



## Article

# Fault Diagnosis of Planetary Gearbox Based on Dynamic Simulation and Partial Transfer Learning

Mengmeng Song<sup>1</sup>, Zicheng Xiong<sup>2</sup>, Jianhua Zhong<sup>2,\*</sup>, Shungen Xiao<sup>1,\*</sup>  and Jihua Ren<sup>3</sup>

<sup>1</sup> College of Information, Mechanical and Electrical Engineering, Ningde Normal University, Ningde 352000, China; t1119@ndnu.edu.cn

<sup>2</sup> College of Mechanical Engineering and Automation, Fuzhou University, Fuzhou 350202, China

<sup>3</sup> Dongguan Xinghuo Gear Co., Ltd., Dongguan 523000, China; sharyren@xhgear.com

\* Correspondence: jhzhong@fzu.edu.cn (J.Z.); xiaoshungen022@163.com (S.X.)

**Abstract:** To address the problem of insufficient real-world data on planetary gearboxes, which makes it difficult to diagnose faults using deep learning methods, it is possible to obtain sufficient simulation fault data through dynamic simulation models and then reduce the difference between simulation data and real data using transfer learning methods, thereby applying diagnostic knowledge from simulation data to real planetary gearboxes. However, the label space of real data may be a subset of the label space of simulation data. In this case, existing transfer learning methods are susceptible to interference from outlier label spaces in simulation data, resulting in mismatching. To address this issue, this paper introduces multiple domain classifiers and a weighted learning scheme on the basis of existing domain adversarial transfer learning methods to evaluate the transferability of simulation data and adaptively measure their contribution to label predictor and domain classifiers, filter the interference of unrelated categories of simulation data, and achieve accurate matching of real data. Finally, partial transfer experiments are conducted to verify the effectiveness of the proposed method, and the experimental results show that the diagnostic accuracy of this method is higher than existing transfer learning methods.

**Keywords:** planetary gearbox; fault diagnosis; dynamics simulation; partial transfer learning



**Citation:** Song, M.; Xiong, Z.; Zhong, J.; Xiao, S.; Ren, J. Fault Diagnosis of Planetary Gearbox Based on Dynamic Simulation and Partial Transfer Learning. *Biomimetics* **2023**, *8*, 361. <https://doi.org/10.3390/biomimetics8040361>

Academic Editors: Yudong Zhang, Shuihua Wang and Huiling Chen

Received: 17 July 2023

Revised: 6 August 2023

Accepted: 9 August 2023

Published: 12 August 2023



**Copyright:** © 2023 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

A planetary gearbox (PG) is a key component of rotating machinery. Due to its advantages of large bearing capacity, small volume, and high transmission efficiency, it has been widely applied in mechanical transmission systems in industries such as wind power, aviation, lifting, and transportation. However, PG often works in harsh environments, making it susceptible to malfunctions. If PG fails, it may cause the entire transmission system to degrade and fail, even causing catastrophic damage and huge economic losses [1]. Therefore, researching fault diagnosis methods for PG has important practical significance for ensuring stable operation and prolonging the service life of mechanical equipment [2]. By conducting relevant research and establishing reliable fault diagnosis models, people can detect problems in a timely manner and take corresponding measures. This can not only reduce the maintenance costs of mechanical equipment but also avoid larger faults and losses.

When using machine learning and deep learning methods for PG fault diagnosis, huge quantities of labeled fault data are usually necessary. However, for actual industrial production, most data is collected while the machine is running normally [3], making it challenging to obtain vast and detailed fault data. To address this issue, an abundance of simulated fault data can be gathered through dynamic simulation analysis, and then transfer learning (TL) methods can be used to narrow the disparity between simulated and real data. Thus, the diagnostic information in the simulated data is able to be applied to the fault diagnosis of real PG. Li et al. [4] simulated vibration signals using a lumped

parameter dynamic model and then used a CNN-based TL network to obtain domain-invariant features from various domains in order to classify faults. Dong et al. [5] generated an abundance of simulated data using dynamic models and then used CNN and parameter transfer methods to apply the learned fault diagnosis knowledge to practical scenarios, solving the problem of small samples. Li et al. [6] trained a deep neural network model using computer-simulated data to deal with the challenge caused by an insufficient amount of labeled fault data and used TL to narrow the discrepancy between the simulated and actual domains. Zhu et al. [7] introduced a defect vibration model for simulating fault vibration signals and used real and simulated signals as the target domain and source domain of TL fault diagnosis methods, demonstrating the method's effectiveness and superiority through experimentation. Liu et al. [8] generated simulated vibration signals using a phenomenological model and then used domain adversarial neural networks to train adversarial data among source and target domains. According to the experimental findings, this means can produce excellent classification accuracy with just relatively little real data.

Diagnostic methods based on dynamic simulation and TL usually presume that the label space of simulation data and real data are identical. But when this method is applied to real planetary gear fault diagnosis, simulation data can contain all possible fault categories while the real planetary gearbox may only have one or a few faults, which means that the label space of real data is a subset of the label space of simulation data. This can lead to interference from outlier label space in simulation data, causing mismatching. Partial transfer learning (PTL) methods can help reduce mismatching. Wang et al. [9] proposed a balanced adversarial domain adaptive network for fault diagnosis tasks in partial transfer scenarios, which alleviated the mismatching problem by introducing balancing strategies and class-level weights. Li et al. [10] proposed a class-weighted adversarial neural network that encourages positive transfers of shared classes and ignores source class outliers through class-weighting strategies. Sun et al. [11] suggested a game theory-enhanced domain adaptation network to solve partial domain adaptation problems. The network constructs three attention matrices using maximum mean discrepancy, Jensen-Shannon divergence, and Wasserstein distance and generates the best probability weight through the combination of game theory weights, thereby filtering out irrelevant source domain samples and improving mechanical fault diagnosis performance. Li et al. [12] suggested a new weighted adversarial transfer network that filters out irrelevant source domain samples and improves the performance of the target task through weighted learning. Jiao et al. [13] proposed a domain adaptive network based on classifier inconsistency. The network uses two discriminative 1D-CNN as the basic architecture and promotes active network training by identifying and emphasizing source domain samples with the same classification as the target domain. At the same time, the classifier inconsistency is added in order to direct the model towards learning discriminative and domain-invariant representations for precise classification of unlabeled target data. Kuang et al. [14] proposed a two-stage double-weight consistency-induced partial domain adaptive network. This network obtains double-level composite weights from class-level and sample-level weights through double-weight consistency-induced weighting strategies, enabling selective mapping of source diagnosis knowledge to the target domain.

To deal with the challenge of scarce labeled fault data in real PG fault diagnosis, this paper establishes a dynamic simulation model of PG to obtain abundant fault simulation data. But simulation data and actual data are distinct, and the label space is also heterogeneous. To solve these problems, this paper introduces multiple domain classifiers and weighted learning schemes on the basis of existing domain adversarial TL methods, evaluates the transferability of simulation data, adaptively measures their contributions to label predictor and domain classifiers, filters out the interference of irrelevant categories of simulation data, achieves accurate matching of real data, and thus improves the diagnostic accuracy of transfer tasks.

The main contributions of this paper are as follows:

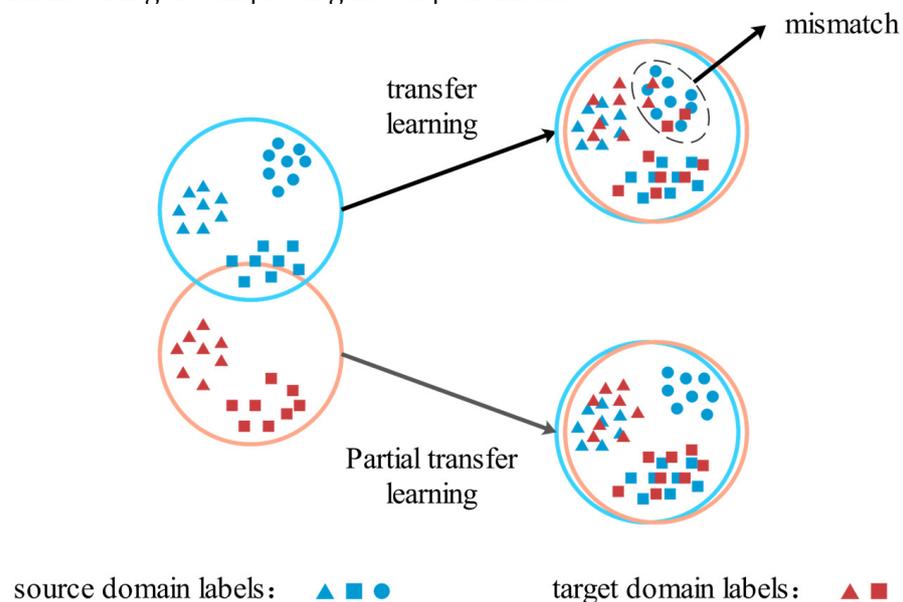
- Through the rigid-flexible coupling dynamic model of PG, a wealth of fault simulation data is obtained, and then the problem of scarcity of labeled fault data in real-world scenarios is solved.
- By introducing multiple domain discriminators and a weighted learning scheme, the interference from simulation data of irrelevant categories is filtered, thereby improving the diagnostic accuracy of partial transfer tasks.

The remainder of this paper is arranged as below: Section 2 introduces the relevant theories. Section 3 describes the proposed means. Section 4 studies a practical case. Section 5 summarizes this paper.

## 2. Theoretical Background

### 2.1. Partial Transfer Learning

In TL, it is often assumed that the label space of the source domain ( $D_s$ ) samples,  $C_s$ , is the same as that of the label space of the target domain ( $D_t$ ) samples,  $C_t$ . However, in real-world applications,  $C_t$  is more likely to be a subset of  $C_s$ . In this case, all labels in  $D_t$  are shared by both  $D_s$  and  $D_t$ , and  $C_t$  is also known as the shared label space. There are also some labels in  $D_s$  that are unique to it, known as the outlier label space ( $C_s - C_t$ ), which can lead to mismatches between  $D_s$  and  $D_t$  samples and affect the accuracy of the transfer task. PTL aims to promote the positive transfer of samples in the shared label space while suppressing the negative transfer of samples in the outlier label space when  $C_t$  is a subset of  $C_s$ , thereby improving the accuracy of the transfer task [15]. Figure 1 illustrates the concept behind traditional TL and PTL. In the figure,  $D_s$  samples have three types of labels:  $\triangle$ ,  $\circ$ , and  $\square$ , while  $D_t$  samples only have two types of labels:  $\triangle$  and  $\square$ . In this case,  $\circ$  in  $D_s$  is an outlier label, which may lead to mismatching with  $D_t$  samples during TL. However, PTL methods can recognize and filter out outlier labels in  $D_s$ , effectively reducing the risk of mismatching and improving model performance.



**Figure 1.** Schematic diagram of TL and PTL.

### 2.2. Residual Neural Network

For deep neural networks, the number of layers is crucial. The deeper the network, the richer its ability to extract hierarchical features and its recognition and classification capabilities are also enhanced. However, for traditional CNN, too many layers can cause gradient vanishing or explosion, making the network difficult to train. To address the issue, He et al. [16] proposed Residual Neural Networks (ResNet) in 2015. ResNet usually includes convolutional layers, pooling layers, residual blocks, and fully connected

layers. Figure 2 illustrates a schematic diagram of residual blocks, where  $x$  is the input,  $H(x) = F(x) + x$  is the output, and  $F(x) = H(x) - x$  is the residual. The residual block has two branches, the residual branch, and the identity mapping branch. The residual branch consists of two convolutional layers, which are used to fit the residual  $F(x)$ , while the identity mapping branch keeps the input  $x$  unchanged. The output  $H(x)$  of the residual block is obtained by element-wise addition of the two branches and then passed through the ReLu activation function. The introduction of identity mapping ensures that the performance of deep networks is not worse than that of shallow networks, and no additional parameters or computation complexity are added.

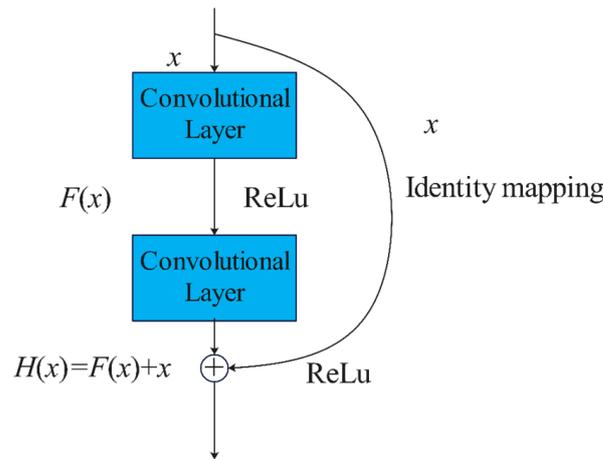


Figure 2. Schematic diagram of residual block.

### 2.3. Domain Adversarial Neural Network

Domain Adversarial Neural Networks (DANNs) have been extensively implemented in TL. Through the adversarial learning process, the network is capable of extracting domain-invariant features from both  $D_s$  and  $D_t$ . The adversarial learning process is able to be seen as a two-player game, with the first player being a domain classifier  $G_d$  taught to differentiate between  $D_s$  and  $D_t$  features, and the second player is a feature extractor  $G_f$  trained to confuse  $G_d$ . The framework of DANN is shown in Figure 3.

To obtain domain-invariant features, during the training process of DANN, the parameters  $\theta_f$  of  $G_f$  are learned by maximizing the loss of the  $G_d$ . The parameters  $\theta_d$  of  $G_d$  are learned by minimizing its loss. In addition, minimizing the loss of the label predictor  $G_y$  ensures a low  $D_s$  classification error. The overall loss function of DANN is shown in Equation (1) [17]:

$$L(\theta_f, \theta_y, \theta_d) = \frac{1}{n_s} \sum_{x_i \in D_s} L_y(G_y(G_f(x_i)), y_i) - \frac{\lambda}{n_s + n_t} \sum_{x_i \in D_s \cup D_t} L_d(G_d(G_f(x_i)), d_i) \tag{1}$$

In the formula,  $L_y$  is the loss function of  $G_y$ ,  $L_d$  is the loss function of  $G_d$ ,  $d_i$  is the domain label of the  $i$ -th sample, and  $\lambda$  is the hyperparameter that balances  $L_y$  and  $L_d$ . The parameter optimization for DANN is:

$$(\hat{\theta}_f, \hat{\theta}_y) = \underset{\theta_f, \theta_y}{\operatorname{argmin}} L(\theta_f, \theta_y, \theta_d) \tag{2}$$

$$(\hat{\theta}_d) = \underset{\theta_d}{\operatorname{argmax}} L(\theta_f, \theta_y, \theta_d) \tag{3}$$

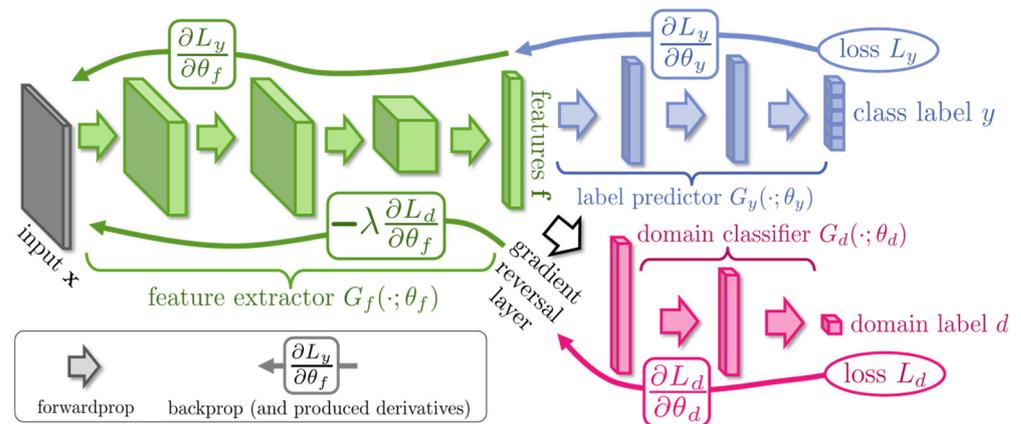


Figure 3. DANN framework.

### 3. Proposed Method

#### Weighted Domain Adversarial Neural Network Diagnostic Model

When  $D_t$  label space  $C_t$  is a subset of  $D_s$  label space  $C_s$ , if existing TL fault diagnosis models are used,  $D_t$  samples may be incorrectly matched with samples belonging to the outlier label space  $C_s - C_t$  in  $D_s$ , resulting in reduced diagnostic accuracy. To deal with this problem, this paper suggests a domain adversarial neural network with a weighted learning strategy to promote the positive transfer of the shared label space  $C_t$  and suppress the negative transfer of the outlier label space  $C_s - C_t$  in  $D_s$ . The network framework is shown in Figure 4.

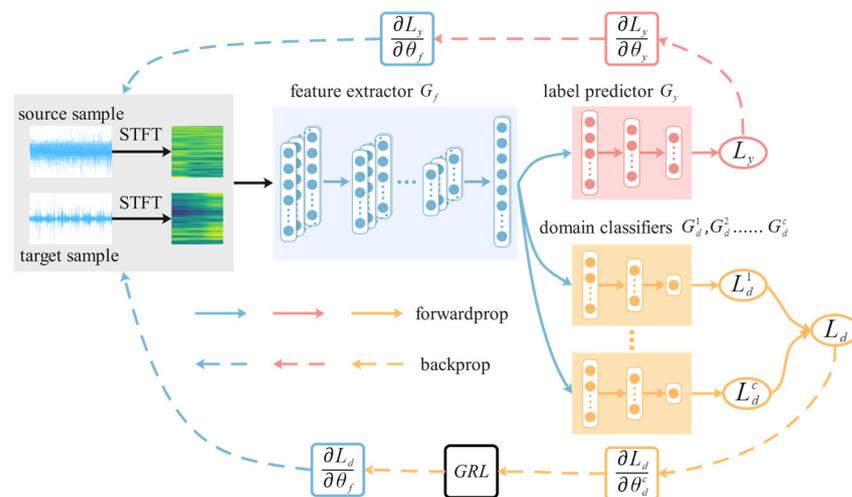


Figure 4. Weighted domain adversarial neural network framework.

This network includes a feature extractor  $G_f$ , a label predictor  $G_y$ , and  $|C_s|$  domain classifiers  $G_d$ .  $G_f$  and  $G_y$  are composed of ResNet-18.  $G_f$  is the feature extraction part of ResNet-18, including convolutional layers, pooling layers, and residual blocks. To extract more effective features, CBAM [18] is added to each residual block.  $G_y$  corresponds to the output of ResNet-18 and includes a fully connected layer and a softmax classification layer.  $|C_s|$  domain classifiers  $G_d$  have the same structure, including three fully connected layers. The  $c$ -th domain classifier  $G_d^c$  ( $c = 1, 2, \dots, |C_s|$ ) is responsible for matching the  $D_s$  sample with label  $c$  and the  $D_t$  sample with label  $c$ . Therefore, in  $G_d^c$ , samples with label  $c$  should be assigned larger weights, while samples with other labels should be assigned smaller weights. In addition, only the domain classifier is accountable for matching the shared label space  $C_t$  can promote positive transfer, while the domain classifier accountable for matching the outlier label space  $C_s - C_t$  will introduce noise. Therefore, it is necessary

to reduce the weight of the domain classifiers responsible for matching the outlier label space  $C_s-C_t$ .

As the labeled samples in  $D_t$  are unknown during model training, it is not possible to determine the weights based on labels. Joint distribution adaptation (JDA) [19] is often used to calculate differences between samples, and this paper will calculate weights using JDA values. The calculation formula for the weight matrix  $W_d$  assigned to the domain classifier is shown in Equations (4)–(6):

$$j_i^c = \begin{cases} j_i^{sc} : \text{mean} \left( \sum_{x_j \in D_s^c} JDA(G_f(x_j), G_f(x_i)) \right), x_i \in D_s \\ j_i^{tc} : \text{mean} \left( \sum_{x_j \in D_s^c} JDA(G_f(x_j), G_f(x_i)) \right), x_i \in D_t \end{cases} \tag{4}$$

$$y_c = \frac{1 / \text{mean} \left( \sum_{i=1}^{n_t} j_i^{tc} \right)}{\sum_{c=1}^{|C_s|} \left( 1 / \text{mean} \left( \sum_{i=1}^{n_t} j_i^{tc} \right) \right)} \tag{5}$$

$$W_d = [y_1, y_2, \dots, y_{|C_s|}] \tag{6}$$

In the formula,  $\text{mean}(\cdot)$  calculates the average value,  $G_f(\cdot)$  represents the extracted features by the feature extractor, and  $JDA(\cdot)$  represents the JDA value used to measure the difference between the two samples. If the JDA value is small, it indicates that the difference between the two samples is small, and there is a significant possibility that they belong to the same label.  $j_i^{sc}$  represents the discrepancy between the  $i$ -th  $D_s$  sample and  $D_s$  sample with label  $c$ , and  $j_i^{tc}$  represents the discrepancy between the  $i$ -th  $D_t$  sample and  $D_s$  sample with label  $c$ . By calculating the discrepancy between each  $D_t$  sample and  $D_s$  sample with label  $c$ , the probability  $y_c$  of label  $c$  belonging to the shared label space  $C_t$  can be obtained. Since the labels in the outlier label space,  $C_s-C_t$ , do not belong to  $C_t$ , their probabilities  $y_c, c \in C_s-C_t$  are small enough to reduce the weight of the domain classifiers responsible for  $C_s-C_t$ .

The calculation formula for the weight matrix  $W_s$  assigned to the samples is shown in Equations (7) and (8):

$$s_i^c = \begin{cases} s_i^{sc} : \frac{1/j_i^{sc}}{\sum_{c=1}^{|C_s|} 1/j_i^{sc}} \\ s_i^{tc} : \frac{1/j_i^{tc}}{\sum_{c=1}^{|C_s|} 1/j_i^{tc}} \end{cases} \tag{7}$$

$$W_s = \begin{bmatrix} s_1^1 & s_1^2 & \dots & s_1^{|C_s|} \\ s_2^1 & s_2^2 & \dots & s_2^{|C_s|} \\ \dots & \dots & \dots & \dots \\ s_{n_s+n_t}^1 & s_{n_s+n_t}^2 & \dots & s_{n_s+n_t}^{|C_s|} \end{bmatrix} \tag{8}$$

In the formula,  $s_i^{sc}$  is the probability that the label of the  $i$ -th  $D_s$  sample is  $c$ , while  $s_i^{tc}$  is the probability that the label of the  $i$ -th  $D_t$  sample is  $c$ .

After incorporating  $W_d$  and  $W_s$ , the total loss function of the model is outlined below:

$$\begin{aligned} L(\theta_f, \theta_y, \theta_d^c |_{c=1}^{|C_s|}) &= \frac{1}{n_s} \sum_{x_i \in D_s} L_y(G_y(G_f(x_i)), y_i) \\ &\quad - \frac{\lambda}{n_s} \sum_{c=1}^{|C_s|} \left( \sum_{x_i \in D_s} y_c L_d^c(G_d^c(s_i^{sc} G_f(x_i)), d_i) \right) \\ &\quad - \frac{\lambda}{n_t} \sum_{c=1}^{|C_s|} \left( \sum_{x_i \in D_t} y_c L_d^c(G_d^c(s_i^{tc} G_f(x_i)), d_i) \right) \end{aligned} \tag{9}$$

In the equation,  $\theta_f$  represents the parameters of  $G_f$ ,  $\theta_y$  represents the parameters of the  $G_y$ ,  $\theta_d^c$  represents the parameters of  $G_d^c$ ,  $L_y$  represents the loss function of the  $G_y$ ,

$L_d^c$  represents the loss function of  $G_d^c$ ,  $d_i$  represents the domain label of the  $i$ -th sample, and  $\lambda$  is a hyperparameter that balances  $L_y$  and  $L_d$ .

The optimization of the model parameters is as follows:

$$\left(\hat{\theta}_f, \hat{\theta}_y\right) = \operatorname{argmin}_{\theta_f, \theta_y} L\left(\theta_f, \theta_y, \theta_d^c \Big|_{c=1}^{|C_s|}\right) \quad (10)$$

$$\left(\hat{\theta}_d^1, \dots, \hat{\theta}_d^{|C_s|}\right) = \operatorname{argmax}_{\theta_d^1, \dots, \theta_d^{|C_s|}} L\left(\theta_f, \theta_y, \theta_d^c \Big|_{c=1}^{|C_s|}\right) \quad (11)$$

Compared with a single domain classifier, the multiple domain classifiers used in this paper have two advantages: (1) by using the weight matrix  $W_d$  of the domain classifiers, the model can emphasize the domain classifiers responsible for the shared label space and suppress the ones responsible for the outlier label spaces, thereby reducing the negative impact of outlier label spaces. (2) The sample weight matrix  $W_s$  allows  $D_i$  samples to only align with  $D_s$  samples of one or multiple most relevant labels, thus reducing mismatching.

## 4. Experiment and Analysis

### 4.1. Dataset Comparison and Analysis

The real data for the PG comes from the Drivetrain Diagnostics Simulator (DDS), which is a comprehensive experimental platform for diagnosing power transmission faults. Figure 5 shows the physical model of the DDS experiment platform. During data collection, the variable-speed drive motor has three speeds: 20 Hz, 30 Hz, and 40 Hz, and the magnetic brake has three currents: 0 A, 0.4 A, and 0.8 A (by adjusting the current of the magnetic brake, various loads can be transferred to the output shaft). The sampling frequency is 12,800 Hz.

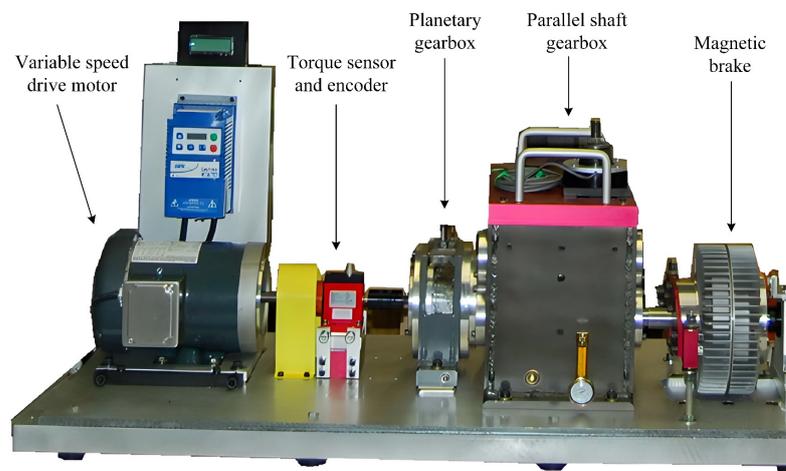
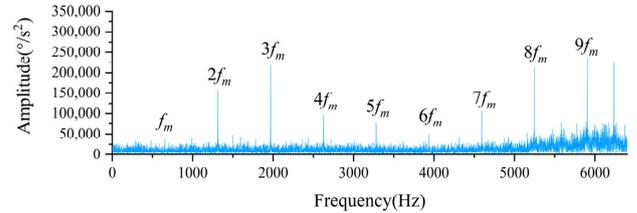
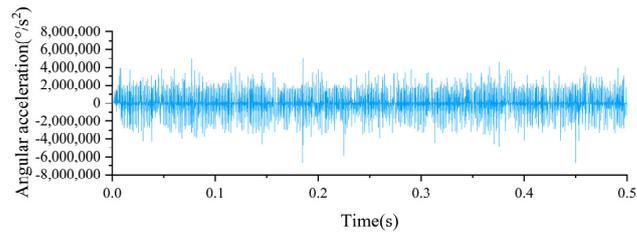
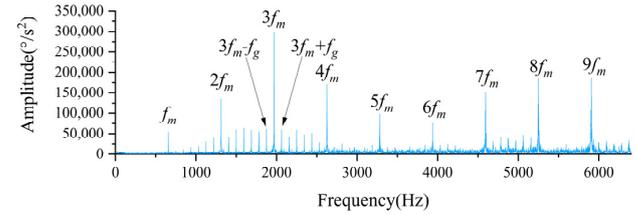
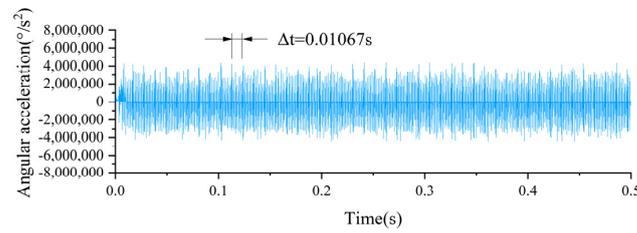


Figure 5. DDS experimental platform.

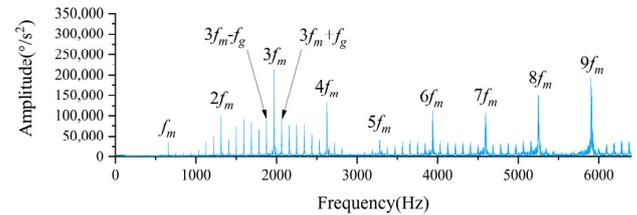
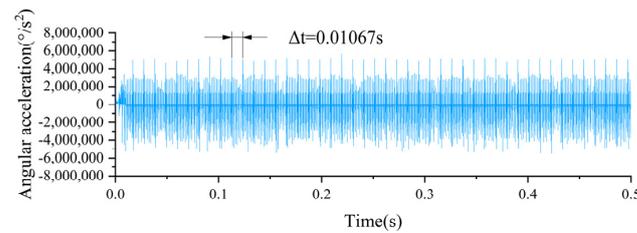
The simulated data for the PG comes from our previous article, where a rigid-flexible coupled model was established [20]. We used this model to obtain simulation data for four different health conditions of the PG: sun gear broken tooth fault (BR), sun gear crack fault (CR), sun gear tooth missing fault (MI), and normal sun gear (NO). During data acquisition, the input shaft speed was 30 Hz with no load, and the simulation time was 10 s with a simulation step of 128,000. This is equivalent to a working conditions of 30 Hz 0 A for the simulation data, with a sampling frequency of 12,800 Hz. In addition, in our previous article, we also analyzed the effect of the simulation step size on the simulation data, and the results are shown in Figure 6 [20], where  $\Delta t$  represents the time interval between two impacts,  $f_m$  represents the meshing frequency, and  $f_g$  represents the fault frequency. It can be seen from Figure 6 that the smaller the simulation step size, the more obvious the periodic shock in the time domain diagram and the sideband in the frequency domain diagram, and they are all consistent with the calculated values of the theoretical formula, which verifies the simulation model and simulation data plausibility.



(a) The simulation step size is  $7.8125 \times 10^{-5}$  s



(b) The simulation step size is  $1.5625 \times 10^{-5}$  s



(c) The simulation step size is  $8.6806 \times 10^{-6}$  s

**Figure 6.** Time domain diagrams and frequency domain diagrams of different simulation step sizes of rigid-flexible coupling model.

The rigid-flexible coupling model has simplified the PG of the DDS experimental platform to a large extent, and the parameter settings in the model are difficult to completely match with the actual PG. This results in a discrepancy between the simulation and real data, even though the simulated data agrees with the theory. In order to more intuitively demonstrate this difference, this section will analyze and compare the simulation and real data from the perspectives of time-domain, frequency-domain, and probability distribution.

(1) Comparison and analysis of time domain diagrams

Figure 7 shows the time-domain plots of simulated data and real data, both of which were obtained under a 30 Hz 0 A operating condition. It can be observed that there is no clear periodic impulse in either the simulated data or the real data due to a lack of sufficiently high sampling frequency. However, the real data exhibits amplitude modulation, while the simulated data does not. This is because the real data was collected using a fixed-position vibration sensor, while in PG, the planetary gear rotates, causing the distance between the planetary gear and sensor to change. When the planetary gear is closer to the sensor, the measured meshing vibration is larger, and when it is farther away, it is smaller, resulting in amplitude modulation in the time-domain waveform. In contrast, the simulated data measures the angular acceleration of the planetary carrier, which is less affected by the position of the planetary gear and hence does not exhibit amplitude modulation.

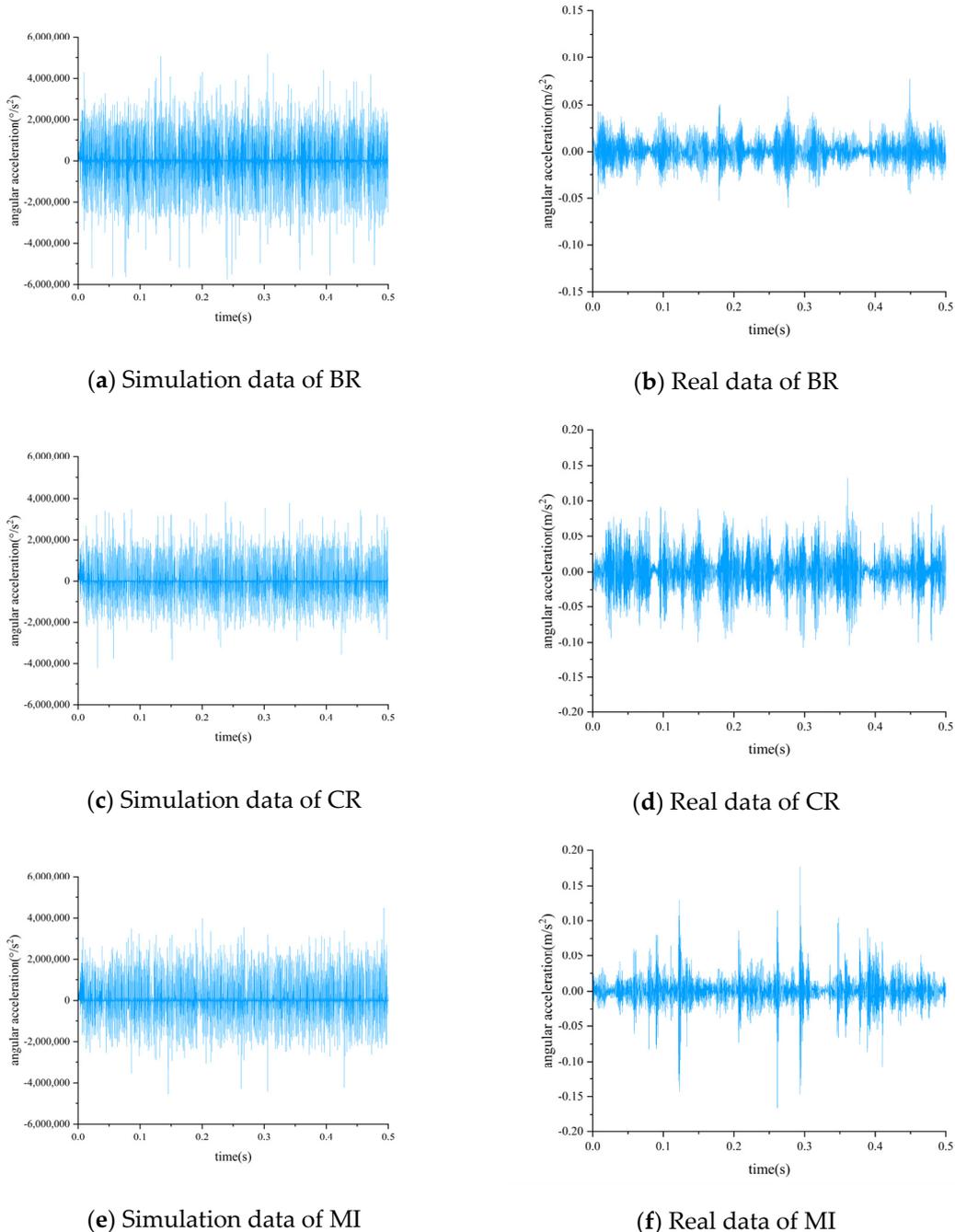


Figure 7. Cont.

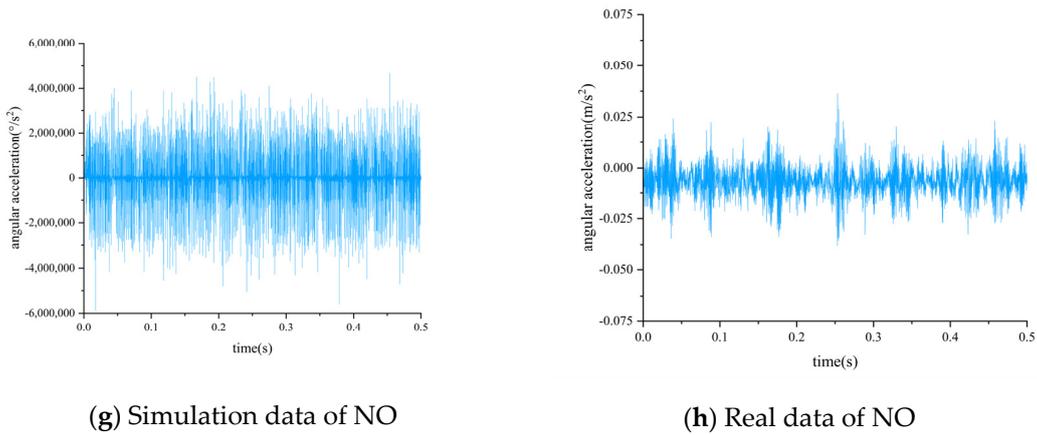


Figure 7. Comparison of simulation data and real data time-domain diagram.

(2) Comparison and analysis of frequency domain diagrams

Figure 8 shows the frequency domain comparison of the simulated and real data, both of which were obtained under the same working conditions of 30 Hz 0 A. The frequency domain spectra of both simulated and real data exhibit obvious meshing frequencies and their low harmonics, indicating similar vibration characteristics between the two groups of data. However, the real data does not show higher harmonics of the meshing frequency, which may be due to environmental noise and other factors affecting the high-frequency components of the frequency domain spectra during data acquisition.

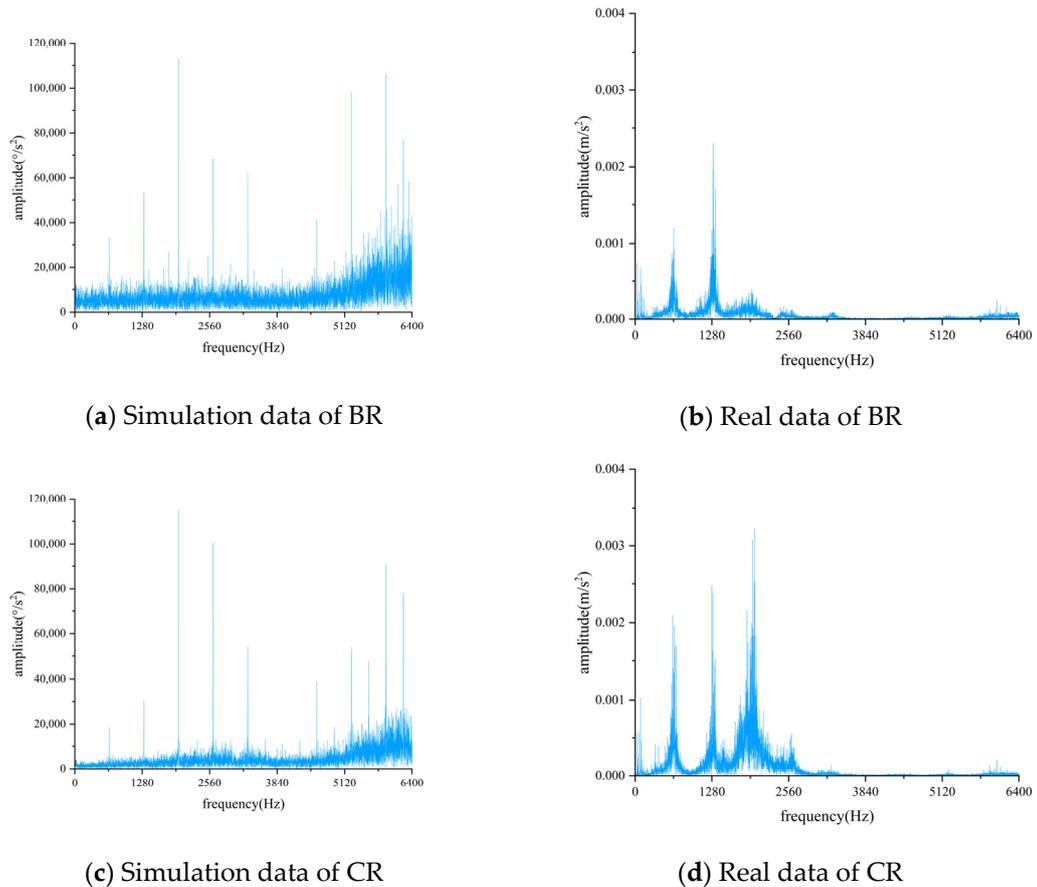
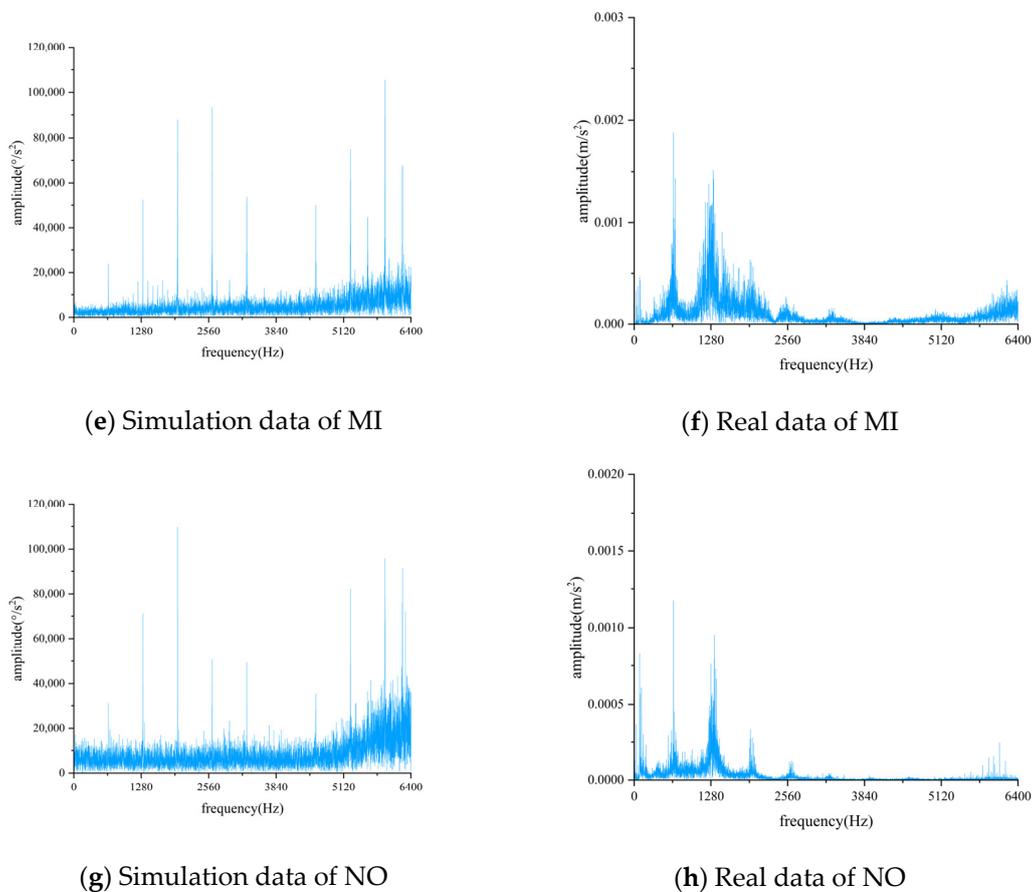


Figure 8. Cont.



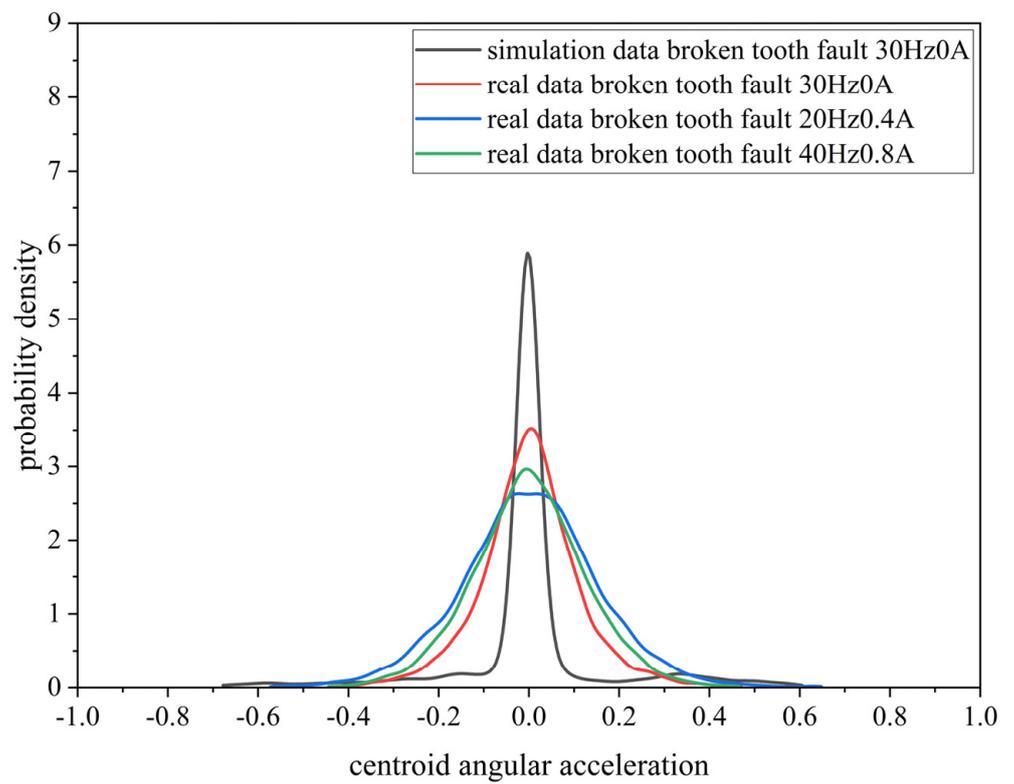
**Figure 8.** Comparison of simulation data and real data frequency domain diagram.

### (3) Comparison and analysis of probability distribution

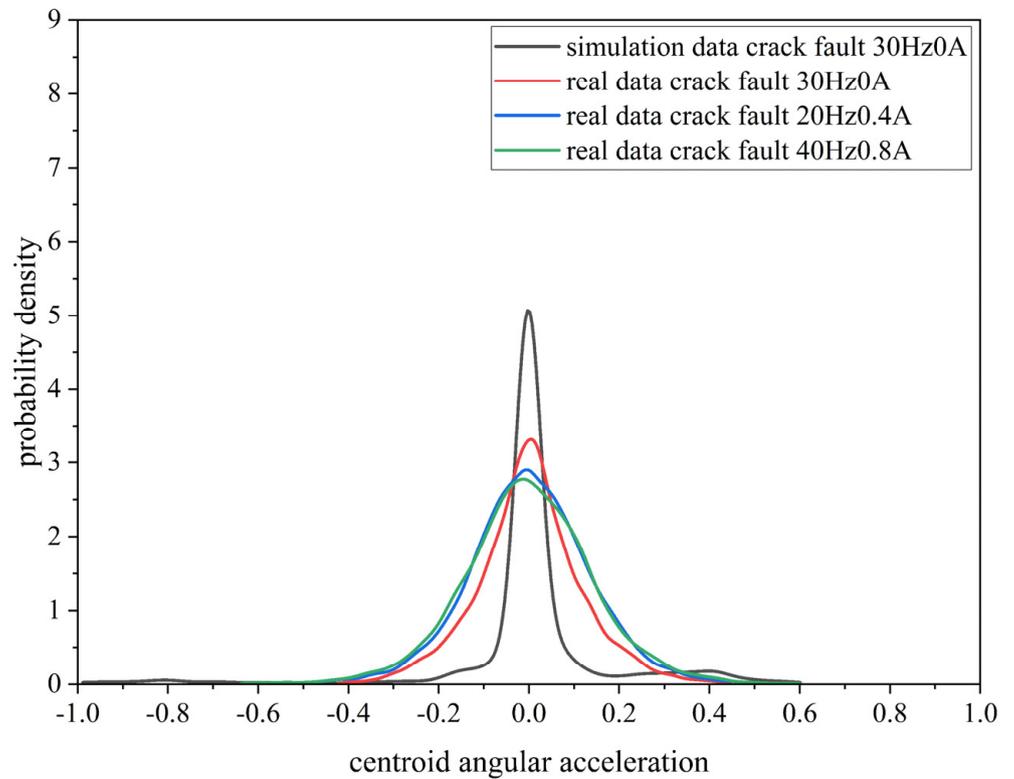
From the simulated data with a frequency of 30 Hz and a current of 0 A, as well as the real data from different conditions (30 Hz 0 A, 20 Hz 0.4 A, 40 Hz 0.8 A), 12,800 data points were selected and normalized to  $[-1,1]$ , resulting in the probability distribution curves shown in Figure 9. Figure 9 illustrates that the probability distribution curve of the simulated data is more concentrated and has a higher peak compared to the real data. This indicates that the probability distribution of the simulated data and the real data differ significantly. In addition, the probability distribution curves of the real data from different conditions are relatively similar, indicating that the difficulty of transfer between the simulated and real data is higher than the difficulty of transfer between real data from various working condition.

#### 4.2. Dataset Description

By using overlapping sampling, 512-length data segments were extracted with a re-sampling step of 50 from both simulated and real data. Then, Short-Time Fourier Transform (STFT) was applied to convert these segments into time-frequency images with a size of 96 pixels in both width and height. There are 2400 time-frequency images for each health condition of simulated and real data, of which 2000 are utilized for training, and the remaining 400 are utilized for testing. Figure 10 shows the time-frequency images of simulated data. After obtaining the time-frequency images, some partial transfer tasks were designed, as shown in Tables 1 and 2. In Table 1, the real data of 30 Hz 0 A is used as  $D_s$ , while the real data of other working conditions are used as  $D_t$ . In Table 2, the simulated data of 30 Hz 0 A is used as  $D_s$ , while the real data are used as  $D_t$ .  $D_s$  in Tables 1 and 2 both contain four health conditions: BR, CR, MI, and NO.

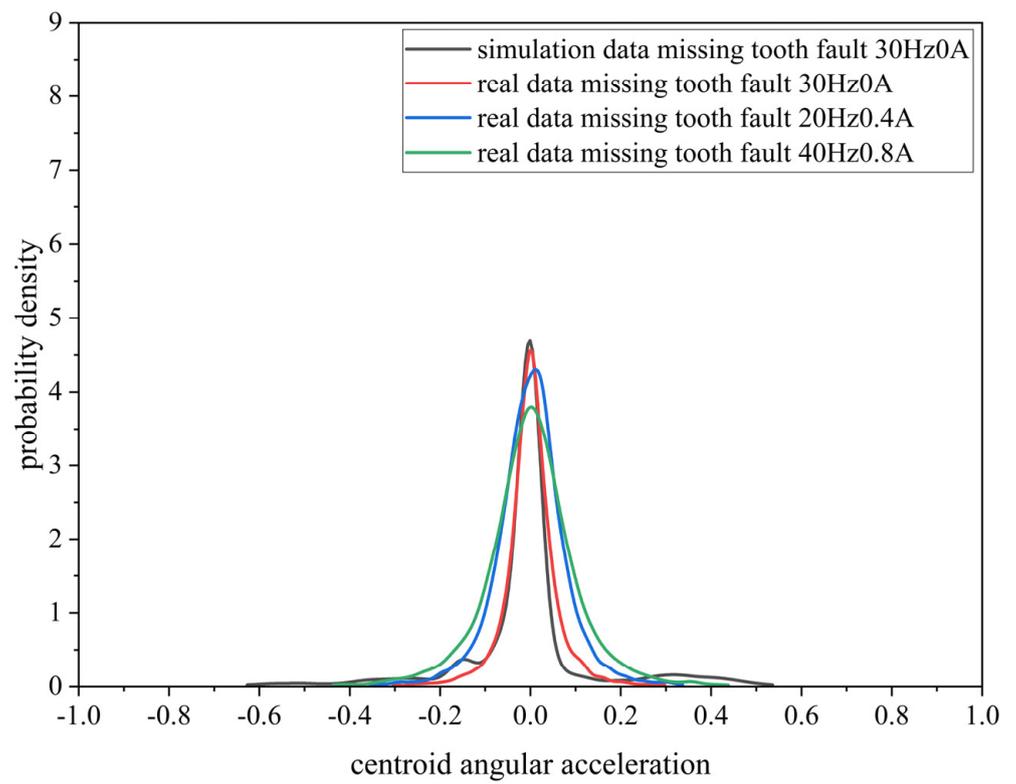


(a) BR

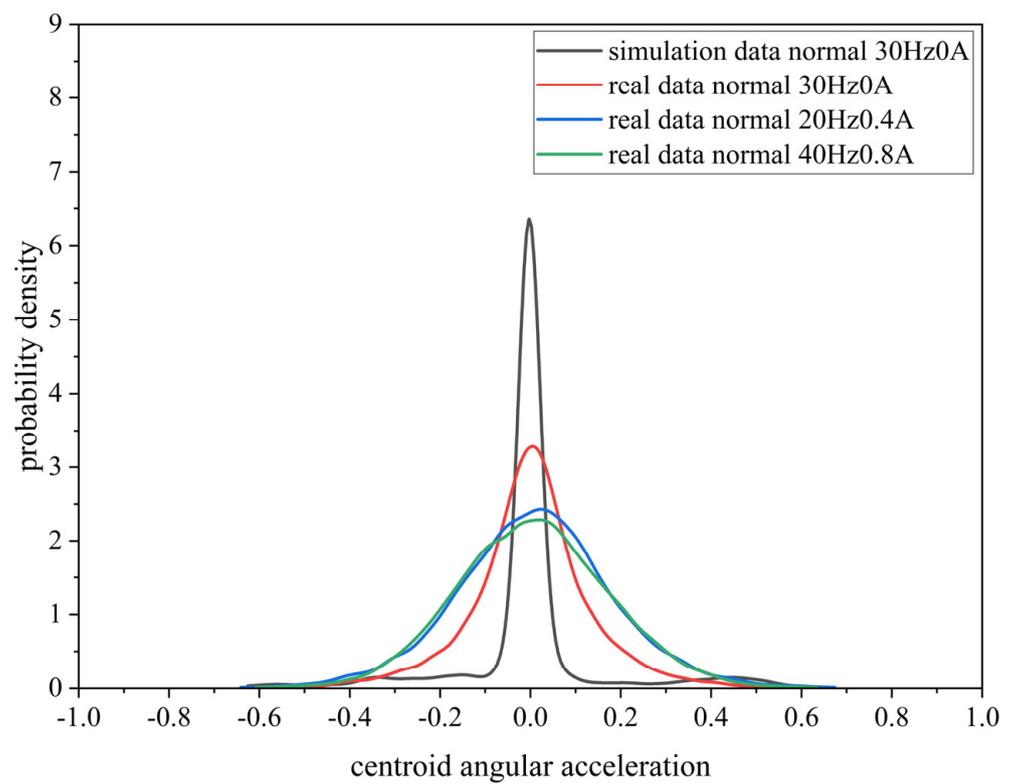


(b) CR

Figure 9. Cont.

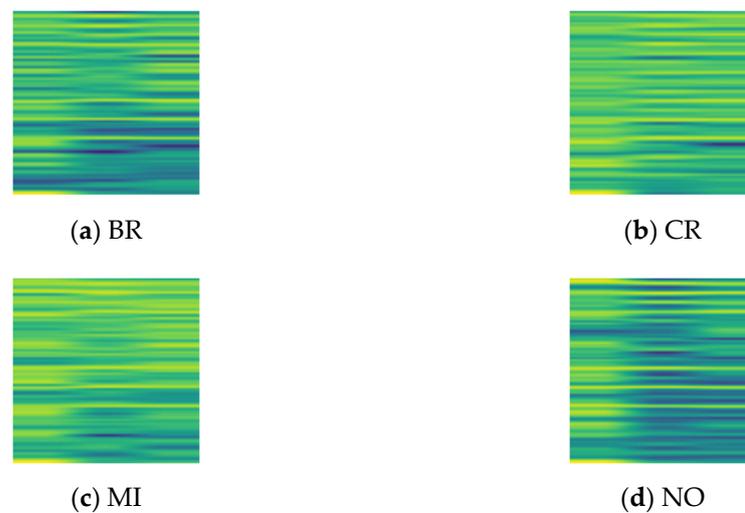


(c) MI



(d) NO

Figure 9. Probability distribution curves of simulated data and real data.



**Figure 10.** Time-frequency diagram of simulation data for different health conditions.

**Table 1.** Partial transfer tasks where both  $D_s$  and  $D_t$  are real data.

Task Name	$D_t$ Conditions	$D_t$ Health Conditions
$C_1$	30 Hz 0.8 A	BR, CR, MI
$C_2$	20 Hz 0 A	BR, CR
$C_3$	20 Hz 0.4 A	CR, NO
$C_4$	40 Hz 0.8 A	CR

**Table 2.** Partial transfer tasks where  $D_s$  is simulation data, and  $D_t$  is real data.

Task Name	$D_t$ Conditions	$D_t$ Health Conditions
$C_5$	30 Hz 0 A	BR, CR, NO
$C_6$	30 Hz 0.8 A	BR, CR, MI
$C_7$	20 Hz 0 A	BR, CR
$C_8$	20 Hz 0.4 A	CR, NO
$C_9$	40 Hz 0.8 A	CR

#### 4.3. Result Comparison

To demonstrate the effectiveness of the proposed method, we compared it with ResNet [16], DeepCoral [21], DDC [22], and DANN [23]. To make an impartial comparison, all methods used the same ResNet network structure and parameters as the proposed method.

Figures 11 and 12 illustrate the diagnostic accuracies of various methods. It can be observed that when both  $D_s$  and  $D_t$  are real data, the proposed method obtains a mean diagnostic precision of 98.02%. When  $D_s$  is simulated data and  $D_t$  is real data, as analyzed in Section 4.1, the transfer difficulty is significantly increased. Nevertheless, the proposed means still achieves a mean diagnostic precision of 83.83%, indicating its practical value. Furthermore, the proposed means outperforms other TL methods in all transfer tasks. This is because the proposed method introduces multiple domain classifiers and a weighted learning strategy, which enables the model to effectively measure the transferability of each label's  $D_s$  sample to  $D_t$  and increase the contribution of shared label  $D_s$  samples and decrease the contribution of outlier label  $D_s$  samples during training. This effectively reduces the mismatching between  $D_t$  samples and outlier label  $D_s$  samples, thereby improving the diagnostic accuracy of transfer tasks.

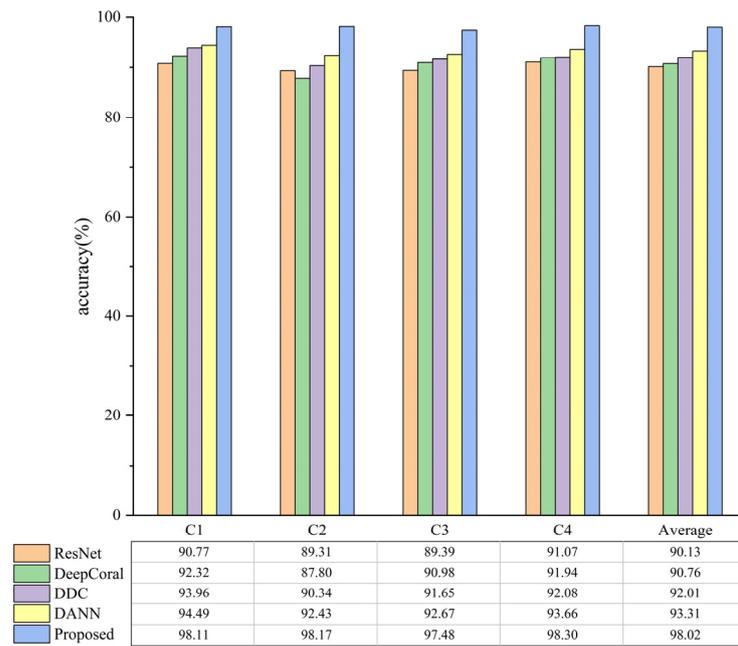


Figure 11. Partial transfer diagnostic accuracy with real data in both  $D_s$  and  $D_t$ .

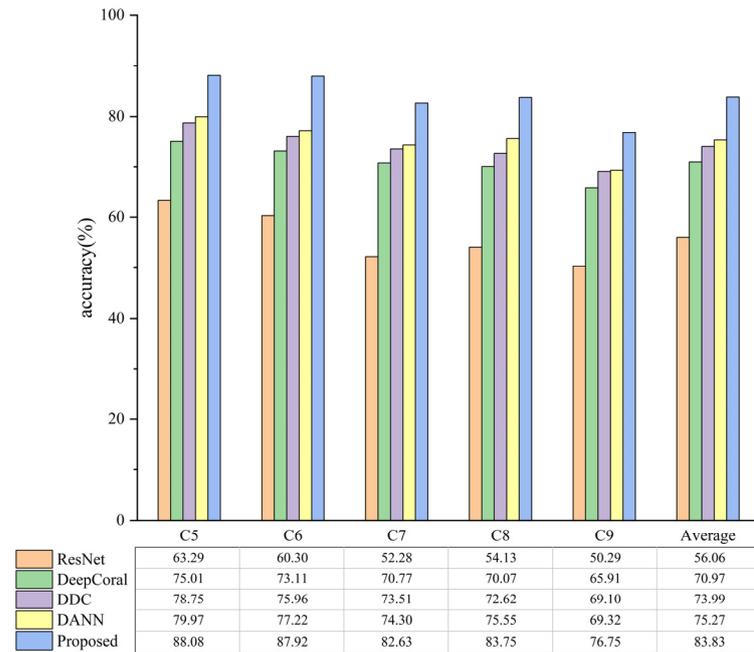
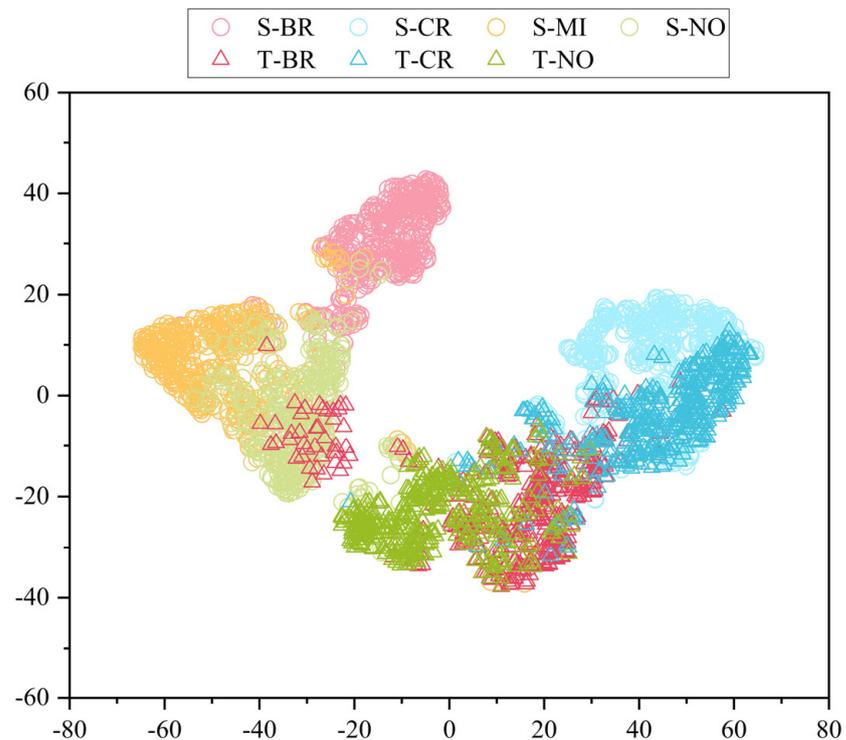


Figure 12. Partial transfer diagnostic accuracy with  $D_s$  being simulated data and  $D_t$  being real data.

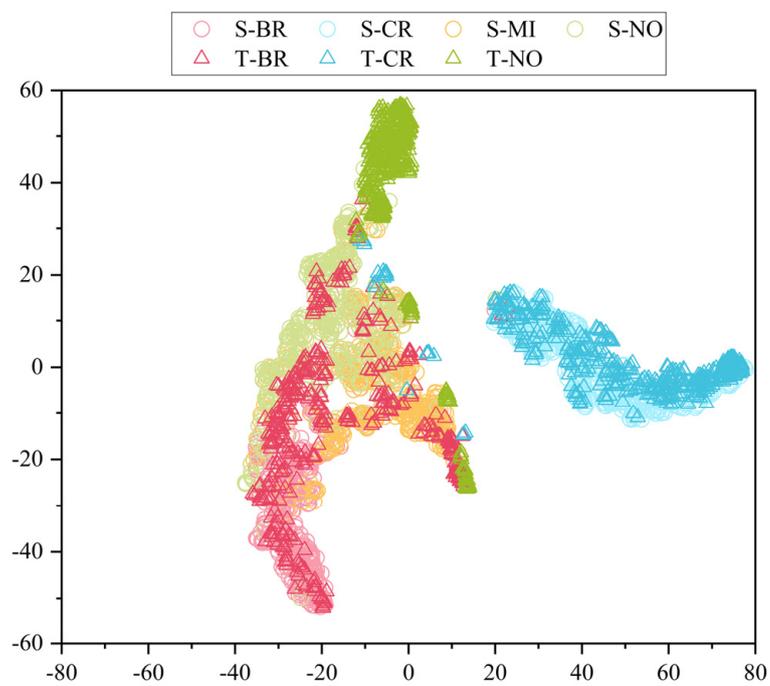
#### 4.4. Feature Visualization Analysis

To clearly demonstrate the feature distribution when simulation data is used as  $D_s$  and real data as  $D_t$ , the features extracted by various methods were visualized using t-SNE. Figure 13 depicts the feature visualization of different methods in task C5. In this task, there are four labels in  $D_s$  samples, while there are only three labels in  $D_t$  samples, and MI in  $D_s$  belongs to the outlier label. From Figure 13, it can be seen that ResNet can accurately distinguish  $D_s$  features of different labels, but the distribution of  $D_t$  features it extracts is quite different from that of  $D_s$  features, which leads to lower accuracy in classifying  $D_t$  samples. In comparison,  $D_s$  features and  $D_t$  features extracted by DeepCoral, DDC, and DANN have a more similar distribution, but there is a large overlap between features of different labels, and some  $D_t$  features are incorrectly aligned with  $D_s$  MI features. In the

proposed method,  $D_t$  features can be correctly aligned with the corresponding  $D_s$  features of the label, and the discriminability between features of different labels is higher, which further validates the effectiveness of the proposed means.

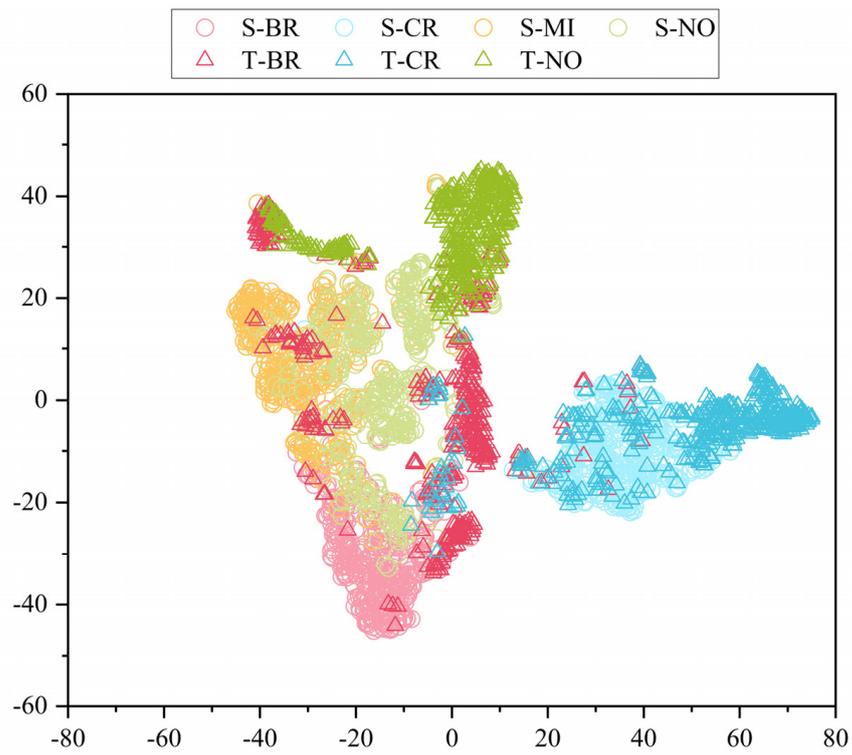


(a) ResNet

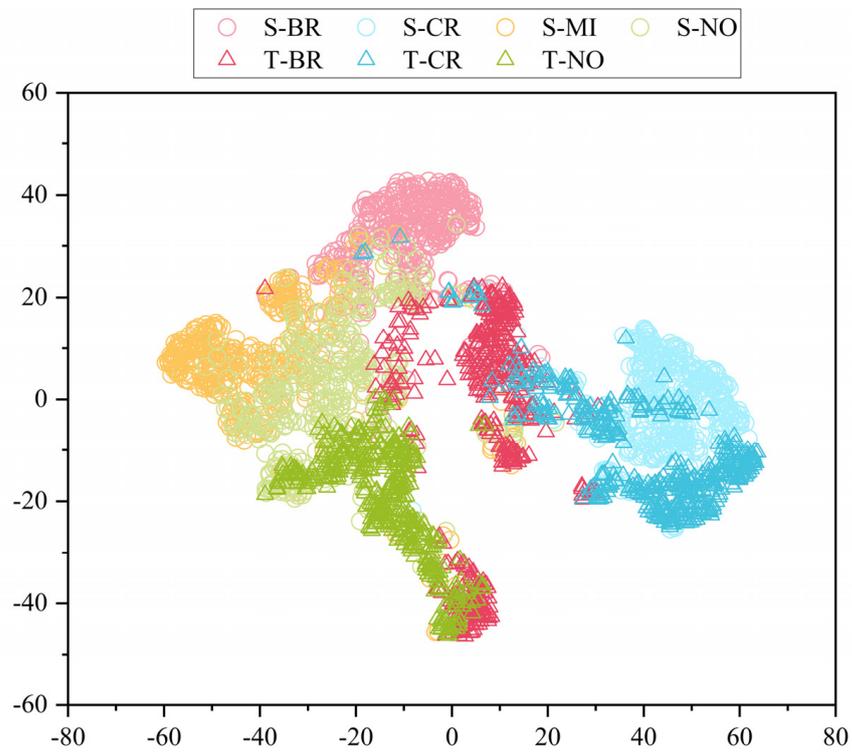


(b) DeepCoral

Figure 13. Cont.

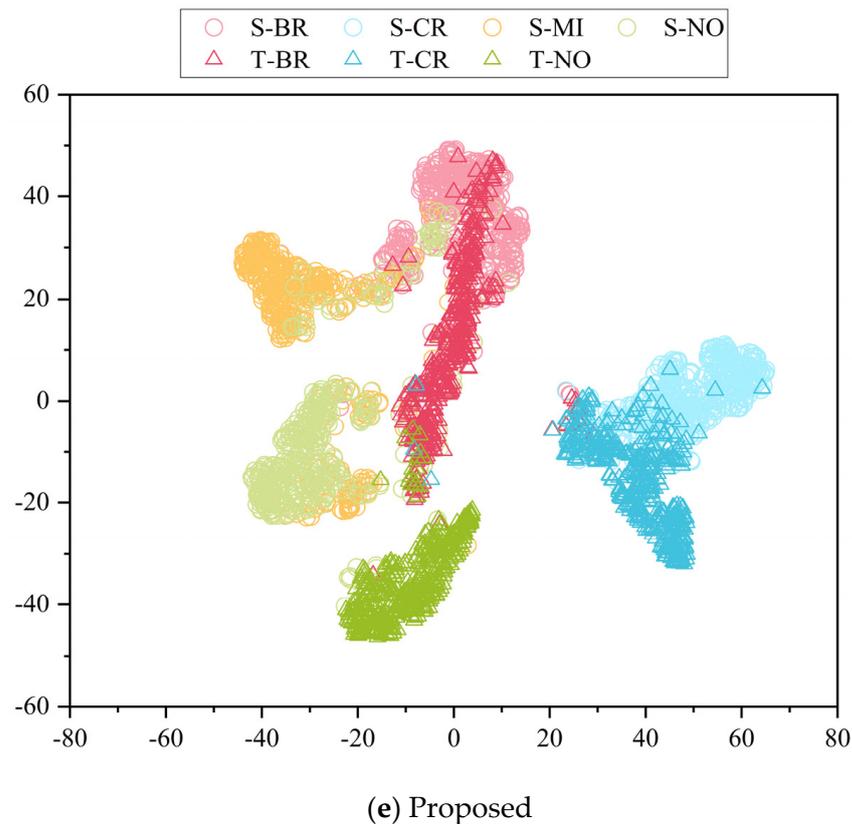


(c) DDC



(d) DANN

Figure 13. Cont.



**Figure 13.** Feature visualization of different methods in partial transfer task C1.

## 5. Conclusions

This paper puts forward a weighted domain adversarial neural network diagnostic model aimed at improving the fault diagnosis performance of PG in partial transfer tasks. Unlike traditional domain adaptation diagnostic methods that directly adapt all  $D_s$  and  $D_t$  class samples, this method considers the influence of outlier label  $D_s$  samples. Specifically, this method uses multiple domain classifiers, each of which is responsible for matching samples of a certain label. And a weighting scheme is introduced to assign smaller weights to outlier label  $D_s$  samples and domain classifiers responsible for matching outlier label source domain samples, effectively reducing the negative impact of outlier label  $D_s$  samples and promoting correct matching of shared labeled  $D_s$  samples and  $D_t$  samples. When both  $D_s$  and  $D_t$  are real data, this means achieved an average diagnostic accuracy of 98.02%; when  $D_s$  is simulated data and  $D_t$  is real data, this method achieved an average diagnostic accuracy of 83.83%, both of which are better than other TL methods. In addition, this method relaxes the requirement that  $D_s$  and  $D_t$  need to have the same label space, which is more in line with practical application scenarios.

**Author Contributions:** Conceptualization, M.S. and S.X.; software, Z.X.; validation, J.Z., J.R.; writing—original draft preparation, Z.X.; writing—review and editing, S.X. and J.Z. All authors have read and agreed to the published version of the manuscript.

**Funding:** This paper was funded by the following research projects: The young and middle-aged science and technology project of Ningde Normal University (Grant No. 2022ZQ102) and the Collaborative innovation center project of Ningde Normal University (Grant No. 2023ZX01).

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

**Data Availability Statement:** The datasets generated during and/or analyzed during the current study are available from the corresponding author on reasonable request.

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

## References

1. Feng, Z.; Zhu, W.; Zhang, D. Time-Frequency demodulation analysis via Vold-Kalman filter for wind turbine planetary gearbox fault diagnosis under nonstationary speeds. *Mech. Syst. Signal Process.* **2019**, *128*, 93–109. [[CrossRef](#)]
2. He, Z.; Shao, H.; Cheng, J.; Zhao, X.; Yang, Y. Support tensor machine with dynamic penalty factors and its application to the fault diagnosis of rotating machinery with unbalanced data. *Mech. Syst. Signal Process.* **2020**, *141*, 106441. [[CrossRef](#)]
3. Kwak, J.; Lee, T.; Kim, C.O. An incremental clustering-based fault detection algorithm for class-imbalanced process data. *IEEE Trans. Semicond. Manuf.* **2015**, *28*, 318–328. [[CrossRef](#)]
4. Li, D.; Zhao, Y.; Zhao, Y. A dynamic-model-based fault diagnosis method for a wind turbine planetary gearbox using a deep learning network. *Prot. Control Mod. Power Syst.* **2022**, *7*, 22. [[CrossRef](#)]
5. Dong, Y.; Li, Y.; Zheng, H.; Wang, R.; Xu, M. A new dynamic model and transfer learning based intelligent fault diagnosis framework for rolling element bearings race faults: Solving the small sample problem. *ISA Trans.* **2022**, *121*, 327–348. [[CrossRef](#)] [[PubMed](#)]
6. Li, W.; Gu, S.; Zhang, X.; Chen, T. Transfer learning for process fault diagnosis: Knowledge transfer from simulation to physical processes. *Comput. Chem. Eng.* **2020**, *139*, 106904. [[CrossRef](#)]
7. Zhu, P.; Dong, S.; Pan, X.; Hu, X.; Zhu, S. A simulation-data-driven subdomain adaptation adversarial transfer learning network for rolling element bearing fault diagnosis. *Meas. Sci. Technol.* **2022**, *33*, 075101. [[CrossRef](#)]
8. Liu, C.; Gryllias, K. Simulation-driven domain adaptation for rolling element bearing fault diagnosis. *IEEE Trans. Ind. Inform.* **2021**, *18*, 5760–5770. [[CrossRef](#)]
9. Wang, Y.; Liu, Y.; Chow, T.W.; Gu, J.; Zhang, M. A Balanced Adversarial Domain Adaptation Method for Partial Transfer Intelligent Fault Diagnosis. *IEEE Trans. Instrum. Meas.* **2022**, *71*, 3526711. [[CrossRef](#)]
10. Li, X.; Zhang, W.; Ma, H.; Luo, Z.; Li, X. Partial transfer learning in machinery cross-domain fault diagnostics using class-weighted adversarial networks. *Neural Netw.* **2020**, *129*, 313–322. [[CrossRef](#)] [[PubMed](#)]
11. Sun, R.; Liu, X.; Liu, S.; Xiang, J. A game theory enhanced domain adaptation network for mechanical fault diagnosis. *Meas. Sci. Technol.* **2022**, *33*, 115501. [[CrossRef](#)]
12. Li, W.; Chen, Z.; He, G. A novel weighted adversarial transfer network for partial domain fault diagnosis of machinery. *IEEE Trans. Ind. Inform.* **2020**, *17*, 1753–1762. [[CrossRef](#)]
13. Jiao, J.; Zhao, M.; Lin, J.; Ding, C. Classifier inconsistency-based domain adaptation network for partial transfer intelligent diagnosis. *IEEE Trans. Ind. Inform.* **2019**, *16*, 5965–5974. [[CrossRef](#)]
14. Kuang, J.; Xu, G.; Tao, T.; Wu, Q.; Han, C.; Wei, F. Dual-weight Consistency-induced Partial Domain Adaptation Network for Intelligent Fault Diagnosis of Machinery. *IEEE Trans. Instrum. Meas.* **2022**, *71*, 3519612. [[CrossRef](#)]
15. Cao, Z.; Long, M.; Wang, J.; Jordan, M.I. Partial transfer learning with selective adversarial networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 2724–2732.
16. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
17. Ganin, Y.; Ustinova, E.; Ajakan, H.; Germain, P.; Larochelle, H.; Laviolette, F.; Marchand, M.; Lempitsky, V. Domain-adversarial training of neural networks. *J. Mach. Learn. Res.* **2016**, *17*, 2096–2130.
18. Woo, S.; Park, J.; Lee, J.-Y.; Kweon, I.S. Cbam: Convolutional block attention module. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 3–19.
19. Long, M.; Wang, J.; Ding, G.; Sun, J.; Yu, P.S. Transfer feature learning with joint distribution adaptation. In Proceedings of the IEEE International Conference on Computer Vision, Sydney, Australia, 1–8 December 2013; pp. 2200–2207.
20. Song, M.-M.; Xiong, Z.-C.; Zhong, J.-H.; Xiao, S.-G.; Tang, Y.-H. Research on fault diagnosis method of planetary gearbox based on dynamic simulation and deep transfer learning. *Sci. Rep.* **2022**, *12*, 17023. [[CrossRef](#)] [[PubMed](#)]
21. Sun, B.; Saenko, K. Deep coral: Correlation alignment for deep domain adaptation. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; pp. 443–450.
22. Tzeng, E.; Hoffman, J.; Zhang, N.; Saenko, K.; Darrell, T. Deep domain confusion: Maximizing for domain invariance. *arXiv* **2014**, arXiv:1412.3474.
23. Ganin, Y.; Lempitsky, V. Unsupervised domain adaptation by backpropagation. In Proceedings of the International conference on machine learning, Lille, France, 6–11 July 2015; pp. 1180–1189.

**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.