A New Adversarial Domain Generalization Network Based on Class Boundary Feature Detection for Bearing Fault Diagnosis

Jingde Li, Changqing Shen, *Senior Member, IEEE*, Lin Kong, Dong Wang, *Member, IEEE*, Min Xia, *Senior Member, IEEE*, and Zhongkui Zhu

***Abstract*— In recent years, many researchers have attempted to achieve cross-domain diagnosis of faults through domain adaptation (DA) methods. However, owing to the complex physical environments, applications of DA-based approach are not guaranteed to unknown operating environments. Some existing domain generalization (DG) methods require enough fully labeled source domains to train, which are often unavailable in practical settings. In this study, an adversarial domain generalization network (ADGN) based on class boundary feature detection is proposed. The ADGN can diagnose faults in unknown operating environments, and only one fully labeled domain is used in training. Although ADGN has to access fully unlabeled auxiliary domains, a large number of unlabeled samples exist under actual working conditions. In our method, fuzzy features at a classification boundary are detected by maximizing the classifier differences. Better feature mapping functions and domain-invariant features are obtained by adversarial training. As the training proceeds, the differences in the distribution of features among the source, auxiliary, and unknown domains become smaller so domain-invariant features can be used for fault diagnosis in unknown operating environments. Comprehensive experiments showed that ADGN can achieve higher fault diagnosis accuracies than other methods when only one fully labeled domain is used in an unknown operating environment. The ADGN can even cope comfortably with complex transfer tasks with different speeds and loads.**

***Index Terms*— Adversarial learning, bearing, domain generalization (DG), fault diagnosis, transfer learning.**

I. Introduction

WITH the development of industrial level, intelligent maintenance of industrial equipment such as engines, trains, and aircraft is becoming increasingly common [1]. Bearings, as an important component of industrial equipment, often work under harsh conditions, and the health of bearings is seriously threatened, resulting in industrial equipment not working properly [2]–[4]. Therefore, ensuring the healthy and stable operation of bearings is crucial for industrial production [5].

Automatic detection and identification of machine faults can be achieved using artificial intelligence methods with the development of deep learning [6]. For example, convolutional neural networks (CNNs) [7], deep belief networks (DBNs) [8], recurrent neural networks [9], auto-encoders (AEs) [10], and multilayer perceptron [11] have been introduced for fault diagnosis. Wang *et al.* [12] used a multi-headed attention mechanism to optimize the CNN model and used it successfully for fault diagnosis. Shao *et al.* [13] improved for AEs and, on this basis, they proposed the method of ensemble deep auto-encoders (EDAEs) with high diagnostic accuracy. An *et al.* [14] introduced long short-term memory networks to fault diagnosis and built a network model to minimize the effect of rotational speed, thus improving the diagnostic accuracy. Pan *et al.* [15] used dynamic learning rate to adjust the training strategy of SAE, effectively improving the diagnostic accuracy. Liu *et al.* [16] improved DBN by proposing a multi-layer feature fusion method to improve the model generalization capability. However, the above method has significant limitations. For one, the model trained by the above method only works for samples with the same distribution as the training data. For another, a large number of labeled samples are needed to train the model.

Bearings often work under different operating conditions, that is, the distribution of training data and test data varies, and the expected results cannot be achieved using the above method. Cross-domain diagnostics are urgently needed. Researchers introduce domain adaptation (DA) [17] to enable cross-domain diagnosis. DA enhances the target domain training by mapping data features from different domains (e.g., two different datasets) to the same feature space. Currently, distributed distance is widely used in DA, for example, maximum mean discrepancy (MMD) [18], correlation alignment (CORAL) [19], joint MMD (JMMD) [20], and local MMD (LMMD) [21]. Siahpour *et al.* [18] used MMD as a regularization term for cross-domain diagnosis. Qin *et al.* [19] combined CORAL with domain adversarial loss to effectively

Manuscript received February 14, 2022; accepted March 21, 2022. Date of publication April 1, 2022; date of current version April 19, 2022. This work was supported in part by the National Natural Science Foundation of China under Grant 51875375, Grant 51875376, and Grant 51975355; and in part by the Research Project of State Key Laboratory of Mechanical System and Vibration, Shanghai Jiao Tong University, under Grant MSV202104. The Associate Editor coordinating the review process was Dr. Arunava Naha. *(Corresponding author: Changqing Shen.)*

Jingde Li, Changqing Shen, and Zhongkui Zhu are with the School of Rail Transportation, Soochow University, Suzhou 215131, China (e-mail: jdlhl@stu.suda.edu.cn; cqshen@suda.edu.cn; zhuzhongkui@suda.edu.cn).

Lin Kong is with Chang Guang Satellite Technology Company Ltd., Changchun 130000, China (e-mail: konglin@charmingglobe.com).

Dong Wang is with the Department of Industrial Engineering and Management, and the State Key Laboratory of Mechanical System and Vibration, Shanghai Jiao Tong University, Shanghai 200000, China (e-mail: dongwang4-c@sjtu.edu.cn).

Min Xia is with the Department of Engineering, Lancaster University, Lancaster LA1 4YW, U.K. (e-mail: m.xia3@lancaster.ac.uk).

Digital Object Identifier 10.1109/TIM.2022.3164163

solve the distribution mismatch between source and target domain features. Liu *et al.* [20] used JMMD for joint distribution adaption, further improving the performance of cross-domain diagnosis. Ramezani [21] narrowed the conditional distribution differences using LMMD, aiming to narrow the intra-class differences. Another group of researchers reduces the domain differences from the perspective of adversarial learning. Jiao *et al.* [22] conducted DA by adversarial learning strategy and introduced a dynamic fitness factor to balance edge distribution and conditional distribution.

However, the DA method still experiences drawbacks in solving the cross-domain diagnosis problem. DA can only achieve cross-domain diagnosis for a specific target domain without good generalization. The diagnostic performance of DA is greatly reduced in an unknown operating environment. The reason is that DA models inevitably overfit when they generalize knowledge from one specific domain to another. In a real environment, the operating environment at the time of failure is not unique, and diagnosis using DA requires training a specific model for each operating environment. Therefore, the research of fault diagnosis based on domain generalization (DG) [23] is necessary. DG learns generic network models with the help of sample features from multiple source domains. Li *et al.* [24] obtained generic features by adversarial training and achieved high diagnostic accuracy for the diagnosis of unknown operating environments. Yang *et al.* [25] minimized the domain differences by jointly monitoring the central loss and softmax loss. Han *et al.* [26] considered intrinsic and extrinsic generalization goals and proposed a hybrid diagnostic network.

Although DG methods can diagnose faults for unknown operating environments, the existing DG methods require many fully labeled source domains for training. In practice, the unlabeled samples are large, whereas the labeled ones are few. After related experimental studies, for the existing DG methods, the diagnostic accuracy of DG is unsatisfactory when the number of fully labeled source domains involved in training is small. For this reason, we put forward our method.

This study mainly investigates the generalization of only one fully labeled source domain and proposes the adversarial domain generalization network (ADGN). Several fully unlabeled domains, which we call auxiliary domains, and one fully labeled source domain are used by ADGN. The neural network of ADGN consists of two modules: a feature generator (G) and two fault classifiers (C). During training in the auxiliary domain, the two fault classifiers attempt to make different decisions from the other fault classifier while learning the diagnostic knowledge. G wants to reconcile the contradiction between the two fault classifiers. Through the adversarial game of G and C, domain-invariant features are finally obtained. In the above training process, the auxiliary domain only needs to provide the sample information without labeling information. The accuracy of the diagnosis is supported by the source domain.

This work has resulted in the following contributions.

1) A new DG framework is proposed. It uses only one fully labeled source domain and some fully unlabeled auxiliary domains for fault diagnosis in unknown operational environments.

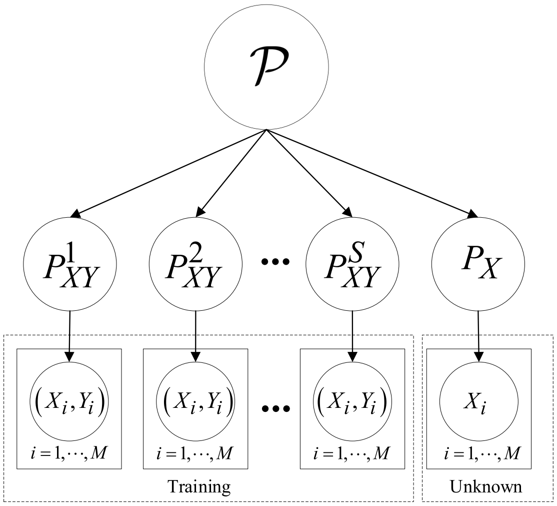


Fig. 1. Schematic of general DG [23].

2) A class boundary feature detection strategy is proposed to enable the model to learn domain-invariant features. Moreover, the strategy can make full use of the supervised capability of the fully labeled domain, to confer the model with higher diagnostic accuracy.
3) The auxiliary domain is introduced to obtain the ideal classification boundary by the adversarial training of the classifier and feature generator.

The rest of this article is organized as follows. Section II popularizes the basics of DG, and Section III details the idea of ADGN and the training steps. Section IV presents the experimental results and the analysis. Finally, Section V draws the conclusions.

II. Preliminaries

A. Domain Generalization

The purpose of DG is to generalize the model learned in the source domain to any new domain. To learn models with sufficient generalization capability, the training data are usually required to come from domains with different data distributions.

Fig. 1 shows the general form of DG, in which all the domains in training are fully labeled domains and different domains follow different feature distributions. In our work, only one domain is fully labeled, and the rest of the domains are unlabeled as auxiliary domains in training.

Each domain D contains a sample space X and a marginal distribution $P(X)$ in this space [27]. Each domain obeys its own data distribution, and all contain M samples. The DG problem has S domains with different but similar data distributions, and it is denoted as $\mathcal{D}_s = \{\mathcal{D}_{si}\}_{i=1}^{S}$. DG requires learning a model f from these S domains such that the prediction error of f on the target domain is minimized. The classification error formula for the target domain is shown as follows:

$$\min{}_f \frac{1}{N_t} \sum_{i=1}^{N_t} l(f(x_i), y_i) \tag{1}$$

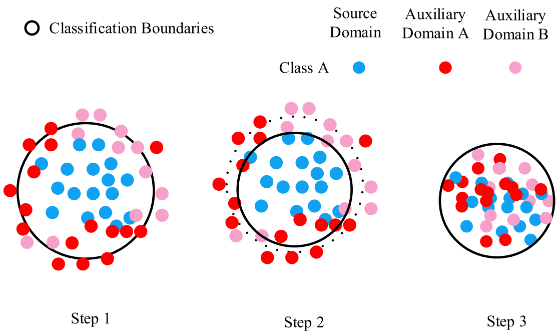


Fig. 2. Idea of the method. Note: In Step 1, features of the same type are clustered near the classification boundary. In Step 2, the classification difficulty is increased so that similar samples in the auxiliary domain are not correctly classified. In Step 3, the samples of the training auxiliary domain are adapted to the classification requirement.

where N_t represents the number of samples in the target domain, and $l(\cdot, \cdot)$ denotes the specific loss function. x_i and y_i are the ith sample and label, respectively.

B. Generative Adversarial Network (GAN)

Generative adversarial network (GAN) [28] consists of two main parts: a generator and a discriminator. The generator is responsible for generating fake samples. The discriminator needs to determine whether the samples are real or generated by the generator. The objective function of GAN is as follows:

$$\begin{aligned}&\min_G \max_D V(D, G)\\&\quad = E_{x\sim p_{\text{data}}}\big[\log D(x)\big] + E_{z\sim p_{\text{noise}}}\big[\log(1 - D(G(z)))\big] \qquad (2)\end{aligned}$$

where G is the generator, D is the discriminator, p_{data} is the true data distribution, and the noise distribution is denoted by p_{noise}. x is the true data, and z is the noise data.

Recently, researchers have introduced GAN into transfer learning, and the ADGN method also borrows ideas from GAN.

III. Adversarial Domain Generalization Network Model

In this section, ADGN is described in detail. The core idea of the method is introduced in Section III-A. The components of the model are further described, along with how to train it, in Section III-B.

A. Overall Idea

In ADGN, a fully labeled source domain $\mathcal{D}_{\text{LS}} = \{(x_{\text{LS},j}, y_{\text{LS},j})\}_{j=1}^{M}$ and N fully unlabeled auxiliary domains $\{\mathcal{D}_{USn}\}_{n=1}^{N} = \{\{x_{USn,j}\}_{j=1}^{M}\}_{n=1}^{N}$ are accessed. $\mathcal{D}_{USn}$ denotes the nth unlabeled domain. $x_{\text{LS},j}$ and $y_{\text{LS},j}$ are the jth sample and label in domain $\mathcal{D}_{\text{LS}}$, respectively. $x_{USn,j}$ denotes the jth sample in domain $\mathcal{D}_{USn}$. M is the number of samples contained in each domain.

This model has a feature extractor G and two independent label classifiers $C1$ and $C2$. First, $x_{\text{LS},j}$ and $x_{USn,j}$ are used as the input of the feature generator G, and high-dimensional features are output through the feature generator G. Subsequently, the output of the feature generator is used as the input of classifiers $C1$ and $C2$. Finally, classification probabilities $\{q_{1,n}(\boldsymbol{y} \mid \boldsymbol{x}_{USn,j})\}_{n=1}^{N}$ and $\{q_{2,n}(\boldsymbol{y} \mid \boldsymbol{x}_{USn,j})\}_{n=1}^{N}$ are obtained. Obviously, the above steps do not obtain domain-invariant feature. Referring to the idea of GANs, classifiers are used as discriminators for training generators to generate domain-invariant features. As shown in Fig. 2, in the feature mapping space, we refer to the intersection of two classifiers as the classification boundary of the model. For example, if a feature is within the classification boundary of class A, then the feature is classified as class A by both the classifiers. The model requires the same classification result for the two classifiers. We refer to this requirement as the classification requirement. In addition, we refer to the features that are made different decisions by the two classifiers as fuzzy features. In step 1, the source domain is trained and its classification boundaries are obtained. Given that the source and auxiliary domains' samples are similar, some of the auxiliary domain features can satisfy the classification requirement. When increasing the classification difference between the two classifiers for the auxiliary domain, the classification boundary is converged to the mapped feature space of the source domain. In other words, only the source domain features can satisfy the classification requirement, and most of the auxiliary domains' features cannot satisfy. Moreover, the number of fuzzy features also increases. Therefore, if the decision differences among the two classifiers for fuzzy features need to be reduced, then the feature mapping function of the model should be optimized so the fuzzy features are more similar to the source domain features, which can always satisfy the classification requirements. During iterations, the decision differences between the two classifiers for the auxiliary domain are smaller and the domain-invariant features are obtained. $d_n(p_1, p_2)$ denotes the decision divergence of the two classifiers on the nth auxiliary domain $\mathcal{D}_{USn}$. p_1 and p_2 represent the predicted output of the two classifiers, respectively.

For the model to gain generalization capabilities, the following work is done. First, a large number of fuzzy features can be obtained by increasing the number of auxiliary domains' samples. Next, the discriminator is trained, which maximizes the decision-making differences. If the decision-making differences in the two classifiers are not controlled artificially, the two classifiers will be very similar and will be unable to detect fuzzy features effectively. Finally, the generator is trained to play against the discriminator, and the generator aims to reduce the decision-making differences in the discriminator. Through adversarial learning, a feature generator that minimizes intraclass feature differences is obtained.

B. Training Steps

Step 1: Classifiers and generators are trained to correctly classify samples of source domain $\mathcal{D}_{\text{LS}}$. This step is crucial to gather sample features of the same class near the classification boundary, which provides a prerequisite for detecting fuzzy features in Step 2. In addition, trained classifiers and generators must first work on source domain $\mathcal{D}_{\text{LS}}$ to generalize better to other domains. The objectives of the optimization are as follows:

$$\min_{G,C_1,C_2} \mathcal{L}(X_{\text{LS}}, Y_{\text{LS}}) \qquad (3)$$

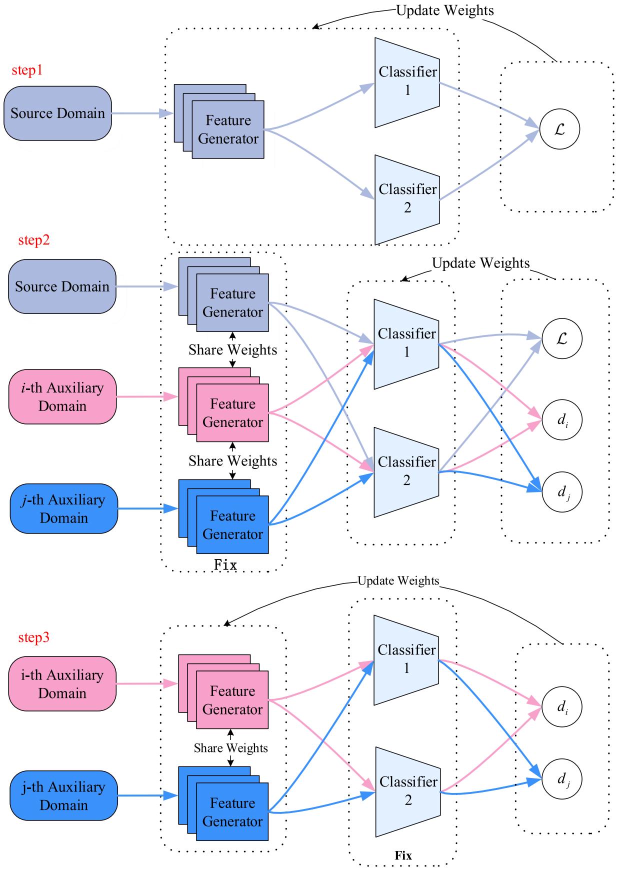


Fig. 3. Flowchart of ADGN. The arrows of a specific color represent the input or output of a specific domain. d_i and d_j denote the classifier differences in the ith and jth auxiliary domains, respectively. $\mathcal{L}$ is the cross-entropy loss.

$$\mathcal{L}(X_{\text{LS}}, Y_{\text{LS}}) = -\frac{1}{M}\sum_{i=1}^{M}\sum_{k=1}^{K} I\left[y_i^{ls} = k\right]\log\frac{\exp\left(x_{i,[k]}^{ls}\right)}{\sum_{j=1}^{K}\exp\left(\boldsymbol{x}_{i,[k]}^{ls}\right)} \tag{4}$$

where $I[]$ is an indicator function that returns a value indicating the probability that the case is true. x_i^{ls} represents the ith sample of the source domains. y_i^{ls} represents the ith label of the source domains. K is the number of classes owned. k is the kth class

Step 2: As shown in Fig. 3, the two classifiers are initialized differently to obtain different classifiers from the beginning. The fuzzy features of the class boundaries are then detected to a greater extent by further increasing the difference between the two classifiers. In addition, more class boundary fuzzy features are obtained when the involved domains are sufficient, and the generalization is stronger. In particular, the feature generator parameters θ_G are fixed in this step, and only the classifier parameters θ_{C1} and θ_{C2} are updated. The purpose is to train the generator and discriminator independently to avoid the mutual influence of parameter updates. To prevent the two classifiers from losing their classification power, we add the source domain classification loss to the objective. The overall objectives are as follows:

$$\min_{C_1,C_2}\mathcal{L}(X_{\text{LS}}, Y_{\text{LS}}) - \mathcal{L}_{\text{adv}}(X_{US1}, \ldots, X_{USN}) \tag{5}$$

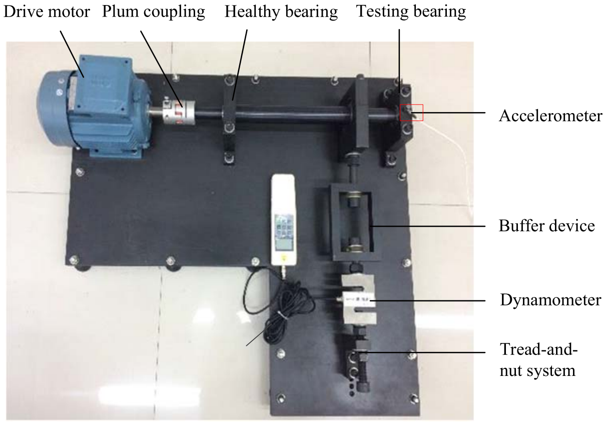


Fig. 4. Test rig of SCU.

TABLE I
DOMAIN SETUP

Name	Speed	Load	Samples
A	800r/min	0.8KN	1400
B	800r/min	1.6KN	1400
C	800r/min	2.5KN	1400
D	1000r/min	1.6KN	1400

$$\mathcal{L}_{\text{adv}}(X_{US1}, \ldots, X_{USN}) = \mathbb{E}_{x_{USn}\sim X_{USn}}\left[\sum_{n=1}^{N} d_n(p_1, p_2)\right]. \tag{6}$$

$d_n(\cdot,\cdot)$ denotes the decision divergence of the two classifiers on the nth auxiliary domain $\mathcal{D}_{USn}$

$$d_n(p_1, p_2) = \frac{1}{K}\sum_{k=1}^{K}\left|p_{1k}(\mathbf{y} \mid \mathbf{x}_{usn}) - p_{2k}(\mathbf{y} \mid \mathbf{x}_{usn})\right|. \tag{7}$$

p_{1k} and p_{2k} represent the probability that $C1$ and $C2$ will predict the features as class k, respectively. The specific calculation formula is as follows:

$$p_{1k}(\mathbf{y} \mid \mathbf{x}_{usn}) = \frac{e^{z_k}}{\sum_{k=1}^{K} e^{z_k}}. \tag{8}$$

Here, z_k denotes the output value of the kth node of the neural network.

Step 3: As shown in Fig. 3, the classifier parameters θ_{C1} and θ_{C2} are fixed, and the feature generator parameters θ_G are updated. A large number of class boundary fuzzy features are detected by Step 2. In Step 3, a feature generator is trained, which captures and amplifies the class features embedded in the fuzzy features and forms a confrontation with the classifier so that the classifier cannot easily detect the fuzzy features. The objectives are as follows:

$$\min_G \mathcal{L}_{\text{adv}}(X_{US1}, \ldots, X_{USN}). \tag{9}$$

Training for these three steps is cycled, and the DG model with strong generalizability is finally obtained by the mutual adversarial of classifiers and generators.

IV. Experimental Analysis

A. Case 1: Soochow University (SCU) Dataset

1) Dataset Description: The data used in this case are obtained from the testbed of Soochow University. As shown

TABLE II

DATASET SETUP

health conditions	label	Description	samples
Ball Fault	0	BF	200
Inner race ball fault	1	IBF	200
Inner race Fault	2	IF	200
Inner and outer race Fault	3	IOF	200
Inner race outer race ball Fault	4	IOBF	200
outer race ball Fault	5	OBF	200
outer race Fault	6	OF	200

TABLE III

MODEL FRAMEWORK OF FEATURE GENERATORS

Layer name	Output size	Channels × kernel size
Input	1×32×32	\
Conv1	64×32×32	64×5×5，stride=1
BN、Relu	64×32×32	\
Max Pool1	64×16×16	64×3×3，stride=2
Conv2	64×16×16	64×5×5，stride=1
BN、Relu	64×16×16	\
Max Pool 2	64×8×8	64×3×3，stride=2
Conv3	128×8×8	128×5×5，stride=1
BN、Relu	128×8×8	\
One-dimensional	1×8192	3072
Fully Connected Layer 1	1×3072	

TABLE IV

PARAMETER SETTING OF CLASSIFIER

Layer name	Activation function	Parameter Structure
Fully Connected Layer 2	Relu	1×3072
BN	\	2048×2048
Fully Connected Layer 3	Softmax	2048×K

TABLE V

PARAMETER SETTING OF DOMAIN DISCRIMINATOR

Layer name	Activation function	Parameter Structure
Fully Connected Layer 4	Relu	3072×1024
Fully Connected Layer 5	Relu	1024×1024
Fully Connected Layer 6	Softmax	1024×(N+1)

TABLE VI

EXPERIMENTAL RESULTS (SCU) (ONLY ONE FULLY LABELED DOMAIN IS USED IN THE TRAINING PROCESS)

Task	M2	M3	M4	**ADGN**
ABC→D	83.57%	92.79%	88%	**98%**
ABD→C	78.93%	**95.57%**	89.21%	95.36%
ACD→B	91.93%	96.71%	93.93%	**98.43%**
BAC→D	81.29%	90.07%	82.29%	**97.57%**
BAD→C	90.86%	94.71%	94.93%	**98.43%**
BCD→A	92.93%	95.85%	97.14%	**98.79%**
CAB→D	77.5%	83.28%	77.21%	**98.36%**
CAD→B	89.14%	**97.14%**	88.64%	96.71%
CBD→A	89.57%	95.43%	87.93%	**98.14%**
DAB→C	86.86%	95.71%	84.14%	**97.79%**
DAC→B	80.14%	97.79%	85.5%	**100%**
DBC→A	86.43%	98.29%	84.21%	**99.21%**
Accuracy fluctuations	77.5%~ 92.93%	83.28%~ 98.29%	77.21%~ 97.14%	95.36%~ 100%

TABLE VII

EXPERIMENTAL RESULTS (SCU) (MULTIPLE FULLY LABELED DOMAINS ARE USED IN THE TRAINING PROCESS)

Task	M1	M2	M3	M4
ABC→D	**96.5%**	85.79%	92.5%	91.14%
ABD→C	95.5%	94.14%	96.5%	**96.86%**
ACD→B	83.64%	97.29%	97.86%	**97.93%**
BCD→A	94.57%	94.93%	98.07%	**97.93%**

in Fig. 4, the test bench consists of couplings, bearings, and dynamometers. Bearings with different health conditions are replaced for data acquisition. The bearing type is 6205-2RS SKF. The nut is adjusted to obtain data for different loads. The load size is measured by SGSF-20K dynamometer. The speed can be adjusted from 800 to 1200 r/min by motor speed control. Faults are set on the outer race, inner race, and ball of the train bearing using the wire cutting method. The groove width is approximately 0.2 mm, and the depth is around 6 mm. It is used to simulate the bearing outer race fault (OF), inner race fault (IF), ball fault (BF), outer race ball fault (OBF), inner race ball fault (IBF), inner race outer race fault (IOF), and inner race outer race ball fault (IOBF).

The working environment settings for the dataset are provided in Table I. Each dataset contains seven types of fault samples, 1400 in total (i.e., 200 samples for each fault). The sample settings for one of the datasets are presented in Table II.

2) Comparison Method: In this section, ADGN is compared with other DG models. To be fair, all the models use the same feature generators and classifiers and domain settings. The comparison method is set as follows.

M1: Combine data from all the domains in one place for training to learn a model trained on all data.

M2: Analogous to DANN [29] in DA, domain-invariant features are obtained using adversarial training of generators and discriminators. As more domains are involved in the training compared with DA, the domain-invariant features obtained will be more generalizable.

M3: DG methods using MMD guidance.

M4: DG methods using CORAL guidance.

3) Network Settings: Tables III–V list the model framework of the feature generator, the model parameters of the classifier, and the model parameters of the domain discriminator required for M2, respectively.

4) Analysis of Results: Upon completion of model training, the best results of this model are compared with those of other methods after convergence. The diagnostic results are summarized in Table VI. Our proposed method is significantly better than other methods and has higher robustness. It can maintain reliable diagnostic accuracy for different generalization tasks.

We also perform a study for the comparison method. The transfer tasks listed in Table VI can access only one source domain with labels. We configure multiple source domains with labels for M1, M2, M3, and M4 based on the original settings. Table VII presents the diagnosis results.

According to the experimental results, we find interesting phenomena. The generalization task is denoted by the symbol **A**BC → E, and A in bold indicates that domain A is the source domain, that is, both the sample and the label can be accessed.

TABLE VIII

DATASET SETUP (SDUST)

Health state	Fault diameter/mm	label	sample	Description
Normal	-	0	200	NO
Inner race	0.2	1	200	IF2
Inner race	0.4	2	200	IF4
Inner race	0.6	3	200	IF6
Outer race	0.2	4	200	OF2
Outer race	0.4	5	200	OF4
Outer race	0.6	6	200	OF5
Ball	0.2	7	200	BF2
Ball	0.4	8	200	BF4
Ball	0.6	9	200	BF6

TABLE IX

DATASET SETUP (SDUST)

Name	Speed	Load	Samples
A	2000r/min	40N	2000
B	2000r/min	60N	2000
C	2500r/min	40N	2000
D	2500r/min	60N	2000
E	3000r/min	40N	2000

TABLE X

EXPERIMENTAL RESULTS (SDUST) (ONLY ONE FULLY LABELED DOMAIN IS USED IN THE TRAINING PROCESS)

Task	M2	M3	M4	**ADGN**
ABCD→E	91.7%	94.75%	92.8%	**97.15%**
ABCE→D	99.45%	99.9%	99.7%	**100%**
ABDE→C	99.85%	99.6%	99.4%	**100%**
ACDE→B	**100%**	99.25%	99.35%	99.85%
BACE→D	97.1%	**99.9%**	**99.9%**	98.5%
BACD→E	96.65%	99.94%	**100%**	**100%**
BADE→C	99.7%	99.55%	94.2%	**100%**
BCDE→A	**100%**	97.55%	99.75%	**100%**
CABD→E	91.7%	99.9%	**99.95%**	97.8%
CABE→D	95.95%	99.95%	**100%**	**100%**
CADE→B	**99.65%**	95.3%	92.8%	99.45%
CBDE→A	**99.65%**	85.15%	86.7%	92.75%
DABC→E	100%	100%	100%	**100%**
DABE→C	81.2%	100%	99.95%	**100%**
DACE→B	74.1%	96.35%	96.1%	**99.9%**
DBCE→A	84.1%	83.1%	87.75%	**98.55%**
EABC→D	83.25%	**100%**	**100%**	**100%**
EABD→C	90.15%	**100%**	99.7%	**100%**
EACD→B	81.75%	94.8%	94.75%	**99.25%**
EBCD→A	73.85%	86.8%	85.25%	**96%**
Accuracy fluctuations	73.85%~ 100%	83.1%~ 100%	85.25%~ 100%	92.75%~ 100%

TABLE XI

EXPERIMENTAL RESULTS (SDUST) (MULTIPLE FULLY LABELED DOMAINS ARE USED IN THE TRAINING PROCESS)

Task	M1	M2	M3	M4
ABCD→E	**100%**	**100%**	99.65%	99.15%
ABCE→D	**100%**	95.95%	99.95%	99.9%
ABDE→C	99.5%	**99.7%**	99.55%	99.35%
ACDE→B	98.8%	**99.65%**	95.3%	94.2%
BCDE→A	89.5%	99.65%	97.55%	**99.75%**

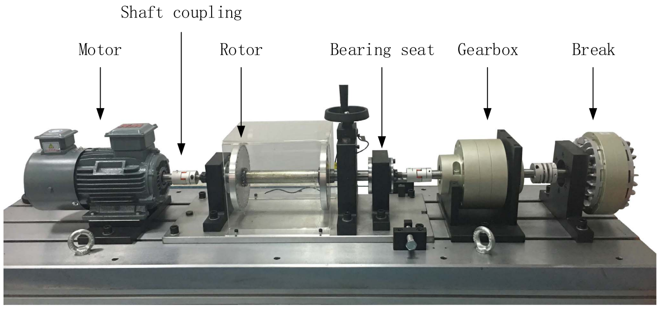


Fig. 5. Test rig of SDUST.

B and C, which are not in bold, are the two auxiliary domains, that is, samples are accessible and labels are not. E is the unknown domain, which is used to verify the generalization performance of the model. In this case, 12 transfer tasks are set up to demonstrate the superiority of the proposed model by fully considering the tasks of different situations.

1) By detecting a large number of fuzzy features and training them, ADGN achieves reliable diagnosis under unknown operating environments. ADGN realizes cross-domain diagnosis by eliminating fuzzy features. An important aspect of the improvement achieved by ADGN is that the present method considers the effect of category boundaries rather than the traditional adaptation of edge probability distributions.
2) ADGN obtains comparable or even higher diagnostic accuracy using only one domain of sample labels for supervision than using three domains of sample labels for simultaneous supervision. ADGN achieves high diagnostic accuracy, thanks to ADGN's unique class boundary feature detection module, which transforms fuzzy features at classification boundaries into robust features. The accuracies in Table VI are all trained using sample labels from only one domain. The accuracies in Table VII are all trained by accessing labels from three domains. Looking at ADGN in Table VI and the other methods in Table VII, ADGN has better diagnostic accuracy and more robustness.
3) ADGN seeks generalized features from classification boundaries and shows higher performance in different transfer tasks. Using the data in Table VI analyzing M1, M2, M3, and M4 independently, these models perform poorly on many transfer tasks. None of these models is well-robust, and they do not respond well to the changes in objective conditions. On the contrary, ADGN presents excellent diagnostic performance with full accuracy in various transfer tasks, which is crucial for industrial production.
4) In this experiment, the diagnostic performance of MMD is slightly better than that of CORAL. This outcome can be proven by comparing the experimental results of M3 and M4. Therefore, a better adaptation can be found by changing different distance metrics.
5) The comparison method exhibits higher diagnostic accuracy and robustness when trained with sample labels from multiple domains than when trained with sample

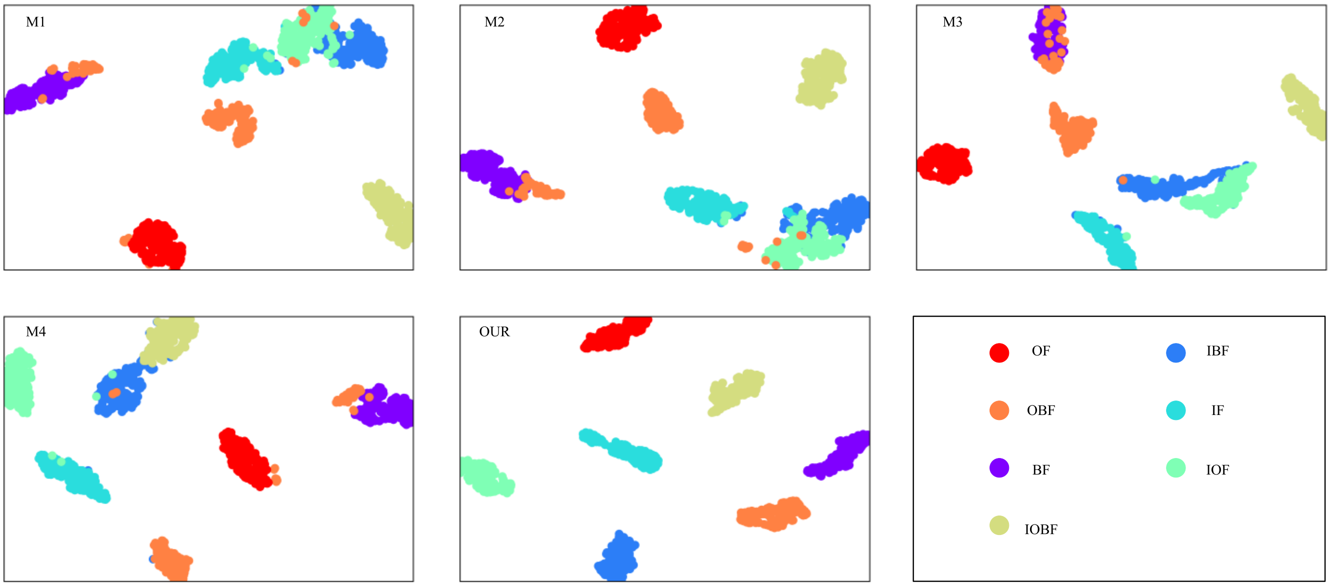


Fig. 6. Classification effect on SCU.

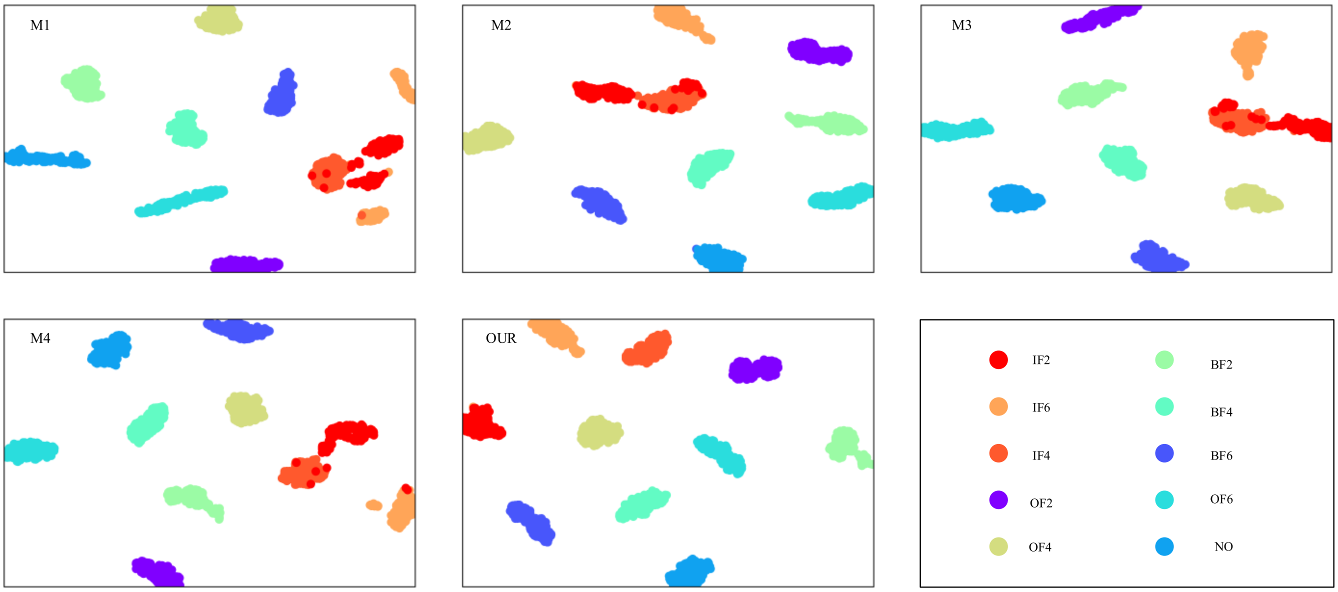
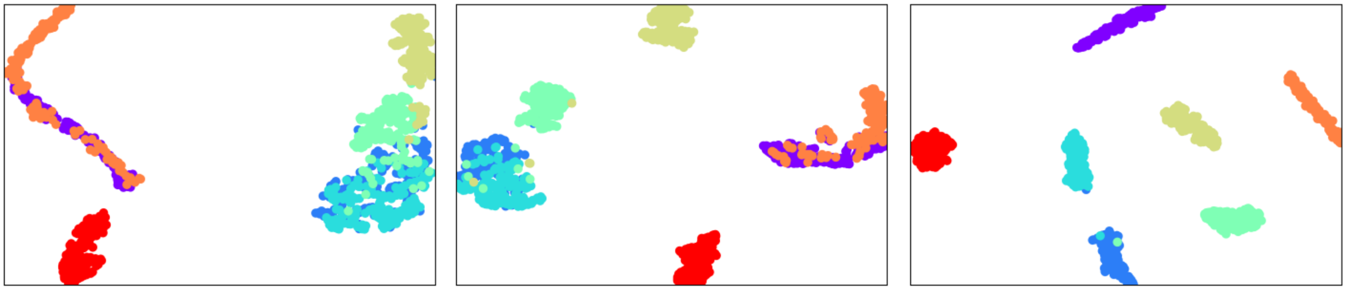


Fig. 7. Classification effect on SDUST.



Early training:Confusing classification

Mid-Training:Gradual reduction of classification error

End of training:Clustering effect is obvious

Fig. 8. Visualization of training effects on SCU.

labels from only one domain. M2, M3, and M4 seek generalization features from the perspective of marginal distribution adaptation, so the robustness of M2, M3, and M4 is poor when only one source domain is involved in the training. Comparing similar methods in Tables VI and VII, they show the same phenomenon, that is, multiple source domains involved in the training outperform the single source domain involved in the training. A model trained on only one source domain will have a classifier that is more inclined to the source domain and is prone to errors in classifying features from other domains. ADGN considers from the classification boundary and effectively collects the class edge features to avoid overfitting of the classifier.

B. Case 2: Shandong University of Science and Technology (SDUST) Dataset

1) Dataset Description: To further verify the effectiveness and robustness of ADGN, experiments are carried out using laboratory data from the Shandong University of Science and Technology (SDUST). The experimental equipment is shown in Fig. 5. The bench consists of a gearbox, a motor, three couplings, two rotors, two bearing blocks, and a brake. The bearing type is N205EU. The test bearing is set up with four basic health conditions (normal, IF, BF, and OF) and three failure sizes (0.2, 0.4, and 0.6 mm). Therefore, ten health states are obtained, and they are listed in Table VIII.

As in Case 1, the five operating environments are set according to the speed and load. The detailed settings are listed in Table IX.

2) Analysis of Results: We continue to use the transfer task and network from Case 1. The experimental results are given in Tables X and XI, and after analysis, similar conclusions as in Case 1 are obtained. In Case 2, the task is less difficult because of the smaller span between loads. Thus, the accuracy of each method is improved, relative to Case 1. Nevertheless, ADGN still performs the best.

3) Visualization of Learned Representation: To understand the classification effect of the model more intuitively, the features extracted by the classifier are transformed into a 2-D feature representation by T-SNE [30], and the classification effect plots of Cases 1 and 2 are shown in Figs. 6 and 7, respectively. The figures only show the classification effect under unknown operating environment, and they demonstrate that the classification effect of ADGN is significantly better than that of other methods. Focusing on the T-SNE plots of M1, M2, M3, and M4, these methods have different degrees of crossover in the classification boundaries, producing many misclassifications. On the contrary, the classification boundary of ADGN is obvious, and the clustering effect is better.

Fig. 8 shows the dynamic process of the model's clustering effect gradually becoming better, from left to right, showing the classification effect of the model at the early stage of training, at the middle stage of training, and after the training is completed. In particular, the classification ability of the model at the early stage of training is obviously insufficient, and no obvious classification boundary is formed. In the middle of the training period, the model has the classification ability for individual categories, and it can achieve reliable classification for a few categories. However, some categories of samples cannot be classified correctly. After repeated training, the model finally has good classification effect with clear classification boundary and reliable classification.

V. Conclusion

This study focuses on bearing fault detection in unknown operating environments and achieves better results using only one source domain with labels for DG studies. The performance of models trained by the general method using only one labeled source domain is significantly worse. The relationship between classification boundaries and domain-invariant features is explored. The fuzzy features detected by the classifier are trained to be robust features with distinct class features through adversarial games between the generator and the classifier. Finally, a fault diagnosis framework for unknown operating environments is successfully developed. ADGN is proven to be a very effective diagnostic tool for fault diagnosis. Achieving such superior results using only one source domain is rare. Through the validation of the two datasets, ADGN can also cope well with complex working conditions with different speeds and complexities. In addition, ADGN provides new ideas for future work to explore domain-invariant features from the classification boundary.

References

[1] Z. Zhao *et al.*, "Applications of unsupervised deep transfer learning to intelligent fault diagnosis: A survey and comparative study," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–28, 2021.

[2] K. Kaplan, Y. Kaya, M. Kuncan, M. R. Minaz, and H. M. Ertunç, "An improved feature extraction method using texture analysis with LBP for bearing fault diagnosis," *Appl. Soft Comput.*, vol. 87, Feb. 2020, Art. no. 106019.

[3] Y. Liao, R. Huang, J. Li, Z. Chen, and W. Li, "Dynamic distribution adaptation based transfer network for cross domain bearing fault diagnosis," *Chin. J. Mech. Eng.*, vol. 34, no. 1, pp. 1–10, Dec. 2021.

[4] C. Zhang, J. Feng, C. Hu, Z. Liu, L. Cheng, and Y. Zhou, "An intelligent fault diagnosis method of rolling bearing under variable working loads using 1-D stacked dilated convolutional neural network," *IEEE Access*, vol. 8, pp. 63027–63042, 2020.

[5] J. Wang, Z. Mo, H. Zhang, and Q. Miao, "A deep learning method for bearing fault diagnosis based on time-frequency image," *IEEE Access*, vol. 7, pp. 42373–42383, 2019.

[6] Y. Lei, B. Yang, X. Jiang, F. Jia, N. Li, and A. K. Nandi, "Applications of machine learning to machine fault diagnosis: A review and roadmap," *Mech. Syst. Signal Process.*, vol. 138, Apr. 2020, Art. no. 106587.

[7] A. Dibaj, M. M. Ettefagh, R. Hassannejad, and M. B. Ehghaghi, "A hybrid fine-tuned VMD and CNN scheme for untrained compound fault diagnosis of rotating machinery with unequal-severity faults," *Expert Syst. Appl.*, vol. 167, Apr. 2021, Art. no. 114094.

[8] E. Bhuiyan, Y. Sarker, S. Fahim, M. A. Mannan, S. Sarker, and S. Das, "A reliable open-switch fault diagnosis strategy for grid-tied photovoltaic inverter topology," in *Proc. Int. Conf. Autom., Control Mechatronics Ind. (ACMI)*, Jul. 2021, pp. 1–4.

[9] T. C. Koylu, C. R. W. Reinbrecht, S. Hamdioui, and M. Taouil, "RNN-based detection of fault attacks on RSA," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, Oct. 2020, pp. 1–5.

[10] X. Kong, X. Li, Q. Zhou, Z. Hu, and C. Shi, "Attention recurrent autoencoder hybrid model for early fault diagnosis of rotating machinery," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–10, 2021.

[11] Z. Zhao *et al.*, "Deep learning algorithms for rotating machinery intelligent diagnosis: An open source benchmark study," *ISA Trans.*, vol. 107, pp. 224–255, Dec. 2020.

[12] H. Wang, J. Xu, R. Yan, C. Sun, and X. Chen, "Intelligent bearing fault diagnosis using multi-head attention-based CNN," *Proc. Manuf.*, vol. 49, pp. 112–118, Jan. 2020.

[13] H. Shao, H. Jiang, Y. Lin, and X. Li, "A novel method for intelligent fault diagnosis of rolling bearings using ensemble deep auto-encoders," *Mech. Syst. Signal Process.*, vol. 102, pp. 278–297, Mar. 2018.

[14] Z. An, S. Li, J. Wang, and X. Jiang, "A novel bearing intelligent fault diagnosis framework under time-varying working conditions using recurrent neural network," *ISA Trans.*, vol. 100, pp. 155–170, May 2020.

[15] H. Pan, W. Tang, J.-J. Xu, and M. Binama, “Rolling bearing fault diagnosis based on stacked autoencoder network with dynamic learning rate,” *Adv. Mater. Sci. Eng.*, vol. 2020, pp. 1–12, Dec. 2020.

[16] S. Liu, J. Xie, C. Shen, X. Shang, D. Wang, and Z. Zhu, “Bearing fault diagnosis based on improved convolutional deep belief network,” *Appl. Sci.*, vol. 10, no. 18, p. 6359, Sep. 2020.

[17] M. Kim, J. U. Ko, J. Lee, B. D. Youn, J. H. Jung, and K. H. Sun, “A domain adaptation with semantic clustering (DASC) method for fault diagnosis of rotating machinery,” *ISA Trans.*, vol. 120, pp. 372–382, Jan. 2022.

[18] S. Siahpour, X. Li, and J. Lee, “Deep learning-based cross-sensor domain adaptation for fault diagnosis of electro-mechanical actuators,” *Int. J. Dyn. Control*, vol. 8, no. 4, pp. 1054–1062, Dec. 2020.

[19] Y. Qin, Q. Yao, Y. Wang, and Y. Mao, “Parameter sharing adversarial domain adaptation networks for fault transfer diagnosis of planetary gearboxes,” *Mech. Syst. Signal Process.*, vol. 160, Nov. 2021, Art. no. 107936.

[20] Z.-H. Liu, B.-L. Lu, H.-L. Wei, X.-H. Li, and L. Chen, “Fault diagnosis for electromechanical drivetrains using a joint distribution optimal deep domain adaptation approach,” *IEEE Sensors J.*, vol. 19, no. 24, pp. 12261–12270, Dec. 2019.

[21] M. Ghorvei, M. Kavianpour, M. T. Beheshti, and A. Ramezani, “An unsupervised bearing fault diagnosis based on deep subdomain adaptation under noise and variable load condition,” *Meas. Sci. Technol.*, vol. 33, no. 2, Feb. 2022, Art. no. 025901.

[22] J. Jiao, M. Zhao, J. Lin, K. Liang, and C. Ding, “A mixed adversarial adaptation network for intelligent fault diagnosis,” *J. Intell. Manuf.*, pp. 1–16, 2021, doi: 10.1007/s10845-021-01777-0

[23] J. Wang *et al.*, “Generalizing to unseen domains: A survey on domain generalization,” 2021, *arXiv:2103.03097*.

[24] X. Li, W. Zhang, H. Ma, Z. Luo, and X. Li, “Domain generalization in rotating machinery fault diagnostics using deep neural networks,” *Neurocomputing*, vol. 403, pp. 409–420, Aug. 2020.

[25] Y. Yang, J. Yin, H. Zheng, Y. Li, M. Xu, and Y. Chen, “Learn generalization feature via convolutional neural network: A fault diagnosis scheme toward unseen operating conditions,” *IEEE Access*, vol. 8, pp. 91103–91115, 2020.

[26] T. Han, Y.-F. Li, and M. Qian, “A hybrid generalization network for intelligent fault diagnosis of rotating machinery under unseen working conditions,” *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–11, 2021.

[27] G. Blanchard, G. Lee, and C. Scott, “Generalizing from several related classification tasks to a new unlabeled sample,” in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 24, 2011, pp. 2178–2186.

[28] I. Goodfellow *et al.*, “Generative adversarial nets,” in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 27, 2014, pp. 1–9.

[29] Y. Ganin *et al.*, “Domain-adversarial training of neural networks,” *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 2030–2096, 2017.

[30] P. E. Rauber, A. X. Falcao and A. C. Telea, "Visualizing time-dependent data using dynamic t-SNE", *Proc. EuroVis*, vol. 2, no. 5, pp. 73–77, 2016.

Jingde Li received the B.S. degree in industrial engineering from the Hubei University of Automotive Technology, Shiyan, China, in 2020. He is currently pursuing the M.S. degree in traffic and transportation with Soochow University, Suzhou, China.

His research interests include deep domain adaptation for mechanical fault diagnosis, adversarial learning, and transfer learning.

Changqing Shen (Senior Member, IEEE) received the B.S. and Ph.D. degrees in instrument science and technology from the University of Science and Technology of China, Hefei, China, in 2009 and 2014, respectively, and the Ph.D. degree in systems engineering and engineering management from the City University of Hong Kong, Hong Kong, in 2014.

He is currently a Professor with the School of Rail Transportation, Soochow University, Suzhou, China. His research interests include signal processing and machine-learning-based fault diagnosis.

Lin Kong received the B.S. degree in instrument science and technology from the University of Science and Technology of China, Hefei, China, in 2009, and the Ph.D. degree in optical engineering from the University of Chinese Academy of Sciences, Beijing, China, in 2014.

He is currently a Research and Development Scientist with Chang Guang Satellite Technology Company Ltd., Changchun, China. His research interests include satellite system design and satellite thermal control.

Dong Wang (Member, IEEE) received the Ph.D. degree from the City University of Hong Kong, Hong Kong, in 2015.

He was a Senior Research Assistant, a Post-Doctoral Fellow, and a Research Fellow with the City University of Hong Kong. He is currently an Associate Professor with the Department of Industrial Engineering and Management and the State Key Laboratory of Mechanical System and Vibration, Shanghai Jiao Tong University, Shanghai, China. His research interests include sparse and complex measures, signal processing, prognostics and health management, condition monitoring and fault diagnosis, statistical learning and machine learning, statistical process control, and non-destructive testing.

Dr. Wang is also an Associate Editor of IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENT, IEEE SENSORS JOURNAL, *Journal of Low Frequency Noise*, *Vibration and Active Control*, and *Journal of Dynamics Monitoring and Diagnostics*; and an Editorial Board Member of *Mechanical Systems and Signal Processing*.

Min Xia (Senior Member, IEEE) received the B.S. degree from Southeast University, Nanjing, China, in 2009, the M.S. degree from the University of Science and Technology of China, Hefei, China, in 2012, and the Ph.D. degree from The University of British Columbia, Vancouver, BC, Canada, in 2017.

He is currently a Lecturer with the Department of Engineering, Lancaster University, Lancaster, U.K. His research interests include smart manufacturing, machine diagnostics and prognostics, deep neural networks, wireless sensor networks, and sensor fusion.

Zhongkui Zhu received the B.S. degree in automobile and tractor (automobile) and the M.S. degree in vehicle engineering from Hefei Polytechnic University, Hefei, China, in 1997 and 2002, respectively, and the Ph.D. degree in instrument science and technology from the University of Science and Technology of China, Hefei, in 2005.

He is currently a Professor with the School of Rail Transportation, Soochow University, Suzhou, China. His current research interests include fault diagnosis of mechanical equipment, vehicle system dynamics and control, and vibration measurement and signal processing.