

# Continuously Indexed Domain Generalization for Fault Diagnosis Under Continuously Varying Working Conditions

Chenhao Wang<sup>1</sup>, Liling Ma<sup>1</sup>, Jiameng Wang<sup>1</sup>, Runjiao Bao<sup>1</sup>, *Student Member, IEEE*,  
Hao Yu<sup>1</sup>, *Member, IEEE*, and Shoukun Wang<sup>1</sup>

**Abstract**—Deep learning-based fault diagnosis often suffers from significant accuracy degradation under varying working conditions. While domain generalization (DG) methods can ensure consistency in fault prediction when the unavailable test data is out of domain, existing research typically involves discretizing continuous operating conditions, which fails to fully utilize the continuity of common factors such as speed and load. This oversight limits improvements in generalization capability. To address this, we propose a fault diagnosis framework suitable for continuously varying working conditions, utilizing continuously indexed DG methods. This framework builds on wavelet transform and convolutional neural networks, incorporating continuous domain mix (CDMix) data augmentation and continuous mutual information minimization (CMIM) loss constraints to achieve cross-DG. Specifically, CDMix determines mixing probabilities by measuring the distance between domain labels of samples, enhancing the reliability of the generated out-of-domain data. Meanwhile, CMIM estimates the mutual information between features and continuous domain labels using kernel methods and then minimizes it to guide the network in extracting fault features that are invariant to continuous working conditions. The proposed methods improve the generalization capability and cross-domain fault classification accuracy of fault diagnosis models. Extensive experiments on gearbox and bearing fault datasets validate the effectiveness of the proposed CDMix–CMIM framework, demonstrating significant superiority over existing methods and good robustness.

**Index Terms**—Continuously indexed domain generalization (DG), continuously varying working conditions, data augmentation, fault diagnosis, mutual information.

## I. INTRODUCTION

IN THE era of smart manufacturing, intelligent fault diagnosis has become crucial for ensuring operational efficiency and minimizing downtime [1], [2], [3]. A common practice is to extract fault features from vibration signals using neural networks for classification [4], or to enhance fault-related patterns by combining advanced signal-processing techniques such as

the wavelet transform [5]. However, these methods generally assume consistent data distributions without domain shifts. As industrial systems operate under varying working conditions, such as changes in speed and load, the resulting domain shifts can significantly impact the performance of diagnostic models. These shifts lead to discrepancies between the training and testing environments, causing traditional fault diagnosis methods to struggle with generalization and accuracy when faced with new operational scenarios [6].

Domain adaptation (DA) techniques aim to reduce discrepancies between the source and target domains, enabling models trained on one domain to perform well in another. Most DA approaches focus on extracting domain-invariant feature representations through adversarial training or discrepancy-based feature alignment [7]. For instance, Zhong et al. [8] aligned marginal distributions using multikernel MMD and minimized conditional distribution differences with a Wasserstein-based adapter, while Qin et al. [9] jointly adapted marginal and conditional distributions to reduce domain shifts in transfer diagnosis. However, DA methods typically require access to target-domain data during training, which is often impractical in dynamic industrial environments. This limitation motivates the use of domain generalization (DG), where models are designed to generalize to unseen domains without target-domain data.

Existing DG approaches can be roughly grouped into three directions, namely domain-invariant feature extraction, data augmentation, and other learning strategies [10]. Domain-invariant feature extraction methods seek to align representations across domains. These include discrepancy-based techniques such as maximum mean discrepancy and CORAL, adversarial-based approaches that confuse domain discriminators, and feature disentanglement methods that leverage causal learning to separate domain-shared and domain-specific components. Pu et al. [11] proposed a domain-relevant joint distribution alignment method to align feature-layer domain-joint distributions, using  $\alpha$ -PE divergence as an evaluation metric and a parameter-free data augmentation module to enhance generalization capability. Jia et al. [12] disentangled fault-related causal factors from domain-related noncausal factors to extract domain-invariant features for cross-domain fault diagnosis. Data augmentation strategies improve robustness by generating perturbed or mixed samples, for example, through

Received 19 July 2025; revised 10 September 2025; accepted 23 September 2025. Date of publication 20 October 2025; date of current version 4 November 2025. This work was supported by the National Natural Science Foundation (NSFC) of China under Grant 62473044. The Associate Editor coordinating the review process was Dr. Yifan Li. (*Corresponding author: Liling Ma.*)

The authors are with the School of Automation, Beijing Institute of Technology, Beijing 100081, China (e-mail: 3120230774@bit.edu.cn; maliling@bit.edu.cn; 3120240799@bit.edu.cn; 3120230765@bit.edu.cn; yuhaocsc@bit.edu.cn; bitwsk@bit.edu.cn).

Digital Object Identifier 10.1109/TIM.2025.3623768

1557-9662 © 2025 IEEE. All rights reserved, including rights for text and data mining, and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission.

See <https://www.ieee.org/publications/rights/index.html> for more information.

Authorized licensed use limited to: BEIJING INSTITUTE OF TECHNOLOGY. Downloaded on November 27, 2025 at 10:05:18 UTC from IEEE Xplore. Restrictions apply.

Mixup or CutMix. Fan et al. [13] proposed a DG method that mixes time–frequency images in both class and domain spaces, utilizing a domain discriminator module and a domain-based discrepancy metric module. Beyond these commonly used strategies, other learning strategies further expand DG, including meta-learning that simulates cross-task adaptation [14], distributionally robust optimization that enhances worst case generalization [15], and federated learning that enables collaborative model training across decentralized data sources [16], [17].

Although these approaches have achieved progress, most existing DG methods mainly focus on discrete domain shifts, where domains are represented by separate identifiers. In contrast, real-world industrial systems often exhibit continuous variations in working conditions, such as gradually changing speed and load. Directly applying discrete-domain DG techniques fails to capture the smooth and evolving nature of such conditions, leaving a critical gap in the literature. This article aims to bridge this gap by proposing a DG framework tailored for continuously indexed domains in fault diagnosis.

To address the challenge of DG with continuous domain labels in fault diagnosis, this article presents a novel solution that extends regular DG methods to this specific context. We design a fault diagnosis framework that includes a continuous domain-compatible mix augmentation method, a fault feature extraction module based on the wavelet transform and a convolutional encoder, a fault classifier, and a continuous mutual information minimization (CMIM) loss constraint. Specifically, we first utilize the continuous domain mix (CDMix) data augmentation method to generate out-of-domain data, enhancing the model's generalization performance. Next, we convert signals into time–frequency representations using wavelet transforms and employ a convolutional encoder for feature extraction. Ultimately, a fault classifier is used to identify feature types. During training, we introduce a soft constraint as an additional loss based on CMIM, which encourages the feature extractor to focus on domain-invariant fault features by minimizing the mutual information between the extracted features and the continuous domain labels. Our contributions are threefold.

- 1) The CDMix method is developed as a mix augmentation technique specifically tailored for continuously indexed DG, enhancing data diversity in fault diagnosis under continuously varying working conditions.
- 2) The CMIM loss is designed for robust domain-invariant feature extraction by estimating and minimizing the mutual information between features and continuous working condition data.
- 3) Extensive experiments are conducted on a gearbox fault dataset and a bearing dataset collected under time-varying working conditions to validate the effectiveness of the proposed methods.

The structure of this article is organized as follows. Section II provides background knowledge on the DG problem and the mix augmentation technique. Section III details the proposed methodologies. Section IV presents the experiments on two types of datasets. Section V discusses model performance through feature visualization, ablation studies, sample

imbalance analysis, and parameter sensitivity analysis. Finally, Section VI concludes this article and discusses potential future work.

## II. PRELIMINARIES

### A. Continuously Indexed DG

In the continuously indexed DG scenario, the source domain with training data is given as  $\mathcal{D}_s = \{(x_i^s, y_i^s, u_i^s)\}_{i=1}^{N_s}$ . The unseen target domain, with unavailable test data, is denoted as  $\mathcal{D}_t = \{(x_i^t, y_i^t, u_i^t)\}_{i=1}^{N_t}$ . Here,  $x \in \mathcal{X}$  and  $y \in \mathcal{Y}$  represent the input data and labels, respectively. The input space is defined as  $\mathcal{X} = \mathcal{X}_s \cup \mathcal{X}_t$ , while the output space is  $\mathcal{Y} = \mathcal{Y}_s \cup \mathcal{Y}_t$ . Additionally,  $u \in \mathcal{U}$  denotes the continuous domain indices, where  $\mathcal{U} = \mathcal{U}_s \cup \mathcal{U}_t$  is part of a metric space. In the context of fault diagnosis,  $x$ ,  $y$ , and  $u$  represent the vibration signals of the system, fault type labels, and continuous working condition information, respectively. Here, “continuous” refers to the fact that  $u$  consists of real-valued measurements of working conditions (e.g., normalized speed and torque), which vary over a continuous range. This differs from discrete domain labels and is a different concept from online or ongoing adaptation to continuously arriving target domains.

The goal of continuously indexed DG, consistent with that of vanilla DG, is to train a robust and generalizable predictive network  $f: \mathcal{X} \rightarrow \mathcal{Y}$  using the training domain  $\mathcal{D}_s$ , aiming to minimize the prediction error on the unseen test domain  $\mathcal{D}_t$ , which can be expressed as

$$\min_f \mathbb{E}_{(x_t, y_t) \in \mathcal{D}_t} [\ell(f(x_t), y_t)] \quad (1)$$

where  $\mathbb{E}$  denotes the expectation and  $\ell(\cdot, \cdot)$  is the loss function [18]. In the context of continuously indexed DG for fault diagnosis, the objective is to train a neural network that is robust to variations in working conditions, using training data with continuous condition information, allowing it to focus on classifying fault categories from data collected under different working conditions.

### B. Mix Data Augmentation

Mixup is a data augmentation technique that generates new training samples by interpolating between existing ones, enhancing model robustness by promoting smoother decision boundaries [19]. For two samples  $(x_i, y_i)$  and  $(x_j, y_j)$ , where  $y$  is a one-hot encoded label, Mixup creates new samples  $(\tilde{x}, \tilde{y})$  as

$$\begin{aligned} \tilde{x} &= \lambda x_i + (1 - \lambda) x_j \\ \tilde{y} &= \lambda y_i + (1 - \lambda) y_j \end{aligned} \quad (2)$$

where  $\lambda$  is sampled from a Beta distribution. This approach has a significant advantage over traditional single-sample data augmentation methods, which typically transform individual samples without considering their relationships. By blending samples, Mixup encourages the model to learn more generalized features and smoother decision boundaries.

A similar approach, CutMix, combines different images and labels, allowing the model to learn from localized features

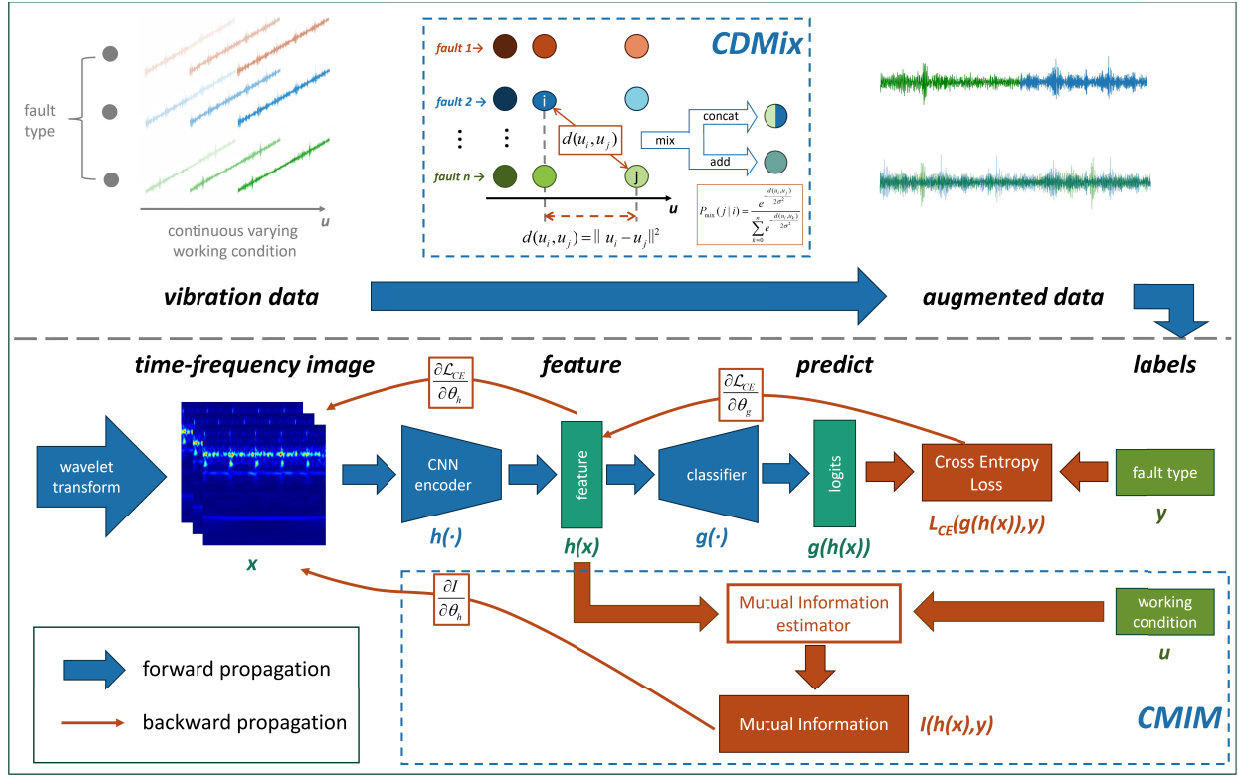


Fig. 1. Framework of the proposed continuously indexed DG method with CDMix and CMIM for fault diagnosis.

while maintaining spatial coherence [20]. For CutMix, the samples are generated as

$$\begin{aligned}\tilde{x} &= M \odot x_i + (1 - M) \odot x_j \\ \tilde{y} &= \lambda y_i + (1 - \lambda) y_j\end{aligned}\quad (3)$$

where the operator  $\odot$  denotes the element-wise multiplication and  $M$  is a binary mask indicating the area of the image used, with a bounding box coordinate  $(r_x, r_y, r_w, r_h)$  according to

$$\begin{aligned}r_x &\sim \text{Unif}(0, W), \quad r_w = W\sqrt{1 - \lambda} \\ r_y &\sim \text{Unif}(0, H), \quad r_h = H\sqrt{1 - \lambda}\end{aligned}\quad (4)$$

where  $W$  and  $H$  are the width and height of the input image, respectively.

In the context of DG, Mixup can be extended to incorporate domain labels by mixing not only the data and labels but also the domain labels. This expansion allows for enhanced adversarial training in DG, represented as

$$\begin{aligned}\tilde{x} &= \lambda x_i^d + (1 - \lambda) x_j^{d'} \\ \tilde{y} &= \lambda y_i + (1 - \lambda) y_j \\ \tilde{d} &= \lambda d_i + (1 - \lambda) d_j\end{aligned}\quad (5)$$

where  $d$  represents the domain labels and  $d_i$  and  $d_j$  are the domain labels for the respective samples. This method leverages domain adversarial training to improve model performance across varying domains [21].

Considering that Mixup is typically used for classification problems, directly using Mixup on continuous labels can result in arbitrarily incorrect labels. To address this issue, C-Mixup extends the Mixup concept specifically for regression tasks by

calculating the distance between labels to sample closer pairs, improving generalization across various regression problems. The probability for mixing is defined as

$$P((x_j, y_j) | (x_i, y_i)) \propto \exp\left(-\frac{d(y_i, y_j)}{2\sigma^2}\right) \quad (6)$$

where  $d(y_i, y_j) = \|y_i - y_j\|^2$ . This approach prioritizes mixing similar samples, increasing the mixing probability for closer labels and decreasing it for those with larger differences. Consequently, C-Mixup enhances the model's generalization ability in regression tasks [22].

### III. METHODOLOGY

#### A. Overall Framework

In this work, we propose a comprehensive fault diagnosis framework designed to enhance model performance across continuously varying working conditions. The framework consists of several key components: a CDMix augmentation method, a feature extraction module based on wavelet transforms and a convolutional encoder, a fault classifier, and a soft constraint with CMIM. The overall framework is illustrated in Fig. 1.

We introduce the CDMix augmentation method, which generates out-of-domain samples tailored for continuously indexed domain scenarios, significantly enhancing the model's generalization ability by allowing it to learn from diverse data variations. To ensure that the model can effectively exploit these diverse inputs, we apply wavelet transforms to convert vibration signals into time-frequency representations,

which are well-suited for capturing localized and nonstationary fault features in rotating machinery. These representations are then processed by a convolutional encoder, denoted as  $h(\cdot)$ , where the convolutional architecture is particularly effective in extracting hierarchical spatial patterns from the time–frequency diagrams. Finally, a fault classifier, denoted as  $g(\cdot)$ , identifies fault types based on the extracted features, while a soft constraint based on CMIM is incorporated during training to minimize the mutual information between the features and continuous domain labels. This mechanism encourages the model to focus on extracting domain-invariant characteristics, thereby enhancing its adaptability across various operational conditions. Overall, our framework addresses the pressing challenge of continuous DG, providing a robust solution for real-world fault diagnosis applications.

In the following parts, we will discuss the CDMix augmentation method and the CMIM loss constraint in detail, emphasizing their specific roles and technical implementations in addressing continuously indexed DG.

### B. CDMix Augmentation

The proposed CDMix augmentation is a data augmentation technique specifically designed for continuous DG tasks. While traditional mixing techniques extend to include domain labels as (5), they often lead to inaccurate label generation for continuous domain labels. This issue becomes more serious as the distance between the domain labels of the mixed data increases. To address this challenge, we introduce CDMix, which adjusts the mixing probability based on the distance between data domains, ensuring that the mixing process prioritizes similar samples. By focusing on label proximity, CDMix enhances the accuracy of generated labels, allowing multidata augmentation to yield positive benefits for continuous DG.

1) *Mixing Probability Between Samples*: In mini-batch training, the mixing probability for any given sample  $(x_i, y_i, u_i)$  with another sample  $(x_j, y_j, u_j)$  depends on the Euclidean distance between their continuous working condition labels  $u_i$  and  $u_j$ . Specifically, this probability can be calculated by

$$P_{\text{mix}}(j|i) = \frac{\exp(-\|u_i - u_j\|^2 / (2\sigma^2))}{\sum_{k=0}^n \exp(-\|u_i - u_k\|^2 / (2\sigma^2))} \quad (7)$$

where  $n$  denotes the number of samples in the mini-batch.

We adopt Euclidean distance as the similarity measure because the domain indices  $u$  are explicit, continuous, and low-dimensional representations of working conditions. Compared with distances in the input or feature space, using domain labels has two main advantages: 1) the domain labels directly encode the underlying variation of working conditions, which makes them a natural choice for measuring domain relevance and 2) the domain label space is much lower in dimensionality, making distance calculation more efficient and stable. Based on (7), CDMix prioritizes samples with similar domain labels, thereby increasing the mixing probability for similar samples while reducing the mixing probability for those that differ significantly. This mechanism reduces the occurrence of semantically inconsistent domain labels and improves the reliability of generated labels, which, in turn, enables mixed

data augmentation techniques to more effectively support continuously indexed DG.

2) *Mix Strategy*: CDMix draws on the mixing methods of addition and slicing from Mixup and CutMix, randomly selecting one of these strategies during application. Specifically, the fault-type labels  $\tilde{y}$  and working condition labels  $\tilde{u}$  of the new samples are

$$\begin{aligned} \tilde{y} &= \lambda y_i + (1 - \lambda) y_j \\ \tilde{u} &= \lambda u_i + (1 - \lambda) u_j \end{aligned} \quad (8)$$

where  $\lambda$  is the mixing coefficient sampled from a Beta distribution  $\text{Beta}(\alpha, \alpha)$ , where  $\alpha$  is set to 1, as most mix augmentation methods show low sensitivity to parameter  $\alpha$ . Besides,  $y$  is the one-hot encoded fault type label, while  $u$  is a vector whose elements are continuous values related to the working conditions.

As for the new input data  $\tilde{x}$ , when the add mix is chosen,  $\tilde{x}$  can be obtained by

$$\tilde{x} = \lambda x_i + (1 - \lambda) x_j \quad (9)$$

and when the cut mix is selected,  $\tilde{x}$  is derived by

$$\tilde{x} = M \odot x_i + (1 - M) \odot x_j \quad (10)$$

where the mask  $M \in \{0, 1\}^L$  is a binary vector with the same length as  $x$ . The starting index  $r_x$  and the length  $r_l$  of the segment with elements equal to 1 in  $M$  satisfy

$$r_x \sim \text{Unif}(0, L), \quad r_l = L(1 - \lambda). \quad (11)$$

---

#### Algorithm 1 Continuous Domain Mix Augmentation

---

**Input:** Batch of data  $\{(x_i, y_i, u_i)\}_{i=1}^N$ .

**Output:** Augmented data batch  $\{(\tilde{x}_i, \tilde{y}_i, \tilde{u}_i)\}_{i=1}^N$ .

```

1 for each sample  $(x_i, y_i, u_i)$  in the batch do
2   Compute the mix probability  $P_{\text{mix}}(j|i)$  with each sample  $(x_j, y_j, u_j)$  using (7) and select a sample based on the computed probabilities.
3   Sample a mixing coefficient  $\lambda \sim \text{Beta}(1, 1)$ .
4   Randomly choose a mixing strategy from add mix and cut mix with equal probability.
5   if add mix is chosen then
6     Compute the mixed input data  $\tilde{x}_i$  by (9).
7   else if cut mix is chosen then
8     Compute the mixed input data  $\tilde{x}_i$  by (10).
9   end if
10  Apply (8) to compute the mixed label  $\tilde{y}_i$  and mixed domain label  $\tilde{u}_i$ .
11 end for
12 return: the augmented batch  $\{(\tilde{x}_i, \tilde{y}_i, \tilde{u}_i)\}_{i=1}^N$ 

```

---

The pseudocode for the CDMix augmentation (CDMix) is presented in Algorithm 1. CDMix effectively balances the tradeoff between diversity and continuous domain label accuracy by prioritizing the mixing of samples with similar domain labels. This approach minimizes the risk of generating semantically inconsistent domain labels, thereby enhancing the reliability of the augmented data. The strategic selection of mixing strategies, combined with a probabilistic approach



based on domain proximity, ensures that the augmented data supports more effective generalization across continuously indexed domains.

### C. Continuous Mutual Information Minimization

In continuous DG, extensive theoretical analysis and experimental validation have shown that extracting domain-invariant features can significantly enhance DG performance, which is equally applicable to continuous DG [18], [23].

Mutual information serves as a powerful tool for quantifying the correlation between the distributions of two variables. In the context of training models to extract domain-invariant features, we utilize mutual information to assess the relationship between the feature and the working conditions. To extract domain-invariant features, we introduce the minimization of mutual information into the model training process, aiming to reduce the dependence of the learned features on specific domain characteristics, thereby enhancing the model's robustness and its ability to generalize across varying conditions.

1) *Kernel-Based Mutual Information Estimation*: The definition of mutual information between feature vectors  $h(x)$  and continuous domain labels  $u$  is given by

$$I(h(X), U) = \mathbb{E}_{(h(x), u) \sim P_{h(X), U}} \left[ \log \frac{p(h(x), u)}{p(h(x))p(u)} \right]. \quad (12)$$

Mutual information  $I(h(X), U)$  is always nonnegative, and it achieves a minimum value of zero if and only if  $h(X)$  and  $U$  are independent. This means that when minimizing mutual information, we are effectively promoting the extraction of domain-invariant features, ensuring that the learned representations are robust to variations in working conditions.

Note that the variational formulation of KL divergence between two probability distributions  $P$  and  $Q$ , introduced by Nguyen et al. [24], is

$$\begin{aligned} D_{\text{KL}}(P, Q) &= \mathbb{E}_{x \sim P} \left[ \log \frac{p(x)}{q(x)} \right] \\ &= \sup_f \left( \mathbb{E}_{x \sim P} [f(x)] - \mathbb{E}_{x \sim Q} [e^{f(x)-1}] \right) \end{aligned} \quad (13)$$

and mutual information can be interpreted as the KL divergence between the joint distribution and the product distribution. Specifically, in the context of (12), these distributions are  $P(h(X), U)$  and  $P(h(X))P(U)$ , respectively. Thus, we can reformulate (12) using (13) as

$$\begin{aligned} I(h(X), U) &= \sup_f \left( \mathbb{E}_{(h(x), u) \sim P_{h(X), U}} [f(h(x), u)] \right. \\ &\quad \left. - \mathbb{E}_{(x, u) \sim P_{h(X)}P_U} [e^{f(h(x), u)-1}] \right). \end{aligned} \quad (14)$$

At this point, when calculating the expectation in (14), the probability is represented in a linear form. In the case of mini-batch training, using  $n$  randomly sampled instances, we can estimate (14) as

$$\begin{aligned} \hat{I}(h(X), U) &= \sup_f \left( \frac{1}{n} \sum_{i=1}^n f(h(x_i), u_i) \right. \\ &\quad \left. - \frac{1}{n^2} \sum_{i, j=1}^n e^{f(h(x_i), u_j)-1} \right). \end{aligned} \quad (15)$$

To estimate the function  $f$ , we draw inspiration from kernel methods [25], [26] and utilize a function approximation that leverages multiple Gaussian kernels, represented with linear parameters  $\theta$

$$\begin{aligned} f(h(x), u; \theta) &= \sum_{i=1}^n \left[ \sum_{l_k=1}^m \theta_{i, l_k} K_{x, l_k}(h(x), h(x_i)) \right] K_u(u, u_i) \end{aligned} \quad (16)$$

where  $K$  denotes the Gaussian kernel function defined as

$$K(x, y) = \exp \left( -\frac{\|x - y\|^2}{2\sigma^2} \right) \quad (17)$$

with bandwidth  $\sigma$ . Given that the inputs  $h(x)$  are finite-dimensional vectors with continuous elements, the Gaussian kernel can implicitly and nonlinearly map these vectors into a high-dimensional space based on their similarities, thereby effectively approximating the function  $f$ . To address the complexity of the feature space, we employ  $m$  kernel functions with varying bandwidths for  $h(x)$ . This approach enhances fitting capability and reduces estimation error. The bandwidths for the kernel functions  $K_x$  are adaptively determined using

$$\sigma_x^2 = \frac{\alpha_k}{n^2} \sum_{i, j=1}^n \|h(x_i) - h(x_j)\|^2 \quad (18)$$

where  $\alpha_k \in \{-3, -2, -1, 0, 1\}$ , and these bandwidths  $\sigma_x$  are constants that do not participate in the gradient optimization process. Besides, the bandwidths for  $K_u$ , denoted as  $\sigma_u$ , are predetermined.

By substituting (16) back into (15), the optimal estimate  $\hat{\theta}$  for the parameter  $\theta$  can be obtained by solving the convex optimization problem

$$\begin{aligned} \min_{\theta \in \mathbb{R}^{n, m}} & -\frac{1}{n} \sum_{i=1}^n f(h(x_i), u_i; \theta) \\ & + \frac{1}{n^2} \sum_{i, j=1}^n e^{f(h(x_i), u_j; \theta)-1} + \lambda_\theta \|\theta\|^2 \end{aligned} \quad (19)$$

where  $\lambda_\theta$  is the  $L_2$  regularization coefficient used to mitigate overfitting. Since (19) represents a convex problem, we employ the limited-memory Broyden–Fletcher–Goldfarb–Shanno (L-BFGS) algorithm [27], which is efficient for large-scale optimization, to determine the optimal parameters  $\hat{\theta}$  as outlined in (19). Finally, by substituting  $\hat{\theta}$  into (15), we can estimate the mutual information as

$$\begin{aligned} \hat{I}(h(X), U) &= \frac{1}{n} \sum_{i=1}^n f(h(x_i), u_i; \hat{\theta}) \\ &\quad - \frac{1}{n^2} \sum_{i, j=1}^n e^{f(h(x_i), u_j; \hat{\theta})-1}. \end{aligned} \quad (20)$$

2) *Training With Mutual Information Minimization*: To address continuously indexed DG with CMIM in fault diagnosis scenarios, we focus on training the network to minimize a combination of the cross-entropy loss and the estimated mutual information in (20). The objective function is defined as

$$\min_{g, h} \mathbb{E}_{(x, y) \sim P(X, Y)} [\mathcal{L}(g(h(x)), y)] + \lambda_I \hat{I}(h(X), U) \quad (21)$$

where  $\mathcal{L}$  represents the cross-entropy loss, which quantifies the difference between the predicted fault types and the actual labels. The parameter  $\lambda_l$  acts as a tradeoff factor, balancing the focus on classification accuracy from the training data with the minimization of feature-domain mutual information.

---

**Algorithm 2** Continuous Mutual Information Minimization
 

---

**Input:** Training data  $X$  with fault labels  $Y$  and continuous domain labels  $U$ .

**Output:** Trained CNN encoder  $h(\cdot)$  and classifier  $g(\cdot)$ .

```

1 Randomly initialize  $h(\cdot)$  and  $g(\cdot)$ .
2 while training does not end do
3   for each training step do
4     Sample a batch  $\{(x_i, y_i, u_i)\}_{i=1}^N$  from  $(X, Y, U)$ .
5     Get features  $h(x)$  and prediction  $g(h(x))$ .
6     Calculate the cross-entropy loss  $\mathcal{L}(g(h(x)), y)$ .
7     Obtain  $\hat{\theta}$  by solving the optimization problem
       described in (19) with  $h(x)$ ,  $g(h(x))$ ,  $u$ .
8     Calculate  $\hat{I}$  in (20) with  $\hat{\theta}$ .
9     Get the total loss by (21) with  $\mathcal{L}(g(h(x)), y)$  and  $\hat{I}$ .
10    Backpropagate the gradients in  $g(\cdot)$  and  $h(\cdot)$ .
11    Update parameters in  $h(\cdot)$  and  $g(\cdot)$ .
12  end for
13 end while

```

---

The pseudocode in Algorithm 2 provides a detailed procedural overview of CMIM, highlighting the steps involved in minimizing mutual information to enhance DG. By employing CMIM, we can enhance the model's robustness and generalization capabilities across continuously varying working conditions.

#### IV. CASE STUDY

In this section, we will demonstrate that the proposed continuously indexed DG fault diagnosis algorithm, incorporating CDMix and CMIM, outperforms existing methods through experiments conducted on two datasets. These datasets include the MCC5-THU gearbox fault dataset [28], collected under time-varying speed and load conditions, and the University of Ottawa (UO) bearing dataset [29], collected under time-varying rotational speed conditions. All experiments are conducted under varying working conditions, including different speeds and torques in MCC5-THU, and continuously varying rotational speeds in the UO dataset. This ensures that the evaluation directly reflects the challenge of fault diagnosis under continuously changing domains.

##### A. Experimental Setup

1) *Implementation Details:* The experiments are conducted on a computer equipped with an L40 GPU running the Ubuntu 20.04 operating system, utilizing the PyTorch framework for algorithm implementation.

We employ the wavelet transform with the generalized Morse wavelet [30] to extract the time–frequency representation of the vibration signals, which are then resized to  $224 \times 224$  pixels. For the convolutional encoder  $h(\cdot)$ , we utilize the ResNet18 architecture [31], and the classifier  $g(\cdot)$  adopts a fully connected network structure without hidden layers.

The model is trained using mini-batch training, extracting data from the training set in batches of size 8. We set the number of training epochs to 100, with early stopping patience set to 10 to prevent overfitting. The AdamW optimizer is adopted with a learning rate of  $1e-5$  and a weight decay of  $1e-6$ .

These hyperparameters are determined through extensive grid search and cross-validation experiments to ensure optimal performance. The application probability  $p$  for CDMix is set to 0.3, with a bandwidth  $\sigma$  of 0.1 in (7). For the CMIM module, the tradeoff weight  $\lambda_l$  in the loss function (21) is set to 0.5, and the bandwidth  $\sigma_u$  in (16) for continuous domain labels is 0.05. The sensitivity of these parameters will be analyzed in Section V.

All experiments are repeated 5 times, and the results presented are the averages of these repetitions to ensure robustness and reliability.

2) *Comparison Algorithms:* In this section, several comparison algorithms are introduced to evaluate the performance of the proposed method. CORAL [32], IRM [33], and MixStyle [34] are outstanding algorithms in the DG field, while CGAN-MSE [35], DMDGN [13], and MGF-PGCN [36] represent state-of-the-art methods for cross-condition DG in fault diagnosis. The detailed introduction to these methods is presented as follows.

- 1) CORAL aims to achieve DG by utilizing linear transformations to align the second-order statistics of feature distributions extracted from various domains, thereby enabling the extraction of domain-invariant features.
- 2) IRM estimates nonlinear, invariant, causal predictors from multiple training environments to achieve out-of-distribution generalization, emphasizing the importance of learning invariances across diverse environments.
- 3) MixStyle uniquely probabilistically mixes the output feature statistics from different layers of the model across source domains, creating diverse styles that enhance DG and improve the model's adaptability to unseen environments.
- 4) CGAN-MSE utilizes a single discriminator and a mean-square-error loss for conditional adversarial training to extract domain-invariant features, improving generalization in fault diagnosis tasks.
- 5) DMDGN applies mixup data augmentation to both class and domain spaces, employs adversarial learning to introduce adversarial perturbations, and utilizes a domain-based discrepancy metric to balance intra- and interdomain distances, effectively learning domain-invariant and discriminative features for different working loads and machines.
- 6) MGF-PGCN combines multiscale Gaussian feature enhancement with a prototype graph convolutional network to achieve robust DG in rolling bearing fault diagnosis, ensuring consistent and compact feature representations across domains.

##### B. Case 1: MCC5-THU Gearbox Dataset

1) *Data Description:* The MCC5-THU gearbox dataset is collected from a two-stage parallel gearbox test platform,

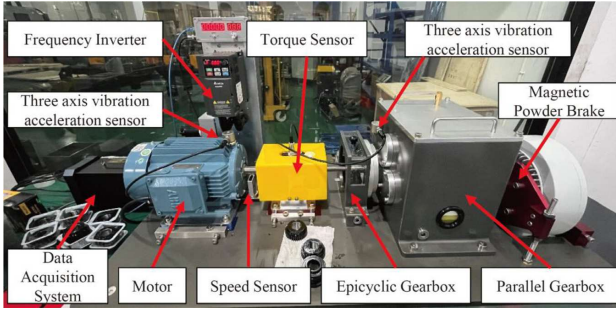


Fig. 2. Gearbox test rig of MCC5-THU [28].

TABLE I  
WORKING CONDITIONS OF EACH MCC5-THU SUBSET

Torque range (Nm)	Speed Range (rpm)		
	500-1000	1500-2000	2500-3000
5-10	Subset 1	Subset 2	Subset 3
15-20	Subset 4	Subset 5	Subset 6

TABLE II  
FAULT TYPES AND DATA COUNT OF MCC5-THU

Label	Fault Type	Data Count
C0	Health	250 per subset
C1	Gear Pitting H	
C2	Gear Wear H	
C3	Missing Teeth	
C4	Teeth Crack H	
C5	Teeth Break H	

equipped with a three-phase asynchronous motor and a magnetic powder brake. The data acquisition system includes two three-axis vibration accelerometers, one positioned at the gearbox output end and the other at the motor end, measuring vibrations along the  $x$ -,  $y$ -, and  $z$ -axes at a sampling frequency of 12.8 kHz for 60 s. Additionally, a torque sensor and a key phase sensor are used to record the gearbox input torque and motor output speed, respectively. The test rig of MCC5-THU is illustrated in Fig. 2.

To create subsets for DG experiments, the data in MCC5-THU is divided into six groups based on three speed ranges and two torque ranges. The details of each subset's working conditions are summarized in Table I.

The selected data in MCC5-THU includes six fault types, focusing on the most severe conditions. Each subset is constructed from the middle 40 s of each data file, segmented into pieces of 4096 sampling points, corresponding to a time length of 320 ms. Table II provides descriptions of the different fault types along with their respective data counts.

For DG fault diagnosis experiments on MCC5-THU, five of the six subsets will be selected as the training set, while the remaining one will serve as the test set. For regular DG methods, the subset indices are used as domain labels, while in our proposed continuously indexed DG method, we directly employ a 2-D working condition vector as continuous domain labels, where the condition values in this vector are normalized to the range of 0–1 using min–max normalization

$$u' = \frac{u - u_{\min}}{u_{\max} - u_{\min}}. \quad (22)$$

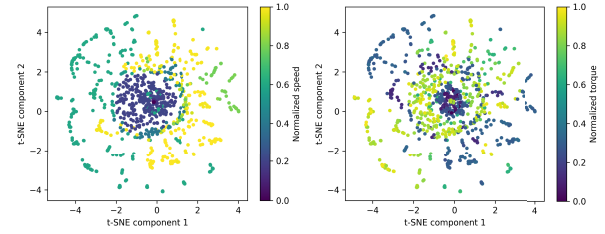


Fig. 3. t-SNE visualization of 1000 samples from the MCC5-THU dataset. (Left) Samples colored by normalized speed. (Right) Samples colored by normalized torque.

2) *Domain Shift Illustration*: To intuitively demonstrate how varying working conditions cause domain shifts, we randomly select 1000 samples from the MCC5-THU dataset and perform a 2-D t-SNE visualization [37]. The samples are colored according to their normalized speed and torque values, as shown in Fig. 3.

The visualization reveals that the feature distributions are not uniform with respect to either speed or torque, indicating that variations in working conditions lead to clear distribution shifts. Consequently, when a model is trained only on data from limited working conditions, the learned decision boundaries may not be well-suited for unseen conditions. This confirms that varying working conditions inherently induce domain shifts, motivating the necessity of continuously indexed DG methods.

3) *Results and Analysis*: The experiments based on the MCC5-THU dataset are divided into two parts: one based on gearbox vibration data and the other based on motor-end vibration data. Each part evaluates the performance of our proposed CDMix–CMIM method in comparison to several classic DG methods. For the six subsets corresponding to the six cross-DG tasks, the accuracy results for each method are presented in Tables III and IV. In the tables, the notation  $\rightarrow$  subset  $n$  indicates that the DG model is tested on subset  $n$  while trained on the other subsets. The highest accuracy for each cross-domain diagnosis task is highlighted in bold to emphasize the best-performing methods.

In Table III, our CDMix–CMIM method demonstrates significant improvements over the base model across almost all tasks, achieving an average accuracy of 88.56%, which represents a notable increase of approximately 8.12% over the base model's average of 80.44%. This consistent gain highlights that the proposed CDMix module enables condition-aware sample generation that reduces semantic inconsistency in mixing, while the CMIM module leverages continuous condition information to enforce smoother domain alignment. Together, these components yield stronger generalization than conventional DG methods.

In contrast, the motor-end vibration data contains more noise, which complicates the detection of gear fault information. Nevertheless, as shown in Table IV, CDMix–CMIM also shows a consistent enhancement over the base model, attaining an average accuracy of 80.14%, which is an increase of about 5.61% compared to the base model's 74.53%. Even in this more challenging setting, the advantage of CDMix–CMIM



TABLE III  
MCC5-THU DG ACCURACY (%) ON GEARBOX VIBRATION

Model	→ Subset 0	→ Subset 1	→ Subset 2	→ Subset 3	→ Subset 4	→ Subset 5	Average
Base	70.07	68.87	91.40	80.20	86.20	85.93	80.44
CORAL	69.47	69.33	91.07	80.27	82.47	87.00	79.94
IRM	72.20	76.13	89.33	77.07	84.20	83.60	80.42
MixStyle	73.67	75.80	93.73	79.33	79.13	88.47	81.69
CGAN-MSE	72.67	77.60	89.00	78.47	83.27	82.80	80.64
DMDGN	74.40	<b>92.53</b>	87.60	82.13	89.53	79.13	84.22
MGF-PGCN	80.75	78.46	88.88	<b>83.32</b>	78.32	89.10	83.14
<b>CDMix-CMIM (ours)</b>	<b>84.40</b>	88.47	<b>95.40</b>	78.73	<b>95.07</b>	<b>89.28</b>	<b>88.56</b>

TABLE IV  
MCC5-THU DG ACCURACY (%) ON MOTOR VIBRATION

Model	→ Subset 0	→ Subset 1	→ Subset 2	→ Subset 3	→ Subset 4	→ Subset 5	Average
Base	75.02	67.22	82.96	72.92	67.71	81.33	74.53
CORAL	<b>78.60</b>	72.60	84.00	73.53	69.73	82.27	76.79
IRM	75.20	69.87	83.33	71.93	70.67	83.00	75.67
MixStyle	74.80	74.47	85.27	67.07	<b>78.53</b>	75.80	75.99
CGAN-MSE	75.47	73.33	83.07	77.87	73.20	83.27	77.70
DMDGN	73.60	75.47	77.93	73.80	76.27	72.33	74.90
MGF-PGCN	74.94	75.47	82.33	<b>82.47</b>	71.73	78.66	77.60
<b>CDMix-CMIM (ours)</b>	76.80	<b>80.27</b>	<b>86.60</b>	73.07	74.27	<b>89.80</b>	<b>80.14</b>

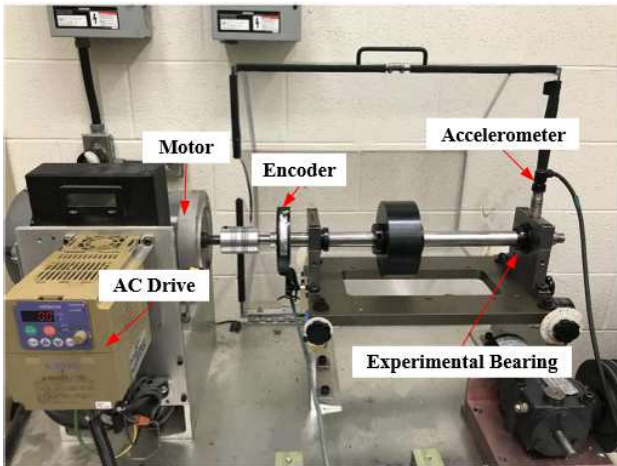


Fig. 4. Experimental platform of UO [29].

confirms that explicitly incorporating condition indices into both sample mixing and feature matching improves robustness to noisy signals.

These results indicate that the proposed CDMix-CMIM method maintains robust performance, despite the increased noise in the motor-end signals. Overall, our method improves accuracy across all tasks and outperforms other established methods on the MCC5-THU dataset, demonstrating its effectiveness in enhancing DG capabilities for both gearbox and motor vibration data.

### C. Case 2: UO Bearing Dataset

1) *Data Description:* The UO Bearing Dataset is collected on a SpectraQuest machinery fault simulator. As illustrated in Fig. 4, the experimental setup includes a drive motor with an ac drive for speed control, an accelerometer to capture

TABLE V  
SPEED RANGE OF EACH UO SUBSET

Index	Subset 1	Subset 2	Subset 3
Speed Range(rpm)	600-1000	1000-1400	1400-1800

vibration signals, and an incremental encoder for measuring the rotational speed.

The UO dataset simulates three fault conditions for bearings, which are inner ring fault, outer ring fault, and ball fault, along with a healthy state, resulting in a total of four fault types. Additionally, the UO dataset is collected under time-varying speeds, ranging from 600 to 1800 r/min. For DG experiments, we divide the dataset into three subsets based on these speed ranges, as shown in Table V.

In preprocessing the UO dataset, the original sampling frequency of 200 kHz is downsampled to 25 kHz. After segmentation, we obtain a total of 2928 data segments, with each fault type represented by 732 segments. Each segment lasts 163.84 ms, which is longer than the maximum fault feature period of 28.01 ms.

Consistent with the experiments on the MCC5-THU dataset, we select two of the UO subsets as training data and use the remaining subset to test the model's generalization capability.

2) *Results and Analysis:* Similarly, we conducted three cross-speed bearing fault diagnosis experiments on the UO dataset, with the results presented in Table VI.

As was expected, the CDMix-CMIM method achieves a remarkable average accuracy of 99.77%, surpassing the baseline model's 99.14%. The baseline model performs well due to the pronounced fault features in bearing systems. However, our method still enhances performance by approximately 0.63%, particularly improving accuracy in the first cross-domain task by 1.52%. This result further validates that CDMix-CMIM is not limited to gear systems but is generally applicable to



TABLE VI  
UO DG ACCURACY (%)

Model	→ Subset 0	→ Subset 1	→ Subset 2	Average
Base	97.97	99.89	99.57	99.14
CORAL	97.51	<b>100.00</b>	99.60	99.04
IRM	98.21	99.91	99.66	99.26
MixStyle	98.38	<b>100.00</b>	99.31	99.23
CGAN-MSE	97.94	<b>100.00</b>	99.60	99.18
DMDGN	98.63	99.94	99.76	99.44
MGF-PGCN	98.27	<b>100.00</b>	<b>99.81</b>	99.36
<b>CDMix-CMIM (ours)</b>	<b>99.49</b>	<b>100.00</b>	<b>99.81</b>	<b>99.77</b>

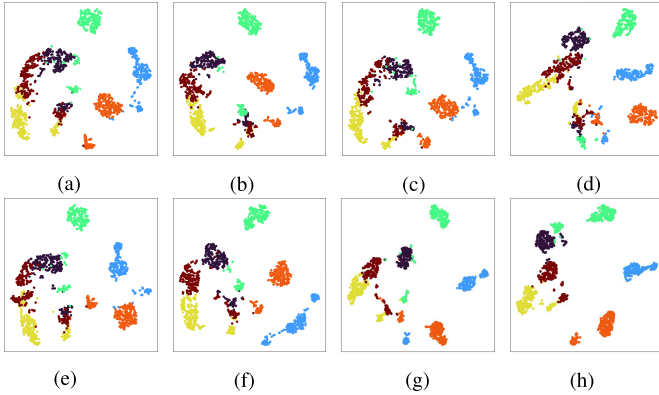


Fig. 5. t-SNE visualization of features from the test set. (a) Base. (b) CORAL. (c) IRM. (d) MixStyle. (e) CGAN-MSE. (f) DMDGN. (g) MGF-PGCN. (h) CDMix-CMIM.

various fault diagnosis tasks, with CMIM effectively exploiting continuous condition indices to maintain cross-condition discriminability.

## V. MODEL DISCUSSION

In this section, we further analyze the proposed method, demonstrating its effectiveness and stability through feature visualization, ablation studies, and parameter sensitivity analysis.

### A. Feature Visualization

To analyze and evaluate the capability of the proposed CDMix and CMIM methods in enhancing the generalization performance of fault diagnosis, we will use t-SNE to visualize the features extracted from the test set. Specifically, we will extract features from models trained using different methods on the gearbox end vibration signals from the MCC5-THU dataset in the first cross-domain task and present the dimensionality reduction results as shown in Fig. 5. Different colors in the figure represent different fault types.

The results indicate that although all fault diagnosis models use the same architecture, the features extracted from the test set show varying degrees of distinguishability based on the DG algorithm. Notably, the model trained using our proposed CDMix-CMIM method exhibits the best distinguishability among fault types. This improvement stems from the combined effect of CDMix, which generates condition-consistent

TABLE VII  
ABLATION STUDY

Method	Description	Accuracy(%)
<b>CDMix-CMIM</b>	Proposed method.	<b>88.56</b>
Mix-CMIM	Replace CDMix with Mixup and CutMix.	83.62
CMIM	Remove CDMix.	86.04
CDMix	Remove CMIM.	83.07

training samples, and CMIM, which aligns features across continuous condition indices. The resulting feature space exhibits clearer class separation in the unseen domains.

### B. Ablation Study

In the ablation study, we systematically evaluate the contributions of different components of the proposed CDMix method. Our experiments are conducted on the gearbox end data from the MCC-THU dataset, where we perform 5 trials to obtain the average accuracy. We first replace the CDMix approach with the vanilla Mix augmentation to assess its impact on performance. Next, we remove the CDMix module and only utilize the CMIM module for DG. Finally, we exclude the CMIM module entirely, which effectively equates to relying solely on the vanilla Mix augmentation. The effectiveness of the method is evaluated using the average accuracy across six cross-domain diagnosis tasks. The results of these different ablation experiments are presented in Table VII.

The results of the ablation experiments are presented in Table VII. The table shows that our proposed CDMix-CMIM method achieves the highest fault diagnosis generalization accuracy among all variants. Specifically, both CDMix alone and CMIM alone yield improvements over the baseline model, and their combination achieves the best overall performance. In contrast, replacing CDMix with conventional mixing techniques such as Mixup or CutMix does not provide the same benefit, as these methods may introduce semantic inconsistency under continuously indexed DG. By maintaining condition-aware semantic consistency, CDMix effectively complements CMIM's domain-invariance enforcement, leading to the strongest generalization.

### C. Sample Imbalance Analysis

To assess the robustness of our proposed method under sample imbalance, we introduced imbalance ratios (IRs) of 1, 2, 3, 4, 5, and 10. Here, IR refers to the ratio of health class data to fault class data, with IR = 1 indicating balanced

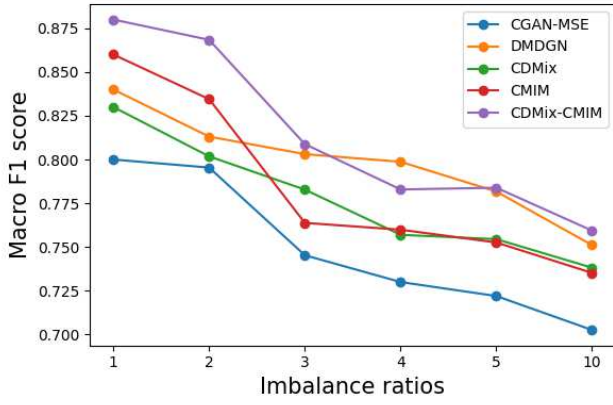


Fig. 6. Macro  $F1$ -scores under different IRs.

data. The experiments are conducted using the gearbox end vibration data from the MCC5-THU dataset, repeating five times to obtain the average results. Considering that accuracy can be misleading in scenarios with sample imbalance, we use the macro  $F1$ -score as the evaluation metric. The macro  $F1$ -score is the mean of the  $F1$ -scores of all classes, providing a more balanced measure of performance across classes. The  $F1$ -score is defined as

$$F1\text{-score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \quad (23)$$

We selected CGAN-MSE and DMDGN methods for comparison with our proposed CDMix-CMIM method. Additionally, we tested CDMix and CMIM individually to analyze the robustness of each module under data imbalance. Fig. 6 illustrates the macro  $F1$ -scores of different methods across varying IRs.

The results demonstrate that as the IR increases, the performance of various DG methods declines to varying extents. However, our CDMix-CMIM method consistently outperforms others. Notably, CMIM maintains excellent DG performance when the imbalance ratio does not exceed 2, but experiences a significant performance drop beyond this point. In contrast, CDMix exhibits greater robustness to changes in imbalance ratio, ensuring that the combined CDMix-CMIM method delivers reliable DG across all imbalance scenarios.

#### D. Parameter Sensitivity Analysis

Here, we analyze the sensitivity of four key hyperparameters, which include the CDMix application probability  $p$ , the bandwidth  $\sigma$ , the CMIM tradeoff weight  $\lambda_I$ , and the bandwidth  $\sigma_u$ . This analysis is conducted using experiments on the gearbox end vibration signals from the MCC5-THU dataset in the first cross-domain task. We utilize a control variable method, fixing three parameters while varying one to evaluate its impact on average accuracy, calculated from five repeated experiments. The experimental results are shown in Fig. 7.

The results show that  $\sigma$  achieves stable performance within 0.01–0.2, with 0.1 being optimal, and the model is generally robust to the choice of  $p$ , with  $p = 0.3$  giving the best accuracy.

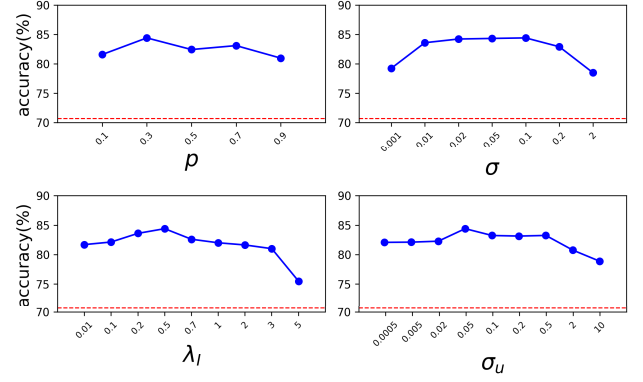


Fig. 7. Sensitivity analysis of hyperparameters.

For the CMIM tradeoff weight  $\lambda_I$ , we observe that it balances feature domain invariance and fault-type discriminability. Too small a value makes CMIM ineffective, while overly large values overemphasize invariance at the expense of classification accuracy. The range 0.2–0.7 yields the appropriate tradeoff, with  $\lambda_I = 0.5$  being optimal.

For the CMIM bandwidth  $\sigma_u$ , very small values make CMIM unable to distinguish domain differences, thus failing to enhance DG, while excessively large values over-discretize the continuous labels, harming DG. A recommended range is 0.02–0.5, with 0.05 as the optimal setting.

Overall, all parameters exhibit a degree of robustness, showing stability in model performance across various configurations.

## VI. CONCLUSION

This article addresses the common challenge of fault diagnosis across continuously varying working conditions by proposing a novel fault diagnosis framework based on continuously indexed DG methods. The framework integrates a CDMix data augmentation method, a feature extraction module based on wavelet transform and a convolutional encoder, a fault feature classification module, and a CMIM loss constraint. Our method generates out-of-domain data suitable for scenarios with continuously indexed domains through CDMix and utilizes CMIM to estimate and minimize the mutual information between features and continuous domain labels, guiding the network to extract domain-invariant features. This enhances the model's generalization capability under different working conditions, thereby improving accuracy in fault classification across continuous domains. Extensive experiments conducted on gearbox and bearing fault datasets demonstrate that the proposed CDMix-CMIM continuously indexed DG framework can fully leverage continuous working condition information, exhibiting significant superiority over existing methods, while the algorithm itself shows good robustness.

Although developed for fault diagnosis, the proposed CDMix-CMIM can also be applied to other fields with similar challenges. Future research may explore other forms of continuously indexed DG methods to enhance DG performance, while also considering the application of these techniques to other common challenges in fault diagnosis, such as sample

imbalance and unknown faults, to improve the practicality of the methods.

## REFERENCES

- [1] J. Cen, Z. Yang, X. Liu, J. Xiong, and H. Chen, "A review of data-driven machinery fault diagnosis using machine learning algorithms," *J. Vibrat. Eng. Technol.*, vol. 10, no. 7, pp. 2481–2507, Oct. 2022.
- [2] R. Vaish, U. D. Dwivedi, S. Tewari, and S. M. Tripathi, "Machine learning applications in power system fault diagnosis: Research advancements and perspectives," *Eng. Appl. Artif. Intell.*, vol. 106, Nov. 2021, Art. no. 104504.
- [3] Z. Zhu et al., "A review of the application of deep learning in intelligent fault diagnosis of rotating machinery," *Measurement*, vol. 206, Jan. 2023, Art. no. 112346.
- [4] K. Liu, Y. Li, Z. Cui, G. Qi, and B. Wang, "Adaptive frequency attention-based interpretable transformer network for few-shot fault diagnosis of rolling bearings," *Rel. Eng. Syst. Saf.*, vol. 263, Nov. 2025, Art. no. 111271.
- [5] Y. Li, K. Feng, Y. Chen, and Z. Chen, "Multioperator morphological undecimated wavelet for wheelset bearing compound fault detection," *IEEE Trans. Instrum. Meas.*, vol. 72, pp. 1–12, 2023.
- [6] X. Chen, R. Yang, Y. Xue, M. Huang, R. Ferrero, and Z. Wang, "Deep transfer learning for bearing fault diagnosis: A systematic review since 2016," *IEEE Trans. Instrum. Meas.*, vol. 72, pp. 1–21, 2023.
- [7] O. Matania, I. Dattner, J. Bortman, R. S. Kenett, and Y. Parmet, "A systematic literature review of deep learning for vibration-based fault diagnosis of critical rotating machinery: Limitations and challenges," *J. Sound Vibrat.*, vol. 590, Nov. 2024, Art. no. 118562.
- [8] J. Zhong, C. Lin, Y. Gao, J. Zhong, and S. Zhong, "Fault diagnosis of rolling bearings under variable conditions based on unsupervised domain adaptation method," *Mech. Syst. Signal Process.*, vol. 215, Jun. 2024, Art. no. 111430.
- [9] Y. Qin, Q. Qian, J. Luo, and H. Pu, "Deep joint distribution alignment: A novel enhanced-domain adaptation mechanism for fault transfer diagnosis," *IEEE Trans. Cybern.*, vol. 53, no. 5, pp. 3128–3138, May 2023.
- [10] C. Zhao, E. Zio, and W. Shen, "Domain generalization for cross-domain fault diagnosis: An application-oriented perspective and a benchmark study," *Rel. Eng. Syst. Saf.*, vol. 245, May 2024, Art. no. 109964.
- [11] H. Pu, S. Teng, D. Xiao, L. Xu, J. Luo, and Y. Qin, "Domain generalization for machine compound fault diagnosis by domain-relevant joint distribution alignment," *Adv. Eng. Informat.*, vol. 62, Oct. 2024, Art. no. 102771.
- [12] L. Jia, T. W. S. Chow, and Y. Yuan, "Causal disentanglement domain generalization for time-series signal fault diagnosis," *Neural Netw.*, vol. 172, Apr. 2024, Art. no. 106099.
- [13] Z. Fan, Q. Xu, C. Jiang, and S. X. Ding, "Deep mixed domain generalization network for intelligent fault diagnosis under unseen conditions," *IEEE Trans. Ind. Electron.*, vol. 71, no. 1, pp. 965–974, Jan. 2024.
- [14] L. Ren, T. Mo, and X. Cheng, "Meta-learning based domain generalization framework for fault diagnosis with gradient aligning and semantic matching," *IEEE Trans. Ind. Informat.*, vol. 20, no. 1, pp. 754–764, Jan. 2024.
- [15] S. Sagawa, P. Wei Koh, T. B. Hashimoto, and P. Liang, "Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization," 2019, *arXiv:1911.08731*.
- [16] B. Yang, Y. Lei, X. Li, N. Li, X. Si, and C. Chen, "A dynamic barycenter bridging network for federated transfer fault diagnosis in machine groups," *Mech. Syst. Signal Process.*, vol. 230, May 2025, Art. no. 112605.
- [17] B. Yang, Y. Lei, N. Li, X. Li, X. Si, and C. Chen, "Balance recovery and collaborative adaptation approach for federated fault diagnosis of inconsistent machine groups," *Knowledge-Based Syst.*, vol. 317, May 2025, Art. no. 113480.
- [18] J. Wang et al., "Generalizing to unseen domains: A survey on domain generalization," *IEEE Trans. Knowl. Data Eng.*, pp. 1–1, May 2022.
- [19] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "Mixup: Beyond empirical risk minimization," 2017, *arXiv:1710.09412*.
- [20] S. Yun, D. Han, S. Chun, S. J. Oh, Y. Yoo, and J. Choe, "CutMix: Regularization strategy to train strong classifiers with localizable features," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6022–6031.
- [21] M. Xu et al., "Adversarial domain adaptation with domain mixup," in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, no. 4, pp. 6502–6509.
- [22] H. Yao, Y. Wang, L. Zhang, J. Zou, and C. Finn, "C-mixup: Improving generalization in regression," in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, pp. 3361–3376.
- [23] I. Albuquerque, J. Monteiro, M. J. D. Bayazi, T. H. Falk, and I. Mitliagkas, "Adversarial target-invariant representation learning for domain generalization," *CoRR*, Jan. 2019.
- [24] X. Nguyen, M. J. Wainwright, and M. I. Jordan, "Estimating divergence functionals and the likelihood ratio by convex risk minimization," *IEEE Trans. Inf. Theory*, vol. 56, no. 11, pp. 5847–5861, Nov. 2010.
- [25] S. Chen, "Multi-source domain adaptation with mixture of joint distributions," *Pattern Recognit.*, vol. 149, May 2024, Art. no. 110295.
- [26] L. Wen, S. Chen, M. Xie, C. Liu, and L. Zheng, "Training multi-source domain adaptation network by mutual information estimation and minimization," *Neural Netw.*, vol. 171, pp. 353–361, Mar. 2024.
- [27] D. C. Liu and J. Nocedal, "On the limited memory BFGS method for large scale optimization," *Math. Program.*, vol. 45, nos. 1–3, pp. 503–528, Aug. 1989.
- [28] S. Chen, Z. Liu, X. He, D. Zou, and D. Zhou, "Multi-mode fault diagnosis datasets of gearbox under variable working conditions," *Data Brief*, vol. 54, Jun. 2024, Art. no. 110453.
- [29] H. Huang and N. Baddour, "Bearing vibration data collected under time-varying rotational speed conditions," *Data Brief*, vol. 21, pp. 1745–1749, Dec. 2018.
- [30] J. M. Lilly and S. C. Olhede, "Generalized Morse wavelets as a superfamily of analytic wavelets," *IEEE Trans. Signal Process.*, vol. 60, no. 11, pp. 6036–6041, Nov. 2012.
- [31] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
- [32] B. Sun and K. Saenko, "Deep CORAL: Correlation alignment for deep domain adaptation," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 443–450.
- [33] M. Arjovsky, L. Bottou, I. Gulrajani, and D. Lopez-Paz, "Invariant risk minimization," 2019, *arXiv:1907.02893*.
- [34] K. Zhou, Y. Yang, Y. Qiao, and T. Xiang, "Domain generalization with MixStyle," 2021, *arXiv:2104.02008*.
- [35] Q. Zhang et al., "Conditional adversarial domain generalization with a single discriminator for bearing fault diagnosis," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–15, 2021.
- [36] X. Cong, Y. Zhuang, Y. Li, D. Wang, Y. Zhang, and Y. Song, "Multi-scale Gaussian feature enhancement and prototype graph convolutional network for domain-generalized rolling bearing fault diagnosis," *Expert Syst. Appl.*, vol. 296, Jan. 2026, Art. no. 129218.
- [37] L. V. D. Maaten and G. E. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. 86, pp. 2579–2605, 2008.