Adversarial domain-invariant generalization: a generic domain-regressive framework for bearing fault diagnosis under unseen conditions

Liang Chen, Qi Li, Changqing Shen, *Senior Member, IEEE,* Jun Zhu, Dong Wang, *Member, IEEE,* Min Xia

***Abstract*—Recently, various fault diagnosis methods based on domain adaptation (DA) have been explored to solve the problem of discrepancy between the source and target domains. However, given complex industrial scenarios, DA-based methods usually fail when the working conditions of machines are unseen, i.e., target data are unavailable during model training. In this work, a generic domain-regressive framework for fault diagnosis, namely, adversarial domain-invariant generalization (ADIG), is proposed. ADIG leverages multiple available domain data to exploit domain-invariant knowledge through adversarial learning between the feature extractor and the domain classifier. Simultaneously, the fault classifier generalizes the knowledge from the source-related domain to diagnose the unseen but related target domain signals. Moreover, customized strategies of feature normalization and adaptive weight are proposed to promote diagnosis performance. Comprehensive case studies show that ADIG achieves satisfactory diagnosis accuracy and robustness under unseen conditions, indicating that ADIG is a remarkably potential diagnosis tool for real-case industrial machines.**

***Index Terms*—Cross-domain fault diagnosis, Domain generalization, Adversarial learning, Rotating machinery.**

Manuscript received December 3, 2020; revised February 24, 2021; revised April 14, 2021; accepted May 6, 2020. Date of publication XXX, XXXX; date of current version May 7, 2021.This work was supported in part by the National Nature Science Foundation of China under Grants 51875375 and 51975355, and in part by the Open Foundation of the State Key Laboratory of Fluid Power and Mechatronic Systems under grant GZKF-202022. Paper no. TII-21-1673. (Corresponding author: Changqing Shen.)

L. Chen, Q. Li are with School of Mechanical and Electric Engineering, Soochow University, Suzhou, China(e-mail: ChenL@suda.edu.cn, 20194229023@stu.suda.edu.cn)

C. Shen is with the School of Mechanical and Electric Engineering, Soochow University, Suzhou, China, and also with the State Key Laboratory of Fluid Power and Mechatronic Systems, Zhejiang University, Hangzhou 310027, China(e-mail: cqshen@suda.edu.cn).

J. Zhu is with the Department of Civil Aviation, Northwestern Polytechnical University, China.(e-mail: j.zhu@u.nus.edu)

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

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

I. INTRODUCTION

Modern industries are moving toward informatization and intelligentization in the fourth industrial revolution era [1], and modern machinery and equipment are widely used in various fields, such as construction, aviation, electric power, and metallurgy. Given the inevitable faults of mechanical devices, health management has been studied for economic benefit and personnel security [2]. In recent years, researchers have studied several techniques, such as signal processing [3], to reveal fault information. Intelligent techniques, including machine learning [4] or deep learning [5], have also been introduced to achieve satisfactory health monitoring systems.

As a major and crucial mechanical component [6], bearings often operate under variable working conditions. Therefore, cross-domain fault diagnosis [7] was proposed to transfer knowledge from the source domain to the target domain. In industrial scenarios, however, independent and identically distributed assumption may often be violated. In other words, training data of source domain with sufficient labels have similar but different distributions from the unlabeled testing data of the target domain. The discrepancy between the source and target domains causes a domain shift through the different working conditions, where the performance of the diagnostic model degenerates when the model is trained by the source domain but is used as an inference engine in the target domain.

Researchers therefore attempted to use domain adaptation (DA) to solve the problem. DA can be regarded as a special case of transfer learning, which aims to transfer shared knowledge across different but related domains. In general study, the distances of distribution, such as maximum mean discrepancy (MMD) [8], correlation alignment (CORAL) [9], and joint MMD (JMMD) [10], are utilized to narrow the domain shift. Xiao et al. [8] reduced the distribution mismatch between source features and target features through the MMD loss term. Wang et al. [9] developed a hierarchical deep DA approach for fault diagnosis by the multiple CORAL loss. JMMD was used by Liu et al. [10] for a deep autoencoder with joint distribution adaptation. To extract the invariant features between the source and target domains, researchers introduced ensemble learning and adversarial training into DA. Li et al. [11] used different kernel MMDs and weighted voting mechanism to construct an ensemble transfer diagnosis model. Han et al. [12] utilized adversarial learning as a regularization method to

boost the generalization ability of convolutional neural networks (CNNs). Jiao et al. [13] presented an unsupervised method called adversarial adaptation network, which uses the maximum classifier discrepancy to learn class-separable and domain-invariant features.

However, DA methods still suffer from some obstacles in solving the cross-domain fault diagnosis problem. First, the industry has difficulty in collecting sufficient samples in the target domain, which is a prerequisite of existing DA methods. For this reason, partial DA [14] and few-shot learning [15] are introduced into fault diagnosis. In the paradigm of partial DA, target data may never have the same health states collected in source data. Therefore, target label space becomes a subset of source label space that may cause negative transfer. The partial DA could leverage domain-asymmetry weight learned by the domain discriminator to train models under the share label spaces [16], [17]. In the paradigm of few-shot learning, limited target training may degenerate the conventional diagnosis model. To acquire a robust diagnosis model with limited data, meta-learning [18] and continual machine learning [19] could learn or transfer prior knowledge through multiple auxiliary tasks. However, when the issues worsen, i.e., the diagnosis model training has no access to target domain data, the methods above cannot be conducted directly. Second, traditional DA models can only generalize the knowledge from a single source domain to a specific target domain. These models could inevitably overfit on a single domain and thus may degenerate the generalization ability of the diagnosis model. Thus, it is necessary to exploit generic diagnosis knowledge across multiple domains [20], which have not been fully researched.

For this motivation, this study introduces domain generalization (DG) [21] into cross-domain fault diagnosis problems to remove the dependency on target domain data. The diagnosis idea based on DG raises a key question, i.e., how to learn a generalized fault feature representation by multiple available source domain data for the unseen target domain. Here, the unseen condition means that the target data are unavailable and have no contribution during the model training process.

In this study, we focus on DG-based cross-domain fault diagnosis and propose a generic domain-regressive framework named adversarial domain-invariant generalization (ADIG). The neural network in ADIG consists of three modules: feature extractor (*E*) with instance normalization (IN) strategy, fault classifier (*C*), and domain classifier (*D*) with spectral normalization (SN) strategy. During model training, *C* continually learns diagnosis knowledge, whereas *E* exploits shared representation from multiple domains. Instead of distinguishing source or target domain as a binary classification [12], *D* performs multiclassification to learn the difference among domains. In the stage of adversarial training, *E* and *D* play an adversarial game through a gradient reversal layer (GRL) [22] to learn fault-related and domain-invariant features. To balance the multitask loss dynamically, a weight coefficient learner is proposed in this work to achieve adaptive weights. Furthermore, model selection is performed on the basis of the validation set in the source domain.

The contributions of this work can be summarized as follows:

1) A novel domain-regressive framework ADIG is provided to diagnose faults under variable and unseen working conditions to fulfill cross-domain generalization. By multiple source domains adversarial learning, the ADIG can fully exploit domain-invariant knowledge to remove the dependency on target domain data, which is crucial but rarely researched.
2) Customized IN and SN strategies achieve cross-domain feature normalization to promote domain generalizability. Besides, an adaptive weight strategy achieves weight self-learning during multitask learning to improve performance.
3) Comprehensive experiments based on two case studies are conducted to evaluate the performance of ADIG. The results reveal that ADIG could be a paradigm facing unseen target condition diagnosis.

The rest of this paper is arranged as follows: Section II presents the preliminaries of this work. Section III provides the details of the proposed ADIG framework. Section IV presents the experimental results. Section V draws the conclusions.

II. PRELIMINARIES

A. Domain adaptation

The definitions of domain and task are given below.

The domain $\mathcal{D} = \{\mathcal{X}, P(X)\}$ includes a data space $\mathcal{X}$ and the marginal distribution $P(X)$ in the data space. Given a source domain $\mathcal{D}_S = \{\mathcal{X}_S, P(X_S)\}$ and a target domain $\mathcal{D}_T = \{\mathcal{X}_T, P(X_T)\}$, $\mathcal{X}_S \neq \mathcal{X}_T$ or (and) $P(X_S) \neq P(X_T)$ lead to $\mathcal{D}_S \neq \mathcal{D}_T$, where the subscripts S and T denote the source and target domains, respectively.

The task $\mathcal{T} = [\mathcal{Y}, C(\cdot)]$ includes the label space $\mathcal{Y}$ and the predictive function $C(\cdot)$. $C(\cdot)$ can be regarded as a conditional distribution description $P(Y|X)$. Thus, $\mathcal{Y}_S \neq \mathcal{Y}_T$ or (and) $C_S(\cdot) \neq C_T(\cdot)$ lead to $\mathcal{T}_S \neq \mathcal{T}_T$.

As shown in Fig. 1, DA aims to learn a predictive function $C(\cdot)$ by using the knowledge in the source domain when $\mathcal{D}_S \neq \mathcal{D}_T$. The source task and target task must be the same, and the unlabeled target data must be available in the model training. In a DA method, $L_S \sim \mathcal{D}_S$ means the labeled source dataset is drawn from source domain, where $L_S = \{(x_s^j, y_s^j)\}_{j=1}^{n_s}$ consists of the n_s data samples and their corresponding labels, and $U_T \sim \mathcal{D}_T$ means that the unlabeled target dataset is drawn from the target domain, and the dataset $U_T = \{(x_t^j)\}_{j=1}^{n_t}$ consists of n_t data samples in the target domain.

Reference [23] stated that the target generalization error of a classifier h can be bounded by the following formula:

$$\epsilon_T(h) \leq \epsilon_S(h) + \frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}\left(\mathcal{D}_T, \mathcal{D}_S\right) + \lambda, \tag{1}$$

where $\mathcal{H}$ is the hypothesis space, and ϵ_T, ϵ_S are the target and source generalization error of any classifier $h \in \mathcal{H}$, respectively. The second term $d_{\mathcal{H}\Delta\mathcal{H}}$ is the $\mathcal{H}\Delta\mathcal{H}$ distance between the source and target domains,

$$d_{\mathcal{H}\Delta\mathcal{H}}\left(\mathcal{D}_T,\mathcal{D}_S\right)=2\sup_{h,h'\in\mathcal{H}}\left|\begin{matrix}\Pr_{x\sim\mathcal{D}_T}\left[h(x)\neq h'(x)\right]\\-\Pr_{x\sim\mathcal{D}_S}\left[h(x)\neq h'(x)\right]\end{matrix}\right|. \quad (2)$$

In Eq. (1), λ is the error under an ideal joint hypothesis h^*,

$$\lambda=\epsilon_S\left(h^*\right)+\epsilon_T\left(h^*\right), \quad (3)$$

$$h^*=\arg\min_{h\in\mathcal{H}}\epsilon_S(h)+\epsilon_T(h). \quad (4)$$

The general DA methods attempt to estimate and reduce the source generalization error and the $\mathcal{H}\Delta\mathcal{H}$ distance, which can be approximated by MMD.

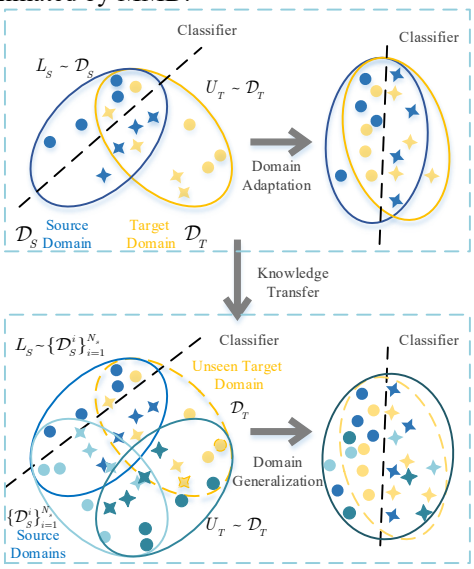

Fig. 1. Idea of knowledge transfer from DA to DG.

B. Domain generalization

DG is a methodology to learn knowledge from various related domains, which can help transfer the knowledge to an unseen target domain. The major differences among DG, DA, and general empirical risk minimization (ERM) are listed in Table I.

TABLE I
MAJOR DIFFERENCES AMONG ERM, DA, AND DG.

Methodology	Target domain data	Training Step	Inference Step
ERM	×	L^1	U^1
DA	available, unlabeled	L_s^1, U_t^1	U_t^1
DG	unavailable, unlabeled	$L_s^1,\ldots L_s^{N_s}$	U_t^1

ERM-based diagnosis methods have only one domain, and the target domain does not exist. The ERM models obey a default assumption wherein training data and testing data are with the same distribution, and they learn knowledge from labeled training dataset L^1 to infer the unlabeled testing dataset U^1. Conversely, DA-based diagnosis methods avoid the same domain/same distribution assumption, and they can take advantage of the available but unlabeled target domain dataset U_t^1 in the training step to extract domain-invariant features and solve the cross-domain fault diagnosis problem. DG is also a cross-domain method except that the task of DG is more difficult because target domain data are unavailable. However, DG is known for its strong ability in generalization from multiple source domains $L_s^1,\ldots,L_s^{N_s}$ to derive the domain-invariant knowledge, which is crucial for industrial fault diagnosis with complex working conditions and without target domain data.

In Fig. 1, we illustrate the knowledge transfer from DA to DG, which emphasizes the capability of the trained model to generalize the knowledge learned from the multiple source domains to the unseen target domain. The source domains can be denoted as $\mathcal{D}_S=\{\mathcal{D}_S^i\}_{i=1}^{N_s}$, where N_s is the number of source domains. The datasets of the source domain are denoted as $L_S=\{\{(x_s^{i,j},y_s^{i,j},d^i)\}_{j=1}^{n_s^i}\}_{i=1}^{N_s}$, where d^i is the domain label. The unlabeled target dataset drawn from target domain $U_T\sim\mathcal{D}_T$ is unavailable in the model training and model selection process.

According to the target generalization error bounded by DA, we can extend the error and hold it by [24],

$$\epsilon_U(h)\le\sum_{i=1}^{N_S}\pi_i\epsilon_S^i(h)+\frac{\gamma+\epsilon}{2}+\lambda, \quad (5)$$

where $\epsilon_U(h)$ is the error in the unseen target domain, π_i is the weight of source errors, and $\gamma+\epsilon$ is the sum of the $\mathcal{H}\Delta\mathcal{H}$ distance among domains.

C. Generative adversarial network

Generative adversarial network (GAN) [25] is a generative model to learn real-world data distribution through adversarial training between the generator and the discriminator. The generator learns the mapping from the latent space to the real-world data space, whereas the discriminator measures the discrepancy between real-world data and generated data. The two-player minimax game can be formulated as follows:

$$\min_G\max_D V(D,G)=E_{x\sim p_{data}(x)}[\log D(x)]+E_{z\sim p_z(z)}[\log(1-D(G(z)))]. \quad (6)$$

Recently, researchers have adapted the GAN model to the DA methods and called it the GRL [22], [26]. The forward behavior of GRL is an identity transformation, whereas the backpropagation behavior of GRL reverses the sign,

$$R(x)=x, \quad (7)$$

$$\frac{dR(x)}{dx}=-I, \quad (8)$$

where I is an identity matrix.

III. PROPOSED METHOD

A. Generic domain-regressive framework.

The framework of the proposed generic domain regression using ADIG is shown in Fig. 2.

First, concerning the existence of multiple available source domains, the domain label d^i of each domain in $L_S=\{L_s^i\}_{i=1}^{N_s}=\{\{(x_s^{i,j},y_s^{i,j},d^i)\}_{j=1}^{n_s^i}\}_{i=1}^{N_s}$ is added into the dataset, whereas the target domain data $U_T=\{(x_t^j)\}_{j=1}^{n_t}$ are unavailable during the training process. Each data sample is preprocessed into a two-dimensional (2-D) image by fast Fourier transform and the reshape operation; specially, 1-D input can also be integrated into the ADIG framework. In a similar way, the validation dataset is constructed for model selection by the preprocessing method above.

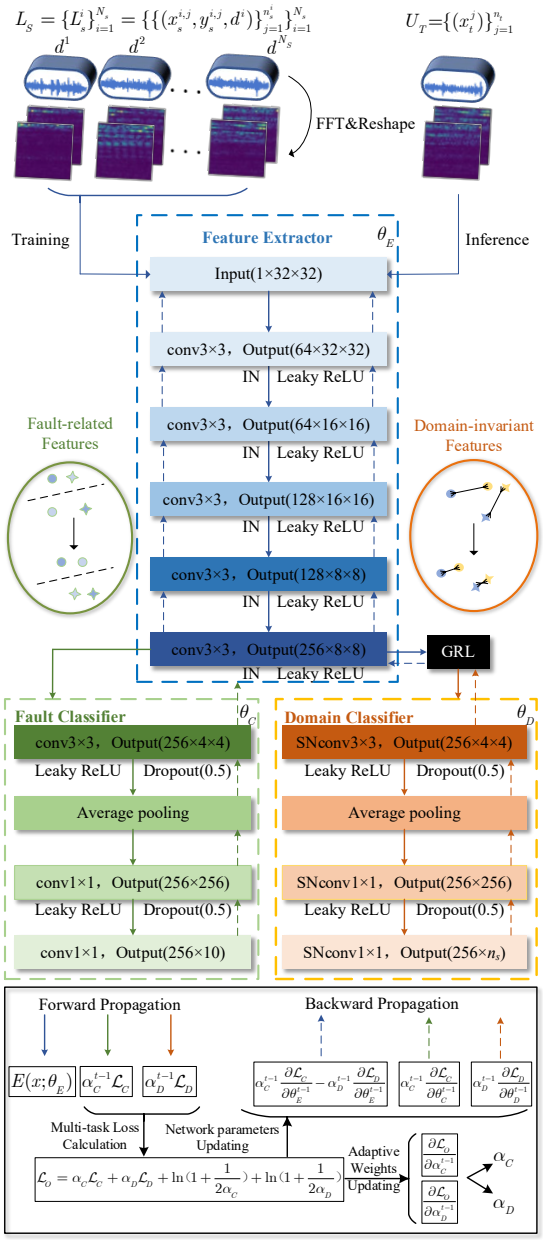

Fig. 2. Flowchart of the ADIG framework.

Second, considering the first two components of the target generalization error in Eq. (5), i.e. source generalization error $\epsilon_S^i(h)$ and $\mathcal{H}\Delta\mathcal{H}$ distance among all domains, we design three modules in the framework, i.e., a fault classifier (C) to reduce $\epsilon_S^i(h)$, a domain classifier (D) to estimate the $\mathcal{H}\Delta\mathcal{H}$ distance, and a feature extractor (E) that can extract domain-invariant features to reduce the estimated $\mathcal{H}\Delta\mathcal{H}$ distance.

In this study, we customize a CNN backbone to realize the above three modules. As shown in Fig. 2, the general block in the feature extractor is a combination of a 3×3 convolutional layer, an IN layer, and a Leaky ReLU activation function. The fault classifier consists of a convolutional layer, Leaky ReLU, Dropout, and average pooling, which can supervise the feature extractor to learn discriminative and fault-related features. To facilitate domain-invariant feature learning, an SN convolutional layer is introduced into the domain classifier. Similarly, the domain classifier consists of an SN convolutional layer, Leaky ReLU, Dropout, and average pooling layer, which promotes E to learn the domain-invariant features. The backbone in the framework can be adjusted to other CNN models, such as ResNet. The adversarial training process between E and D can be achieved by the GRL, which can reverse the gradient in the backpropagation procedure.

B. Feature normalization strategy

1) Instance normalization

IN is known to be a powerful tool in the style transfer task [27], which can be regarded as distribution alignment. Therefore, the IN layer is adopted in the backbone of the diagnosis model and is formulated as,

$$y_{lm} = \varphi\left(\frac{x_{lm}-\mu}{\sqrt{\sigma^2+\epsilon}}\right)+\beta\,, \tag{9}$$

where φ and β are the weight and bias of the IN layer, respectively. μ and σ^2 are the mean and variance of the single instance, respectively, with width W and height H.

$$\mu = \frac{1}{HW}\sum_{l=1}^{W}\sum_{m=1}^{H} x_{lm}\,, \tag{10}$$

$$\sigma^2 = \frac{1}{HW}\sum_{l=1}^{W}\sum_{m=1}^{H}(x_{lm}-\mu)^2\,, \tag{11}$$

2) Spectral normalization

Stable training is a troublesome problem for adversarial training. In ADIG, we consider the Lipschitz continuity and use the SN layer [28] to control the Lipschitz constant of D by the spectral norm,

$$\Omega(A) = \max_{h:h\neq 0}\frac{\|Ah\|_2}{\|h\|_2} = \max_{\|h\|_2\leq 1}\|Ah\|_2\,. \tag{12}$$

SN is likewise embedded in the backbone of ADIG by:

$$\overline{\theta}_D = \theta_D / \Omega(\theta_D)\,, \tag{13}$$

where θ is the parameter of the weight or bias in the network.

C. Optimization of adversarial domain generalization

1) Loss functions with adaptive weight strategy

The detailed optimization is introduced in this subsection. In the forward propagation, E computes the abstract feature from the prepared signal by multiple layers of nonlinear mapping. To learn the diagnosis knowledge between deep features and the fault types across multiple available domains, C predicts the fault pattern as the predicted value. Symmetrically, D builds a bond between deep feature and working conditions and predicts the domain pattern as the predicted value. Cross-entropy loss $\mathcal{L}$ is chosen to minimize the discrepancy between actual label and predicted value,

$$\mathcal{L}(y,\hat{y}) = -\frac{1}{M}\sum_{m=1}^{M} y \ln\frac{1}{\sum_{k=1}^{K} e^{((w_k)^T O+b_k)}}\begin{bmatrix} e^{((w_1)^T O+b_1)} \\ e^{((w_2)^T O+b_2)} \\ \vdots \\ e^{((w_K)^T O+b_K)} \end{bmatrix}. \tag{14}$$

In the formula above, the predicted value is calculated by the softmax function, and K is the number of categories. In this manner, two optimization objectives, i.e., fault classifier loss $\mathcal{L}_C$ and domain classifier loss $\mathcal{L}_D$ are calculated as follows:

$$\mathcal{L}_C = \sum_{i=1}^{N_s}\sum_{j=1}^{n_s^i/M}\mathcal{L}\left(C\left(E(x_s^{i,j};\theta_E);\theta_C\right), y_s^{i,j}\right), \tag{15}$$

$$\mathcal{L}_D = \sum_{i=1}^{N_s} \sum_{j=1}^{n_s^i/M} \mathcal{L}\left(D\left(R\left(E(x_s^{i,j};\theta_E);\theta_D\right)\right), d^i\right), \quad (16)$$

where θ_E , θ_C ,and θ_D are the parameters of *E*, *C*, and *D*, respectively; M is the batch size. Specifically, GRL, which is embedded between *E* and *D*, can be regarded as an identity transformation in the forward propagation rather than reverse transformation. Consequently, the optimization loss can be formulated as:

$$\mathcal{L}_O = \alpha_C \mathcal{L}_C + \alpha_D \mathcal{L}_D , \quad (17)$$

where α_C and α_D are the trade-off factors that can be further optimized. In the ADIG, the loss function of the diagnosis model above can be regarded as a multitask loss. For multitask learning [29], the performance of such model is strongly correlated with each task's loss, and these trade-off weights are tuned by hand, which is difficult and labor intensive. Thus, in this paper, these weights must be automatically set to save labor and control the contribution of the two losses in multitask learning. In [19], adaptive knowledge transfer was achieved by calculating the adaptive weights to constrain the parameter updating conditionally, i.e., a serial model learning the knowledge task by task. However, the model in our work performs multitask learning, which is a paralleled procedure to learn diagnosis and domain knowledge simultaneously. Therefore, in our method, we add these trade-off weights into the learned parameter sets as $\Theta=\{\theta_C,\theta_D,\theta_E,\alpha_C,\alpha_D\}$ to form a hyperparametric learning strategy, which is named adaptive weight strategy in the ADIG framework. Furthermore, these trade-off weights can be learned by the model itself. The final loss function is defined as follows:

$$\mathcal{L}_O = \alpha_C \mathcal{L}_C + \alpha_D \mathcal{L}_D + \ln(1+\frac{1}{2\alpha_C}) + \ln(1+\frac{1}{2\alpha_D}), \quad (18)$$

where $\ln(1+\frac{1}{2\alpha_C})$ is a regulation term with the negative gradient $-\frac{1}{\alpha_C(1+2\alpha_C)}$ when weights are positive, which can avoid the weight vanishing to become 0. Likewise, α_D can be restricted by its regulation term $\ln(1+\frac{1}{2\alpha_D})$. They are the essential terms because once the weights become 0, $\theta_C,\theta_D,\theta_E$ cannot be updated anymore.

2) Parameter optimization

Moreover, the parameter optimization of neural networks can be found through adversarial training by jointly satisfying:

$$\widehat{\theta_E} = \arg\left\{\min_{\theta_E} \alpha_C \mathcal{L}_C\left(\theta_E, \widehat{\theta_C}\right), \max_{\theta_E} \alpha_D \mathcal{L}_D\left(\theta_E, \widehat{\theta_D}\right)\right\}, \quad (19)$$

$$\widehat{\theta_C} = \arg\min_{\theta_C} \alpha_C \mathcal{L}_C\left(\widehat{\theta_E}, \theta_C\right), \quad (20)$$

$$\widehat{\theta_D} = \arg\min_{\theta_D} \alpha_D \mathcal{L}_D\left(\widehat{\theta_E}, \theta_D\right). \quad (21)$$

In this manner, the fault classifier with available labels supervises the feature extractor to learn discriminative and fault-related features. Simultaneously, the feature extractor is facilitated by the domain classifier to learn domain-invariant features through adversarial training. The stochastic gradient descent (SGD) is utilized in the adversarial game. Given that the GRL layer reverses the sign of the gradient in the backpropagation procedure, the optimization objective is inversed between *E* and *D*. Therefore, the parameter updating can be formulated as:

$$\theta_E^t \leftarrow \theta_E^{t-1} - \gamma\left(\alpha_C^{t-1}\frac{\partial \mathcal{L}_C}{\partial \theta_E^{t-1}} - \alpha_D^{t-1}\frac{\partial \mathcal{L}_D}{\partial \theta_E^{t-1}}\right), \quad (22)$$

$$\theta_D^t \leftarrow \theta_D^{t-1} - \gamma\alpha_D^{t-1}\frac{\partial \mathcal{L}_D}{\partial \theta_D^{t-1}}, \quad (23)$$

$$\theta_C^t \leftarrow \theta_C^{t-1} - \gamma\alpha_C^{t-1}\frac{\partial \mathcal{L}_C}{\partial \theta_C^{t-1}}, \quad (24)$$

where γ is the learning rate of the optimization algorithm. Synchronously, the trade-off weights are updated by:

$$\alpha_C^t \leftarrow \alpha_C^{t-1} - \gamma\frac{\partial \mathcal{L}_O}{\partial \alpha_C^{t-1}}, \quad (25)$$

$$\alpha_D^t \leftarrow \alpha_D^{t-1} - \gamma\frac{\partial \mathcal{L}_O}{\partial \alpha_D^{t-1}}. \quad (26)$$

The adaptive weights in our work are learned continually by SGD. The similar adaptive weight technique in reference [19] utilized iterative equation to fulfill continual learning, which is different from our learning strategy. Overall, the ADIG with adaptive weight strategy can be summarized in **Algorithm 1.**

Algorithm 1: ADIG

 # Training stage

Input: Multiple source dataset $L_S = \{\{(x_s^{i,j}, y_s^{i,j}, d^i)\}_{j=1}^{n_s^i}\}_{i=1}^{N_s}$.

Initialization: The module *E, C, D* with initialized parameter $\Theta=\{\theta_C,\theta_D,\theta_E,\alpha_C,\alpha_D\}$ and other pre-setting hyperparameters.

1: **for** *epoch* = 1 to *epochs* **do**

2: Randomly sample source data from L_S .

3: Forward propagation to calculate Eq. (18).

4: Backward propagation to update *E, C, D* by Eqs. (22-24).

5: Backward propagation to update α_C,α_D by Eqs. (25-26).

6: **end for**

Return: The optimal *E, C, D* selected by validation set.

 # Testing stage

Input: Unseen target dataset $U_T=\{(x_t^j)\}_{j=1}^{n_t}$.

Model: ADIG with optimal *E, C, D*.

Output: Diagnosis result of U_T by optimal *E* and *C*.

IV. EXPERIMENT ANALYSIS

A. Case 1: SCU

1) Experiment and dataset description

The bearing vibration data of the first case study are acquired from a test rig of our lab (SCU). As shown in Fig. 3, the running end of the bearing includes a motor, a plum coupling, a healthy bearing, a testing bearing (6205-2RS SKF), and an accelerometer connected with a NIPXle-1082 data acquisition system. In the loading end, a tread-and-nut system is utilized to adjust the loading from 0 KN to 3 KN, which can be measured by a SGSF-20K dynamometer.

In this case study, we collect 10 health conditions (one normal condition and nine fault conditions) into a dataset with a sampling frequency of 10 kHz. These bearing faults through wire cutting include outer race fault (O), inner race fault (I), and ball fault (B) with 0.2, 0.4, and 0.6 mm fault diameters, respectively. Thus, 10 health conditions can be abbreviated as

Nor, O02, O04, O06, I02, I04, I06, B02, B04, and B06. The dataset for each working condition is set by the data of the 10 health conditions, as well as their fault labels and domain labels. Each sample consists of 2048 data points, which can cover the fault information. In this work, we perform four DG tasks: T0, T1, T2, and T3. In each task, the datasets of the seen source domain have three working conditions. For example, task T0 means the ADIG model generalizes the domain-invariant knowledge from available source datasets under working condition 1 KN, 2 KN, and 3 KN to the unseen target dataset under working condition 0 KN.

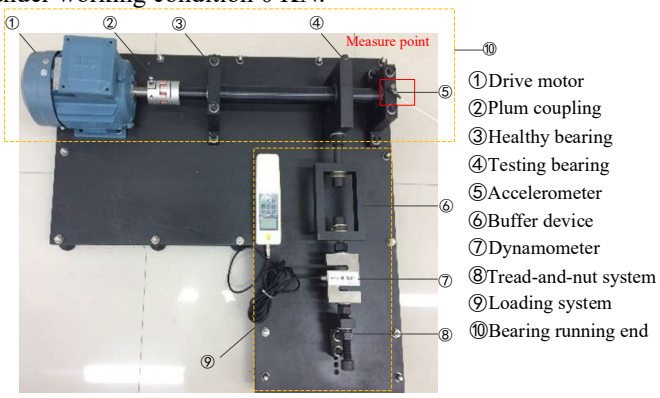

Fig. 3 Test rig of SCU.

2) Compared methods

Research on DG-based diagnosis methods to diagnose the unseen domain fault is relatively rare. Hence, in this work, we verify the ADIG framework based on the design of experiments and ablation experiments. The details are given in TABLE II. The backbone of ADIG is utilized in all compared methods for fair comparison.

In the first part, M1–M6 are designed as compared experiments extended from existing DA-based methods to show the effectiveness and superiority of the ADIG.

TABLE II
COMPARED METHODS

Method	Description
M1	CNN trained by single source domain data.
M2	CNN trained by multiple source domain data.
M3	CNN with JMMD [10] trained by multiple source domain data.
M4	CNN with MMD [8] trained by multiple source domain data.
M5	CNN with CORAL [9] trained by multiple source domain data.
M6	RTN [26] trained by multiple source domain data.
A1	Remove adaptive weight strategy. α_C is set by 0.1, α_D is set by 1.
A2	Remove adaptive weight strategy. α_C is set by 0.5, α_D is set by 1.
A3	Remove adaptive weight strategy. α_C is set by 1, α_D is set by 1.
A4	Remove IN strategy.
A5	Remove SN strategy.
A6	Replace IN strategy by batch normalization (BN).
A7	ADIG 1-D version.
ADIG	The proposed method.

M1 has different results according to which domain data are utilized. Thus, the best one in these results is chosen for comparison. M2–M6 are incorporated into the proposed generic domain-regressive framework. M2 uses the same fault classifier loss $\mathcal{L}_C$. The loss functions with adaptive weights in M3–M5 can be calculated on the basis of $\mathcal{L}_{JMMD}$ [10], $\mathcal{L}_{MMD}$ [8], and $\mathcal{L}_{CORAL}$ [9], respectively,

$$\mathcal{L}_{M3} = \alpha_C \mathcal{L}_C + \frac{\alpha_{M3}}{2} \sum_{p=1}^{N_s} \sum_{q \neq p}^{N_s} \mathcal{L}_{JMMD}(\mathcal{D}_S^p, \mathcal{D}_S^q), \quad (27)$$

$$\mathcal{L}_{M4} = \alpha_C \mathcal{L}_C + \frac{\alpha_{M4}}{2} \sum_{p=1}^{N_s} \sum_{q \neq p}^{N_s} \mathcal{L}_{MMD}(\mathcal{D}_S^p, \mathcal{D}_S^q), \quad (28)$$

$$\mathcal{L}_{M5} = \alpha_C \mathcal{L}_C + \frac{\alpha_{M5}}{2} \sum_{p=1}^{N_s} \sum_{q \neq p}^{N_s} \mathcal{L}_{CORAL}(\mathcal{D}_S^p, \mathcal{D}_S^q). \quad (29)$$

To show the versatility of the domain-regressive framework further, M6 is introduced into our backbone, i.e., a method similar with M4 but with a residual block in the fault classifier.

In the second part, we conduct ablation experiments, i.e., A1–A7, to show the necessity and importance of each strategy used in the ADIG framework. A1-A3 are used to verify adaptive weight strategy, while A4-A6 try to prove the effectiveness of normalization strategy. A7 is a 1-D version of ADIG with 1-D convolutional layer and 1-D inputs.

3) Parameter settings

The hyperparameters of ADIG are summarized in Table III.

TABLE III
HYPERPARAMETER SETTING

Hyperparameters	Value	Hyperparameters	Value
Learning rate γ	0.0001	Weight decay	0.0001
Batch size	128	Dropout Rate	0.5
α Initialization	C: 0.5; D: 1	Epoch	50(Case1);200(Case2)

These parameters are set in accordance with related works and grid search method. For example, grid search experiments are conducted to select the best hyperparameters. Specifically, the results of α_C and learning rate are shown in Fig. 4. The learning rate is searched in the set {0.1,0.01,0.001,0.0001,0.00001}, whereas the initialization of α_C is searched in the range [0.1,1]. The results show that the best selections of the initialization of α_C and learning rate are 0.5 and 0.0001. Moreover, the initialization of α_C is insensitive when the learning rate is set automatically by self-learning. Through the same method, α_D is set as 1, and the batch size is set as 128. The epoch in case 1 is 50, whereas that in case 2 is 200 due to the complexity of the problem. In addition, common hyperparameters, such as weight decay and dropout rate, are set as default referring to [8] and [12]. In all experiments, the hyperparameter settings are the same.

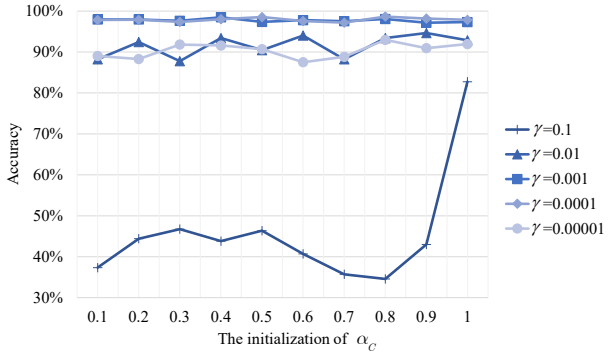

Fig. 4 Grid search experiments for α_C and γ

4) Result discussion

After model training and model selection, the average results of experiments in ADIG are compared with other methods in Fig. 5, and the ablation experiment results are given in Fig. 6.

Each experiment had five trials to eliminate contingency. The overall diagnosis results are listed in Table IV. It is obvious that the proposed ADIG outperforms other methods at the average level. Several interesting insights are revealed from the results.

TABLE IV
DIAGNOSIS RESULTS IN CASE 1(%)

Method	T0	T1	T2	T3	Average
M1	97.23±3.59	92.40±1.43	**97.97±6.48**	93.06±3.25	95.17
M2	97.11±0.82	97.70±0.30	97.47±1.47	95.02±0.79	96.83
M3	97.05±0.14	98.90±0.41	97.76±0.64	96.73±0.41	97.61
M4	96.67±0.29	98.61±0.36	97.41±0.77	95.96±0.50	97.16
M5	97.04±0.52	99.05±0.43	97.97±0.76	96.91±0.59	97.74
M6	97.50±0.18	98.87±0.32	97.22±1.32	95.98±0.52	97.39
A1	96.74±0.30	**99.01±0.23**	97.86±0.78	96.74±0.46	97.60
A2	96.64±0.30	98.78±0.31	97.83±0.59	97.13±0.42	97.60
A3	95.57±0.26	98.64±0.33	97.15±0.71	96.03±0.47	96.85
A4	14.80±3.93	19.33±4.31	23.59±5.18	17.12±4.11	18.71
A5	97.39±0.32	98.79±0.48	97.71±0.72	96.82±0.53	97.68
A6	**97.91±0.43**	98.07±0.37	97.32±0.90	96.50±0.49	97.45
A7	97.69±0.40	98.86±0.83	97.78±1.69	97.48±0.89	97.95
ADIG	97.67±0.29	98.89±0.21	97.82±0.76	**97.49±0.44**	**97.97**

Domain-invariant feature learning achieves reliable diagnosis under unseen condition by ADIG framework. The power of ADIG roots in that D in ADIG learns the distribution discrepancy among domains, whereas E learns the feature out of the domain distribution by $\mathcal{L}_D$, resulting in improved diagnosis performance in the unseen domain. From the perspective of generalization error, minimizing the empirical risk by $\mathcal{L}_C$ cannot generalize the knowledge well to the unseen domain, which can be seen from the results of M1 and M2. And the great aspect of ADIG to obtain improvement is that adversarial training rather than the traditional discrepancy metric in M2-M6 can further promote E to learn the domain-invariant features to narrow $\mathcal{H}\Delta\mathcal{H}$ distance through multiple available domains.

Each strategy and module in ADIG is indispensable and has its role in improving diagnosis performance. In Fig. 6, the effectiveness of adaptive weight strategy and the IN and SN strategies are verified by ablation experiments. Without adaptive weight, the diagnosis result worsens, especially in T0 and T3 tasks. The IN strategy has an advantage in promoting the diagnosis accuracy for the normalized first and second moments of each instance feature. In this case, A4 cannot fully learn the knowledge of the source domain. A5 shows that SN limits the Lipschitz continuity to facilitate adversarial training. Moreover, ADIG with 1-D and 2-D input have similar results when the $\mathcal{H}\Delta\mathcal{H}$ distance among domains is narrow.

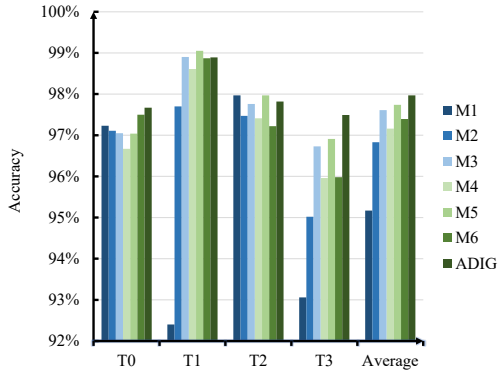

Fig. 5 Diagnosis performance comparisons in case 1.

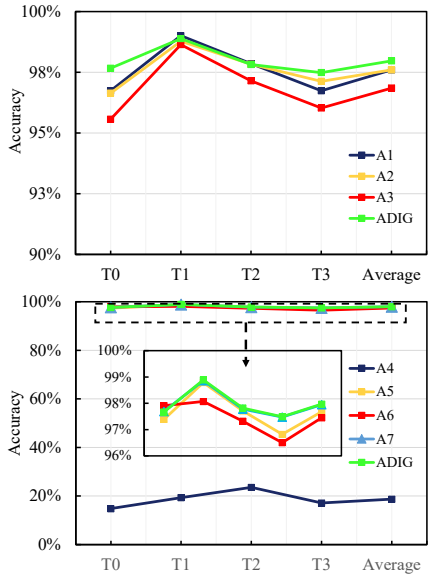

Fig. 6 Diagnosis performance of the variant ADIG in case 1.

Multiple domain adversarial training promotes model robustness. Comparing M1 with M2 shows that using multiple source domains to learn the cross-domain knowledge is a feasible way to diagnose unseen faults. Although M1 can obtain slightly higher accuracy in T0 and T2, the standard deviation (std) in M1 is remarkably higher than the std in M2, which validates the better robustness of M2 trained by multiple source domain data. In addition, the reason why M1 achieves higher accuracy in some tasks is that the source domain with best result has the narrowest $\mathcal{H}\Delta\mathcal{H}$ distance to the unseen domain. In other words, without the domain-regressive framework, it cannot be strictly guaranteed that multiple domain training is always positive.

Learning Low $\mathcal{H}\Delta\mathcal{H}$ distance features facilitate diagnosis. In case 1, the four DA-extended methods M3-M6 with distribution discrepancy loss achieve similar performance, indicating that source generalization error rather than $\mathcal{H}\Delta\mathcal{H}$ distance plays a major role, whereas ADIG can further exploit domain-invariant features with lower $\mathcal{H}\Delta\mathcal{H}$ distance among these methods. Additionally, DG is relatively easy to achieve when the unseen domain is between several seen domains. The evidence is that the tasks T0 and T3 are much more difficult to realize in Fig. 5 due to the huge difference in $\mathcal{H}\Delta\mathcal{H}$ distance between the available source domain and unseen target domain.

B. Case 2: SDUST

1) Dataset description

To verify the superiority of the ADIG framework, a bearing fault dataset from Shandong University of Science and Technology (SDUST) is adopted [30]; data are collected at variable speeds. The experimental setup for data collection is shown in Fig. 7. The sampling frequency is set at 25.6 kHz. The engine rotates at a speed of 2000 r/min, and the bearing type is N205EU. The dataset contains four health states as case 1, i.e., normal, inner race fault, outer race fault, and rolling element fault. Each fault type has three damage dimensions, specifically, 0.2, 0.4, and 0.6 mm. Four datasets are collected under different speeds, i.e., 500, 1000, 1500, and 2000 r/min. Thus, a domain

dataset has 10 health conditions, and four domain datasets have different speeds.

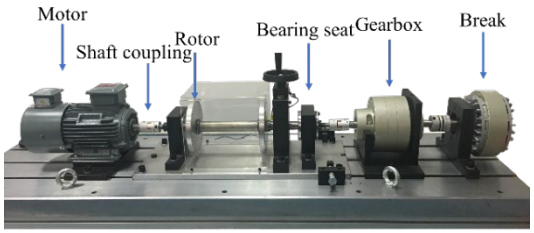

Fig. 7 SDUST Experimental setup for machinery fault diagnosis.

In this case study, we perform three DG tasks: T1000, T1500, and T2000. For example, task T1000 is to generalize the domain-invariant knowledge from the seen source datasets under the speeds 500, 1500, and 2000 r/min to the unseen target dataset under the speed 1000 r/min. T1500 and T2000 follow the same definition.

2) Result discussion

The diagnosis results are shown in Fig. 8, Fig. 9 and TABLE V. Some new findings are highlighted beyond case study 1.

TABLE V
DIAGNOSIS RESULTS IN CASE 2(%)

Method	T1000	T1500	T2000	Average
M1	48.32±5.73	61.34±10.35	71.74±2.55	60.47
M2	73.25±4.22	94.71±3.44	81.48±1.89	83.15
M3	85.94±2.66	92.81±2.08	89.98±3.17	89.58
M4	76.35±2.37	**98.03±1.84**	76.92±7.83	83.77
M5	53.79±3.48	92.06±2.26	69.28±6.08	71.71
M6	86.05±1.69	91.77±1.97	90.95±3.11	89.59
A1	85.24±1.33	92.62±2.98	92.59±3.86	90.15
A2	86.61±3.88	94.45±3.29	87.84±1.78	89.63
A3	83.94±3.01	93.42±2.95	91.98±4.04	89.78
A4	69.26±12.65	79.39±21.24	46.92±13.46	65.19
A5	88.25±4.75	96.46±1.94	**92.40±3.83**	92.37
A6	85.30±2.93	91.45±0.86	88.05±3.20	88.27
A7	82.13±3.21	95.82±1.10	88.01±5.62	88.65
ADIG	**91.09±0.94**	96.47±2.60	91.43±3.10	**93.00**

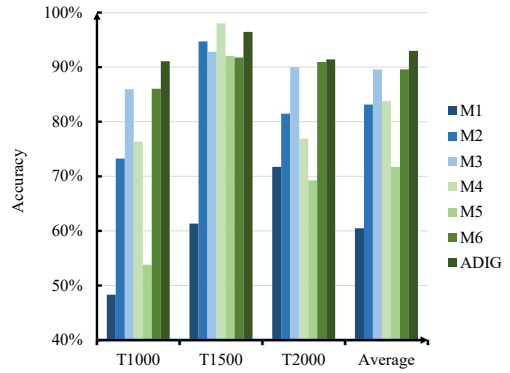

Fig. 8 Diagnosis performance comparisons in case 2.

High $\mathcal{H}\Delta\mathcal{H}$ distance degenerates cross domain diagnosis performance, while ADIG still achieves best results in case 2. The gaps between ADIG and other methods are widened due to the higher $\mathcal{H}\Delta\mathcal{H}$ distance among domains caused by variant rotating speed. For instance, M2 obtains over 80% accuracy by learning from multiple domains, whereas M1 obtains only 60.47%. The $\mathcal{H}\Delta\mathcal{H}$ distance could be further narrowed by M3–M6, indicating that $\mathcal{H}\Delta\mathcal{H}$ distance plays a more important role in case 2. When the DG task becomes more difficult, negative knowledge transfer inevitably occurs, e.g., CORAL loss-based method in M5 is no longer effective for case 2. By contrast, significant improvements of ADIG further prove the superiority of domain-invariant knowledge learned by adversarial learning from multiple domains. The ADIG can estimate the $\mathcal{H}\Delta\mathcal{H}$ distance by D to avoid the degradation in different case studies when facing huge distribution discrepancies.

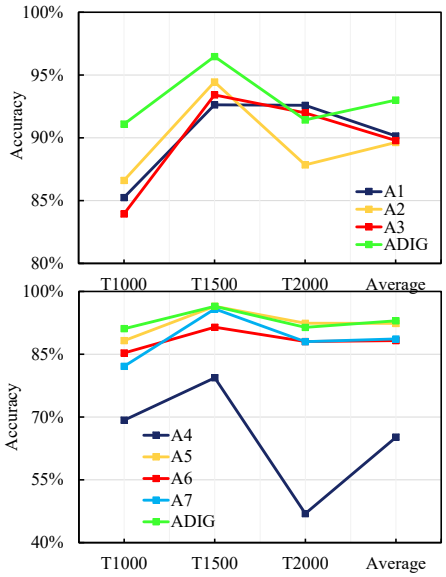

Fig. 9 Diagnosis performance of the variant ADIG in case 2.

Given more complex generalization tasks, strategies in ADIG still have their effectiveness. Fig. 9 demonstrates that the best weight about $\mathcal{L}_C$ and $\mathcal{L}_D$ is not constant. The optimized weights in different tasks seem to be different; in this case, the adaptive weight strategy is of great significance. Compared with ADIG and A4, IN improves nearly 28% accuracy, which shows a significant attribute of IN when facing a huge domain discrepancy. By contrast, A6 performs worse than ADIG, which again proves the importance of the IN strategy in the generalization problem. Similarly, the result of A5 supports that the SN strategy can facilitate diagnosis performance.

The 2-D input facilities domain-invariant feature learning under dramatically changed unseen work condition. Compared the results of A7 and ADIG in case 2, it can be found that when variant rotating speed enlarges $\mathcal{H}\Delta\mathcal{H}$ distance, using 2-D input can further exploit domain-invariant features to promote diagnosis performance. In other words, facing more difficult generalization tasks, ADIG with 2-D can learn features with lower $\mathcal{H}\Delta\mathcal{H}$ distance than 1-D.

3) Visualization of learned representation

To demonstrate the fault feature distribution and explain the mechanism of ADIG, the high-dimension features extracted by E are reduced into a 2-D feature representation by T-SNE [31] for improved visual understanding. The feature vectors learned under task T2000 are illustrated in Fig. 10 for illustration purposes. We only plot four health conditions (Nor, I06, B06, and O06). M1 (worst) means the model is trained by the dataset under 500 r/min, whereas M1 (best) is the result under 1500 r/min.

In Fig. 10, the green marks are the features in the unseen target domain, whereas the rest are the features in available source domains. ADIG aims to learn the generalized feature representations by using the available data in the source domain. Specifically, the unseen domain feature representation (the

green marks) extracted by the learned model aims to overlap with the available source domain feature representation.

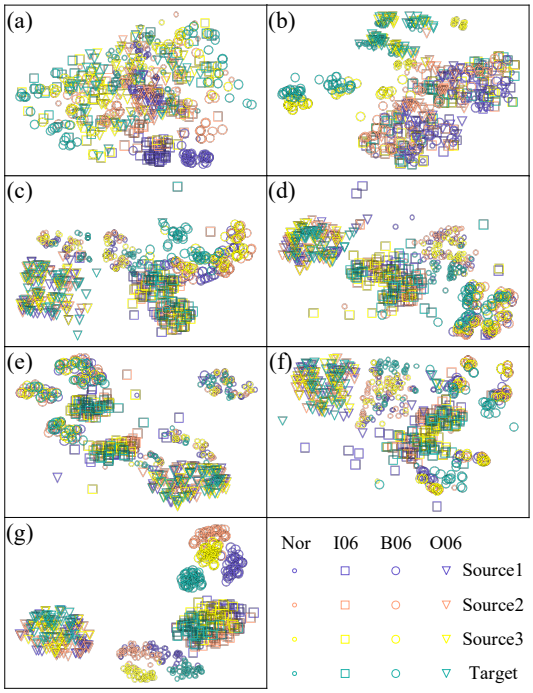

Fig. 10 Feature visualization for task T2000. (a) M1(worst), (b) M1(best), (c) M2, (d) M5, (e) M4, (f) M3, (g) ADIG.

A comparison of Fig. 10(a) and Fig. 10(b) reveals that only the trained domain features are clustered by the learned knowledge and that single domain knowledge is difficult to generalize to the unseen domain. Given the low $\mathcal{H}\Delta\mathcal{H}$ distance, the most related domain can more or less be clustered. Thus, the knowledge learned by the single domain cannot be generalized to other domains well without multiple related domains. Fig. 10(c) illustrates the feature vectors learned from multiple related source domains, whereas the methods in Fig. 10(d)–Fig. 10(f) take advantage of the distribution discrepancy loss. In these methods, feature distribution biases or domain shifts may still occur between the source and target domains, although the model learns the knowledge from different source domains. The features learned from the related source domain by ADIG are clustered better than those by other methods. In ADIG, the fault features of three source domains and an unseen target domain can be gathered into a cluster, thus proving the effectiveness and the domain-invariant feature learning of the proposed framework.

V. CONCLUSION

Three contributions are presented in this study. First, we propose a novel insight, generic domain-regressive framework, which can diagnose unseen fault patterns in the target domain. Second, we propose a novel ADIG fault diagnosis framework to diagnose the unseen domain faults with three customized modules, i.e., feature extractor, domain classifier, and fault classifier. Through adversarial training, domain-invariant and fault-related knowledge are learned from multiple domains. Third, the adaptive weight strategy together with the normalization strategy facilitates training to learn and transfer additional domain-invariant features to the target domain. Visualization representation manifests the internal feature cluster of the proposed method, and comprehensive experiments prove that ADIG can generalize the knowledge to an unseen target domain, thus proving the superiority of ADIG for real industrial applications.

REFERENCES

[1] R. C. Luo and H. Wang, "Diagnostic and Prediction of Machines Health Status as Exemplary Best Practice for Vehicle Production System," *IEEE Veh. Technol. Conf.*, vol. 2018-Augus, pp. 1–5, 2018.

[2] Y. Lei, F. Jia, J. Lin, S. Xing, and S. X. Ding, "An Intelligent Fault Diagnosis Method Using Unsupervised Feature Learning Towards Mechanical Big Data," *IEEE Trans. Ind. Electron.*, vol. 63, no. 5, pp. 3137–3147, 2016.

[3] X. Jiang, C. Shen, J. Shi, and Z. Zhu, "Initial center frequency-guided VMD for fault diagnosis of rotating machines," *J. Sound Vib.*, vol. 435, pp. 36–55, 2018.

[4] Z. Pan, Z. Meng, Z. Chen, W. Gao, and Y. Shi, "A two-stage method based on extreme learning machine for predicting the remaining useful life of rolling-element bearings," *Mech. Syst. Signal Process.*, vol. 144, p. 106899, 2020.

[5] C. Shen, Y. Qi, J. Wang, G. Cai, and Z. Zhu, "An automatic and robust features learning method for rotating machinery fault diagnosis based on contractive autoencoder," *Eng. Appl. Artif. Intell.*, vol. 76, no. 8, pp. 170–184, 2018.

[6] L. Cui, X. Wang, H. Wang, and J. Ma, "Research on Remaining Useful Life Prediction of Rolling Element Bearings Based on Time-Varying Kalman Filter," *IEEE Trans. Instrum. Meas.*, vol. PP, no. c, pp. 1–1, 2019.

[7] H. Zheng *et al.*, "Cross-Domain Fault Diagnosis Using Knowledge Transfer Strategy: A Review," *IEEE Access*, vol. 7, pp. 129260–129290, 2019.

[8] D. Xiao, Y. Huang, L. Zhao, C. Qin, H. Shi, and C. Liu, "Domain Adaptive Motor Fault Diagnosis Using Deep Transfer Learning," *IEEE Access*, vol. 7, pp. 80937–80949, 2019.

[9] X. Wang, H. He, and L. Li, "A Hierarchical Deep Domain Adaptation Approach for Fault Diagnosis of Power Plant Thermal System," *IEEE Trans. Ind. Informatics*, vol. 15, no. 9, pp. 5139–5148, 2019.

[10] 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 Sens. J.*, vol. 19, no. 24, pp. 12261–12270, 2019.

[11] X. Li, H. Jiang, R. Wang, and M. Niu, "Rolling bearing fault diagnosis using optimal ensemble deep transfer network," *Knowledge-Based Syst.*, vol. 213, p. 106695, 2021.

[12] T. Han, C. Liu, W. Yang, and D. Jiang, "A novel adversarial learning framework in deep convolutional neural network for intelligent diagnosis of mechanical faults," *Knowledge-Based Syst.*, vol. 165, pp. 474–487, 2019.

[13] J. Jiao, M. Zhao, and J. Lin, "Unsupervised Adversarial Adaptation Network for Intelligent Fault Diagnosis," *IEEE Trans. Ind. Electron.*, vol. 67, no. 11, pp. 9904–9913, 2020.

[14] Z. H. Liu, B. L. Lu, H. L. Wei, L. Chen, X. Li, and C. T. Wang, "A Stacked Auto-Encoder Based Partial Adversarial Domain Adaptation Model for Intelligent Fault Diagnosis of Rotating Machines," *IEEE Trans. Ind. Informatics*, vol. 3203, no. MMD, 2020.

[15] C. Li, S. Li, A. Zhang, and Q. He, "Meta-Learning for Few-Shot Bearing Fault Diagnosis under Complex Working Conditions," *Neurocomputing*, vol. 439, pp. 197–211, 2021.

[16] B. Yang, C. G. Lee, Y. Lei, N. Li, and N. Lu, "Deep partial transfer learning network: A method to selectively transfer diagnostic knowledge across related machines," *Mech. Syst. Signal Process.*, vol. 156, 2021.

[17] Y. Deng, D. Huang, S. Du, G. Li, C. Zhao, and J. Lv, "A double-layer attention based adversarial network for partial transfer learning in machinery fault diagnosis," *Comput. Ind.*, vol. 127, p. 103399, 2021.

[18] Z. He, H. Shao, P. Wang, J. (Jing) Lin, J. Cheng, and Y. Yang, "Deep transfer multi-wavelet auto-encoder for intelligent fault diagnosis of gearbox with few target training samples," *Knowledge-Based Syst.*, vol. 191, p. 105313, 2020.

[19] S. Xing, Y. Lei, B. Yang, and N. Lu, "Adaptive Knowledge Transfer by Continual Weighted Updating of Filter Kernels for Few-shot Fault

Diagnosis of Machines," *IEEE Trans. Ind. Electron.*, vol. 0046, no. c, 2021.

[20] J. Zhu, N. Chen, and C. Shen, "A New Multiple Source Domain Adaptation Fault Diagnosis Method Between Different Rotating Machines," *IEEE Trans. Ind. Informatics*, vol. 17, no. 7, pp. 4788–4797, 2021.

[21] K. Muandet, D. Balduzzi, and B. Schölkopf, "Domain generalization via invariant feature representation," *30th Int. Conf. Mach. Learn. ICML 2013*, vol. 28, no. PART 1, pp. 10–18, 2013.

[22] Y. Ganin *et al.*, "Domain-adversarial training of neural networks," *J. Mach. Learn. Res.*, vol. 17, pp. 1–35, 2016.

[23] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan, "A theory of learning from different domains," *Mach. Learn.*, vol. 79, no. 1–2, pp. 151–175, 2010.

[24] I. Albuquerque, J. Monteiro, M. Darvishi, T. H. Falk, and I. Mitliagkas, "Generalizing to unseen domains via distribution matching," pp. 1–15, 2019.

[25] I. J. Goodfellow *et al.*, "Generative adversarial nets," *Adv. Neural Inf. Process. Syst.*, vol. 3, no. January, pp. 2672–2680, 2014.

[26] M. Long, H. Zhu, J. Wang, and M. I. Jordan, "Unsupervised Domain Adaptation with Residual Transfer Networks," no. Nips, 2016.

[27] Y. Li, N. Wang, J. Liu, and X. Hou, "Demystifying neural style transfer," *IJCAI Int. Jt. Conf. Artif. Intell.*, vol. 0, pp. 2230–2236, 2017.

[28] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral normalization for generative adversarial networks," *6th Int. Conf. Learn. Represent. ICLR 2018 - Conf. Track Proc.*, 2018.

[29] R. Cipolla, Y. Gal, and A. Kendall, "Multi-task Learning Using Uncertainty to Weigh Losses for Scene Geometry and Semantics," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 7482–7491, 2018.

[30] S. Jia, J. Wang, B. Han, G. Zhang, X. Wang, and J. He, "A novel transfer learning method for fault diagnosis using maximum classifier discrepancy with marginal probability distribution adaptation," *IEEE Access*, vol. 8, pp. 71475–71485, 2020.

[31] P. E. Rauber, A. X. Falcão, and A. C. Telea, "Visualizing Time-Dependent Data Using Dynamic t-SNE," in *Eurographics Conference on Visualization*, 2016, p. 2016.