Expert Systems With Applications 296 (2026) 129106

Contents lists available at ScienceDirect

Expert Systems With Applications

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

An improved lightweight residual network model deployed on the edge device for the unsupervised cross-domain fault diagnosis

Changbo He [a,1], Qi Meng [a,2], Xuefang Xu [b,3], Peng Chen [c,4,*], Pengfei Liang [d,5], Jiahui Cao [e,6]

[a] School of Electrical Engineering and Automation, Anhui University, Hefei 230601, China
[b] School of Electrical Engineering, Yanshan University, Qinhuangdao 066004, China
[c] College of Engineering, Shantou University, Shantou 515063, China
[d] School of Mechanical Engineering, Yanshan University, Qinhuangdao 066004, China
[e] School of Mechanical Engineering, Xi'an Jiaotong University, Xi'an 710049, China

ARTICLE INFO

Keywords:
Cross-domain Fault diagnosis
Unsupervised transfer learning
Depthwise separable convolution
Group normalization
Lightweight residual network
Edge deployment

ABSTRACT

In cross-domain fault diagnosis, samples usually lack labels because of different working conditions. Therefore, unsupervised deep transfer learning is more suitable than deep learning to tackle it. Moreover, deploying these methods on edge devices can reduce diagnostic latency. Thus, diagnostic methods of unsupervised deep transfer learning deployed on edge devices deserve attention from researchers. Due to high computational costs, reducing unnecessary model complexity while maintaining competitive performance for edge computing is a critical issue. To address the issue, this paper proposes an unsupervised transfer learning model based on an improved lightweight residual network, which achieves higher accuracy than complex models while significantly reducing parameters size, making it suitable for edge deployment. First, an improved lightweight residual network is proposed, which incorporates two novel types of residual blocks that utilize depthwise separable convolution and group normalization. Then, a new feature extraction network is introduced by combining the improved lightweight residual network with the Spatial and Channel Reconstruction Convolution (SCConv) module. Based on the proposed feature extraction network, an unsupervised cross-domain fault diagnosis model is constructed, incorporating Joint Maximum Mean Discrepancy (JMMD) and adversarial network loss for domain adaptation. Furthermore, two bearing datasets are utilized to validate the effectiveness of the proposed method and the improved model is deployed on an edge device to demonstrate its feasibility in practical applications. Experimental results show that the proposed method achieves higher accuracy than traditional complex models while maintaining fewer parameters and high computational efficiency, making it a practical solution for edge-based fault diagnosis.

1. Introduction

Traditional signal processing methods preprocess the acquired fault signals to extract characteristic frequencies, including Empirical Mode Decomposition (EMD) (Yu et al., 2005), wavelet transform (Burrus et al., 1998), Short-Time Fourier Transform (STFT) (Cocconcelli et al., 2012), Wigner-Ville Distribution (WVD) (Boashash and Black, 1987), singular value decomposition (Cong et al., 2013), and blind source separation (Nguyen et al., 2012), among others. For example, Bao et al. (2022) proposed a fault diagnosis model that combines EMD and Convolutional

* Corresponding author.
E-mail addresses: changbh@ahu.edu.cn (C. He), Z23301059@stu.ahu.edu.cn (Q. Meng), xuefangxu@ysu.edu.cn (X. Xu), pengchen@alu.uestc.edu.cn, dr.pengchen@foxmail.com (P. Chen), liangpf@ysu.edu.cn (P. Liang), caojiahui@stu.xjtu.edu.cn (J. Cao).
[1] 0000-0003-4180-5334.
[2] 0009-0004-8800-6833.
[3] 0000-0002-3861-8733.
[4] 0000-0002-3265-3079.
[5] 0000-0003-1938-895X.
[6] 0000-0002-1705-7965.

https://doi.org/10.1016/j.eswa.2025.129106
Received 25 February 2025; Received in revised form 24 June 2025; Accepted 19 July 2025
Available online 20 July 2025
0957-4174/© 2025 Elsevier Ltd. All rights are reserved, including those for text and data mining, AI training, and similar technologies.

Sparse Filtering (CSF) (Wohlberg, 2015). Traditional fault diagnosis methods require manual feature extraction, which introduces subjectivity and uncertainty. The diagnostic performance is heavily influenced by prior expert knowledge, making it difficult to establish corresponding diagnostic models for varying working conditions. Meanwhile, due to the high latitude and nonlinearity of modern data, traditional methods are difficult to extract complex features, resulting in insufficient reliability of diagnosis results.

In recent years, deep learning-based methods possess the capability of automatic feature learning, which can extract valuable features directly from raw data, thereby handling complex nonlinear relationships and achieving intelligent fault diagnosis. Compared to traditional methods, deep learning effectively addresses the shortcomings of the former and greatly improves the accuracy of fault diagnosis tasks. Various deep networks are applied to intelligent fault diagnosis, including Multi-Layer Perceptron (MLP) (ALTobi et al., 2019), Autoencoder (AE) (Ma et al., 2018), Deep Belief Networks (DBN) (Wang et al., 2020), Long Short-Term Memory (LSTM) (Han et al., 2020), and Convolutional Neural Network (CNN) (Wen et al., 2017). For example, Abdeljaber et al. (2017) utilized a one-dimensional adaptive CNN that integrates feature extraction and classification modules within a single learning framework tailored for fault diagnosis. Ma et al. (2019) introduced a deep residual learning approach for diagnosing non-stationary operating states of planetary gearboxes using demodulation time–frequency features.

However, in practical engineering applications, samples often lack labels due to variations in equipment operating conditions. Deep learning-based methods struggle to handle the classification of such samples, making it difficult to accomplish cross-domain fault diagnosis tasks. These samples often share certain common characteristics within a specific feature space. Leveraging the powerful feature extraction capabilities of deep learning, unsupervised deep transfer learning (UDTL) (Tan et al., 2018) is introduced for fault diagnosis (Li et al., 2020) under the condition of missing labels and has demonstrated significant progress. UDTL involves extracting features through a feature extractor and domain adaption to map the features of the source domain and target domain into a higher-dimensional space for discrimination, so that it can transfer knowledge to address the issue of lacking labels in target domain.

In light of this, researchers propose various transfer models for cross-domain fault diagnosis. Shao et al. (2018) introduced an improved framework based on VGG16 and wavelet transform to achieve high-precision fault diagnosis. Gao et al. (2021) proposed a novel method based on data self-production and a new network named SP-CNN, which enhances the diagnostic accuracy of typical chiller faults through data augmentation techniques. Zhao et al. (2023) proposed the Conditional Weighted Transfer Wasserstein Autoencoder, to address the challenges of cross-domain fault diagnosis. Yang et al. (2023) realized unbalanced fault diagnosis in wind turbine generators based on GAN and wavelet packet transform. Wang et al. (2023) combined graph labels and manifold connections to enhance the generalization capability of sparse data for fault diagnosis. Meng et al. (2022) showed a CNN-based method using grayscale images to achieve two-dimensional fault diagnosis. These cross-domain fault diagnosis models all utilize complex and large-scale backbone networks, focusing solely on accuracy without paying attention to operational efficiency.

Related studies pay more attention to constructing an overly complex network, which may be unnecessary for just minor precision improvement. That's because although overly complex models can achieve insignificant accuracy improvement, it incurs computational resource costs that don't proportionally match the increase in accuracy. However, the related studies overlooked the critical need to optimize the balance between complexity and accuracy. In addition, fault diagnosis algorithms are closely related to industrial practice, so it is meaningful to deploy the model to edge devices and consider practical applications. Unfortunately, related studies have rarely considered the practical application and failed to incorporate lightweight processing considerations into block designs for the deployment of edge devices.

Aiming at the above issues, this paper proposes an unsupervised cross-domain fault diagnosis model that can be deployed on edge device. The main contributions of this paper are as follows:

(1) Proposing two improved lightweight residual blocks to improve the residual network. These blocks utilize depthwise separable convolution and group normalization for reducing computing resource consumption.
(2) Designing an unsupervised transfer learning model for fault diagnosis. The feature extraction network consists of improved lightweight residual blocks and the SCConv module, and the domain adaptation module is composed of JMMD and adversarial network loss.
(3) Deploying the source domain-trained model on edge computing devices to simulate and validate fault diagnosis scenarios in the target domain for actual engineering applications.

The structure of the paper is organized as followed: Section 2 introduces the related research. Section 3 shows the improvements of this paper's transfer learning model. Section 4 presents experimental results of the proposed model. Section 5 shows the limitation of the research and has a prospect for the future research. Finally, Section 6 concludes this paper.

2. Related research

2.1. Depthwise separable convolution

Depthwise separable convolution (DSConv) (Chollet, 2017) consists of the depthwise convolution (DW) and the pointwise convolution (PW) (Sifre and Mallat, 2014). It operates by grouping features dimensionally, applying depthwise convolution independently to each channel, and then using pointwise convolution to fuse all channels, thereby obtaining features and reducing computational resource consumption.

Assume that the input data for depthwise separable convolution is $[H_{in}, W_{in}]$ with C_{in} channels, the output data is $[H_{out}, W_{out}]$ with C_{out} channels, the convolution kernel size is k, the padding is P and the stride is S. The ratios between depthwise separable convolution and standard convolution are given by the Eqs. (1) ~ (4):

$$H_{out} = \frac{H_{in} + 2P - k}{S} \tag{1}$$

$$W_{out} = \frac{W_{in} + 2P - k}{S} \tag{2}$$

$$r_1 = \frac{k \times k \times C_{in} + C_{in} \times C_{out}}{k \times k \times C_{in} \times C_{out}} = \frac{1}{C_{in}} + \frac{1}{k^2} \tag{3}$$

$$r_2 = \frac{H_{out} \times W_{out} \times k \times k \times C_{in} + H_{out} \times W_{out} \times C_{in} \times C_{out}}{H_{out} \times W_{out} \times k \times k \times C_{in} \times C_{out}} = \frac{1}{C_{in}} + \frac{1}{k^2} \tag{4}$$

where, r_1 presents the ratio of paraments and r_2 presents the ratio of computational load.

2.2. Group normalization

Group Normalization (GN) (Wu and He, 2018) divides feature into multiple groups and normalizes each group's features. Compared to Batch Normalization (BN) (Bjorck et al., 2018), GN does not require calculating statistical information for each channel, reducing computational consumption under small batches, as shown in Fig. 1.

In the case of one-dimensional data, it is assumed that the input feature X has a shape of $N \times C \times H$, where N represents the batch size, C represents the number of channels, and H represents the feature length.

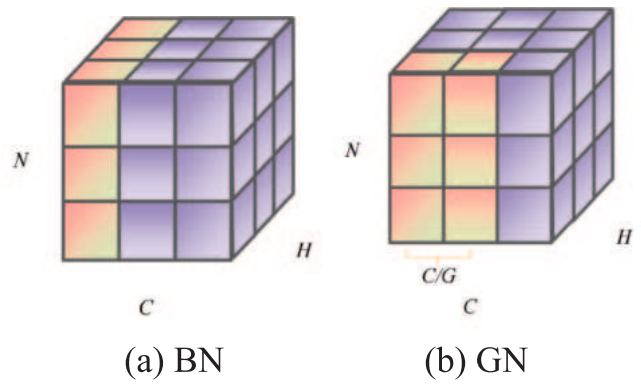


Fig. 1. The normalization: (a) BN and (b) GN.

GN divides C feature channels into G groups, so that X can be divided into G subsets $\{X_1, X_2, \ldots X_G\}$, representing the features of the i group. Finally, the statistics (mean and standard deviation) are obtained by normalization.

2.3. Spatial and channel reconstruction convolution

Spatial and Channel reconstruction Convolution (SCConv) (Li et al., 2023) includes two special modules: the spatial reconstruction unit (SRU) and the channel reconstruction unit (CRU). The SCConv module not only effectively suppresses feature redundancy but also improves computational efficiency.

The SRU primarily operates in the spatial dimension by separating redundant features based on weights to suppress spatial feature redundancy. Its main structure is shown in Fig. 2. Specifically, the SRU is described by Eqs. (5) ~ (9) as follows:

$$X_1^w = W_1 \otimes X \tag{5}$$

$$X_2^w = W_2 \otimes X \tag{6}$$

$$X_{11}^w \oplus X_{22}^w = X^{w1} \tag{7}$$

$$X_{21}^w \oplus X_{12}^w = X^{w2} \tag{8}$$

$$X^{w1} \cup X^{w2} = X^w \tag{9}$$

where, W_1, W_2 represent the weights for the input X. X_1^w represents the space features with high information content. X_2^w represents the redundant features with less information content. Finally, the output X^w of the SRU is obtained through cross-reconstruction.

N represents the weight of normalization, S is a sigmoid function, T represents a threshold to separate the features and C represents the concatenation.

The CRU reduces feature redundancy along the channel dimension through a Split-Transform-and-Fuse strategy, with its structure illustrated in Fig. 3. The CRU first uses the Split to divide the channel features into two parts, X_{up} and X_{low}. Then it obtains Y_1 and Y_2 through Group-wise Convolution (GWC) and Point-wise Convolution (PWC):

$$Y_1 = M^G X_{up} + M^{P_1} X_{up} \tag{10}$$

$$Y_1 = M^{P_2} X_{low} \cup X_{low} \tag{11}$$

Where, M^G, M^{P_1}, M^{P_2} are the weight matrixes from GWC and PWC.

Finally, combine Y_1 and Y_2:

$$Y = \beta_1 Y_1 + \beta_2 Y_2 \tag{12}$$

$$\beta_1 + \beta_2 = 1 \tag{13}$$

where, β_1 and β_2 are all importance vectors.

3. Proposed method

3.1. Improved lightweight residual block

In residual networks (He et al., 2016), residual blocks are crucial components of the network. This paper designs two types of residual blocks: LRBB-A and LRBB-B, in order to reduce computational resource consumption while maintaining feature extraction capabilities.

LRBB-A (Fig. 4(a)) primarily replaces the convolution and BN in traditional residual blocks with DSConv and GN, respectively.

Assume the input to the residual block is X , and the output Y after passing through LRBB-A is:

$$Y = \boldsymbol{ReLU}(G(X) + X) \tag{14}$$

where, $G(X)$ is the output of the main path branch and x is the input connected to $G(X)$ through skip connection.

The first DSConv in LRBB-B changes the dimension of input by downsampling to extract more essential features. The residual connection part of LRBB-B adjusts the number of channels and dimensions of the input features through convolution and GN, as shown in Fig. 4(b), ensuring that the original input features are preserved as much as possible.

Assume the input to the residual block is X , and the output Y after passing through LRBB-B is:

$$Y = \boldsymbol{ReLU}(\boldsymbol{G}(\boldsymbol{X}) + \boldsymbol{H}(\boldsymbol{X})) \tag{15}$$

$$H(x) = GN(conv(x)) \tag{16}$$

As mentioned above (Eqs. (1) ~ (4)), compared to ordinary convolution, depthwise separable convolution can reduce the number of parameters and computational load to about 1/8. Additionally, GN has the significant advantage of not relying on batch size, especially under small batch training, which greatly reduces computational complexity and improves computational efficiency. Therefore, the residual blocks in this paper, compared to those using ordinary convolution and BN, significantly reduce computational resources.

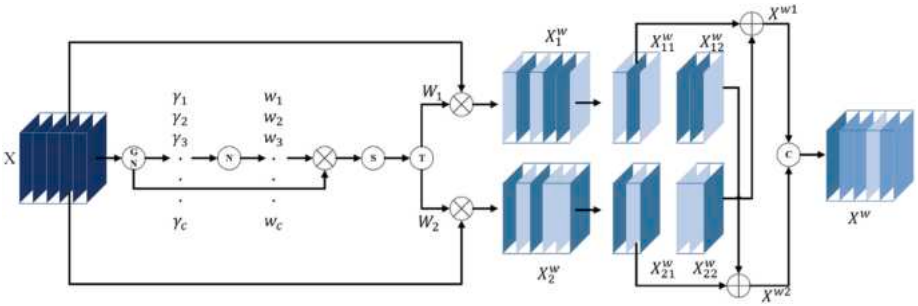


Fig. 2. Spatial reconstruction unit.

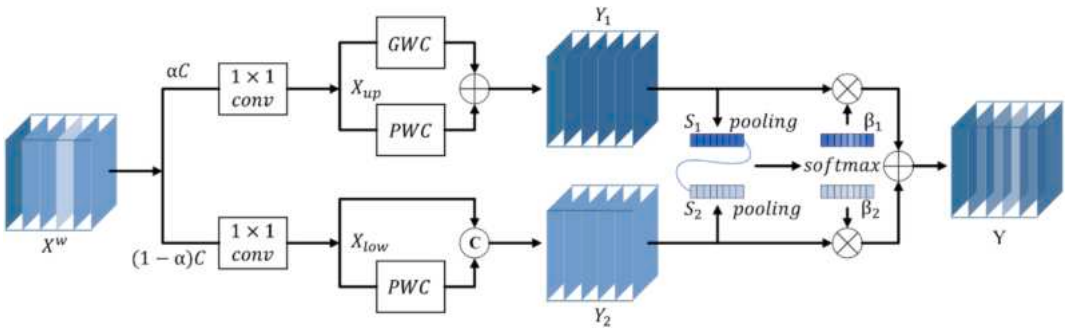


Fig. 3. Channel reconstruction unit.

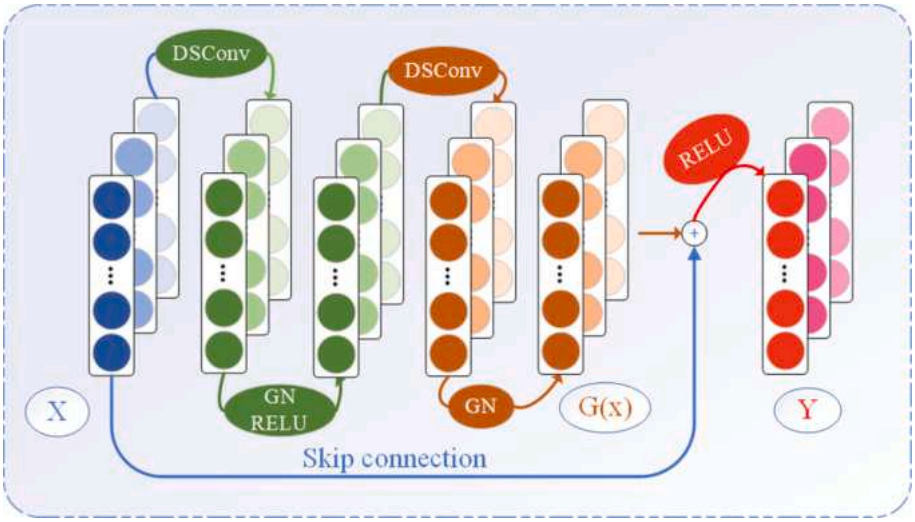


Fig. 4a. LRBB-A.

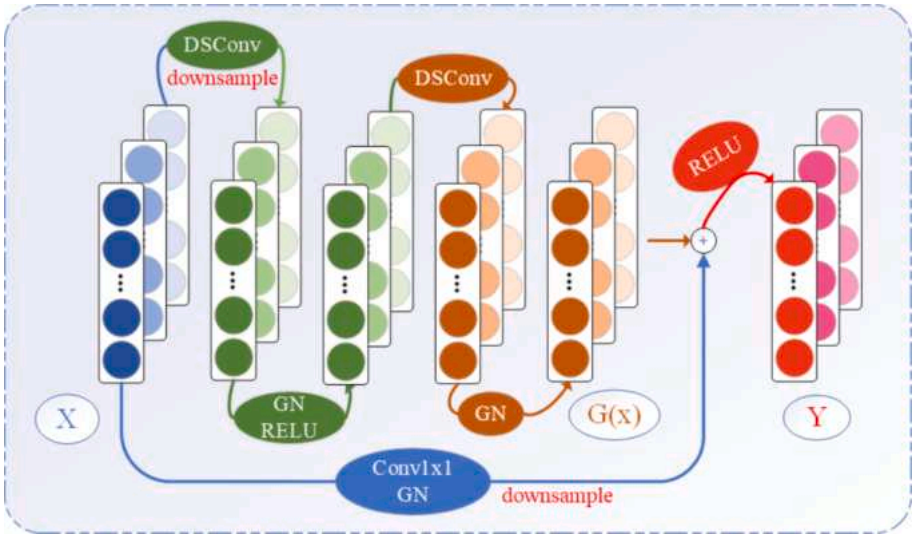


Fig. 4b. LRBB-B.

3.2. Construction of the lightweight residual network model

The lightweight of the model is also worth noting in transfer learning, as it determines the inference efficiency of the model. Therefore, this paper constructs a lightweight residual network model for unsupervised cross-condition rotating machinery bearing fault diagnosis, as shown in Fig. 5.

The lightweight residual network model consists of a lightweight feature extraction network and a domain adaptation module. The feature extraction network is composed of the above-mentioned lightweight blocks and SCConv.

The domain adaptation module in this paper primarily constitutes

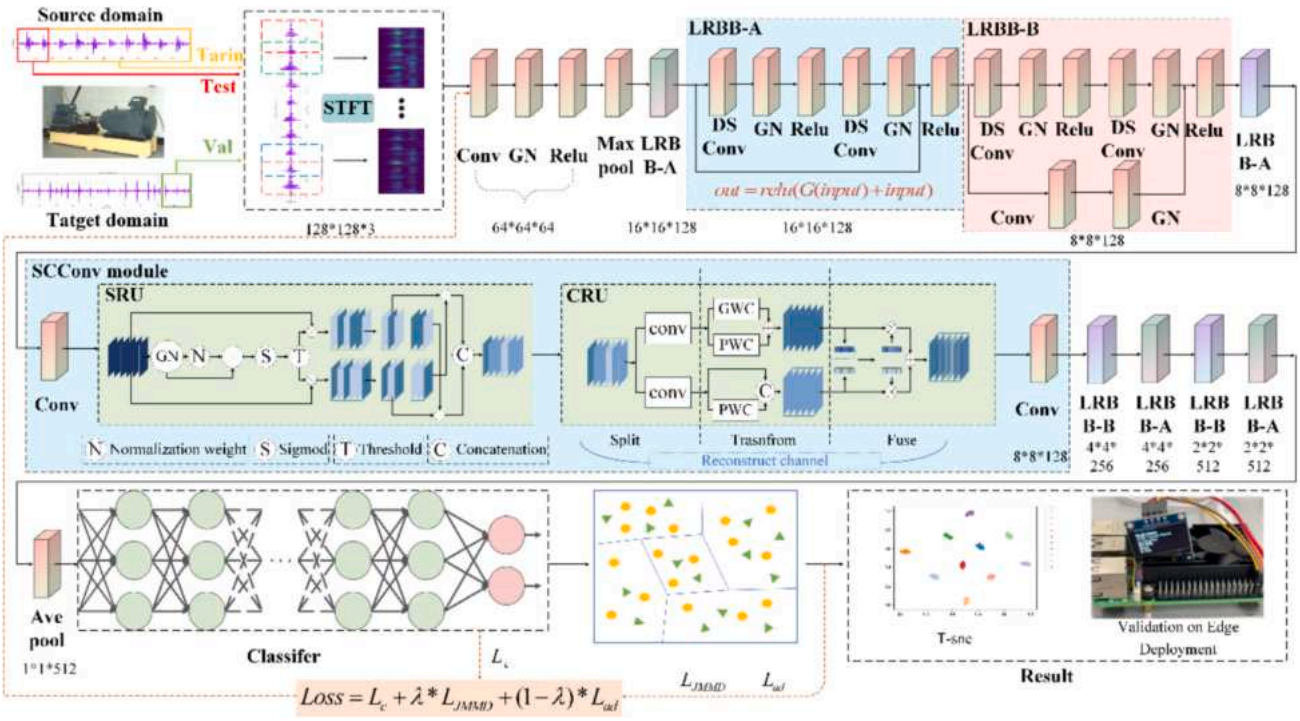


Fig. 5. The improved unsupervised cross-domain fault diagnosis.

the optimization objective, which includes the adversarial networks loss and JMMD (Long et al., 2017), as given by the Eqs. (17) ~ (18).

Assuming that the source domain and target domain are D_s and D_t, respectively.$X_s = \left\{x_1^{s1}, \cdots, x_{n_s}^{s|L|}\right\}andX_t = \left\{x_1^{t1}, \cdots, x_{n_t}^{t|L|}\right\}$are the corresponding sample data. The data feature distributions are$Z_s = \left\{z_1^{s1}, \cdots, z_{n_s}^{s|L|}\right\}$and $Z_t=\left\{z_1^{t1}, \cdots, z_{n_t}^{t|L|}\right\}$, respectively. n_s and n_t are the numbers of samples.

$$L_{JMMD}(P,Q) = \frac{1}{n_s^2}\sum_{i=1}^{n_s}\sum_{j=1}^{n_s}\prod_{l\in L}K^l\left(z_i^{sl}, z_j^{sl}\right) + \frac{1}{n_t^2}\sum_{i=1}^{n_t}\sum_{j=1}^{n_t}\prod_{l\in L}K^l\left(z_i^{tl}, z_j^{tl}\right) - \frac{1}{n_s n_t}\sum_{i=1}^{n_s}\sum_{j=1}^{n_t}\prod_{l\in L}K^l\left(z_i^{sl}, z_j^{tl}\right) \tag{17}$$

The adversarial loss L_{ad} is from the discriminator loss function in the GAN and it is utilized to maximize the distribution of source and target domain data, as shown in the Eq. (18). D represents the discriminator in the GAN.

$$L_{ad} = \mathrm{E}_{(X_s,D_s)}\left[\log \boldsymbol{D}(x_s)\right] + \mathrm{E}_{(X_t,D_t)}\left[\log(\mathbf{1} - \boldsymbol{D}(x_t))\right] \tag{18}$$

Cross-entropy loss is an effective performance measure that provides clear gradient information during the training process, enabling the optimization algorithm to adjust model parameters more effectively. Additionally, it is widely used to quantify the discrepancy between actual labels and model predictions. By minimizing this distance, L_c can improve classification accuracy of the model. The cross-entry loss is given by Eq. (19) as follows:

$$L_c = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} y_{i,c}\log\left(p_{i,c}\right) \tag{19}$$

where, N represents the number of samples and C represents the number of labels. $y_{i,c}$ is the true label of the sample i and $p_{i,c}$ is the corresponding prediction probability.

Cross-domain fault diagnosis is essentially a classification problem. The final optimization objective in the paper is given by Eq. (20).

$$\begin{aligned} &\min_f L_c(f(x)) + \lambda \cdot Domain\ Discrepancy \\ \Rightarrow &\min_f \max_D L_c(f(x)) + \lambda \cdot L_{JMMD}(P,Q) + (1-\lambda)\cdot L_{ad}(D(x)) \end{aligned} \tag{20}$$

Therefore, the final optimization objective function of the model is given by the Eq. (21):

$$Loss = L_c + \lambda \cdot L_{JMMD} + (1-\lambda)\cdot L_{ad} \tag{21}$$

where, λ represents the weight coefficient, which is selected based on different requirements to balance the different loss functions.

3.3. Inference on edge devices

Edge devices have developmental advantages in portability, low energy consumption, scalability, and deployment flexibility, which align with the low computational requirements and low latency characteristics of lightweight networks. Deploying a trained fault diagnosis classification model on these devices enables rapid inference and real-time functionality, thereby offering broad application prospects.

This paper selects the Raspberry Pi 5 (Fig. 6) as the edge device, and the proposed lightweight model as the fault diagnosis model. The fault diagnosis model is deployed on the Raspberry Pi 5 to enable inference on bearing data, achieving portable fault diagnosis.

4. Experiment

- **Case Western Reserve University Data**

4.1. Data set introduction

The CWRU dataset (Smith and Randall, 2015) is collected from the drive-end bearing (SKF6205) with a sampling frequency of 12 kHz, as shown in Fig. 7. The dataset primarily includes four operating conditions (motor loads of 0, 1, 2, and 3 HP). It consists of one healthy state (N) and

Fig. 6. Raspberry Pi 5 used in this paper.

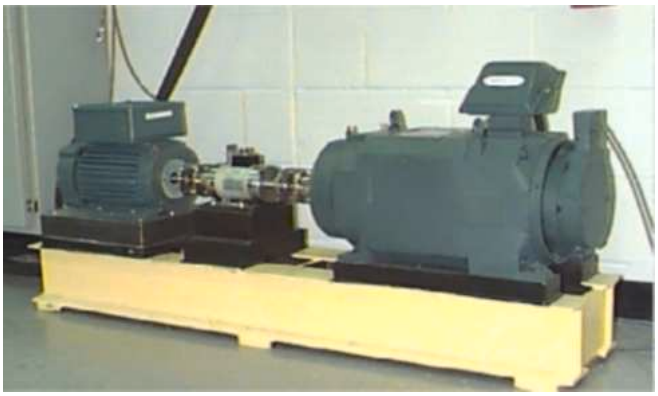


Fig. 7. CWRU data test bench.

three fault states: ball fault (B), inner race fault (IR), and outer race fault (OR). Each fault state corresponds to three fault diameters: 0.007 in., 0.014 in., and 0.021 in.. Therefore, the CWRU dataset can be categorized into nine different fault severity levels and one healthy state. The dataset is shown in Table 1.

In the experiments of this section, each domain contains 2000 samples, with 200 samples per category. Among them, 80 % (1600 samples) of the source domain compose the training set, and 20 % (400 samples) compose the test set. For the target domain, 20 % (400 samples) is used as the validation set.

4.2. Experimental results

4.2.1. Data preprocessing experiment

Since one-dimensional data only contains time-domain or frequency-domain information, the data features are relatively simple, so it is difficult to deeply acquire certain characteristic information. Therefore, two-dimensional time–frequency image data is introduced for preprocessing.

Firstly, this paper uses the STFT to convert vibration signals into two-dimensional time–frequency images, with image sizes of 128 × 128. To demonstrate the advantages of STFT, this paper compares the results of fault diagnosis transfer learning using the above model when the dataset consists of one-dimensional vibration signals, two-dimensional time–frequency images generated by Continuous Wavelet Transform (CWT) (Rioul and Duhamel, 1992), two-dimensional time–frequency images generated by Fast Spectral Correlation (FAST_sc) (Antoni et al., 2017), and two-dimensional time–frequency images generated by STFT. The results are shown in Table 2. Examples of datasets generated by CWT, FAST_sc, and STFT are visible in Fig. 8.

As shown in Fig. 9, the average accuracy of any two-dimensional data is at least 0.71 % higher than that of one-dimensional data. Moreover, the STFT dataset used in this paper improves the average accuracy by 1.17 % compared to one-dimensional data.

Among the two-dimensional datasets, STFT achieves the highest average accuracy, indicating that the time–frequency images of STFT contain the most abundant feature information and are the easiest to be extracted by the network used in this paper. Additionally, the STFT dataset achieves 100 % accuracy in eight transfer tasks, while the CWT and FAST_sc datasets only achieves 100 % accuracy in two of the tasks. Furthermore, the STFT dataset outperforms in nine of the tasks, with an average improvement of 0.5 %. It is evident that the dataset preprocessed by STFT is more suitable for the model in this paper to perform fault diagnosis transfer learning tasks.

In order to avoid erroneous conclusions caused by random fluctuations, the paired *t*-test is used to analyze the accuracies of different models on transfer tasks and the results can be seen in Fig. 10. Here, paired t-tests are performed to compare STFT with 1D, CWT, and FAST_sc separately.

Each model's accuracies of all transfer tasks are aggregated into a single dataset. In the paired *t*-test, the null hypothesis (H_0) is that the average accuracy difference between the two models is 0, and the alternative hypothesis (H_1) is that the average accuracy difference is non-zero. The p-values in the Fig. 10 represents the probability observed under the assumption that the null hypothesis holds true.

Fig. 10 shows the three p-values and they are lower than 0.05, which means H_0 is false and the accuracy difference between STFT and other models is significant. Combing that the accuracy in Table 2, it can be found that the accuracy of STFT is significantly better than two other methods.

4.2.1.1. Model comparison experiment. By comparing with several existing advanced models, this section demonstrates the significant advantages of the proposed model in terms of accuracy and lightweight, thereby proving its potential in practical applications. Several different models are selected, as follows:

Table 1
CWRU data set fault information.

Domain	loads	Diagnosis task
A	0HP	A → B(T1) A → C(T2) A → D(T3)
B	1HP	B → A(T4) B → C(T5) B → D(T6)
C	2HP	C → A(T7) C → B(T8) C → D(T9)
D	3HP	D → A(T10) D → B(T11) D → C(T12)
Class: ①N; ②0.007_IR; ③0.007_B; ④0.007_OR; ⑤0.014_IR; ⑥0.014_B; ⑦0.014_OR; ⑧0.021_IR; ⑨0.021_B; ⑩0.021_OR;		

Table 2
Results of data preprocessing experiments.

Task	1D	CWT	FAST_sc	**STFT**
T1	100	99.5	99.5	**100**
T2	100	99.25	99.25	**100**
T3	98.47	99.75	99.25	**100**
T4	99.58	99.25	**99.75**	99.50
T5	99.86	99.5	99.75	**100**
T6	99.03	**99.75**	99.25	99.50
T7	98.61	99	**99.75**	99
T8	99.44	98.75	99	**100**
T9	99.31	100	99.5	**100**
T10	94.86	99.25	99.25	**99.75**
T11	95.42	98.50	99.5	**100**
T12	99.17	100	99.75	**100**
Average	98.64	99.375	99.438	**99.81**

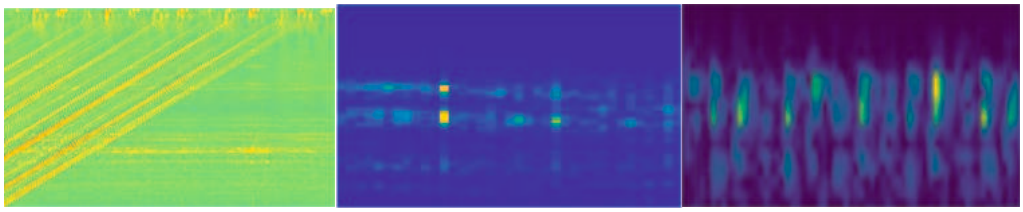


Fig. 8. The time–frequency images of the normal state in Domain A: (a) The CWT image, (b) The FAST_sc image and (c) The STFT image.

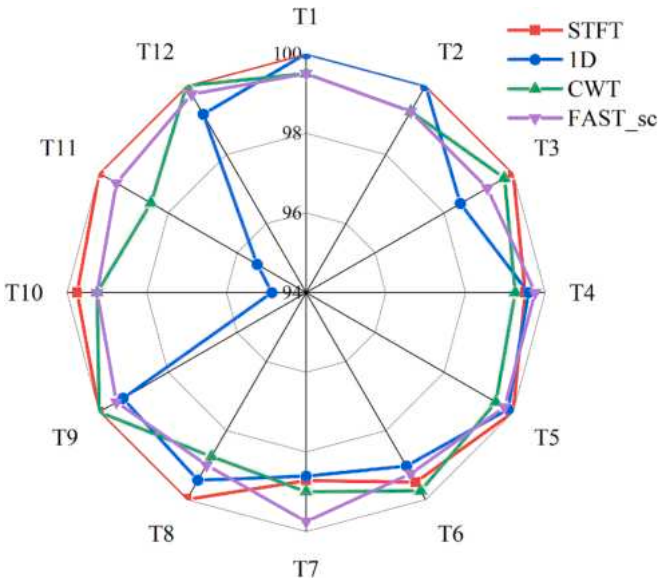


Fig. 9. The CWRU data preprocessing experiments.

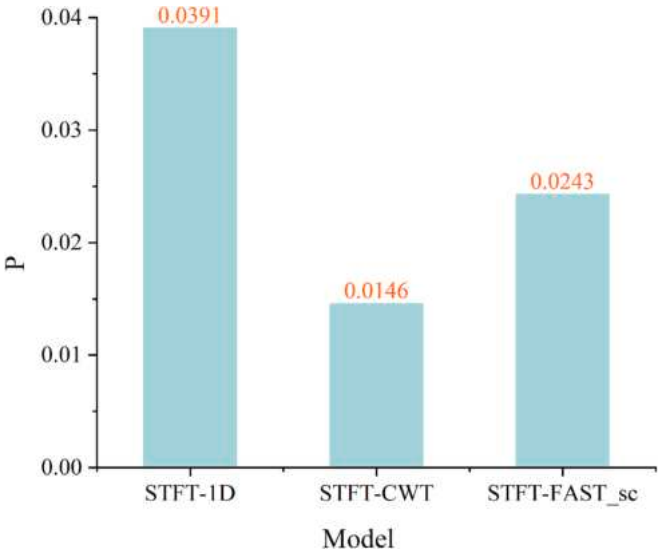


Fig. 10. The paired *t*-test result between different preprocessing methods.

(1) JAN (Long et al., 2017): CNN as the feature network with JMMD
(2) ResNet (He et al., 2016): ResNet18 as the feature network with JMMD
(3) Repvgg (Ding et al., 2021): RepvggA1 as the feature network with JMMD
(4) LRSAN (Yu et al., 2023): LRSAN as the feature network with MK-MMD
(5) WMGRNMM (Yu et al., 2022): WPT (Gao et al., 2011) as the input and MGRN as the feature network with MK-MMD
(6) Ours: The model proposed in this paper

Table 3 shows the accuracy of various transfer tasks for different models. Firstly, as shown in Fig. 11, Ours achieves the highest average accuracy of 99.81 %. In addition, the average accuracy achieved by other models is approximately 98.92 %, which is about 1 % lower than Ours. Compared to Repvgg, which reduces parameters by transforming parameters to accelerate neural network models during inference, Repvgg can only achieve an accuracy of 98.87 %, barely reaching the average level. Although Ours has a slightly lower accuracy of 98.72 % in T7 compared to Repvgg and LRSAN, it achieves 100 % accuracy in three of the four transfer tasks, ensuring the reliability of Ours in completing transfer tasks.

The paired *t*-test is used to compare Ours to other models, and the results are shown in the Fig. 12. From this figure, except the p-value between Ours and LRSAN, all the p-values are lower than 0.05, which means that Ours has the significant differences with other models. By comparing the accuracy in Table 3, it could be concluded that the accuracy of Ours is significantly better than these models.

Although the difference in accuracy between ours and LRSAN is not significant, the accuracy distribution (Fig. 13) shows that the average accuracy of Ours is higher than LRSAN. Meanwhile, the standard deviation of the proposed model's accuracy is lower than LRSAN's, which means the accuracy of Ours is more stable and the proposed model is more suitable for practical application.

Table 4 presents the number of parameters, floating point operations (FLOPs), and average accuracy of different models. As shown in the parameter comparison chart (Fig. 14), ours achieves high accuracy with a smaller number of parameters and computational complexity. Although JAN has minimal parameters (2.38 MB), the small number of parameters in its feature network results in an accuracy of only 93.12 % for transfer tasks, making it difficult to achieve a convincing level. Ours has the second smallest number of parameters (7.18 MB), which is far

Table 3
Results of model comparison experiments.

Task	**Ours**	JAN	ResNet	Repvgg	LRSAN	WMG RNMM
T1	**100**	92.09	98.31	98.89	100	97.78
T2	**100**	91.31	99.68	99.12	99.72	100
T3	**100**	87.15	99.42	99.36	99.72	99.72
T4	99.50	94.96	98.37	98.47	**99.72**	97.50
T5	**100**	95.86	98.59	99.03	100	100
T6	99.50	92.14	99.26	99.12	**99.72**	97.50
T7	**99**	92.44	98.36	98.94	98.89	97.22
T8	**100**	93.69	98.92	98.72	98.89	97.50
T9	**100**	95.86	99.21	99.23	100	100
T10	**99.75**	92.05	96.08	98.21	99.17	98.61
T11	**100**	92.95	97.13	98.16	99.44	97.22
T12	**100**	96.84	99.76	99.25	100	100
Ave	**99.81**	93.12	98.59	98.87	99.61	98.59

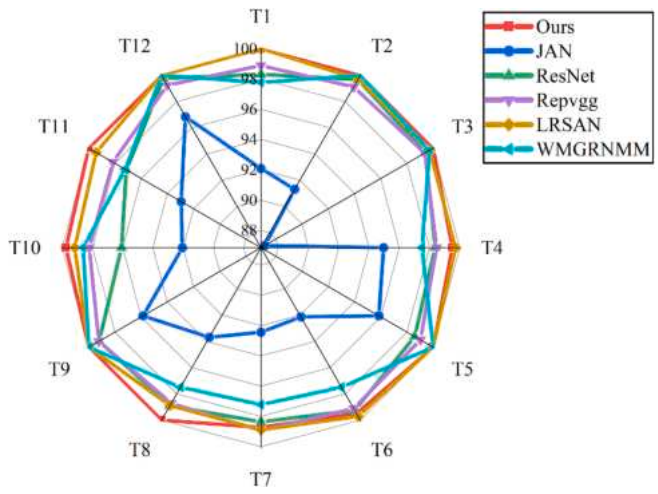


Fig. 11. The CWRU comparison experiments of different models.

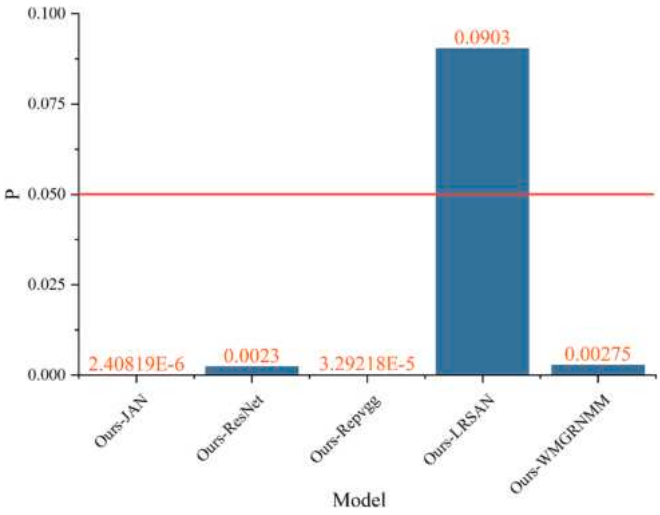


Fig. 12. The paired *t*-test result between different models.

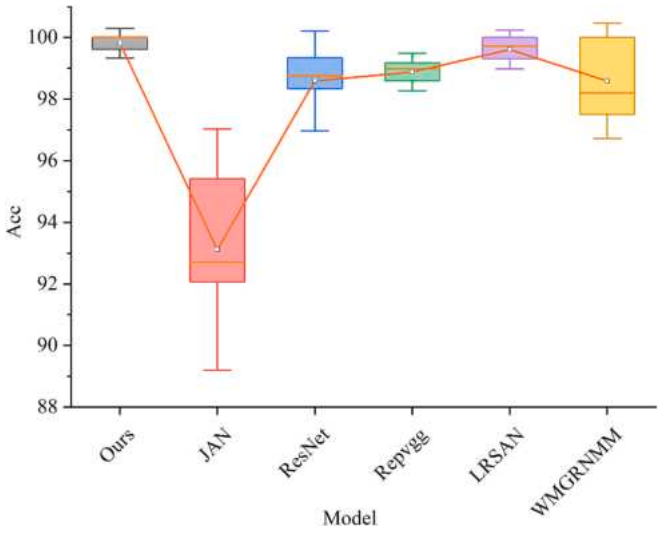


Fig. 13. The accuracy distribution of transfer tasks.

Table 4
Results of parameter comparison experiments.

Model	**Ours**	ResNet18	JAN	Repvgg
Parameters (MB)	7.18	42.61	2.38	55.12
FLOPs (10^6)	**93.99**	592.99	404.32	860.77
Ave Acc	**99.79**	98.59	93.12	98.87

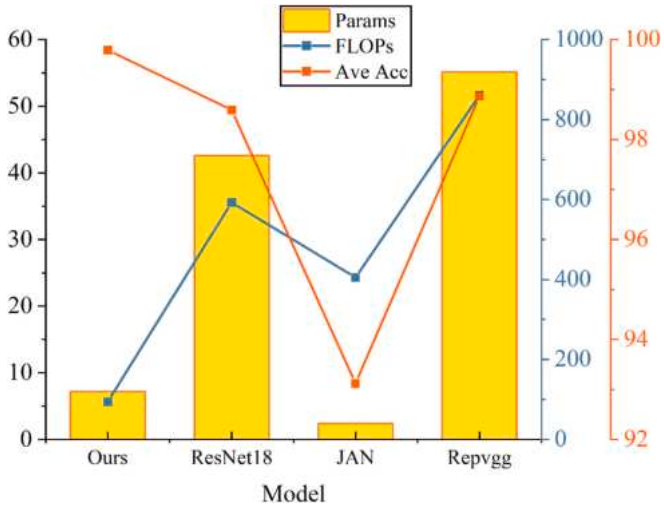


Fig. 14. The comparison chart of different models.

lower than the other models. However, in terms of FLOPs, it is 76.75 % lower than CNN, while the accuracy is increased by approximately 6.67 %. Compared to Repvgg, the model achieves an 86.97 % reduction in parameters, an 89.08 % reduction in FLOPs, and a 0.92 % increase in accuracy.

4.2.1.2. Ablation experiment. This section compares several different design methods, as illustrated in Table 5, to verify the rationality of model's module design in achieving a balance between accuracy and lightweight performance. Table 5 describes the specific content of different design methods, and Table 6 shows the impact of different designs on parameters, FLOPs, and the accuracy of various transfer tasks.

M0 represents the model in this paper. M1 represents replacing the DSConv in the improved lightweight residual blocks with traditional convolution layers. M2 represents changing the two convolution layers in the improved lightweight residual blocks to DSConv and traditional convolution respectively. M3 represents changing the two convolution layers to traditional convolution and DSConv respectively.

As shown in Fig. 15, M0 has the smallest number of parameters and FLOPs with an average accuracy of 99.81 %, second only to M2's 99.77 %. Compared to M1, M0 has a significant advantage, with 36.11 MB fewer parameters, a 56.19 % reduction in FLOPs, and an increase in average accuracy. Similarly, compared to M2 and M3, M0 maintains accuracy while reducing parameters and FLOPs. Therefore, among the various models, M0 ensures accuracy while demonstrating lightweight characteristics. In summary, this paper's model is a fault diagnosis transfer learning model that balances accuracy and lightweight design.

Table 5
Introduction to different models.

M0	The model in this paper
M1	Model based on residual blocks with traditional convolution
M2	Model based on residual blocks with DSConv + Conv
M3	Model based on residual blocks with Conv + DSConv

Table 6
Results of ablation experiments.

STFT	**M0**	M1	M2	M3
Para (MB)	**7.18**	43.29	27.38	22.12
FLOPs (10^6)	**93.99**	214.58	160.55	148.03
T1	**100**	100	100	100
T2	**100**	100	100	100
T3	**100**	100	100	100
T4	99.50	99.34	99.48	**100**
T5	**100**	100	100	100
T6	99.50	99.29	**99.52**	99.75
T7	**99**	**99.07**	98.56	98.50
T8	**100**	100	100	100
T9	**100**	100	100	99.5
T10	**99.75**	99.31	99.72	99.25
T11	**100**	100	100	100
T12	**100**	100	100	100
Ave	**99.81**	99.74	99.77	99.75

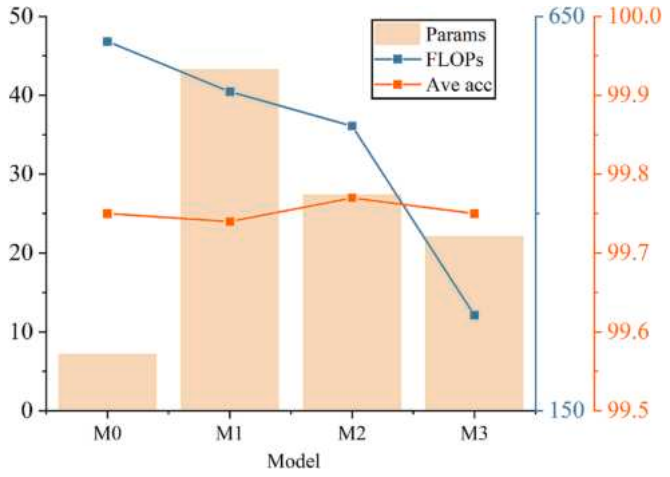


Fig. 15. The comparison chart of ablation experimental models.

4.3. Validation on edge deployment

This section deploys the model on a Raspberry Pi to validate that the lightweight nature of the proposed model is sufficient to be driven by lightweight devices. The environment configuration for the Raspberry Pi in this section is: Python 3.9, PyTorch 1.11. Firstly, a randomly selected fault state image from the CWRU STFT dataset is chosen as the input for fault diagnosis on the Raspberry Pi. Then, the corresponding transfer model trained on the GPU is selected, and the model is run on the Raspberry Pi to obtain the final diagnosis result.

In the example, this section selects a STFT time–frequency image from domain A with a fault category of 8 (0.021_IR) as the input. Then, transfer learning models T4, T7, and T10 are selected to simulate three operating conditions: B-A, C-A, and D-A.

Finally, the Raspberry Pi runs the three models separately for fault diagnosis under the three operating conditions. The final diagnosis results are shown in Fig. 16, with results visible in Table 7.

As shown in Table 7, the accuracy of the three cross-condition fault diagnosis tasks is maintained around 98 %, with a diagnosis time of only 0.13 s. This proves the reliability of our model for cross-domain fault diagnosis on edge devices.

- **HUST bearing Data**

4.4. Data set introduction

The HUST bearing dataset (Zhao et al., 2024) is collected from the Spectra-Quest mechanical fault test bench, as shown in Fig. 17. The components on the test bench from left to right are speed control, motor, shaft, acceleration sensor, bearing, and data acquisition. The HUST bearing dataset primarily includes four operating conditions: 65 Hz, 70 Hz, 75 Hz, and 80 Hz. For convenience, these conditions are named A, B, C, and D, respectively. Under each operating condition, the data is divided into nine fault categories, as shown in Table 8.

In the experiments of this section, the dataset contains 2000 samples, with 200 samples per category. 80 % (1600 samples) of the source domain is the training set, and 20 % (400 samples) is the test set. For the target domain, 20 % (400 samples) is the validation set.

4.5. Experimental results

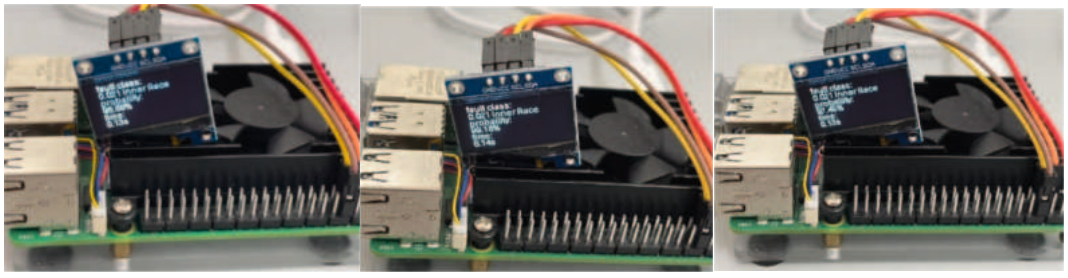
4.5.1.1. Data preprocessing experiment

To verify advantages of STFT selected in this paper, the accuracy of fault diagnosis using CWT, FAST_sc, and STFT under the proposed model is compared. The results are detailed in Table 9.

As shown in Fig. 18, it is evident that in terms of two-dimensional data preprocessing methods, STFT has significantly higher accuracy and stability. In terms of the average accuracy, STFT improves by approximately 20 % and 40 % compared to FAST_sc and CWT, respectively. From the perspective of various transfer tasks, the accuracy of STFT is generally around 95 %, with the worst D → A task still achieving an accuracy of 82.78 %, far higher than the other two methods. In comparison, the accuracy of CWT ranges from 31.94 % to 86.39 %, indicating a large distribution interval and poor stability. FAST_sc has an accuracy range of approximately 50 % to 90 %, showing improved

Table 7
Fault diagnosis results deployed on Raspberry Pi.

Task	B-A	C-A	D-A
Result	0.021_IR	0.021_IR	0.021_IR
Probability	98.89 %	99.16 %	97.46 %
Time (s)	0.13	0.14	0.13



(a) B-A (b) C-A (c) D-A

Fig. 16. Fault diagnosis with Raspberry Pi: (a) Transfer from B to A, (b) Transfer from C to A, (c) Transfer from D to A.

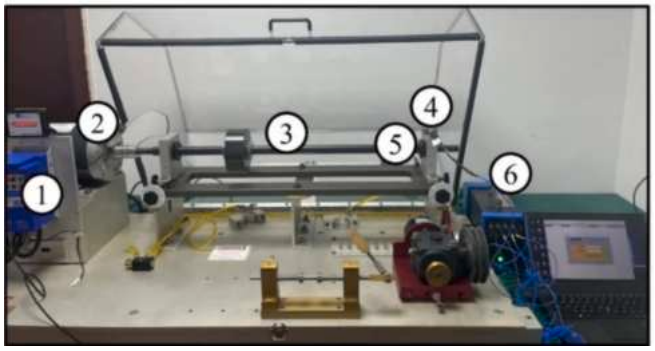


Fig. 17. HUST data test bench.

Table 8
Fault categories of HUST data.

Domain	A:65 Hz, B:70 Hz, C:75 Hz, D:80 Hz
Class	1) Normal 2) Medium inner 3) Sever inner 4) Medium outer 5) Sever outer 6) Medium ball 7) Sever ball 8) Medium combination 9) Sever combination
Combination indicates that both the outer race and inner race have faults	

Table 9
Results of data processing experiments.

HUST Bearing	CWT	FAST_sc	**STFT**
A → B	86.39	80.83	**96.39**
A → C	65.00	66.67	**93.33**
A → D	41.67	56.94	**93.72**
B → A	68.33	85.83	**95.83**
B → C	50.83	80.83	**93.33**
B → D	41.11	61.94	**97.22**
C → A	38.89	75.83	**94.44**
C → B	72.23	84.17	**92.50**
C → D	48.90	85.56	**95.38**
D → A	31.94	69.17	**82.78**
D → B	41.39	76.94	**90.28**
D → C	68.89	73.06	**92.78**
Avg	54.63	74.81	**93.16**

stability and accuracy compared to CWT, but still weaker than STFT.

Similarly, the paired *t*-test is used to analyze the accuracy of three preprocessing methods. From Fig. 19, the p-value between STFT and CWT is much lower than 0.05, which means the accuracy of STFT is significantly different from the accuracy of CWT. The average accuracy of STFT is 38.53 % higher than the accuracy of CWT, demonstrating the superiority of STFT. With the same analytical approach, the accuracy of STFT is significantly better than the accuracy of FAST_sc, because the p-value is 0.00004 and the average accuracy of STFT is 18.35 % higher than FAST_sc.

4.5.1.2. Model comparison experiment

To verify the reliability and parameter advantage of our model, the above CNN, Repvgg, and ResNet18 are still selected, and the Ours (M1) is also added as a comparative experimental model. The results are shown in Table 10.

As shown in Fig. 20, CNN and Repvgg exhibit significantly lower accuracy compared to other models, with an average accuracy of approximately 80 %. In addition, as shown in Fig. 21, the accuracy of Ours (M1) is slightly better, with an average accuracy about 1 % higher

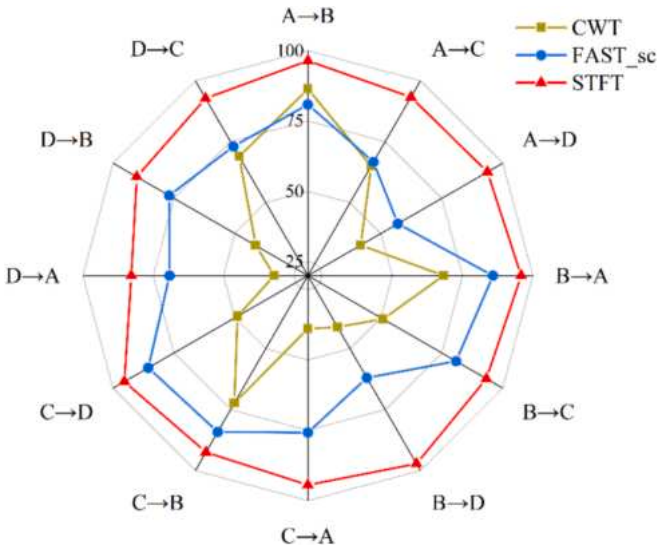


Fig. 18. The HUST data processing results.

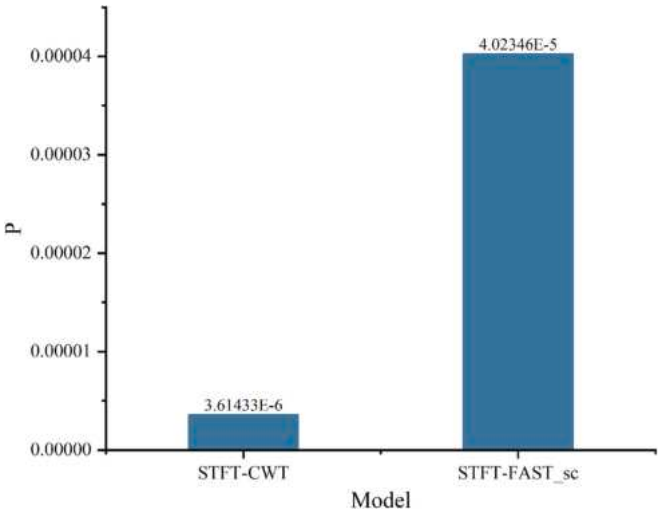


Fig. 19. The paired *t*-test result between different preprocessing methods.

Table 10
Results of HUST comparison experiments.

HUST Bearing	CNN	Repvgg	ResNet18	Ours(M1)	**Ours**
A → B	89.17	86.94	**96.67**	95.83	96.39
A → C	84.44	76.39	94.72	**94.72**	93.33
A → D	79.17	90.83	93.61	**95.28**	93.72
B → A	92.78	88.61	94.72	**97.50**	95.83
B → C	90.56	81.94	95.56	**97.22**	93.33
B → D	86.94	72.22	95.83	97.22	**97.22**
C → A	84.44	90.28	92.78	93.06	**94.44**
C → B	85.56	88.61	93.33	**95.56**	92.50
C → D	85.28	83.61	94.72	94.72	**95.38**
D → A	68.33	63.06	84.72	**87.50**	82.78
D → B	77.22	73.06	88.61	**90.83**	90.28
D → C	78.61	69.17	91.94	90.56	**92.78**
Para (MB)	2.38 M	55.12 M	42.61 M	42.72 M	6.21 M
FLOPs (10^6)	404.85	860.77	592.99	510.94	**93.99**
Avg acc	83.54	80.39	93.10	**94.19**	93.16

than Ours. However, in terms of parameters and FLOPs, it far exceeds Ours, and the improvement in accuracy does not match its FLOPs. Similarly, Ours has advantages over ResNet18 in all aspects. Parameters

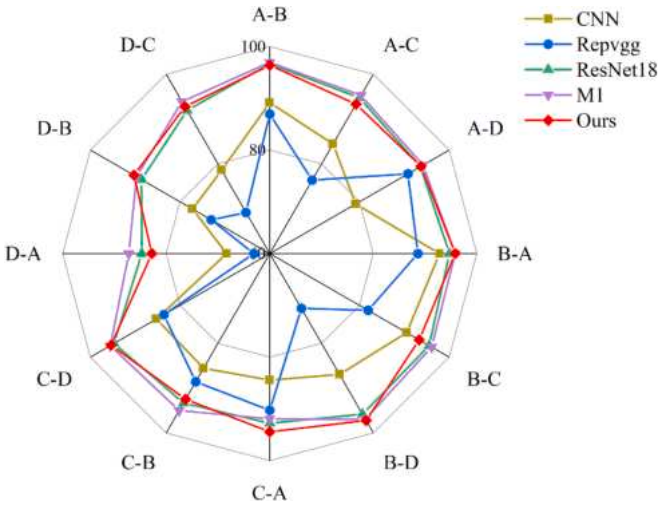


Fig. 20. The HUST accuracy of different models.

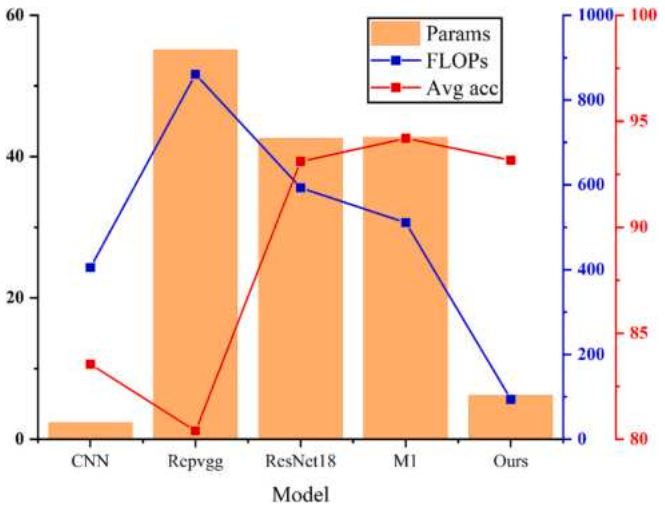


Fig. 21. The comparison chart of the model data.

and FLOPs are both reduced by about 85 %, while the accuracy is slightly improved.

Consistent with the above analysis, the paired *t*-test is used here to analyze the accuracy of the models and evaluate their differences, as shown in Table 11. This table shows the p-values from the paired *t*-test between Ours and other models. It is easy to see that the accuracy of Ours is significantly different from the accuracy of CNN and Repvgg. The higher average accuracy of ours compared to the two other models proves that Ours is significantly better than CNN and Repvgg. Although the performances of Ours, ResNet18 and M1 are similar from Fig. 22 and Table 11, it indicates that the performance of the proposed model is more powerful, because the proposed model reduces the parameters and FLOPs by about 85 %.

4.6. Validation on edge deployment

To further validate the use of our model on the Raspberry Pi, the same environment and methods as above are selected. In this section, a STFT time–frequency image with an operating condition of C and a fault category of medium outer is chosen as the input for this device. Then, three cross-condition fault diagnosis tasks are simulated, and fault diagnosis is performed using the Raspberry Pi. The results are shown in Table 12 and Fig. 23.

Table 11
The p-values from the paired *t*-test between two models.

Model	Ours-CNN	Ours-Repvgg	Ours-ResNet18	Ours-Ours(M1)
p-value	5.54E-6	1.25E-4	0.87	0.13

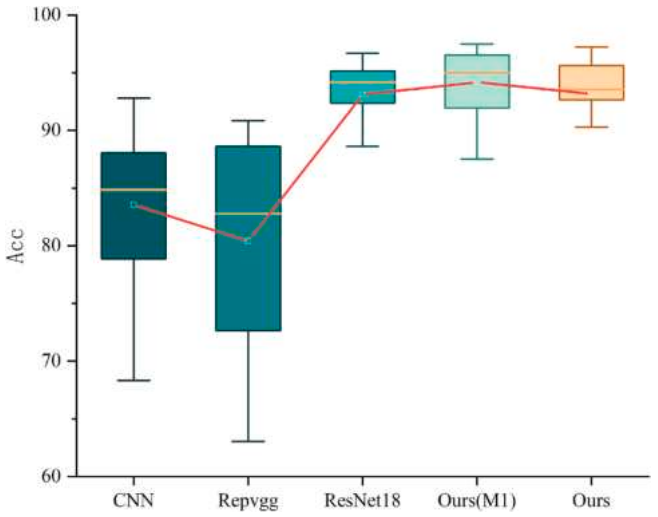


Fig. 22. The accuracy distribution of transfer tasks.

5. Limitations

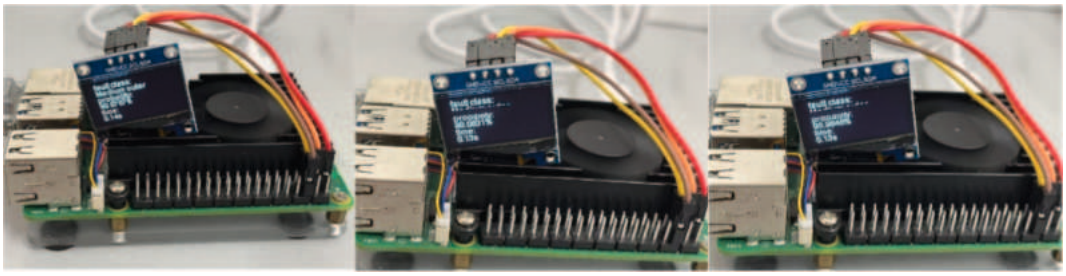
Although this paper introduces an improved method which demonstrates its strong performance for cross-domain fault diagnosis, it is still necessary to admit that there are some small limitations. Two important points are summarized as follows. Firstly, its diagnostic accuracy may degrade when facing extreme domain (e.g., severe noise in data or extremely small sample data). This limitation arises as the domain adaptation method primarily aligns feature distributions but may struggle with highly divergent source and target domains. The future works should explore a more appropriate method for the extreme domain adaptation. Secondly, edge computing devices may have difficulties in the diagnosis of unknown faults. Due to the limitation of computation resources, models deployed on the edge devices are difficult to infer unknown fault categories in real-time. Future studies should focus on the real-time inference on unknown fault categories with the more advanced models that can be deployed on edge devices.

6. Conclusion

Considering the practical engineering needs, this paper proposes an improved lightweight residual network model that can be deployed on edge devices for the cross-domain fault diagnosis. Firstly, the data is preprocessed using STFT to generate time–frequency images, which serve as the model's input. Then, two lightweight residual blocks are designed based on DSConv and GN. Additionally, an improved lightweight residual network is constructed with SCConv, serving as the feature extraction network for transfer learning. Subsequently, the domain adaptation module is utilized to form the final cross-domain fault diagnosis model. Finally, through two experimental studies, it is

Table 12
Fault diagnosis results deployed on Raspberry Pi.

Task	A-C	B-C	D-C
Result	Medium outer	Medium outer	Medium outer
Probability	99.87 %	99.06 %	98.98 %
Time (s)	0.14	0.13	0.13



(a) A-C (b) B-C (c) D-C

Fig. 23. Fault diagnosis with Raspberry Pi. (a) Transfer from A to C, (b) Transfer from B to C, (c) Transfer from D to C.

found that the model in this paper, compared to other models, has advantages such as high accuracy, small number of parameters, and high computational efficiency.

Declaration of competing interest

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

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Nos. 52305085, 52105111 and 52405076), China Postdoctoral Science Foundation (2023 M740021), Guangdong Basic and Applied Basic Research Foundation (Grant 2025A1515012256), Industry-Academia Cooperation Project from the Guangdong Institute of Special Equipment Inspection and Research Shunde Branch (XTJ-KY01-202503-030).

Data availability

The data that has been used is confidential.

References

Yu, D., Cheng, J., & Yang, Y. (2005). Application of EMD method and Hilbert spectrum to the fault diagnosis of roller bearings. *Mechanical Systems and Signal Processing, 19*, 259–270.

Burrus, C. S., Gopinath, R. A., & Guo, H. (1998). Wavelets and wavelet transforms. *Rice University, Houston Edition*, 98.

Cocconcelli, M., Zimroz, R., Rubini, R., & Bartelmus, W. (2012). STFT based approach for ball bearing fault detection in a varying speed motor. In *Condition Monitoring of Machinery in Non-Stationary Operations: Proceedings of the Second International Conference" Condition Monitoring of Machinery in Non-Stationnary Operations" CMMNO'2012*, 41-50.

Boashash, B., & Black, P. (1987). An efficient real-time implementation of the Wigner-Ville distribution. *IEEE Transactions on Acoustics, Speech, and Signal Processing, 35*, 1611–1618.

Cong, F., Chen, J., Dong, G., & Zhao, F. (2013). Short-time matrix series based singular value decomposition for rolling bearing fault diagnosis. *Mechanical Systems and Signal Processing, 34*, 218–230.

Nguyen, V. H., Rutten, C., & Golinval, J.-C. (2012). Fault diagnosis in industrial systems based on blind source separation techniques using one single vibration sensor. *Shock and Vibration, 19*, 795–801.

Bao, H., Wei, Y., Zhang, Z., Wang, J., Zhang, G., & Tian, Z. (2022). EarlyWeak bearing fault diagnosis method based on EMD-CSF. *Noise and Vibration Control, 42*, 105.

Wohlberg, B. (2015). Efficient algorithms for convolutional sparse representations. *IEEE Transactions on Image Processing, 25*, 301–315.

ALTobi, M. A. S., Bevan, G., Wallace, P., Harrison, D., & Ramachandran, K. (2019). Fault diagnosis of a centrifugal pump using MLP-GABP and SVM with CWT. *Engineering Science and Technology, an International Journal, 22*, 854–861.

Ma, M., Sun, C., & Chen, X. (2018). Deep coupling autoencoder for fault diagnosis with multimodal sensory data. *IEEE Transactions on Industrial Informatics, 14*, 1137–1145.

Wang, Y., Pan, Z., Yuan, X., Yang, C., & Gui, W. (2020). A novel deep learning based fault diagnosis approach for chemical process with extended deep belief network. *ISA Transactions, 96*, 457–467.

Han, Y., Ding, N., Geng, Z., Wang, Z., & Chu, C. (2020). An optimized long short-term memory network based fault diagnosis model for chemical processes. *Journal of Process Control, 92*, 161–168.

Wen, L., Li, X., Gao, L., & Zhang, Y. (2017). A new convolutional neural network-based data-driven fault diagnosis method. *IEEE Transactions on Industrial Electronics, 65*, 5990–5998.

Abdeljaber, O., Avci, O., Kiranyaz, S., Gabbouj, M., & Inman, D. J. (2017). Real-time vibration-based structural damage detection using one-dimensional convolutional neural networks. *Journal of Sound and Vibration, 388*, 154–170.

Ma, S., Chu, F., & Han, Q. (2019). Deep residual learning with demodulated time-frequency features for fault diagnosis of planetary gearbox under nonstationary running conditions. *Mechanical Systems and Signal Processing, 127*, 190–201.

Tan, C., Sun, F., Kong, T., Zhang, W., Yang, C., & Liu, C. (2018). A survey on deep transfer learning. In *Artificial Neural Networks and Machine Learning–ICANN 2018: 27th International Conference on Artificial Neural Networks, Rhodes, Greece, October 4-7, 2018, Proceedings*, Part III 27, 270-279.

Li, C., Zhang, S., Qin, Y., & Estupinan, E. (2020). A systematic review of deep transfer learning for machinery fault diagnosis. *Neurocomputing, 407*, 121–135.

Shao, S., McAleer, S., Yan, R., & Baldi, P. (2018). Highly accurate machine fault diagnosis using deep transfer learning. *IEEE Transactions on Industrial Informatics, 15*, 2446–2455.

Gao, J., Han, H., Ren, Z., & Fan, Y. (2021). Fault diagnosis for building chillers based on data self-production and deep convolutional neural network. *Journal of Building Engineering, 34*, Article 102043.

Zhao, K., Jia, F., & Shao, H. (2023). A novel conditional weighting transfer Wasserstein auto-encoder for rolling bearing fault diagnosis with multi-source domains. *Knowledge-Based Systems, 262*, Article 110203.

Yang, S., Zhou, Y., Chen, X., Deng, C., & Li, C. (2023). Fault diagnosis of wind turbines with generative adversarial network-based oversampling method. *Measurement Science and Technology, 34*, Article 044004.

Wang, G., Zhao, S., Chen, J., & Zhong, Z. (2023). A novel compound fault diagnosis method for rolling bearings based on graph label manifold metric transfer. *Measurement Science and Technology, 34*, Article 065010.

Meng, Z., Zhao, Z., Zhu, B., & Fan, F. (2022). Online diagnosis for rolling bearings based on multi-channel convolution and transfer learning. *Measurement Science and Technology, 33*, Article 115116.

Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1251–1258.

Sifre, L., & Mallat, S. (2014). Rigid-motion scattering for texture classification. *arXiv preprint arXiv:1403.1687*.

Wu, Y., & He, K. (2018). Group normalization. In *In Proceedings of the European Conference on Computer Vision* (pp. 3–19).

Bjorck, N., Gomes, C. P., Selman, B., & Weinberger, K. Q. (2018). Understanding batch normalization. *Advances in Neural Information Processing Systems, 31*.

Li, J., Wen, Y., & He, L. (2023). Scconv: Spatial and channel reconstruction convolution for feature redundancy. In *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 6153–6162).

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 770–778).

Long, M., Zhu, H., Wang, J., & Jordan, M. I. (2017). Deep transfer learning with joint adaptation networks. *In International Conference on Machine Learning*, 2208–2217.

Smith, W. A., & Randall, R. B. (2015). Rolling element bearing diagnostics using the Case Western Reserve University data: A benchmark study. *Mechanical Systems and Signal Processing, 64*, 100–131.

Rioul, O., & Duhamel, P. (1992). Fast algorithms for discrete and continuous wavelet transforms. *IEEE Transactions on Information Theory, 38*, 569–586.

Antoni, J., Xin, G., & Hamzaoui, N. (2017). Fast computation of the spectral correlation. *Mechanical Systems and Signal Processing, 92*, 248–277.

Ding, X., Zhang, X., Ma, N., Han, J., Ding, G., & Sun, J. (2021). Repvgg: Making vgg-style convnets great again. In *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 13733–13742).

Yu, X., Wang, Y., Liang, Z., Shao, H., Yu, K., & Yu, W. (2023). An adaptive domain adaptation method for rolling bearings' fault diagnosis fusing deep convolution and

self-attention networks. *IEEE Transactions on Instrumentation and Measurement, 72*, 1–14.

Yu, X., Liang, Z., Wang, Y., Yin, H., Liu, X., Yu, W., & Huang, Y. (2022). A wavelet packet transform-based deep feature transfer learning method for bearing fault diagnosis under different working conditions. *Measurement, 201*, Article 111597.

Gao, R. X., Yan, R., Gao, R. X., & Yan, R. (2011). Wavelet packet transform. *Wavelets: Theory and Applications for Manufacturing*, 69-81.

Zhao, C., Zio, E., & Shen, W. (2024). Domain generalization for cross-domain fault diagnosis: An application-oriented perspective and a benchmark study. *Reliability Engineering & System Safety, 109964*.