Abstract: Domain adaptation techniques are crucial for addressing the discrepancies between training and testing data distributions caused by varying operational conditions in practical bearing fault diagnosis. However, transfer fault diagnosis faces significant challenges under complex conditions with dispersed data and distinct distribution differences. Hence, this paper proposes CWT-SimAM-DAMS, a domain adaptation method for bearing fault diagnosis based on SimAM and an adaptive weighting strategy. The proposed scheme first uses Continuous Wavelet Transform (CWT) and Unsharp Masking (USM) for data preprocessing, and then feature extraction is performed using the Residual Network (ResNet) integrated with the SimAM module. This is combined with the proposed adaptive weighting strategy based on Joint Maximum Mean Discrepancy (JMMD) and Conditional Adversarial Domain Adaption Network (CDAN) domain adaptation algorithms, which minimizes the distribution differences between the source and target domains more effectively, thus enhancing domain adaptability. The proposed method is validated on two datasets, and experimental results show that it improves the accuracy of bearing fault diagnosis.

Keywords: bearing fault diagnosis; domain adaptation; adaptive weighting strategy; CWT; SimAM

1. Introduction

With the advent of Industry 4.0, modern information technology has become deeply integrated with manufacturing, leading to significant advancements in machine manufacturing and industrial production. Rotating machinery equipment is extensively used in these fields, and bearings, as key mechanical components of such machinery, affect the safe operation of the equipment in its entirety [1–3]. Statistics show that about 40% of failures in rotating equipment are caused by bearing faults [4]. Thus, accurate and real-time detection of bearing faults is essential for the smooth progress of mechanical manufacturing and industrial production.

With the boom in big data and artificial intelligence technology, data-driven intelligent fault diagnosis methods have become a key research focus in recent years [5–7]. Data processing plays a key role in the effectiveness of fault diagnosis. Raw bearing fault signals typically reflect time domain information, and, after serial processing, they can reveal frequency domain information. However, considering only time or frequency domain information, the model’s fault diagnosis performance is often suboptimal when dealing with nonlinear bearing fault diagnosis signals. Therefore, attention has been given to the Continuous Wavelet Transform (CWT), which can simultaneously reflect both time and frequency domain information. Gu et al. [8] proposed a hybrid deep learning model for fault diagnosis that effectively extracts fault features from bearings and handles small sample datasets. This model uses variational modal decomposition (VMD) [9] and CWT algorithms for data processing and employs a convolutional neural network (CNN) [10] for
model training. Cheng et al. [11] introduced a rotational machinery diagnosis method based on the CWT and Local Binary Convolutional Neural Networks. Stable and accurate fault diagnosis technology can reliably detect the types of faults in motors, providing reliable support for the operational monitoring and maintenance of rotating machinery [5–7]. As an intelligent algorithm, the deep learning model can extract fault features from bearing data for end-to-end fault diagnosis. Wang et al. [12] proposed a method combining an improved residual network and wavelet transform for intelligent gearboxes. This approach effectively extracts features and diagnoses single faults, compound faults, and unbalance faults. Jiang et al. [13] introduced a multi-scale convolutional neural network featuring channel attention utilizing both max pooling and average pooling layers to identify bearing fault characteristics at different scales. Regarding nonlinear feature extraction methods, Zhang et al. [14] developed an adaptive activation function with a $\tanh$ function and slope thresholding. These were incorporated into the Residual Network (ResNet), allowing the network to extract features that are significantly different between faults types. However, deep learning requires a large number of data for training, and deep learning algorithms need training and testing data to have the same operational conditions, meaning they must share the same distribution. Real-time changes in operating conditions such as humidity, voltage, speed, current fluctuations, and load can cause data distribution variations during the normal operation of actual rotating equipment. These changes decrease the accuracy of deep learning algorithms when processing test dataset data [15].

Recently, bearing fault diagnosis methods based on domain adaptation transfer learning have addressed several challenges. Specifically, they have resolved the issues of low generalizability and low robustness owing to limited data in deep learning. They have tackled the problems associated with the source and target data being in different feature spaces or distributions. Schwendemann et al. [16] proposed the Layered Maximum Mean Discrepancy (LMMD) method, an extension of the Maximum Mean Discrepancy (MMD) that incorporates the unique characteristics of the proposed intermediary domain. Lu et al. [17] developed an architecture in which the conditional and marginal distributions are adapted across multiple neural network layers. This method uses the MMD to measure the distribution discrepancies and introduces an adaptive weighting strategy to ascertain the importance of different distributions. Mao et al. [18] combined the adaptability of Domain Adversarial Neural Networks (DANNs) with structured relational information across various failure models to enhance transfer learning effectiveness. Chen et al. [19] proposed the Multi-Gradient Hierarchical Domain Adaptation Network, which concurrently acquires transferable domain invariance and class-discriminative insights, improving the diagnostic transferability of bearing faults. All of these methods have achieved satisfactory results in some respects. However, traditional bearing fault diagnosis methods based on transfer learning and CWT time–frequency images still face the following major challenges in feature capture and domain adaptation:

1. When the fault signal is weak, the data are smooth, or feature contrast is not apparent. CWT alone may not clearly display bearing fault characteristics. Therefore, enhancing data contrast through sharpening methods to improve the discriminative power of data features is particularly important.

2. In the process of feature extraction for fault diagnosis, models need strong feature capture capabilities. Traditional residual networks often struggle to adequately focus on important features when capturing complex fault patterns, leading to suboptimal feature extraction.

3. Domain adaptation algorithms based on kernel methods, such as the MMD, LMMD, and Joint Maximum Mean Discrepancy (JMMD), rely heavily on the selection and tuning of the kernel function to achieve feature alignment. When dealing with data exhibiting complex nonlinear distributions, the choice of kernel function greatly influences the algorithm’s ability to capture feature differences and interactions within the data. Domain adaptation algorithms based on adversarial learning, such as DANNs and Conditional Adversarial Domain Adaption Networks (CDANs), align features between the source and target domains through adversarial training. Although adversarial learning excels at capturing complex nonlinear distribution
differences, the training process is prone to gradient instability, mode collapse, and vanishing gradients, making it difficult for the model to converge. Additionally, a DANN primarily focuses on aligning feature distributions and lacks explicit alignment of class conditions, which can adversely affect classification performance.

In order to solve the problems mentioned above, this paper proposes the CWT-SimAM-DAMS model. The specific innovations and contributions are as follows:

1. The one-dimensional bearing fault signal is integrated using a sliding window, the segmented data are processed with the CWT algorithm, and, finally, the resulting CWT time–frequency images are enhanced by overlaying high-frequency features using the Unsharp Masking (USM) algorithm. This method is named CWT-USM.

2. The SimAM attention mechanism is integrated into the Residual Network to enhance the model’s feature extraction capability for input images and provide a robust feature extraction foundation for JMMD and CDAN domain adaptation algorithms. This model is named SimAM-ResNet.

3. The model’s generalization ability is enhanced utilizing the JMMD and CDAN domain adaptation algorithms and designing an adaptive weighting strategy. The JMMD domain adaptive algorithm provides stable distribution alignment to make adversarial training more stable, and the CDAN domain adaptive algorithm mitigates the JMMD domain adaptive algorithm’s dependence on the kernel method by capturing complex nonlinear distribution differences through adversarial learning. CDAN and JMMD domain adaptive algorithms focus on both the joint distribution of labels and features. The adaptive weighting strategy considers the classification, JMMD, and CDAN loss, effectively reducing the discrepancy in joint distributions and achieving global domain alignment. Additionally, parameters are adaptively adjusted at various stages of model training to ensure the model’s optimal performance.

The rest of the paper is organized as follows: Section 2 describes the theoretical concepts of transfer learning, CWT, USM, SimAM, and ResNet. Section 3 presents a new domain adaptive method for diagnosing bearing faults, including the SimAM attention mechanism, the JMMD and CDAN domain adaptation algorithms, and the weight adaptive strategy. Section 4 describes the specifics of the dataset and the parameter settings used in this study. Section 5 provides experimental results and conducts an analysis. Section 6 summarizes the paper.

2. Theoretical Background
2.1. Description of Transfer Learning Problems

In domain adaptation [5], the source domain is defined as \( D_s = \{ \chi^s, P(x_s) \} \) and the target domain as \( D_t = \{ \chi^t, P(x_t) \} \). The dataset for the source domain is \( X_s = \{ (x^s_i) \}_{i=1}^{N_s} \), with labels \( Y_s = \{ (y^s_i) \}_{i=1}^{N_s} \), where \( y_i \in 1, 2, \ldots, K \), and \( K \) denotes the total number of categories. The dataset for the target domain is \( X_t = \{ (x^t_i) \}_{i=1}^{N_t} \). The main problem addressed in this paper is that the feature space of the source and target domains are the same, i.e., \( \chi^s = \chi^t \), but their marginal distributions differ, i.e., \( P(x_s) \neq P(x_t) \).

2.2. Continuous Wavelet Transform

When dealing with the continuous one-dimensional vibration signals of motor faults, an effective feature extraction strategy is to convert the signals into two-dimensional time–frequency images. This method not only enriches the representation of frequency domain information, but also makes it more suitable for the learning process of neural networks due to its two-dimensional structure. In the time–frequency plots, one can directly observe the changes in signal frequency components over time. The CWT is particularly adept due to its window scaling, which overcomes the limitations of the Short-Time Fourier Transform (STFT) [20,21], where window sizes do not vary with frequency or time, making
it more suitable for handling transient signals like those in motor faults. The CWT is mathematically formulated as follows [22]:

\[
W_x(a, b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{+\infty} x(t) \psi^* \left( \frac{t - b}{a} \right) dt
\]

(1)

where \(W_x(a, b)\) represents the wavelet coefficients, \(a\) is the scaling parameter, and \(b\) is the translation parameter. The choice of the mother wavelet is crucial in wavelet transforms as it determines the accuracy and efficiency of the transform. Common mother wavelets include the Daubechies Wavelet, Reverse Biorthogonal Wavelet, Bior Wavelet, and Morlet Wavelet. This paper selects the cmor wavelet as the mother wavelet for the CWT because the cmor wavelet, a complexified version of the Morlet Wavelet, possesses excellent time–frequency localization properties and effective filtering and signal reconstruction capabilities.

2.3. Unsharp Masking

Unsharp Masking is a widely used technique for sharpening enhancement. The USM algorithm acquires high-frequency components by subtracting the low-pass filtered blurred image from the original image. These high-frequency parts are then multiplied by a gain coefficient and added back to the original image, enhancing the contrast of these high-frequency components and thereby improving the visual clarity of the image details and edges in the image. The processing steps of Unsharp Masking are as [23] follows:

**Step 1:** Use a Gaussian filter to create a blurred version of the original image and reduce its high-frequency content.

\[
G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2 + y^2}{2\sigma^2}}
\]

(2)

where \(x\) and \(y\) denote the positions relative to the center pixel, and \(\sigma\) is the standard deviation of the Gaussian distribution, which controls the extent of blurring.

**Step 2:** Use a high-pass filter to extract the edges and texture information from the image, i.e., the high-frequency components.

\[
H = I - (G * I)
\]

(3)

where \(I\) represents the original image, \(G * I\) represents the image after applying Gaussian filtering, and \(H\) represents the image containing the high-frequency components.

**Step 3:** Add the high-frequency image to the original image according to a coefficient, adjust the sharpening intensity, and merge them.

\[
B = I + \alpha H
\]

(4)

where \(\alpha\) represents the sharpening intensity and \(B\) is the final image after sharpening.

2.4. Residual Network

For richer image features, a common method is to increase the network. However, as the network depth increases, the model may encounter vanishing or exploding gradient problems, which can decrease its accuracy. He et al. [24] proposed the Residual Network to simplify the training of deep networks. ResNet improves the traditional CNN and effectively addresses this issue. The structure of a residual module is illustrated in Figure 1. The output \(G(X)\) of the residual network is composed of a combination of input \(x\) and mapping function \(F(x)\).
Sensors 2024, 24, 4251

Figure 1. The basic architecture of ResNet.

2.5. SimAM

SimAM is an attention mechanism distinct from traditional channel attention mechanisms or spatial attention mechanisms [25]. SimAM identifies neurons with higher spatial suppression effects by defining an energy function and assigning them higher weights. Its specific framework is illustrated in Figure 2, and the energy function is expressed as follows:

$$\varepsilon_t(w_t, b_t, y, x_i) = (y_t - \tilde{y}_t)^2 + \frac{1}{M-1} \sum_{i=1}^{M-1} (y_0 - \tilde{x}_i)^2$$

where $t$ and $x_i$ represent the target neuron and the estimated values of other neurons on a single channel of input $X \in \mathbb{R}^{C \times H \times W}$ (C, H, and W denote the number of channels, height, and width, respectively, and $\mathbb{R}$ is the set of real numbers). $w_t$ and $b_t$ denote the weight and bias, $M = H \times W$ represents the number of neurons in that channel, and $\tilde{t}_i = w_t t + b_t$ and $\tilde{x}_i = w_t x_i + b_t$ are the linear transformations of $t$ and $x_i$.

Figure 2. The architecture of the SimAM attention mechanism.

Introducing the regularization coefficient $\lambda$ into the weights, the energy formula is as follows:

$$\varepsilon_t(w_t, b_t, y, x_i) = \frac{1}{M-1} \sum_{i=1}^{M-1} (-1 - (w_t x_i + b_t))^2 + (1 - (w_t t + b_t))^2 + \lambda w_t^2$$

The solutions for $w_t$ and $b_t$ are obtained as follows:

$$\begin{cases} w_t &= -\frac{2(t - \mu_t)}{(t - \mu_t)^2 + 2\sigma_t^2 + 2\lambda} \\ b_t &= -\frac{1}{2} (t + \mu_t) w_t \end{cases}$$

where $\mu_t = \frac{1}{M-1} \sum_{i=1}^{M-1} x_i$ and $\sigma_t^2 = \frac{1}{M-1} \sum_{i=1}^{M-1} (x_i - \mu_t)^2$ represent the mean and variance of the channel excluding the target neuron.

The final simplified minimum energy is as follows:

$$\varepsilon_t^* = \frac{4(\delta^2 + \lambda)}{(t - \mu_t)^2 + 2\sigma^2 + 2\lambda}$$
where \( \sigma \) represents the covariance value. Equation (8) reveals that the smaller the energy value, the greater the separability between the target neuron and the rest of the neurons, indicating an inverse relationship between the energy value and the separability of the target neuron from the rest. Therefore, the attention parameter is denoted by \( \frac{1}{E} \).

Finally, the enhanced input with attention is obtained as follows:

\[
\tilde{X} = \text{sigmoid}(\frac{1}{E}) \odot X
\] (9)

3. The Proposed Method

The proposed CWT-SimAM-DAMS method is illustrated in Figure 3. The process begins with converting vibration signals into time–frequency images using the CWT algorithm. These images are then enhanced with the USM algorithm, and the enhanced data serve as input for the model. In the source domain, the feature extraction model first extracts the bearing fault features. These features are then subjected to dimensionality reduction and nonlinear transformation through a bottleneck layer, which includes a Dropout layer \((p = 0.5)\), a fully connected layer, and a ReLU activation function. The transformed features are passed through a linear classifier to compute the classification loss. Simultaneously, the features from the bottleneck layer are used with the JMMD and CDAN domain adaptation algorithms to align the joint distribution between the source and target domains and calculate the domain adaptation loss for these two algorithms. The JMMD provides a smooth and continuous alignment target, making adversarial training more stable, while the CDAN captures complex nonlinear distribution differences through adversarial learning. Additionally, the proposed weight adaptive algorithm can adjust the weights of each part in real time based on the losses from the classification, JMMD, and CDAN during model training, achieving the optimal fault monitoring state.

3.1. Data Processing Based on Unsharp Masking and Continuous Wavelet Transform

The core principle in information theory is that information inevitably suffers loss or degradation during transmission. Therefore, when converting one-dimensional raw data into two-dimensional images, some loss of data information is unavoidable. This paper adopts a 50% data overlap strategy [26] when generating CWT images effectively. The specific procedure is as follows. First, calculate the number of samples per cycle based on the following equation:

\[
N_{\text{min}} = \frac{f_Z r}{60} \quad (10)
\]

where \( f_Z \) represents the sampling frequency of the vibration signal [27], and \( r \) is the rotation speed of the bearing. Therefore, the minimum number of samples for the bearing is calculated as \( N_{\text{min}} \). However, to maintain the completeness of the sampling data, we consider \( N \geq 1.5 N_{\text{min}} \). Then, the data are segmented through a sliding window, where the moving step size of the data segmentation window is half the number of samples per cycle. Continue this process until the end of the data is reached. This procedure is illustrated in Figure 3. After the data are segmented using a sliding window, the steps for image processing are as follows:

**Step 1:** Normalize the segmented data according to

\[
f(x) = \frac{(x - \min)}{\max - \min} \quad (11)
\]

**Step 2:** Apply the CWT algorithm to the data to transform them into a two-dimensional time–frequency image.

**Step 3:** Set different values for the key parameters and the USM algorithm and enhance each image generated by the CWT algorithm using the USM algorithm.

**Step 4:** Conduct comparative experiments on the images generated by the USM algorithm under different parameters, and select the images processed with the parameters yielding the highest fault diagnosis accuracy as the experimental input data.
3.2. Domain Adaptation Model Based on SimAM and ResNet

3.2.1. Residual Network Integrated with SimAM

In fault diagnosis, neural networks play a crucial role in feature extraction. However, traditional residual models often struggle to effectively identify fault features when dealing with complex time–frequency images, mainly due to their limited feature extraction capabilities. Hence, to address this issue, we propose the SimAM-ResNet model, which relies on ResNet as the backbone. ResNet addresses the vanishing gradient problem in deep networks by introducing residual connections, enabling easier training and optimization. Specifically, ResNet adds residual connections across the layers by directly adding the input signal to the output signal during forward propagation, thus implementing “skip connections”. This connection method enables the network to learn more accurate feature representations and significantly reduces training errors. Additionally, ResNet employs batch normalization techniques and pre-activation structures to enhance network performance and stability further. Moreover, the SimAM attention mechanism is introduced on this basis. SimAM determines attention weights by computing similarity scores between elements in the input sequence. Specifically, it calculates the similarity of each element in the input sequence to other elements, typically using the similarity or dot product op-
erations. Then, for each element, it weights and sums the other elements based on their similarity scores to obtain the attention representation of that element. SimAM is unique because it introduces a similarity threshold which automatically filters out low-quality elements, reducing the impact of noise and redundant information. Thus, SimAM improves the robustness and generalization ability of the fault diagnosis model, which is used for refining feature mapping. The specific network structure of the fault classification module is presented in Figure 4 and Table 1.

Table 1. Parameters of the residual network structure integrated with the SimAM attention mechanism.

<table>
<thead>
<tr>
<th>Layer</th>
<th>Parameter</th>
<th>Output Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>SimAM_module</td>
<td>Conv2d</td>
<td>32 x 32 x 3</td>
</tr>
<tr>
<td></td>
<td>Conv2d kernel_size = 7, stride = 2, pad = 3</td>
<td>16 x 16 x 64</td>
</tr>
<tr>
<td>Layer1</td>
<td>Conv2d kernel_size = 3, stride = 1, pad = 1</td>
<td>16 x 16 x 64</td>
</tr>
<tr>
<td></td>
<td>Conv2d kernel_size = 3, stride = 1, pad = 1</td>
<td>16 x 16 x 64</td>
</tr>
<tr>
<td></td>
<td>Conv2d kernel_size = 3, stride = 1, pad = 1</td>
<td>16 x 16 x 64</td>
</tr>
<tr>
<td></td>
<td>Conv2d kernel_size = 3, stride = 1, pad = 1</td>
<td>16 x 16 x 64</td>
</tr>
<tr>
<td>Layer2</td>
<td>Conv2d kernel_size = 3, stride = 2, pad = 1</td>
<td>8 x 8 x 128</td>
</tr>
<tr>
<td></td>
<td>Conv2d kernel_size = 3, stride = 1, pad = 1</td>
<td>8 x 8 x 128</td>
</tr>
<tr>
<td></td>
<td>Conv2d kernel_size = 3, stride = 1, pad = 1</td>
<td>8 x 8 x 128</td>
</tr>
<tr>
<td></td>
<td>Conv2d kernel_size = 3, stride = 1, pad = 1</td>
<td>8 x 8 x 128</td>
</tr>
<tr>
<td>Layer3</td>
<td>Conv2d kernel_size = 3, stride = 2, pad = 1</td>
<td>4 x 4 x 256</td>
</tr>
<tr>
<td></td>
<td>Conv2d kernel_size = 3, stride = 1, pad = 1</td>
<td>4 x 4 x 256</td>
</tr>
<tr>
<td></td>
<td>Conv2d kernel_size = 3, stride = 1, pad = 1</td>
<td>4 x 4 x 256</td>
</tr>
<tr>
<td></td>
<td>Conv2d kernel_size = 3, stride = 1, pad = 1</td>
<td>4 x 4 x 256</td>
</tr>
<tr>
<td>Layer4</td>
<td>Conv2d kernel_size = 3, stride = 2, pad = 1</td>
<td>2 x 2 x 512</td>
</tr>
<tr>
<td></td>
<td>Conv2d kernel_size = 3, stride = 1, pad = 1</td>
<td>2 x 2 x 512</td>
</tr>
<tr>
<td></td>
<td>Conv2d kernel_size = 3, stride = 1, pad = 1</td>
<td>2 x 2 x 512</td>
</tr>
<tr>
<td></td>
<td>Conv2d kernel_size = 3, stride = 1, pad = 1</td>
<td>2 x 2 x 512</td>
</tr>
</tbody>
</table>

Figure 4. Residual network structure integrated with the SimAM attention mechanism.

3.2.2. Joint Maximum Mean Discrepancy

The Maximum Mean Discrepancy [28] is a non-parametric metric for evaluating the difference in distributions of different datasets. It operates by mapping the feature representations of the source and target domains into the Regenerative Kernel Hilbert Space.
(RKHS), where the distribution discrepancy is determined by calculating the marginal distributions \(P(X_s)\) and \(Q(X_t)\) between the two domains. The MMD is defined as follows:

\[
\text{MMD}^2(P, Q) = \sup_{||\phi||_{\mathcal{H}} \leq 1} ||E[\phi(x)] - E[\phi(y)]||_{\mathcal{H}}^2
\]

where \(\sup\) denotes the supremum, \(\phi\) represents the mapping function, which maps the original dataset into the reproducing kernel Hilbert space, \(\mathcal{H}\) denotes the reproducing kernel Hilbert space, and the subscript \(||\phi||_{\mathcal{H}} \leq 1\) indicates that the norm of the function in the Hilbert space is less than or equal to 1. The empirical estimate of the MMD is given by

\[
\text{MMD}^2(P, Q) = ||\frac{1}{n_s} \sum_{i=1}^{n_s} \phi(x_i) - \frac{1}{n_t} \sum_{j=1}^{n_t} \phi(x'_j)||_{\mathcal{H}}^2
\]

\[
= ||\frac{1}{n_s} \sum_{i=1}^{n_s} \sum_{j=1}^{n_t} k(x_i, x'_j) - \frac{2}{n_sn_t} \sum_{i=1}^{n_s} \sum_{j=1}^{n_t} k(x_i, x'_j) + \frac{1}{n_t} \sum_{j=1}^{n_t} \sum_{l=1}^{n_s} k(x'_j, x'_l)||_{\mathcal{H}}
\]

where \(k(\cdot, \cdot)\) is the kernel function, and \(k(x_i, y_j) = \exp(-||x_i - y_j||^2/(2\sigma^2))\).

The MMD, serving as a kernel-based two-sample test statistic, is extensively utilized to assess the distinction between marginal distributions but has not been employed to gauge the difference between joint distributions. Moreover, the MMD exhibits limited domain adaptation capability under complex multimodal conditions, and optimizing kernel parameters poses challenges. Therefore, the JMMD [29] is proposed by considering the empirical joint distributions \(P(X_s, Y_s)\) and \(Q(X_t, Y_t)\) between the two domains. The JMMD is defined as follows:

\[
L_{\text{JMMD}}(P, Q) = ||\mathbb{E}_P(\otimes_{l=1}^{L} \phi^l(z_l)) - \mathbb{E}_Q(\otimes_{l=1}^{L} \phi^l(z_l))||_{\mathcal{H}}^2
\]

3.2.3. Conditional Adversarial Domain Adaption

A DANN is a domain adaptive network model based on adversarial concepts. DANNs optimize learning through adversarial training between a feature extractor and a domain classifier. During the training process, domain adaptation is embedded into the model’s learning, enabling the model to extract and recognize domain-invariant features. The Category classifier trained based on adversarial concepts demonstrates good generalization in the target domain. However, DANNs do not consider the joint distribution of features and labels, which can lead to the neglect of class-specific features during training. Additionally, when the data distribution exhibits a multimodal structure, focusing solely on feature distribution makes it challenging for DANNs to accurately align the source and target domains. Long et al. [30] proposed a Conditional Domain Adversarial Network. The CDAN divides the entire network structure into three modules: a feature extractor, a Category classifier, and a domain discriminator. The CDAN addresses the problem of the DANN neglecting the joint distribution of features and labels by introducing a multilinear conditioning mechanism. Specifically, it optimizes the joint distribution of features \(f\) and labels \(g\) through multilinear mapping, thereby considering the joint distribution of features and labels. \(T_f(f, g)\) and \(T_g(f, g)\) are the multilinear mapping methods proposed by the CDAN. When \(d_f \times d_g \leq 4096\), the CDAN takes \(T_f(f, g)\) as input for the domain discriminator. When \(d_f \times d_g > 4096\), to avoid the dimensionality explosion, the CDAN randomly selects certain dimensions of the features and labels for multilinear mapping. In this case, the CDAN takes \(T_f(f, g)\) as the input for the domain discriminator. The multilinear mapping method can capture the distribution characteristics of multimodal complex data. The loss function of the CDAN can be expressed as

\[
L_{\text{CDAN}}(\theta_f, \theta_g) = -\mathbb{E}_{x_s' \in D_s} \log[D(f_s', g_s')] - \mathbb{E}_{x_t' \in D_t} \log[1 - D(f_t', g_t')]
\]
where \( f_t = G_f(x_t^i, \theta_f) \), \( g_s^i = G_c(G_f(x_t^i, \theta_f)) \), \( D(f, g) = G_d(f \otimes g, \theta_d) \).

Substituting \( f_t \), \( g_s^i \), and \( D(f, g) \) into Equation (15), the loss function of the CDAN can be expressed as

\[
L_{CDAN}(\theta_f, \theta_d) = -\mathbb{E}_{x_t^i \in D_i} \log [G_d(G_f(x_t^i) \otimes G_c(G_f(x_t^i)))] \\
-\mathbb{E}_{x_t^j \in D_t} \log [1 - G_d(G_f(x_t^j) \otimes G_c(G_f(x_t^j)))]
\]  

(16)

3.2.4. Domain Adaptation Based on Adaptive Weighting Strategy

This paper proposes a domain adaptation method that improves the accuracy of cross-domain fault diagnosis by enabling the model to reduce marginal distribution discrepancies like the JMMD and achieve global domain alignment like the CDAN. Additionally, this paper designs an adaptive weighting strategy based on the principle that the parts of the loss function with larger values should receive more attention during the training process. The objective of this strategy is to allocate higher weights to objectives that are difficult to achieve in the current stage of the model, thereby prioritizing these parts during training. To ensure that the model learns effective source domain features in the early stages of training and improves domain adaptation ability in the later stages during target optimization, the final loss function \( \mathcal{L}_{JMMMD} \) is obtained by multiplying the JMMD loss function by a parameter \( \lambda_{JMMMD} \). The CDAN requires minimizing the label classification loss and maximizing the domain classification loss during the optimization process. To eliminate the simultaneous maximization and minimization optimization problem, a Gradient Reversal Layer (GRL) is introduced between the feature extractor and the domain discriminator. Specifically, during forward propagation, the GRL does not perform any operation and passes the features normally through the network. During backward propagation, the GRL takes the gradient from the subsequent network, multiplies it by the parameter \(-\lambda_{CDAN}\), and passes it to the previous layer. Through the above operations, the final loss function of the CDAN is obtained as \( \mathcal{L}_{CDAN} \). To effectively integrate the JMMD and CDAN, we introduce three key weights: the classifier weight \( W_c \), distance weight \( W_{JMMMD} \), and adversarial weight \( W_{CDAN} \). The adaptive weighting strategy dynamically adjusts these weights in real time based on the model’s performance during training and optimization objectives.

Algorithm 1 describes the process of the CWT-SimAM-DAMS model training. The overall loss function of the CWT-SimAM-DAMS model can be expressed as

\[
\mathcal{L} = W_c \mathcal{L}_c + W_{JMMMD} \mathcal{L}_{JMMMD} + W_{CDAN} \mathcal{L}_{CDAN}
\]  

(17)

\[
W_{\text{classifier}}^k = \frac{3 \times \mathcal{L}^k_c}{\mathcal{L}^k_c + \mathcal{L}^k_{JMMMD} + \mathcal{L}^k_{CDAN}}
\]  

(18)

\[
W_{\text{JMMMD}}^k = \frac{3 \times \mathcal{L}^k_{JMMMD}}{\mathcal{L}^k_c + \mathcal{L}^k_{JMMMD} + \mathcal{L}^k_{CDAN}}
\]  

(19)

\[
W_{\text{CDAN}}^k = \frac{3 \times \mathcal{L}^k_{CDAN}}{\mathcal{L}^k_c + \mathcal{L}^k_{JMMMD} + \mathcal{L}^k_{CDAN}}
\]  

(20)

where \( \mathcal{L} \) represents the overall loss function of the SimAM-DAMS model, and \( \mathcal{L}_c \) is the classification loss of the model. \( W_{\text{classifier}}^k \) represents the weight corresponding to \( \mathcal{L}_c \) at the \( K \)-th epoch. \( \mathcal{L}_{JMMMD} \) is the loss function of the JMMD algorithm, \( W_{\text{JMMMD}}^k \) represents the weight corresponding to \( \mathcal{L}_{JMMMD} \) at the \( K \)-th epoch. \( \mathcal{L}_{CDAN} \) is the loss function of the CDAN algorithm. \( W_{\text{CDAN}}^k \) represents the weight corresponding to \( \mathcal{L}_{CDAN} \) at the \( K \)-th epoch. In this paper, there are three loss functions. To ensure that the sum of the weights is a fixed parameter of 3, each weight is multiplied by 3, which has no special significance.
Algorithm 1: CWT-SimAM-DAMS algorithm.

**Input:** epoch: max iteration; 
\( D \): CWT-USM images; 
Randomly initialized network parameter parameterized by \( \theta \);  
**Training:**  
set \( n = 0 \),  
**While** \( n < \text{epoch} \) **do**  
for each batch in \( D \) **do**  
obtain SimAM-RseNet results: \( f = F(X) \)  
obtain classification loss  
obtain JMM loss by Equation (14) and CDAN loss by Equation (16)  
calculate CWT-SimAM-DAMS loss by Equation (17)  
obtain Predicted results: \( G(z) \)  
update \( \theta \) by using the method of gradient descent;  
set \( n++ \);  
**end**  
**Output:** Predicted label \( y = G(z) \)

4. Data Description

The experimental platform involves a Windows 11 64-bit operating system using a 13th Gen Intel(R) Core(TM) i9-13900HX at 2.20 GHz and an NVIDIA GeForce RTX 4060 laptop GPU. The program runs in the PyCharm 2023.3.4 × 64 environment.

4.1. Dataset Introduction

This paper primarily utilizes two publicly available bearing fault datasets: the Case Western Reserve University bearing dataset and the dataset from the laboratory of the University of Padova. The number of epochs for the CWRU dataset is 80, and, for the PU dataset, it is 800. According to Equation (10), the signal period sampling points are 800 for the CWRU dataset and 3840 for the PU dataset. The data are split into training and testing data with a ratio of 75:25. Below is a detailed introduction to the datasets.

4.1.1. Case Western Reserve University Dataset

The CWRU collected vibration acceleration data [31] from the motor drive and fan end (Figure 5). The dataset includes normal bearing and faulty bearing operation data. This paper utilizes a sample frequency of 12 kHz for the faulty samples from the drive end.

![Figure 5. CWRU testing platform.](image)

The bearing speed is categorized into four speeds, labeled as “0, 1, 2, 3”, with different loads under each speed. The data are divided into four operating conditions, as reported in Table 2. The CWRU dataset comprises 10 bearing health conditions, including one
normal and three types of faults. “IF” represents an inner ring fault, “BF” represents a ball fault, “OF” stands for outer ring fault, and “NA” represents normal bearings, as presented in Table 3. The transfer task 0-1 represents the migration from source domain operating condition 0 to target domain operating condition 1.

Table 2. CWRU dataset operating conditions and data splitting.

<table>
<thead>
<tr>
<th>Task</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Speed (rpm)</td>
<td>1797</td>
<td>1772</td>
<td>1750</td>
<td>1730</td>
</tr>
<tr>
<td>Load (HP)</td>
<td>0HP</td>
<td>1HP</td>
<td>2HP</td>
<td>3HP</td>
</tr>
<tr>
<td>Dataset</td>
<td>train 75</td>
<td>75</td>
<td>75</td>
<td>75</td>
</tr>
<tr>
<td></td>
<td>test 25</td>
<td>25</td>
<td>25</td>
<td>25</td>
</tr>
</tbody>
</table>

Table 3. CWRU dataset fault condition information.

<table>
<thead>
<tr>
<th>Class Label</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fault Location</td>
<td>NA</td>
<td>IF</td>
<td>BF</td>
<td>OF</td>
<td>IF</td>
<td>BF</td>
<td>OF</td>
<td>IF</td>
<td>BF</td>
<td>OF</td>
</tr>
<tr>
<td>Fault Size (mils)</td>
<td>0</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>14</td>
<td>14</td>
<td>14</td>
<td>21</td>
<td>21</td>
<td>21</td>
</tr>
</tbody>
</table>

4.1.2. Paderborn University Dataset

The PU dataset [32] contains two sets of data: an artificial dataset and an actual bearing damage dataset. This paper selects the actual bearing damage data collected from an accelerated life experiment. The experimental setup [33] is illustrated in Figure 6. The electric motor comprises a drive motor, adjusting nut, spring package, and housing. The vibration acceleration signal sampling frequency for the PU dataset is 64 kHz. Based on the changes in load, radia, and speed in the PU dataset, this paper selects three working conditions for the motor, as reported in Table 4. Six transfer learning tasks are constructed accordingly. This paper investigates transfer learning tasks under different operating conditions using data from 13 bearings damaged due to accelerated life experiments. Table 5 provides the classification information.

Figure 6. PU testing platform.

Table 4. PU dataset operating conditions and data splitting.

<table>
<thead>
<tr>
<th>Task</th>
<th>0</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Load (Nm)</td>
<td>0.7</td>
<td>0.1</td>
<td>0.7</td>
</tr>
<tr>
<td>Radia (N)</td>
<td>1000</td>
<td>1000</td>
<td>400</td>
</tr>
<tr>
<td>Speed (rpm)</td>
<td>1500</td>
<td>1500</td>
<td>1500</td>
</tr>
<tr>
<td>Dataset</td>
<td>train 75</td>
<td>75</td>
<td>75</td>
</tr>
<tr>
<td></td>
<td>test 25</td>
<td>25</td>
<td>25</td>
</tr>
</tbody>
</table>
Table 5. PU dataset fault condition information.

<table>
<thead>
<tr>
<th>Class Label</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bearing Code</td>
<td>KA04</td>
<td>KA15</td>
<td>KA16</td>
<td>KA22</td>
<td>KA30</td>
<td>KB23</td>
<td>KB24</td>
<td>KB27</td>
<td>KI14</td>
<td>KI16</td>
<td>KI17</td>
<td>KI18</td>
<td>KI21</td>
</tr>
<tr>
<td>Bearing Element Combination</td>
<td>OR</td>
<td>OR</td>
<td>OR</td>
<td>OR</td>
<td>OR</td>
<td>IR(+OR)</td>
<td>IR(+OR)</td>
<td>IR</td>
<td>IR</td>
<td>IR</td>
<td>IR</td>
<td>IR</td>
<td>IR</td>
</tr>
</tbody>
</table>

S: single damage; M: multiple damage; R: repetitive damage; IR: inner ring; OR: outer ring

4.2. Experimental Parameter Settings

This paper employs the Adam algorithm as the optimizer. The $\lambda_{CDAN}$ and $\lambda_{JMMD}$ settings for this article are as follows:

$$\lambda_{CDAN} = 1, \lambda_{JMMD} = \frac{2}{1 + e^{-10 \times \left(\frac{current\_epoch - middle\_epoch}{max\_epoch - middle\_epoch}\right)} - 1}$$

where $middle\_epoch$ is set to 0. Different datasets have varying maximum numbers of epochs, and the parameter $max\_epoch$ differs accordingly. The maximum number of epochs for the CWRU dataset is 80, and, for the PU dataset, it is 800.

When $current\_epoch \in [0, 40)$, the learning rate is set to $10^{-3}$. When $current\_epoch \in [40, 60)$, the learning rate is set to $10^{-4}$. When $current\_epoch \in (60, max\_epoch)$, the learning rate is set to $10^{-5}$.

5. Experimental Verification

5.1. Experiment on Unsharp Mask Parameter Settings

We selected nine values for the $\sigma$ and $\lambda$ parameters in the Unsharp Masking algorithm based on reference [34], and the experiments were conducted on the CWRU dataset. The ResNet model was utilized, and each experiment was repeated five times. Table 6 reports the corresponding results. This demonstrates the feasibility of the USM algorithm.

By analyzing Table 6, it can be seen that, after applying the USM algorithm, the overall performance of fault diagnosis improved. In transfer task 2-3, the accuracy of all nine parameter configurations selected in this study was higher than the results using the original CWT images. Moreover, different parameter configurations had a certain impact on the final results. Specifically, when the parameters were set to $\sigma = 1.0, \lambda = 1.5$, the algorithm performed best, achieving an average accuracy of 86.59%, which is higher than the average accuracy of 84.89% achieved without using the USM algorithm. $\sigma = 1.0$ and $\lambda = 1.5$ represent a moderately blurred and strongly enhanced image edge and detail processing in sharpening. While reducing minor noise, it also avoids losing too many detailed features. This parameter setting makes the fault features more pronounced without excessively emphasizing noise, effectively balancing the signal-to-noise ratio and showcasing the frequency and time information of CWT images at different scales. Therefore, $\sigma = 1.0, \lambda = 1.5$ were selected as the parameter settings for the Unsharp Masking algorithm.

Table 6. Experimental results of USM algorithm parameter comparison.

<table>
<thead>
<tr>
<th>Parameters of USM</th>
<th>0-0</th>
<th>0-2</th>
<th>0-3</th>
<th>1-0</th>
<th>1-2</th>
<th>1-3</th>
<th>2-0</th>
<th>2-1</th>
<th>2-3</th>
<th>3-0</th>
<th>3-1</th>
<th>3-2</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>Not using USM</td>
<td>79.96</td>
<td>68.88</td>
<td>92.72</td>
<td>94.8</td>
<td>84.84</td>
<td>88.12</td>
<td>90.48</td>
<td>93.7</td>
<td>80.56</td>
<td>79.12</td>
<td>89.12</td>
<td>85.12</td>
<td>84.89</td>
</tr>
<tr>
<td>$\sigma = 0.7, \lambda = 1.0$</td>
<td>69.04</td>
<td>71.76</td>
<td>92.48</td>
<td>95.52</td>
<td>90.24</td>
<td>88.32</td>
<td>89.84</td>
<td>95.68</td>
<td>83.12</td>
<td>78.96</td>
<td>91.6</td>
<td>85.01</td>
<td></td>
</tr>
<tr>
<td>$\sigma = 1.0, \lambda = 0.5$</td>
<td>85.2</td>
<td>72.16</td>
<td>90.92</td>
<td>96.67</td>
<td>90.2</td>
<td>86.17</td>
<td>91.84</td>
<td>97.12</td>
<td>80.48</td>
<td>78</td>
<td>89.28</td>
<td>86.19</td>
<td></td>
</tr>
<tr>
<td>$\sigma = 1.0, \lambda = 0.8$</td>
<td>76.88</td>
<td>71.04</td>
<td>87.44</td>
<td>91.28</td>
<td>86.8</td>
<td>87.6</td>
<td>89.4</td>
<td>96.16</td>
<td>81.04</td>
<td>77.88</td>
<td>91.12</td>
<td>84.97</td>
<td></td>
</tr>
<tr>
<td>$\sigma = 1.0, \lambda = 1.0$</td>
<td>80.96</td>
<td>76.16</td>
<td>70.32</td>
<td>90.6</td>
<td>95.28</td>
<td>88.48</td>
<td>87.84</td>
<td>90.08</td>
<td>97.84</td>
<td>77.76</td>
<td>91.12</td>
<td>88.4</td>
<td>85.24</td>
</tr>
<tr>
<td>$\sigma = 1.0, \lambda = 1.3$</td>
<td>80</td>
<td>75.28</td>
<td>72.16</td>
<td>90.6</td>
<td>95.28</td>
<td>88.48</td>
<td>88.6</td>
<td>92.8</td>
<td>95.64</td>
<td>80.4</td>
<td>80.08</td>
<td>90.48</td>
<td>85.88</td>
</tr>
<tr>
<td>$\sigma = 1.0, \lambda = 1.5$</td>
<td>85.04</td>
<td>82.04</td>
<td>72.32</td>
<td>95.12</td>
<td>95.28</td>
<td>88.27</td>
<td>89.82</td>
<td>95.64</td>
<td>79.84</td>
<td>79.2</td>
<td>89.84</td>
<td>85.88</td>
<td></td>
</tr>
<tr>
<td>$\sigma = 1.3, \lambda = 1.0$</td>
<td>74.56</td>
<td>78.56</td>
<td>73.44</td>
<td>91.28</td>
<td>89.48</td>
<td>83.68</td>
<td>88.92</td>
<td>98.68</td>
<td>78.56</td>
<td>80.16</td>
<td>91.84</td>
<td>85.16</td>
<td></td>
</tr>
<tr>
<td>$\sigma = 1.3, \lambda = 1.5$</td>
<td>79.56</td>
<td>76.16</td>
<td>70.32</td>
<td>91.12</td>
<td>91.574</td>
<td>85.866</td>
<td>89.76</td>
<td>91.04</td>
<td>95.84</td>
<td>79.84</td>
<td>78.64</td>
<td>88.24</td>
<td>84.93</td>
</tr>
<tr>
<td>$\sigma = 1.5, \lambda = 1.0$</td>
<td>78.56</td>
<td>72.32</td>
<td>68.48</td>
<td>91.44</td>
<td>97.76</td>
<td>87.28</td>
<td>91.52</td>
<td>88.88</td>
<td>95.48</td>
<td>80.4</td>
<td>80.08</td>
<td>90.48</td>
<td>85.22</td>
</tr>
</tbody>
</table>
5.2. Comparative Experiment of Image Processing Method

Comparative experiments were conducted on the CWRU dataset to validate the effectiveness of the proposed image extraction method (CWT-USM) by combining the CWT and USM in the signal feature extraction process. In the experiment, several different image transformation methods [35] were selected for comparison, including Gramian Angular Summation Fields (GASF), Gramian Angular Difference Fields (GADF), Recurrence Plot (RP), and Markov Transition Fields (MTF) methods. The corresponding two-dimensional images are shown in Figure 7.

![Figure 7. Data preprocessing 2D images: (a) Gramian Angular Difference Fields; (b) Recurrence Plot; (c) Markov Transition Fields; (d) Gramian Angular Summation Fields.]

The ResNet model was selected for the experiments. Where 50% data overlap was selected for all image processing methods, the signal period sampling points were the same as that of the CWT algorithm. Figure 8 depicts the results after conducting five experiments for each method and averaging the results. By observing the experimental results, it is evident that the accuracy of images processed using GASF, GADF, RP, and MTF methods in the transfer task 0-1 was below 60%. In contrast, the accuracy of images processed with the CWT-USM method in transfer task 0-1 was 85.04%, which is a 30.4% improvement compared to the second-highest accuracy achieved by MTF (54.64%), showing a significant enhancement. Additionally, in other transfer tasks, the accuracy of the RP and MTF was significantly improved compared to that of GASF and GADF, but their accuracy was still lower than that of the CWT-USM method proposed in this study. The results indicate that the CWT-USM method can extract richer and more accurate data features, significantly improving the accuracy of bearing fault diagnosis.

![Figure 8. Experimental results of comparison between different two-dimensional images.]

5.3. Comparative Experiments with Different Dimensional Inputs

To compare the impact of different dimensional inputs on fault diagnosis outcomes, we evaluated the original one-dimensional (1D) time domain signal, the 1D frequency domain signal processed by FFT, and the proposed CWT-USM method, which includes time–frequency domain information. The experiments were conducted using the ResNet model, and the results are shown in Table 7.

The results indicate that CWT-USM outperformed the 1D frequency domain input across all transfer tasks. Although the accuracy of CWT-USM was slightly lower in transfer tasks 0-3, 1-2, 1-3, and 2-1 compared to the one-dimensional time domain input, the overall average accuracy of CWT-USM was higher. Specifically, the CWT-USM method improved the average accuracy by 3.61% compared to the 1D time domain input and by 14.46% compared to the 1D frequency domain input.

These experimental results demonstrate the superiority of using CWT-USM as input. By encompassing both frequency domain and time domain information relating to the vibration signal, CWT-USM provides richer feature information, leading to better fault diagnosis performance.

Table 7. Comparative experiments with different dimensional inputs for the CWRU dataset.

<table>
<thead>
<tr>
<th>Methods</th>
<th>0-1</th>
<th>0-2</th>
<th>0-3</th>
<th>1-0</th>
<th>1-2</th>
<th>1-3</th>
<th>2-0</th>
<th>2-1</th>
<th>2-3</th>
<th>3-0</th>
<th>3-1</th>
<th>3-2</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>1D time domain input</td>
<td>80.32</td>
<td>79.84</td>
<td>75.65</td>
<td>81.77</td>
<td>99.84</td>
<td>96.69</td>
<td>74.19</td>
<td>93.87</td>
<td>94.68</td>
<td>69.11</td>
<td>72.34</td>
<td>77.42</td>
<td>82.98</td>
</tr>
<tr>
<td>1D frequency domain input</td>
<td>73.22</td>
<td>67.31</td>
<td>54.48</td>
<td>72.45</td>
<td>79.01</td>
<td>81.34</td>
<td>67</td>
<td>83.43</td>
<td>82.33</td>
<td>55.78</td>
<td>70.35</td>
<td>78.86</td>
<td>72.13</td>
</tr>
<tr>
<td>CWT-USM</td>
<td>85.04</td>
<td>80.24</td>
<td>72.32</td>
<td>95.12</td>
<td>95.28</td>
<td>87.28</td>
<td>88.6</td>
<td>89.92</td>
<td>95.64</td>
<td>79.84</td>
<td>79.2</td>
<td>90.56</td>
<td>86.59</td>
</tr>
</tbody>
</table>

5.4. Comparative Experiments on Different Domain Adaptation Strategies

To enhance the persuasiveness and general applicability of the experiments, this study introduced the PU bearing and the existing CWRU datasets. The experiments extensively compared several transfer strategies, including the baseline model without any transfer strategy (SimAM-ResNet), utilizing the Conditional Domain Adaptation Network (SimAM-ResNet-CDAN), utilizing the Maximum Mean Discrepancy (SimAM-ResNet-JMMD), a model combining CDAN and JMMD but without the adaptive weighting algorithm (SimAM-ResNet-CDAN-JMMD), and the proposed method (CWT-SimAM-DAMS). The experiments were repeated five times, and the results on the CWRU and PU datasets are presented in Tables 8 and 9, as well as Figures 9 and 10, respectively.

Figure 9. Average accuracy (%) of different domain adaptation methods on the CWRU dataset.
On the CWRU dataset, compared to SimAM-ResNet without domain adaptation algorithm or SimAM-ResNet-CDAN and SimAM-ResNet-JMMD using one domain adaptation algorithm alone, SimAM-ResNet-CDAN-JMMD, a method that combines two domain adaptation algorithms, had an improved average fault diagnostic accuracy. However, there was still a problem where the accuracy rate decreased in migration tasks compared to when using a domain adaptation algorithm alone, e.g., migration task 0-2. The proposed method, CWT-SimAM-DAMS, achieves an accuracy rate that is greater than or equal to that of other domain adaptation algorithms across all migration tasks. Additionally, this method addresses the decreased accuracy of the SimAM-ResNet-CDAN-JMMD method compared to the SimAM-ResNet-CDAN and SimAM-ResNet-JMMD methods on migration task 0-2. On the PU dataset, the proposed method showed a significant improvement in migration task 0-2 and migration task 2-1. Although it decreased in migration tasks 0-1, 1-0, and 2-0, the conditions 0-2 and 2-1 were improved by 14.03% and 13.42%, respectively, compared to the SimAM-ResNet-CDAN-JMMD method. Thus, it significantly improves the average fault diagnosis accuracy. Compared to other domain adaptation algorithms, the proposed CWT-SimAM-DAMS method exhibits stronger adaptability and accuracy. Because it adjusts the optimization objectives in real time, by comprehensively considering three optimization objectives, the classification, JMMD, and CDAN, this method reduces the distribution differences between the source and target domains, enhancing the model’s ability to diagnose bearing failures.
5.5. Model Comparison Experiment

To verify the feasibility of our proposed bearing fault diagnosis model compared to other bearing fault diagnosis models, we selected several common algorithms and models in bearing fault diagnosis for comparative verification. Each model was applied five times to obtain the average diagnostic accuracy. Figures 11 and 12, as well as Tables 10 and 11, present the experimental results comparing the CWT-SimAM-DAMS model with the competitor models. Tables 12 and 13 present the training and testing times of different models. Additionally, the performance of each method was assessed through confusion matrices, as shown in Figures 13 and 14.

Figure 11. Average diagnostic accuracy (%) of different models on the CWRU dataset.

Figure 12. Average diagnostic accuracy (%) of different models on the PU dataset.

Table 10. Average diagnostic accuracy (%) of different models on the CWRU dataset.

<table>
<thead>
<tr>
<th>Methods</th>
<th>0-1</th>
<th>0-2</th>
<th>0-3</th>
<th>1-0</th>
<th>1-2</th>
<th>1-3</th>
<th>2-0</th>
<th>2-1</th>
<th>2-3</th>
<th>3-0</th>
<th>3-1</th>
<th>3-2</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>VGG</td>
<td>33.52</td>
<td>36</td>
<td>35.76</td>
<td>40</td>
<td>42.4</td>
<td>30.8</td>
<td>36.48</td>
<td>40.4</td>
<td>49.72</td>
<td>38.96</td>
<td>32.88</td>
<td>33.84</td>
<td>37.56</td>
</tr>
<tr>
<td>AlexNet</td>
<td>76.54</td>
<td>75.04</td>
<td>74.72</td>
<td>99.12</td>
<td>90.96</td>
<td>82.4</td>
<td>85.84</td>
<td>88.8</td>
<td>97.12</td>
<td>80.48</td>
<td>80.16</td>
<td>82.24</td>
<td>84.45</td>
</tr>
<tr>
<td>LeNet</td>
<td>84.96</td>
<td>73.76</td>
<td>73.38</td>
<td>98.64</td>
<td>89.68</td>
<td>80</td>
<td>92.24</td>
<td>92.26</td>
<td>85.4</td>
<td>79.44</td>
<td>76.24</td>
<td>79.52</td>
<td>83.79</td>
</tr>
<tr>
<td>CNN</td>
<td>79.36</td>
<td>73.2</td>
<td>71.52</td>
<td>93.92</td>
<td>94.96</td>
<td>91.44</td>
<td>89.52</td>
<td>89.68</td>
<td>97.84</td>
<td>84</td>
<td>83.36</td>
<td>96.56</td>
<td>87.11</td>
</tr>
<tr>
<td>ResNet</td>
<td>85.04</td>
<td>80.24</td>
<td>72.32</td>
<td>95.12</td>
<td>95.28</td>
<td>87.28</td>
<td>88.6</td>
<td>89.92</td>
<td>95.64</td>
<td>79.84</td>
<td>79.2</td>
<td>90.56</td>
<td>86.59</td>
</tr>
<tr>
<td>CWT-SimAM-DAMS</td>
<td>100</td>
<td>99.84</td>
<td>92.08</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>99.92</td>
<td>99.84</td>
<td>99.84</td>
<td>99.29</td>
</tr>
</tbody>
</table>
Table 11. Average diagnostic accuracy (%) of different models on the PU dataset.

<table>
<thead>
<tr>
<th>Methods</th>
<th>0-1</th>
<th>0-2</th>
<th>1-0</th>
<th>1-2</th>
<th>2-0</th>
<th>2-1</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>VGG</td>
<td>41.34</td>
<td>14.74</td>
<td>38.36</td>
<td>11.08</td>
<td>19.16</td>
<td>22.548</td>
<td>24.54</td>
</tr>
<tr>
<td>AlexNet</td>
<td>71.48</td>
<td>40.12</td>
<td>79.22</td>
<td>34.41</td>
<td>29.23</td>
<td>24.62</td>
<td>43.86</td>
</tr>
<tr>
<td>LeNet</td>
<td>79.79</td>
<td>27.99</td>
<td>67.14</td>
<td>46.95</td>
<td>38.26</td>
<td>42.5</td>
<td>56.27</td>
</tr>
<tr>
<td>CNN</td>
<td>85.16</td>
<td>38.03</td>
<td>86.7</td>
<td>49.5</td>
<td>42.5</td>
<td>48.16</td>
<td>61.51</td>
</tr>
<tr>
<td>ResNet</td>
<td>83.82</td>
<td>47.38</td>
<td>87.95</td>
<td>59.26</td>
<td>42.48</td>
<td>48.16</td>
<td>61.51</td>
</tr>
<tr>
<td>CWT-SimAM-DAMS</td>
<td>97.05</td>
<td>83.81</td>
<td>98.09</td>
<td>80.37</td>
<td>71.81</td>
<td>90.46</td>
<td>86.93</td>
</tr>
</tbody>
</table>

Table 12. The average training and testing time of each model in the CWRU dataset.

<table>
<thead>
<tr>
<th>Methods</th>
<th>VGG</th>
<th>AlexNet</th>
<th>LeNet</th>
<th>CNN</th>
<th>ResNet</th>
<th>CWT-SimAM-DAMS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training time (min)</td>
<td>1.856917</td>
<td>1.658083</td>
<td>1.841167</td>
<td>2.258319</td>
<td>2.541681</td>
<td>2.431865</td>
</tr>
<tr>
<td>Test time (min)</td>
<td>0.282778</td>
<td>0.268042</td>
<td>0.316833</td>
<td>0.381056</td>
<td>0.385319</td>
<td>0.33694</td>
</tr>
</tbody>
</table>

Table 13. The average training and testing time of each model in the PU dataset.

<table>
<thead>
<tr>
<th>Methods</th>
<th>VGG</th>
<th>AlexNet</th>
<th>LeNet</th>
<th>CNN</th>
<th>ResNet</th>
<th>CWT-SimAM-DAMS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training time (min)</td>
<td>27.183528</td>
<td>25.044139</td>
<td>23.018417</td>
<td>23.274472</td>
<td>32.809889</td>
<td>30.7255</td>
</tr>
<tr>
<td>Test time (min)</td>
<td>4.097833</td>
<td>3.8351389</td>
<td>3.8351944</td>
<td>3.7904167</td>
<td>4.8703611</td>
<td>4.2688889</td>
</tr>
</tbody>
</table>

Figure 13. Visualization of confusion matrix for different models in the target domain of CWRU dataset 0-1 migration task.

The experimental results show that the CWT-SimAM-DAMS model achieved an average accuracy of 99.29% on the CWRU dataset and 86.93% on the PU dataset. Compared to several traditional bearing fault diagnosis methods, the CWT-SimAM-DAMS method significantly improves average accuracy. Specifically, the accuracy of the CWT-SimAM-DAMS method on the CWRU and PU datasets was 13.56% and 25.42% higher, respectively, than that of the traditional ResNet model. Similarly, compared to the CNN model, the CWT-SimAM-DAMS method achieved an average accuracy improvement of 12.18% and 30.66%.
on the CWRU and PU datasets, respectively. This indicates that the CWT-SimAM-DAMS model has superior feature extraction and domain alignment capabilities.

![Visualizations](image)

**Figure 14.** Visualization of confusion matrix for different models in the target domain of PU dataset 0-1 migration task.

Tables 12 and 13 show that the ResNet model had the longest training and testing times on both the CWRU and PU datasets. Although the CWT-SimAM-DAMS model has relatively long training times compared to other models, its testing time does not significantly increase. For the CWRU dataset, the training time difference between all models was less than one minute, and the testing time of the CWT-SimAM-DAMS model was only 0.068898 minutes longer than the fastest AlexNet model. For the PU dataset, the training time difference between all models was within 10 minutes, and the testing time of the CWT-SimAM-DAMS model was only 0.4784722 minutes longer than the fastest CNN model. Considering that, in practical industrial applications, model training is usually conducted offline, training time is not a critical issue compared to model accuracy. Additionally, the difference in testing time between models is not significant. Taking both model accuracy and testing time into account, the CWT-SimAM-DAMS model still has a significant advantage.

### 5.6. Ablation Study

Various ablation experiments were conducted on the CWRU dataset to verify the proposed CWT-SimAM-DAMS method, referred to as Method 1 in Table 14, and the efficacy of each component. These ablation experiments involved systematically removing key modules of Method 1 and observing their impact on the final performance, thereby revealing the contributions and importance of each module.

By comparing different combinations, it was found that removing the adaptive weighting module led to a significant decrease in performance, indicating the critical importance of the adaptive weighting module for the effectiveness of the CWT-SimAM-DAMS method. Conversely, when CWT-USM was replaced with regular CWT or the Residual Network integrated with SimAM was substituted by a standard residual network, although there was a decrease in performance, the impact was relatively minor. This indicates that, while the module image processing and the residual network integrated with SimAM contribute
to performance enhancement, their effect is not as pronounced as that of the weight adaptive strategy module. The complete Method 1 model outperformed all other combinations, validating that integrating all modules achieves the best performance.

### Table 14. CWUR dataset ablation experiment result.

<table>
<thead>
<tr>
<th>Image Processing</th>
<th>Model Selection</th>
<th>Domain Adaptation</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>CWT - USM</td>
<td>✓✓✓</td>
<td>0-1 0-2 0-3 1-0 1-2 1-3 2-0 2-1 2-3 3-0 3-1 3-2 Average</td>
<td></td>
</tr>
<tr>
<td>SimAM - ResNet</td>
<td>✓✓✓</td>
<td>100 99.84 92.08 100 100 100 100 100 100 99.92 99.84 99.84 99.29</td>
<td></td>
</tr>
<tr>
<td></td>
<td>✓✓</td>
<td>81.04 80.4 73.68 96 94.4 85.6 89.44 89.92 94.8 77.92 79.2 93.8 86.28</td>
<td></td>
</tr>
<tr>
<td></td>
<td>×</td>
<td>85.04 80.24 72.32 95.12 95.28 87.28 88.6 89.92 95.64 79.84 79.2 90.56 86.59</td>
<td></td>
</tr>
<tr>
<td></td>
<td>✓</td>
<td>96.16 90.56 93.36 99.92 99.76 99.92 97.76 98.8 100 99.92 99.9 99.8 97.92</td>
<td></td>
</tr>
<tr>
<td></td>
<td>×</td>
<td>80.08 73.5 75 93.92 94.32 84.72 85.2 89.68 94.4 78.32 77.12 85.6 84.76</td>
<td></td>
</tr>
<tr>
<td></td>
<td>✓</td>
<td>100 99.92 93.44 100 100 100 100 100 100 92.16 84.48 99.76 97.48</td>
<td></td>
</tr>
<tr>
<td></td>
<td>×</td>
<td>79.96 76.4 68.88 92.72 94.8 84.84 88.12 90.48 93.7 80.56 79.12 89.12 84.89</td>
<td></td>
</tr>
<tr>
<td></td>
<td>✓</td>
<td>99.92 97.52 90.08 99.84 99.84 99.76 100 100 100 95.92 85.68 99.84 97.37</td>
<td></td>
</tr>
</tbody>
</table>

✓ represents selecting the corresponding module, × represents not selecting the corresponding module.

### 6. Conclusions

This study proposes a bearing fault diagnosis method based on SimAM and an adaptive weighting transfer strategy. The proposed method transforms one-dimensional vibration time series signals of bearing faults into CWT images and enhances the detailed features of the images using the USM algorithm, facilitating feature extraction by the model. Integrating the SimAM attention mechanism into the residual network enhances the model’s feature extraction capability in the source domain. Additionally, by combining the JMMD and CDAN algorithms and employing a weight adaptive strategy, the domain adaptation transfer capability of the model is strengthened.

The proposed method is validated on both the CWRU and PU datasets, achieving an accuracy of 99.29% on the CWRU dataset and 86.93% on the PU dataset, representing a significant improvement compared to other models. Moreover, ablation experiments conducted on the CWRU dataset verify the importance and effectiveness of each component. The experimental results demonstrate that this method effectively reduces the distribution difference between the source and target domains, improving fault diagnosis accuracy.

In future research, further optimization of the model architecture will be pursued to enhance its generalization, and application in more realistic industrial scenarios will be explored. Additionally, refinement of model parameters will be conducted to improve both training and testing times for fault diagnosis while maintaining the model’s accuracy.

### Author Contributions:

Conceptualization, Z.T., X.H. (Xinhao Hou) and X.W.; methodology, Z.T. and X.H. (Xinhao Hou); software, Z.T., X.H. (Xinhao Hou), X.H. (Xinheng Huang) and J.Z.; validation, Z.T., X.H. (Xinhao Hou), X.H. (Xinheng Huang) and J.Z.; formal analysis, Z.T.; investigation, Z.T., X.H. (Xinhao Hou) and X.W.; resources, Z.T. and X.W.; data curation, Z.T. and J.Z.; writing—original draft preparation, Z.T.; writing—review and editing, Z.T. and X.W.; funding acquisition, Z.T. and X.W. All authors have read and agreed to the published version of the manuscript.

### Funding:

This work were supported by the key project of the National University Innovation and Entrepreneurship Training Programs Foundation under grant number 202210060002.

### Institutional Review Board Statement:

Not applicable.

### Informed Consent Statement:

Not applicable.

### Data Availability Statement:

The detailed data information of this article is included in the article. For detailed information, please contact the corresponding author.

### Conflicts of Interest:

The authors declare no conflicts of interest.
References


**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.