Point-to-Set Metric-Gated Mixture of Experts for Multisource Domain Adaptation Fault Diagnosis

Boyuan Yang, *Member, IEEE*, Jinyuan Zhang, Ruonan Liu, *Senior Member, IEEE*, Di Lin, *Member, IEEE*, Ping Li, *Member, IEEE*, and C. L. Philip Chen, *Life Fellow, IEEE*

***Abstract*—The multisource unsupervised domain adaptation (MUDA) scenario poses a significant challenge in the field of intelligent fault diagnosis (IFD), where the goal is to transfer the knowledge learned from multiple labeled source domains to an unlabeled target domain. Existing IFD-oriented MUDA approaches frequently fail to recognize the distinct importance of each source domain relative to specific target samples, or lack flexibility in integrating diagnostic insights from multiple sources. In response, a novel MUDA approach is proposed for IFD, termed point-to-set metric-gated mixture of experts (PSMMoEs). This method leverages a mixture-of-experts (MoEs) framework to automatically integrate the complementary information from multiple source domains. It develops a deep point-to-set distance (PSD) metric learning technique within the MoE's gating mechanism, effectively fusing domain-specific features by assessing the similarity between individual target samples and each source domain. The method ensures balanced training across progressive stages, harmonizing multitask learning with joint training for the MoE framework. Furthermore, a multilayer maximum mean discrepancy (MMD) measurement is employed for domain alignment, ensuring feature alignment across different domains at multiple levels. In order to assess the efficacy of the proposed method, it is compared with several leading domain adaptation methods on publicly available and laboratory-based rotating machinery fault datasets. The experimental results demonstrate superior classification and adaptation capabilities of the proposed fault diagnosis method.**

Received 8 December 2023; revised 19 July 2024 and 12 December 2024; accepted 27 February 2025. This work was supported in part by the National Natural Science Foundation of China under Grant 62206199 and Grant U2141234, in part by the Young Elite Scientist Sponsorship Program under Grant YESS20220409, in part by the Science and Technology Program of Hebei under Grant 225676162GH, in part by Gusu Innovation Leading Talents Program under Grant ZXL2024356, in part by Hainan Province Science and Technology Special Fund under Grant ZDYF2024GXJS003, and in part by the National Science and Technology Major Project under Grant 2024ZD01NL00104. *(Corresponding author: Ruonan Liu.)*

Boyuan Yang is with the Center for Advanced Control and Smart Operations, Nanjing University, Suzhou 215163, China, and also with the State Key Laboratory of Mechanical Behavior and System Safety of Traffic Engineering Structures, Shijiazhuang 050043, China.

Jinyuan Zhang is with the Institute of Robotics and Automatic Information System, College of Artificial Intelligence, Nankai University, Tianjin 300350, China.

Ruonan Liu is with the Department of Automation, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: ruonan.liu@sjtu.edu.cn).

Di Lin is with the College of Intelligence and Computing, Tianjin University, Tianjin 300350, China.

Ping Li is with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong.

C. L. Philip Chen is with the School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China, also with the Navigation College, Dalian Maritime University, Dalian 116026, China, and also with the Faculty of Science and Technology, University of Macau, Macau, China.

Digital Object Identifier 10.1109/TNNLS.2025.3548894

***Index Terms*—Intelligent fault diagnosis (IFD), mixture of experts (MoEs), multisource domain adaptation, point-to-set distance (PSD).**

I. Introduction

As SYSTEMS, including machinery, equipment, and processes, become increasingly complex, the role of fault diagnosis becomes more critical. Fault diagnosis involves identifying and locating potential issues in systems, which is crucial for ensuring the reliability, safety, and efficiency of industrial operations. With the rapid advancement of artificial intelligence (AI), fault diagnosis is evolving into a more sophisticated technology known as intelligent fault diagnosis (IFD). This evolution is characterized by the integration of machine learning, deep learning, and other advanced technologies, which collectively automate and enhance the diagnostic process. The application of these technologies improves the accuracy and robustness of fault detection, thereby mitigating potential risks associated with system failures [1], [2], [3].

Conventional IFD approaches predominantly focus upon supervised learning methods, which are typically trained on labeled data specific to a particular domain, such as a particular machine type or specific operating conditions [4], [5], [6]. However, these methods suffer from two major limitations.

1) *Data Dependence and Costly Labeling:* These approaches require a large amount of labeled data, the acquisition of which is both cost-intensive and time-consuming.
2) *Distribution Assumption:* There is an underlying assumption that the training and testing datasets are drawn from the same distribution. This assumption is frequently invalidated in real-world scenarios due to various factors, such as environmental changes, operational variations, or sensor noise [7]. Therefore, these constraints hinder the ability of conventional IFD methods to generalize effectively to new or unseen domains.

To address the inherent limitations of traditional IFD methods, researchers have increasingly turned to unsupervised domain adaptation (UDA) techniques. These techniques facilitate the transfer of knowledge from a source domain, where labeled data are available, to a target domain that possesses only unlabeled data. This knowledge transfer is primarily achieved by minimizing the distribution discrepancies between these two domains. Within the context of IFD, UDA methods can be broadly categorized into four types based on the strategies they employ to align domains.

1) *Discrepancy-Based Methods:* These approaches focus on reducing domain discrepancy by minimizing various distance metrics between source and target distributions. Notable examples include maximum mean discrepancy (MMD) [8], correlation alignment (CORAL) [9], and Wasserstein distance [10].
2) *Adversarial Learning-Based Methods:* Employing adversarial learning techniques, these methods aim to align source and target distributions within a shared feature space. Examples include generative adversarial networks (GANs) [11] and domain adversarial neural networks (DANNs) [12].
3) *Reconstruction-Based Methods:* This approach employs reconstruction techniques to develop domain-invariant features by reconstructing input samples across different domains. Examples involve autoencoders [13] and cycle-consistent learning [14].
4) *Self-Training-Based Methods:* These strategies utilize self-training techniques to leverage the unlabeled data in the target domain by training models with target pseudo-labels. Examples include pseudo-labeling [15], [16] and co-training [17]. The growing interest in UDA methods is primarily due to their broader applicability and superior generalization capabilities, which are crucial for meeting practical diagnostic demands. Consequently, an increasing number of IFD methodologies are now being developed based on UDA principles, indicating a significant shift toward more robust and flexible diagnosis solutions.

In the field of cross-domain fault diagnosis, most existing UDA approaches focus on learning from a single source domain, typically referred to as single-source UDA (SUDA). However, real-world scenarios frequently involve multiple source domains that are relevant to the target domain, such as various machine types or operational conditions. These source domains may exhibit varying degrees of similarity or dissimilarity to the target domain, challenging the fundamental assumption of SUDA, which posits homogeneous source sample distributions. This discrepancy underscores the necessity for research in multisource UDA (MUDA), where leveraging knowledge from multiple sources could harness more valuable information and enhance the generalization capability of fault diagnosis models. However, MUDA introduces several new challenges.

1) *Source-Domain Selection or Fusion:* Recent strategies address how to select and combine multiple source domains based on their relevance to the target domain. Methods such as output weighting [18], output combination [19], and output alignment [20] have been developed. However, output weighting relies on broad domain-level operations, output combination directly integrates outputs without considering flexibility, and output alignment lacks guaranteed consistency across all source domains.
2) *Management of Conflicting or Noisy Information:* To mitigate the impact of conflicting or noisy information from various sources, recent studies have explored approaches such as knowledge distillation [21], multiadversarial learning [22], and feature selection [23]. Nevertheless, knowledge distillation may struggle to distill accurate insights from noisy data, multiadversarial learning might not completely reconcile conflicts, and feature selection risks omitting critical informative features.
3) *Balancing Domain Alignment With Task-Specific Learning:* To address the tradeoff between aligning domains and focusing on task-specific objectives, recent research has proposed methods such as weighted learning [24] and multistage alignment [25]. However, weighted learning can struggle with adapting to varying degrees of domain dissimilarity, and multistage alignment introduces increased complexity and computational demands.

In response, a novel model, termed point-to-set metric-gated mixture of experts (PSMMoEs), is proposed to address the challenges of MUDA in fault diagnosis. Structurally, this model is composed of a deep convolutional shared feature extractor, an mixture-of-expert (MoE)-specific feature extractor, and a single classifier. Functionally, the model diagnoses faults by exploiting both domain-shared and domain-specific features extracted from raw vibration signals. Within the gating mechanism of the MoE feature extractor, a deep point-to-set distance (PSD) metric learning approach is developed. This approach trains a transferability perception metric, which subsequently facilitates the aggregation of features. These ensemble features are utilized by the classifier to predict fault types. To address the three aforementioned challenges associated with MUDA, the model incorporates the following strategies.

1) Upon receiving an input signal, the trained transferability measure dynamically integrates domain-specific features from multiple experts based on the PSD between the input and each source domain.
2) To deal with conflicting or noisy information from different sources, a multitask learning loss is developed to foster the independent training of each expert model, enhancing specialization and diversity. Additionally, a joint training loss is implemented to ensure collaborative operation among all experts.
3) Domain alignment is achieved using a multilayer MMD criterion that aligns both domain-shared and domain-specific features. A dynamic parameter, which evolves throughout the training process, is employed as a tradeoff between the MMD criterion and classification error, thus striking a balance between domain alignment and task-specific learning.

The major contributions of this article can be outlined as follows.

1) A novel deep PSD metric learning method is proposed, which incorporates multiple adaptive coefficient vectors for leveraging source subsets to facilitate convenient distance computation. A transformation network is introduced to train the PSD metric simultaneously with the entire model utilizing deep contrastive learning techniques. This PSD metric is both interpretable and effective in reflecting the similarity between the

input and each source domain, thus functioning as a transferability perception metric to integrate expert-specific outputs.

2) Both domain-shared and domain-specific features are leveraged to determine the final fault prediction. Using the metric-gated MoE feature extractor, domain-specific features are integrated based on each source domain's transferability to the input signal. Two collaborative losses, namely, multitask learning and joint training losses, are introduced within the MoE framework to enhance expert specialization and collaborative performance, respectively. A multilayer MMD criterion is implemented to ensure the alignment of both domain-shared and domain-specific features within the MoE module. Additionally, a dynamic tradeoff mechanism is developed to prioritize different training objectives as the model optimization progresses. These strategies collectively empower the PSMMoE model to effectively address the challenges in MUDA.
3) Extensive experiments are conducted on a variety of publicly available and laboratory-based rotating machinery datasets to evaluate the performance of the proposed method. The results demonstrate that PSMMoE significantly outperforms several leading SUDA and MUDA methods, showcasing its superiority in fault diagnosis.

The structure of the rest of this article is arranged as follows. Section II covers the related work. Section III presents the preliminaries. Section IV details the problem formulation and proposed PSMMoE method. The experimental validation is elaborated in Section V, and Section VI concludes this article.

II. Related Work

A. Multisource Adaptation Fault Diagnosis

While SUDA has proven beneficial, its utility is often constrained by the limited diversity of a single source domain and a pronounced sensitivity to discrepancies between the source and target domains. In response, MUDA has garnered increasing attention, particularly for addressing more complex scenarios in IFD.

Many MUDA methodologies extend classic SUDA techniques to accommodate the difference of multiple source domains. For instance, adversarial domain adaptation methods, which leverage the DANN framework, have been further developed by researchers such as Zhang et al. [19] and Zhu et al. [22]. Concurrently, domain-specific feature adaptation techniques grounded in MMD have been introduced by Wang et al. [20], Tian et al. [18], and Zhu et al. [25]. Expanding upon these foundational approaches, Feng et al. [21] implemented dual alignment modules to synchronize both local and global distributions, complemented by a distillation strategy to optimize classifier performance. Further enhancing the versatility of MUDA, Li and Yu [23] employed a multitask learning framework to capture both domain-invariant and task-specific features, utilizing a weighting scheme for source-domain fusion. Additionally, Chen et al. [24] introduced an open-set recognition module to address unknown fault types in the target domain, incorporating a weighted approach to balance the alignment of known and unknown classes effectively.

Distinguishing from these methods, the proposed model not only emphasizes the unique importance of each source domain relative to specific target samples, but also ensures robust domain alignment across the source and target domains while concurrently facilitating effective task-specific learning.

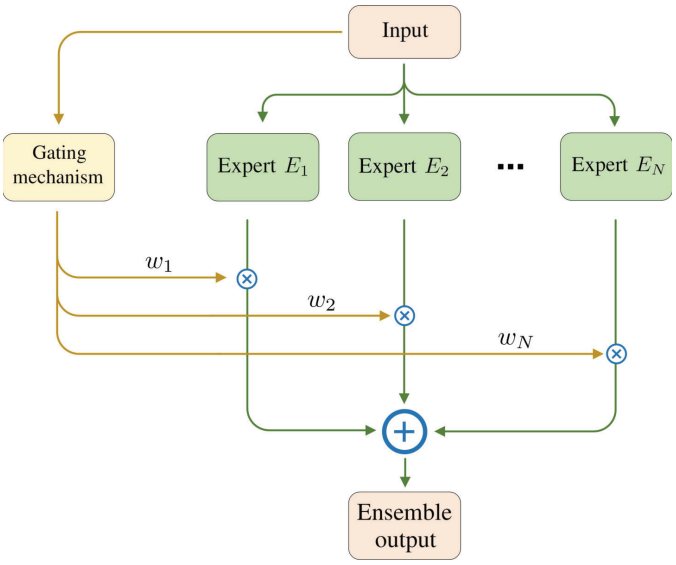


Fig. 1. Architecture of the MoE model. w_i is a parametric metric that measures how much information from expert E_i is utilized for a given input.

B. MoEs Neural Networks

Introduced by Jacobs et al. [26], the MoE framework has been widely adopted across various research domains. As depicted in Fig. 1, MoE utilizes multiple specialized models, known as experts, to tackle distinct subtasks within complex problems. Each expert is trained on a specific subset of input data, enhancing its specialization. A gating mechanism determines the selection of the appropriate expert for a given input. The final output is a combination of the experts' results, weighted according to the gating mechanism's decisions. Subsequent research has expanded MoE's capabilities, focusing on enhancements in model capacity [27], gating mechanisms [28], and network structures [29].

MoE has gained prominence in machine learning fields such as computer vision, natural language processing, and speech recognition. For instance, Riquelme et al. [30] introduced vision MoE, a sparse variant of the vision Transformer, tailored for image recognition tasks. Lepikhin et al. [31] developed an MoE approach for large-scale language modeling, employing a top-k gating mechanism that assigns tokens to thousands of experts simultaneously. Furthermore, Kumatani et al. [32] utilized a sparsely gated MoE for multisource domain adaptation in speech recognition, enhancing both network capacity and accuracy. Despite its extensive contributions to machine learning, MoE has been relatively underexplored in the realm of IFD. For instance, in neural machine translation (NMT), MoE has seen considerable development in comparison, mainly due to the following three factors: 1) the readily available

multilingual data; 2) MoE's capability to enhance performance and generalization by leveraging diverse network architectures and parameters, which is crucial for managing large-scale, high-dimensional NMT data; and 3) MoE's ability to improve efficiency and scalability through sparse activation and parallel computation. Recently, Chen et al. [33] employed a multigated MoE in multitask learning for bearing and gear fault diagnosis tasks. However, this article suggests that the potential of MoE is underdeveloped in the context of MUDA fault diagnosis.

By resolving conflicts from diverse sources and integrating the expertise of multiple domain-specific experts through an innovative gating mechanism, the proposed method seeks to demonstrate the versatile applications of MoE in complex diagnostic scenarios in the field of IFD.

III. Preliminaries

A. Point-to-Set Distance

The PSD is a metric used to quantify the distance between a single point $p \in \mathbb{R}^m$ and a set of points $\boldsymbol{D} \in \mathbb{R}^{m\times n}$. The specific definition of PSD can vary depending on the application, with the concept rooted in point-to-point distances. The point-to-point distance is denoted as $d_{\mathrm{p2p}}(\cdot,\cdot)$, which may represent any distance metric, such as Euclidean or Manhattan distance. One of the most common measures of PSD is the minimum distance, representing the shortest distance between the point p and any point q in the set $\boldsymbol{D}$

$$d_{\mathrm{p2s}}^{\min}(p,\boldsymbol{D}) = \min_{q\in\boldsymbol{D}} d_{\mathrm{p2p}}(p,q). \tag{1}$$

Similarly, PSD can be defined for the maximum distance, which measures the farthest distance between the point p and any point q in $\boldsymbol{D}$. However, these PSD measures are highly sensitive to outliers in the set $\boldsymbol{D}$, as a single distant point can significantly influence the measure.

To address this sensitivity, more robust PSD metrics have been proposed. One such metric is the Mahalanobis distance [34], which accounts for correlations between variables in the set and measures the distance between a point and a set in a multivariate space

$$d_{\mathrm{p2s}}^{\mathrm{M}}(p,\boldsymbol{D}) = \sqrt{(p-\mu)^{\top}\Sigma^{-1}(p-\mu)}. \tag{2}$$

Here, μ and Σ are the mean vector and the covariance matrix of the set $\boldsymbol{D}$, respectively. Specifically, the Mahalanobis distance assumes that the set $\boldsymbol{D}$ follows a Gaussian distribution, which may not always be accurate.

An alternative approach is the linear combination distance, which uses a coefficient $\boldsymbol{\alpha} \in \mathbb{R}^n$ to compute a linear sum of the set $\boldsymbol{D}$

$$d_{\mathrm{p2s}}^{\mathrm{lin}}(p,\boldsymbol{D}) = d_{\mathrm{p2p}}(p,\boldsymbol{D}\boldsymbol{\alpha}) = d_{\mathrm{p2p}}\left(p,\sum_{i=1}^{n}\alpha_i d_i\right) \tag{3}$$

where α_i are the elements of the coefficient, and each $d_i \in \mathbb{R}^m$ is a point in the set $\boldsymbol{D}$. This method allows for a weighted combination of distances, making it applicable even when the distribution of $\boldsymbol{D}$ is complex or non-Gaussian. To obtain the optimal value of $\boldsymbol{\alpha}$, Zhu et al. [35] proposed using least-squares regression and ridge regression to solve the minimization problem $\hat{\boldsymbol{\alpha}} = \arg\min_{\alpha} d(p,\boldsymbol{D}\boldsymbol{\alpha})$. However, this solution has the following issues.

1) The entire set $\boldsymbol{D}$ is employed concurrently for matrix operations. The computational complexity associated with these operations will increase exponentially with the size of $\boldsymbol{D}$.
2) If the set $\boldsymbol{D}$ is singular or ill-conditioned, the desired solution for $\hat{\boldsymbol{\alpha}}$ cannot be obtained using the normal equation.

B. Metric Learning

Metric learning is a subfield of machine learning focused on developing algorithms to learn distance metrics directly from data. These learned metrics are tailored to specific tasks, enabling more accurate and effective distance measurements compared to predefined metrics like the Euclidean distance. The fundamental goal in point-to-point distance metric learning is to learn a distance function $d_{\mathrm{p2p}}(p,q)$ that quantifies the similarity or dissimilarity between data points p and q.

Linear metric learning methods involve learning a linear transformation of the input space to optimize a specific distance metric. A common approach in linear metric learning is Mahalanobis distance learning [36], which involves learning a positive semidefinite matrix $\boldsymbol{M}$ that defines the distance between two data points p and q as follows:

$$d_{\mathrm{p2p}}^{\mathrm{M}}(p,q) = \sqrt{(p-q)^{\top}\boldsymbol{M}(p-q)}. \tag{4}$$

While effective for many tasks, these linear methods may be inadequate when dealing with complex, high-dimensional data.

Nonlinear metric learning methods extend linear approaches by introducing nonlinear transformations, which can be achieved through kernel methods or, more powerfully, deep learning techniques. Utilizing deep neural networks to learn a distance metric leads to the field of deep metric learning (DML) [37]. Here, a neural network f_θ, parameterized by θ, maps input data points to feature representations in a learned feature space. The distance metric in this feature space is typically represented as

$$d_{\mathrm{p2p}}^{\mathrm{DML}}(p,q) = d_{\mathrm{p2p}}(f_\theta(p), f_\theta(q)). \tag{5}$$

DML aims to learn representations of data such that similar data points are closer together in the learned feature space, while dissimilar points are farther apart. This is achieved by training the parameter θ to optimize the distance metric using specialized loss functions, such as contrastive loss and triplet loss. These loss functions and training methods enable DML to create highly discriminative and task-specific feature spaces through the trained neural network.

IV. Proposed Method

Problem Formulation: Fault data collected from various machinery conditions exhibit significant distribution differences, making it inappropriate to treat these data as originating from a single distribution. Consequently, N labeled source domains $\{\mathcal{S}_i\}_{i=1}^{N}$ are considered, where each source domain $\mathcal{S}_i = \left\{\left(x_i^j, y_i^j\right)\right\}_{j=1}^{n_i}$ consists of n_i sample pairs. In this context,

x_i^j denotes the jth data sample in the ith source domain, and y_i^j represents the corresponding label. The data in $\mathcal{S}_i$ are sampled from the distribution P_{s_i}. The target domain $\mathcal{T} = \{x_t^j\}_{j=1}^{n_t}$ consists of n_t unlabeled samples, which are drawn from the distribution P_t. It is assumed that the input feature space and label space are shared across all domains. However, the feature distributions of P_{s_i} and P_t may differ due to domain shift, i.e., $P_{s_i} \neq P_{s_j} \neq P_t$ for any $i, j \in \{1, 2, \ldots, N\}, i \neq j$. The objective is to utilize $\{\mathcal{S}_i\}_{i=1}^N$ and $\mathcal{T}$ to train a target model θ_T capable of accurately predicting the corresponding label for a given sample $x \sim P_t$.

Overall Framework: The proposed model, termed PSMMoE, is designed for the problem of MUDA in the context of IFD. The architecture of PSMMoE is illustrated in Fig. 2. Structurally, the model consists of three main components: a deep convolutional shared feature extractor $\mathcal{G}_s$, an MoE specific feature extractor $\mathcal{G}_p$, and a single classifier $\mathcal{C}$. Within the MoE feature extractor, the gating mechanism employs multiple adaptive coefficient vectors $\{\boldsymbol{V}_i\}_{i=1}^N$, which facilitates the extraction of salient information from each source domain. Additionally, a transformation network ϕ is optimized using both positive and negative sample pairs to learn the optimal PSD metric. Each expert in the MoE is dedicated to learn specialized features for distinct source domains, and their outputs are integrated using the learned PSD metric. To address the challenge of domain adaptation, the model leverages the multilayer MMD criterion, which enhances the alignment between the source and target domains.

A. Deep Point-to-Set Metric Learning

In the context of MUDA, two critical challenges should be addressed: the integration of outputs from multiple source domains and the alignment between source and target domains. In the proposed method, the PSD metric serves as a measure of transferability, adaptively combining the output features from N source domains to ensure effective integration.

Given a point x and a source domain $\mathcal{S} \in \mathbb{R}^{m \times n}$ comprising n samples, the PSD is defined considering the linear combination method in (3). In response to the existing issues in the method by Zhu et al. [35], a trainable coefficient matrix $\boldsymbol{V} \in \mathbb{R}^{n' \times m}$ is introduced to collaborate with x and replace $\boldsymbol{\alpha}$, where n' is substantially smaller than n ($n' \ll n$). Furthermore, to rectify the assumption in the linear distance metrics that features maintain a linear relationship with the distance measures, a nonlinear transformation is implemented via a transformation network ϕ. The PSD formulation now considers a subset $\mathcal{S}'$, where $\mathcal{S}' \subseteq \mathcal{S}$ and $\mathcal{S}' \in \mathbb{R}^{m \times n'}$. It is characterized by the squared Euclidean distance as follows:

$$d\left(x, \mathcal{S}'\right) = \left\|\phi\left(x\right) - \phi\left(\mathcal{S}'\boldsymbol{V}x\right)\right\|_2^2. \tag{6}$$

This formulation extends the point-to-point distance metric from (5) into the PSD context. Consequently, $d(x, \mathcal{S}')$ not only enhances computational efficiency by avoiding the need to consider all samples in $\mathcal{S}$, but also adeptly captures the inherent nonlinearity of the dataset. However, fully optimizing this PSD metric necessitates addressing two primary challenges: determining the optimal configuration of $\boldsymbol{V}$ that captures the most salient features of $\mathcal{S}$ effectively and refining the transformation network ϕ to ensure accurate transformation of the input data into the desired feature space.

The coefficient matrix $\boldsymbol{V}$ fulfills two crucial roles: first, it projects the input x onto a subspace delineated by the most salient features of $\mathcal{S}$, and second, it quantifies the significance or contribution of each feature within this new subspace. By reducing the dimensionality of a dataset while retaining as much information as possible, optimizing $\boldsymbol{V}$ can be akin to conducting a form of principal component analysis. The objective function for optimizing $\boldsymbol{V}$ is proposed as follows:

$$J\left(\boldsymbol{V}\right) = \left\|x - \mathcal{S}'\boldsymbol{V}x\right\|_2^2 + \beta\left\|\boldsymbol{V}\right\|_2^2 \tag{7}$$

where β represents the parameter controlling the regularization term. The L2-norm regularization term, $\|\boldsymbol{V}\|_2^2$, not only penalizes the magnitude of the coefficients to balance the model's complexity, but also ensures that the objective function remains convex, thereby simplifying the optimization process.

To minimize this objective function, the gradient of $J(\boldsymbol{V})$ with respect to $\boldsymbol{V}$ can be derived as follows:

$$\nabla_{\boldsymbol{V}} J\left(\boldsymbol{V}\right) = 2\mathcal{S}'^{\top}\mathcal{S}'\boldsymbol{V} - 2\mathcal{S}'^{\top}x + 2\beta\boldsymbol{V}. \tag{8}$$

Given that the objective function $J(\boldsymbol{V})$ with respect to $\boldsymbol{V}$ is convex, the optimal $\hat{\boldsymbol{V}}$ can theoretically be determined by setting $\nabla_{\boldsymbol{V}} J(\boldsymbol{V}) = 0$, where $\hat{\boldsymbol{V}} = (\mathcal{S}'^{\top}\mathcal{S}' + \beta\boldsymbol{I})^{-1}\mathcal{S}'^{\top}x$. However, given the challenges in fully capturing all characteristics of $\mathcal{S}$ with a considerably smaller subset $\mathcal{S}'$, it is proposed to employ a gradient descent method for gradually leveraging data from the entire $\mathcal{S}$. This approach iteratively updates $\boldsymbol{V}$ using mini-batches of $\mathcal{S}$ until $\hat{\boldsymbol{V}}$ is optimally determined. In this context, n' can be set as the size of each mini-batch.

The objective of deep PSD metric learning is to leverage an optimal $\hat{\boldsymbol{V}}$ to train a transformation network ϕ that effectively quantifies the similarity between a sample x and a source domain $\mathcal{S}'$. Utilizing the mapping function $\phi(\cdot)$ facilitates nonlinear transformations and restructures features within a new space. The distance metric, modified from (6), is defined as follows:

$$d\left(x, \mathcal{S}'\right) = \left\|\frac{\phi\left(x\right) - \phi\left(\mathcal{S}'\hat{\boldsymbol{V}}x\right)}{d_{\max}}\right\|_2^2. \tag{9}$$

This formulation employs the predefined normalization factor $d_{\max}$, bounding the distance metric $d(x, \mathcal{S}')$ within a range from 0 to 1. This normalization ensures that the PSD metric is comparable across different samples and source domains. Moreover, it avoids the metric becoming overly large or small due to variations in the scale of $\phi(\cdot)$.

The transformation network ϕ plays a pivotal role in the computation of $d(x, \mathcal{S})$. For each sample x and a subset $\mathcal{S}_i' \subseteq \mathcal{S}_i$, an effective ϕ should minimize the PSD between x and $\mathcal{S}_i'$ when $x \in P_{s_i}$, while enlarging it when $x \notin P_{s_i}$. To facilitate the training of ϕ, both positive and negative pairings are generated

$$\begin{aligned}\mathcal{D}_+ &= \left\{\left(x, \mathcal{S}_i'\right) \middle| x \in \mathcal{S}_i, i \in [N]\right\} \\ \mathcal{D}_- &= \left\{\left(x, \mathcal{S}_k'\right) \middle| x \in \mathcal{S}_i, k = \arg\min_{j \neq i} d\left(x, \mathcal{S}_j'\right), i, j \in [N]\right\}.\end{aligned} \tag{10}$$

Here, $[N]$ denotes the set $\{1, 2, \ldots, N\}$. Each positive pair $(x, \mathcal{S}_i')$ is constructed using x and a subset of its respective source $\mathcal{S}_i$, while negative pairs $(x, \mathcal{S}_k')$ are identified by determining the minimal PSD distances $d(x, \mathcal{S}_j')$ for $j \neq i$, indicating the closest incorrect source domain. This selection process is applied across all samples and source domains, forming a collection of positive and negative pairs, denoted as $\mathcal{D}_+$ and $\mathcal{D}_-$, which accurately reflect the similarities and differences between the samples and source domains.

The training of parameters within ϕ utilizes a contrastive loss function designed to minimize the PSD metric for positive pairs in $\mathcal{D}_+$ and maximize it for negative pairs in $\mathcal{D}_-$. Fig. 3 demonstrates the contrastive learning process for PSD metric optimization. The deep PSD metric learning loss is formulated as

$$\mathcal{L}_{\text{psd}} = \sum_{(x,\mathcal{S})\in\mathcal{D}_+} d(x, \mathcal{S}) + \sum_{(x,\mathcal{S})\in\mathcal{D}_-} \max(0, 1 - d(x, \mathcal{S})) + \sum_{(x,\mathcal{S})\in\mathcal{D}_-} \max(0, d(x, \mathcal{S}) - 1). \quad (11)$$

Given that the distance metric $d(x, \mathcal{S})$ is normalized, the second and third terms of the loss employ a dual strategy to ensure that $d(x, \mathcal{S})$ remains close to 1 for negative pairs. Consequently, setting $d_{\max}$ to an appropriately estimated value will enable the loss function $\mathcal{L}_{\text{psd}}$ to adaptively constrain $d(x, \mathcal{S})$ to $d_{\max}$. Through iterative optimization of the contrastive loss, ϕ is refined to capture and discriminate the relationships between samples and their respective source domains.

B. Mixture of Experts

Considering an input sample from the target domain $x \in \mathcal{T}$, x is first processed through a deep convolutional shared feature extractor, denoted by $z = \mathcal{G}_s(x)$. The primary function of this extractor is to map the input sample into a feature space that captures general information useful across all source domains. Following this, domain-specific features are derived through the MoE feature extractor, which is expressed as

$$\mathcal{G}_p(z) = \sum_{i=1}^{N} w(z, \mathcal{S}_i) \cdot \mathcal{G}_{p_i}(z) \quad (12)$$

where $\mathcal{G}_{p_i}(z)$ corresponds to the output features produced by the ith domain-specific extractor (or expert). The weighting function $w(x, \mathcal{S}_i)$ serves as a confidence metric to assess the importance of domain $\mathcal{S}_i$ for the input x. The weights are computed using softmax normalization, leveraging a pretrained PSD metric to evaluate the transferability of each source domain with respect to z, as shown in the following:

$$w(z, \mathcal{S}_i) = \frac{\exp(1 - d(z, \mathcal{G}_s(\mathcal{S}_i)))}{\sum_{j=1}^{N} \exp\left(1 - d\left(z, \mathcal{G}_s\left(\mathcal{S}_j\right)\right)\right)}. \quad (13)$$

A lower PSD value $d(z, \mathcal{S}_i)$ indicates a higher similarity between the sample z and the source domain $\mathcal{S}_i$. Consequently, $1 - d(z, \mathcal{S}_i)$ inversely scales the distance, making it a direct indicator of transferability. This transformation ensures that domains with greater relevance (lower distances) are assigned exponentially higher weights, aligning with the objective of giving more importance to more relevant domains.

For training the MoE feature extractor with source domains, two distinct loss functions are employed: the multitask learning loss and the joint training loss. The multitask learning loss, aimed at optimizing each expert model independently for its respective domain data, is defined as

$$\mathcal{L}_{\text{mtl}} = -\frac{1}{N} \sum_{i=1}^{N} \frac{1}{n_i} \sum_{j=1}^{n_i} \log \mathcal{C}\left(y_i^j \middle| \mathcal{G}_{p_i}\left(\mathcal{G}_s\left(x_i^j\right)\right)\right). \quad (14)$$

This loss promotes the independent training of each expert model, enhancing specialization and diversity within the MoE framework.

Conversely, the joint training loss seeks to optimize all expert models collaboratively, using the combined outputs from the ensemble

$$\mathcal{L}_{\text{joint}} = -\frac{1}{N} \sum_{i=1}^{N} \frac{1}{n_i} \sum_{j=1}^{n_i} \log \mathcal{C}\left(y_i^j \middle| \mathcal{G}_p\left(\mathcal{G}_s\left(x_i^j\right)\right)\right). \quad (15)$$

This loss ensures that all experts work together harmoniously, aligning with the outputs of the gating mechanism. The dual loss strategy balances the need for expert specialization and collaborative performance, making the MoE module robust and effective for domain-specific feature extraction.

C. Domain Alignment

In the context of MUDA, the integration of outputs from multiple source domains is addressed through the PSD metric-gated MoE. The other significant challenge in this context is the alignment of features between the source and target domains. If discrepancies are not addressed in the initial stages, they may propagate and potentially become amplified in the deeper layers of the network. Consequently, the initial step involves the alignment of the input features of the MoE module (the domain-shared features), formulated as

$$\mathcal{L}_{\text{mmd-s}} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{D}^2\left(\mathcal{G}_s(\mathcal{S}_i), \mathcal{G}_s(\mathcal{T})\right) \quad (16)$$

where

$$\mathbb{D}^2(\mathcal{X}, \mathcal{Y}) = \frac{1}{n^2} \sum_{j=1}^{n} \sum_{k=1}^{n} k(x_j, x_k) + \frac{1}{m^2} \sum_{j=1}^{m} \sum_{k=1}^{m} k(y_j, y_k) - \frac{2}{nm} \sum_{j=1}^{n} \sum_{m=1}^{m} k(x_j, y_k)$$

represents the MMD between two sets of samples, $\mathcal{X} = \{x_j\}_{j=1}^{n}$ and $\mathcal{Y} = \{y_j\}_{j=1}^{m}$, in a reproducing kernel Hilbert space (RKHS). $k(\cdot, \cdot)$ is a positive-definite kernel function, with the Gaussian kernel being utilized here.

The feature extractor $\mathcal{G}_s$ typically learns low-level, generic features that are universally applicable and less specific to any particular domain. However, aligning these features might not fully address the more complex and domain-specific variations present in higher level features. To tackle this issue, the concept of multilayer MMD is introduced, which simultaneously aligns low-level and high-level features across domains. To mitigate the impact of discrepancies in high-level features on final predictions, MMD is also applied to the outputs of the

MoE feature extractor. For the ensemble output of the MoE module, the discrepancy in the weighting metric w between source- and target-domain samples is inevitable. Therefore, it is proposed to align the output features of each expert individually

$$\mathcal{L}_{\text{mmd-p}} = \frac{1}{N}\sum_{i=1}^{N}\mathbb{D}^2\left(\mathcal{G}_{p_i}\left(G_s\left(\mathcal{S}_i\right)\right), \mathcal{G}_{p_i}\left(G_s\left(\mathcal{T}\right)\right)\right). \tag{17}$$

The comprehensive MMD loss is then formulated by combining these two terms

$$\mathcal{L}_{\text{mmd}} = \mathcal{L}_{\text{mmd-s}} + \mathcal{L}_{\text{mmd-p}}. \tag{18}$$

Through the optimization of this composite loss function, both the domain-shared and domain-specific features are aligned within the framework of the MoE module.

D. Optimization and Fault Diagnosis

The training of the PSMMoE model utilizes a composite loss function that consists of four distinct components: the PSD metric learning loss $\mathcal{L}_{\text{psd}}$, the MoE multitask learning loss $\mathcal{L}_{\text{mtl}}$, the MoE joint training loss $\mathcal{L}_{\text{joint}}$, and the MMD loss $\mathcal{L}_{\text{mmd}}$. This combined loss function is formulated as

$$\mathcal{L} = \lambda\mathcal{L}_{\text{joint}} + (1-\lambda)\,\mathcal{L}_{\text{mtl}} + \lambda\mathcal{L}_{\text{mmd}} + \mathcal{L}_{\text{psd}} \tag{19}$$

where $\lambda > 0$ modulates the balance among these terms. Notably, the training of the PSD metric learning occurs simultaneously with that of the backbone network, yet their parameters are independent. Therefore, there is no need for a distinct balancing factor for the $\mathcal{L}_{\text{psd}}$ component.

The adjustment of λ during the training process is described by

$$\lambda = \frac{2}{1+\exp\left(-\zeta p\right)} - 1 \tag{20}$$

where p increases gradually from 0 to 1 throughout the training period, and ζ is a parameter that determines the rate of this increase. Initially, λ is near zero and progressively converges toward one as training progresses. This dynamic adjustment of λ serves two primary functions.

1) Initially, when the PSD metric is underdeveloped, the MoE ensemble output may not be fully effective. During this phase, the loss function primarily emphasizes the individual losses of each expert, optimizing them for their respective source domains. As training progresses, the emphasis shifts toward the joint loss of the ensemble output, thereby optimizing the expert networks to cooperate with the gating mechanism output.
2) In the early stages of training, minimizing the MMD loss is less effective due to the lack of meaningful or representative characteristics from the feature extractors. The early focus is on enhancing the feature discriminability for classification tasks. As training advances, the parameter adjustment for $\mathcal{L}_{\text{mmd}}$ increases, indicating a shift in focus toward enhancing feature transferability and thus improving generalization to the target domain.

Algorithm 1 outlines the pseudocode for the optimization algorithm based on gradient descent. Algorithm 2 describes the practical application of the trained model for fault diagnosis upon receiving a new fault signal.

Algorithm 1 Training PSMMoE

Input: Labeled source domains $\{\mathcal{S}_i\}_{i=1}^N$, unlabeled target domain $\mathcal{T}$, learning rate η, hyperparameters ζ, β and $d_{\max}$.
Output: Trained shared feature extractor $\mathcal{G}_s$, MoE specific feature extractor $\mathcal{G}_p$, transformation layer ϕ, classifier $\mathcal{C}$, and vectors $\{\boldsymbol{V}_i\}_{i=1}^N$.

Initialize the coefficient vectors $\{\boldsymbol{V}_i\}_{i=1}^N$ as $\boldsymbol{I}$.
repeat
  Sample N mini-batches $\{(\mathcal{X}_i^s, \mathcal{Y}_i^s)\}_{i=1}^N$ from each source domain.
  Sample a mini-batch $\mathcal{X}^t$ from the target domain $\mathcal{T}$.
  Extract domain-shared features $\{\mathcal{Z}_i^s\}_{i=1}^N \leftarrow \ \mathcal{Z}_i^s = \mathcal{G}_s(\mathcal{X}_i^s)$.
  for each source domain $i = 1$ to N **do**
    Calculate the gradient $\nabla_{\boldsymbol{V}_i} J(\boldsymbol{V}_i) \leftarrow$ Eq. (8).
    Update coefficient vector $\boldsymbol{V}_i = \boldsymbol{V}_i - \eta\nabla_{\boldsymbol{V}_i} J(\boldsymbol{V}_i)$.
    Compute PSD $\{d(\mathcal{Z}_i^s, \mathcal{G}_s(\mathcal{S}_j))\}_{j=1}^N \leftarrow$ Eq. (9).
    Determine transferability metric $\{w(\mathcal{Z}_i^s, \mathcal{S}_j)\}_{j=1}^N \leftarrow$ Eq. (13).
  end for
  Formulate positive and negative sets $\mathcal{D}_+$ and $\mathcal{D}_- \leftarrow$ Eq. (10).
  Compute $\mathcal{L}_{\text{psd}}$ using $\mathcal{D}_+$ and $\mathcal{D}_- \leftarrow$ Eq. (11).
  Compute $\mathcal{L}_{\text{mtl}}$ with $\{(\mathcal{X}_i^s, \mathcal{Y}_i^s)\}_{i=1}^N \leftarrow$ Eq. (14).
  Calculate $\mathcal{L}_{\text{joint}}$ with $\{(\mathcal{X}_i^s, \mathcal{Y}_i^s)\}_{i=1}^N$ and $w \leftarrow$ Eq. (15).
  Compute $\mathcal{L}_{\text{mmd}}$ with $\{\mathcal{X}_i^s\}_{i=1}^N$ and $\mathcal{X}^t \leftarrow$ Eq. (18).
  Update trade-off parameter $\lambda \leftarrow$ Eq. (20).
  Determine the combined loss $\mathcal{L} \leftarrow$ Eq. (19).
  Update parameters of $\mathcal{G}_s$, $\mathcal{G}_p$, ϕ and $\mathcal{C}$ with gradients of $\mathcal{L}$.
until converge.
return $\mathcal{G}_s$, $\mathcal{G}_p$, ϕ, $\mathcal{C}$, and $\{\boldsymbol{V}_i\}_{i=1}^N$.

Algorithm 2 Fault Diagnosis With Trained PSMMoE

Input: Trained $\mathcal{G}_s$, $\mathcal{G}_p$, ϕ, $\mathcal{C}$ and $\{\boldsymbol{V}_i\}_{i=1}^N$, input signal $x \sim P^t$, the mean μ and standard deviation σ of $\mathcal{T}$, source embeddings $\{\mathcal{S}_i'\}_{i=1}^N$, hyperparameters β and $d_{\max}$.
Output: Fault diagnosis result.

Normalize the input signal $\hat{x} \leftarrow \ \hat{x} = \frac{x-\mu}{\sigma}$.
Extract domain-shared features $z \leftarrow \ z = \mathcal{G}_s(\hat{x})$.
for each source domain $i = 1$ to N **do**
  Compute PSD $d(z, \mathcal{G}_s(\mathcal{S}_i')) \leftarrow$ Eq. (9).
  Determine transferability $w(z, \mathcal{S}_i') \leftarrow$ Eq. (13).
  Extract domain-specific feature $z_i' \leftarrow \ z_i' = \mathcal{G}_{p_i}(z)$.
end for
Aggregate $\{z_i'\}_{i=1}^N$ to form the ensemble feature $\hat{z} \leftarrow$ Eq. (12).
Obtain the fault prediction $\hat{y} \leftarrow \ \mathcal{C}(\hat{z})$.
return $\hat{y}$.

V. Experiments

To assess the efficacy of the proposed PSMMoE, comprehensive experiments are conducted using both publicly available and laboratory-collected datasets, derived from diverse operational conditions. The performance of the PSMMoE is benchmarked against several leading methods under identical training configurations to ensure a fair and

TABLE I
CONDITION INFORMATION OF CWRU

Condition	Load (hp)	Revolution (rpm)	Samples
1	0	1797	558 (62 per class)
2	1	1772	3348 (372 per class)
3	2	1750	4266 (474 per class)
4	3	1730	4257 (473 per class)

TABLE II
CONDITION INFORMATION OF MFPT

Condition		Normal	Inner race	Outer race
1	Load	270	0, 50	270
	Samples	572	286	572
2	Load	270	100, 150, 200	25, 50, 100, 150
	Samples	572	429	572
3	Load	270	250, 300	200, 250, 300
	Samples	572	286	429

rigorous comparison. Additionally, subsequent experiments are designed to specifically investigate and analyze the performance contributions of the PSD metric component, multitask learning, and joint training within the PSMMoE framework.

A. Data Description

1) CWRU Dataset: The CWRU dataset originates from Case Western Reserve University [38]. Defects are intentionally introduced into bearings through electrical discharge machining at a single point. Vibration data are collected using an accelerometer at a sampling rate of 48000 samples/s. The dataset includes recordings from four motor loads: 0, 1, 2, and 3 horsepower (hp), corresponding to 1797, 1772, 1750, and 1730 revolutions per minute (r/min), respectively. It categorizes bearing faults into inner ring faults (IF), outer ring faults (OF), and rolling element faults (RF), each presented in three fault sizes: 7, 14, and 21 mil. This configuration results in nine distinct fault classes under each working condition, such as 7-mil IF, 14-mil IF, and 7-mil OF. Data segmentation is performed using a sliding window with a length of 1024 samples. The condition information is summarized in Table I.

2) Mfpt Dataset: The MFPT dataset is provided by the Society for Machinery Failure Prevention Technology [39]. It features NICE bearings under normal and fault conditions, with vibration data captured for three operational states: normal, OF, and IF. The dataset is recorded under varying loads at the sampling frequencies of 97.656 kHz for normal conditions and 48.828 kHz for fault conditions. Each condition is divided into segments of 1024 samples for experiments. Despite identical load for the normal state across all conditions, the vibration data are derived from different bearings. The organization of these conditions is detailed in Table II.

3) FB Dataset: The faulty bearing (FB) laboratory dataset is collected using the experimental bench shown in Fig. 4. The setup includes a two-stage planetary gearbox with a 27:1 gear ratio, featuring four planets in stage 1 and three in stage 2, and a two-stage parallel shaft gearbox. The parallel shaft gearbox contains three in-line parallel shafts configurable as single- or two-stage reduction/increaser, with six rolling element bearings. The setup is driven by a 3-hp variable frequency ac motor with a multifeatured front panel programmable controller, powered by a 220-VAC one-phase supply. Vibration signals are recorded at a frequency of 12.8 kHz under three distinct conditions: 1) a 0.2-A load with a constant speed of 30 revolutions per second (r/s); 2) a 0.5-A load at 30 r/s; and 3) a 0.5-A load with a variable speed ranging from 0 to 30 r/s. Faults are induced at four points on either bearings or gears, resulting in eight specific fault types for each condition. The data are sampled using a length of 1024, with label information and sample counts detailed in Table III.

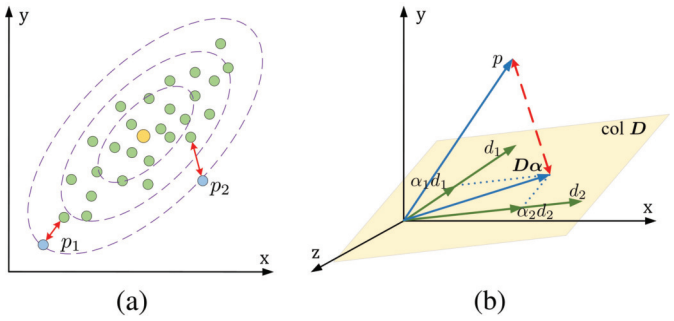


Fig. 2. Illustration of PSD metrics. (a) Minimum distance (red double arrow) and Mahalanobis distance (with purple dashed lines indicating different distance levels). Query points p_1 and p_2 have equal Mahalanobis distances, as they reside on the same distance level. (b) Linear combination distance (red double arrow). The distance can be the residual vector when p is projected onto the linear subspace defined by the columns of $\boldsymbol{D}$. Better viewed in color.

TABLE III
LABEL INFORMATION OF FB

Label	Working status	Samples of three conditions		
		No.1	No.2	No.3
0	Normal	762	725	587
1	Spur gear surface damage	731	737	750
2	Spur gear dedendum cracks	731	731	581
3	Spur gear teeth cutting	725	762	706
4	Spur gear teeth missing	718	718	731
5	Bearing compound fault	675	768	731
6	Bearing rolling element fault	718	725	843
7	Bearing inner ring fault	737	737	675
8	Bearing outer ring fault	731	737	893

B. Experimental Settings

1) Compared Methods: To assess the performance of the proposed model, PSMMoE is compared with several leading SUDA and MUDA methods.

1) *Convolutional Neural Network:* This baseline approach utilizes the convolutional neural network (CNN) without incorporating any adaptation techniques. It is trained on the source domain and evaluated directly on the target domain.
2) *DCTLN [40]:* This SUDA approach incorporates a condition recognition module for fault classification and employs a domain adaptation module based on MMD.

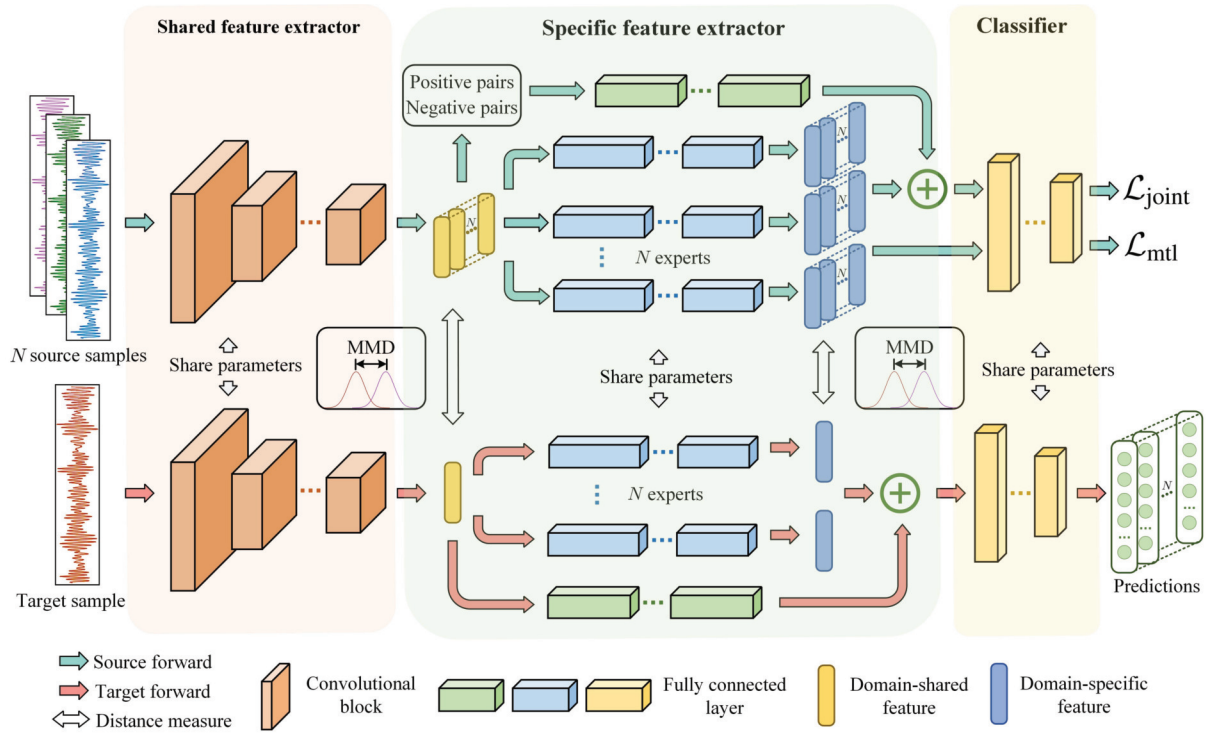


Fig. 3. Proposed architecture of PSMMoE.

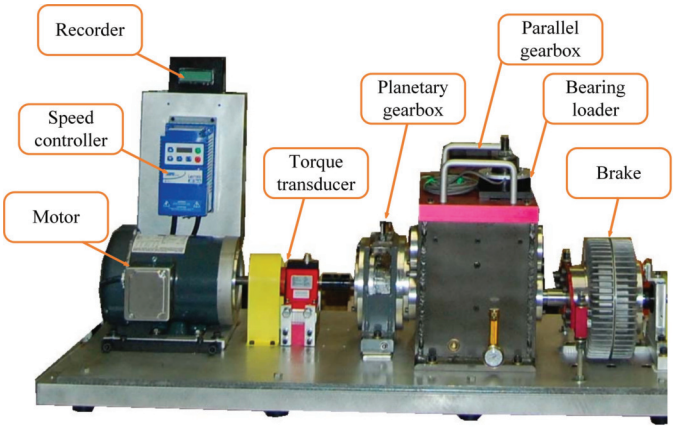


Fig. 4. Bearing and gear testing bench for the FB dataset.

3) *DATN [41]:* This SUDA method features dual asymmetric encoder networks for feature extraction and a domain alignment module leveraging DANN.
4) *ACDANN [42]:* An improved iteration of DANN, this SUDA method enhances domain alignment by utilizing both the feature outputs from the generator and the predictive insights from the classifier.
5) *ADACL [19]:* This MUDA strategy uses adversarial learning and incorporates domain classifier alignment to manage discrepancies among multiple classifiers.
6) *MSSA [18]:* This MUDA method utilizes a multibranch network for feature extraction, employs local MMD for distribution alignment, and calculates a weighted score computed from MMD for prediction combination.

TABLE IV
IMPLEMENTATION DETAILS OF EXPERIMENTS

Hyper-parameter	Value	Hyper-parameter	Value
Batch size	64	Increasing rate ζ	10
Initial learning rate (LR)	0.01	Weight decay rate	5×10^{-4}
Momentum for SGD	0.9	Max epoch	30
LR decay epoch	10	LR decay multiplier	0.2
Normalization factor $d_{\max}$	1	Trade-off β	1

7) *MFSAN [43]:* This MMD-based MUDA method aligns not only the source- and target-domain distributions, but also predictions across multiple classifiers.
8) *MANMoE [44]:* This adversarial learning-based MUDA method employs domain-invariant and domain-specific features extracted by MoE networks, with the MoE output features integrated by a multilayer perceptron (MLP) gating network.

Methods 2–6 were originally developed for IFD, Method 7 was evaluated on image datasets, and Method 8 was designed for NMT. Here, both Methods 7 and 8 are adapted for application in IFD.

2) Implementation Details: To facilitate a fair comparison across all methods, including the proposed model, the network architectures and hyperparameters are standardized. A multiscale convolutional network serves as the backbone for all methods. This architecture integrates five CNNs, each with varying kernel sizes of 4, 8, 16, 24, and 32, to effectively capture diverse signal features at different scales. Additionally, an MLP is employed as the classifier for these methods. Stochastic gradient descent (SGD) is utilized as the

TABLE V
CASES OF EXPERIMENTS

Case	Dataset	Source	Target
C_1	CWRU	Condition 1, 2 and 3	Condition 4
C_2	CWRU	Condition 1, 2 and 4	Condition 3
C_3	CWRU	Condition 1, 3 and 4	Condition 2
C_4	MFPT	Condition 1 and 2	Condition 3
C_5	MFPT	Condition 1 and 3	Condition 2
C_6	MFPT	Condition 2 and 3	Condition 1
C_7	FB	Condition 1 and 2	Condition 3
C_8	FB	Condition 1 and 3	Condition 2
C_9	FB	Condition 2 and 3	Condition 1

optimization technique for training. Detailed settings of the parameters are enumerated in Table IV.

The datasets used in this experiment include CWRU, MFPT, and FB, which are segmented into nine multisource transfer scenarios, as detailed in Table V. Given that Methods 2–4 are inherently designed for single-source applications, they are adapted for multisource scenarios through a source-combined standard. For this standard, all source domains are integrated into a single composite source for the purpose of SUDA training. During training, models have access to the complete set of signals and labels from the source domains but are limited to a specific subset of the target-domain signals. The remaining segments of the target domain are reserved for testing. In the experimental setup, 80% of the target-domain samples are randomly selected for training, and the residual 20% are reserved for testing purposes.

C. Fault Classification Performance

Table VI details the fault diagnosis results across nine cases, with outcomes reported as the mean ± variance of accuracies over five trials. From these results, several critical conclusions can be drawn.

1) Methods incorporating UDA techniques consistently outperform those without UDA, thereby effectively mitigating domain-shift issues. Methods 2–8, along with the proposed method, consistently achieve higher accuracy compared to the CNN across all cases, underscoring the benefits of integrating UDA strategies.
2) The homogeneity of the source-domain distribution plays a crucial role in the effectiveness of SUDA methods. Combining source domains with diverse distributions often results in minimal improvements or even a reduction in accuracy compared to the best single-source transfer. For instance, ACDANN achieves a peak accuracy of 99.48% in single-source transfer at C_6, but its performance drops to 90.95% when trained with combined sources. This discrepancy underscores the need for enhanced MUDA to address these shortcomings.
3) Some existing MUDA techniques struggle to effectively fuse information from multiple source domains, sometimes resulting in accuracies that do not surpass those of SUDA methods. This issue highlights a common challenge: many MUDA methods, despite employing specialized feature extractors or classifiers for multisource scenarios, still fall short in efficiently synthesizing outputs from various sources. Methods based on MoE, such as MANMoE and the proposed model, demonstrate distinct advantages, particularly in scenarios like C_1, C_2, and C_3, where the number of source domains is more.
4) The proposed method exhibits outstanding multisource transfer capabilities. PSMMoE generally achieves the highest accuracies in comparison with other SUDA and MUDA approaches, with the exception of a slight underperformance in C_4 and C_5. Remarkably, in C_9, PSMMoE surpasses other methods by at least 2.78%.

Fig. 5 employs t-SNE for feature visualization, illustrating the features input to the classifier from both source and target domains after dimensionality reduction in condition C_1. For clarity, 100 random samples from each fault status are selected for visualization. Fig. 5(a)–(c) delineate s the top three comparison methods (excluding the proposed method) that demonstrate the highest accuracy under source-combined or multisource standards. Fig. 5(d)–(f) present s the feature distributions from individual experts within PSMMoE for their respective source domains. Fig. 5(g) displays the integrated feature distribution achieved by PSMMoE. The analysis yields the following insights.

1) Comparison of Fig. 5(a)–(c) with Fig. 5(g) demonstrates PSMMoE's superior feature extraction and domain adaptation capabilities relative to other methods. The second-best method, MANMoE, shows significant feature mixing (e.g., between RF-14 and IF-14) and apparent disparities (especially for IF-14) in feature distributions for the same fault across source and target domains. This suggests that while the accuracy improvement might be modest, PSMMoE offers substantial benefits in terms of more pronounced feature distributions.
2) Drawing comparisons of Fig. 5(d)–(f) with Fig. 5(g), it illustrates that PSMMoE, which integrates insights across all source domains, outperforms the individual domain experts. Although the feature overlap by the $\mathcal{S}_3$ expert shows reasonable alignment, certain health states (e.g., RF-07 and OF-07) demonstrate poor performance. In contrast, PSMMoE's fusion of source-domain features adeptly aligns all health states, with only minor mixing observed between IF-14 and IF-21.
3) Observations from Fig. 5(d)–(f) reveal that feature overlap tends to increase with the similarity of working conditions. Specifically, $\mathcal{S}_1$ and the target domain $\mathcal{T}$ exhibit the least overlap due to their contrasting conditions. Conversely, the proximity of working conditions between $\mathcal{S}_3$ and $\mathcal{T}$ facilitates more substantial feature overlap.

D. Effect of Core Components

To empirically validate the efficacy of the PSD metric, multitask learning, and joint training within the proposed PSMMoE framework, a detailed comparative analysis is conducted using a series of modified models.

TABLE VI
CLASSIFICATION ACCURACY (%) FOR CASES

Standards	Methods	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9
Single-best	CNN	91.51±2.01	92.80±1.08	89.95±1.67	80.62±3.29	82.97±2.09	85.16±4.33	74.61±1.92	81.04±0.85	83.20±1.93
	DCTLN	94.91±1.32	94.47±0.04	96.62±0.86	**89.06**±4.64	96.09±2.11	99.74±0.21	81.38±2.98	85.81±1.05	84.58±1.81
	DATN	95.35±1.06	94.31±0.06	96.04±0.61	85.42±2.52	93.23±2.10	97.40±0.57	78.72±0.58	83.70±1.42	84.86±1.26
	ACDANN	96.67±1.46	95.15±0.86	95.63±1.30	86.29±5.33	96.15±1.32	99.48±0.50	77.89±1.12	85.63±2.48	84.54±1.37
Source-combined	CNN	91.51±1.12	92.80±0.43	89.95±0.27	80.62±1.93	82.97±1.56	85.16±8.49	74.61±0.38	81.04±0.45	83.20±1.44
	DCTLN	94.99±0.21	94.99±0.21	95.25±0.40	88.02±6.35	89.06±1.61	93.49±0.00	81.20±0.95	84.72±0.37	84.90±0.44
	DATN	94.59±0.15	93.24±0.49	94.03±0.13	88.80±1.44	91.14±1.12	92.55±1.71	80.73±1.59	85.07±1.06	83.05±1.54
	ACDANN	95.92±0.41	95.20±0.18	95.14±0.27	85.85±1.11	87.63±1.03	90.95±0.28	81.28±1.23	84.12±0.93	82.55±0.67
Multi-source	ADACL	96.55±0.37	96.11±0.15	97.07±0.05	86.46±4.89	94.40±1.51	96.72±0.16	81.66±1.11	87.01±1.12	83.18±0.89
	MFSAN	95.67±0.51	96.35±0.25	96.51±0.18	86.46±6.39	**96.23**±1.96	99.48±0.34	81.43±0.40	89.09±1.09	85.57±0.25
	MSSA	95.87±0.52	95.83±0.34	96.20±0.38	85.94±5.69	93.88±2.08	99.22±0.34	81.35±1.36	88.36±0.87	85.44±0.89
	MANMoE	96.59±0.49	97.37±0.26	97.56±0.79	85.42±2.68	92.19±0.54	98.10±0.49	82.82±1.07	87.34±0.57	84.03±1.00
	PSMMoE	**98.67**±0.18	**98.81**±0.05	**98.82**±0.18	87.26±5.29	94.62±1.72	**99.65**±0.18	**85.08**±0.25	**90.94**±0.12	**88.35**±0.19

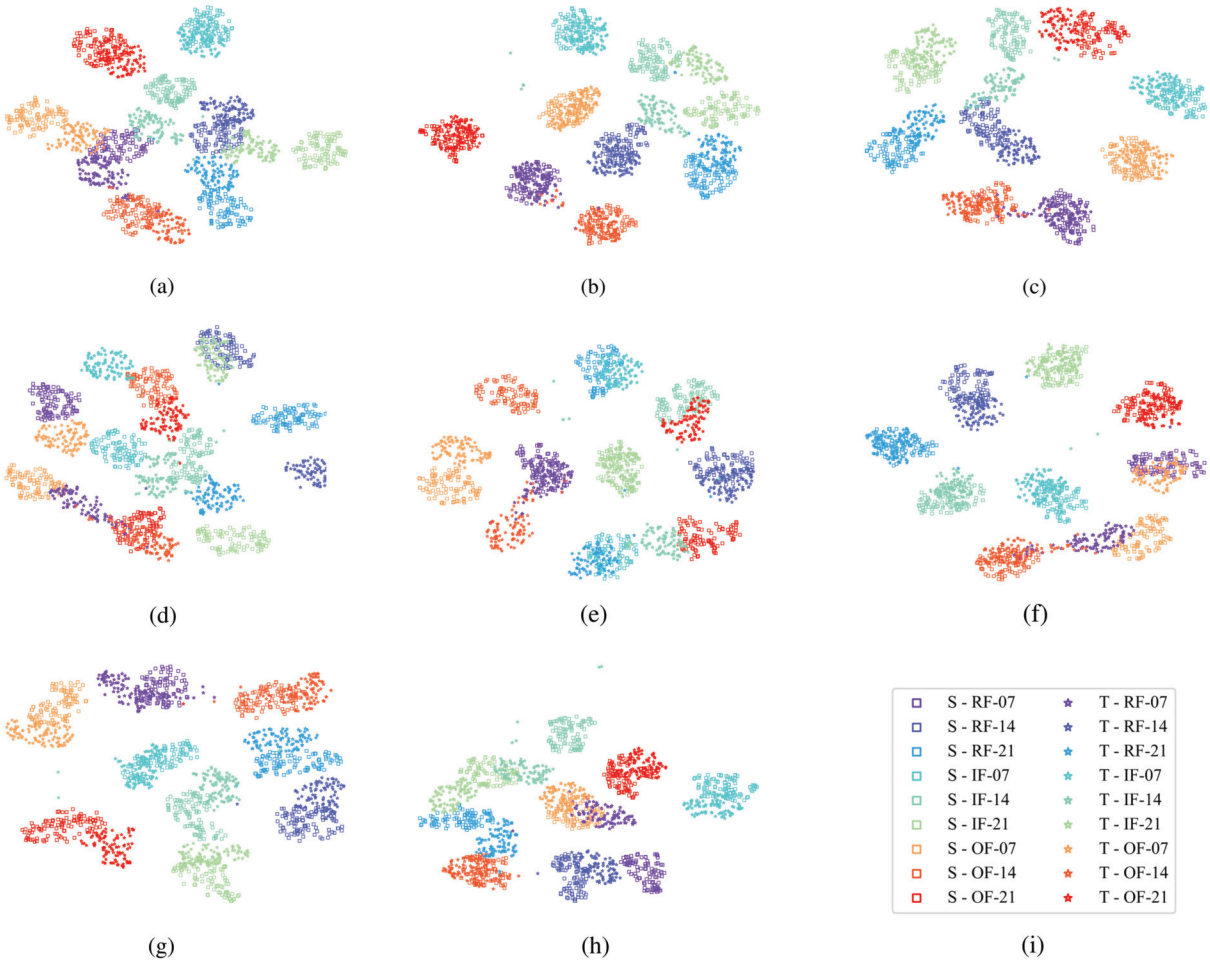


Fig. 5. Feature visualization. Identical colors indicate the same category, while identical shapes represent the same domain. In the legend, S represents the source domain, and T denotes the target domain. (a) ACDANN. (b) ADACL. (c) MANMoE. (d) $\mathcal{S}_1$ expert. (e) $\mathcal{S}_2$ expert. (f) $\mathcal{S}_3$ expert. (g) PSMMoE. (h) CNN. (i) Legend.

1) *Sum Combination:* This method directly aggregates the outputs from multiple domain-specific feature extractors to obtain the ensemble feature.
2) *MMD Weights:* Utilized within the MSSA framework, this technique employs the MMD metric as a weighting mechanism to gauge the similarity between target and source domains during feature fusion.
3) *MLP Gating:* Predominantly applied in MoE networks, this strategy utilizes an MLP as a gating network, akin to the approach in the MANMoE framework.

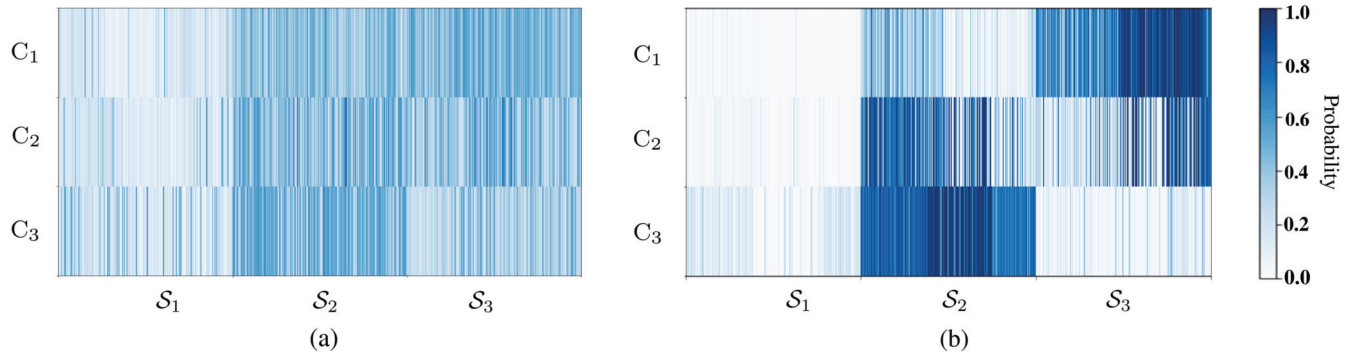


Fig. 6. w distributions across source domains for randomly selected 200 target samples. (a) PSMMoE. (b) MLP gating.

TABLE VII
ACCURACY (%) FOR CASES

Methods	C_1	C_2	C_3	C_7	C_8	C_9
Sum combination	95.42	95.28	96.32	80.83	87.77	83.32
MMD weights	95.75	94.72	96.19	81.46	87.01	85.87
MLP gating	96.82	96.40	96.64	83.32	89.51	85.96
PSD meta-training	95.28	96.38	97.28	82.81	88.29	86.10
PSMMoE w/o $\mathcal{L}_{\text{mtl}}$	98.48	98.74	98.62	82.75	87.36	88.33
PSMMoE w/o $\mathcal{L}_{\text{joint}}$	97.80	98.32	98.39	83.10	88.55	87.19
PSMMoE	**98.67**	**98.81**	**98.82**	**85.08**	**90.94**	**88.35**

4) *PSD Meta-Training:* A meta-learning-based PSD metric learning strategy proposed for MoE networks, as detailed in [45].
5) *PSMMoE w/o $\mathcal{L}_{\text{mtl}}$ or $\mathcal{L}_{\text{joint}}$:* The PSMMoE framework excludes either the multitask learning loss ($\mathcal{L}_{\text{mtl}}$) or the joint training loss ($\mathcal{L}_{\text{joint}}$).

For this experiment, six cases are chosen: three with class-balanced data (C_1–C_3) and three with class-imbalanced data (C_7–C_9), to assess the core components of the proposed method. The experimental outcomes, as shown in Table VII, consistently demonstrate that PSMMoE surpasses other fusion methods and ablated models in accuracy across all cases. These findings confirm that the PSD metric learning strategy effectively promotes the MoE-based networks. Moreover, the collaborative effect of the multitask learning and joint training losses markedly enhances the functionality of the MoE extractor. Notably, the exclusion of $\mathcal{L}_{\text{joint}}$ leads to a more pronounced performance drop than removing $\mathcal{L}_{\text{mtl}}$, underscoring the importance of optimizing the classification capabilities of the MoE networks in conjunction with the gating mechanisms.

Fig. 6 illustrates the distributions of the transferability factor w across source domains for selected samples in experiments of C_1–C_3. For the proposed model, the heatmap depicted in Fig. 6(a) underscores that different target samples benefit from distinct combinations of source-domain insights. Typically, the distribution of w exhibits a preference for certain source domains, indicating that the model assigns greater importance to domains that are more informative for the specific target sample. In contrast, the MLP gating method, as shown in Fig. 6(b), displays an extreme bias, indicating that it inclines to select the most informative domain instead of integrating insights from all source domains. However, the proposed deep PSD metric learning strategy adeptly combines insights from a variety of domains, thereby emphasizing its advanced capability to leverage a holistic domain knowledge.

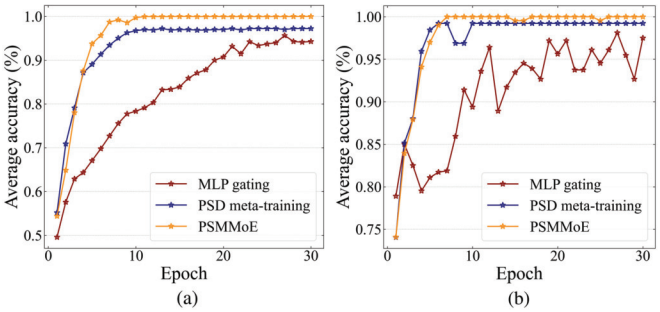


Fig. 7. Classification accuracy of the gating mechanisms. (a) C_1. (b) C_4.

Fig. 7 provides a detailed depiction of the classification accuracy of identifying the source domain from which domain-shared features of a sample are derived. This analysis highlights several key findings.

1) PSD-based gating mechanisms demonstrate faster convergence and superior accuracy compared to the commonly used MLP gating mechanisms. This observation underscores the effectiveness and potential for the development of PSD-based strategies within this application.
2) The advanced PSD metric learning method manifests its strengths early in the training process. Remarkably, it achieves a classification accuracy of nearly 100% within just a few epochs.
3) An interesting observation emerges when integrating insights from both Figs. 6(a) and 7. Despite achieving exceptional classification accuracy for source-domain samples, the gating mechanism maintains balanced PSD outputs for target-domain samples. This ensures that the PSD across different source domains is distinct yet not excessively different, facilitating a balanced and comprehensive consideration of source knowledge.

E. Parameter Sensitivity

This section delves into the sensitivity of the hyperparameter ζ and its influence on the model's performance through modulation of the loss function behavior. Fig. 8 delineates the

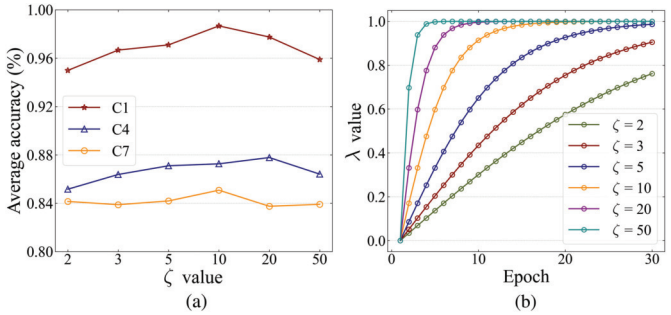


Fig. 8. Influence of hyperparameter ζ. (a) Fault classification accuracy with different ζ. (b) Values of λ over successive training epochs.

fault classification accuracy for selected cases (C_1, C_4, and C_7) as a function of varying ζ values, alongside the evolution of λ during training under different ζ settings. Analysis of these observations reveals the following.

1) The rate of increase in λ, which is critical for controlling the balance between different loss terms, is significantly affected by ζ. A smaller ζ results in a gradual increase of λ, sometimes failing to reach 1.0 even by the training's end. This slow pace can impede adequate learning of domain alignment and MoE joint training, negatively impacting the overall training effectiveness. Conversely, an excessively high ζ causes λ to rise too quickly, potentially reaching 1.0 too early. This rapid escalation can hamper the early phase of MoE multitask learning and may cause domain alignment to excessively influence the model before its classification capabilities are fully developed. Optimal tuning of ζ is crucial as both extremities can detrimentally affect model performance.
2) Further insights are drawn by comparing Fig. 8(b) with Fig. 7, which shows a clear link between the progression of λ and the accuracy trajectory of the PSD gating mechanism. Notably, during the initial training epochs, the PSD metric is in its learning stages. It is only after approximately five epochs, when the accuracy of the PSD gating mechanism has sufficiently improved that the joint output of the MoE network becomes reliable. This pivotal observation validates the strategic selection of λ, which is carefully designed to ensure that both the MoE multitask learning and joint training losses are optimized to contribute effectively to the total loss, thus enhancing the robustness of the training process.

VI. Conclusion

In this article, a novel model termed PSMMoE is introduced, specifically designed for MUDA cross-domain fault diagnosis. The model is characterized by several innovative contributions.

1) A deep PSD metric learning method is proposed within the gating mechanism of the MoE architecture. This method adaptively and effectively integrates domain-specific features based on their transferability for each individual target sample.
2) The model incorporates multitask learning and joint training techniques to collaboratively train the MoE module, which strikes a balance between expert specialization and the optimization of ensemble output.
3) A multilayer MMD is tailored to address complex domain shifts, ensuring alignment of both domain-shared and domain-specific features. Extensive experiments conducted on publicly available and laboratory datasets for IFD demonstrate that PSMMoE consistently outperforms several leading SUDA and MUDA methods.

For the advantages, the PSMMoE model adeptly integrates complementary information from multiple source domains, thus enhancing diagnostic accuracy across varied operating environments. This model develops key components that offer valuable insights for addressing diagnostic challenges in MUDA scenarios. However, the simultaneous optimization of the PSD metric with the overall network demands more computational resources and introduces complexity in training. These are potential drawbacks that need to be addressed in future developments to enhance the model's applicability in industrial applications.

References

[1] X. Zhao et al., "Intelligent fault diagnosis of gearbox under variable working conditions with adaptive intraclass and interclass convolutional neural network," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 9, pp. 6339–6353, Sep. 2023.
[2] Y. Wang et al., "Coarse-to-fine: Progressive knowledge transfer-based multitask convolutional neural network for intelligent large-scale fault diagnosis," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 2, pp. 761–774, Feb. 2023.
[3] J. Li, Y. Wang, Y. Zi, and Z. Zhang, "Whitening-Net: A generalized network to diagnose the faults among different machines and conditions," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 10, pp. 5845–5858, Oct. 2022.
[4] T. de Bruin, K. Verbert, and R. Babuška, "Railway track circuit fault diagnosis using recurrent neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 3, pp. 523–533, Mar. 2017.
[5] J. Shi, D. Peng, Z. Peng, Z. Zhang, K. Goebel, and D. Wu, "Planetary gearbox fault diagnosis using bidirectional-convolutional LSTM networks," *Mech. Syst. Signal Process.*, vol. 162, Jan. 2022, Art. no. 107996.
[6] H. Wang, Z. Liu, D. Peng, and Y. Qin, "Understanding and learning discriminant features based on multiattention 1DCNN for wheelset bearing fault diagnosis," *IEEE Trans. Ind. Informat.*, vol. 16, no. 9, pp. 5735–5745, Sep. 2020.
[7] W. Lu, B. Liang, Y. Cheng, D. Meng, J. Yang, and T. Zhang, "Deep model based domain adaptation for fault diagnosis," *IEEE Trans. Ind. Electron.*, vol. 64, no. 3, pp. 2296–2305, Mar. 2017.
[8] Y. Li, Y. Song, L. Jia, S. Gao, Q. Li, and M. Qiu, "Intelligent fault diagnosis by fusing domain adversarial training and maximum mean discrepancy via ensemble learning," *IEEE Trans. Ind. Informat.*, vol. 17, no. 4, pp. 2833–2841, Apr. 2021.
[9] Q. Qian, Y. Qin, Y. Wang, and F. Liu, "A new deep transfer learning network based on convolutional auto-encoder for mechanical fault diagnosis," *Measurement*, vol. 178, p. 109352, Jun. 2021.
[10] F. Ferracuti, A. Freddi, A. Monteriù, and L. Romeo, "Fault diagnosis of rotating machinery based on Wasserstein distance and feature selection," *IEEE Trans. Autom. Sci. Eng.*, vol. 19, no. 3, pp. 1997–2007, Jul. 2022.
[11] J. Wang, B. Han, H. Bao, M. Wang, Z. Chu, and Y. Shen, "Data augment method for machine fault diagnosis using conditional generative adversarial networks," *Proc. Inst. Mech. Engineers, D, J. Automobile Eng.*, vol. 234, no. 12, pp. 2719–2727, Jun. 2020.
[12] W. Mao, Y. Liu, L. Ding, A. Safian, and X. Liang, "A new structured domain adversarial neural network for transfer fault diagnosis of rolling bearings under different working conditions," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–13, 2021.
[13] M. Ma, C. Sun, and X. Chen, "Deep coupling autoencoder for fault diagnosis with multimodal sensory data," *IEEE Trans. Ind. Informat.*, vol. 14, no. 3, pp. 1137–1145, Mar. 2018.

[14] X. Li, W. Zhang, H. Ma, Z. Luo, and X. Li, "Degradation alignment in remaining useful life prediction using deep cycle-consistent learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 10, pp. 5480–5491, Oct. 2022.

[15] B. Zhao, C. Cheng, S. Zhao, and Z. Peng, "Hybrid semi-supervised learning for rotating machinery fault diagnosis based on grouped pseudo labeling and consistency regularization," *IEEE Trans. Instrum. Meas.*, vol. 72, pp. 1–12, 2023.

[16] N. Lu, H. Xiao, Z. Ma, T. Yan, and M. Han, "Domain adaptation with self-supervised learning and feature clustering for intelligent fault diagnosis," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 6, pp. 7657–7670, Jun. 2022.

[17] T. S. Abdelgayed, W. G. Morsi, and T. S. Sidhu, "Fault detection and classification based on co-training of semisupervised machine learning," *IEEE Trans. Ind. Electron.*, vol. 65, no. 2, pp. 1595–1605, Feb. 2018.

[18] J. Tian, D. Han, M. Li, and P. Shi, "A multi-source information transfer learning method with subdomain adaptation for cross-domain fault diagnosis," *Knowl.-Based Syst.*, vol. 243, May 2022, Art. no. 108466.

[19] Y. Zhang, Z. Ren, S. Zhou, and T. Yu, "Adversarial domain adaptation with classifier alignment for cross-domain intelligent fault diagnosis of multiple source domains," *Meas. Sci. Technol.*, vol. 32, no. 3, Dec. 2020, Art. no. 035102.

[20] R. Wang, W. Huang, J. Wang, C. Shen, and Z. Zhu, "Multisource domain feature adaptation network for bearing fault diagnosis under time-varying working conditions," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–10, 2022.

[21] Y. Feng, J. Chen, S. He, T. Pan, and Z. Zhou, "Globally localized multisource domain adaptation for cross-domain fault diagnosis with category shift," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 6, pp. 3082–3096, Jun. 2023.

[22] J. Zhu, N. Chen, and C. Shen, "A new multiple source domain adaptation fault diagnosis method between different rotating machines," *IEEE Trans. Ind. Informat.*, vol. 17, no. 7, pp. 4788–4797, Jul. 2021.

[23] S. Li and J. Yu, "A multisource domain adaptation network for process fault diagnosis under different working conditions," *IEEE Trans. Ind. Electron.*, vol. 70, no. 6, pp. 6272–6283, Jun. 2023.

[24] Z. Chen et al., "A multi-source weighted deep transfer network for open-set fault diagnosis of rotary machinery," *IEEE Trans. Cybern.*, vol. 53, no. 3, pp. 1982–1993, Mar. 2023.

[25] J. Zhu, N. Chen, C. Shen, and D. Wang, "Multi-source unsupervised domain adaptation for machinery fault diagnosis under different working conditions," in *Proc. IEEE 18th Int. Conf. Ind. Informat. (INDIN)*, vol. 1, Jul. 2020, pp. 755–762.

[26] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton, "Adaptive mixtures of local experts," *Neural Comput.*, vol. 3, no. 1, pp. 79–87, 1991.

[27] D. Eigen, M. Ranzato, and I. Sutskever, "Learning factored representations in a deep mixture of experts," 2013, *arXiv:1312.4314*.

[28] N. Shazeer et al., "Outrageously large neural networks: The sparsely-gated mixture-of-experts layer," 2017, *arXiv:1701.06538*.

[29] R. D. De Veaux, "Mixtures of linear regressions," *Comput. Statist. Data Anal.*, vol. 8, no. 3, pp. 227–245, Nov. 1989.

[30] C. Riquelme et al., "Scaling vision with sparse mixture of experts," in *Proc. NIPS*, 2021, pp. 8583–8595.

[31] D. Lepikhin et al., "GShard: Scaling giant models with conditional computation and automatic sharding," 2020, *arXiv:2006.16668*.

[32] K. Kumatani et al., "Building a great multi-lingual teacher with sparsely-gated mixture of experts for speech recognition," 2021, *arXiv:2112.05820*.

[33] Y. Chen, L. Ren, H. Xia, Z. Wang, C. Gao, and F. Wang, "A compound fault diagnosis method based on multi-task learning with multi-gate mixture-of-experts," in *Proc. 14th Int. Conf. Measuring Technol. Mechatronics Autom. (ICMTMA)*, Jan. 2022, pp. 281–285.

[34] R. De Maesschalck, D. Jouan-Rimbaud, and D. L. Massart, "The Mahalanobis distance," *Chemometrics Intell. Lab. Syst.*, vol. 50, no. 1, pp. 1–18, 2000.

[35] P. Zhu, L. Zhang, W. Zuo, and D. Zhang, "From point to set: Extend the learning of distance metrics," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 2664–2671.

[36] J. V. Davis, B. Kulis, P. Jain, S. Sra, and I. S. Dhillon, "Information-theoretic metric learning," in *Proc. 24th Int. Conf. Mach. Learn. (ICML)*, 2007, pp. 209–216.

[37] D. Yi, Z. Lei, S. Liao, and S. Z. Li, "Deep metric learning for person re-identification," in *Proc. 22nd Int. Conf. Pattern Recognit.*, Aug. 2014, pp. 34–39.

[38] W. A. Smith and R. B. Randall, "Rolling element bearing diagnostics using the case western reserve university data: A benchmark study," *Mech. Syst. Signal Process.*, vols. 64–65, pp. 100–131, Dec. 2015.

[39] *Society For Machinery Failure Prevention Technology*. Accessed: Mar. 28, 2023. [Online]. Available: https://mfpt.org/fault-data-sets/

[40] 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.

[41] Z. Chen, G. He, J. Li, Y. Liao, K. Gryllias, and W. Li, "Domain adversarial transfer network for cross-domain fault diagnosis of rotary machinery," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 11, pp. 8702–8712, Nov. 2020.

[42] Q. Wang, C. Taal, and O. Fink, "Integrating expert knowledge with domain adaptation for unsupervised fault diagnosis," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–12, 2022.

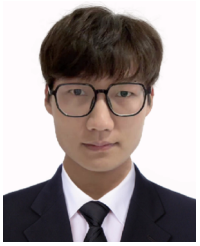
[43] Y. Zhu, F. Zhuang, and D. Wang, "Aligning domain-specific distribution and classifier for cross-domain classification from multiple sources," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 5989–5996.

[44] X. Chen, A. Hassan Awadallah, H. Hassan, W. Wang, and C. Cardie, "Multi-source cross-lingual model transfer: Learning what to share," 2018, *arXiv:1810.03552*.

[45] J. Guo, D. J. Shah, and R. Barzilay, "Multi-source domain adaptation with mixture of experts," 2018, *arXiv:1809.02256*.

Boyuan Yang (Member, IEEE) received the B.A., M.A., and Ph.D. degrees in mechanical engineering from the School of Mechanical Engineering, Xi'an Jiaotong University, Xi'an, China, in 2013, 2015, and 2019, respectively.

He was a Research Associate with the School of Electrical and Electronic Engineering, The University of Manchester, Manchester, U.K. He joined the Center for Advanced Control and Smart Operations, Nanjing University, Nanjing, China, as an Associate Professor. His research interests include intelligent manufacturing, machine learning, condition monitoring, and fault diagnosis.



Jinyuan Zhang received the B.S. degree in automation from the School of Electrical and Information Engineering, Zhengzhou University, Zhengzhou, China, in 2018. He is currently pursuing the M.S. degree in artificial intelligence with the School of Artificial Intelligence, Nankai University, Tianjin, China.

His research interests include deep learning and transfer learning, with a special interest in applying these techniques to the fault diagnosis of mechanical systems.

Ruonan Liu (Senior Member, IEEE) received the B.S., M.S., and Ph.D. degrees from Xi'an Jiaotong University, Xi'an, China, in 2013, 2015, and 2019, respectively.

She was a Post-Doctoral Researcher with the School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA, in 2019, and an Alexander von Humboldt Fellow with the University of Duisburg-Essen, Duisburg, Germany, from 2022 to 2024. She is currently an Associate Professor with the Department of Automation, Shanghai Jiao Tong University, Shanghai, China. Her research interests include machine learning, intelligent manufacturing, and vision-language navigation.

Dr. Liu received the 2021 Outstanding Paper Award from IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, the Runner-Up Paper Award from IJCAI-W 2024, the Best Paper Award Finalist from IEEE ICARM 2024, and the Best Paper Award from RCAE 2024, and selected in the Young Elite Scientist Sponsorship Program by CAST in 2022.

Di Lin (Member, IEEE) received the bachelor's degree in software engineering from Sun Yat-sen University, Guangzhou, China, in 2012, and the Ph.D. degree from The Chinese University of Hong Kong, Hong Kong, in 2016.

He is currently an Associate Professor with the College of Intelligence and Computing, Tianjin University, Tianjin, China. His research interests include computer vision and machine learning.

Ping Li (Member, IEEE) received the Ph.D. degree in computer science and engineering from The Chinese University of Hong Kong, Hong Kong, in 2013.

He is currently a Research Assistant Professor with The Hong Kong Polytechnic University, Hong Kong. He has one image/video processing national invention patent and has excellent research project reported worldwide by ACM TechNews. His current research interests include image/video stylization, GPU acceleration, and creative media.

C. L. Philip Chen (Life Fellow, IEEE) received the M.S. degree from the University of Michigan at Ann Arbor, Ann Arbor, MI, USA, in 1985, and the Ph.D. degree in electrical engineering from Purdue University, West Lafayette, IN, USA, in 1988.

He is currently a Chair Professor and the Dean of the School of Computer Science and Engineering, South China University of Technology, Guangzhou, China. His current research interests include systems, cybernetics, and computational intelligence.

Dr. Chen is a fellow of AAAS, IAPR, CAA, and HKIE, and a member of Academia Europaea (AE), European Academy of Sciences and Arts (EASA), and the International Academy of Systems and Cybernetics Science (IASCYS). He was a recipient of the 2016 Outstanding Electrical and Computer Engineers Award from his alma mater, Purdue University, in 1988. He received the IEEE Norbert Wiener Award in 2018 for his contribution to systems and cybernetics, and machine learning. He was the Chair of TC 9.1 Economic and Business Systems of International Federation of Automatic Control from 2015 to 2017. He was the IEEE Systems, Man, and Cybernetics Society President from 2012 to 2013 and the Editor-in-Chief of IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS from 2014 to 2019. He is the Editor-in-Chief of IEEE TRANSACTIONS ON CYBERNETICS and an Associate Editor of IEEE TRANSACTIONS ON FUZZY SYSTEMS. He is the Vice President of Chinese Association of Automation (CAA). He is also a Highly Cited Researcher by Clarivate Analytics in 2018 and 2019.