

## Article

# Deep Domain Adaptation with Correlation Alignment and Supervised Contrastive Learning for Intelligent Fault Diagnosis in Bearings and Gears of Rotating Machinery

Bo Zhang <sup>1,\*</sup> , Hai Dong <sup>1</sup> , Hamzah A. A. M. Qaid <sup>2</sup>  and Yong Wang <sup>2</sup>

<sup>1</sup> School of Computer Science and Technology, China University of Mining and Technology, Xuzhou 221116, China; ts21170062p31@cumt.edu.cn

<sup>2</sup> School of Mechanical and Electrical Engineering, China University of Mining and Technology, Xuzhou 221116, China; fb23050001e@cumt.edu.cn (H.A.A.M.Q.); lb21050025@cumt.edu.cn (Y.W.)

\* Correspondence: zbcumt@cumt.edu.cn

**Abstract:** Deep domain adaptation techniques have recently been the subject of much research in machinery fault diagnosis. However, most of the work has been focused on domain alignment, aiming to learn cross-domain features by bridging the gap between source and target domains. Despite the success of these methods in achieving domain alignment, they often overlook the class discrepancy present in cross-domain scenarios. This can result in the misclassification of target domain samples that are located near cluster boundaries or far from their associated class centers. To tackle these challenges, a novel approach called deep domain adaptation with correlation alignment and supervised contrastive learning (DCASCL) is proposed, which synchronously realizes both domain distribution alignment and class distribution alignment. Specifically, the correlation alignment loss is used to enforce the model to generate transferable features, facilitating effective domain distribution alignment. Additionally, classifier discrepancy loss and supervised contrastive learning loss are integrated to carry out feature distribution alignment class-wisely. The supervised contrastive learning loss leverages class-specific information of source and target samples, which efficiently promotes the compactness of samples of the same class and the separation of samples from different classes. Moreover, our approach is extensively validated across three diverse datasets, demonstrating its effectiveness in diagnosing machinery faults across different domains.

**Keywords:** domain adaptation; intelligent fault diagnosis; correlation alignment; supervised contrastive learning; rotating machinery



**Citation:** Zhang, B.; Dong, H.; Qaid, H.A.A.M.; Wang, Y. Deep Domain Adaptation with Correlation Alignment and Supervised Contrastive Learning for Intelligent Fault Diagnosis in Bearings and Gears of Rotating Machinery. *Actuators* **2024**, *13*, 93. <https://doi.org/10.3390/act13030093>

Academic Editor: Giorgio Olmi

Received: 15 January 2024

Revised: 24 February 2024

Accepted: 26 February 2024

Published: 27 February 2024



**Copyright:** © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Bearings and gears are two essential components in rotating mechanical equipment, widely utilized in various fields such as automobiles, aircraft engines, and wind turbines. However, due to their operation under harsh conditions, such as high speeds and heavy loads, these components are prone to failure, significantly impacting the performance and reliability of the equipment. Therefore, achieving rapid and accurate intelligent fault diagnosis for these components is crucial [1]. Some early fault diagnosis techniques, such as signal analysis [2,3] and machine learning [4,5], have been extensively employed in machinery defect diagnosis. Nevertheless, these methods require a certain level of expertise and manual feature extraction, limiting their widespread application. With the increase in computational resources and advancements in big data and sensing technologies, deep learning-based intelligent fault diagnosis has been increasingly investigated. This is attributed to their capability for end-to-end learning, reducing the necessity for extensive human involvement in model development. Janssens et al. [6] established a feature learning-based approach using a CNN to detect different types of bearing defects. This approach was also compared with traditional machine learning-based methods such as support

vector machine (SVM) and random forest (RF) using the same data. The results significantly demonstrated the superiority of end-to-end learning methods over traditional ones. Wang et al. [7] employed a CNN model to learn the discriminative features for gearbox fault diagnosis from time-frequency graphs. Shao et al. [8] introduced a new deep autoencoder method that is effective and robust for feature learning and has been successfully applied to identify bearing or gearbox failures. While these methods have demonstrated potential effectiveness in practice, they also have several considerable limitations. First, they rely on supervised learning, requiring large amounts of training data with different labels, including normal states and various fault situations. However, in a real industrial environment, obtaining data with a fault label is very difficult because the frequency of failures is usually very low. In addition, labeling failure data is exhausting and time-intensive [9]. Therefore, supervised learning methods are limited in practical industrial applications. Second, these methods often assume that the feature distributions of the training and test samples are identical or comparable [10]. Nevertheless, operational circumstances, deteriorated states, and background noise levels of industrial machinery are frequently inconsistent in real-world industrial settings. This implies the presence of a mismatch between the feature distributions of training and test samples, resulting in a notable decline in the performance of fault diagnosis based on deep learning methodologies. As a consequence of the previously outlined factors, the application of these deep learning methods becomes challenging when confronted with data exhibiting distribution discrepancies.

Domain adaptation (DA) emerged as a viable technique for addressing variations in feature distributions across diverse domains. Generally speaking, it is a transfer learning technique that concentrates on utilizing the knowledge learned from a source domain (where labeled data are abundant) to improve the model performance in a target domain (where labeled data are scarce or nonexistent). The primary objective here is to make the gap between the feature representations or distributions of source and target samples as small as possible, enabling the model to generalize and generate accurate predictions on the target samples. In recent years, several methods have been developed based on DA to realize the identification of machine failures across different domains [11–13]. Most of the existing methodologies can generally be divided according to the mechanisms of minimizing the discrepancy between domains into two categories, including discrepancy-based techniques and adversarial-based techniques. Discrepancy-based techniques aim to reduce the gap between different domains by minimizing some defined statistical discrepancy metrics. Representative examples of such techniques include maximum mean discrepancy (MMD) [14], correlation alignment (CORAL) loss [15,16], central moment discrepancy (CMD) [17], and so on. For instance, researchers in prior works [18–20] utilized MMD as part of their objective function, aiming to minimize the distribution discrepancies between the source and target domains for fault diagnosis. Che et al. [21] computed the multiple kernel maximum mean discrepancy (MK-MMD) across selected hidden layers and integrated it into the loss function. This integration aimed to improve the effectiveness of domain adaptation in the context of bearing fault diagnosis. But in methods based on MK-MMD, it is imperative to manually choose an appropriate kernel or multiple kernels within the Reproducing Kernel Hilbert Space (RKHS). In order to reduce the manual interference, Qian et al. [22] combined the CORAL loss with an adversarial mechanism, thereby reducing the distribution discrepancy between the two domains. On the other hand, adversarial-based techniques [23–25] introduce an additional classifier to predict the domain label of the input samples called domain discriminator and force the feature extractor to confuse it during the training process. This strategy allows for the extraction of features that are domain-invariant. Li et al. [26] proposed an adversarial training scheme to reduce the gap between cross-sensor data in the feature space, thereby achieving cross-domain fault diagnosis. Chen et al. [27] attempted to address the large domain shift problem in cross-domain fault diagnosis scenarios by employing a domain adversarial transfer network. However, although these methods have demonstrated encouraging results in domain adaptation, these domain alignment methods can only reduce domain shift rather than

completely eliminate it. Consequently, target samples located near cluster boundaries or far from their respective class centers are more prone to misclassification due to the learned hyperplane from the source domain [28]. In other words, these methods focused solely on acquiring shared feature representations by minimizing distribution discrepancies among various domains but failed to preserve class distinctive features, which can lead to the misclassification of the target samples distributed near class decision boundaries [29].

In an attempt to leverage class-specific decision boundaries, Saito et al. [30] introduced a new method called maximum classifier discrepancy (MCD), which is relevant to our approach. They utilized a feature extractor and two differently initialized classifiers in a mini-max game during domain adaptation. The approach focuses on maximizing the prediction discrepancy of unlabeled target domain samples, thereby facilitating the detection of ambiguous target samples falling outside the support of the source domain. Simultaneously, it minimizes this discrepancy when optimizing the feature extractor to generate target features that reside within the source feature regions. However, MCD exclusively addresses class-level alignment but overlooks global domain alignment. Such an approach highly depends on the accuracy of the domain source classifier, leading to significant performance degradation, especially when the gap between domains is substantial [31]. Furthermore, MCD does not incorporate class label information when minimizing classifier discrepancies, potentially causing some target samples to be assigned to incorrect category.

To address the aforementioned challenges, a deep domain adaptation with correlation alignment and supervised contrastive learning (DCASCL) method is developed in this work, which carries out feature distribution alignment on both global domain and class-level scales. In particular, global domain alignment is achieved through correlation alignment loss. By minimizing this loss, the model is trained to generate generalizable feature representations that are robust to domain shift. Additionally, to align the feature distributions on the class-level scale, a cross-domain supervised contrastive learning loss is combined with the classifier discrepancy loss. This allows the model to utilize class-label information and learn discriminative features with improved intra-class compactness and inter-class separability. The following is a brief summary of our work's main contributions.

1. We propose DCASCL, a novel domain adaptation (DA) framework applied to fault diagnosis of mechanical machinery. DCASCL simultaneously considers domain distribution alignment and class distribution alignment. We experimentally validate that these two aspects complement each other.
2. The correlation alignment is used to realize the domain distribution alignment by minimizing the difference between the covariance matrices of the source and target domain features. The supervised contrastive learning loss is combined with classifier discrepancy loss to align the feature distributions class-wisely. Unlike other methods, DCASCL utilizes class label information through the supervised contrastive learning loss term, which makes it possible to align the features of samples of the same class more tightly while pushing apart those of dissimilar classes.
3. Three different datasets with distinct transfer tasks are employed to validate the feasibility of DCASCL. Furthermore, extensive comparison experiments are carried out to demonstrate the effectiveness of DCASCL over several popular cross-domain diagnostic methods.

## 2. Methods

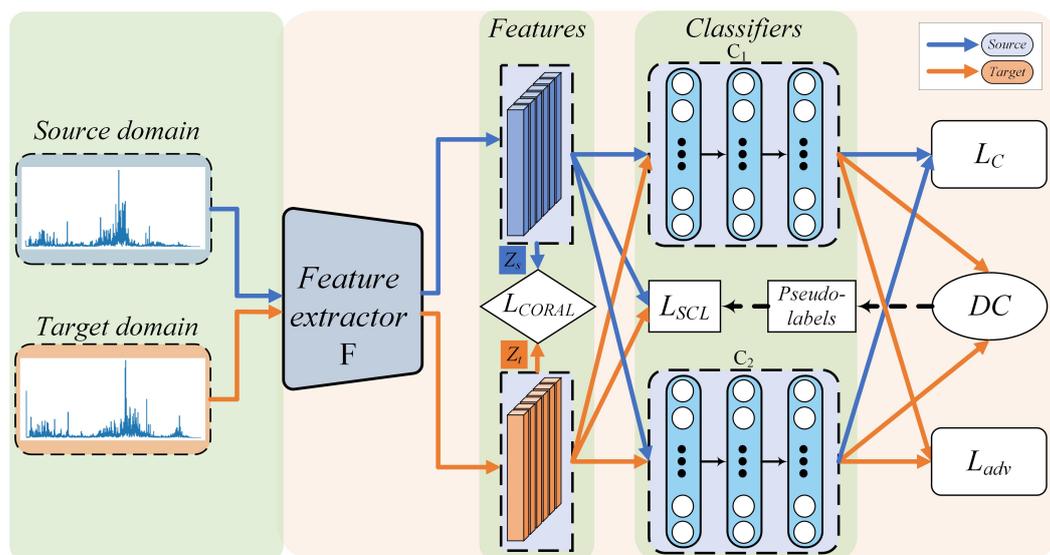
### 2.1. Problem Description

Let a source domain  $\mathcal{D}_s$  with  $n_s$  labeled fault samples be represented as  $\mathcal{D}_s = \{(x_i^s, y_i^s)\}_{i=1}^{n_s}$ , where  $x_i^s$  and  $y_i^s$  denote the  $i$ -th source domain fault sample and its corresponding fault label, respectively. Similarly, let a target domain  $\mathcal{D}_t$  with  $n_t$  unlabeled fault samples be represented as  $\mathcal{D}_t = \{(x_i^t)\}_{i=1}^{n_t}$ , where  $x_i^t$  is the  $i$ -th target domain fault sample. It is assumed that the source domain samples  $X^s$  and target domain samples  $X^t$  share a common feature space but have distinct distributions  $\mathcal{P}(X^s) \neq \mathcal{P}(X^t)$ . This implies that the source and target domains exhibit divergent marginal distributions, resulting in a domain shift. Therefore, our goal is

to develop a novel strategy to effectively transfer knowledge from the source to the target domain, allowing the model to reliably identify the target domain samples.

## 2.2. Model Structure

DCASCL comprises three components, as illustrated in Figure 1: a feature extractor  $F$  is employed to acquire high-dimensional feature vectors, and two classifiers  $C_1$  and  $C_2$  for label prediction. To mitigate the impact of high-frequency noise on the classification accuracy of bearing signals, the structure of the feature extractor  $F$  in DCASCL is similar to the WDCNN method proposed by Zhang et al. [32], which implemented a large kernel in the first convolutional layer. The feature extractor  $F$  is composed of five 1-D convolutional blocks and two dense blocks. Each convolutional block consists of a convolutional layer, ReLU activation, batch normalization (BN), and max pooling. The first convolutional layer utilizes larger convolutional kernels ( $64 \times 1$ ), while the subsequent layers utilize smaller convolutional kernels ( $3 \times 1$ ). Each dense block consists of a fully connected layer, a ReLU layer, and a BN layer. The classifiers  $C_1$  and  $C_2$  consist of two dense blocks and a fully connected layer, and the structure of the dense blocks is consistent with that of the feature extractor. To enhance the network's capacity for generalization and avoid overfitting, dropout layers are included following every dense block. The final layer is a fully connected layer that maps the model's output to the probability distribution of each corresponding category through the softmax activation function. The detailed model design is provided in Table 1.



**Figure 1.** The model architecture diagram of DCASCL. The source domain samples  $X^s$  and the target domain samples  $X^t$  act as inputs to the feature extractor  $F$  after FFT processing, ultimately obtaining high-dimensional features  $Z_s$  and  $Z_t$ . The  $L_{CORAL}$  is employed to measure the difference in the covariance matrices between  $Z_s$  and  $Z_t$ . Pseudo-labels for target domain samples are acquired using the double-confirmation (DC) strategy and are subsequently utilized for supervised contrastive learning ( $L_{SCL}$ ) alongside labeled source domain samples. The  $L_{adv}$  represents the classifier discrepancy and is computed from the probability outputs of two classifiers,  $C_1(F(X^t))$  and  $C_2(F(X^t))$ . The  $L_C$  is a classification loss.

**Table 1.** The detailed parameters of DCASCL.

Module Name	Block Name	Layer Type	In/Out Channel	Kernel Size/Stride	Activation Function
Feature extractor	Conv1	Convolutional	1/16	64/4	ReLU
		BatchNorm	16	/	/
		Max Pooling	/	2/1	/
	Conv2	Convolutional	16/32	3/2	ReLU
		BatchNorm	32	/	/
		Max Pooling	/	2/1	/
	Conv3	Convolutional	32/64	3/2	ReLU
		BatchNorm	64	/	/
		Max Pooling	/	2/1	/
	Conv4	Convolutional	64/64	3/2	ReLU
BatchNorm		64	/	/	
Max Pooling		/	2/1	/	
Conv5	Convolutional	64/64	3/1	ReLU	
	BatchNorm	64	/	/	
	Max Pooling	/	2/1	/	
Dense1	Linear	$64 \times 56/2048$	/	ReLU	
	BatchNorm	2048	/	/	
Dense2	Linear	2048/1024	/	ReLU	
	BatchNorm	1024	/	/	
Classifier	/	Linear	1024/512	/	ReLU
		BatchNorm	512	/	/
		Dropout	/	/	/
	/	Linear	512/256	/	ReLU
		BatchNorm	256	/	/
		Dropout	/	/	/
/	Linear	256/num classes	/	Softmax	

### 2.3. Optimization Objectives of DCASCL

**Classification Loss Term:** To enable the model to learn discriminative features of the input fault samples, we first utilize the source domain to train the whole model, including the feature extractor  $F$  and the classifiers  $C_1$  and  $C_2$ . This training phase aims to minimize the classification loss of both classifiers. The classification loss used is the cross-entropy, which is computed for both classifiers and integrated as follows:

$$\mathcal{L}_C(X^s, Y^s) = -\mathbb{E}_{(x^s, y^s) \sim (X^s, Y^s)} \left[ \sum_{k=1}^K 1_{[k=y^s]} \log p_1(y|x^s) + \sum_{k=1}^K 1_{[k=y^s]} \log p_2(y|x^s) \right] \quad (1)$$

where  $p_1(y|x^s)$  and  $p_2(y|x^s)$  represent the probabilistic output from the classifiers  $C_1$  and  $C_2$ , respectively.  $k$  stands for the fault label.

During the network training, the primary objective is to seek the optimal  $\hat{\theta}_F$ ,  $\hat{\theta}_{C_1}$ , and  $\hat{\theta}_{C_2}$  by reducing  $\mathcal{L}_C(X^s, Y^s)$ . The entire process can be represented as

$$(\hat{\theta}_F, \hat{\theta}_{C_1}, \hat{\theta}_{C_2}) = \arg \min_{\theta_F, \theta_{C_1}, \theta_{C_2}} \mathcal{L}_c(X^s, Y^s) \quad (2)$$

**Correlation Alignment Loss Term:** The correlation alignment (CORAL) loss is a statistical matching-based domain adaptation strategy. Its primary objective is to align the feature distributions of the source and target domains by minimizing the difference in the covariance matrices in the feature space. This aids in extracting domain-invariant

feature representations that are robust to the domain shift. The CORAL loss is calculated as follows:

$$\mathcal{L}_{CORAL} = \frac{1}{4d^2} \|C_S - C_T\|_F^2 \quad (3)$$

where  $d$  stands for the feature space's dimension,  $\|\cdot\|_F^2$  denotes the squared matrix Frobenius norm, and  $C_S$  and  $C_T$  represent the covariance matrices of the features of the source and target domains, respectively. The covariance matrices can be expressed as

$$C_S = \frac{1}{N_s - 1} \left( \mathbf{Z}_s^T \mathbf{Z}_s - \frac{1}{N_s} \left( \mathbf{1}^T \mathbf{Z}_s \right)^T \left( \mathbf{1}^T \mathbf{Z}_s \right) \right) \quad (4)$$

$$C_T = \frac{1}{N_t - 1} \left( \mathbf{Z}_t^T \mathbf{Z}_t - \frac{1}{N_t} \left( \mathbf{1}^T \mathbf{Z}_t \right)^T \left( \mathbf{1}^T \mathbf{Z}_t \right) \right) \quad (5)$$

in which  $\mathbf{Z}_s$  and  $\mathbf{Z}_t$  represent domain features of the source and target domains obtained with batch size  $N_s$  and  $N_t$ , respectively, and  $\mathbf{1}^T$  is a vector wherein all elements equal 1.

In this study, the training process of the network aims to minimize the  $\mathcal{L}_{CORAL}$  between the features extracted from the source and target domains. The optimization of parameters  $\hat{\theta}_F$  is performed to achieve global domain alignment. The process of training  $\hat{\theta}_F$  as follows:

$$(\hat{\theta}_F) = \arg \min_{\theta_F} \mathcal{L}_{CORAL} \quad (6)$$

**Discrepancy Loss Term:** Due to the presence of domain shift, global domain alignment can only alleviate it but not entirely remove it. This results in target samples near the class boundaries being prone to mis-classification by the classifier. Therefore, it is imperative to implement class alignment as well. In this work, class alignment is realized by adopting the discrepancy loss between the classifiers  $C_1$  and  $C_2$ . Since the classifiers are initialized differently, their predictions for the target samples near class boundaries that fall outside the support of the source domain are inconsistent. By intentionally maximizing this inconsistency in the predictions between  $C_1$  and  $C_2$ , the model can effectively identify the target samples situated near class boundaries. We can measure the difference between the predictions of the two classifiers on the target domain samples as follows:

$$dis(p_1(y|x^t), p_2(y|x^t)) = \frac{1}{K} \sum_{k=1}^K |p_{1_k}(y|x^t) - p_{2_k}(y|x^t)| \quad (7)$$

where  $p_{1_k}(y|x^t)$  and  $p_{2_k}(y|x^t)$  represent the probability outputs of  $p_1(y|x^t)$  and  $p_2(y|x^t)$  for class  $k$ , respectively; and  $|\cdot|$  denotes the  $l_1$  norm.

In addition, the cross-entropy loss  $\mathcal{L}_C(X^s, Y^s)$  is added to ensure that the source domain samples are correctly classified while maximizing the classifier discrepancy loss. Finally, the maximization of the classifier discrepancy loss can be achieved by the following formula:

$$(\hat{\theta}_{C_1}, \hat{\theta}_{C_2}) = \arg \min_{\theta_{C_1}, \theta_{C_2}} \{ \mathcal{L}_C(X^s, Y^s) - \mathcal{L}_{adv}(X^t) \} \quad (8)$$

$$\mathcal{L}_{adv} = \mathbb{E}_{x^t \sim X^t} [dis(p_1(y|x^t), p_2(y|x^t))] \quad (9)$$

To align the feature distributions of the source and target domains class-wisely and encourage the features of the target samples to be generated under the influence of the source domain, we also need to minimize the discrepancy between classifiers. In this

case, we need to keep the two classifiers fixed and update the feature extractor using the following formula:

$$(\hat{\theta}_F) = \arg \min_{\theta_F} \mathcal{L}_{adv}(X^t) \quad (10)$$

**Supervised Contrastive Learning Loss Term:** In the process of minimizing classifier discrepancy, the class label information is neglected. In order to utilize this information effectively, drawing inspiration from supervised contrastive learning (SCL) [33], we introduce a novel cross-domain supervised contrastive learning loss aimed at learning representations with both intra-class compactness and inter-class separability. Applying this loss, the model is trained to project samples of the identical category nearer in the feature space while pushing apart the samples of distinct categories, whether these samples originate from the source or target domain. We regard the  $\ell_2$ -normalized features  $z_i^t$  obtained from the  $i$ -th sample  $x_i^t$  in the target domain as an anchor. It forms a positive pair with a sample from the source domain belonging to the identical category, denoted as  $z_p^s$ . The cross-domain supervised contrastive learning loss is then defined as follows:

$$\mathcal{L}_{SCL}^{t,s} = \sum_{i=1}^N \frac{-1}{|\mathcal{P}_s(\hat{y}_i^t)|} \sum_{p \in \mathcal{P}_s(\hat{y}_i^t)} \log \frac{\exp(z_i^{tT} \cdot z_p^s / \tau)}{\sum_{j \in I_t} \exp(z_i^{tT} \cdot z_j^s / \tau)} \quad (11)$$

where the  $\cdot$  symbol denotes the inner dot product;  $\mathcal{P}_s(\hat{y}_i^t) = \{k | y_k^s = \hat{y}_i^t\}$  is the set of positive samples in the source domain that share the identical category as the target domain anchor  $x_i^t$ ,  $\hat{y}_i^t$  is the pseudo-label of the target domain sample  $x_i^t$ .  $I_t$  represents the set of source samples in a batchsize;  $\tau$  is a temperature hyper-parameter. Additionally, we can calculate  $\mathcal{L}_{SCL}^{s,t}$  by using the features of source domain samples as anchor, where  $\mathcal{P}_t(\hat{y}_i^s) = \{k | \hat{y}_k^t = \hat{y}_i^s\}$  represents the set of positive samples in the target domain with the identical category as the anchor samples from the source domain. Finally, combining  $\mathcal{L}_{SCL}^{t,s}$  and  $\mathcal{L}_{SCL}^{s,t}$ , the ultimate cross-domain supervised contrastive loss can be expressed as follows:

$$\mathcal{L}_{SCL} = \mathcal{L}_{SCL}^{t,s} + \mathcal{L}_{SCL}^{s,t} \quad (12)$$

Although we cannot directly obtain the true labels of target domain samples, we can acquire a set of pseudo-labels through the predictions made by classifiers. A common practice is to use the highest prediction probability of the classifier as a pseudo-label for each sample [34]. However, not all pseudo-labels are accurate, as there may be some misleading ones that could misguide the model during training. To ensure the generation of high-quality pseudo-labels, we propose a strategy named double confirmation (DC), comprising two steps. Firstly, employing a pre-defined fixed threshold  $T_{thre}$  to filter out low-confidence samples predicted by the classifiers. Subsequently, we select samples for which pseudo-label predictions are consistent between two classifiers. Specifically, we use a fixed threshold of 0.95 to filter and obtain two sets, denoted as  $\hat{Y}_1$  (predicted by  $C_1$ ) and  $\hat{Y}_2$  (predicted by  $C_2$ ), which consist of samples with predicted probabilities greater than or equal to the threshold. We then extract samples from these sets where both classifiers  $C_1$  and  $C_2$  predict consistent pseudo-labels. These consistent samples are utilized to create a new dataset, denoted as  $\hat{Y}_t$ , which will be used in the previously mentioned supervised contrastive learning process.  $\hat{Y}_t$  is expressed as follows:

$$\hat{Y}_t = \{ \{ \hat{Y}_1 = \{ \max(P_1(y|x^t)) \geq T_{thre} \} \} \cap \{ \hat{Y}_2 = \{ \max(P_2(y|x^t)) \geq T_{thre} \} \} \} \quad (13)$$

Minimizing the supervised contrastive learning loss helps bring closer the distances between samples of the identical category originating from different domains or the same domain, while simultaneously increasing the distances between samples of distinct cate-

gories in the feature space. By pursuing this training objective, we guide the model to learn more discriminative feature representations. Therefore, the training process for F can be formulated as follows:

$$(\hat{\theta}_F) = \arg \min_{\theta_F} \alpha \mathcal{L}_{SCL} \tag{14}$$

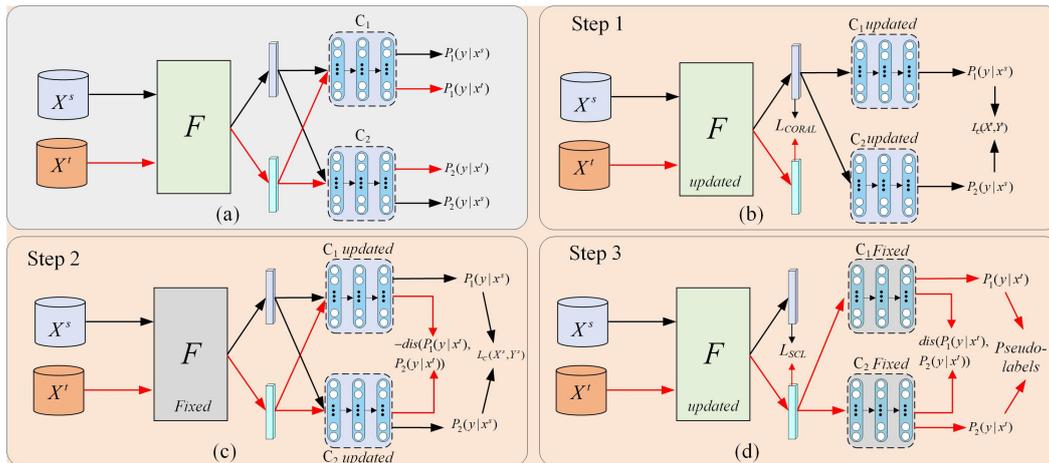
where  $\alpha$  is a hyper-parameter. Despite employing the DC strategy to obtain reliable pseudo-labels as much as possible, there exists some noise in the pseudo-labels of the target domain, particularly during the initial stages of network training. This noise can have an impact on supervised contrastive learning. To alleviate this issue, we introduce a hyper-parameter  $\alpha$  during the optimization of the supervised contrastive learning loss. The  $\alpha$  is gradually adjusted from 0 to 1 as the number of iterations progresses to mitigate the influence of noisy pseudo-labels. The variation of  $\alpha$  can be expressed as [35]

$$\alpha = \frac{2}{1 + \exp\left(-\gamma\left(\frac{i}{E}\right)\right)} - 1 \tag{15}$$

where  $i$  is the  $i$ -th epoch,  $E$  represents the total number of epochs and is set to 200, and  $\gamma$  is set to 10.

2.4. Training Process

As demonstrated in Figure 2, the training process involves three steps.



**Figure 2.** The detailed training process of the DCASCL model. (a) represents the prediction process, (b) represents Step 1, (c) represents Step 2, and (d) represents Step 3. In the figure, the black arrow indicates the flow of source domain data, while the red arrow indicates the flow of target domain data.

**Step 1:** the feature extractor F and the two classifiers  $C_1$  and  $C_2$  are trained synchronously using the cross-entropy loss term and the CORAL loss term, as shown in Figure 2b. The overall optimization goal is achieved by combining Equations (2) and (6) as follows:

$$(\hat{\theta}_F, \hat{\theta}_{C_1}, \hat{\theta}_{C_2}) = \arg \min_{\theta_F, \theta_{C_1}, \theta_{C_2}} \mathcal{L}_C(X^s, Y^s) + \lambda \arg \min_{\theta_F} \mathcal{L}_{CORAL} \tag{16}$$

**Step 2:** the feature extractor F is fixed, and the two classifiers  $C_1$  and  $C_2$  are trained using the discrepancy loss term, as displayed in Figure 2c. The objective of this step is to maximize the discrepancy of the prediction distributions between  $C_1$  and  $C_2$  on target samples using Equation (8).

**Step 3:** the two classifiers  $C_1$  and  $C_2$  are fixed, and the feature extractor  $F$  is trained using SCL loss term and discrepancy loss term, as depicted in Figure 2d. The objective of this step is to minimize the discrepancy loss along with the SCL loss. Integrating Equations (10) and (14), the training process for  $F$  in this step can be expressed as follows:

$$(\hat{\theta}_F) = \arg \min_{\theta_F} \{ \mathcal{L}_{adv}(X^t) + \alpha \mathcal{L}_{SCL} \} \quad (17)$$

where  $\alpha$  and  $\lambda$  are the trade-off parameters. The above three steps are iterated until the model is trained. And Algorithm 1 summarizes the overall algorithm and training process of DCASCL.

---

**Algorithm 1:** Training process of DCASCL

---

**Input:** the labeled samples  $X^s$  and the corresponding label  $Y^s$ , the unlabeled samples  $X^t$ , number of epochs (E), number of batch size (B), initial learning rate, and the trade-off parameter  $\lambda$

**Output:** Optimal parameters  $\theta_F$  of  $F$ , Optimal parameters  $\theta_{C_1}$  and  $\theta_{C_2}$  of  $C_1$  and  $C_2$

1. **For** epoch = 1 **to** E **do**
  2.    $\alpha$  increases from 0 to 1
  3.   **For** i = 1 **to** B **do**
  4.     **#Step 1:** Simultaneously update parameters of  $F$ ,  $C_1$ , and  $C_2$ ,
  5.     Calculate classification loss  $\mathcal{L}_C(X^s, Y^s)$  and correlation alignment loss  $\mathcal{L}_{CORAL}$  using Equations (1) and (3)
  6.     Update parameters of  $F$ ,  $C_1$ ,  $C_2$  using the Equation (16)
  7.     **#Step 2:** Update parameters of  $C_1$  and  $C_2$ , fix parameters of  $F$
  8.     Calculate classifier discrepancy  $dis(p_1(y|x^t), p_2(y|x^t))$  using Equation (7)
  9.     Update parameters of  $C_1$ ,  $C_2$  using the Equation (8)
  10.    **#Step 3:** Update parameters of  $F$ , fix parameters of  $C_1$  and  $C_2$
  11.    Calculate classifier discrepancy  $dis(p_1(y|x^t), p_2(y|x^t))$  and supervised contrastive learning loss using Equations (7) and (12)
  12.    Update parameters of  $C_1$ ,  $C_2$  using the Equation (17)
  13.    **End**
  14. **End**
- 

### 3. Experimental Results and Discussion

#### 3.1. Dataset Description

We conducted experiments on three distinct datasets to assess the performance of the DCASCL model. These datasets include the Case Western Reserve University (CWRU) dataset, the Southeast University (SEU) dataset, and the Jiangnan University (JNU) dataset, respectively. Details about these three datasets are described below.

**CWRU Bearing Dataset:** The CWRU dataset is commonly utilized for fault diagnosis [36], and its experimental configuration is depicted in Figure 3, which is adapted from [37]. Downloads for it are available at [38]. The dataset was acquired at either 12 kHz or 48 kHz. In this paper, we specifically utilized the data collected at 12 kHz. They contain vibration signals obtained from bearings, covering four distinct operating conditions: 0, 1, 2, and 3 hp. Each operating condition comprises ten distinct health states of the bearings, including normal (N), inner race fault (IF) with defect diameters of 0.007, 0.014, and 0.021 inches, as well as both ball fault (BF) and outer race fault (OF) with the corresponding defect diameters. Twelve different transfer tasks are established based on the distinct operating conditions. In the transfer task 0 hp  $\rightarrow$  1 hp, it indicates the utilization of 0 hp and 1 hp operating conditions as the source domain and target domain datasets, respectively. Detailed information can be found in Table 2.

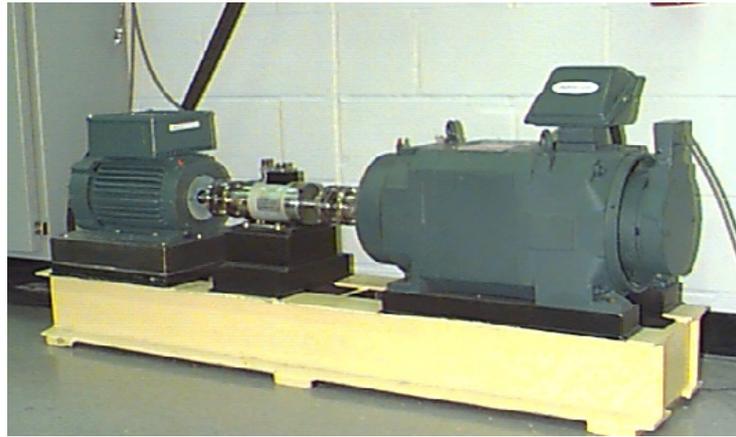


Figure 3. CWRU bearing test rig [37].

Table 2. Description of the CWRU dataset.

<b>Fault Type</b>	BF	BF	BF	IF	IF	IF	OF	OF	OF	N
<b>Fault Size (Inches)</b>	7	14	21	7	14	21	7	14	21	0
<b>Class Label</b>	0	1	2	3	4	5	6	7	8	9
<b>Load (hp)</b>	0, 1, 2, 3									
<b>Total number of samples</b>	4000									
<b>Train and test set ratio</b>	7:3									

**JNU Bearing Dataset:** Jiangnan University provided this dataset [39], which is also commonly used for research in bearing fault diagnosis. Its experimental signal acquisition system is illustrated in Figure 4, which is sourced from [39], and it can be obtained from [40]. The vibration data in the JNU bearing dataset were acquired at three different speeds: 600, 800, and 1000 rpm. Normal health (N), inner fault (IF), outer fault (OF), and ball fault (BF) are the four health states included in each speed. Six transfer tasks were established based on the three different speed conditions, where 600 rpm  $\rightarrow$  800 rpm indicates that the dataset obtained at the speed condition of 600 rpm is used for the source domain and 800 rpm is employed for the target domain. Table 3 provides comprehensive information.

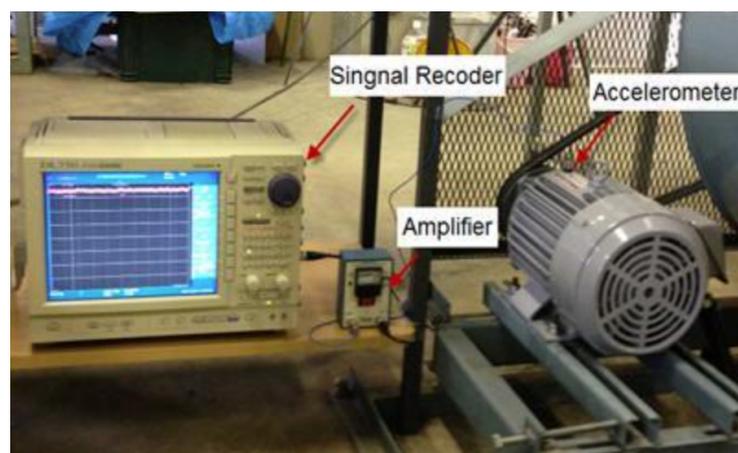
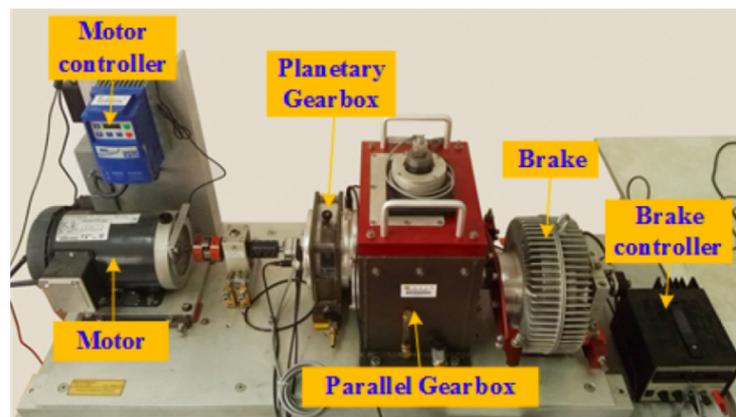


Figure 4. Experimental signal acquisition system for the JNU dataset [39].

**Table 3.** Description of the JNU dataset.

Fault Type	IF	N	OF	BF
Class Label	0	1	2	3
Speed (rpm)	600, 800,1000			
Total number of samples	2400			
Train and test set ratio	7:3			

**SEU Gearbox Dataset:** This dataset is from the Southeast University in China [41], and its experimental setup is depicted in Figure 5, which is adapted from [37]. It can be downloaded from [42]. The dataset is separated into two sub-datasets that provide information on the health of bearings and gearboxes. The data were collected using a Drivetrain Dynamics Simulator (DDS) and included eight vibration information channels. The data from the second channel of the gearbox dataset are used in this work. This dataset contains two operating conditions based on speed and load: 20 HZ-0V and 30 HZ-2V. Each operating condition includes one healthy condition and four faulty conditions, namely Health, Chipped, Root, Miss, and Surface. Based on these two operating conditions, two transfer tasks are created, where 20 HZ-0V → 30 HZ-2V represents the utilization of the 20 HZ-0V dataset for the source domain and the 30 HZ-2V dataset for the target domain. For more comprehensive information, refer to Table 4.

**Figure 5.** The experimental setup of SEU dataset [37].**Table 4.** Description of the SEU dataset.

Fault Type	Chipped	Health	Miss	Root	Surface
Class Label	0	1	2	3	4
RS-LC	20 HZ-0V, 30 HZ-2V				
Total number of samples	3000				
Train and test set ratio	7:3				

### 3.2. Data Processes

In this paper, the same approach is employed for preprocessing the three datasets mentioned above. First, we applied a random sampling approach to sample the raw vibration data, with a sample length of 4096 for each sample. Then, we applied the Fast Fourier Transform (FFT) technique to each randomly sampled data. Due to the symmetry of the spectral coefficients, we used only half of the length of the FFT-transformed samples, which is 2048. Finally, the FFT-processed samples underwent Z-score standardization, with the Z-score standardization formula as follows:

$$x_i^n = \frac{x_i - x_i^{mean}}{x_i^{std}} \quad (18)$$

Here, the mean and standard deviation of  $x_i$  are indicated by  $x_i^{mean}$  and  $x_i^{std}$ , respectively.

### 3.3. Implementation Details

We constructed the entire model using the PyTorch framework and conducted both training and testing on a PC equipped with an NVIDIA RTX 3060 GPU. The batch size for both the source and target domains is 128. The Adam optimizer is utilized for model optimization, with an initial learning rate of 0.0001 for both the feature extractor and the two classifiers. The dropout ratio is established at 0.2. The accuracy of diagnosis on the target domain is utilized as a metric to assess the performance of our suggested model. To mitigate randomness and bolster result robustness, each experiment undergoes ten iterations, and the average performance over these repetitions is reported as the final result.

### 3.4. Comparison Methods

We compared our proposed strategy to numerous approaches in order to assess its efficacy and superiority:

- (1) No Domain Adaptation: 1D-CNN serves as a baseline method, utilizing only source domain data to directly train the model for diagnostic tasks in the target domain.
- (2) Only Domain Distribution Alignment: Both MK-MMD [43] and CORAL [15] align distributions by matching statistical differences between two domains. DANN [44] introduces a domain discriminator to differentiate between domains and encourages the model to learn representations that are invariant across domains by confusing the discriminator.
- (3) Only Class Distribution Alignment: MCD [30] method maximizes discrepancy in predictions on unlabeled target samples between two separate classifiers during optimization. Meanwhile, it minimizes this discrepancy when optimizing the feature extractor to generate target features under the support of the source domain.

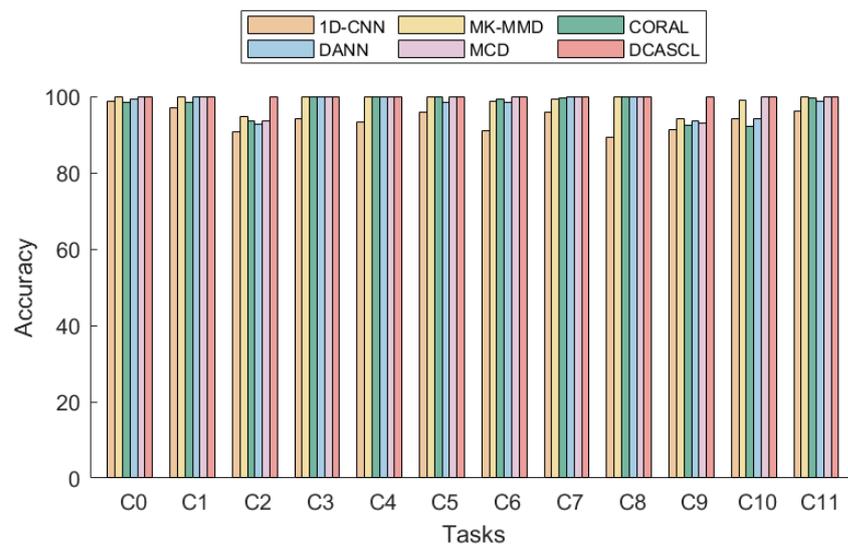
### 3.5. Experimental Results and Analysis

Tables 5–7 and Figures 6–8 display detailed diagnostic performance for our method and other approaches in the transfer tasks across three datasets.

As presented in Table 5, DCASCL consistently demonstrated top accuracy in all twelve transfer tasks within the CRWU dataset, achieving a remarkable 100% accuracy. Although MCD achieved perfect accuracy in ten tasks, its accuracy decreased to 93.45% and 92.96% in the 0 hp → 3 hp and 3 hp → 0 hp tasks, respectively, primarily due to the significant variations in working conditions between these two tasks.

**Table 5.** Results of DCASCL and other approaches on CRWU dataset.

Tasks Symbol	Tasks	1D-CNN	MK-MMD	CORAL	DANN	MCD	DCASCL
C0	0 hp → 1 hp	98.67	100	98.53	99.35	100	<b>100</b>
C1	0 hp → 2 hp	97.08	100	98.54	100	100	<b>100</b>
C2	0 hp → 3 hp	90.84	94.65	93.65	92.76	93.45	<b>100</b>
C3	1 hp → 0 hp	94.04	99.81	100	99.78	100	<b>100</b>
C4	1 hp → 2 hp	93.19	100	100	100	100	<b>100</b>
C5	1 hp → 3 hp	95.79	99.84	100	98.43	100	<b>100</b>
C6	2 hp → 0 hp	90.99	98.85	99.23	98.46	100	<b>100</b>
C7	2 hp → 1 hp	95.73	99.35	99.68	100	100	<b>100</b>
C8	2 hp → 3 hp	89.19	100	100	100	100	<b>100</b>
C9	3 hp → 0 hp	91.21	94.13	92.50	93.51	92.96	<b>100</b>
C10	3 hp → 1 hp	94.18	99.03	92.21	94.28	100	<b>100</b>
C11	3 hp → 2 hp	96.24	100	99.68	98.68	100	<b>100</b>
Average	-	93.93	98.81	97.84	97.94	98.87	<b>100</b>



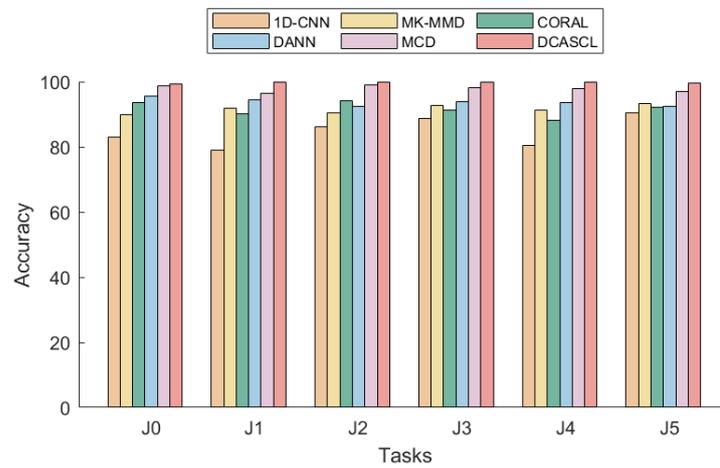
**Figure 6.** The experimental results of the CRWU dataset.

Due to the relatively straightforward nature of diagnosing bearing faults across different domains in the CRWU dataset, the advantage of DCASCL is not as pronounced. Hence, the evaluation is extended to the JNU dataset. Compared to the CRWU dataset, the JNU dataset involves significant variations in rotation speed, making domain adaptation more challenging due to substantial differences in features implied by similar signal states. Table 6 presents the experimental outcomes of various methods across six transfer tasks using the JNU dataset. The experimental results highlight the adaptability of the DCASCL model to the signals of the same faults that bear large differences between different working conditions (in this case, different rotation speeds). In each transfer task, DCASCL outperformed other methods in terms of diagnostic performance. Particularly, DCASCL attained remarkable accuracies of 100% and 99.68%, for the challenging tasks of 600 rpm  $\rightarrow$  1000 rpm and 1000 rpm  $\rightarrow$  600 rpm, respectively. These results highlight the superiority of DCASCL, even when faced with substantial variations in working conditions.

**Table 6.** Results of DCASCL and other approaches on JNU dataset.

Tasks Symbol	Tasks	1D-CNN	MK-MMD	CORAL	DANN	MCD	DCASCL
J0	600 rpm $\rightarrow$ 800 rpm	83.01	89.73	93.45	95.57	98.67	<b>99.37</b>
J1	600 rpm $\rightarrow$ 1000 rpm	78.95	91.80	90.27	94.32	96.34	<b>100</b>
J2	800 rpm $\rightarrow$ 600 rpm	86.09	90.36	94.05	92.51	99.07	<b>100</b>
J3	800 rpm $\rightarrow$ 1000 rpm	88.65	92.63	91.25	93.83	98.12	<b>100</b>
J4	1000 rpm $\rightarrow$ 600 rpm	80.49	91.27	88.14	93.67	97.81	<b>99.68</b>
J5	1000 rpm $\rightarrow$ 800 rpm	90.35	93.27	92.13	92.49	97.06	<b>100</b>
Average	-	84.59	91.51	91.55	93.73	97.85	<b>99.84</b>

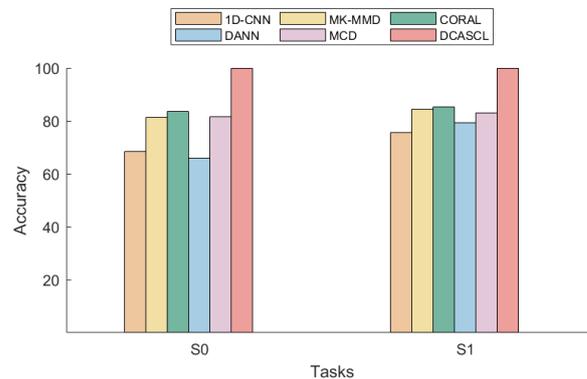
Table 7 presents the results achieved from the experiments performed on two transfer tasks from the SEU dataset, where our method consistently showed the best performance. In particular, our method has shown a fault diagnosis accuracy of 100% in both tasks. It significantly improves the accuracy by 31.65% in the 20 HZ-0V  $\rightarrow$  30 HZ-2V task compared to the accuracy results achieved when no domain adaptation is used.



**Figure 7.** The experimental results of the JNU dataset.

**Table 7.** Results of DCASCL and other approaches on SEU dataset.

Tasks Symbol	Tasks	1D-CNN	MK-MMD	CORAL	DANN	MCD	DCASCL
S0	20 HZ-0V → 30 HZ-2V	68.35	81.35	83.62	65.86	81.48	<b>100</b>
S1	30 HZ-2V → 20 HZ-0V	75.43	84.52	85.17	79.32	82.93	<b>100</b>
Average	-	71.89	82.94	84.40	72.59	82.21	<b>100</b>



**Figure 8.** The experimental results of the SEU dataset.

From these three experimental result tables, it is observed that all domain adaptation methods outperform baseline methods in terms of average accuracy. Notably, DCASCL excels in this aspect, surpassing other comparative domain adaptation methods, those that solely focus on either domain distribution alignment or class distribution alignment. This further underscores the importance of simultaneously considering both types of distribution alignment in the context of fault diagnosis domain adaptation.

## 4. Model Analysis

### 4.1. Ablation Studies

To assess the impact of each component in the DCASCL model on performance, we conducted ablation experiments on transfer tasks selected from the CWRU, JNU, and SEU datasets, specifically, the 0 hp → 3 hp, 600 rpm → 1000 rpm, and 20 HZ-0V → 30 HZ-2V tasks. MCD was chosen as our baseline method, and thus we categorized the ablation experiments into four groups: (1) MCD, (2) MCD combined with the CORAL loss (MCD+CORAL), (3) MCD combined with supervised contrastive learning loss (MCD+SCL), and (4) DCASCL. We utilized accuracy and F1-score as evaluation metrics, where F1-score, a commonly used

comprehensive metric, considers both precision and recall and can be represented as the following formula:

$$F1 = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \quad (19)$$

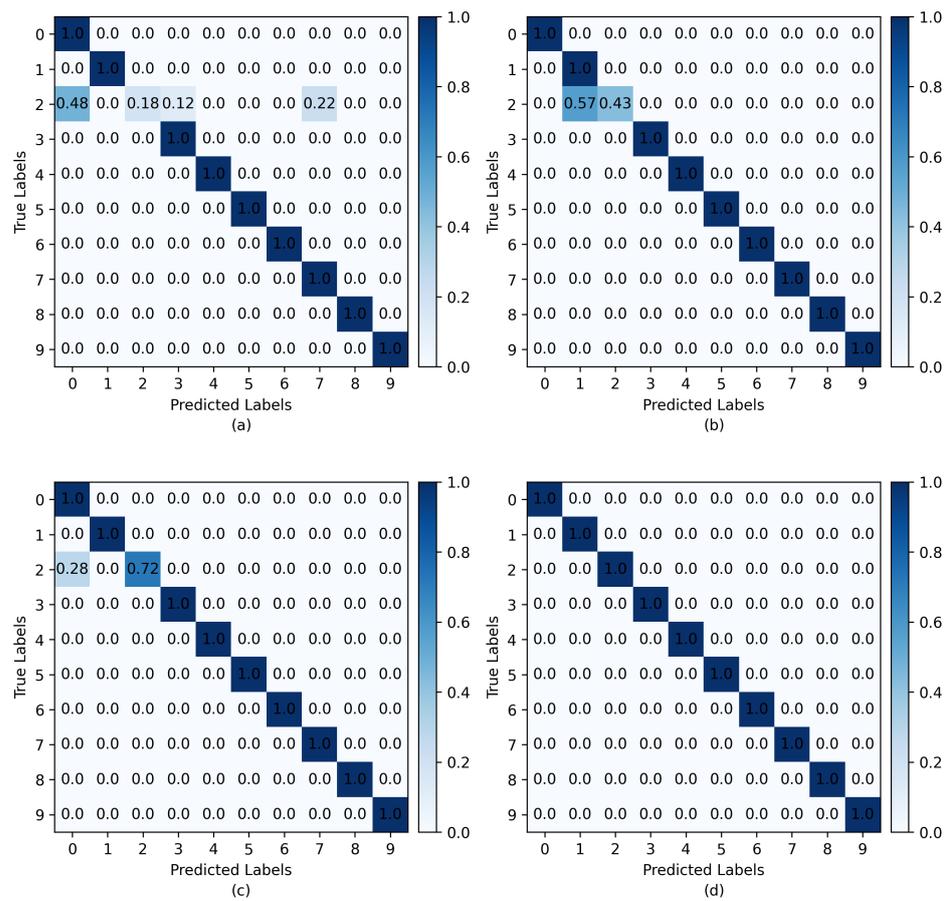
In addition to these two evaluation metrics, we also introduced confusion matrices for each experiment. Confusion matrices are tools used to visualize model prediction results by comparing the true classes with the predicted classes, enabling a better understanding of the model's performance across different classes. Table 8 presents the results of each experiment, while Figures 9–11 depict the corresponding confusion matrices.

**Table 8.** Performance comparison of methods on different transfer tasks.

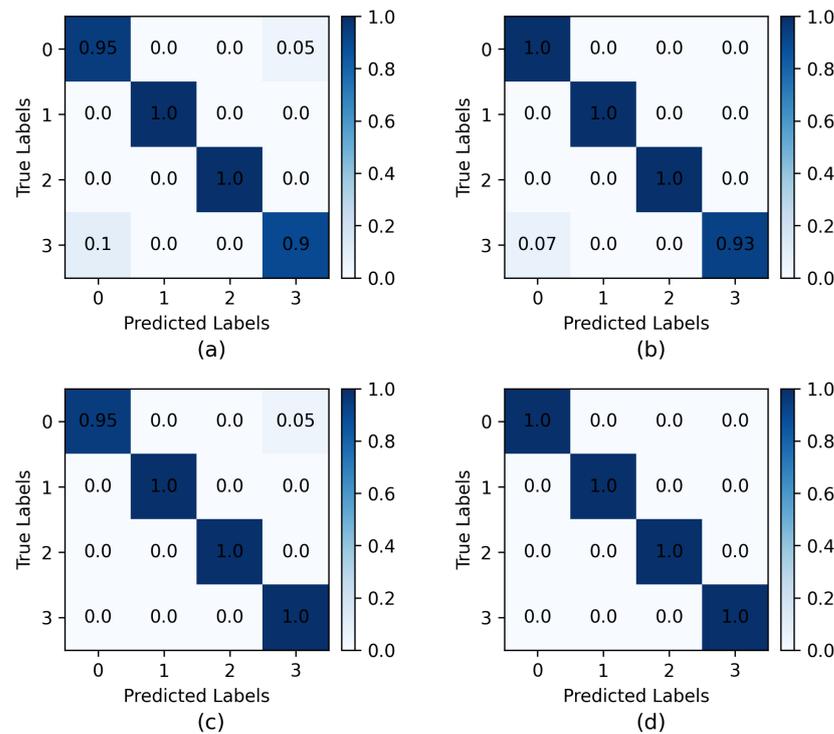
Method	0 hp → 3 hp		600 rpm → 1000 rpm		20 HZ-0V → 30 HZ-2V	
	Accuracy	F1-Score	Accuracy	F1-Score	Accuracy	F1-Score
MCD	91.83	89.62	96.25	96.25	81.56	81.84
MCD+CORAL	94.33	93.84	98.19	98.19	91.44	91.52
MCD+SCL	97.25	97.20	98.75	98.75	93.89	93.88
DCASCL	<b>100.0</b>	<b>100.0</b>	<b>100.0</b>	<b>100.0</b>	<b>100.0</b>	<b>100.0</b>

In Table 8 of the experimental results, it is evident that the MCD method consistently achieves the lowest accuracy and F1-score in each transfer task. However, upon incorporating the CORAL and SCL losses into the MCD method separately, we observed improvements in accuracy and F1-score across all transfer tasks. Particularly notable is the significant enhancement in accuracy, with increases of 9.88% and 12.33% observed in the 20 HZ-0V → 30 HZ-2V transfer task. Our proposed method, DCASCL, integrates both these components, resulting in the highest accuracy and F1-score in each transfer task, both reaching 100%.

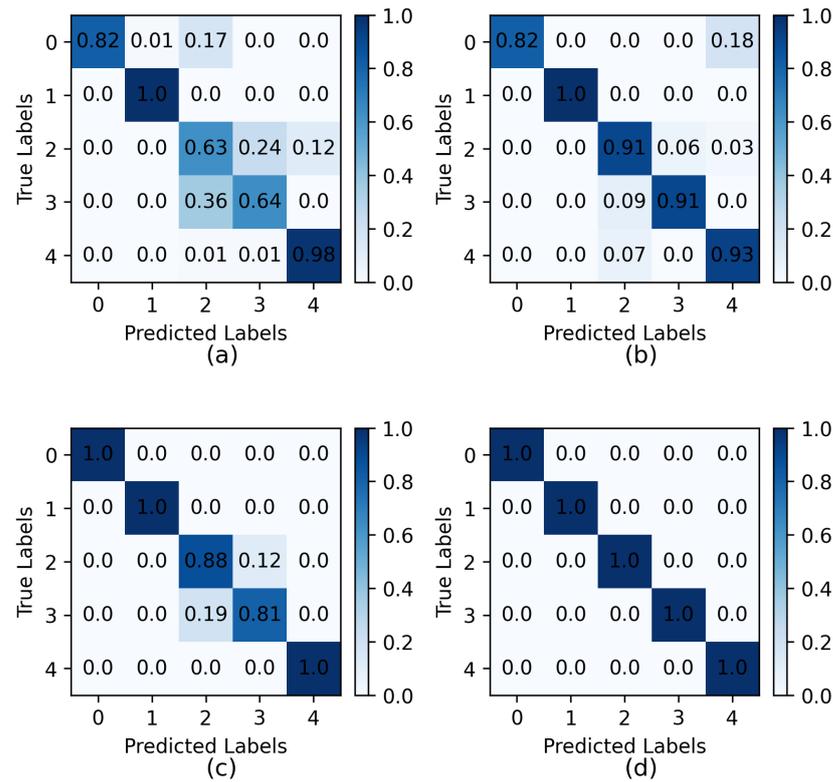
Further examination of the confusion matrices in Figures 9a and 10a reveals that the misclassification of the MCD method is not severe in the 0 hp → 3 hp and 600 rpm → 1000 rpm transfer tasks. In the 0 hp → 3 hp transfer task, some confusion exists primarily in the third category, wherein samples are incorrectly classified into multiple classes. Similarly, in the 600 rpm → 1000 rpm transfer task, only a small number of misclassification occurs between classes 1 and 4. However, from Figure 11a, it is evident that the misclassification of the MCD method is more pronounced in the 20 HZ-0V → 30 HZ-2V transfer task, particularly between classes 3 and 4, where samples are erroneously classified into multiple classes. After incorporating the CORAL and SCL components, we observed some improvements in the misclassification situation from Figures 9–11. We attribute this misclassification to the inadequate consideration of global domain alignment and the absence of class-specific information in the MCD method. The lack of such information may lead to ambiguous target samples being incorrectly matched to the wrong classes. Given that the CORAL component strengthens knowledge transfer between domains and the SCL component leverages class-discriminative information to enhance the network's ability to differentiate between different fault classes, both components contribute to improved fault diagnosis accuracy. Therefore, our method integrates both these components, effectively resolving the misclassification issue across the three transfer tasks, with all classes being correctly classified. These analytical results further validate the superior performance and robustness of our method in fault diagnosis.



**Figure 9.** Confusion matrix of combinations between different components on the 0 hp → 3 hp transfer task: (a) MCD; (b) MCD+CORAL; (c) MCD+SCL; (d) DCASCL.



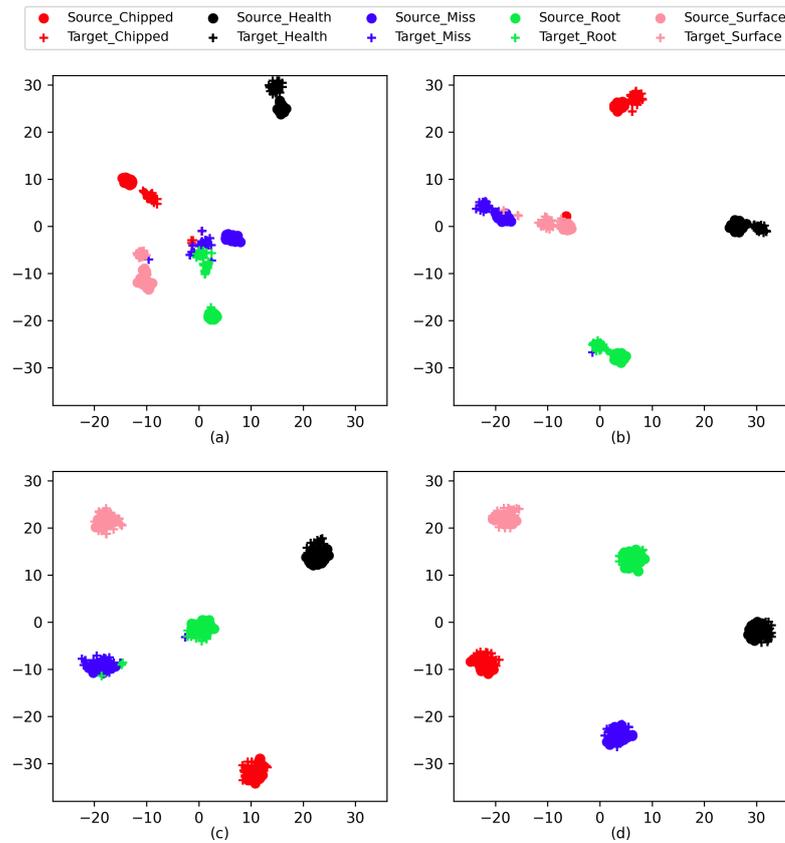
**Figure 10.** Confusion matrix of combinations between different components on the 600 rpm → 1000 rpm transfer task: (a) MCD; (b) MCD+CORAL; (c) MCD+SCL; (d) DCASCL.



**Figure 11.** Confusion matrix of combinations between different components on the 20 Hz-0V → 30 Hz-2V transfer task: (a) MCD; (b) MCD+CORAL; (c) MCD+SCL; (d) DCASCL.

#### 4.2. Feature Visualization

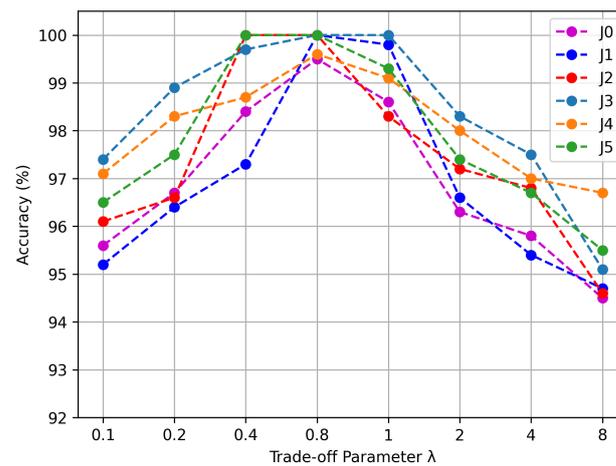
To provide a more intuitive visualization of the impact of different components on the feature extraction capabilities of DCASCL, we employed t-SNE [45] technology. T-SNE is a commonly used technique for dimensionality reduction and visualization, aiming to map high-dimensional data to two-dimensional or three-dimensional space, allowing for easier interpretation. Figure 12 illustrates the effect of utilizing different components on the model's feature extraction. Figure 12a visually illustrates the effects of training the model using the conventional MCD method on the extracted features from samples in both the source and target domains. From the visualization, it is evident that there are significant overlapping regions among the features of the three categories, and there is a noticeable separation between the features extracted from the source and target samples. In Figure 12b, after incorporating the CORAL loss for global alignment, the overlapping regions of category features are reduced, and the distance between the features of the source and target domains is relatively minimized but not sufficiently compact. However, there are still some individual samples that are close to the class boundaries of different categories. When only the supervised contrastive learning loss is added, as shown in Figure 12c, the feature distances between samples of the same category from the source and target domains show a more compact arrangement, whereas the distances between samples of different categories increase. In Figure 12d, after incorporating both the CORAL and SCL losses, the aforementioned issues are addressed. The class boundaries become clearer, and the feature distances between different domains are highly compact. The results from the ablation experiments indicate that both domain distribution alignment (achieved through the CORAL loss) and class distribution alignment (achieved through MCD and the SCL loss) play a vital role in enhancing the model's fault diagnosis performance across diverse domains.



**Figure 12.** Feature visualization of combinations between different components in the 20 HZ-0V → 30 HZ-2V transfer task of the SEU dataset: (a) MCD; (b) MCD+CORAL; (c) MCD+SCL; (d) DCASCL.

#### 4.3. Parameter Analysis

In this study, we conducted a trade-off parameter analysis of model performance. Our model has two main trade-off parameters:  $\alpha$  and  $\lambda$ . Among them,  $\alpha$  is a dynamically changing trade-off parameter, as defined in Equation (15). Therefore, we focused on analyzing the impact of the trade-off parameter  $\lambda$  on model performance. We conducted experiments on the  $\lambda$  trade-off parameter across all transfer learning tasks on the JNU dataset to observe its effect on model performance. From Figure 13, it can be seen that, as  $\lambda$  gradually increases from a small value of 0.1 to 0.8, the model's accuracy shows a gradually increasing trend. The highest classification accuracy across all six tasks on the JNU dataset was achieved at  $\lambda = 0.8$ . Around this value, the model's performance exhibits relative stability. However, as the value of  $\lambda$  continues to increase, surpassing 1, the model's accuracy begins to decline. Therefore, we set the hyperparameter  $\lambda$  to 0.8. Overall,  $\lambda$  has a significant impact on model performance, and selecting an appropriate value of  $\lambda$  can enhance model performance, while avoiding excessively large values of  $\lambda$  that may lead to performance degradation.



**Figure 13.** The impact of the trade-off parameter  $\lambda$  on the JNU dataset.

## 5. Conclusions

This paper proposes a method named DCASCL to diagnose rotating mechanical equipment failures across different operating conditions. Unlike most existing approaches, our method simultaneously conducts the alignment of feature distributions on global domain and class-level scales to cope with the differences in domains and classes present in cross-domain scenarios. We initially utilize the correlation alignment loss to measure the distance between feature distributions of source and target domains in the high-dimension feature space. We then minimize this loss to bring the feature distributions of the two domains closer to each other, achieving feature representations that are domain-invariant and robust to the domain disparity. Moreover, we align the feature distributions class-wisely by incorporating supervised contrastive learning loss and classifier discrepancy loss into our model. The supervised contrastive learning loss leverages the class-label information of target domain samples obtained through the double-confirmation (DC) strategy, along with labeled source domain samples. This enables it to effectively cluster samples of the same class in the feature space and to effectively separate samples of different classes in the feature space, regardless of whether these samples come from the source domain or the target domain. By computing the supervised contrastive learning loss between target domain and source domain samples, the feature extractor is updated, thereby enhancing the performance of DCASCL. Additionally, we experimentally validate our proposed DCASCL approach on three publicly available datasets. The results obtained from these experiments show the capability of the DCASCL in diagnosing machine faults under various operating conditions, surpassing many works carried out in the field of domain adaptation fault diagnosis.

Although our method has achieved significant results in diagnosing faults in rotating mechanical equipment, especially in bearing and gear faults, we also recognize some limitations. Firstly, our method has shown good performance in diagnosing faults in rotating mechanical equipment under certain specific conditions. However, there may be challenges when facing working conditions or fault types that have not been previously encountered. Additionally, our model has not addressed the issue of data imbalance, which may affect its performance in certain scenarios. Therefore, we will focus on addressing these limitations in future research and further improve our method to enhance its robustness and applicability in various scenarios.

**Author Contributions:** Data curation, B.Z. and H.D.; methodology, B.Z. and H.D.; software, B.Z., H.D. and H.A.A.M.Q.; formal analysis, Y.W.; writing—original draft, B.Z., H.D. and H.A.A.M.Q.; writing—review and editing, B.Z., H.D., H.A.A.M.Q. and Y.W. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work is supported by the Fundamental Research Funds for the Central Universities 2019ZDPY08.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Data are contained within the article.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. Zhang, S.; Lei, S.; Jiefei, G.; Ke, L.; Lang, Z.; Pecht, M. Rotating machinery fault detection and diagnosis based on deep domain adaptation: A survey. *Chin. J. Aeronaut.* **2023**, *36*, 45–74. [[CrossRef](#)]
2. Joksimović, G.M.; Riger, J.; Wolbank, T.M.; Perić, N.; Vašak, M. Stator-current spectrum signature of healthy cage rotor induction machines. *IEEE Trans. Ind. Electron.* **2012**, *60*, 4025–4033. [[CrossRef](#)]
3. Hong, L.; Dhupia, J.S. A time domain approach to diagnose gearbox fault based on measured vibration signals. *J. Sound Vib.* **2014**, *333*, 2164–2180. [[CrossRef](#)]
4. Yan, X.; Jia, M. A novel optimized SVM classification algorithm with multi-domain feature and its application to fault diagnosis of rolling bearing. *Neurocomputing* **2018**, *313*, 47–64. [[CrossRef](#)]
5. Lu, J.; Qian, W.; Li, S.; Cui, R. Enhanced K-nearest neighbor for intelligent fault diagnosis of rotating machinery. *Appl. Sci.* **2021**, *11*, 919. [[CrossRef](#)]
6. Janssens, O.; Slavkovikj, V.; Vervisch, B.; Stockman, K.; Loccufier, M.; Verstockt, S.; Van de Walle, R.; Van Hoecke, S. Convolutional neural network based fault detection for rotating machinery. *J. Sound Vib.* **2016**, *377*, 331–345. [[CrossRef](#)]
7. Wang, P.; Yan, R.; Gao, R.X. Virtualization and deep recognition for system fault classification. *J. Manuf. Syst.* **2017**, *44*, 310–316. [[CrossRef](#)]
8. Shao, H.; Jiang, H.; Zhao, H.; Wang, F. A novel deep autoencoder feature learning method for rotating machinery fault diagnosis. *Mech. Syst. Signal Process.* **2017**, *95*, 187–204. [[CrossRef](#)]
9. Lei, Y.; Yang, B.; Jiang, X.; Jia, F.; Li, N.; Nandi, A.K. Applications of machine learning to machine fault diagnosis: A review and roadmap. *Mech. Syst. Signal Process.* **2020**, *138*, 106587. [[CrossRef](#)]
10. Zhao, Z.; Zhang, Q.; Yu, X.; Sun, C.; Wang, S.; Yan, R.; Chen, X. Applications of unsupervised deep transfer learning to intelligent fault diagnosis: A survey and comparative study. *IEEE Trans. Instrum. Meas.* **2021**, *70*, 1–28. [[CrossRef](#)]
11. Li, X.; Zhang, W. Deep learning-based partial domain adaptation method on intelligent machinery fault diagnostics. *IEEE Trans. Ind. Electron.* **2020**, *68*, 4351–4361. [[CrossRef](#)]
12. Liao, Y.; Huang, R.; Li, J.; Chen, Z.; Li, W. Deep semisupervised domain generalization network for rotary machinery fault diagnosis under variable speed. *IEEE Trans. Instrum. Meas.* **2020**, *69*, 8064–8075.
13. Lu, N.; Xiao, H.; Sun, Y.; Han, M.; Wang, Y. A new method for intelligent fault diagnosis of machines based on unsupervised domain adaptation. *Neurocomputing* **2021**, *427*, 96–109. [[CrossRef](#)]
14. Borgwardt, K.M.; Gretton, A.; Rasch, M.J.; Kriegel, H.P.; Schölkopf, B.; Smola, A.J. Integrating structured biological data by kernel maximum mean discrepancy. *Bioinformatics* **2006**, *22*, e49–e57. [[CrossRef](#)]
15. Sun, B.; Saenko, K. Deep coral: Correlation alignment for deep domain adaptation. In *Proceedings of the Computer Vision–ECCV 2016 Workshops, Amsterdam, The Netherlands, 8–10. 15–16 October 2016*; Proceedings, Part III 14; Springer: Berlin/Heidelberg, Germany, 2016; pp. 443–450.
16. Sun, B.; Feng, J.; Saenko, K. Correlation alignment for unsupervised domain adaptation. In *Domain Adaptation in Computer Vision Applications*; Springer: Cham, France, 2017; pp. 153–171.
17. Zellinger, W.; Grubinger, T.; Lughofer, E.; Natschläger, T.; Saminger-Platz, S. Central moment discrepancy (cmd) for domain-invariant representation learning. *arXiv* **2017**, arXiv:1702.08811.
18. Long, M.; Cao, Y.; Wang, J.; Jordan, M. Learning transferable features with deep adaptation networks. In *Proceedings of the International Conference on Machine Learning, Lille, France, 7–9 July 2015*; pp. 97–105.
19. Wen, L.; Gao, L.; Li, X. A new deep transfer learning based on sparse auto-encoder for fault diagnosis. *IEEE Trans. Syst. Man Cybern. Syst.* **2017**, *49*, 136–144. [[CrossRef](#)]
20. Yang, B.; Lei, Y.; Jia, F.; Xing, S. An intelligent fault diagnosis approach based on transfer learning from laboratory bearings to locomotive bearings. *Mech. Syst. Signal Process.* **2019**, *122*, 692–706. [[CrossRef](#)]
21. Che, C.; Wang, H.; Ni, X.; Fu, Q. Domain adaptive deep belief network for rolling bearing fault diagnosis. *Comput. Ind. Eng.* **2020**, *143*, 106427. [[CrossRef](#)]
22. Qian, Q.; Qin, Y.; Wang, Y.; Liu, F. A new deep transfer learning network based on convolutional auto-encoder for mechanical fault diagnosis. *Measurement* **2021**, *178*, 109352. [[CrossRef](#)]
23. Li, Q.; Shen, C.; Chen, L.; Zhu, Z. Knowledge mapping-based adversarial domain adaptation: A novel fault diagnosis method with high generalizability under variable working conditions. *Mech. Syst. Signal Process.* **2021**, *147*, 107095. [[CrossRef](#)]
24. Wang, H.; Bai, X.; Tan, J.; Yang, J. Deep prototypical networks based domain adaptation for fault diagnosis. *J. Intell. Manuf.* **2022**, *33*, 1–11. [[CrossRef](#)]

25. Wu, H.; Li, J.; Zhang, Q.; Tao, J.; Meng, Z. Intelligent fault diagnosis of rolling bearings under varying operating conditions based on domain-adversarial neural network and attention mechanism. *ISA Trans.* **2022**, *130*, 477–489. [[CrossRef](#)] [[PubMed](#)]
26. Li, X.; Zhang, W.; Xu, N.X.; Ding, Q. Deep learning-based machinery fault diagnostics with domain adaptation across sensors at different places. *IEEE Trans. Ind. Electron.* **2019**, *67*, 6785–6794. [[CrossRef](#)]
27. Chen, Z.; He, G.; Li, J.; Liao, Y.; Gryllias, K.; Li, W. Domain adversarial transfer network for cross-domain fault diagnosis of rotary machinery. *IEEE Trans. Instrum. Meas.* **2020**, *69*, 8702–8712. [[CrossRef](#)]
28. Chen, C.; Chen, Z.; Jiang, B.; Jin, X. Joint domain alignment and discriminative feature learning for unsupervised deep domain adaptation. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33; pp. 3296–3303.
29. Dai, S.; Cheng, Y.; Zhang, Y.; Gan, Z.; Liu, J.; Carin, L. Contrastively smoothed class alignment for unsupervised domain adaptation. In Proceedings of the Asian Conference on Computer Vision, Kyoto, Japan, 30 November 2020.
30. Saito, K.; Watanabe, K.; Ushiku, Y.; Harada, T. Maximum classifier discrepancy for unsupervised domain adaptation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 3723–3732.
31. Li, S.; Liu, C.H.; Xie, B.; Su, L.; Ding, Z.; Huang, G. Joint adversarial domain adaptation. In Proceedings of the 27th ACM International Conference on Multimedia, Nice, France, 21–25 October 2019; pp. 729–737.
32. Zhang, W.; Peng, G.; Li, C.; Chen, Y.; Zhang, Z. A new deep learning model for fault diagnosis with good anti-noise and domain adaptation ability on raw vibration signals. *Sensors* **2017**, *17*, 425. [[CrossRef](#)] [[PubMed](#)]
33. Khosla, P.; Teterwak, P.; Wang, C.; Sarna, A.; Tian, Y.; Isola, P.; Maschinot, A.; Liu, C.; Krishnan, D. Supervised contrastive learning. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 18661–18673.
34. Sohn, K.; Berthelot, D.; Carlini, N.; Zhang, Z.; Zhang, H.; Raffel, C.A.; Cubuk, E.D.; Kurakin, A.; Li, C.L. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 596–608.
35. Ganin, Y.; Lempitsky, V. Unsupervised domain adaptation by backpropagation. In Proceedings of the International Conference on Machine Learning, Lille, France, 7–9 July 2015; pp. 1180–1189.
36. Smith, W.A.; Randall, R.B. Rolling element bearing diagnostics using the Case Western Reserve University data: A benchmark study. *Mech. Syst. Signal Process.* **2015**, *64*, 100–131. [[CrossRef](#)]
37. Zhang, B.; Zhou, C.; Li, W.; Ji, S.; Li, H.; Tong, Z.; Ng, S.K. Intelligent bearing fault diagnosis based on open set convolutional neural network. *Mathematics* **2022**, *10*, 3953. [[CrossRef](#)]
38. Case Western Reserve University Bearings Vibration Dataset. Available online: <https://engineering.case.edu/bearingdatacenter/download-data-file> (accessed on 15 August 2022).
39. Li, K.; Ping, X.; Wang, H.; Chen, P.; Cao, Y. Sequential fuzzy diagnosis method for motor roller bearing in variable operating conditions based on vibration analysis. *Sensors* **2013**, *13*, 8013–8041. [[CrossRef](#)]
40. Li, K. School of Mechanical Engineering, Jiangnan University. Available online: <http://mad-net.org:8765/explore.html?t=0.5831516555847212> (accessed on 20 August 2019).
41. Shao, S.; McAleer, S.; Yan, R.; Baldi, P. Highly accurate machine fault diagnosis using deep transfer learning. *IEEE Trans. Ind. Inform.* **2018**, *15*, 2446–2455. [[CrossRef](#)]
42. SEU Gearbox Dataset. Available online: <https://github.com/cathysiyu/mechanical-datasets> (accessed on 3 August 2022).
43. Li, X.; Zhang, W.; Ding, Q.; Sun, J.Q. Multi-layer domain adaptation method for rolling bearing fault diagnosis. *Signal Process.* **2019**, *157*, 180–197. [[CrossRef](#)]
44. Ganin, Y.; Ustinova, E.; Ajakan, H.; Germain, P.; Larochelle, H.; Laviolette, F.; March, M.; Lempitsky, V. Domain-adversarial training of neural networks. *J. Mach. Learn. Res.* **2016**, *17*, 1–35.
45. Van der Maaten, L.; Hinton, G. Visualizing data using t-SNE. *J. Mach. Learn. Res.* **2008**, *9*, 2579–2605.

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.