Rolling bearing fault diagnosis under varying operating conditions using a convolutional-Transformer multi-alignment approach

Yiying Wang[1], **Fulu Sui**[2], **Xiaoling Li**[3], **Xiaoxin Zhang**[4], **Mingxian Liu**[5], **Chen Liu**[6], **Jie Wu**[7]
Handan Vocational College of Technology, Handan, China
[4]Corresponding author
E-mail: [1]*yatesyy@163.com*, [2]*18330763025@163.com*, [3]*wolxl0@126.com*, [4]*A17631230069@163.com*, [5]*210229258@qq.com*, [6]*L15136759793@163.com*, [7]*jiewu06@163.com*

Received 26 September 2025; accepted 28 December 2025; published online 21 February 2026
DOI https://doi.org/10.21595/jve.2025.25460

Abstract. In this study, a novel deep transfer learning method, termed the Universal Domain Alignment and Multi-level Alignment Network (UDAM-Net), is established to address the significant feature distribution discrepancy between the source and target domains caused by the difficulty of feature extraction for rolling bearings under complex operating conditions. In the feature extraction stage, a one-dimensional convolutional neural network is integrated with a Transformer-based multi-head attention mechanism to effectively enhance the feature representation capability of raw signals. Additionally, the complementary advantages of Joint Maximum Mean Discrepancy (JMMD), Multi-kernel Maximum Mean Discrepancy (MK-MMD), and a domain discriminator are explored by designing a collaborative mechanism combining distribution metrics and adversarial learning in conjunction with transfer learning. Furthermore, an uncertainty-based weighting strategy is introduced to adaptively adjust the loss function and dynamically balance the contributions of different alignment modules. Finally, the Adam optimizer is employed to accelerate model convergence. Experimental results on the CWRU public dataset and a laboratory-built test rig dataset demonstrate that UDAM-Net achieves classification accuracies of 96.67 % and 96.17 %, respectively. This study enlightens fault diagnosis based on feature extraction and cross-domain feature alignment.

Keywords: rolling bearing fault diagnosis, one-dimensional convolutional neural network, joint maximum mean discrepancy, multi-kernel maximum mean discrepancy, domain discriminator, transformer multi-head attention mechanism.

1. Introduction

While rolling bearings are critical components in rotating machinery, their wide range of use and harsh operating conditions make them one of the most failure-prone components in such practical industrial systems. Since mechanical equipment typically operates under varying working conditions, conducting rolling bearing fault diagnosis under complex operating conditions is of great significance for engineering applications [1, 2].

In traditional learning-based approaches, literature [3] utilized nonlinear dimensionality reduction methods, including Diffusion Maps, Locally Linear Embedding (LLE), and Autoencoders, to lessen signal dimensionality and extract features, followed by fault identification through GK clustering and k-medoids. literature [4] proposed a CEEMDAN-IWSO-VMD-based denoising method, in which secondary signal decomposition was performed with CEEMDAN and VMD, and statistical features were subsequently extracted and fed into a BiLSTM network for fault identification. However, the above-mentioned methods rely heavily on signal decomposition for denoising and fault feature enhancement under practical operating conditions, hindering the direct extraction of discriminative features from raw vibration signals. Moreover, these methods strongly depend on manual parameter tuning, degrading feature quality and leading to issues such as mode mixing, over-decomposition, and high computational complexity. This limits their applicability to complex signals under varying operating conditions.

To overcome the aforementioned limitations, literature [5] leveraged the advantages of convolutional neural networks (CNN) in spatial feature extraction, noise robustness, and computational efficiency to improve the accuracy of bearing fault diagnosis. literature [6] proposed a hybrid method combining CNNs with long short-term memory (LSTM) networks to capture temporal dependencies in the data, followed by fault diagnosis through an optimized least squares support vector machine. While CNNs can automatically extract local features and reduce the reliance on manual feature engineering, their receptive fields are constrained by kernel size and network depth, rendering it difficult to capture long-range global temporal dependencies. Existing studies have combined CNNs with LSTM or GRU networks, whereas such architectures generally suffer from low training efficiency when handling long sequences and features under complex operating conditions. In contrast, the self-attention mechanism of the Transformer allows for the computation of dependency weights between arbitrary positions, enabling the integration of global temporal features.

For example, literature [7] first mapped time-domain vibration signals into the frequency domain using the fast Fourier transform (FFT) and employed CNNs to process time-frequency representations and extract local impulsive patterns and frequency energy distributions. Moreover, they captured long-term dependencies following the Transformer's multi-head attention mechanism, thereby improving diagnostic accuracy. With respect to feature extraction, literature [8] constructed a multi-branch Transformer network to extract global features from different perspectives and achieved fault diagnosis under noisy environments through a multi-supervision strategy combined with local feature enhancement. literature [9] decomposed raw signals through variational mode decomposition (VMD) to remove noise, extracted multi-scale features via CNNs, and employed a Transformer to capture long-range dependencies for fault diagnosis. In practical machinery operation, nevertheless, CNN-Transformer-based methods primarily focus on the fusion of local and global feature representations, and their feature alignment capability remains limited. Consequently, it is difficult to capture feature discrepancies across different samples, resulting in poor feature alignment under varying operating conditions.

Transfer learning, which is an effective approach for addressing varying operating conditions, aims to transfer knowledge from the source domain to the target domain for reducing the discrepancy between them [10]. The literature [11] employed an improved convolutional neural network to extract multi-scale features and utilized multi-kernel maximum mean discrepancy (MK-MMD) to reduce feature discrepancies between the source and target domains, contributing to the improved accuracy of cross-domain feature discrimination, literature [12] combined CNNs with an attention mechanism to enhance signal feature representation and addressed domain shift through JMMD, literature [13] applied EMD for denoising and achieved feature alignment by integrating CNNs and attention mechanisms with a domain discriminator. Although MK-MMD, JMMD, and domain discriminators can extract domain-invariant features and enhance model robustness, each of them faces inherent challenges. Specifically, JMMD suffers from high computational complexity and slow training speed, and is sensitive to multi-scale feature representations and kernel selection; MK-MMD improves flexibility but only aligns marginal distributions while neglecting conditional distributions; domain discriminators are prone to training instability or gradient vanishing, bringing about the weakened discriminative capability of learned features.

In this study, the self-attention mechanism of the Transformer is introduced to address the performance degradation of conventional CNN-based methods under varying operating conditions triggered by limited receptive fields and insufficient long-range dependency modeling. By employing a single-layer architecture, a global receptive field is achieved while noisy features are dynamically suppressed, enabling the enhancement of feature extraction capability and more robust deep representations of raw signals. With the purpose of mitigating the large feature discrepancies between the source and target domains, a joint alignment strategy integrating JMMD and MK-MMD is proposed, together with a domain discriminator for domain feature alignment.

This approach overcomes the limitations of conventional metrics in computational efficiency, conditional distribution alignment, and training stability. Furthermore, the label classification loss, domain alignment loss, JMMD loss, and MK-MMD loss are dynamically balanced following an uncertainty-adaptive weighting mechanism to alleviate the issue of inconsistent loss scales and lower reliance on manual hyperparameter tuning. Furthermore, the Adam optimizer is adopted to alleviate the limitations of traditional optimization methods concerning gradient noise, sparsity, and learning rate adjustment, thereby improving training stability and convergence speed. Based on these improvements, a novel deep transfer learning framework, termed the UDAM-Net, is constructed in this study, providing a new solution for cross-domain fault diagnosis of rolling bearings.

The main contributions of this study are summarized as follows:

(1) A multi-scale joint feature extraction framework is constructed by incorporating a Transformer layer on top of a one-dimensional convolutional neural network. By integrating local temporal features with global dependency modeling, the proposed network improves long-range feature capture and alleviates insufficient feature representation under varying operating conditions.

(2) JMMD, MK-MMD, and a domain discriminator are jointly integrated to achieve multi-dimensional alignment of marginal distributions, conditional distributions, and discriminative features, contributing to mitigated distribution shift between domains and enhanced model generalization capability.

(3) An uncertainty-adaptive loss weighting strategy is employed to dynamically allocate weights among different loss components, avoiding imbalance stemming from manual hyperparameter tuning. In conjunction with the Adam optimizer, training stability and convergence speed are further improved, allowing the model to maintain robustness under noisy and cross-operating conditions.

2. Fault diagnosis model

In this study, a transfer learning-based rolling bearing fault diagnosis model is designed for varying operating conditions. First, a Transformer module is introduced on top of a 1D-CNN to reduce the excessive time consumption and manual parameter tuning arising from reliance on handcrafted feature extraction, enabling direct feature learning from raw vibration signals. Meanwhile, transfer learning is incorporated by combining explicit distance-based metrics with implicit adversarial alignment mechanisms. In this way, a cross-domain feature alignment framework is constructed, achieving class discriminability, multi-scale global feature alignment, and dynamic adaptability. Finally, an uncertainty-adaptive weighting strategy is applied to the label classification loss, JMMD loss, MK-MMD loss, and domain discriminator loss, and the Adam optimizer is employed to enhance training stability and convergence speed. The overall architecture of the proposed model is illustrated in Fig. 1. The corresponding pseudocode is detailed in Table 1.

2.1. Fault recognition classifier

This study adopts a feature extractor that combines a 1D-CNN with a Transformer. The 1D-CNN is effective at capturing local patterns in time-series signals and highlights features such as edges, peaks, and harmonic components of fault signals. Nevertheless, its receptive field is inherently limited. The Transformer can model long-range dependencies and attend to periodic patterns and strongly correlated regions from a global perspective. Unfortunately, it may lose fine-grained local information and typically requires large amounts of training data. With the purpose of handling these issues, the Transformer layer is placed after the convolution and pooling operations of the 1D-CNN, which preserves local details while enhancing global modeling capability. Additionally, sequence down-sampling is employed to lower the computational cost of the attention mechanism, allowing for efficient and comprehensive feature extraction.

During the feature extraction stage, the basic 1D-CNN–Transformer architecture consists of an input layer, convolutional layers, pooling layers, a Transformer layer, fully connected layers, and an output layer (Fig. 1). The convolutional layers contain multiple convolutional units that perform local receptive field operations on the input data to extract salient temporal local features.

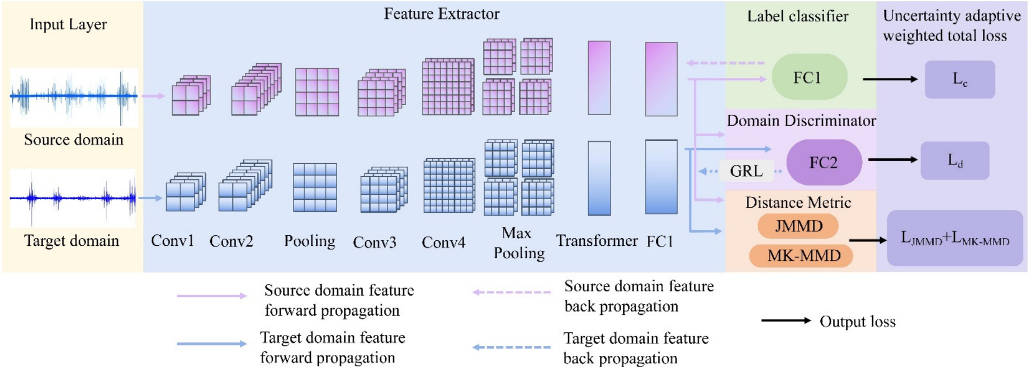


Fig. 1. Model network structure diagram

Table 1. Pseudo-code of the model

```
Algorithm: universal domain alignment and multi-level alignment network
Require: Source set $D_s = x^s, y^s$, target set $D_t = x^s$, class number $K$; trade-off weights $\lambda_1\lambda_2$;
Learnable uncertainty scales $\sigma_c\sigma_d\sigma_a, \sigma_c^2, \sigma_d^2, \sigma_a^2 > 0$; GRL schedule parameter $\gamma$
Initialize feature extractor $\theta_f$, classifier $\theta_y$, domain discriminator $\theta_d$, learnable logs, training progress $p \leftarrow 0$
While not converged do
Sample minibatches $x^s, y^s \sim D_s, x^s \sim D_t$
Feature extraction: $f_s, J_s \leftarrow G_f(x_s; \theta_f), f_t, J_t \leftarrow G_f(x_t; \theta_f)$
Label prediction: $z_s \leftarrow G_y\left(f_s; \theta_y\right), L_c \leftarrow CrossEntropy(Z_s, y_s)$
Domain discrimination with GRL: $d_s \leftarrow G_d(GRL(f_s); \theta_d), d_t \leftarrow G_d(GRL(f_t); \theta_d), L_d \leftarrow BCE(d_s, 0) + BCE(d_t, 1)$
Multi-level distribution alignment: $D_J \leftarrow JMMD(J_s, J_t), D_M \leftarrow MKMMD(f_s, f_t), D_{fussdion} \leftarrow \lambda_1 D_J + \lambda_2 D_M$
Dynamic adversarial weight ($\beta_{adv}$ with progress $p \in [0,1]$), $\beta_{adv} \leftarrow \frac{2}{1+exp\,(-\gamma p)} - 1$
Uncertainty-weighted total loss: $L_{total} \leftarrow \frac{L_c}{2\sigma_c^2} + \frac{\beta_{adb}}{2\sigma_d^2} L_d + \frac{D_{fusion}}{2\sigma_a^2} + \log\sigma_c + \log\sigma_d + \log\sigma_a$
One optimization step $\theta_f, \theta_y, \theta_d \leftarrow$ Adam Update $\left\{\theta_f, \theta_y, \theta_d\right\}, \nabla L_{total}$
Update training progress: $p \leftarrow$ UpdateProgess ()
end while
Return $\theta = \{\theta_f, \theta_y, \theta_d\}$
```

(1) Input layer.

The input layer receives a vibration signal of length L, expressed as:

$$S = \{S_1, S_2, \cdots, S_L\}, \quad S \in R^{L \times 1}, \tag{1}$$

where each s_i denotes the vibration signal value at the i-th sampling point.

(2) Convolution operation.

The convolutional layer is employed for feature learning and local feature extraction. One-dimensional convolution learns features by convolving a kernel with the j-th segmented signal. The input sequence is denoted as $s^{(l-1)}$, and it is supposed that the kernel size, kernel weights, and bias of the l-th convolutional layer are $k^{(l)}$, $\omega^{(l)}$ and $b^{(l)}$, respectively. Then, the output of the convolutional layer can be expressed as:

$$c_j = RELU\left(\sum_{i=1}^{k^{(l)}} \omega_i^{(l)} \times s_{j-k^{(l)}}^{(l-1)} + b^{(l)}\right), \tag{2}$$

where ReLU denotes the nonlinear activation function; c_j represents the j-th element of the convolutional output; multi-scale feature extraction is achieved across different layers by adjusting the kernel size $k^{(l)}$.

(3) Pooling operation.

A max-pooling layer is applied after the convolutional layer to reduce feature dimensionality and prevent overfitting:

$$P_j = \max\{c_{j-k+1:j}\}, \tag{3}$$

where P_j signifies the pooled feature, and $c_{j-k+1:j}$ represents a local segment with window size k ending at position j. In this study, a pooling window size of $k = 2$ is adopted. It effectively halves the feature length and reduces computational complexity while preserving salient local features during dimensionality reduction, avoiding the loss of fault-related impulsive information caused by excessive pooling [14].

(4) Transformer attention mechanism.

The Transformer model initially demonstrated outstanding performance in the field of natural language processing and has been increasingly introduced into various time-series signal modeling tasks. Its core concept lies in the Multi-Head Self-Attention (MHSA). MHSA captures global dependencies between different positions in a sequence through parallelized attention computations, thereby effectively compensating for the limitations of traditional convolutional or recurrent architectures in long-range dependency modeling [15].

A self-attention mechanism is introduced on the feature map $P_j \in R^{L\times d_p}$ obtained after convolution and pooling to further capture long-range dependencies across different time steps. Specifically, L denotes the down-sampled sequence length, and d_p represents the number of feature channels. In the attention mechanism, the input features are first linearly projected into a query vector Q, a key vector K, and a value vector V, where Q is used to issue queries, K to measure relevance, and V to represent the actual feature information. The attention weights are computed based on the similarity between Q and K and then employed to perform a weighted aggregation over V, enabling dynamic feature aggregation:

$$Q = PW_Q, \quad K = PW_K, \quad V = PW_V, \tag{4}$$

where W_Q, W_K, $W_V \in R^{d_p\times d_k}$ embody learnable parameter matrices, and d_k signifies the internal dimensionality of the attention mechanism.

The self-attention is computed as:

$$\text{Attention}(Q,K,V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V. \tag{5}$$

Under the multi-head attention mechanism, let the number of attention heads be h, and then the i-th attention head is defined as:

$$head_i = Attention\left(QW_Q^i, KW_K^i, VW_V^i\right). \tag{6}$$

After concatenating all attention heads, the output is obtained as:

$$MultiHead(Q,K,V) = Concat(head_1,\cdots,head_h)W_0, \tag{7}$$

where $W_0 \in R^{hd_v \times d}$ indicates the output projection matrix. The number of heads in the multi-head attention mechanism exerts a significant impact on model performance and is typically set to 2, 4, or 8 in practice. An excessively large number of heads may boost the risk of overfitting and slow down convergence, whereas too few heads may induce underfitting. In this study, the number of attention heads is set to 4 [16], and the dimensionality of each head is defined as $d_k = d_v = d/4$, where d_v represents the dimensionality of the value vector in each attention head.

(5) Fully connected layer.

The feature representation $f \in R^d$, obtained from the multi-head attention mechanism is fed into a fully connected layer:

$$f' = \sigma\left(W_f^T f + b_f\right), \quad W_f \in R^{d \times d_{fc}}, \tag{8}$$

where $\sigma(\cdot)$ denotes the activation function, and the ReLU activation function is adopted in this study; b_f represents the bias term; d_{fc} signifies the output dimensionality of the fully connected layer.

(6) Output layer.

Finally, the output layer employs the Softmax function to transform the outputs of the fully connected layer into a probability distribution, enabling fault category classification:

$$y_c = \frac{\exp\,(W_c^T f' + b_c)}{\sum_{k=1}^{K} \exp W_k^T f' + b_k)}, \quad c = 1,2,\cdots,K, \tag{9}$$

where K denotes the total number of fault categories; W_c and b_c are the class-specific weight vector and bias term for class c, respectively; b_c represents the bias associated with class c, and b_k embodies the bias term corresponding to class kin the summation. The output vector $y = [y_1, y_2, \cdots y_k]$ provides the predicted probabilities for all fault categories. The final classification result is determined by selecting the category with the maximum probability.

2.2. Domain adaptive module

The domain adaptation module of the proposed model consists of a JMMD-based distance metric, an MK-MMD-based distance metric, and a domain discriminator module.

2.2.1. Feature distribution difference measurement module

Under varying operating conditions, a single distribution alignment method is typically insufficient to simultaneously ensure cross-layer joint distribution consistency and capture multi-scale distribution discrepancies, consequently degrading diagnostic accuracy. Given this issue, JMMD and MK-MMD are integrated in this study. The former constrains the joint distributions of multi-layer features; the latter enhances robustness to discrepancies at different scales through multi-kernel weighting. They jointly curtail feature shifts between the source and target domains in both marginal and conditional distributions. Compared with the conventional single Maximum Mean Discrepancy method, JMMD [17] integrates multi-level feature mappings to more comprehensively characterize and lessen distribution discrepancies across different domains, enabling the optimization of the joint distributions of multiple feature representations and enhancement of feature extraction and domain alignment performance [18].

In the fault identification classifier, deep feature vectors, f_1^s and f_1^t, are first extracted from the source and target domains through the feature extraction network. Subsequently, they are further projected into a unified feature space through the fully connected layer FC1. With the purpose of achieving cross-domain distribution alignment, the feature representations of the source and target domains in this space are denoted as $\phi(f_1^s)$ and $\phi(f_1^t)$, respectively, where $\phi(\cdot)$ embodies a kernel mapping function that projects features into a reproducing kernel Hilbert space (RKHS). JMMD enforces collaborative consistency between the source and target domains across both convolutional features and high-level semantic features, preventing situations in which lower-level features are well aligned while substantial discrepancies remain at higher levels. Thus, discriminative capability can be reinforced under varying operating conditions.

The corresponding formulation is:

$$JMMD(P,Q) = \sum\nolimits_{l=1}^{L} \left\| E_{f_1^s}{\sim}P\left[\phi^{(l)}(f_1^s)\right] - E_{f_1^t}{\sim}Q\left[\phi^{(l)}(f_1^t)\right] \right\|_{H_l}^2, \tag{10}$$

where P and Q denote the feature distributions of the source and target domains, respectively; $\phi^{(l)}(\cdot)$ indicates the feature mapping function of the l-th layer that maps features into a reproducing kernel Hilbert space for computing mean embeddings; f_1^s, f_1^t represent the feature representations of the source and target domains; H_l embodies the RKHS associated with the l-th layer; $\|\cdot\|_{H_l}$ refers to the norm defined in the Hilbert space H_l.

Multi-kernel Maximum Mean Discrepancy (MK-MMD) [19] indicates an extension of the conventional MMD that further enhances distribution alignment capability. MMD characterizes the distribution discrepancy between the source and target domains by mapping samples into a RKHS and computing the distance between their mean embeddings [20], defined as:

$$d_k^2(p,q) = \left\| E_{f_1^s}{\sim}P[\phi_k(f_1^s)] - E_{f_1^t}{\sim}Q[\phi_k(f_1^t)] \right\|_{H_l}^2, \tag{11}$$

where $\phi_k(\cdot)$ denotes the feature mapping associated with the k-th kernel function into a RKHS; H_k represents the RKHS induced by the k-th kernel; $d_k^2(p,q) = 0$ and only if the distributions p and q are identical.

In practical applications, the choice of a single kernel function can significantly impact alignment performance. Since MK-MMD constructs multiple kernel functions at different scales and combines them through weighted summation, the model can capture both global and local feature variations when measuring distribution discrepancies. Accordingly, MK-MMD constructs a composite kernel through a non-negative convex combination of multiple kernels:

$$K(f_1^s, f_1^t) = \sum\nolimits_{u=1}^{m} \beta_u k_u(f_1^s, f_1^t), \ \text{subject to} \sum\nolimits_{u=1}^{m} \beta_u = 1, \quad \beta_{u\geq 0}, \tag{12}$$

where $k(\cdot,\cdot)$ denotes the u-th base kernel function, β_u represents the weight of the u-th kernel, and m denotes the number of kernel functions. This strategy alleviates the sensitivity to single-kernel selection while fully exploiting the advantages of different kernel functions at multiple scales. Therefore, combining Eqs. (10) and (12) yields the resulting distance metric as:

$$\begin{cases} D_{JMMD} = \sum\nolimits_{l=1}^{L} \left\| E_{f_1^s}{\sim}P\left[\phi^{(l)}(f_1^s)\right] - E_{f_1^t}{\sim}Q\left[\phi^{(l)}(f_1^t)\right] \right\|_{H_l}^2, \\ D_{MK-MMD} = \left\| E_{f_1^s}{\sim}P[\phi_K(f_1^s)] - E_{f_1^t}{\sim}Q[\phi_K(f_1^t)] \right\|_{H_l}^2, \\ \qquad \phi_K(f) = \sum\limits_{i=1}^{m} \alpha_i \phi_{k_i}(f), \quad \sum\limits_{i=1}^{m} \alpha_i = 1, \quad \alpha_i \geq 0, \\ D_{\text{fussion}} = \lambda_1 D_{JMMD} + \lambda_2 D_{MK-MMD}, \end{cases} \tag{13}$$

where L denotes the number of feature layers jointly aligned by JMMD; $\phi_{k_i}(f)$ represents the mapping of the feature vector f induced by the i-th base kernel; $\phi_K(f)$ signifies the composite multi-kernel mapping used in MK-MMD; α_i refers to the weight of the i-th kernel function; λ_1, λ_2 are the balancing coefficients for integrating JMMD and MK-MMD, respectively. Through the fusion of JMMD and MK-MMD, the model aligns the joint feature distributions between the source and target domains while simultaneously capturing distribution discrepancies at multiple scales, thereby alleviating distribution shift under varying operating conditions.

2.2.2. Domain discriminator

After feature extraction of bearing vibration signals with a convolutional neural network, the feature distributions under different operating conditions may differ significantly. As a result, the features learned from the source condition become difficult to transfer directly to the target condition. Given this issue, a domain discriminator is introduced after the integration of JMMD and MK-MMD, and an adversarial framework is constructed by incorporating a gradient reversal layer (GRL) [21]. The domain discriminator attempts to identify the operating condition from which the input features originate, while the feature extractor is continuously optimized during training to confuse the discriminator. Thus, it learns more domain-invariant feature representations and improves cross-condition fault diagnosis performance.

The domain classifier is primarily composed of two components: a fully connected layer (FC2) and a domain discriminator output layer (DO). These two layers jointly ensure that the network can effectively distinguish data features from different domains.

Fully connected layer FC2: FC2 receives the feature representation f_1 from FC1 and produces a new feature representation f_2 through linear transformation and nonlinear activation:

$$f_2 = \sigma\left(W_f^T f_1 + b_f\right), \tag{14}$$

where f_1 denotes the sigmoid activation function; W_f and b_f refer to the weight matrix and bias term of this layer, respectively.

Domain discriminator output layer (DO) is a binary classifier specifically designed to distinguish whether the input features belong to the source domain or the target domain. The output of this layer, denoted as d, is given by:

$$d = \frac{1}{1 + \exp\left(-(W_d^T f_2 + b_f\right)}, \tag{15}$$

where W_d denotes the output weight matrix of the domain discriminator, and b_d represents the corresponding bias term. Therefore, the output $d \in (0,1)$ can be interpreted as the probability that the input feature belongs to the source or target domain. Through the adversarial training mechanism, the domain discriminator continuously enhances its capability to distinguish features from different domains. Meanwhile, the feature extractor, under the effect of reversed gradients, gradually learns more domain-invariant feature representations. Consequently, it achieves multi-dimensional alignment of discriminative features, alleviates distribution shift between domains, and effectively reduces cross-domain distribution discrepancies under varying operating conditions.

2.2.3. Definition of loss function

The source domain dataset is denoted as $D_s = \{(x_i^s, y_i^s)\}_{i=1}^{n_s}$, where x_i^s represents the i-th source-domain input sample, y_i^s signifies the corresponding label, and n_s embodies the number of source-domain samples. Besides, the target domain dataset is defined as $D_t = \left\{(x_j^t)\right\}_{j=1}^{n_t}$, where x_j^t and n_t refer to the j-th target-domain input sample and the number of target-domain samples, respectively. The feature extractor is denoted as $G_f(\cdot)$, which is used to extract features from the input data, and its output is defined as $f_1(x) = G_f(x)$ with the corresponding source-domain and target-domain feature representations given by $f_1^s = G_f(x_i^s)$, $f_1^t = G_f\left(x_j^t\right)$. The label classifier is denoted as $G_y(\cdot)$, whose logits output is given by $z(x) = W_c^T f_1(x) + b_c$, and the Softmax probability vector is defined as $p(x) = \text{softmax}(z(x))$.

(1) Classification loss L_c: a multi-class cross-entropy loss applied to labeled source-domain samples, which constrains the classification capability of the classifier on the source domain, defined as:

$$L_c = -\frac{1}{m}\sum_{i=1}^{m}\sum_{k=1}^{K} I[y_i^s = k]\log\frac{\exp\left(W_c^{(k)T} f_1(x_i^s) + b_c^{(k)}\right)}{\sum_{j=1}^{K}\exp\left(W_c^{(j)T} f_1(x_i^s) + b_c^{(j)}\right)}, \tag{16}$$

where $f_1(x_i^s) = G_f(x_i^s) \in R^d$ denotes the feature vector extracted by the feature extractor $G_f(\cdot)$, with dimensionality d; $W_c = [W_c^{(1)}, \cdots, W_c^{(k)}] \in R^{d\times K}$ indicates the classifier weight matrix, and $W_c^{(k)}$ stands for the weight column vector corresponding to the k-th class; $b_c = [b_c^{(1)}, \cdots, b_c^{(k)}]^T$ represents the bias vector, and each element, $b_c^{(k)}$, corresponds to the bias term of the k-th class; $I(\cdot)$ embodies the indicator function, and $I[y_i^s = k] = 1$ if $y_i^s = k$ and 0 otherwise; K signifies the total number of fault categories, and m denotes the mini-batch size of the source domain. The fraction in the formulation is related to the Softmax probability, and the predicted probability of the k-th class $p_k(x_i^s)$ is defined as:

$$p_k(x_i^s) = \frac{\exp\left(W_c^{(k)T} f_1(x_i^s) + b_c^{(k)}\right)}{\sum_{j=1}^{K}\exp\left(W_c^{(j)T} f_1(x_i^s) + b_c^{(j)}\right)}, \tag{17}$$

where $p_k(x_i^s)$ denotes the predicted probability that the input sample x_i^s belongs to the k-th class.

(2) Domain discrimination loss L_d: The domain discrimination loss is employed to guide adversarial training between the feature extractor and the domain discriminator, allowing for the learning of domain-invariant features. Binary cross-entropy loss is computed for both source-domain and target-domain samples through the domain discriminator. The feature extractor and the domain discriminator are jointly optimized in an adversarial manner by computing the binary cross-entropy loss for source and target samples separately. Let $\hat{d} = \sigma\left(G_d\left(GRL(F)\right)\right)$, and then the loss for a single sample is defined as:

$$l_d = (f, d) = -\left[d\log\hat{d} + (1-d)\log\left(1-\hat{d}\right)\right], \tag{18}$$

where $\hat{d}$ denotes the model output; d represents the domain label, 0 or 1; they correspond to the source domain and the target domain, respectively. Accordingly, the domain discrimination loss L_d is defined as the average loss over source-domain and target-domain samples:

$$L_d = \frac{1}{n_s}\sum_{i=1}^{n_s} L_d\left(f_1^{s_i}, 0\right) + \frac{1}{n_t}\sum_{j=1}^{n_t} L_d\left(f_1^{s_j}, 1\right), \tag{19}$$

where $f_1^{s_i}$ and $f_1^{s_j}$ represent the feature vectors of the source-domain and target-domain samples, respectively, extracted by the feature extractor G_f; $L_d(\cdot)$ denotes the binary cross-entropy loss for a single sample, measuring the discrepancy between the domain discriminator output $\hat{d}$ and the domain label d; n_s, n_t signify the numbers of samples in the source and target domains, respectively; G_d indicates the domain discriminator, which aims to distinguish whether a sample belongs to the source or target domain by minimizing the loss L_d during training; GRL stands for the gradient reversal layer, which reverses the gradient propagated to the feature extractor G_f during backpropagation, thereby encouraging G_f to learn domain-invariant feature representations.

(3) Distribution distance loss D_{fusion}: With the purpose of further aligning the feature distributions between the source and target domains, the JMMD and MK-MMD losses are integrated into a unified metric for measuring the discrepancy between the joint feature distributions of the source and target domains. Through this fusion, the model can align multi-level feature distributions between the source and target domains while simultaneously capturing distribution discrepancies at multiple scales, thereby alleviating distribution shift stemming from varying operating conditions. According to Eq. (13), the final fused loss is defined as:

$$D_{fusion} = \lambda_1 D_{JMMD} + \lambda_2 D_{MK-MMD}, \tag{20}$$

where D_{JMMD} and D_{MK-MMD} denote the JMMD and MK-MMD loss terms, respectively; λ_1, λ_2 control the relative weights of the two metrics.

(4) Uncertainty-Adaptive Weighted Overall Loss: An uncertainty-based adaptive weighting strategy is introduced [22] to avoid manual hyperparameter tuning. Specifically, three objectives, namely the classification loss L_c, the domain discrimination loss L_d, and the distribution distance loss D_{fusion}, are each associated with a learnable uncertainty scale satisfying $\sigma_c^2, \sigma_d^2, \sigma_a^2 > 0$. The model is trained by minimizing the total loss L_{total} to jointly improve domain adaptation and classification accuracy. The overall loss is defined as:

$$L_{total} = \frac{1}{2\sigma_c^2} L_c + \frac{\beta_{adb}}{2\sigma_d^2} L_d + \frac{1}{2\sigma_a^2} D_{fusion} + \log\sigma_c + \log\sigma_d + \log\sigma_a \tag{21}$$

where σ_c^2, σ_d^2 and σ_a^2 are learnable variances associated with the classification loss, domain discrimination loss, and distribution alignment loss, respectively; $\log\sigma_d$, $\log\sigma_a$, $\log\sigma_c$ denote regularization terms that constrain the magnitude of the variances and prevent excessive adjustment; β_{adv} represents the adversarial training strength, with $\beta_{adv} \in (0,1]$, which is progressively increased with the training progress $p \in [0,1]$ following $\beta_{adv} = \frac{2}{1+\exp(-\gamma p)} - 1$ to stabilize adversarial training

3. Test verification and result analysis

3.1. Western Reserve University data validation

The rolling bearing vibration signal dataset used in this study was obtained from the Electrical Engineering Laboratory of Case Western Reserve University (CWRU), USA [23]. During the experiments, the vibration signals of rolling bearings were recorded as multivariate vibration time series and collected by a data acquisition system at sampling frequencies of 12 kHz and 48 kHz, among which the 12 kHz sampling frequency was adopted in this work. Rolling bearing datasets under four operating conditions were obtained as per different combinations of rotational speed and load. Each operating condition in the CWRU dataset includes four bearing states: normal (N), inner race fault (IF), outer race fault (OF), and ball fault (BF), as detailed in Table 2. Considering both bearing condition and fault size, each operating condition contains ten bearing states: N, IF18, BF18, OF18, IF36, BF36, OF36, IF54, BF54, and OF54. Concerning data preparation, the signals were segmented into approximately 100 samples per class, with each sample having a length of 1024 points. With respect to each operating condition, 80 % of the samples were employed for training and the remaining 20 % for testing. Additionally, the number of training epochs was set to 99, the Adam optimizer was applied, the initial learning rate was set to 0.001, and the batch size was set to 64.

Table 2. Specific data of working conditions

Condition / domain	revolution speed (r/min)	Load (HP)	Fault size
0hp	1797	0	0.1778 0.3556 0.5334
1hp	1772	1	
2hp	1750	2	
3hp	1730	3	

3.1.1. Different training batch tests

Under varying operating conditions, the batch size setting influences the estimation accuracy of alignment losses, the convergence behavior of distribution distances between the source and target domains, and the stability of domain adversarial training. Thus, comparative experiments were conducted on the proposed UDAM-Net model across ten transfer tasks to investigate this influence, and the average results were reported. The batch size was set to 32, 50, 64, 90, and 128, respectively. The experimental results are summarized in Table 3.

Table 3. Average experimental results of different training batches of 10 migration tasks

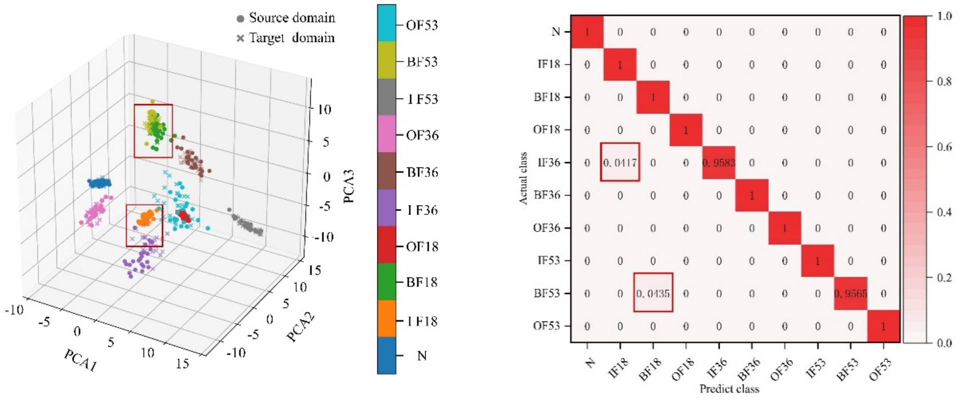
Batch	Accuracy	Precision	Recall	F1-score
32	95.16 %	94.00 %	95.00 %	94.50 %
50	94.33 %	90.66 %	92.00 %	90.67 %
64	96.67 %	96.67 %	96.00 %	96.00 %
90	96.33 %	96.33 %	96.00 %	96.00 %
128	95.83 %	95.17 %	95.75 %	95.46 %

As observed from Table 3, the UDAM-Net model achieves its best performance when the batch size is set to 64. Across the 12 transfer tasks, the average diagnostic accuracy, precision, recall, and F1-score reach 96.67 %, 96.67 %, 96 %, and 96 %, respectively, all of which are higher than those obtained with the other four batch sizes. Thus, the training batch size is uniformly set to 64 in all subsequent experiments, ensuring model stability and optimal performance.

3.1.2. Contrast test

Comparative experiments were conducted to verify the effectiveness of the proposed UDAM-Net in reducing the feature distribution discrepancy between the source and target domains under variable operating conditions. The UDAM-Net model was compared with 1DCNN-Transformer, CORAL, KL divergence, Proxy-A distance, as well as the existing methods CNN-BiLSTM-Transformer [24] and CNN-MDD [25]. The performance was evaluated through four metrics: accuracy, precision, recall, and F1-score. The results, averaged over 12 transfer tasks, are summarized in Table 4. With the 1 hp to 2 hp operating condition as an example, Fig. 2(a) illustrates PCA-based feature clustering, where high-dimensional features are projected into a three-dimensional space for visualization. Fig. 2(b) exhibits the corresponding confusion matrix. Table 4 lists the averaged experimental results.

As illustrated in Fig. 2(a), the distribution discrepancy between the source and target domain features has been largely reduced, with only a small portion of features failing to fully overlap. Fig. 2(b) reflects that 4.17 % of IF36 samples are misclassified as IF18, and BF53 samples are misclassified as IF53, while all other fault types are correctly identified. As revealed from Table 4, UDAM-Net outperforms the other methods across all four evaluation metrics. Compared with the baseline models CNN-BiLSTM-Transformer and CNN-MDD, UDAM-Net achieves accuracy improvements of 3.16 % and 6.41 %, respectively, demonstrating its advantage in achieving higher diagnostic accuracy.



a) Feature clustering results

b) Confusion matrix

Fig. 2. Fig. 5(a) Feature clustering results and confusion matrix

Table 4. Comparison of model index results

	Accuracy	Precision	Recall	F1-Score
1DCNN-Transformer	93.75 %	93.41 %	91.89 %	92.65 %
CORAL	94.00 %	93.75 %	93.08 %	93.42 %
KL	73.83 %	79.00 %	72.75 %	75.88 %
Proxy a distance	90.10 %	92.34 %	87.67 %	90.01 %
CNN-BiLSTM-Transformer	93.51 %	90.00 %	93.00 %	91.50 %
CNN-MDD	90.26 %	94.00 %	91.00 %	92.50 %
UDAM-Net	96.67 %	96.67 %	96.00 %	96.34 %

3.1.3. Ablation test

The effects of 1DCNN-Transformer, JMMD, MK-MMD, JMMD–MK-MMD, and UDAM-Net without the Transformer layer (denoted as UDAM-Net (none)) on overall model performance were investigated to evaluate the adaptability of different modules in the proposed model under variable operating conditions and their contributions to fault diagnosis performance. The performance is evaluated with accuracy, precision, recall, and F1-score. The experimental results are presented in Table 5.

Table 5. Comparison results of the ablation test model indexes

	Accuracy	Precision	Recall	F1-score
1DCNN-Transformer	92.33 %	92.99 %	92.08 %	92.54 %
JMMD	93.75 %	93.41 %	91.89 %	91.83 %
MKMMD	94.08 %	93.58 %	94.00 %	93.79 %
JMMD-MK-MMD	95.83 %	95.33 %	95.75 %	95.54 %
UDAM-Net	96.67 %	96.67 %	96.00 %	96.00 %
UDAM-Net(none)	89.67 %	89.00 %	88.50 %	88.75 %

As suggested in Table 5, the model accuracy increases by 1.42 % and 1.75 % after the introduction of JMMD and MK-MMD, respectively. The accuracy improvement reaches 3.5 % when the two metrics are combined. Furthermore, the model accuracy is further improved to 96.67 % after the introduction of the domain discriminator. In contrast, removing the Transformer layer yields a significant accuracy decrease of 7 %, specifying the importance of the Transformer-based self-attention mechanism for feature extraction under complex operating conditions. Thus, the 1D-CNN, JMMD–MK-MMD, domain discriminator, and Transformer modules play critical roles in enhancing overall model performance.

3.1.4. Optimizer comparison test

An optimizer comparison experiment was conducted to address the challenges of accelerating model convergence, improving training stability, and avoiding local optima under varying operating conditions. Considering the effects of distribution discrepancy and gradient fluctuations during training, Adam and SGD optimizers were selected for comparison, with the learning rate set to 0.001 for both. Among the 12 transfer tasks, three representative tasks (1 hp to 2 hp, 1 hp to 3 hp, and 2 hp to 1 hp) were selected as examples to evaluate their effects on model convergence speed, training stability, and final transfer capability. The corresponding accuracy and loss curves are illustrated in Fig. 3.

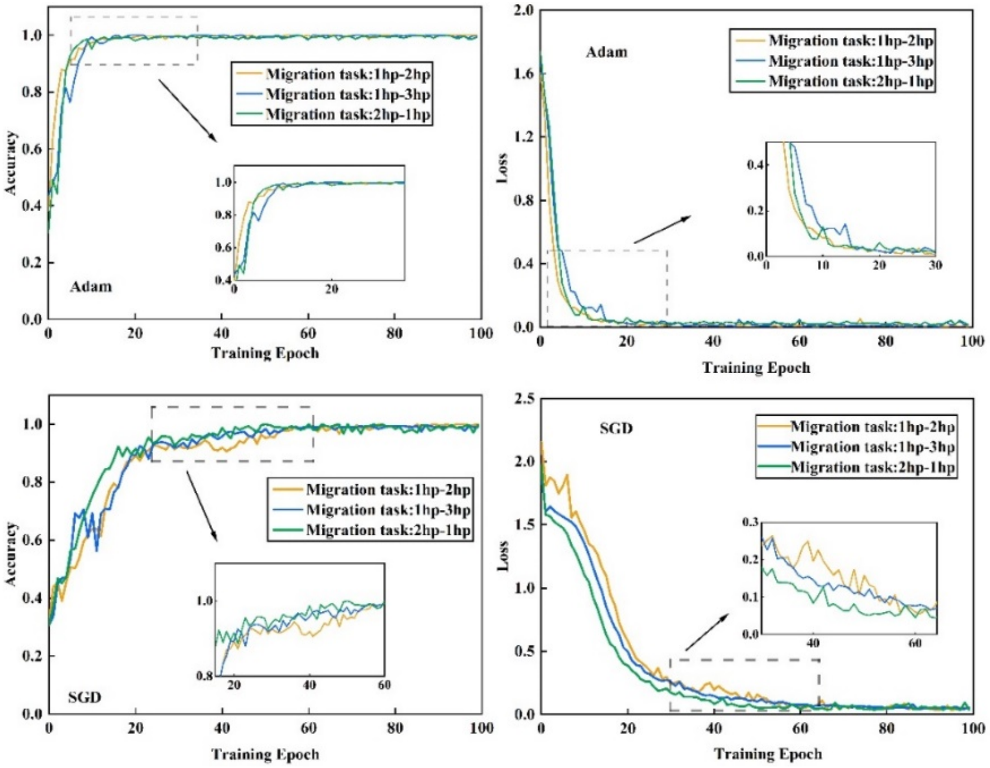


Fig. 3. Optimizer sparrow rate and loss rate comparison

Fig. 5 reveals that, for the selected transfer tasks, the model optimized with Adam achieves stable accuracy and loss values after approximately 20 iterations. In contrast, the model using the SGD optimizer requires nearly 60 iterations to reach convergence. To sum up, the Adam optimizer outperforms SGD in convergence speed and stability under varying operating conditions, making it more suitable for the proposed UDAM-Net model.

3.2. Fault test bench test verification

Experiments were conducted with data collected from a laboratory-built test rig to verify the feasibility and generalization capability of the proposed model in practical applications. The bearing fault experimental setup is illustrated in Fig. 4. The fault categories and corresponding parameters are detailed in Tables 6 and 7. The test ring mainly consists of accelerometers, a data acquisition card, a data acquisition system, and a speed controller. Two experiments were conducted in this study. During the experiments, accelerometers 1, 2, and 3 correspond to channels ch01, ch02, and ch03 in the acquired CSV files, respectively. The data from channel ch02 were selected for analysis. The data acquisition unit employs a 24-bit ADC with eight vibration and voltage channels, supporting a maximum sampling frequency of 77.16 kHz per channel, along with an additional speed channel. It is equipped with a front-end Butterworth low-pass filter, bringing about a measurement error below 0.4 %. The data acquisition software provides three gain levels and supports data acquisition, visualization, storage, and analysis. The acquisition parameters are set with a sampling frequency of 77.16 kHz and a sampling duration of approximately 20 s, as well as a load range of 0-100 kg and a rotational speed range of 200-1500 r/min. The tested fault types comprise outer race faults and rolling element faults. After data segmentation, approximately 1500 samples are obtained, each with a length of 1024 points, of which 80 % are adopted for training and 20 % for testing. Additionally, six variable operating condition fault diagnosis tasks are constructed based on the four bearing fault categories in Table 6 and the three operating conditions in Table 7. The experimental settings are consistent with those of the Case Western Reserve University dataset. Comparative experiments, ablation studies, and optimizer comparison experiments are conducted. The experimental results are evaluated with accuracy, precision, recall, and F1-score. The photograph of the experimental platform in Fig. 4 was taken by the laboratory team at Room 305, School of Mechanical and Equipment Engineering, Hebei University of Engineering, on August 23, 2024.

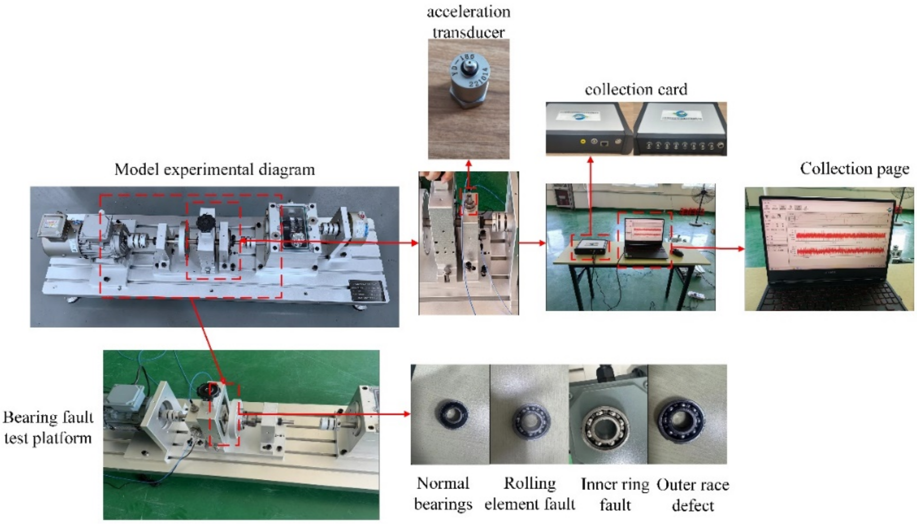


Fig. 4. The main components of the test bench

Table 6. Classification of bearing state

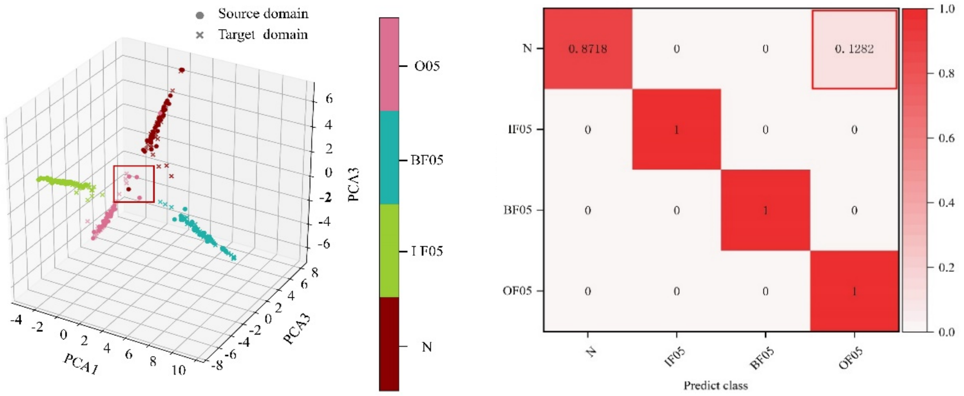
Fault category	Numbering	Fault state	Degree of injury /mm
1	N00	normal state	0
2	IF05	Rolling element fault	0.5
3	BF05	Inner ring fault	0.5
4	OF05	outer race defect	0.5

Table 7. Test condition parameters

Condition / domain	revolution speed (r/min)	Load (HP)
0 hp	800	25%
1 hp	900	50%
2 hp	1000	100%

3.2.1. Comparative test of the fault test bench

Comparative models consistent with those used in the Case Western Reserve University experiments are adopted in this section to verify the effectiveness of the proposed method. The evaluation is performed by calculating the average accuracy, precision, recall, and F1-score across six different transfer tasks. With the 2 hp to 0 hp transfer task as an example, the feature clustering results and confusion matrix of the proposed model are presented in Fig. 5(a) and Fig. 5(b), respectively. The averaged evaluation metrics are summarized in Table 8. The classification accuracies of the seven models across the six transfer tasks are detailed in Table 9.



a) Feature clustering results b) 1DCNN-UDAM-Net confusion matrix

Fig. 5. Feature clustering results and 1DCNN-UDAM-Net confusion matrix

Table 8. Test bench comparison test model diagnosis results

	Accuracy	Precision	Recall	F1-score
1DCNN-Transformer	80.33 %	77.67 %	82.00 %	79.84 %
CORAL	80.00 %	81.67 %	80.00 %	80.84 %
KL	71.00 %	72.17 %	71.00 %	71.59 %
Proxy a distance	86.83 %	86.70 %	86.80 %	86.75 %
CNN-BiLSTM-Transformer	90.83 %	89.33 %	86.67 %	88.01 %
CNN-MDD	86.17 %	83.00 %	83.00 %	83.00 %
UDAM-Net	96.16 %	96.00 %	95.50 %	95.75 %

Table 9. Accuracy of different model migration tasks

	0 hp-1 hp	0 hp-2 hp	1 hp-0 hp	1 hp-2 hp	2 hp-0 hp	2 hp-1 hp
1DCNN-Transformer	96.00 %	75.00 %	84.00 %	81.00 %	72.00 %	74.00 %
CORAL	93.00 %	71.00 %	91.00 %	72.00 %	62.00 %	91.00 %
KL	72.00 %	76.00 %	82.00 %	78.00 %	49.00 %	69.00 %
Proxy a Distance	97.00 %	75.00 %	92.00 %	89.00 %	83.00 %	85.00 %
CNN-BiLSTM-Transformer	93.00 %	92.00 %	86.00 %	96.00 %	86.00 %	92.00 %
CNN-MDD	86.00 %	94.00 %	77.00 %	78.00 %	93.00 %	89.00 %
UDAM-Net	98.00 %	92.00 %	99.00 %	97.00 %	95.00 %	96.00 %

As observed from Fig. 5(a), for the 1 hp to 2 hp transfer task, the feature distribution discrepancy between the source and target domains is not fully overlapped only for the outer race fault and normal conditions. Fig. 5(b) suggests that 12.82 % of normal bearing samples are misclassified as outer race faults, while all other fault types are correctly identified. As reported in Table 8, UDAM-Net outperforms the other metric-based methods and baseline models across all four evaluation metrics. Table 9 reflects that, particularly for the 0 hp to 2 hp task, the proposed model achieves accuracy comparable to that of ResNet-DANN and 1DCNN-DANN, with a slight decrease. Nonetheless, the proposed model consistently outperforms the competing methods under the other operating conditions, achieving the best overall performance.

3.2.2. Failure test bench ablation test

With the purpose of evaluating the overall effectiveness of each metric module under practical operating conditions, ablation experiments were conducted on a laboratory-built experimental platform to further validate the generalization and feasibility of the proposed model. The ablation models adopted in this experiment are consistent with those used in the Case Western Reserve University experiments. The averaged results are summarized in Table 10. The experimental results under six operating conditions are detailed in Table 11.

Table 10. Comparison of diagnostic results of ablation test model indexes on the test bench

	Accuracy	Precision	Recall	F1-score
1DCNN- Transformer	80.33 %	77.67 %	82.00 %	79.84 %
JMMD	88.50 %	89.67 %	89.00 %	89.34 %
MKMMD	89.83 %	91.83 %	91.00 %	91.42 %
JMMD-MK-MMD	92.33 %	90.50 %	90.67 %	90.59 %
UDAM-Net	96.16 %	96.00 %	95.50 %	95.75 %
UDAM-Net(none)	76.50 %	75.67 %	75.50 %	75.59 %

Table 11. Diagnostic results of the migration task of the ablation test model on the test bench

	0 hp-1 hp	0 hp-2 hp	1 hp-0 hp	1 hp-2 hp	2 hp-0 hp	2 hp-1 hp
1DCNN- Transformer	96.00 %	75.00 %	84.00 %	81.00 %	72.00 %	74.00 %
JMMD	94.00 %	83.00 %	95.00 %	84.00 %	83.00 %	92.00 %
MK-MMD	92.00 %	79.00 %	94.00 %	92.00 %	92.00 %	90.00 %
JMMD-MK-MMD	95.00 %	83.00 %	96.00 %	97.00 %	87.00 %	96.00 %
UDAM-Net	98.00 %	92.00 %	99.00 %	97.00 %	95.00 %	96.00 %
UDAM-Net(none)	94.00 %	71.00 %	79.00 %	68.00 %	77.00 %	70.00 %

As revealed in Table 10, the model accuracy reaches 96.16 % after the introduction of JMMD, MK-MMD, and the domain discriminator. In contrast, removing the Transformer layer leads to a significant accuracy decrease of 19.66 %. In other words, the Transformer-based self-attention mechanism substantially enhances feature extraction capability on top of the 1D-CNN backbone. Table 11 reflects that, under the 1 hp to 0 hp and 2 hp to 0 hp operating conditions, introducing the metric modules on top of the 1DCNN-Transformer increases the accuracy by 15 % and 23 %, respectively. Conversely, removing the Transformer layer yields accuracy reductions of 20 % and 18 %, respectively. These results further verify the critical roles of each module in improving overall model performance.

3.2.3. Optimizer comparison test

Experiments are conducted on data collected from the bearing fault test rig to verify the advantage of the Adam optimizer. Specifically, the performance of Adam and SGD was evaluated across six transfer tasks, with the 0 hp to 1 hp and 0 hp to 2 hp operating conditions selected as representative examples. The accuracy and loss curves of the two optimizers under different tasks are illustrated in Fig. 6.

When the Adam optimizer is employed, the model's accuracy and loss values tend to stabilize after approximately 40 iterations, whereas the SGD optimizer requires around 60 iterations to reach convergence, as suggested in Fig. 6. These results specify that the Adam optimizer exhibits higher convergence efficiency and stability in the 1DCNN-UDAM-Net model, further confirming its advantage in terms of convergence speed and training stability.

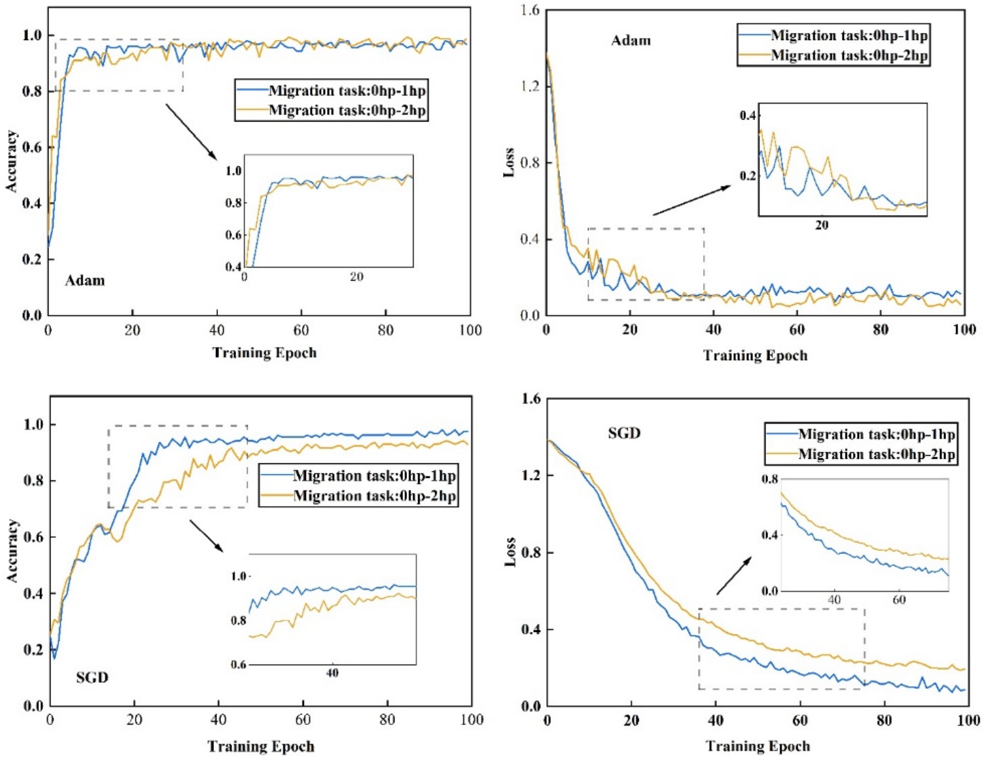


Fig. 6. Comparison of accuracy and loss rate of self-built test bench optimizer

4. Conclusions

In this study, a deep transfer learning approach that integrates a multi-head attention mechanism with explicit and implicit distance metric learning was established to overcome the challenges of directly extracting discriminative features from raw signals under complex operating conditions, as well as the reduced diagnostic accuracy stemming from cross-domain data distribution imbalance under variable operating conditions. The contribution of each module to the overall framework is systematically validated by comparing the performance of different models. Experimental results demonstrate that, under identical parameter settings, UDAM-Net achieves superior fault diagnosis accuracy and maintains consistent effectiveness across different datasets.

1) The feature extraction capability is significantly enhanced by building upon a 1D-CNN architecture and incorporating a Transformer layer. Additionally, both marginal and conditional distribution alignment capabilities are improved by introducing JMMD and MK-MMD metrics together with a domain discriminator, contributing to the formation of a cross-domain feature alignment framework that combines class discriminability with multi-scale global feature alignment. An uncertainty-based adaptive weighting mechanism is employed to dynamically balance the contributions of different loss terms, thereby reducing the complexity of hyperparameter tuning and significantly enhancing model robustness and cross-domain generalization ability. Finally, the Adam optimizer is adopted to accelerate convergence and improve training stability.

2) The performance of UDAM-Net is validated on both the public CWRU dataset and a self-built experimental platform dataset. Experimental results specify that UDAM-Net achieves average accuracy improvements of 2.67 %, 22.84 %, and 6.57 % over CORAL, KL divergence, and Proxy-A distance, respectively, on the CWRU dataset. Compared with the baseline models

CNN-BiLSTM-Transformer and CNN-MDD, further improvements of 3.16 % and 6.41 % are obtained, respectively. Under the operating conditions of the self-built experimental platform, UDAM-Net improves the average accuracy by 13 % and 9 % for the 1 hp to 0 hp and 2 hp to 0 hp transfer tasks, respectively, compared with CNN-BiLSTM-Transformer. In comparison with CNN-MDD, UDAM-Net achieves accuracy improvements of 22 % and 19 % for the 1 hp to 0 hp and 1 hp to 2 hp tasks, respectively. These results further confirm the high diagnostic accuracy of UDAM-Net under practical operating conditions and its superiority across multiple datasets and application scenarios.

3) In the ablation experiments conducted on the CWRU dataset, introducing JMMD and MK-MMD on top of the 1DCNN-Transformer yields an improvement of the accuracy of 1.42 % and 1.75 %, respectively, while their combination brings about an accuracy improvement of 3.5 %. Thus, enhanced feature distribution alignment strengthens cross-domain adaptability. After the introduction of the domain discriminator, the model accuracy is further increased to 96.67 %. This result suggests that the domain discriminator effectively optimizes feature alignment between the source and target domains, reduces inter-domain discrepancies, and enhances adaptability and generalization under variable operating conditions and cross-domain tasks. In contrast, removing the Transformer layer leads to a 7 % decrease in accuracy, further validating the critical role of the Transformer-based self-attention mechanism in feature extraction and long-range dependency modeling. In the ablation experiments conducted on the laboratory-built experimental platform, particularly under the 1 hp to 2 hp and 2 hp to 0 hp operating conditions, introducing the three metric modules into the 1DCNN-Transformer increases the accuracy by 15 % and 23 %, respectively; removing the Transformer layer results in accuracy reductions of 29 % and 18 %, respectively. These results validate that the Transformer self-attention mechanism, the combined JMMD-MK-MMD metrics, and the domain discriminator make critical contributions to performance improvement, enabling UDAM-Net to achieve higher diagnostic accuracy and stronger adaptability under complex operating conditions and cross-domain tasks.

4) The impact of different optimizers on model convergence speed and training stability is evaluated. The results from the experiments conducted on both the CWRU dataset and the self-built dataset unveil that the Adam optimizer achieves rapid convergence and stabilizes after approximately 20 and 40 iterations, respectively, whereas the SGD optimizer requires around 60 iterations to reach a stable state. These findings highlight the clear advantage of the Adam optimizer in accelerating convergence and improving model stability.

5) In future work, the performance of the proposed model will be further investigated under more complex operating conditions, particularly in multi-source domain scenarios involving multiple operating states. Furthermore, its generalization ability and high-accuracy diagnostic performance will be evaluated in situations with limited or missing samples.

Acknowledgements

This work was supported by the Science Research Project of the Hebei Education Department (No. CXY2023002).

Data availability

The datasets generated during and/or analyzed during the current study are available from the corresponding author on reasonable request.

Author contributions

Yiying Wang: conceptualization, investigation, methodology, writing-original draft, writing-review and editing, validation, funding acquisition. Fulu Sui: conceptualization, supervision, project administration, formal analysis, data curation. Xiaoling Li: supervision, project

administration. Xiaoxin Zhang: conceptualization, funding acquisition, supervision. Mingxian Liu: conceptualization, project administration. Chen Liu: supervision, project administration. Jie Wu: conceptualization, supervision.

Conflict of interest

The authors declare that they have no conflict of interest.

References

[1] G. Yang, W. Xu, Q. Deng, Y. Wei, and F. Li, "A review of rolling bearing compound fault diagnosis based on vibration signals," (in Chinese), *Journal of Xihua University*, Vol. 43, pp. 48–69, 2024, https://doi.org/10.12198/j.issn.1673-159x.5096

[2] J. Wang, Z. Xu, W. Liu, Y. Wang, and L. Liu, "Research on health monitoring and fault diagnosis mechanism of rolling bearings," (in Chinese), *Journal of Frontier of Computer Science and Technology*, Vol. 18, No. 4, pp. 878–898, 2024.

[3] M. Demetgul, K. Yildiz, S. Taskin, I. N. Tansel, and O. Yazicioglu, "Fault diagnosis on material handling system using feature selection and data mining techniques," *Measurement*, Vol. 55, pp. 15–24, Sep. 2014, https://doi.org/10.1016/j.measurement.2014.04.037

[4] M. Dou, Y. Zhang, F. Sun, and H. Pei, "Combined model: CEEMDAN and AVMD quadratic decomposition noise reduction and BiLSTM-BP classification model rolling bearing fault diagnosis," *Engineering Research Express*, Vol. 7, No. 3, p. 035560, Sep. 2025, https://doi.org/10.1088/2631-8695/adfe38

[5] X. He, "Research on bearing fault diagnosis based on the fusion of CNN and LSTM algorithms," in *Journal of Physics: Conference Series*, Vol. 3057, No. 1, p. 012054, Jul. 2025, https://doi.org/10.1088/1742-6596/3057/1/012054

[6] G. Xu, J. Cao, W. Liu, D. Song, J. Zhong, and L. Meng, "Anovel fault diagnosis method for rolling bearing based on SGMD and improved CNN-LSTM," *Engineering Research Express*, Vol. 7, No. 3, p. 035567, Sep. 2025, https://doi.org/10.1088/2631-8695/adf93b

[7] J. Zou, W. Qiu, Z. Liu, J. Su, T. Chen, and Q. Liu, "Fault diagnosis method for rotating machinery based on FFT-CNN-transformer-crossattention," *International Journal of High Speed Electronics and Systems*, Aug. 2025, https://doi.org/10.1142/s0129156425408927

[8] X. Shao and C.-S. Kim, "Multi-branch global Transformer-assisted network for fault diagnosis," *Applied Soft Computing*, Vol. 182, p. 113572, Oct. 2025, https://doi.org/10.1016/j.asoc.2025.113572

[9] J. Hu, "Intelligent fault diagnosis of rolling bearings based on VMD-CNN-transformer," *World Journal of Engineering Research*, Vol. 3, No. 2, Jan. 2025, https://doi.org/10.61784/wjer3029

[10] I. Goodfellow et al., "Generative adversarial networks," *Communications of the ACM*, Vol. 63, No. 11, pp. 139–144, Oct. 2020, https://doi.org/10.1145/3422622

[11] X. Li, P. Yuan, K. Su, D. Li, Z. Xie, and X. Kong, "Innovative integration of multi-scale residual networks and MK-MMD for enhanced feature representation in fault diagnosis," *Measurement Science and Technology*, Vol. 35, No. 8, p. 086108, Aug. 2024, https://doi.org/10.1088/1361-6501/ad4380

[12] L. Wan, Y. Li, K. Chen, K. Gong, and C. Li, "A novel deep convolution multi-adversarial domain adaptation model for rolling bearing fault diagnosis," *Measurement*, Vol. 191, p. 110752, Mar. 2022, https://doi.org/10.1016/j.measurement.2022.110752

[13] H. Wu, J. Li, Q. Zhang, J. Tao, and Z. Meng, "Intelligent fault diagnosis of rolling bearings under varying operating conditions based on domain-adversarial neural network and attention mechanism," *ISA Transactions*, Vol. 130, pp. 477–489, Nov. 2022, https://doi.org/10.1016/j.isatra.2022.04.026

[14] Y. Tian and X. Liu, "A deep adaptive learning method for rolling bearing fault diagnosis using immunity," *Tsinghua Science and Technology*, Vol. 24, No. 6, pp. 750–762, Dec. 2019, https://doi.org/10.26599/tst.2018.9010144

[15] G. Xiao, J. Yao, L. Zhong, Z. Xiao, and J. Lu, "MB-ViT: MBConv vision transformer with time-frequency feature fusion for bearing fault diagnosis," *Neural Computing and Applications*, Vol. 37, No. 27, pp. 22801–22825, Aug. 2025, https://doi.org/10.1007/s00521-025-11509-7

[16] Y. Jin, L. Hou, and Y. Chen, "A Time Series Transformer based method for the rotating machinery fault diagnosis," *Neurocomputing*, Vol. 494, pp. 379–395, Jul. 2022, https://doi.org/10.1016/j.neucom.2022.04.111

[17] M. Long, H. Zhu, J. Wang, and M. I. Jordan, "Deep transfer learning with joint adaptation networks," in *Proceedings of the 34th International Conference on Machine Learning (ICML 2017), Proceedings of Machine Learning Research*, Vol. 70, pp. 2208–2217, 2017.

[18] Y. Gao and X. Li, "Fault diagnosis of centrifugal fan bearings based on I-CNN and JMMD in the context of sample imbalance," *Eksploatacja i Niezawodność – Maintenance and Reliability*, Vol. 26, No. 4, Jul. 2024, https://doi.org/10.17531/ein/191459

[19] G. Y. Ganin et al., "Domain adversarial training of neural networks," *Journal of Machine Learning Research*, Vol. 17, No. 59, 2016.

[20] J. Li, Z. Ye, J. Gao, Z. Meng, K. Tong, and S. Yu, "Fault transfer diagnosis of rolling bearings across different devices via multi-domain information fusion and multi-kernel maximum mean discrepancy," *Applied Soft Computing*, Vol. 159, p. 111620, Jul. 2024, https://doi.org/10.1016/j.asoc.2024.111620

[21] X. Li et al., "Research on unsupervised domain adaptive bearing fault diagnosis method based on migration learning using MSACNN-IJMMD-DANN," *Machines*, Vol. 13, No. 7, p. 618, Jul. 2025, https://doi.org/10.3390/machines13070618

[22] Z. Wang et al., "Weighted joint maximum mean discrepancy enabled multi-source-multi-target unsupervised domain adaptation fault diagnosis," *arXiv:2310.14790*, Jan. 2023, https://doi.org/10.48550/arxiv.2310.14790

[23] W. A. Smith and R. B. Randall, "Rolling element bearing diagnostics using the Case Western Reserve University data: A benchmark study," *Mechanical Systems and Signal Processing*, Vol. 64-65, pp. 100–131, Dec. 2015, https://doi.org/10.1016/j.ymssp.2015.04.021

[24] X. Yuan and W. Wu, "A bearing fault diagnosis method based on fusion of CNN-BiLSTM-transformer and cross-attention," *Academic Journal of Science and Technology*, Vol. 14, No. 2, pp. 191–199, Mar. 2025, https://doi.org/10.54097/594q8441

[25] C. Wang, S. Wu, and X. Shao, "Unsupervised domain adaptive bearing fault diagnosis based on maximum domain discrepancy," *EURASIP Journal on Advances in Signal Processing*, Vol. 2024, No. 1, pp. 1–12, Jan. 2024, https://doi.org/10.1186/s13634-023-01107-x

Yiying Wang is an Associate Professor in School of Mechanical and Equipment Engineering, Hebei University of Engineering. In 2017, he received a doctorate in control theory and engineering from China University of Mining and Technology (Beijing). His research direction is the power electronic transformer, coal mine filling mining.

Fulu Sui is a postgraduate student in the School of Mechanical and Equipment Engineering of Hebei University of Engineering. In 2022, he received a Bachelor's degree in mechanical design, manufacturing and automation from the School of Light Industry, North China University of Technology. His research direction is bearing fault diagnosis.

Xiaoling Li is an Associate Professor in the computer department of Handan Vocational and Technical College. In 2017, he was awarded a Master's degree in the computer department of Hebei University of Engineering. The research direction is computer application technology.

Xiaoxin Zhang is a Lecturer in the computer department of Handan Vocational and Technical College. In 2021, he received a Master's degree in electronic and communication engineering from North China Electric Power University. Her research direction is electronic and communication engineering.

Mingxian Liu is a postgraduate student in the School of Mechanical and Equipment Engineering of Hebei University of Engineering. In 2023, he received a Bachelor's degree in mechanical design, manufacturing and automation from Hebei University of Engineering. His research direction is power electronic transformer.

Chen Liu is a postgraduate student in the School of Mechanical and Equipment Engineering of Hebei University of Engineering. In 2023, he received a Bachelor's degree in vehicle engineering from Hebei University of Engineering. His research direction is power electronic transformer.

Jie Wu is an Associate Professor of Anyang Institute of Technology, School of Mechanical and Aeronautical Manufacturing Engineering, and received a doctorate in mechanical engineering from Xi'an Jiaotong University in 2019.