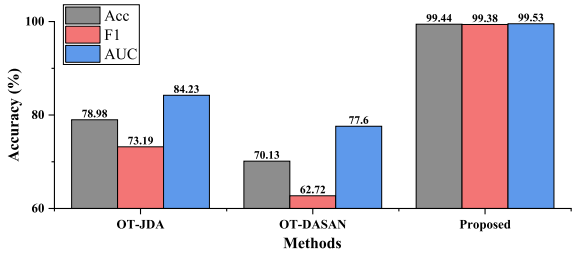

Fig. 10. Comparison of the effects of different methods.

V. Conclusion

In this article, a novel time-varying speed mechanical fault diagnosis method, named DMsrTTLN with DKLDW, was proposed. The MSR in DMsrTTLN can eliminate interference from speed fluctuations, thereby significantly improving the model's performance. In domain adaptive tasks across devices, the developed KMMD metric can automatically select appropriate kernel functions from data, thereby unleashing the potential of MMD. Additionally, the proposed DKLDW demonstrates excellent capabilities in balancing distance and adversarial domain metrics, playing a crucial role in enhancing the training stability and diagnostic accuracy of the model. The experimental results also demonstrate the high performance of the proposed fault diagnosis method. In the future, we believe that models based on the DMsrTTLN can play a greater role in time-varying speed fault diagnosis tasks.

Although the proposed DMsrTTLN method has demonstrated the best diagnostic performance in various time-varying speed scenarios defined in this article, its ability to identify small and unbalanced samples or unknown fault types may be limited, leading to a decrease in diagnostic performance. Subsequent research will focus on improving the model structure, exploring few-shot learning methods, and expanding the generalization ability to fault types.

References

[1] K. Feng, J. C. Ji, Y. Zhang, Q. Ni, Z. Liu, and M. Beer, "Digital twin-driven intelligent assessment of gear surface degradation," *Mech. Syst. Sig. Process.*, vol. 186, 2023, Art. no. 109896.

[2] Q. Ni, J. C. Ji, K. Feng, Y. Zhang, D. Lin, and J. Zheng, "Data-driven bearing health management using a novel multi-scale fused feature and gated recurrent unit," *Rel. Eng. System Saf.*, vol. 242, 2024, Art. no. 109753.

[3] H. Chen et al., "M^{3}FuNet: An unsupervised multivariate feature fusion network for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, pp. 1–15, 2024, Art. no. 5513015, doi: 10.1109/TGRS.2024.3380087.

[4] S. Pal, A. Roy, P. Shivakumara, and U. Pal, "Adapting a swin transformer for license plate number and text detection in drone images," *Artif. Intell. Appl.*, vol. 1, no. 3, pp. 145–154, 2023.

[5] R. G. M. Helali, "An exploratory study of factors affecting research productivity in higher educational institutes using regression and deep learning techniques," *Artif. Intell. Appl.*, 2023, doi: 10.47852/bonviewAIA3202660.

[6] T. Akande, O. Alabi, and S. Ajagbe, "A deep learning-based CAE approach for simulating 3D vehicle wheels under real-world conditions," *Artif. Intell. Appl.*, 2024, doi: 10.47852/bonviewAIA42021882.

[7] L. Guo, Y. Lei, S. Xing, T. Yan, and N. Li, "Deep convolutional transfer learning network: A new method for intelligent fault diagnosis of machines with unlabeled data," *IEEE Trans. Ind. Electron.*, vol. 66, no. 9, pp. 7316–7325, Sep. 2019.

[8] D. Liu, L. Cui, and H. Wang, "Rotating machinery fault diagnosis under time-varying speeds: A review," *IEEE Sens. J.*, vol. 23, no. 24, pp. 29969–29990, Dec. 2023.

[9] Q. Ni, J. C. Ji, B. Halkon, K. Feng, and A. K. Nandi, "Physics-Informed Residual Network (PIResNet) for rolling element bearing fault diagnostics," *Mech. Syst. Sig. Process.*, vol. 200, 2023, Art. no. 110544.
[10] C. He, H. Shi, X. Liu, and J. Li, "Interpretable physics-informed domain adaptation paradigm for cross-machine transfer diagnosis," *Knowl.-Based Syst.*, vol. 288, 2024, Art. no. 111499.
[11] F. Lu et al., "Towards multi-scene learning: A novel cross-domain adaptation model based on sparse filter for traction motor bearing fault diagnosis in high-speed EMU," *Adv. Eng. Inf.*, vol. 60, 2024, Art. no. 102536.
[12] Y. Xiao, H. Shao, J. Wang, S. Yan, and B. Liu, "Bayesian variational transformer: A generalizable model for rotating machinery fault diagnosis," *Mech. Syst. Sig. Process.*, vol. 207, 2024, Art. no. 110936.
[13] S. Shao, S. McAleer, R. Yan, and P. Baldi, "Highly accurate machine fault diagnosis using deep transfer learning," *IEEE Trans. Ind. Inf.*, vol. 15, no. 4, pp. 2446–2455, Apr. 2019.
[14] B. Yang, S. Xu, Y. Lei, C.-G. Lee, E. Stewart, and C. Roberts, "Multi-source transfer learning network to complement knowledge for intelligent diagnosis of machines with unseen faults," *Mech. Syst. Sig. Process.*, vol. 162, 2022, Art. no. 108095.
[15] B. Yang, Y. Lei, S. Xu, and C.-G. Lee, "An optimal transport-embedded similarity measure for diagnostic knowledge transferability analytics across machines," *IEEE Trans. Ind. Electron.*, vol. 69, no. 7, pp. 7372–7382, Jul. 2022.
[16] B. Yang, Y. Lei, X. Li, and C. Roberts, "Deep targeted transfer learning along designable adaptation trajectory for fault diagnosis across different machines," *IEEE Trans. Ind. Electron.*, vol. 70, no. 9, pp. 9463–9473, Sep. 2023.
[17] Y. Chang, J. Chen, Q. Chen, S. Liu, and Z. Zhou, "CFs-focused intelligent diagnosis scheme via alternative kernels networks with soft squeeze-and-excitation attention for fast-precise fault detection under slow & sharp speed variations," *Knowl.-Based Syst.*, vol. 239, 2022, Art. no. 108026.
[18] P. Liang, L. Xu, H. Shuai, X. Yuan, B. Wang, and L. Zhang, "Semisupervised subdomain adaptation graph convolutional network for fault transfer diagnosis of rotating machinery under time-varying speeds," *IEEE/ASME Trans. Mechatron.*, vol. 29, no. 1, pp. 730–741, Feb. 2023.
[19] J. Chen, J. Chen, Z. Chen, S. Liu, and S. He, "Hybrid augmented network with balance domain window for few-shot fault diagnosis under sharp speed variation," *Mech. Syst. Sig. Process.*, vol. 207, 2024, Art. no. 110944.
[20] Y. Zhou, Y. Dong, H. Zhou, and G. Tang, "Deep dynamic adaptive transfer network for rolling bearing fault diagnosis with considering cross-machine instance," *IEEE Trans. Instrum. Meas.*, vol. 70, 2021, Art. no. 3525211.
[21] Y. Liu, Y. Wang, T. W. S. Chow, and B. Li, "Deep adversarial subdomain adaptation Network for intelligent fault diagnosis," *IEEE Trans. Ind. Inf.*, vol. 18, no. 9, pp. 6038–6046, Sep. 2022, doi: 10.1109/TII.2022.3141783.
[22] J. Ngiam, Z. Chen, S. Bhaskar, P. Koh, and A. Ng, "Sparse filtering," in *Advances in Neural Information Processing Systems*, vol. 24, J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Q. Weinberger, Eds., 2011, Red Hook, NY, USA: Curran Assoc., pp. 1125–1133. [Online]. Available: https://proceedings.neurips.cc/paper/2011/file/192fc044e74dffea144f9ac5dc9f3395-Paper.pdf
[23] A. Müller, "Integral probability metrics and their generating classes of functions," *Adv. Appl. Probability*, vol. 29, no. 2, pp. 429–443, 1997.
[24] H. Huang and N. Baddour, "Bearing vibration data collected under time-varying rotational speed conditions," *Data Brief*, vol. 21, pp. 1745–1749, 2018.
[25] N. D. Thuan and H. S. Hong, "HUST bearing: A practical dataset for ball bearing fault diagnosis," *BMC Res. Notes*, vol. 16, no. 1, 2023, Art. no. 138.
[26] S. Liu, J. Chen, S. He, Z. Shi, and Z. Zhou, "Subspace Network with Shared representation learning for intelligent fault diagnosis of machine under speed transient conditions with few samples," *ISA Trans.*, vol. 128, pp. 531–544, 2022.
[27] F. Lu, Q. Tong, Z. Feng, and Q. Wan, "Unbalanced bearing fault diagnosis under various speeds based on spectrum alignment and deep transfer convolution neural network," *IEEE Trans. Ind. Inf.*, vol. 19, no. 7, pp. 8295–8306, Jul. 2023.
[28] Q. Qian, Y. Wang, T. Zhang, and Y. Qin, "Maximum mean square discrepancy: A new discrepancy representation metric for mechanical fault transfer diagnosis," *Knowl.-Based Syst.*, vol. 276, 2023, Art. no. 110748.
[29] T. Han, C. Liu, W. Yang, and D. Jiang, "Deep transfer network with joint distribution adaptation: A new intelligent fault diagnosis framework for industry application," *ISA Trans.*, vol. 97, pp. 269–281, 2020.
[30] Q. Qian, Y. Qin, J. Luo, Y. Wang, and F. Wu, "Deep discriminative transfer learning network for cross-machine fault diagnosis," *Mech. Syst. Sig. Process.*, vol. 186, 2023, Art. no. 109884.

Feiyu Lu (Graduate Student Member, IEEE) received the B.S. degree in electrical engineering and automation from Beihua University, Jilin, China, in 2018, and the M.S. degree in power electronics and power drives from Shijiazhuang Tiedao University, Shijiazhuang, China, in 2021. He is currently working toward the Ph.D. degree in electrical engineering with Beijing Jiaotong University, Beijing, China.

His main research interests include rotating machinery condition monitoring and fault diagnosis.

Qingbin Tong received the Ph.D. degree in instrument science and technology from the Harbin Institue of Technology, Harbin, China, in 2008.

He is currently a Professor with the School of Electrical Engineering, Beijing Jiaotong University. He is mainly engaged in the research of artificial intelligence and intelligent testing under rail transit, power and electronics, and energy, including the dynamic modeling of key components of the system; fault diagnosis, damage assessment and life prediction; dynamic nonlinear and nonstationary signal analysis and processing.

Xuedong Jiang received the Ph.D. degree in electrical engineering from Beijing Jiaotong University, Beijing, China, in 2018. He is currently a Professor with the School of Electrical Engineering, Beijing Jiaotong University, Beijing, China. He is mainly engaged in the research of artificial intelligence and intelligent testing under rail transit, power and electronics, and energy, including the dynamic modeling of key components of the system; fault diagnosis, damage assessment and life prediction; dynamic nonlinear and non-stationary signal analysis and processing.

Ziwei Feng (Graduate Student Member, IEEE) received the B.S. degree in mechanical and electronic engineering from the Northeast Forestry University, Harbin, China, in 2021. She is currently working toward the Ph.D. degree in electrical engineering with Beijing Jiaotong University, Beijing, China.

She is mainly engaged in the research of dynamic nonlinear and nonstationary signal analysis and processing.

Jianjun Xu received the master of engineering degree in power system and automation from Beijing Jiaotong University, Beijing, China, in 2003.

He is currently a Senior Experimenter with Beijing Jiaotong University. He is currently the Director of the Experimental Center, School of Electrical Engineering, Beijing Jiaotong University. His research interests include automation, new energy and artificial intelligence.

Jingyi Huo received the master's degree in power system and its automation from Beijing Jiaotong University, Beijing, China, in 2018.

She is currently with the School of Electrical Engineering, Beijing Jiaotong University, as a Teacher in the experimental center and a mentor for undergraduate innovation and entrepreneurship. Her current research interests include intelligent detection, automation control, and intelligent electrical equipment.