Intelligent fault diagnosis and health state recognition of rotating machinery under variable working conditions using deep transfer learning

by

Zehui Hua

A Dissertation Submitted to the University of Ottawa in Fulfillment of the Requirement for the Degree of

Doctorate of Philosophy

in

Mechanical Engineering

Ottawa-Carleton Institute for Mechanical and Aerospace Engineering

Faculty of Engineering

University of Ottawa

Abstract

The health of key rotating components in machinery systems, such as rolling element bearings and gears, is critical for meeting design requirements and ensuring safe operation. These components degrade over time, leading to faults that can cause unplanned downtime, economic loss, or catastrophic accidents. Vibration signal-based intelligent fault diagnosis (IFD) enables real-time condition monitoring with reduced reliance on human expertise. However, traditional machine learning methods often assume that vibration data from a source domain and a target domain share similar feature distributions, an assumption that rarely holds under variable working conditions in industrial settings. Transfer learning mitigates distribution discrepancies, yet important challenges remain: (1) effective IFD under changing operating conditions, (2) learning models that generalize across multiple domains simultaneously, and (3) transferring knowledge to a totally unseen target domain.

This thesis investigates domain generalization for vibration-based IFD under distribution shifts induced by variable working conditions and develops four methods. First, by leveraging inter- and intra-domain invariances, condition-robust representations are learned and achieve consistent improvements over strong baselines on two benchmark datasets across diverse cross-condition settings. Second, to better handle operating-condition variations, a multi-subdomain alignment strategy that introduces multiple condition-related subdomains within a single source domain and aligns them to reduce condition-dependent discrepancies is proposed, improving diagnostic performance on two bearing datasets. Third, a feature disentanglement mechanism is introduced to decouple domain-invariant from domain-specific features, enhancing discriminability and robustness under unseen conditions. Extensive experiments, including low-data regimes, show superiority over several state-of-the-art approaches. Finally, the framework for simulation-to-experiment transfer is extended, and a transferable diagnostic model that captures time-varying characteristics is developed, enabling reliable fault detection based on the obtained time-frequency representations.

Acknowledgements

First of all, I would like to sincerely thank my supervisor, Dr. Patrick Dumond, for his continuous guidance throughout my Ph.D. study. His broad knowledge, professional suggestions, and comments have been invaluable to the development of this thesis. His passion for research and his thoughtful advice have had a lasting influence on my academic growth. I am especially grateful for the trust and freedom he gave me to explore my research ideas, for his guidance during the difficult stages of this work, and for his support of my participation in academic conferences.

My sincere thanks also go to Dr. Natalie Baddour, Dr. Eric Lanteigne, Dr. Jie Liu of Carleton University, and Dr. Yuejian Chen of the University of Manitoba for serving as my thesis committee members. Their insightful comments helped improve the thesis, and their constructive questions motivated me to broaden my research perspective.

I would also like to express my sincere gratitude to my master's supervisor, Dr. Juanjuan Shi, for her long-term support and encouragement. Her academic advice and discussions have continued to benefit my research and broaden my horizons throughout my studies.

I am grateful to my colleagues in the Dumond Design Lab for their help, discussions, and friendship. The research journey would have been much more difficult without the encouragement I received from my colleagues and friends.

I gratefully acknowledge the financial support provided by the University of Ottawa and the China Scholarship Council, which made it possible for me to pursue this research and complete my doctoral studies.

Most importantly, I would like to thank my family for their love, patience, and belief in me. Their support gave me strength during many difficult moments.

Table of Contents

List of Figures

List of Tables

Acronyms

AM	Amplitude Modulation
CWT	Continuous Wavelet Transform
CNN	Convolutional Neural Network
CORAL	Correlation Alignment
DA	Domain Adaptation
DANN	Domain Adversarial Neural Network
DBN	Deep Belief Network
DG	Domain Generalization
DIFE	Domain-invariant Feature Exploration
DIS	Domain Interference Suppression
DTL	Deep Transfer Learning
EEMD	Ensemble Empirical Mode Decomposition
EMD	Empirical Mode Decomposition
ERM	Empirical Risk Minimization
FCC	Fault Characteristic Coefficient
FCF	Fault Characteristic Frequency
FCO	Fault Characteristic Order
FFT	Fast Fourier Transform
FD	Fault Diagnosis
FM	Frequency Modulation
FSL	Few-Shot Learning
GAN	Generative Adversarial Network
GPU	Graphics Processing Unit
GRL	Gradient Reversal Layer
IF	Instantaneous Frequency
IFD	Intelligent Fault Diagnosis
IID	Independent and Identically Distributed
IMSST	Iterative Matching Synchrosqueezing Transform
ISRF	Instantaneous Shaft Rotating Frequency

JMMD	Joint Maximum Mean Discrepancy
KD	Knowledge Distillation
KNN	K-Nearest Neighbor
LCT	Linear Chirplet Transform
LMD	Local Mean Decomposition
MK-MMD	Multi-Kernel Maximum Mean Discrepancy
MMD	Maximum Mean Discrepancy
MSE	Mean Squared Error
MSST	Multiple Synchrosqueezing Transform
PHM	prognostics and health management
ResNet	Residual Network
RKHS	Reproducing Kernel Hilbert Space
RM	Reassignment Method
RNN	Recurrent Neural Network
RUL	Remaining Useful Lifetime
SAM	Self-Attention Mechanism
Semi-SL	Semi-Supervised Learning
SET	Synchroextracting Transform
SNR	Signal-to-Noise Ratio
SRF	Shaft Rotating Frequency
SSL	Self-Supervised Learning
SST	Synchrosqueezing Transform
SSTFA	Self-Supervised Time-Frequency Analysis
STFT	Short-time Fourier Transform
TF	Time-Frequency
TFA	Time-Frequency Analysis
TFR	Time-Frequency Representation
t-SNE	t-Distributed Stochastic Neighbor Embedding
VMD	Variational Mode Decomposition
WT	Wavelet Transform

Chapter 1 Introduction

1.1 Basic concepts

Fault diagnosis is of great importance in assuring the safe operation of rotating machinery systems. Bearings, the key rotating components in rotational machinery, ensure the smooth operation of the whole system when in their normal state. If an unexpected bearing failure happens, system failure results, causing delays and even accidents. To help monitor the running operation of a system, bearing fault diagnosis and prognosis play an important role [1]. To perform fault diagnosis, vibration data is usually analyzed since this data contains vast amounts of health information. With the development of signal processing techniques, researchers and expert engineers can understand the health state of systems and then make a maintenance decision based on the analysis of results. This has become known as prognostics and health management (PHM). IFD utilizes an end-to-end deep learning model to automatically extract features from collected vibration signals [2]. Driven by mass data processing, IFD is also playing a pivotal role in Industry 4.0. Once a model is well-trained, it can provide accurate and real-time fault diagnosis results. Combined with transfer learning, models have the potential to provide promising results for new data collected under variable working conditions [3].

For vibration signals, the data collection procedure can be considered to be a stochastic process due to strong non-stationarity, especially when the working conditions are changing (e.g., varying speed and load) [4]. By first establishing the relationship between given labels and the collected vibration signals using deep learning, then embedding a transfer strategy to find the generalized domain-invariant features, a more generalized model can be simultaneously applied to different objects, even though the working conditions are variable. Thus, it is possible to realize the health state recognition of critical rotating components in machines and schedule maintenance with the goal of identifying potential failures, minimizing shutdowns, and avoiding heavy economic losses or catastrophic accidents [5].

According to existing research, IFD requires the availability of voluminous data to train constructed diagnosis models, otherwise models may perform poorly. Available data refers to data that is collected from sufficient typical faults and that has been correctly labeled. However, in engineering scenarios, such data are limited. First, fault data is more difficult to collect than normal data, resulting in insufficient types of fault patterns. Second, labeling monitoring data is very expensive. For example, it is unrealistic to frequently inspect a mechanical system's health state, which is a way to manually label data. Though sometimes a specific condition can be acknowledged by employing signal processing techniques, it also requires costly professional engineers with expert knowledge to spend a great deal of time making maintenance decisions.

Transfer learning is a promising way to solve the problem of limited labeled data [2]. The need for transfer learning stems from the fact that deep learning assumes that training data and testing data share the same distribution. However, this assumption is not always applicable to transfer learning. For instance, labels from the sample of interest in the target domain are often unavailable, so a reliable model cannot be trained based on these samples. Nonetheless, there exists another domain of interest in which sufficient data with labels is available, but the data in these two domains share different distributions. Therefore, successful knowledge transfer means that the performance and effectiveness of learning and training procedures can be improved without expensive and complicated data labeling. The goal of transfer learning is then to use knowledge learned from the source domain to achieve high accuracy when testing the target domain [6]. Since data in the source and target domains share some similarities, diagnosis knowledge learned from the source domain can be further used for the target domain to help recognize the health state of the target data, rather than training a new model. Hence, it is necessary to further study the feature distribution discrepancies between different domains to generate a more generalized and robust model that applies to multiple domains simultaneously. As a result, the demand for high model generalization promotes the development of deep transfer learning (DTL), in which deep learning acts as a powerful feature-extracting tool and transfer learning improves the performance of the trained model by sharing learned knowledge when dealing with limited available data.

The final goal of developing IFD based on DTL is to train a more generalized model that works well on unseen target data. Specifically, DTL will be used to transfer knowledge learned from one or multiple known source domains to an unknown target domain using an unsupervised machine learning method wherein labels are unavailable. To make the health state recognition task more challenging, target domain data can also be unavailable ahead of time (i.e., both samples and labels for the target domain data are unavailable and unseen beforehand).

1.2 Objectives

According to the aforementioned challenges, the objectives of the thesis are to improve the generalization ability of intelligent bearing fault diagnosis models under variable operating conditions and to extend the representation learning perspective to adaptive time-frequency (TF) analysis (TFA) by studying whether representations learned from synthetic signals can be transferred to real nonstationary signals. Specifically, the objectives of this thesis are:

(1) Improve capabilities for domain generalization across variable working conditions and unknown domain shifts, ensuring stable performance for new tasks;
(2) Design algorithms that integrate semi-supervised learning to effectively utilize scarce labeled samples and abundant unlabeled data from real industrial systems;

(3) Investigate domain generalization methods that allow models to retain performance on previously known working conditions without requiring access to target operating conditions during training;
(4) Develop automated feature extraction or end-to-end representation learning approaches that eliminate extensive manual preprocessing and extract discriminative time-varying features directly from raw signals, especially for nonstationary feature extraction under time-varying speeds.

1.3 Contributions

By targeting the objectives, the contributions of the thesis are:

(1) Development of a more generalized model for bearing fault diagnosis by exploring both mutually and internally domain-invariant features;
(2) Development of a more generalized model for bearing fault diagnosis by introducing pseudo domain labels that consider the dynamic feature distributions of data collected under variable working conditions;
(3) Achievement of better domain generalization for bearing fault diagnosis by regulating the domain-specific features induced by domain variations;
(4) Development of a self-supervised TFA model for fault-related feature representation.

1.4 Organization of the thesis

This manuscript-based thesis aims to improve bearing fault diagnosis under varying operating conditions by developing methods that can learn more transferable and fault-discriminative representations while reducing the influence of domain variations. To achieve this goal, the thesis addresses several related challenges, including the extraction of informative fault features, the modeling of domain shift, the suppression of domain-specific interference, and the development of signal representations for non-stationary signals. The chapters are organized to present these research problems in a coherent manner, with each chapter focusing on a specific aspect that contributes to the overall study.

Chapter 2 reviews existing methods for bearing fault diagnosis, including signal-processing-based methods for bearing fault signature extraction and DTL-based diagnostic methods. This chapter establishes the research background and highlights the limitations of current approaches in cross-domain fault diagnosis. Building on this foundation, Chapter 3 investigates internally domain-invariant features to enable the extraction of more diverse and transferable representations. Chapter 4 further studies dynamic feature distributions by introducing pseudo domain labels, thereby providing a more refined way to characterize domain variation during feature learning. Based on these insights, Chapter 5 proposes a new model for IFD that regulates domain-specific

features by suppressing the influence of domain variations while retaining fault-discriminative information. In addition, Chapter 6 proposes a self-supervised deep-learning-based TFA method for non-stationary signals, which can provide adaptive TFA results without requiring instantaneous frequency (IF) information. Finally, Chapter 7 concludes the thesis and suggests directions for future work.

Chapter 2 Literature review

2.1 Fault diagnosis: signal processing and intelligent methods

Acting as key components in many mechanical systems, rotating machinery connect and drive different parts of these systems, ensuring efficient and smooth operation. Vibration signals collected from accelerometers placed on this machinery contain considerable health information about key rotating parts, including bearings and gears, which in turn make it possible to perform fault diagnosis using vibration signal analysis [1]. Generally, signal processing techniques applied to vibration signals can perform a fault diagnosis task by exploring local maximum amplitude peaks in the corresponding Fourier spectrum [7]. For a constant speed condition, transforming the original time domain signal into the frequency domain allows fault-related frequency components to be detected by comparing to known local peaks in the spectrum and the calculated fault characteristic frequencies (FCFs) based on bearing or gear parameter specifications [8,9]. Specifically, since the shaft rotation frequency (SRF) is constant, the FCFs are also constant and should be proportional to the SRF, where the ratios are also known as fault characteristic coefficients (FCCs). The relationship between the FCFs, FCCs, and SRFs can be expressed as FCF = FCC × SRF, where the FCC is only determined by the specific parameters of the bearing or gear [3]. Tracking the FCCs in a signal is known as fault characteristic order (FCO) analysis [10].

Due to limitations in the ability to mount a rotational speed encoder directly on the shaft in many industrial applications, the actual speed of rotating machinery is often unavailable. In this case, advanced signal processing techniques that do not require an estimate of the rotation speed are developed further. Besides, background noise and inevitable interference can cause the vibration signal to suffer from heavy contamination, making it much more difficult to identify local maximum peaks related to faults, especially for early fault diagnosis when fault-related features are weak compared to heavy noise. To deal with this problem, some researchers have proposed filtering-based methods to minimize the side effects of noise, as well as adaptive signal decomposition techniques like empirical mode decomposition (EMD) [11], local mean decomposition (LMD) [12], ensemble empirical mode decomposition (EEMD) [13], and variational mode decomposition (VMD) [14]. These signal decomposition methods have been proven effective in tackling heavy noise. The contributions of these methods lie in their ability to divide the original signal into several different modes to minimize the side effects of heavy noise. Combining filtering methods with signal decomposition methods is also a recommended approach. That is, filtering the original signal first, based on the target frequency range, and then performing signal decomposition to further analyze frequency components with lower energy levels works well.

However, these methods only apply when rotating machinery operates under constant speed conditions. These methods become ineffective when rotating machinery operates under time-varying speed conditions. The typical problem associated with signal processing under time-varying speed conditions involves the corresponding Fourier spectrum suffering from mode aliasing, which makes local peaks of the spectrum appear over a wide frequency range rather than a fixed frequency point. Therefore, TFA is widely adopted to perform signal analysis in these cases. TFA provides additional insight into the changing pattern of the time-varying frequency trajectory over time. The resulting time-frequency representation (TFR) is a two-dimensional matrix with a certain value at each TF point compared to the original Fourier spectrum. TFRs are a stacked figure of different spectra truncated by a window function centered at each time instance, where the amplitudes are usually reflected by different colors, corresponding to energy concentration levels of the target frequency components. Thus, for rotating machinery operating under time-varying speed conditions, the FCO analysis approach cannot be directly applied to judge fault types based on peaks in the Fourier spectrum. To effectively use FCO analysis, signal resampling must be conducted first. For instance, the original signal should be resampled based on the estimated instantaneous shaft rotating frequency (ISRF), by mapping the original fault-related features to a new fault phase domain, the changing pattern of fault orders of the resampled signal can then be considered the same as those under a constant speed condition.

However, to properly estimate the ISRF remains difficult. To solve this problem, there are two different approaches: (1) developing a more advanced or accurate ridge detection algorithm rather than extracting the trajectories with maximal amplitude at each time instant, and (2) matching the time-varying pattern of the frequency components to enhance the energy concentration levels of the feature components of interest so that a peak searching algorithm still works. By implementing these two strategies, the IF trajectories under time-varying speed conditions can be detected. Then, specific fault types can be determined according to the ratios of the detected IF ridges by comparing the FCCs. This is because the ratios remain the same as those under a constant speed condition, as determined by the specific parameters.

A class of TFA involves post-processing techniques. The main contribution of post-processing is reallocating the energy distribution of the TFR generated by a short-time Fourier transform (STFT), or wavelet transform (WT), under the guidance of the IF estimator, when truncating the signal [15]. Following the definition, the IF estimator is free from noise and only determined by the ratio of the TFRs generated by using different types of window functions. Methods such as the synchrosqueezing transform (SST) [16,17], synchroextracting transform (SET) [18], reassignment method (RM) [17], and multi-synchrosqueezing transform (MSST) have been studied by many researchers [19]. It is worth noting that the IF estimator in the SST and SET is almost the same.

Namely, the SST reassigns the TF energy along the frequency axis, while the SET only extracts the TF energy from the original TF plane guided by the calculated IF estimator. More recently, the MSST has been shown to employ multiple reassignment operations to generate more energy-condensed TFRs compared with the original SST. Other methods include the iterative matching synchrosqueezing transform (IMSST), as well as the high-order synchroextracting transform [20,21]. The main contributions of these methods are that more accurate IF estimators are adopted to guide the reassignment operation, especially when analyzing signals with rapidly changing frequencies. Lastly, since the SST and SET methods only reassign the TF energy in the frequency dimension, signal reconstruction is possible, and the main frequency components of interest can be recovered from the resulting TFRs. Also, it is worth noting that the side effects of noise are also reassigned, guided by the IF estimator, leading to a lower TF concentration level. Therefore, to characterize the time-varying frequency components, more advanced TFA methods can generate a more condensed TFR with higher energy concentration levels. TFA can be used to help prepare a dataset for IFD using two-dimensional inputs by transforming the original one-dimensional time domain signal to a two-dimensional TF plane, which can characterize TF details of the data in the time and frequency domains at the same time.

Although the above studies have provided effective tools for vibration-based fault diagnosis and TFA, several limitations remain. First, fault-related feature extraction becomes more difficult when frequency components vary over time. Under time-varying speed conditions, fault-related components no longer appear as stable peaks at fixed frequency locations, and their energy may spread over a wider frequency range, which weakens the effectiveness of conventional spectrum-based diagnosis. Second, practical vibration signals usually contain multiple interacting components, including shaft-related components, modulation sidebands, structural resonances, background noise, and interference from other mechanical parts. These components may overlap or cross in the time-frequency plane, making fault-related ridges difficult to separate reliably. Third, fault-related impulses usually exhibit nonstationary oscillatory decay rather than ideal periodic impulse patterns. This issue becomes more pronounced under time-varying speed conditions, where the impulse intervals, local waveform morphology, and energy distribution are affected by speed variations and component coupling. Therefore, obtaining an informative and concentrated signal representation is necessary for subsequent fault characteristic analysis and diagnostic decision-making.

2.2 Signal processing-based vibration data analysis

To illustrate the basic procedure of signal-based fault diagnosis, a simulated bearing fault signal is first constructed in this section. Under a local bearing fault, the vibration signal usually contains a sequence of fault-related impulses. The occurrence rate of these impulses is determined

by the FCF, which depends on the rotation frequency $f_{\mathrm{ISRF}}(t)$. For instance, the rotation frequency is expressed as:

$$f_{ISRF}(t)=\begin{cases}30t+15, 0\le t<1\\ 45, 1\le t\le 2\end{cases} \tag{2-1}$$

The simulated FCC is set as 3.6, which means that the FCF can be further written as:

$$f_{FCF}(t)=3.6f_{ISRF}(t)=\begin{cases}108t+54, 0\le t<1\\ 162, 1\le t\le 2\end{cases} \tag{2-2}$$

The fault-related impulses are generated according to the accumulated fault phase, defined as:

$$\theta(t)=2\pi\int_0^t f_{FCF}(\tau)\mathrm{d}\tau \tag{2-3}$$

The kth fault impulse occurs when the accumulated phase reaches $2\pi k$, namely:

$$\int_0^{t_k} f_{FCF}(\tau)\mathrm{d}\tau=k \tag{2-4}$$

This phase-based formulation indicates that the fault-related impulses are not uniformly spaced in the time domain when the rotating frequency varies with time. Each fault impulse is assumed to excite a damped sinusoidal response. The impulse response is defined as:

$$h(t)=e^{-\alpha t}\sin(2\pi f_n t)u(t) \tag{2-5}$$

where $u(t)$ is the unit step function, f_n =2500 Hz is the resonance frequency, and α=900 is the damping coefficient. Therefore, the simulated clean vibration signal can be written as:

$$x(t)=\sum_{k=1}^{K} e^{-\alpha(t-t_k)}\sin\left(2\pi f_n(t-t_k)\right)u(t-t_k) \tag{2-6}$$

To show the procedure of signal-based processing, a simulated signal is first constructed for illustration, where the frequency first increases and then remains constant. The simulated signal $x(t)$ and the rotation frequency $f_{\mathrm{ISRF}}(t)$ are given in Fig. 2-1 (a)-(b). It can be found that both the ISRF and FCF are time-varying. Under variable speed conditions, the rotating frequency $f_{\mathrm{ISRF}}(t)$ first increases with time. As a result, the time interval between two adjacent fault impulses changes with the instantaneous rotating speed.

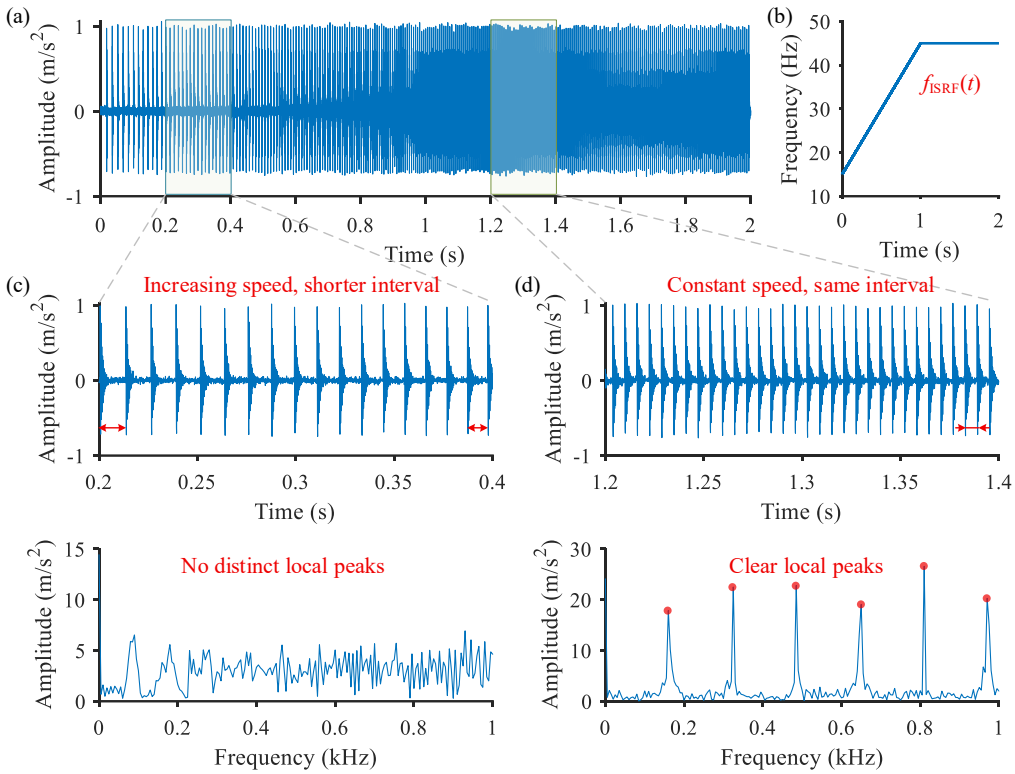


Fig. 2-1. Simulated noise-free bearing fault signal: (a) time domain signal waveform, (b) the rotating frequency, (c) signal with increasing speed (0.2-0.4 s) and its corresponding Fourier spectrum, and (d) signal with constant speed (1.2-1.4 s) and its corresponding Fourier spectrum.

The difference between variable speed and constant speed conditions can be observed from the Fourier spectra of two selected signal segments. During the increasing speed period from 0.2 s to 0.4 s, the shaft rotating frequency changes from 21 Hz to 27 Hz. Accordingly, the fault characteristic frequency varies from 75.6 Hz to 97.2 Hz. Since the FFT represents the selected signal segment using stationary sinusoidal components, the time-varying FCF cannot be concentrated at a single spectral peak. Instead, the fault-related energy is distributed over the corresponding frequency range, as shown in Fig. 2-1 (c). In contrast, during the constant speed period from 1.2 s to 1.4 s, the fault impulses occur with uniform time intervals. Therefore, the fault-related components are more concentrated in the Fourier spectrum, leading to clear spectral peaks at the FCF and its harmonics, as shown in Fig. 2-1 (d).

However, identifying fault-related spectral peaks is only a preliminary step in fault diagnosis. In practical applications, the measured vibration signals are often contaminated by noise and other interfering components. To simulate this scenario, additive noise is introduced into the generated signal, and the signal-to-noise ratio (SNR) is set to 0 dB. This means that the average signal power and the average noise power are equal. The noisy simulated signal waveform is shown in Fig. 2-2 (a). Similar to the noise-free case, two segments are picked for FFT analysis, and their results are

given in Fig. 2-2 (c)-(d), respectively. Compared with the noise-free case, the fault-related impulses and local peaks become less distinguishable in both the time and frequency domains.

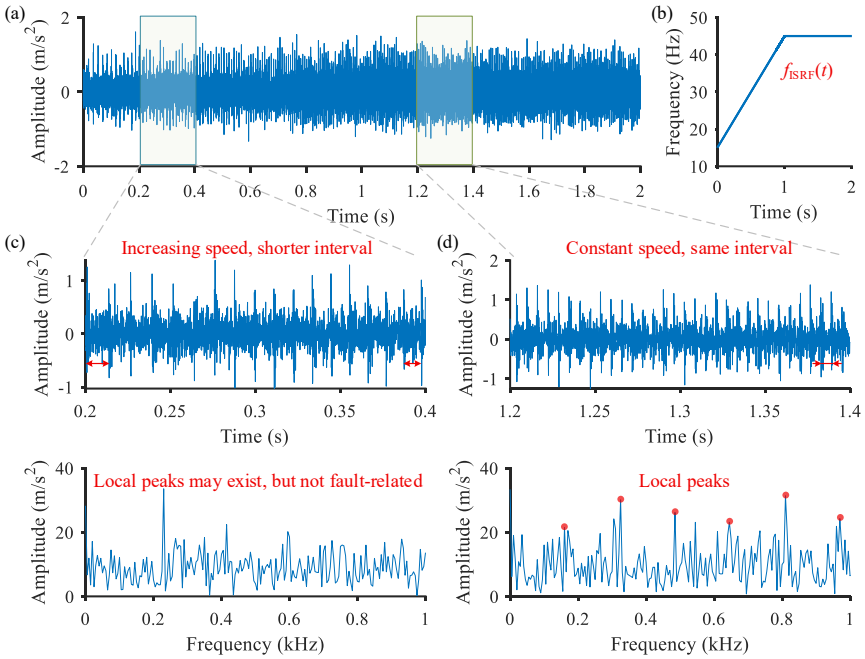


Fig. 2-2. Simulated noisy bearing fault signal (SNR=0 dB): (a) time domain signal waveform, (b) the rotating frequency, (c) signal with increasing speed (0.2-0.4 s) and its corresponding Fourier spectrum, and (d) signal with constant speed (1.2-1.4 s) and its corresponding Fourier spectrum.

For the increasing speed segment in Fig. 2-2 (c), spectral components are difficult to identify because the fault frequency variation and the added noise jointly cause spectral smearing and amplitude masking. Compared to the varying speed case, some fault-related spectral peaks in Fig. 2-2 (d) can still be clearly observed, but several components are weakened or partially buried by noise. With further increases in noise intensity, these fault-related components may become almost indistinguishable from the background noise.

These observations indicate that direct diagnosis based on manually identified spectral peaks can be unreliable under variable speed and noisy conditions. This motivates the use of data-driven fault classification models, which aim to learn discriminative representations from vibration signals and reduce the reliance on manually selected frequency domain features.

2.3 Deep learning-based intelligent fault diagnosis

With the development of the Internet of Things (IoT), massive monitoring data can now be acquired, leading to the big data era. IFD is now primed to meet many new opportunities, but it also continues to face many challenges. Nowadays, modern mechanical systems usually need to

cooperate with different system groups to achieve tasks, which means that condition monitoring systems must continuously return monitoring data. Furthermore, it has also become necessary to set a high sampling frequency for certain applications so that more health information can be acquired across a larger frequency band. However, in a run-to-failure life cycle, the healthy state accounts for most of the data collected, while failure may only occur within a short time period at very large intervals. Therefore, collected data is usually unbalanced, where healthy data is more convenient and easier to collect than faulty data. Similarly, due to sensor mounting limitations or the inevitable interference of background noise, it is also difficult to ensure the quality of the collected vibration signals themselves. Due to the complicated design of mechanical systems, multiple sensors may need to be used for health condition monitoring (e.g., accelerometer, acoustic emission, encoder, and current clamp), capturing data that reflects complementary information. Since there are mathematical transfer functions between different sensors, this provides an opportunity for data fusion. Finally, monitored data represents dynamic time sequences, which contain sufficient real-time information. Thus, it is necessary to adopt actions once the incipient fault can be detected.

As such, it is beneficial to improve IFD through monitoring big data. Sufficient data ensures that IFD can produce fault diagnosis results accurately, and even shows promise in making diagnosis decisions. By using edge computing and graphics processing units (GPUs), IFD has the capability of handling large volumes of data, potentially achieving accurate and online diagnosis.

In conventional signal-based fault diagnosis, sensitive features are extracted from vibration signals before making diagnostic decisions. However, diagnosis results rely greatly on the extracted feature distributions of the fault patterns. Moreover, the procedure of feature extraction is conducted manually, where engineers or users must design algorithms to artificially obtain features, either in the original time domain or in the frequency domain. However, for a large volume of monitoring data, extracting specialized features is unrealistic with expert knowledge due to the large labor cost. Moreover, feature selection is also required in further analysis.

The advantage of IFD is that it is an end-to-end solution that avoids human intervention and does not require relevant in-depth knowledge of the rotating machinery. Nonetheless, high accuracy is recommended to shorten the maintenance cycle. Once a high-accuracy model has been well trained, it can process large amounts of data efficiently and accurately, making it possible to automatically detect and recognize the health status of machines. However, there are also drawbacks or constraints that need to be addressed further. First, deep learning depends greatly on the quality of the dataset. Specifically, abundant labeled samples should be used during the training of a network, which is usually difficult to satisfy. Second, deep learning requires strict requirements for the distributions between training and test data, like ensuring that training and

test datasets are independent and identically distributed (IID). It is obvious that if training and testing data show a distribution discrepancy, the accuracy of the transfer task between these two datasets will decrease. Therefore, to train a more generalized model that ensures high accuracy between data coming from variable working conditions (also known as different domains), transfer learning is considered by minimizing differences between source and target domains.

Combining the advantages of deep learning for feature representations and the benefits of transfer learning for knowledge transfer, DTL, a new paradigm of machine learning, is rapidly being developed. DTL is preferred in many practical applications since it can be more easily integrated with deep learning models and can make deep learning models more reliable and accessible, while providing more robust results when performing fault diagnosis on rotating machinery. There are three main strategies for DTL, listed as follows: (1) instance-based DTL, (2) model-based DTL, and (3) feature-based DTL. However, these types of DTL approaches are interrelated, making it difficult to categorize them explicitly. The instance-based method aims to train a more precise deep model under a transfer scenario where differences between source and target domains only come from either the marginal probability distribution or the conditional probability distribution, assuming that labeled samples/instances in the target domain are too limited for training a satisfactory diagnosis model. Model-based methods focus on the transfer assumption that tasks between source and target domains share some common knowledge at the model level, indicating that transferable knowledge is well embedded into a pre-trained source deep model whose parameters and structure are general and helpful for learning a powerful target model. Feature-based methods endow deep models with the ability to transfer knowledge by learning common representations at the feature space level, rather than in the instance and model levels, which further relaxes assumptions and allows differences in feature spaces to exist in source and target domains. An intuitive solution behind feature-based DTL is to learn mapping functions as a bridge that converts raw data in source and target domains from different feature spaces to a common latent feature space (i.e., reproducing kernel Hilbert space (RKHS)) [23]. There should be a space in which raw data from source and target domains have a minimum feature distribution discrepancy. A crucial problem of feature-based DTL in learning domain-invariant features is how to estimate and learn representation invariance between source and target domains. There are several solutions to this problem: (1) leveraging discrepancy-based criteria to reduce distribution differences, (2) adding domain discriminative architectures to encourage confusion through an adversarial mechanism, and (3) combining data reconstruction to help improve representation invariance.

Aside from the methods and categories introduced above, it is worth noting that there also exist many hybrid methods to build DTL models using several of these techniques simultaneously.

The core idea of these hybrid methods is that domain-invariant knowledge between source and target domains can be learned in any two or more of the (instance, model, and feature) levels.

Finally, some researchers aim to model the evolutionary fault process of bearings from healthy to unacceptable failures. By constructing a health indicator and predicting the remaining useful lifetime (RUL) of the bearing, the specific health state of a bearing can be determined during each period, making it possible to provide a potential solution for detecting faults at an early stage. Specifically, collected vibration signals are used for bearing condition degradation assessment by generating a health indicator using the characteristics of bearing fault signals. Then, combined with a statistical model and Bayesian inference on its parameters, RUL predictions may be possible.

Various methods have been developed to perform IFD and TFA via domain adaptation (DA). Li et al. propose a multi-layer DA method to minimize the multi-kernel maximum mean discrepancy (MMD, MK-MMD) for adapting to learned representations [22,23]. Han et al. were the first to elaborate joint distribution adaptation for conditional distribution and marginal distribution at the same time to better perform IFD [24]. Ma et al. propose a weighted transfer component analysis method to reduce distribution differences [25]. Feng et al propose a similarity-based meta-learning method guided by adversarial learning to study the domain-discriminative error [26]. It can be found that these DA methods focus primarily on the alignment of two domains (a single source domain and a single target domain), and the training procedure involves access to unlabeled target domain data, which, of course, limits the use of this kind of method in real industrial applications. To tackle a more challenging IFD problem, some researchers have also studied multi-source DA, which assumes that there are multiple known source domains at the same time. In this case, the goal was to make full use of the known samples. Some typical methods focus on adjusting the coefficients of each source domain to simulate the distribution of the target domain. For instance, Xu et al. propose multi-source alignment DA by measuring the similarity in cross-domain fault diagnosis [27]. Zhang et al. propose a cross-supervised multisource prototypical network for multi-source few-shot fault diagnosis [28]. To further increase the level of difficulty, more limitations are added (i.e., limiting available samples in the training set). Two typical applications include IFD combined with few-shot learning (FSL) and semi-supervised learning (Semi-SL) [29–32]. FSL only requires a few samples in the training set, while Semi-SL only requires partial, fully labeled source domain data. These two types of applications are also in line with real industrial applications, simulating scenarios where sufficient training data are unavailable. Furthermore, when the target domain is totally unseen during training, IFD falls within the area of domain generalization (DG), which is also known as out-of-distribution generalization. The main novelty of DG lies in the absence of target-domain data during training, requiring the model to acquire truly domain-invariant knowledge and to generalize effectively to new operating

conditions or unseen environments without any prior exposure to their potential distributions. By removing the reliance on target samples, DG methods take an important step toward realistic industrial deployment, where collecting, labeling, or even accessing target-domain data is often impractical or impossible. Consequently, DG provides a more robust and practical framework for developing IFD models that can maintain stable performance under variable working conditions. Aimed at these scenarios, multi-source DA can be extended by only aligning the samples across multiple source domains, which has been widely studied in the current literature. Some other researchers have also studied how to better extract domain-invariant features by designing their own frameworks. Once more generalized features can be learned, a model's performance on new data can be further improved. To better extract domain-invariant features, metric learning and adversarial learning are effective [33,34]. There have also been some attempts at modifying network structures to maintain an appropriate gradient update through deeper layers, including short-cuts in the residual network (ResNet) and structural reparameterization [35,36]. By adding shortcuts and scale factors into the CNN networks themselves, nonlinear features can be learned. Data augmentation and domain augmentation methods also serve as promising solutions for DG-based IFD. Data augmentation focuses on synthesizing samples that mimic potential target-domain conditions, while domain augmentation enriches the training set by combining or perturbing multiple known source domains. Rather than explicitly estimating the unknown target-domain distribution, which is inherently difficult, these augmentation strategies aim to approximate the range of possible target distributions through controlled simulations. This approach provides a more practical pathway for improving model robustness against unseen working conditions [37].

In industry, there exist many IFD implementation problems that have attracted considerable attention, and significant emphasis has been placed on solving these problems. It is important to understand the types of problems that IFD faces and how to solve them. In the case of current problems faced by the manufacturing industry, the most pressing issues encountered when applying intelligent methods for machines are summarized as follows:

(1) Unknown domain shifts under variable working conditions: due to unknown distribution differences between data collected under variable working conditions, trained models are not robust enough to be generalized to new tasks.

(2) Limited labeled data in the known source domains: if sufficient training data samples are available in the source domains, the model can be well-trained after a certain number of epochs. Unfortunately, sufficient training data and labels are not usually available in practical applications. Therefore, how to effectively perform IFD combined with fewer labeled data must be explored.

(3) Invisible data in the target domain: in line with the actual needs of industrial applications, domain generalization-based fault diagnosis algorithms need to be studied so that models can sustain effective performance on unseen target domains without accessing target data.

(4) Dependence on manual data preprocessing and feature characterization: although machine learning models can perform IFD, many pipelines still rely on manual feature characterization, such as generating two-dimensional time-frequency representations via TFA, since TFA is a useful tool when revealing time-varying features in the TF plane.

2.4 Motivation

According to the literature, it is hard to train IFD models if there are unknown domain shifts, especially when target domain data is totally unseen. To ensure high accuracy performance of these training models, more generalized models should be studied by investigating shared features.

To enhance the DG capacity of IFD models, a deeper exploration of domain-invariant features is essential. Although existing studies have proposed several effective approaches for extracting such features, current methods remain limited in diversity and scope. For instance, the accuracy performance for unseen target data cannot always be guaranteed, so more effective domain-invariant features should be extracted to ensure the model is still effective under unknown domain shifts. At the same time, limited training sample scenarios also make it challenging to accurately predict actual health states. Investigating a broader range of domain-invariant representations would not only improve a model's robustness across unseen operating conditions but also contribute to a clearer interpretation of what constitutes domain invariance in rotating machinery systems. This expanded understanding is crucial for developing more reliable and interpretable DG-based fault diagnosis frameworks.

Inspired by the strong feature-characterization capability of deep learning models, this study proposes an adaptive TFA method that will further improve the efficiency of IFD under variable working conditions. The adaptive TFA framework is designed to automatically adjust its representation process to the underlying characteristics of the input signal, enabling more accurate extraction of discriminative fault-related features across diverse operating environments. After characterizing features using the proposed TFA model, generated TFRs will be fed directly into an end-to-end IFD model. This not only eliminates the need for manual feature engineering or expert knowledge but also provides a more user-friendly diagnostic workflow.

Chapter 3 Domain-invariant feature exploration for intelligent fault diagnosis under unseen and time-varying working conditions

This chapter addresses objectives 1, 2, and 3. Specifically, domain-invariant features are studied by exploring both mutually and internally invariant features. The tasks designed here aim to deal with unseen target domain data. The scenario involving limited samples is also tested by limiting access to training data.

The contents of this chapter have been published in *Mechanical Systems and Signal Processing*.

Zehui Hua, Juanjuan Shi, Patrick Dumond, Domain-invariant feature exploration for intelligent fault diagnosis under unseen and time-varying working conditions, *Mechanical Systems and Signal Processing*, 224, 2025, 112193.

Authorship contribution statement:

Zehui Hua: writing of original draft, writing review and editing, validation, algorithm development and implementation, methodology, funding acquisition, conceptualization;

Juanjuan Shi: writing review and editing, funding acquisition;

Patrick Dumond: writing review and editing, validation, supervision, project administration.

3.1 Abstract

DTL is effective when performing IFD because of its strong feature representation performance when characterizing vibration signals under variable working conditions. However, when target domain data is not available, the ability to train a model effectively could be very challenging since the feature distribution of the target domain does not contribute to the training procedure of the model. To deal with this scenario, a domain-invariant feature exploration (DIFE) method is proposed for IFD under unseen target working conditions. As the name suggests, domain-invariant features refer to the shared and common features that do not change among different working conditions when performing IFD. To further explore these transferable features, DIFE first divides the originally invariant features into two different groups: (1) the internally invariant features, which are embedded in an individual domain and obtained by using a Fourier transform, and (2) the mutually invariant features—features shared across multiple working conditions by aligning these domains. To increase computational efficiency, knowledge distillation (KD) is also used here to capture the internally domain-invariant features, which also helps save on FFT operations for unseen target domain data. Feature fusion is used to formulate the final domain-invariant features since the originally invariant features are divided into two different groups. To ensure diversity of the extracted features, their differences should be maximized. Two experiments indicate that the proposed DIFE method could provide a better domain-invariant feature representation and successfully solve the cross-domain diagnosis problem under unseen working conditions.

3.2 Introduction

3.2.1 Background

Due to varying operational conditions, DL models typically only retain high accuracy when training data and testing data share the same feature distribution (i.e., the IID assumption). However, this is not often the case, especially when it comes to variable working conditions (i.e., continuously varying speeds and loads).

3.2.2 Domain adaptation

To train a model that works both on the source and target domains, transfer learning is proposed. The key to performing transfer learning is to tackle domain shifts caused by different working conditions and minimize the domain discrepancy. By minimizing differences, the trained model could be applied to these two domains and get satisfactory accuracy at the same time. DTL utilizes the strong feature representation embedded in deep learning, while also taking advantage of the DA provided by transfer learning. After automatically extracting high-dimensional features, the distribution discrepancy between the source and target domains is minimized in model training,

making these two domains share similar features. DA involves learning cross-domain features by using distance learning and adversarial learning to help align the two different domains. Ma et al. used weighted transfer component analysis to study how to perform better transfer tasks across diverse domains [25]. Cheng et al. designed a fault attention mechanism and combined this with metrics to learn diagnosis-relevant features [38]. Yu et al. proposed a model by fusing non-stationary signal processing, residual block, and a self-attention mechanism (SAM) [39]. Li et al. proposed a multi-layer DA method to study the feature distribution in each convolutional layer [22]. Zhao et al. published a review paper on unsupervised DTL, where transfer tasks are performed between single-source and single-target domains [5]. In their paper, a benchmark study for several public datasets is provided. Li et al. proposed a multi-receptive field graph convolutional network to first extract features and then used adversarial learning to align source and target domains [40]. These methods can deal with domain discrepancy and obtain reasonable results, but these methods can only be applied between two different working conditions.

Like transferring from a single source to a single target domain, some researchers have also conducted transfer tasks for IFD from multiple source domains. The idea behind involving multiple source domains is to leverage knowledge learned from different but related conditions (i.e., data collected with supervised samples under variable working conditions). Adversarial learning is widely used in multi-source DA by involving a domain discriminator, which is designed to help identify corresponding domain labels. The adversarial learning strategy involves a min-max game, where a generator and a domain discriminator compete with each other to help to recognize both domain-specific and domain-invariant features across different domains. Xu et al. introduced the similarity measurement combined with correlation alignment (CORAL) across multiple domains to minimize the difference between outputs [27]. Zhu et al. proposed a framework for multiple source scenarios by employing a multi-adversarial learning strategy and transfer tasks were performed using different bearing datasets [41]. Wu et al. proposed a knowledge dynamic matching unit-guided network and two classifiers with an attention mechanism [42]. It is also worth noting that, in some cases, the contribution of each domain should also be considered when performing better multi-source DA. By combining data from multiple source domains, it is possible to make full use of known data to train a more robust and generalized model.

3.2.3 Domain generalization

Domain-invariant features imply the sharing of features between different domains (i.e., source and target domains), which is widely used in DG. In DG, adversarial learning is also widely employed to help identify which domain the sample belongs to, which greatly improves feature characterizing learning. However, different from DA, DG means that an adapted model cannot be trained for the target domain since the target domain is totally unseen during training. DG, or out-

of-distribution generalization, learns a generalized model from multiple training domains that generalize well to unseen domains. For DG, data augmentation is usually also used to increase data diversity. By artificially generating new data from existing data using linear superposition, data augmentation can increase the size of the dataset by making small changes to the original data. Li et al. proposed a cross-domain augmentation method to boost the ability to generalize at both the instance and feature levels [33]. Cong et al. studied federated domain generalization under the condition of data privacy protection by designing contributing weights for local model aggregation [43]. Qian et al. used a relationship transfer network to boost the generalization performance of the diagnostic model by measuring and reducing the distribution discrepancy [6]. Fan et al. proposed a deep mixed domain generalization network by applying data augmentation to both class and domain spaces for cross-domain fault diagnosis when the target domain is unseen [44]. These recent studies show that DG is currently a topic of interest in the field of IFD since DG requires limited access to target domain data, making it more challenging, but also providing a more representative approach for applications found in industry.

3.2.4 Motivation

As the name suggests, domain-invariant feature learning is used to learn feature representations that remain invariant across different domains. To train a more generalized model across multiple domains, the presence of shared and invariant features among different working conditions also helps make the transfer task more interpretable. Many researchers have considered the design of domain-adversarial neural networks (DANNs) using adversarial learning by trying to confuse the domain discriminator so that it cannot distinguish which domain the features belong to, thus achieving domain-invariant feature learning through min-max games [45]. However, understanding the domain-invariant features and how to further improve a model's performance for DG is worth further study. Following the alignment scheme often proposed for DA, feature alignment is usually adopted (i.e., most IFD between different working conditions is conducted based on feature transfer). However, this is not enough for domain generalization, which requires samples in the optimization process. Unfortunately, these samples, as well as labels, are not available for unseen target domain data.

Unlike generally defined domain-invariant features, the invariant features mentioned here are further divided into two different groups: (1) z_1 denotes internally domain-invariant features (do not change with other domains and only exist in an individual domain), and (2) z_2 represents mutually domain-invariant features (transferable across different domains). Specifically, internal features exist with the input data and do not change with the existence of other domains, while mutually invariant features are used to understand cross-domain transferable knowledge that is mined from different distributions. Therefore, the integration of these two different features using

feature fusion can ensure better generalization for unseen domains. Most current research only focuses on how to better align different domains when performing transfer tasks (i.e., alignment between source and target domains). However, the invariant features that exist in the individual domains are often neglected. To explore the internally invariant features, the FFT is considered since it is widely used in vibration signal analysis and is also mathematically reversible. That is, the FFT is first used to get the corresponding spectra of the vibration signal, then the extracted features generated by using the original time domain signal are compared to the Fourier spectra to make them similar, so that self-invariant features can be learned. These kinds of domain-invariant features in a specific domain are recorded as internally invariant features. Also, it is worth noting that during the model training procedure, to accelerate computation and save memory, KD is used to avoid calculating the extra Fourier spectra of the testing samples in the unseen target domain [46,47]. To do this, another module with a different feature extractor and an auxiliary classifier is trained. Specifically, the teacher network in the distillation framework uses Fourier spectra and labels as inputs and outputs, respectively, while the student network uses the original time domain signal as an input. To ensure the student network can extract features close to the ones generated through the teacher network, the mean squared error (MSE) is introduced. By doing so, the KD framework can guarantee that features extracted from the original vibration signal are similar to the ones from the Fourier spectra, saving FFT operations and increasing computational efficiency. Then, mutually invariant features can be extracted by leveraging the inter-domain knowledge that exists among different domains [48]. To distinguish the difference between these two kinds of features, their divergence is maximized to ensure that internally invariant features are different from the mutually invariant features. A feature fusion strategy can then be used to combine these two kinds of features to better capture the invariant features across multiple domains. The main contributions are summarized as follows: (1) a DG-based IFD method is proposed, by exploring both internally- and mutually invariant features; (2) the divergence between internally- and mutually invariant features is considered to further ensure the diversity of the extracted invariant features; (3) the FFT is used to learn internally invariant features, which are individually invariant within one specific domain; and (4) KD is introduced to guide the student network in learning features in the frequency domain so that FFT operations can be saved in the inference stage to further increase computational efficiency. It is also worth noting that the proposed method could be developed further by introducing other adversarial learning metrics. Testing results on two public bearing datasets show the superiority and effectiveness of the proposed method.

The rest of this chapter is organized as follows: Section 3.3 introduces IFD using transfer learning, FFT, and KD strategies. Section 3.4 provides the details of the proposed domain-invariant feature exploration (DIFE) method. Experimental results and analyses on the two public bearing

datasets are presented in Section 3.5. Section 3.6 first discusses the robustness of the proposed method against noise and how to further develop and extend the proposed DIFE method using different kinds of inputs. The computation time and sensitivity analysis are given in Sections 3.6.3 and 3.6.4, respectively. Then, potential applications of the proposed method when dealing with limited training samples are discussed in Section 3.6.5. Also, since two different invariant features are explored, ablation experiments are conducted to further verify the effectiveness of the proposed method, as given in Section 3.6.6. In Section 3.6.7, the accuracy performance of the proposed method is compared with some state-of-the-art methods. Conclusions are then given in Section 3.7.

3.3 Proposed approach

3.3.1 Deep transfer learning for intelligent fault diagnosis

The basic idea of IFD is to use collected data to train an end-to-end model to help predict the health state of key rotating components in mechanical systems. Usually, training of the model is guided by loss backpropagation, where the loss is calculated based on given labels.

The main contribution of DTL, when compared to traditional machine learning or deep learning, is to deal with feature distribution discrepancies across different domains, that is $P(\mathcal{X}_S) \neq Q(\mathcal{X}_T)$, where P and Q represent different distributions. The accuracy of the training model will decrease greatly if domain discrepancy exists [49]. So, DTL is implemented to deal with the domain discrepancy caused by variable working conditions of the rotating machinery. The specified domain discrepancy can be minimized by aligning the different domains, which is also known as DA.

For normal DTL, which is widely studied in DA, transfer tasks are performed under the hypothesis that both the source domain data and the corresponding labels are available, recorded as $\mathcal{D}_S = \{x_S, y_S\}$. But, for the target domain, only samples are available, recorded as $\mathcal{D}_T = \{x_T\}$. Corresponding labels y_T are not involved in the training procedure and are only used to test the final accuracy of the training model.

Then, DTL between a single source domain and a single target domain can be developed further by assuming that there are multiple source domains available. This simulates real industrial applications where data is available from multiple sources, and where full use of this data is desired. The same assumption applies to multi-source domain adaptation. That is, there is access to both samples and labels for source domain data, but only samples are available in the target domain data. Therefore, multi-source domain adaptation could be more challenging if the target domain data is totally unseen, which is widely known as DG. Thus, the goal is to train an end-to-end model that still applies to an unseen target domain.

Traditional machine learning methods require that data follow the IID hypothesis. However, there may be domain shifts in unseen target domain data, especially under variable working conditions. Without considering domain-specific knowledge embedded in the unseen target domain, a trained model will degenerate greatly, with a relatively low accuracy, indicating low performance. Therefore, to determine the latent distribution of unseen target domain signals collected under different working conditions, the domain-invariant features that exist across multiple domains should be explored to boost the generalization performance.

3.3.2 Fast Fourier transform

The Fourier transform is used to map the original time domain signal to the frequency domain, expressed as,

$$X(f)=\int_{-\infty}^{+\infty}x(t)e^{-i2\pi ft}\mathrm{d}t \tag{3-1}$$

where $x(t)$ is the original time domain signal, $i=\sqrt{-1}$, and $X(f)$ denotes the amplitude with respect to frequency f. The discrete Fourier transform is used to map the original discrete-time signal to the frequency domain, expressed as,

$$X(k)=\sum_{n=0}^{N-1}x(n)e^{-i2\pi kn/N} \tag{3-2}$$

where $x(n)$ is the original discrete-time signal, N is the number of sampling points, and $X(k)$ denotes the frequency-domain representation at frequency index k.

Moreover, the resulting Fourier spectrum $X(f)$ can be used to recover the original input $x(t)$ by using an inverse FT, written as,

$$x(t)=\frac{1}{2\pi}\int_{-\infty}^{+\infty}X(f)e^{i2\pi ft}\mathrm{d}f \tag{3-3}$$

Using Eqs. (3-1) and (3-3) indicate that the original time domain signal can be recovered, demonstrating that the original input remains invariant after such operations [20].

If a rotating machine operates under constant speed and a failure exists, local peaks can be identified in the corresponding Fourier spectrum, also known as fault characteristic frequencies. Fault characteristic frequencies are determined by the rotational speed of the machine and the bearing specifications. Using signal processing, the actual failure type can be identified by comparing the ratios of the detected frequency peaks. The Fourier spectrum acquired by applying the Fourier transform can provide additional insight when it comes to fault-related feature representation since it is more robust against varying working conditions and noise. For instance,

if a bearing fault exists, applying a variable load does not change the resulting fault characteristic frequency and frequency changing pattern, but may lead to different amplitudes.

3.3.3 Knowledge distillation

KD is a technique that enables knowledge transfer from large models to smaller ones without losing validity [47,50]. It allows for deployment on less powerful hardware, making evaluation faster and more efficient. The key to KD is that knowledge learned from the teacher network (large, computationally expensive models) can be distilled to a student network (a smaller network). The student network usually has a simpler network structure with inference results similar to those of the teacher network. KD requires training the teacher network first, and then training the student network so that its inferences are infinitely close to the teacher network. If p and q represent the inferences by the student and teacher networks, respectively, the objective loss function of KD can be expressed as

$$\mathcal{L}_{KD} = \mathcal{L}\left(y,p\right) + \lambda \mathcal{L}\left(p,q\right) \tag{3-4}$$

where y is the true label, and $\mathcal{L}\left(\cdot\right)$ is the loss function (i.e., cross-entropy loss). The first term denotes the training loss in the student network, and the second term denotes the closeness of the outputs between the student and teacher networks with a trade-off parameter λ.

However, if the outputs of the models are directly used to evaluate the performance of KD, Hinton et al. further propose a SoftMax function with temperature T, written as

$$q_i = \frac{e^{z_i/T}}{\sum_j e^{z_i/T}} \tag{3-5}$$

where z_i is the logit output. After feeding into the subsequent SoftMax function, the probability of the tested sample belonging to each class can be easily acknowledged. A special case occurs when T=1. In this case, Eq. (3-5) will degenerate to a normal SoftMax function. Adding a temperature T can make the network output smoother, which can help the knowledge transfer from the teacher network to the student network.

3.3.4 Domain-invariant feature exploration

In this section, the goal is to develop an algorithm to better characterize the common and shared features associated with different health states of bearings under variable working conditions and enable generalized models to maintain a high performance when testing unseen target domain data. Therefore, to better explore the invariant features among different working conditions and boost the generalization ability of the model, the DIFE method is proposed. Specifically, domain-invariant features are divided into internally invariant features that only exist

in an individual domain and mutually invariant features across multiple domains. By considering the diversity of the extracted domain-invariant features, more information is retained.

To extract internally invariant features that exist within the domain, the FFT is introduced. That is, both the original time domain signals and the corresponding Fourier spectra are utilized during training, whereas most current studies are conducted in either the time domain or the frequency domain exclusively. Moreover, for unseen target domain data, KD is used to further improve computing efficiency so that FFT operations for unseen target domain data can be avoided. It can be understood that both time and frequency domain data can help boost the diversity of the feature representations, which is essential for model training because more invariant features are explored. The FFT is utilized for generating a series of Fourier spectra for the teacher network when distilling knowledge. Once the teacher network is well-trained with the auxiliary classifier, the student network is subsequently trained to learn similar feature representations to those obtained with the teacher network. Therefore, the teacher network can be understood to help the student network realize FFTs through convolutional operations. Since the generated Fourier spectra do not change with other working domains, these kinds of features can be considered as internally invariant. Moreover, a diversity in the learned invariant features can be ensured to help improve the generalization ability of the model.

To learn the mutually invariant feature representations, a domain alignment between different domains can be used to learn cross-domain-invariant features. Popular methods can be divided into two families: metric learning and adversarial learning. In this chapter, the proposed DIFE method is implemented by introducing CORAL between two source domains. By aligning these domains and minimizing their discrepancy, common and shared feature representations can be learned.

Then the combination of both intra- and inter-domain-invariant features can provide a better insight into IFD when the target domain is totally unseen. After internally and mutually invariant features are extracted, feature fusion is then used, where these two features are concatenated together, wherein the first half is internally invariant, and the other half is mutually invariant. For inference, the unseen target domain data are input into the student network branch without using the FFT embedded into the teacher network to increase computational efficiency.

3.4 Methodology

In this section, a flowchart of the proposed DIFE method for IFD is first given. Then, details for internally- and mutually invariant features are introduced, respectively. Finally, these different features are fused to generate the final domain-invariant features for the unseen target domain.

3.4.1 Domain-invariant feature exploration for IFD

A DIFE framework is introduced to explore domain-invariant features for DG. Shared features are learned so that satisfactory performance can be achieved under different working conditions. Currently, there are numerous studies that have extracted shared and common features by simply aligning different domains via metric and adversarial learning. However, simply aligning multiple domains may not be enough to boost the capacity for generalization. As the name suggests, these inter-domain-invariant features exist across multiple domains at the same time and can be recorded as mutually invariant features. The intra-domain-invariant features that exist in each individual domain should also be studied. The combination of these two different kinds of invariant features can ensure the diversity of extracted features and hence boost the capacity for domain generalization. The flowchart for the proposed IFD method is provided in Fig. 3-1.

The entire procedure can be summarized as follows: (1) original vibration data is collected, preprocessing and sliding windows are used to generate a series of samples that act as inputs to the training model, (2) the features automatically extracted from the network are divided into internally intra-domain-invariant and mutually inter-domain-invariant features by using different strategies, (3) feature fusion is used to combine these two different kinds of invariant features, and (4) the learned features are fed into the classifier to help construct and then predict the corresponding health states or specific fault types. It is also worth noting that since the original domain-invariant features are divided into two different groups, internally invariant features z_1 and mutually invariant features z_2, their divergence should be maximized to ensure the diversity of the learned features from the original signals (i.e., using exploration loss, or L2 normalization, to regularize these features).

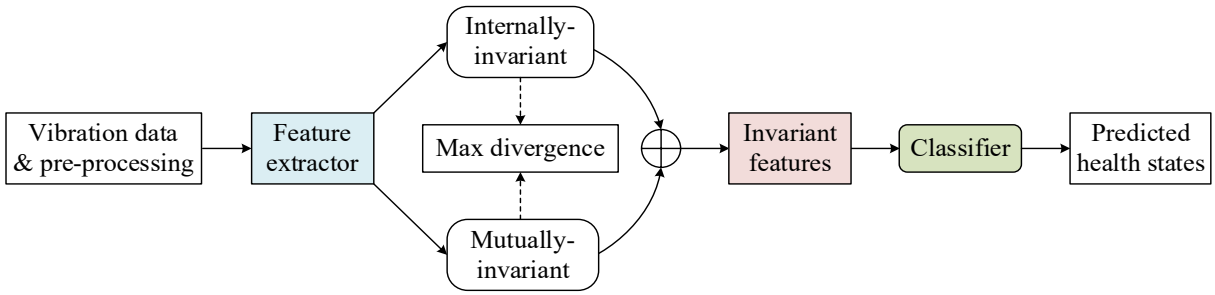


Fig. 3-1. Diverse domain-invariant features in the proposed DIFE method.

To extract internal features that only exist in an individual domain, a KD framework is employed to capture high-level features using the FFT. For mutual features that exist across different domains, CORAL is used to align the feature distributions across any two known domains. To allow different feature exploration, regularization is added to maximize the divergence and ensure the diversity of these two kinds of invariant features.

3.4.2 Knowledge distillation for internally invariant features

Internally invariant features are features directly related to classification that are only embedded in each domain and not affected by other domains. In Fig. 3-2, the KD network is illustrated, where the teacher network utilizes the amplitudes of the Fourier spectra versus frequency and class labels as inputs to obtain Fourier information features for classification.

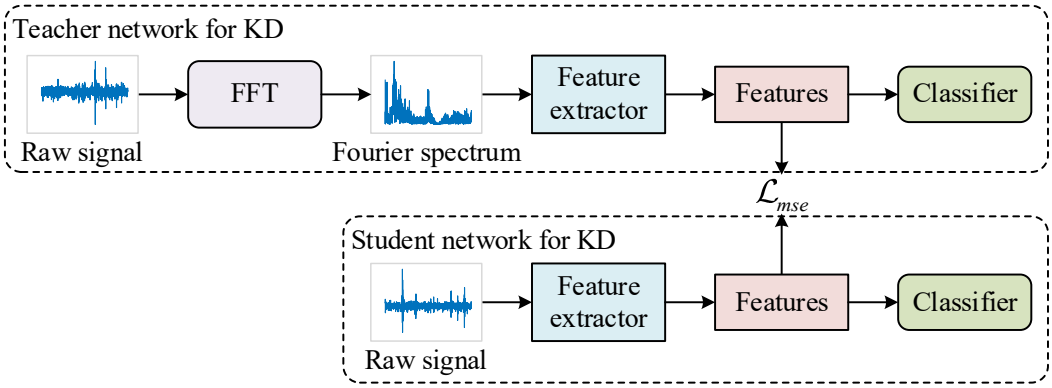


Fig. 3-2. The KD network used to extract internally invariant features.

The Fourier spectrum of the original vibration data can be calculated using the FFT, following Eq. (3-2), which is recorded as $\mathcal{F}(x)$. The teacher network in Fig. 3-2 is trained with $(\mathcal{F}(x), y)$ as

$$\min_{\theta_T^{F_1},\theta_T^{B_1},\theta_T^{AC_1}} \mathbb{E}_{(x,y)\sim\mathbb{P}^{tr}} \mathcal{L}_{cls}\left(G_T^{AC_1}\left(G_T^{B_1}\left(G_T^{F_1}\left(\mathcal{F}(x)\right)\right)\right), y\right) \tag{3-6}$$

where $\theta_T^{F_1}, \theta_T^{B_1}, \theta_T^{AC_1}$ are the learnable parameters of the feature extractors $G_T^{F_1}$, $G_T^{B_1}$, and auxiliary classifier $G_T^{AC_1}$ in the teacher network, respectively. $\mathbb{P}^{tr}$ represents the distribution of the training data, $\mathbb{E}$ denotes expectation, and $\mathcal{L}_{cls}$ is the cross-entropy loss function, which is widely used for classification problems.

Once the teacher network T has been trained with a certain number of epochs, KD is used to guide the student network S to learn Fourier information. The objective function of the distillation process is written as

$$\min_{\theta_S^{F_2},\theta_S^{B_2},\theta_S^{C_2}} \mathbb{E}_{(xx,yy)\sim\mathbb{P}^{tr}} \mathcal{L}_{cls}\left(G_S^{C_2}\left(G_S^{B_2}\left(G_S^{F_2}(x)\right)\right), y\right) + \lambda_1 \mathcal{L}_{mse}\left(z_1, G_T^{B_1}\left(G_T^{F_1}\left(\mathcal{F}(x)\right)\right)\right) \tag{3-7}$$

$$z_1 = G_S^{B_2}\left(G_S^{F_2}(x)\right)_{1,2,\ldots,\lfloor n/2 \rfloor} \tag{3-8}$$

where $\theta_S^{F_2}, \theta_S^{B_2}, \theta_S^{C_2}$ represent the learnable parameters of the feature extractor for the student network. λ_1 is a tradeoff hyperparameter and $\mathcal{L}_{mse}$ is the MSE loss, which makes the features of the student network close to the features of the teacher network. As expressed by Eq. (3-8), the learned internally invariant features z_1 are recorded by $G_S^{B_2}\left(G_S^{F_2}(x)\right)_{1,2,\ldots,\lfloor n/2 \rfloor}$, where $\lfloor\ \rfloor$ denotes a round-

down operation, so that the features extracted in the first half through the bottleneck layer are retained.

3.4.3 CORAL for domain alignment of mutually invariant features

To understand mutually invariant features z_2, cross-domain knowledge embedded in multiple training domains is explored. Given two domains $\mathbf{X}_i$, $\mathbf{X}_j$, the CORAL approach is used, such that the corresponding alignment loss $\mathcal{L}_{align}$ can be given as:

$$\mathcal{L}_{align}=\frac{2}{N(N-1)}\sum_{i\neq j}^{N}\left\|\mathbf{C}_i-\mathbf{C}_j\right\|_F^2,\ \mathbf{C}_i=\frac{1}{n_i-1}\left(\mathbf{X}_i^T\mathbf{X}_i-\frac{1}{n_i}\left(1^{\mathrm{T}}\mathbf{X}_i\right)^{\mathrm{T}}\left(1^{\mathrm{T}}\mathbf{X}_i\right)\right) \tag{3-9}$$

where $\mathbf{C}$ is the covariance matrix and $\|\ \|_F$ denotes the matrix Frobenius norm. From Eq. (3-9), the specific mutually invariant features z_2 in $\mathbf{X}$ can be expressed as

$$z_2=G_S^{B_2}\left(G_S^{F_2}\left(x\right)\right)_{\lfloor n/2\rfloor+1,\lfloor n/2\rfloor+2,\ldots,n} \tag{3-10}$$

Additionally, adversarial learning and other metric learning methods can also be extended to deal with mutually invariant feature exploration. Simultaneously, to reduce the redundancy between the internally invariant features z_1 and mutually invariant features z_2, corresponding features should be different, regularized by maximizing their divergence, expressed as

$$\mathcal{L}_{exp}\left(z_1,z_2\right)=-d\left(z_1,z_2\right)=-\left\|z_1,z_2\right\|_2^2 \tag{3-11}$$

where $d(\ )$ is a distance function (here, the L2 distance is used for simplification). It is worth noting that for different datasets, different measurements can be trialed, which may lead to better results (i.e., the L1 distance).

Finally, the objective function for this network structure is optimized as

$$\min_{\theta_S^{F_2},\theta_S^{B_2},\theta_S^{C_2}}\mathbb{E}_{(x,yy)\sim\mathbb{P}^{tr}}\mathcal{L}_{cls}\left(G_S^{C_2}\left(G_S^{B_2}\left(G_S^{F_2}\left(x\right)\right)\right),y\right)+\lambda_1\mathcal{L}_{mse}\left(z_1,G_T^{F_1}\left(G_T^{B_1}\left(\mathcal{F}\left(x\right)\right)\right)\right)+\lambda_2\mathcal{L}_{align}+\lambda_3\mathcal{L}_{exp}\left(z_1,z_2\right) \tag{3-12}$$

where the first and second terms denote the class loss when training the student network and during the process where the student network learns knowledge from the teacher network, respectively, the third term is the loss between every two known source domains, and the last term ensures that the extracted internally- and mutually invariant features z_1 and z_2 are different. The effectiveness of the proposed objective function is verified by conducting an ablation study.

3.4.4 Overview of the proposed method

The detailed network structure of how the proposed DIFE method works for IFD is given in Fig. 3-3.

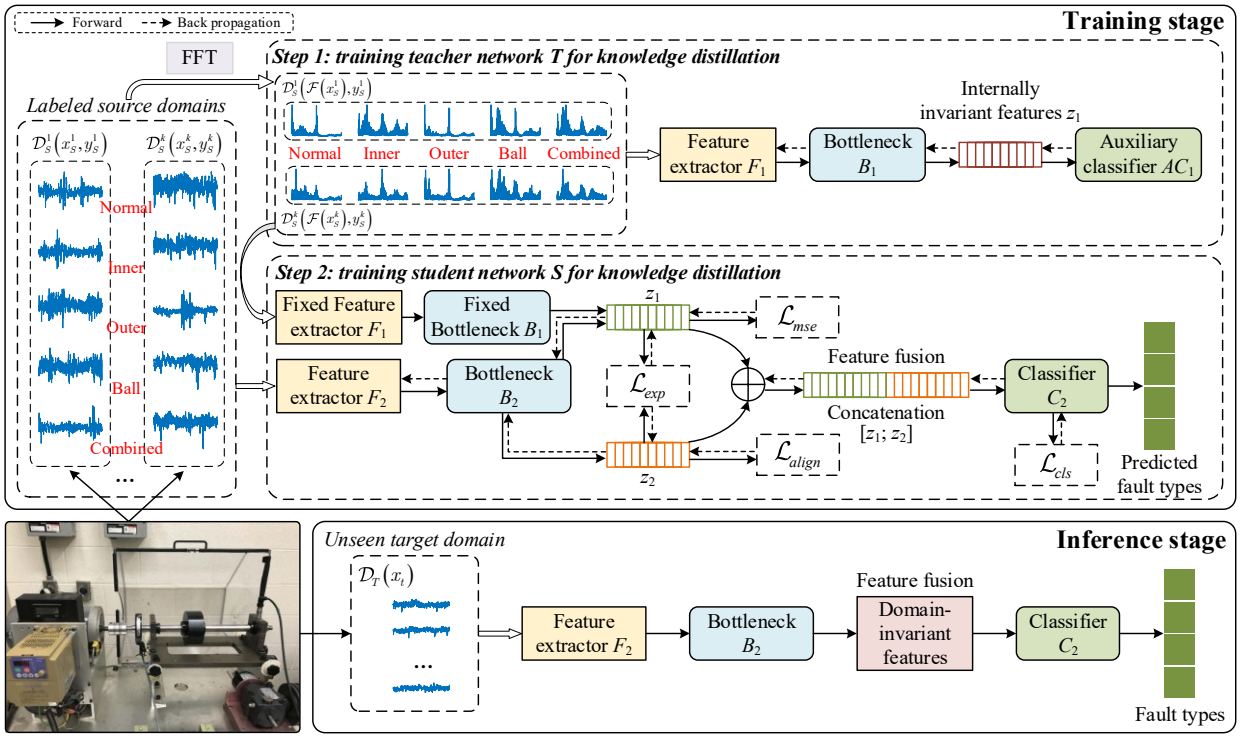


Fig. 3-3. Detailed network of the proposed DIFE method.

First, vibration signals are collected and data preparation is adopted. Then, a teacher network T is trained to help learn the internally invariant features z_1 through feature extractor F_1 and bottleneck layer B_1, guided by an auxiliary classifier AC_1, wherein the loss is calculated based on the known samples from multiple source domains by using a cross-entropy loss function. Then, the student network S with a new feature extractor F_2 is trained, which enables the student network to learn Fourier information. The newly generated domain-invariant features, z_1 and z_2 through bottleneck layer B_2, can be acquired at the same time using feature fusion (feature concatenation is used here), as represented by a series of orange and green rectangles during the training stage in step 2 in Fig. 3-3. Domain-invariant features are divided into two separate parts, wherein the first half is designed to learn intra-domain-invariant features z_1 that exist only in an individual domain. This helps the newly trained feature extractor in the student network to learn intra-domain features without applying the FFT. The MSE is required to minimize the differences between the features learned by the two feature extractors in the KD so that Fourier information can be learned, while the FFT operation for target domain data can be saved. The left half of the fused features represented by green rectangles is designed to learn mutually invariant features z_2 that exist across different domains, which can be realized by simply aligning different source domains (i.e., using CORAL).

The student network S is then guided by a classifier C_2. When testing unseen target domain data during the inference stage, only the student network S is involved. Therefore, FFT operations for internally invariant features z_1 can be reduced to increase computational efficiency. The specific network structure employed in the proposed method is listed in Table 3.1.

Table 3.1. Network structure for the proposed DIFE method.

Blocks	Layers
	Feature extractor F_1, F_2
Block 1	Conv1d(1, 16, 15), BN1d(16), ReLU
Block 2	Conv1d(16, 32, 3), BN1d(32), ReLU, Maxpool(2)
Block 3	Conv1d(32, 64, 3), BN1d(64), ReLU
Block 4	Conv1d(64, 128, 3), BN1d(128), ReLU, AdaptiveMaxpool(4), Flatten
Block 5	Linear(512, 256), ReLU, Dropout
	Bottleneck B_1 for internally invariant feature exploration
Block 6	Linear(256, 128)
	Bottleneck B_2 for both internally- and mutually invariant feature exploration
Block 7	Linear(256, 256)
	Auxiliary classifier AC_1
Block 8	Linear(128, *Nc*)
	Classifier C_2
Block 9	Linear(256, *Nc*)

The training procedure for the proposed DIFE method is summarized in Algorithm 3.1.

Algorithm 3.1: DIFE

Input: Multiple source domain datasets $\mathcal{D}_S = \left\{\mathcal{D}_S^1, \mathcal{D}_S^2, ..., \mathcal{D}_S^k\right\}$.

Initialization: the initialized parameters and other pre-setting hyperparameters.

Training stage:

(step 1: training the teacher network *T* for KD)

1. **for** *epoch* = 1 to *epochs* **do**
2. Randomly select source domain samples and create a training stream $\mathcal{F}\left(\mathcal{D}_S\right) = \left\{\mathcal{F}\left(\mathcal{D}_S^1\right), \mathcal{F}\left(\mathcal{D}_S^2\right), ..., \mathcal{F}\left(\mathcal{D}_S^k\right)\right\}$.
3. Calculate the Fourier spectrum of the original input data using Eq. (3-1).
4. Forward propagation to calculate the class loss following Eq. (3-6).
5. Backward propagation to update $G_T^{F_1}, G_T^{B_1}, G_T^{AC_1}$.
6. **end for**

Return: $G_T^{F_1}, G_T^{B_1}$.

(step 2: training the student network *S* for KD)

7. **for** *epoch* = 1 to *epochs* **do**
8. Randomly select samples from source domains.
9. Forward propagation to extract features by feature extractor F_2 and bottleneck B_2.
10. Learn internally invariant features z_1 by knowledge distilling z_1 following Eqs. (3-7) and (3-8).

11. Learn mutually invariant features z_2 by aligning different domains following Eqs. (3-9) and (3-10).
12. Maximize divergence between z_1 and z_2 to ensure the diversity of invariant features by Eq. (3-11).
13. Feature fusion to generate final domain-invariant features.
14. Backward propagation to update $G_S^{F_2}, G_S^{B_2}, G_S^{C_2}$ by Eq. (3-12).
15. **end for**

Return: $G_S^{F_2}, G_S^{B_2}, G_S^{C_2}$.

Inference stage

Input: Unseen target domain dataset $\mathcal{D}_T$ after training.

Model: DIFE with optimal $G_S^{F_2}, G_S^{B_2}, G_S^{C_2}$.

Output: Predicted labels of the unseen target domain samples.

3.5 Experimental results

3.5.1 Experiment setup

In this section, the proposed DIFE method is verified using two publicly available datasets, the University of Ottawa (UO) bearing dataset and the SQV bearing dataset from Xi'an Jiaotong University (XJTU) [51], [52].

3.5.1.1 UO bearing dataset

In this subsection, the UO bearing dataset under time-varying speeds is described. There are a total of five artificially made fault types: healthy (H), inner race fault (I), outer race fault (O), ball fault (B), and combined fault (C), which includes an inner race, an outer race, and a ball fault at the same time. Also, the time-varying speed conditions can be divided into four groups: (i) increasing speed, (ii) decreasing speed, (iii) increasing then decreasing speed, and (iv) decreasing then increasing speed. The test rig is shown in Fig. 3-4 [51]. The experiments are conducted using the SpectraQuest fault simulator (MFS-PK5M), which holds two ER16K ball bearings. In this case, the bearing on the right side has the fault. The shaft is powered by an AC drive. An accelerometer is mounted to record vibration data, and an encoder (EPC model 775) is used to measure the rotational speed of the motor.

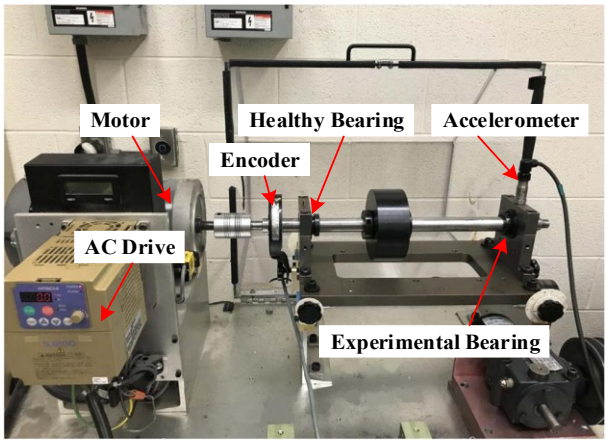


Fig. 3-4. UO bearing dataset test rig [51].

The detailed health states of the test bearings under different working conditions are listed in Table 3.2.

Table 3.2. Domains created from the UO bearing dataset.

Domains	Speeds	Health states	Sample size
A	increasing	Healthy (H)	5×600×4096
B	decreasing	Inner race fault (I)	5×600×4096
C	increasing then decreasing	Outer race fault (O)	5×600×4096
D	decreasing then increasing	Ball fault (B) Combined faults (C)	5×600×4096

To show the time-varying speed conditions, rotation speeds of 4 working conditions are illustrated in Fig. 3-5, where the rotation speeds vary from almost 13 Hz to about 30 Hz. The sampling frequency is set as 200 kHz and 10 seconds of data are collected. In this case study, each sample is designed to have a signal length of 4096 and the overlap between two adjacent samples is set as 1024 (25 % overlap). Then 600 samples are generated for each fault type.

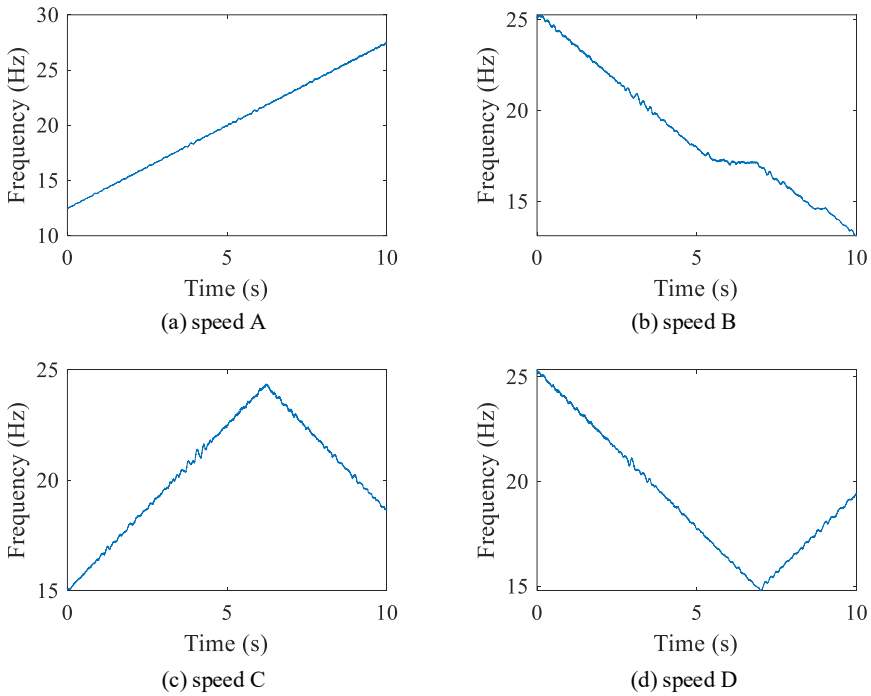


Fig. 3-5. Rotation speed versus time for the UO bearing dataset under 4 different speed conditions: (a) increasing speed, (b) decreasing speed, (c) increasing then decreasing, and (d) decreasing then increasing.

The transfer tasks on the UO dataset are conducted between 4 different speed conditions (domains). For instance, TA means that unseen target domain data is collected under speed condition A and data collected under conditions B, C, and D are involved in model training.

3.5.1.2 SQV bearing dataset

The second public bearing dataset comes from Xi'an Jiaotong University. The test rig is shown in Fig. 3-6 [52]. Experiments are conducted using the SpectraQuest simulator, which has two NSK6203 bearings in the motor. The bearing in the motor closest to the accelerometer is faulty. The fault types include healthy (H), inner race and outer race faults (IF, OF) combined with three different fault sizes, recorded as IF1, IF2, IF3, OF1, OF2, and OF3, respectively.

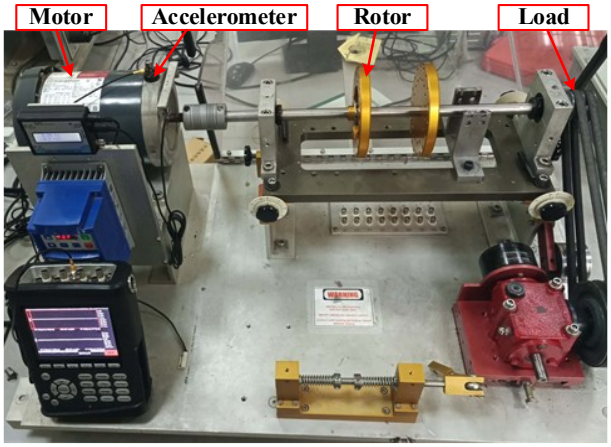
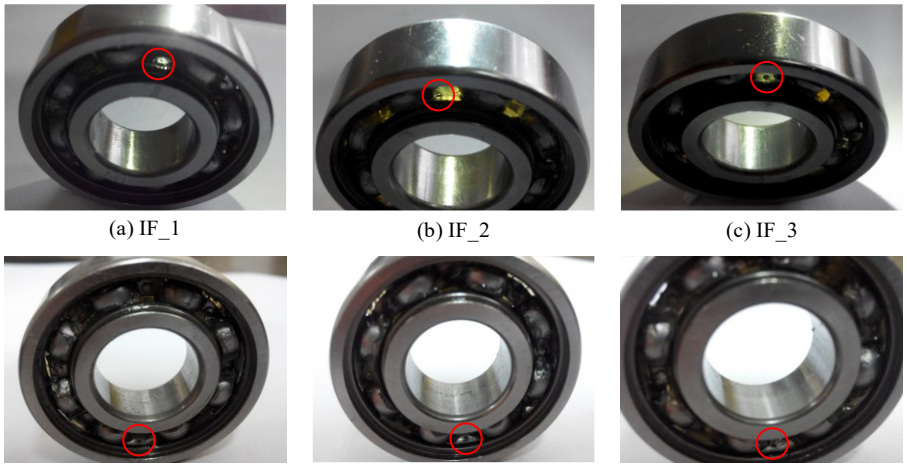


Fig. 3-6. SQV bearing dataset from XJTU [52].

Faulty bearings are pictured in Fig. 3-7, where local faults are marked by red circles. The sampling frequency is set as 25.6 kHz and the motor speed is set to accelerate from 0 to 3000 rpm (equivalent to 50 Hz), then maintain a constant speed for a period of time and finally decelerate to 0 again. 6 experiments are repeated, but with different signal durations.



(a) IF_1 (b) IF_2 (c) IF_3

(d) OF_1 (e) OF_2 (f) OF_3

Fig. 3-7. Fault simulation bearings, where local faults are marked by red circles [52].

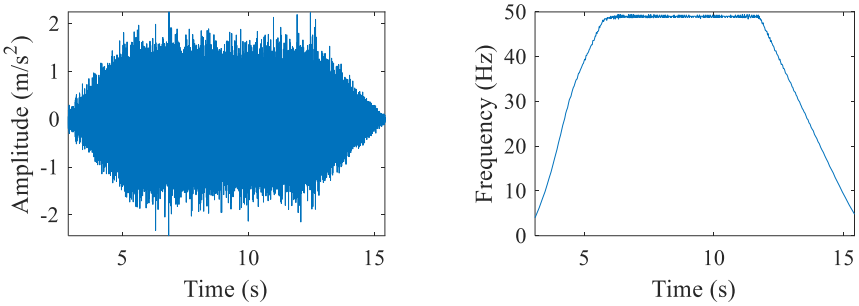
Specific bearing fault types and given labels are listed in Table 3.3.

Table 3.3. Domains created from the SQV bearing dataset.

Domains	Speeds	Health states	Sample number
[1, 2, 3, 4, 5, 6]	First increasing, then stable and finally decreasing with 6 different durations	H, IF1, IF2, IF3, OF1, OF2, OF3	7×100×2048

An example of how the rotation speed changes is plotted in Fig. 3-8, where the first and last several seconds are filtered by measuring the rotation speed. Note that in this case, since the

rotation speed is designed to increase from 0, the collected data may be unstable at the beginning and at the end. Therefore, the signal segment given below is filtered by calculating the rotation speed so that it starts from almost 5 Hz rather than 0 Hz.



(a) signal waveform (b) rotation speed versus time

Fig. 3-8. SQV bearing dataset: (a) original time domain signal, (b) rotation speed versus time.

The transfer task for the XJTU-SQV dataset is also conducted between different speed conditions and the goal of IFD is to accurately predict the corresponding fault types even if the target domain data does not contribute to model training. Focusing more on signal segments with varying speeds, overlap between the truncated signal segments is used to generate more training and testing samples.

In this case study, each sample is designed to have a signal length of 2048 and the overlap between two adjacent samples is also set as 1024 (50 % overlap). Signal segments with time-varying speeds are also preferred so that signal segments that occur during increasing and decreasing speed periods are selected (2 s signal segments are used for both increasing and decreasing speeds).

3.5.2 Methods used for comparison

To better illustrate the improvement of the proposed DIFE method, a set of typical or updated methods is also used for comparison. These include empirical risk minimization (ERM), as well as metric learning and adversarial learning strategies based on ERM [50]. As shown in Table 3.4, all methods are tested using the same network structure (i.e., blocks defined in Table 3.1) and the number of training epochs is set to be the same for a fair comparison.

Table 3.4. Methods used for comparison.

Methods	Description
M1	ERM [50]
M2	ERM with MMD [49]
M3	ERM with CORAL [53]
M4	DANN [45]
M5	Mixup [33]
M6	Proposed DIFE method

In particular, ERM (M1) can be used as a baseline to evaluate whether the proposed transfer task works. Then, to minimize the domain shift between the two domains, M2 is performed by adding the MMD distance loss to align these known source domains.

Compared with M2, M3 could be understood as simply using another metric learning strategy. Also, M3 could be treated as an ablation study of the proposed M6 method by reducing the MSE loss $\mathcal{L}_{mse}$ and exploration loss $\mathcal{L}_{exp}$ without considering internally invariant features. M4 explores domain-invariant features by introducing a domain discriminator and the distance of the learned features is studied at the same time. M5 acts as a data augmentation method to study DG by adding newly generated data to help train the model, where augmented data are linear combinations of the original collected signals from known working conditions.

The hyperparameters used in this study are summarized in Table 3.5.

Table 3.5. Hyperparameter settings.

Hyperparameter	Learning rate	Batch size	Weight decay	Max epoch	λ_1, λ_2, λ_3
Value	0.01	32	0.0005	150	{0.001, 0.5, 0.1}

3.5.3 Results and discussion

3.5.3.1 UO bearing dataset

Transfer tasks performed on the UO bearing dataset are listed in Table 3.6, wherein the best accuracy is in bold, and the second-best accuracy is underlined. It can be seen that the proposed DIFE method (M6) achieves the highest average accuracy of 97.91 %. The second-best method is found to be Mixup, a data augmentation method, achieving an accuracy of 93.59 %. From Table 3.6, it is also worth noting that almost every method fails to provide accurate enough results when testing data under speed condition A. Compared with the other three tasks, task TA could be a little more challenging. Nonetheless, it can be seen that the proposed method M6 can achieve the highest accuracy of 98.13 % on task TA while other methods hover around 80 % accuracy by setting $\{\lambda_1 = 0.001, \lambda_2 = 0.5, \lambda_3 = 0.1\}$. Among all tasks, M6 (proposed method) achieves the two best accuracy results, the two second-best accuracy results, and performs significantly better than other

comparison methods in the UO-TA task, hence leading to the highest accuracy overall, as well as the lowest standard deviation. The comparison indicates the effectiveness of the proposed method.

Table 3.6. UO bearing dataset accuracy results (%).

Methods	TA	TB	TC	TD	Average
M1	81.11±2.72	94.74±0.46	99.91±0.05	94.81±0.50	92.64±8.06
M2	82.54±2.05	90.35±1.84	98.29±0.46	95.26±1.89	91.61±6.87
M3	79.41±0.72	85.98±3.40	97.28±2.05	92.79±2.72	88.87±7.83
M4	81.25±3.45	94.72±0.97	99.88±0.04	95.32±0.80	92.79±8.03
M5	79.89±0.52	**96.16±0.28**	99.91±0.08	**98.41±0.56**	93.59±9.26
M6	**98.13±1.43**	96.12±0.67	**99.98±0.02**	97.41±0.26	**97.91±1.61**

To figure out which samples are classified incorrectly, confusion matrices of all methods used are plotted in Fig. 3-9 for the TA task due to relatively low performance.

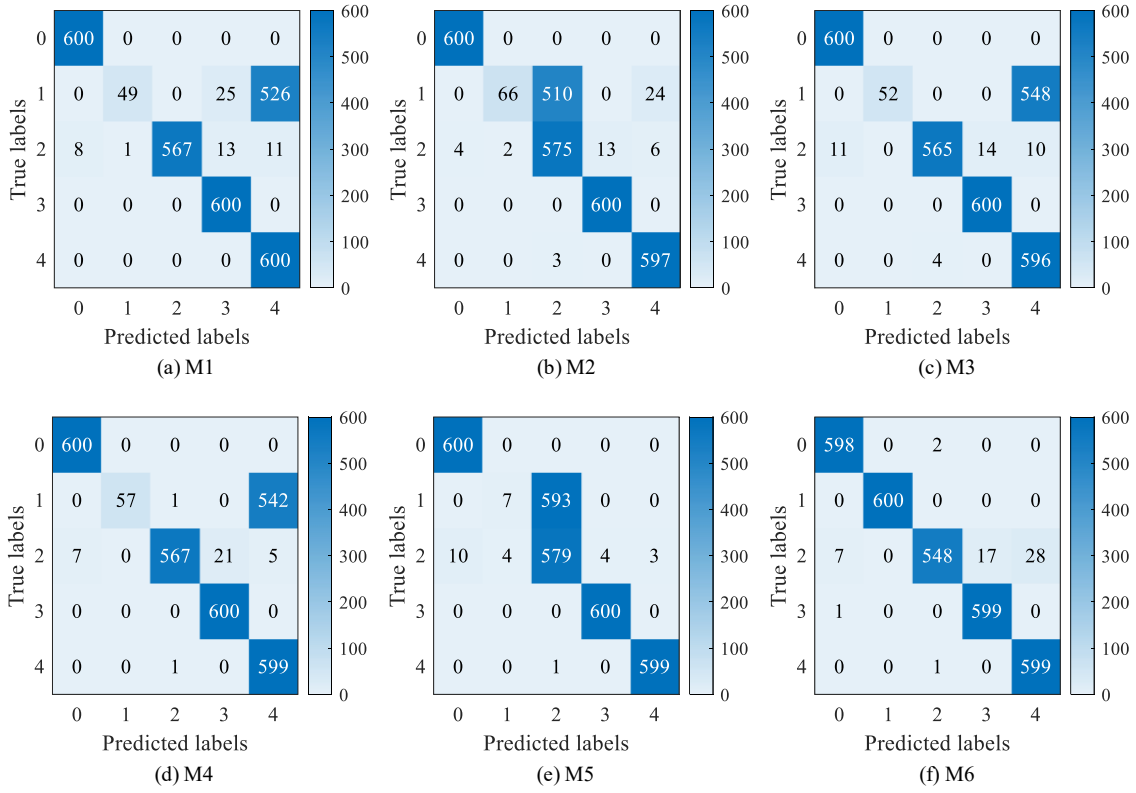


Fig. 3-9. Confusion matrices corresponding to the UO-TA task.

Generally, it can be seen that samples with an inner race fault (label 1, Table 3.2) are more commonly misidentified. By comparing M1, M3, and M4, it can be seen that under unseen target domain A, most samples which should have an inner race fault (label 1) are misclassified as combined faults (where inner race, outer race, and ball faults exist together, label 4), resulting in unsatisfactory accuracy. For M2 and M5, samples having an inner race fault and an outer race fault

are misclassified. In Fig. 3-9 (f), the proposed M6 method successfully predicts all the inner race fault samples, but some samples with an outer race fault are misidentified. A specific case can also be found from the shown feature visualization results, where only the proposed method fails to accurately predict the actual health states of the healthy samples. Two healthy samples are misidentified as an outer race fault. More details are provided in the subsequent feature visualization results of Section 3.5.4.1.

3.5.3.2 SQV bearing dataset

All methods used on the UO dataset are applied again to analyze the SQV bearing dataset, and the results are tabulated in Table 3.7. Here, 6 tasks are conducted because the experiments were performed 6 times. For instance, T1 means that the first experiment was treated as the unseen target domain, and data from the other 5 experiments are used as labeled source domains. The specific accuracies by method are plotted in a radar diagram shown in Fig. 3-10. It can be seen that the proposed M6 method (light blue colored line) outperforms other methods in all tasks, indicating the effectiveness of the proposed method.

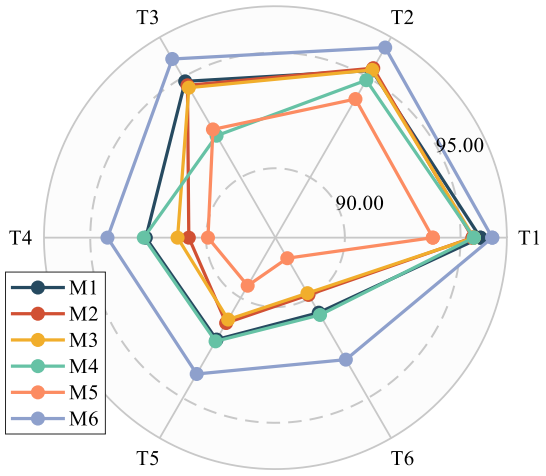


Fig. 3-10. Accuracy of different methods for all unseen target domains.

As listed in Table 3.7, the proposed M6 method achieves the best average accuracy of 95.02 %, as well as the smallest standard deviation error of 1.40 %, indicating that the proposed method is more robust and generalized on the SQV dataset. However, compared with the UO dataset, M4 (DANN) is the second-best method, indicating that adversarial learning could also help boost the generalization ability of the model. Considering the speed changes on the SQV dataset, data augmentation in M5 also fails to help model generalization as more nonlinear features are expected. In this case, the T6 task appears to be the most challenging task. Corresponding confusion matrices

are plotted for further analysis, as shown in Fig. 3-11.

Table 3.7. SQV bearing dataset accuracy results (%).

Methods	T1	T2	T3	T4	T5	T6	Average
M1	95.86±0.39	95.34±0.79	94.80±0.22	92.60±0.70	92.09±1.28	90.74±0.77	93.57±2.05
M2	95.52±0.84	95.46±0.75	94.60±1.12	90.74±0.63	91.26±0.77	89.86±1.33	92.90±2.56
M3	95.54±0.67	95.55±1.32	95.11±0.36	91.40±1.07	91.29±1.47	89.94±1.10	93.14±2.54
M4	95.80±1.28	95.23±1.08	94.20±0.97	92.69±0.34	92.37±0.78	91.54±0.78	93.64±1.70
M5	94.14±0.99	93.92±0.84	92.77±0.91	90.06±0.35	89.31±0.72	88.40±0.40	91.43±2.49
M6	**96.37±0.52**	**96.49±0.08**	**95.91±0.49**	**94.26±0.64**	**93.80±0.46**	**93.31±0.62**	**95.02±1.40**

By comparing the confusion matrices, it can be seen that among all the bearing fault types, IF3 is the most commonly misidentified fault, typically being classified as IF1, IF2 and OF1 instead. The proposed M6 method performs best when predicting health states of IF3, OF2, OF3 and H, achieving 82 %, two 96 % and 97 % results.

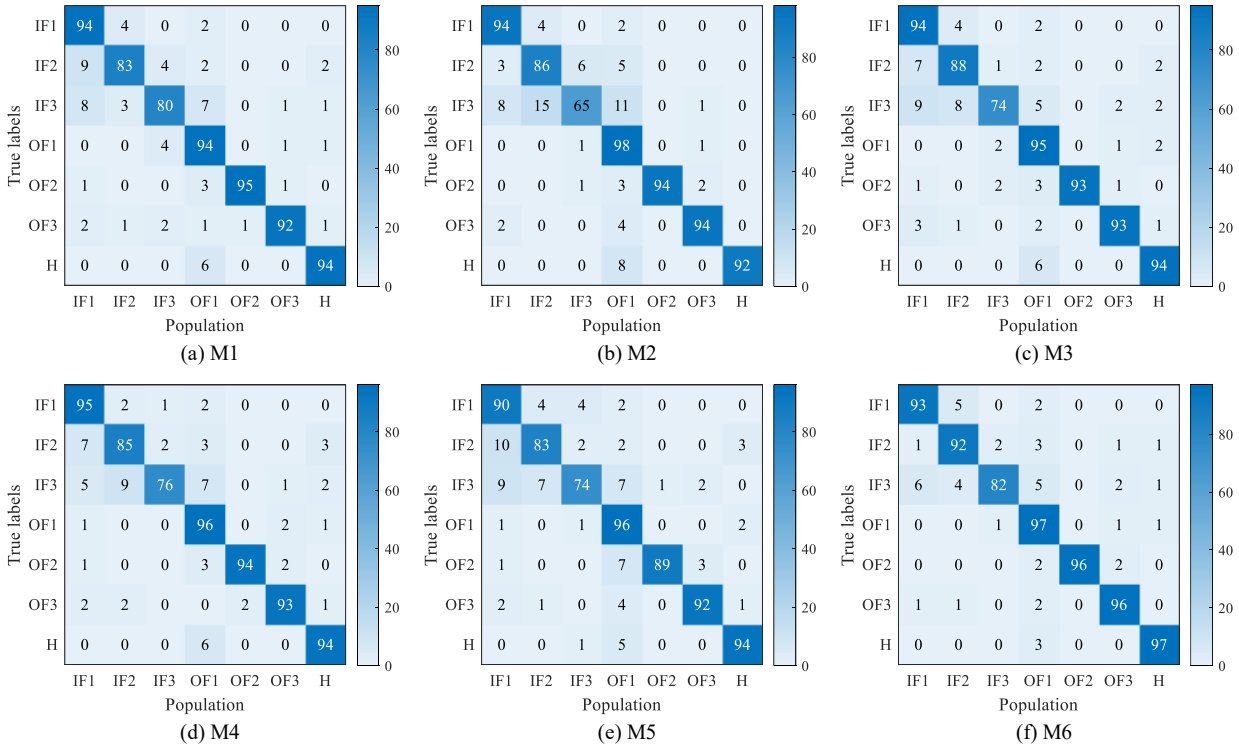


Fig. 3-11. Confusion matrices corresponding to the SQV-T6 task.

3.5.4 Feature visualization

To evaluate clustering performance of the different methods, feature visualization results of the features extracted through bottleneck B_2 using the t-distributed stochastic neighbor embedding (t-SNE) method are analyzed.

3.5.4.1 UO dataset feature visualization results

Invariant features extracted by the proposed DIFE method are visualized. Since original domain-invariant features are divided into internally- and mutually invariant features (z_1 and z_2), these two kinds of features are analyzed, respectively, by reducing the features' dimensions into a two-dimensional feature space, wherein each sample is marked by a pair of coordinates. The distribution of fault-related features extracted from the known and unseen domains is analyzed. For an intuitive understanding, the feature vectors of the UO dataset under task TA are plotted. The feature visualization results are marked differently by 30 pairs of coordinates generated using t-SNE. The DG-based method aims to learn the generalized features clustered together across different domains, including unseen domains. Samples from the same working conditions are plotted using the same color, and different shapes of each sample denote different health states (i.e., fault types). For example, the diamond shape means that the bearing is healthy, while a circle shape denotes a sample with an inner race fault.

Then all methods are employed to see their feature visualization results, where features are extracted through bottleneck B_2 before feeding them into a classifier. The results are given in Fig. 3-12. It can be seen that all 6 methods fail to align the samples having inner race faults (denoted by red circles) because the target domain data is totally unseen during model training. Nonetheless, differences in the misclustered red circles could still be used to evaluate the performance of different methods. For instance, in Fig. 3-12, M1 fails to cluster well on the seen domains, especially the misclassified outer race faults represented by red triangles, which can be found in 4 different clusters. M2, M3, M4 and M5 methods also fail to capture generalized feature representations when predicting inner race faults because of the larger domain discrepancy between seen and unseen domains. Inter-class clustering can be observed more vividly compared to other methods, indicating the decision boundary for each kind of fault type can be easily detected from the two two-dimensional plots, which in turn show the effectiveness of the proposed DIFE method when exploring the diversity of the extracted domain-invariant features.

To further study the effectiveness of the proposed method, feature visualization results based on both internally- and mutually invariant features are analyzed. The corresponding feature distribution results on the UO-TA task are illustrated in Fig. 3-12 (g)-(h). It can be seen that samples with an inner race fault in an unseen target domain (speed A), represented by red circles, are forming an extra cluster away from the cluster center formed by other samples from source domains B, C, and D, demonstrating that domain shifts between the known source domains and the unseen target domain exist.

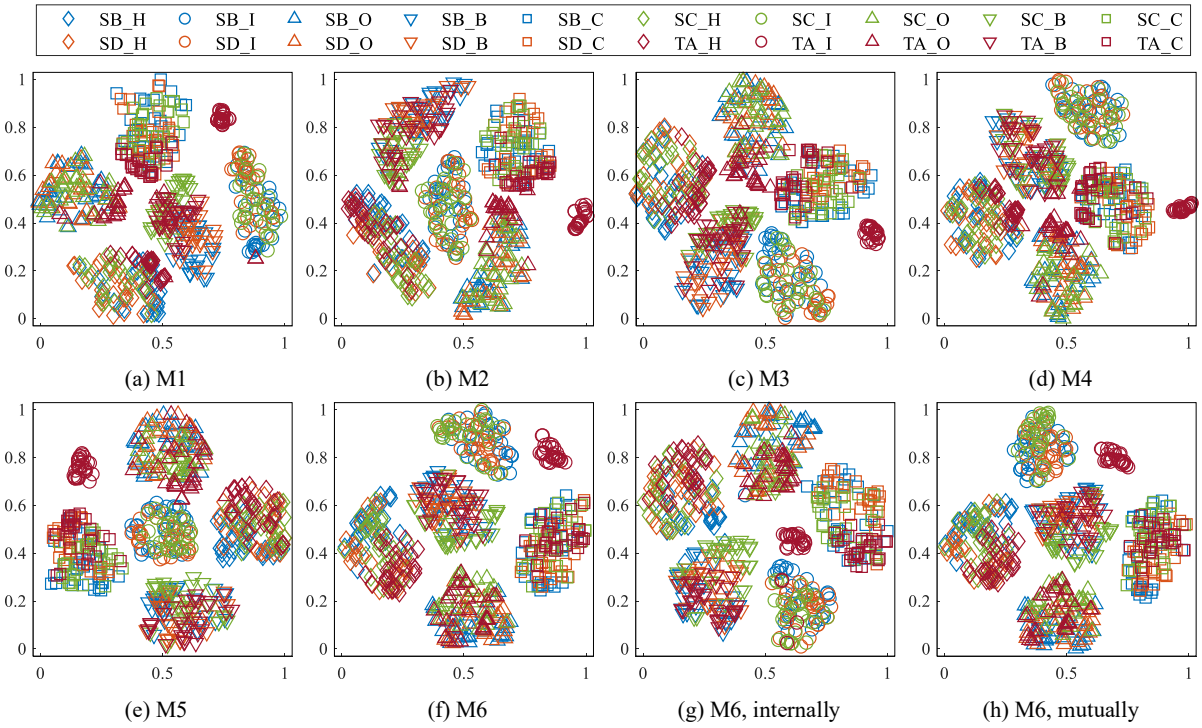


Fig. 3-12. Feature visualization results using M1-M6 on the UO-TA task (normalized axes 0-1 are used for feature visualization purposes).

Also, by comparing Fig. 3-12 (g) and (h), it can be concluded that since the unseen target domain is not involved in the training procedure, a separate cluster may exist to some extent. It can also be found that these red circles (sample with an inner race fault) are more closely located to samples having combined faults (red squares) when analyzing the extracted internally invariant features z_1. For mutually invariant features z_2, the samples in red are, to some extent, closely located to samples having ball faults from the seen source domains. However, by only comparing samples in red, it cannot be concluded that inner race faults may be misidentified as a ball fault or combined faults because red circles are closely located near the known source domain samples, as clear inter-class boundary decisions could still be observed.

3.5.4.2 SQV dataset feature visualization results

For the SQV dataset, the same analysis is conducted. Since the SQV dataset has 7 classes of health states and 6 different domains, another two shapes and colors are introduced but with fewer samples. In this case study, each fault class is presented with 20 data points.

Then, all 6 methods are carried out to visualize the corresponding feature results on the SQV-T6 task, and the resulting feature distributions are plotted in Fig. 3-13. In Fig. 3-13, it can be seen, by focusing on the feature distributions of samples in purple, that M1, M3, and M4 provide good generalization for unseen domain data. However, in the middle of the feature map, the decision

boundaries for seen domains are not clear enough, especially compared to the proposed M6 method in Fig. 3-13 (f). M2 and M5 fail to capture the generalized feature representations for the unseen target domain. The proposed M6 method can provide a relatively clear decision boundary, and the features are well generalized for both the seen and unseen target domains, even though some samples with IF2 and IF3 faults are misidentified. A comparison between the proposed M6 method and the M3 method shows the effectiveness of exploring the internally domain-invariant features by additionally applying the FFT and KD. Similarly, the feature distribution results of the different invariant features on the SQV-T6 task by using the proposed method are plotted in Fig. 3-13 (g) and (h). It can be seen that by using the proposed method, both internally- and mutually invariant features can form clusters with clear decision boundaries, especially in the center of the feature map. However, some samples are misidentified in Fig. 3-13 (h) (i.e., purple circles – sample with a fault type of IF2 are misclassified as IF1).

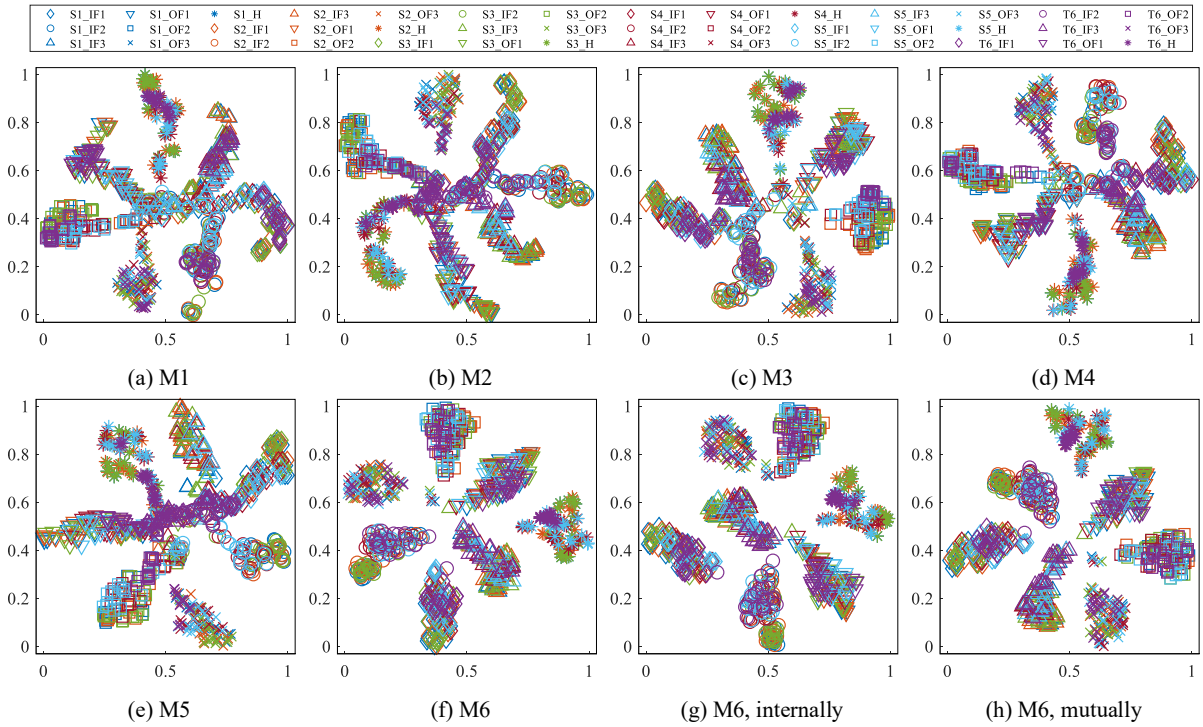


Fig. 3-13. Feature visualization results using M1-M6 on the SQV-T6 task.

3.6 Discussion

3.6.1 Robustness against noise

It is widely acknowledged that vibration signals collected are often contaminated by noise since background noise is almost always unavoidable. However, for industrial applications, experiments conducted in a lab may not be enough, and the extracted internally domain-invariant features z_1 used in the proposed method are highly sensitive to noise. So, the performance of

different methods mentioned in Section 3.5.2 is verified again by adding Gaussian white noise into the vibration signals of the datasets. The SNR used in the artificially generated noisy signal is defined as [54]

$$SNR = 10\log_{10}\left(\frac{P_{signal}}{P_{noise}}\right) \tag{3-13}$$

The SNR is designed to vary from -10 dB to 10 dB. By referring to Eq. (3-13), it can be found that when SNR = 0 dB, the noise energy is equal to the signal energy. The corresponding accuracies using all methods are given in Fig. 3-14. It can be seen that the accuracy of all methods decreases as the SNR decreases. This is because more noise is added to the original signal, leading to a lower performance. However, compared with other methods, the proposed DIFE method still yields a relatively higher accuracy, especially at lower SNR values such as -10 dB and -5 dB. This demonstrates the reliability and robustness of the proposed method against noise and interference.

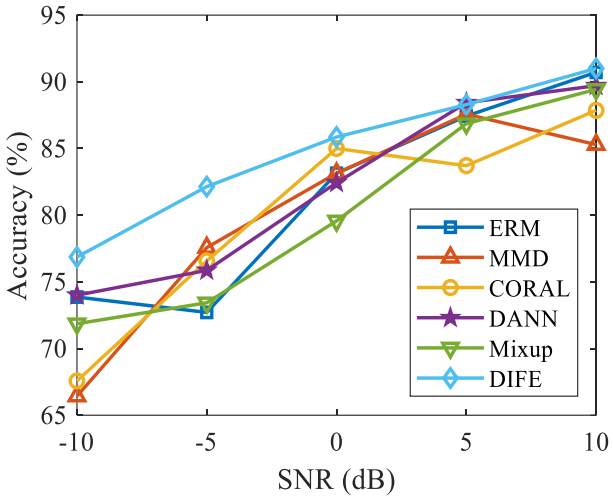


Fig. 3-14. Test accuracy of the SQV-T6 task when artificially adding extra Gaussian noise.

3.6.2 Image inputs based on a pre-trained ResNet18

Following the same structure, the proposed method can be easily developed or extended by using different inputs and feature extractors (i.e., using TFRs presented as RGB figures as inputs and using a pre-trained image classification model as a feature extractor). The figure inputs can be generated by using a time-frequency analysis method, expressed as:

$$S(\tau,\omega) = \int_{-\infty}^{+\infty} x(t)\,g(t-\tau)\,e^{-i\omega(t-\tau)}\mathrm{d}t \tag{3-14}$$

where $x(t)$ is the collected vibration signal, $g(t)$ is the window function used to truncate the signal $x(t)$ (here, a Gaussian window is selected), τ and ω denote time and frequency, respectively. Fig. 3-15 shows figure samples using vibration signals under speed condition *C*.

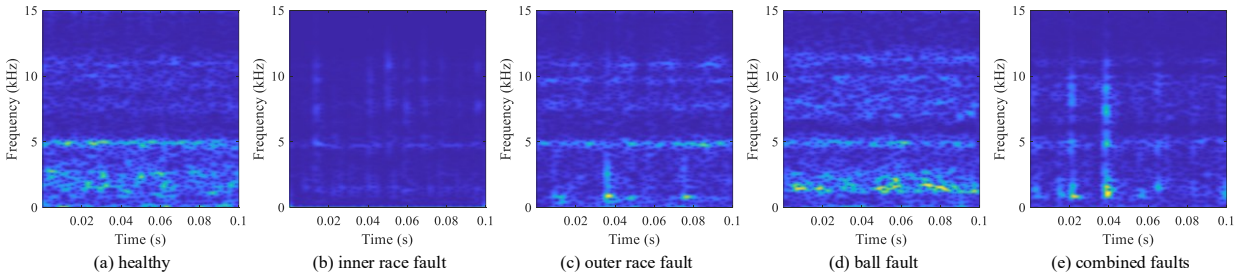


Fig. 3-15. Sample figures from the UO bearing dataset with different health states.

This may be better than randomly initialized parameters for training since this kind of training can be classified as model transfer in DTL. To apply this change, vibration data preprocessing using a time-frequency analysis method is required, and the corresponding feature extractors F_1 and F_2 in Table 3.1 need to be changed. The rest of the network remains the same. By performing an FFT on the figure inputs, the image is transformed from the spatial domain to the frequency domain. It provides a better understanding of the image's features (i.e., edges of fault-related features in the resulting TFRs). An illustration of the FFT taken from the TFR is given in Fig. 3-16.

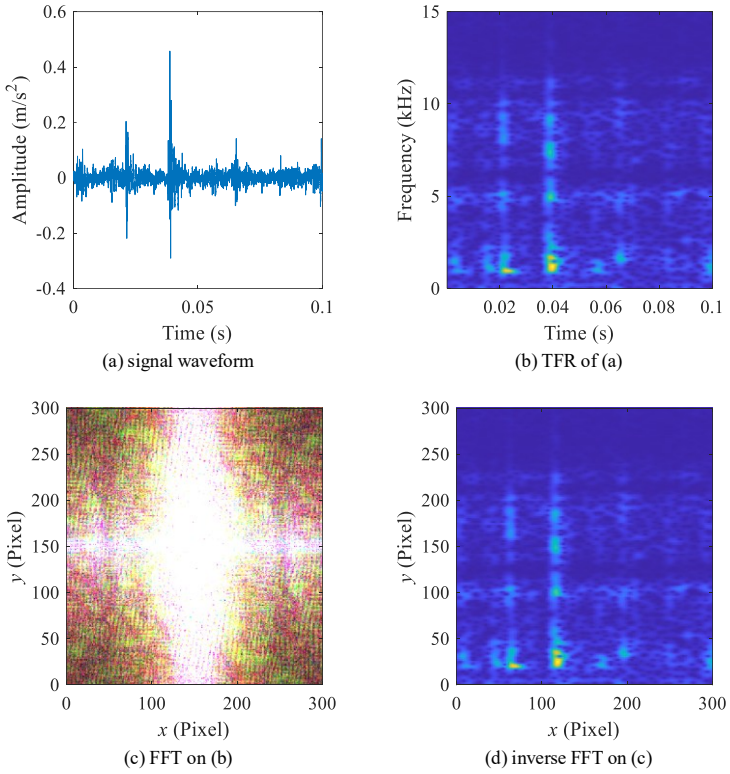


Fig. 3-16. Example of applying FFT on images.

In Fig. 3-16 (a), a signal segment with a duration of 0.1 s is truncated from the original vibration signal, thus 100 images are generated in each class. Then, its corresponding TFR is drawn in Fig. 3-16 (b). Then the FFT is applied, and the amplitude versus spatial and vertical frequency

is given in Fig. 3-16 (c). Finally, the inverse FFT result is illustrated in Fig. 3-16 (d), where the *x*-axis and *y*-axis respectively represent the resolution of the resulting TFR (300 is used here). It can be seen that the resulting TFR can be perfectly restored by performing an inverse FFT, demonstrating that the deformation is individually invariant. This represents the basic idea for why the Fourier transform can be applied to extract intra-domain-invariant features.

The accuracy results of the compared methods are listed in Table 3.8. The highest accuracy is in bold, and the second-best is underlined. It can be found that the proposed method obtains the highest average accuracy of 96.45 %, and the proposed method outperforms other methods in each task.

Table 3.8. Accuracy result on the UO bearing dataset using image inputs (%).

Method	TA	TB	TC	TD	Average
M1	95.2	89.2	96.4	89.0	92.40±3.85
M2	96.0	<u>91.6</u>	95.8	88.4	92.95±3.65
M3	95.2	91.0	95.2	88.4	92.45±3.35
M4	87.6	89.2	<u>97.2</u>	89.2	90.80±4.33
M5	<u>97.2</u>	90.4	96.4	<u>89.6</u>	<u>93.40±3.95</u>
M6	**99.2**	**95.8**	**98.6**	**92.2**	**96.45±3.20**

Feature visualization results using different methods are also given in Fig. 3-17. It can be seen that the proposed method has clearer decision boundaries (as shown in Fig. 3-17 (f) - (h)) for all samples among 5 different classes compared to M1-M5, showing that the proposed method can help boost the generalization capacity for IFD using different kinds of inputs.

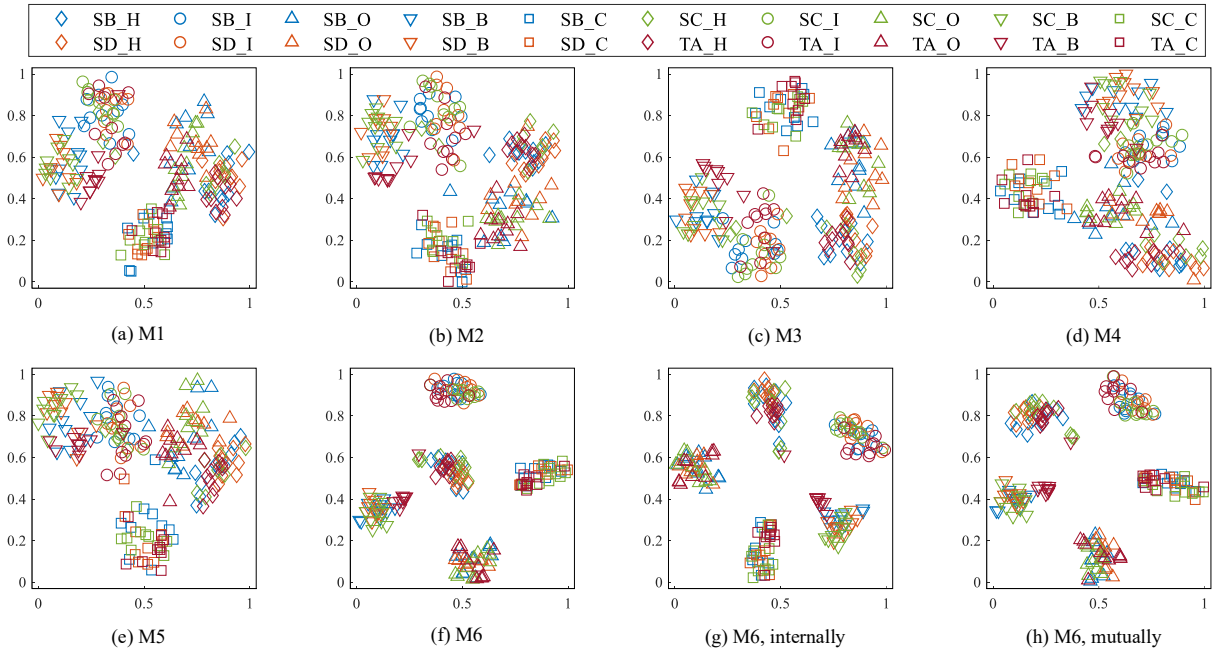


Fig. 3-17. Feature visualization results by using different methods.

The same visualization technique is used again to analyze the UO bearing dataset with figure inputs, wherein a pretrained ResNet18 is used to act as the feature extractor dealing with three-channel image inputs [35].

3.6.3 Computation time

The average computation time over three trials for each method is provided in Table 3.9. Here, the UO-TA task is analyzed by using methods M1 to M6 with 150 epochs (12000 samples are generated, 9000 of them are used during training, while the remaining 3000 samples are unseen during training). All methods are used for comparison. There are two steps in the proposed method (i.e., teacher network and student network). Therefore, these times are also provided separately. The M1 (ERM) method is fast because only cross-entropy loss is used. M2 (MMD) and M3 (CORAL) use metric learning. By comparison, it can be found that CORAL involves matrix multiplication, which is typically more efficient than MMD. M4 (DANN) introduces a domain discriminator to learn the domain-invariant features across different domains through an adversarial learning strategy. M5 uses a mix-up strategy by artificially generating more training samples, which in turn makes it the most time-consuming method. For M6 (proposed method), the teacher network for intra-domain-invariant features is trained first, which takes about 144 seconds. Then the model is saved and loaded to guide intra-domain-invariant feature learning in the student network, taking about 226 seconds. Also, note that only the student network is used during the inference stage.

Table 3.9. Average computation time by using different methods on the UO bearing dataset.

Method	M1	M2	M3	M4	M5	M6	M6-T	M6-S
Time (s)	165.94	269.45	224.77	171.85	378.53	371.52	144.73	226.79

3.6.4 Sensitivity analysis

Three hyperparameters are used in the proposed method. To study their sensitivities, two of them are fixed as 1 (1 is the default coefficient, considering that all terms contribute equally), and the third parameter is varied according to $\{0.001, 0.01, 0.05, 0.1, 0.5, 1\}$.

Here, two tasks—UO-TA and SQV-T1 are borrowed for the sensitivity analysis. The accuracy results versus different parameter combinations are plotted in Fig. 3-18. By taking turns to test the final accuracy on these two tasks, it can be seen that, compared to the SQV-T1 task, accuracy on the UO-TA task is more sensitive to a variation in parameter values, as more fluctuations can be observed for all three hyperparameters. The default accuracy on the UO-TA task when using $\lambda_1 = \lambda_2 = \lambda_3 = 1$ is 86.13 % when the random seed is set to 0. By looking at the changing pattern in the accuracy results, it can be seen that a local peak is obtained when setting $\lambda_2 = 0.5$ and $\lambda_3 = 0.1$. No local peak could be observed when studying the sensitivity of the proposed method

when changing λ_1. However, by setting $\lambda_1 = 0.001$, a higher accuracy can be acquired compared to setting $\lambda_1 = 1$, as seen in Fig. 3-18.

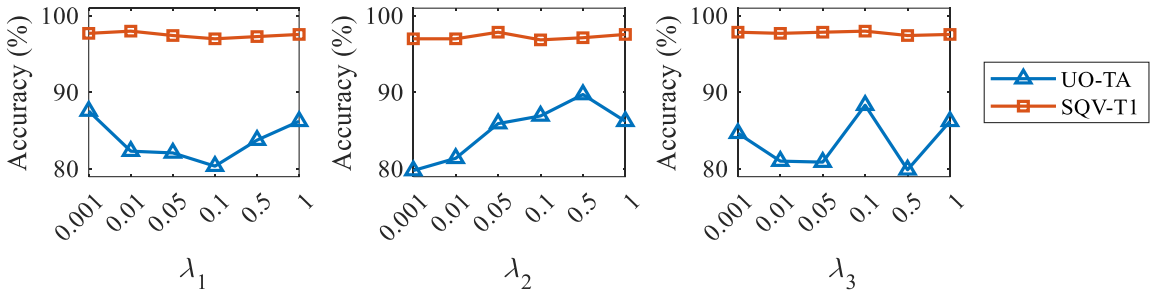


Fig. 3-18. Accuracy versus hyperparameters.

Then, for the UO-TA task, if $\{\lambda_1 = 0.001, \lambda_2 = 0.5, \lambda_3 = 0.1\}$ is used, an accuracy of 98.13 % can be obtained, compared to an accuracy of 86.23 % when using $\{\lambda_1 = 1, \lambda_2 = 1, \lambda_3 = 1\}$. Also, for other tasks on the UO bearing dataset, a smaller value of λ_1 is always preferred. At the same time, for the SQV-T1 task, the effects of hyperparameters are not obvious. Therefore, the default combination could be used directly for this task. To further improve results, a similar analysis could be applied again to search for the optimal trade-off parameters.

3.6.5 Accuracies with limited training samples

To further explore the application of the proposed method, the accuracy of the proposed method when trained with limited source domain samples is tested. Here, the UO-TC task is used since all compared methods perform well on this task. That is, all methods achieve an accuracy of almost 100 % on this task if sufficient data is available, while all the other methods used for comparison only achieve 80 % on the most challenging UO-TA task.

Since there are 3000 samples in each domain, 80 % of them are used for training, and the remaining 20 % are used for validation (2400 for training and 600 for validation). If 1 % of samples are used, it means that only 24 samples in each domain are selected to train the model, while other training samples are not involved in training. The test result is recorded by using the accuracy of 3000 unseen target domain samples. Also, in this case, a random seed equal to 0 is used, which means that the same training samples are used to train the model for all comparison methods. The accuracy results obtained with full access to the training samples (100% are used) are taken from Table 3.6.

A variety of different sample percentages are used, and the accuracy results are listed in Table 3.10. It can be seen that accuracy increases when more samples are involved during training, as expected.

Table 3.10. Accuracies on the UO-TC task with limited training samples and digit inputs.

Method	1 %	2 %	5 %	10 %	20 %	50 %	100 %
M1	70.63	84.60	97.00	97.97	99.17	99.87	99.91
M2	53.93	53.77	81.43	87.50	94.93	98.13	98.29
M3	47.03	48.57	50.87	75.03	83.20	91.67	97.28
M4	74.67	91.37	96.07	98.87	99.73	99.87	99.88
M5	63.03	63.17	92.87	96.70	99.03	99.93	99.91
M6	**77.30**	**95.47**	**98.63**	**99.70**	**99.90**	**99.97**	**99.98**

To more vividly compare the accuracies versus different percentages of training samples, the results are plotted in Fig. 3-19. It is found that the proposed method achieves the highest accuracy under all percentages of training samples, indicating that the proposed method can still outperform other comparison methods even when only a small number of training samples are used.

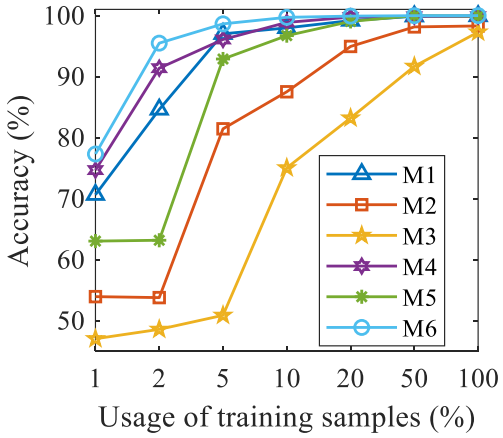


Fig. 3-19. Accuracy results on the UO-TC task with limited training samples.

3.6.6 Ablation study

Ablation experiments are conducted to verify the effectiveness of the proposed method. The compared methods are summarized in Table 3.11, where 4 loss function terms are studied. By removing $\mathcal{L}_{align}$, that is, mutually invariant features are neglected, and only internally invariant features are considered when considering $\mathcal{L}_{mse}$ by minimizing the MSE using knowledge distillation. In the original CORAL, two loss functions are used, the first one is the cross-entropy loss using the predicted labels and the true labels and the second is the distance loss. CORAL can also be considered as a partial solution of the proposed method by only considering inter-domain-invariant features (i.e., by removing $\mathcal{L}_{mse}$ and $\mathcal{L}_{exp}$). This method is recorded as A2, as shown below. Then, $\mathcal{L}_{exp}$ is also neglected for comparison by ignoring the dissimilarities of the learned invariant features. Finally, the full version of the proposed method (M6) is borrowed again for comparison.

Table 3.11. Ablation experiments using partial loss functions.

Methods	$\mathcal{L}_{cls}$	$\mathcal{L}_{mse}$	$\mathcal{L}_{align}$	$\mathcal{L}_{exp}$	Note
A1	✓	✓	✗	✗	internally only
A2	✓	✗	✓	✗	mutually only
A3	✓	✓	✓	✗	w/o feature divergence
M6	✓	✓	✓	✓	proposed method

The accuracy results obtained when using the UO bearing dataset via an ablation study are listed in Table 3.12. It can be seen that the accuracy greatly decreases when neglecting the divergence between the extracted internally invariant and mutually invariant features. Such a phenomenon indicates that these two features are different; that is, the diversity of the invariant features can further boost the generalization capacity of the proposed method. Also, if only one type of invariant features is extracted, the accuracy performance is not good enough.

Table 3.12. Accuracy results of the ablation study on the UO bearing dataset with digit inputs.

Methods	TA	TB	TC	TD	Average
A1	81.27	94.77	<u>99.90</u>	95.10	92.76
A2	79.93	94.93	99.83	<u>96.73</u>	92.86
A3	<u>87.57</u>	<u>95.53</u>	99.87	96.60	<u>94.89</u>
M6	**98.13**	**96.03**	**100.00**	**97.07**	**97.81**

3.6.7 Comparison to results in the literature

The proposed method has been verified to be effective when compared to other methods with the same network structure. The method is further compared to other methods in the literature using two different publicly available datasets. Results are provided in Table 3.13. All methods use digit inputs but differ in sample lengths and network structures.

Li et al. also conducted IFD on the UO dataset using a data augmentation strategy on features at different levels [33]. Results are similar to those obtained herein among four different speed conditions. However, on the most challenging UO-TA task, their highest accuracy was almost 67 %, whereas the accuracy achieved by the proposed method was 98.13 %. Pu et al. proposed a restricted sparse network (RSN) and obtained average accuracies of 95.74 % for the UO dataset and 90.86 % for the SQV dataset [55]. On the other hand, dual branch structures were used by Zhao et al. for testing, obtaining average accuracy results of 93.74 % for the UO dataset and 85.32 % for the SQV dataset [56].

By comparing the proposed method with recent results available in the literature, it can be found that the proposed DIFE method provides favorable accuracy performance when compared with state-of-the-art methods. Specifically, the method demonstrates improved performance when used with both the UO and the SQV datasets.

Table 3.13. Average accuracy result (%) comparison.

Methods	UO dataset	SQV dataset
ADAG [33]	90.11	-
RSN [55]	<u>95.74</u>	<u>90.86</u>
DBANet [56]	93.74	85.32
DIFE (proposed)	**97.91**	**95.02**

3.7 Conclusions and future work

To better study how domain-invariant features work across different working conditions when performing IFD, and a generalized model can be trained when the target domain is unseen, domain-invariant features are divided into internally invariant and mutually invariant features. Internally invariant features z_1 only exist in the individual domain and do not change with other domains, which is realized by applying the FFT. On the other hand, mutually invariant features z_2 can be acquired by using existing algorithms (i.e., CORAL and adversarial learning) to align different domains. Then, since different domain-invariant features are unified, their divergence can be maximized to guarantee their difference. Ablation experiments confirm that the diversity of invariant features should be considered. Two experimental analyses show the effectiveness of the proposed method. Nonetheless, challenging tasks remain to be explored (e.g., reducing the computational cost associated with using three-channel images as inputs since more convolutional operations are needed). The novelty of the proposed method is introduced by the diversity of the domain-invariant features, wherein the model's generalization ability is boosted by considering both internally- and mutually invariant features. For internally invariant features extracted by applying the FFT, it is worth noting that in real industrial applications, collected signals are usually contaminated by heavy noise and interference when compared to experimental setups in the lab. Therefore, further research on the impact of data quality should be carried out. For mutually invariant features, there are many options to achieve domain alignment. Metric learning using MMD, multi-kernel MMD (MK-MMD), and joint MMD (JMMD) could be explored and compared to CORAL and adversarial learning embedded in DANN. Alternatively, hybrid methods that combine metric learning and adversarial learning could also improve generalization and should be considered.

Chapter 4 Latent subdomain assignment based on pseudo domain labels for fault diagnosis of unseen data

This chapter addresses objectives 1, 2, and 3. In this chapter, a new IFD method is proposed by considering the dynamic feature distributions of the collected data. The effectiveness of the proposed method is verified using unseen target data and limited training sample scenarios.

The content of this chapter has been published in *Advanced Engineering Informatics*.

Zehui Hua, Juanjuan Shi, Patrick Dumond, Latent subdomain assignment based on pseudo domain labels for fault diagnosis of unseen data, *Advanced Engineering Informatics*, 67, 2025, 103526.

Authorship contribution statement:

Zehui Hua: writing of the original draft, algorithm development and implementation, writing review and editing, methodology, funding acquisition, conceptualization;

Juanjuan Shi: writing review and editing, validation, funding acquisition;

Patrick Dumond: writing review and editing, supervision, project administration.

4.1 Abstract

IFD is important for rotating machinery maintenance. Unfortunately, fault diagnosis training models often degenerate if unknown domain shifts exist between different working conditions when performing IFD. To deal with this problem, more generalized features related to rolling element bearing faults should be explored so that the generalized capacity of the training model is boosted for unseen target domain data. In this chapter, a new algorithm using pseudo domain labels is proposed to explore subdomain distributions within each subdomain at the domain level. The idea behind the proposed method is that the domain shifts caused by variable working conditions, like varying speeds, should also be considered since the data may show a dynamic distribution of temporal features that are not limited to spatial distributions. That is, the original domain distribution could be further divided into several latent subdomains by introducing pseudo domain labels, which enables the proposed method to learn domain-specific features. Furthermore, the diversity of learned features across subdomains ensures comprehensive feature coverage during model training, while the inherent similarities between these domains enhance the capacity of the model for domain generalization. To figure out how the domain label updates, a domain-class label is initially introduced to facilitate fine-grained feature learning, enabling the model to capture as many features as possible. Then an adversarial learning strategy is employed to separate the domain and class information. Specifically, pseudo domain labels are determined using class invariant features, while class labels are distinguished using features that are invariant across multiple latent subdomains. These two steps are equivalent to a min-max game, like adversarial learning. By exploring features from the class and domain levels, the domain generalization capabilities of the model can be improved, thereby further increasing the accuracy of results. Experiments on two public bearing datasets show that the proposed method outperforms state-of-the-art methods. Additionally, by limiting the number of accessible data from known source domains, the proposed method shows the potential to maintain satisfactory domain generalization capacities when combined with limited training samples.

4.2 Introduction

Recently, IFD has garnered attention from researchers and engineers due to its high accuracy performance and reliability. IFD has the ability to provide real-time fault diagnosis once a model is well-trained and can process a large amount of data at the same time. As is widely acknowledged, IFD based on deep learning models relies on a huge amount of data, which is a typical shortcoming of model training [2]. However, due to the rapid development of Industry 4.0, a large amount of data is now being collected, which has greatly promoted the development of IFD based on deep learning methods. Many network structures can be adopted when performing IFD (e.g., CNNs), recurrent neural networks (RNNs), deep belief networks (DBNs)). ResNet is a typical example in

the family of deep learning models that are enacted by stacking more layers, wherein the shortcut structure enables parameters to be updated through deeper layers [57,58]. However, if the training samples and the testing samples share different feature distributions, violating the prerequisite traditional machine learning assumption, the accuracy will drop significantly [59].

With the development of transfer learning, the limitations imposed by limited data and existing domain distribution discrepancies can be addressed or alleviated to some extent by studying the domain shift between different working conditions. The key step in performing transfer learning is to minimize the intraclass loss and to maximize the interclass loss [60]. By doing so, the samples sharing the same fault types across different working conditions will be aligned together, and the samples with different fault types will be separated with a much clearer decision boundary so that the model can acquire high accuracy performance on the source and target domains at the same time. For instance, Wang et al. proposed a dynamic joint distribution alignment network by studying the marginal and conditional distributions at the same time [61]. Zhao et al. released their framework and comparative study on several public datasets [5]. These methods focused more on transferring from one single domain to another domain by studying domain alignment strategies (e.g., metric learning and adversarial learning). Metric learning utilizes metrics such as MMD to measure domain discrepancies. By mapping the data from both source and target domains to a reproducing kernel Hilbert space (RKHS), the domain distribution discrepancy is minimized so that these two domains share similar distributions, and then the model can obtain high accuracy for both domains [23,48]. Some researchers have also focused on training models with unbalanced data. For example, Shi et al. proposed a graph embedding-based deep board learning system through an encoding and decoding mechanism [62]. Data imbalance is a hot topic in fault diagnosis since most run-to-failure data is healthy during a machine's lifetime, except for failures that are artificially generated. Chen et al utilize two sample enhancement methods to help the model in generating more samples and increasing the diversity of the learned features [63].

However, it can be shown that these domain adaptation methods can only be applied based on the availability of target domain data. Unfortunately, it is more likely that the working conditions of the test data are unseen during the model training phase, especially in particular industrial scenarios, which greatly limits the model's generalization capacity for new data. Therefore, it is necessary to explore how to train the model more efficiently without access to the target domain data when training the model. Nonetheless, there are several approaches for boosting a model's generalizability without access to target domain data. For instance, Li et al. propose a data augmentation method, wherein more data are artificially generated by using a linear combination to facilitate the model's training [37]. Li et al. propose a multi-mode data augmentation method using an auxiliary classifier generative adversarial network (ACGAN) to

improve the quality of generation [64]. These data augmentation methods help boost the generalization capacity of the trained model by simulating feature distributions of the target domain data [65,66].

In recent years, IFD methods based on domain generalization have been widely investigated [67]. The idea of domain generalization is to generalize the knowledge learned from multiple source domains for fault diagnosis on an unseen target task. This requires the model to be able to deal with new data by capturing potential distribution shifts. However, due to the limitation that the target domain data is unavailable, the corresponding distribution shift cannot be calculated. At the same time, most current literature treats the unseen target domain as a single domain to study the potential feature distribution, while neglecting the dynamic distribution caused by time-varying working conditions. For instance, Ren et al. focus on fusing domain-invariant features across two branches at the same time [32].

Furthermore, to make the domain generalization problem more challenging, some researchers limit the number of available training data to see the generalizability of the training model when simulating scenarios where massive data cannot be acquired [31]. Since training data is insufficient, data augmentation and domain augmentation methods are developed to help learn domain-invariant features that make the model robust to out-of-distribution data, thereby generalizing well to unseen domains [68]. Ren et al. limited the accessible source domains, where one of the source domains is completely labeled, while samples from other source domains are available with labeling [32]. Shi et al. propose multisource domain augmentation combined with an adversarial learning strategy to boost the diversity of domain-invariant representations [69]. Gao et al. conducted uncertainty analysis by further studying the confidence of the predicted labels [70]. Also, contrastive learning is widely used by comparing the similarity and dissimilarity between samples. The key to contrastive learning is to make the representations of positive pairs closer, while pushing apart the representations of negative pairs. This is like the triplet loss since they share the same goal of using metric learning to map data points into an embedding space where similar points are closer and dissimilar points are farther apart. Han et al. propose IEDGNet by using triplet loss to facilitate intraclass aggregation and interclass separation at the class level [71]. Song et al. propose a contrast-assisted domain-specificity-removal network to alleviate the side effects related to variable working conditions [68]. By studying both the domain-invariant features and domain-specific features, a comprehensive understanding of the feature distribution in the collected signal can be acquired. Zhang et al. propose a cross-supervised multisource prototypical network by using domain alignment and a pseudo class label [28]. Shao et al. propose universal federated domain adaptation by studying a credible pseudo-label generation mechanism based on a Gaussian mixture model [66].

Where samples are aligned across different working conditions, as most current studies do, the generalization capacity of the model may be limited. Alignment is designed to learn shared features from different working conditions (i.e., domain-invariant features could be specific bearing fault types, while domain-specific features are the time-varying working conditions, such as loads, speeds, and fault sizes). It is widely acknowledged that fault-related features will be in the format of a series of impulses, which are unique signatures when compared to the signal collected under normal states. If the speed remains unchanged, the collected signals will show strong cyclical stationary characteristics, so that fault-related features appear periodically at the same intervals [72]. However, under time-varying speed conditions, amplitudes will also be time-varying. Specifically, amplitudes tend to increase with increasing speeds [37]. Therefore, the feature distributions of vibration signals collected under variable working conditions will show strong feature distribution discrepancies, which can be considered as domain-specific features and should be further analyzed.

To ensure effective domain generalization, following the same idea of transfer learning, the intraclass loss and interclass loss are considered by combining domain and class information as class-invariant features across different subdomains, and should form different clusters. Through the introduction of domain-class labels, samples with different fault types under variable working conditions are treated as a new class. This enables the model to learn fine-grained features from domain and class levels at the same time. To deal with the dynamic distribution that exists across multiple subdomains after introducing pseudo domains, the objective is for the trained model to divide the original domain into multiple subdomains and align these subdomains based on class-invariant features. Here, multiple subdomains refer to the existence of at least two subdomains, whereas a single subdomain can be considered as a special case where no pseudo domain label is introduced. To separate the domain- and class-level information, adversarial learning has been proven to be effective [73]. Specifically, a class discriminator that leverages an adversarial learning strategy will be designed. In this case, the corresponding output will be the class label, which facilitates effective differentiation between classes. Simultaneously, when the class discriminator fails to distinguish which class the sample comes from, the model will focus on learning class-invariant features. Then, samples will be assigned different pseudo domain labels based on their distances to the centroid of each subdomain, which ensures accurate representation of the diverse features at the domain level. It is worth noting that in this step, the diversity of the extracted features across different domains should be ensured. Furthermore, by studying the similarities between the subdomains, domain-invariant features will be learned, which in turn improves the generalizability of the proposed method.

The main contributions of the proposed method can be summarized as follows:

(1) The dynamic feature distribution inherent in variable working conditions is considered, especially when performing bearing fault diagnosis using vibration signals collected under time-varying working conditions. Most existing literature can be considered as a simplified case where no pseudo domain label is introduced. Additionally, improved accuracy performance is achieved when implementing IFD under constant speeds.
(2) Fine-grained features are learned to improve the intraclass aggregation and interclass separation at the class level, which improves the feature characterization capacity of the training model. Newly generated pseudo domain-class labels enable the model to capture richer information from the domain- and class-level at the same time.
(3) Multiple subdomains are determined by assigning samples pseudo domain labels based on measuring their distances to the centroid of each subdomain. Furthermore, adversarial learning strategies, as used in most of the current literature, are further developed to separate domain and class information. Class invariant features are designed to help update the pseudo domain labels when characterizing domain-specific features among multiple subdomains, while domain-invariant features help generalize the model's knowledge for IFD to unseen target domain data.

The rest of this chapter starts with related work in Section 4.3. First, domain generalization-based IFD is introduced. DG-based IFD aims to generalize knowledge from multiple source domains to an unseen target domain, to help improve the model's fault diagnosis performance when testing data is invisible. Preliminary knowledge about adversarial learning and pseudo labeling strategies is also included. A detailed illustration of the proposed pseudo domain assignment method is presented in Section 4.4. Experimental studies, including result analyses and discussions based on two bearing datasets are presented in Section 4.5 to validate the performance of the proposed method. Finally, conclusions are given in Section 4.7.

4.3 Preliminaries

4.3.1 Domain generalization-based intelligent fault diagnosis

IFD here means that a deep learning model is trained by constructing a relationship between the collected vibration signals and their health states. Once the model is well-trained with known data, it can be used to properly predict the corresponding fault types and can potentially be applied to new data as well. However, due to variable working conditions, new data appears to show different feature distributions so that the accuracy of the model decreases. To tackle this problem, the goal then is to train a model that can be applied to multiple working conditions at the same time [74].

Under the assumption that labels for the samples in the target domain are unavailable, the

conditional distribution cannot be calculated because the labels are unavailable. For instance, the conditional distribution can be written as $P\left(y_s|\mathbf{A}^{\mathrm{T}}x_s\right)$ and $P\left(y_t|\mathbf{A}^{\mathrm{T}}x_t\right)$, where *s* denotes the source domain and *t* denotes the target domain, $\mathbf{A}$ is a transform with a transpose operation T, *x* denotes the samples, and *y* represents the labels. Then, a conditional distribution is applied to align $P\left(y_s|\mathbf{A}^{\mathrm{T}}x_s\right)$ and $P\left(y_t|\mathbf{A}^{\mathrm{T}}x_t\right)$ to minimize the discrepancy. This kind of domain alignment strategy is widely used in performing IFD between two different working conditions, which is also known as DA [59].

If multiple source domains are available and the target domain data does not contribute to the model training procedure, the transfer task will be a DG problem, as shown in Fig. 4-1. In this case, the final goal is to train a generalized model that applies to unseen target domain data by generalizing prior knowledge learned from known source domains.

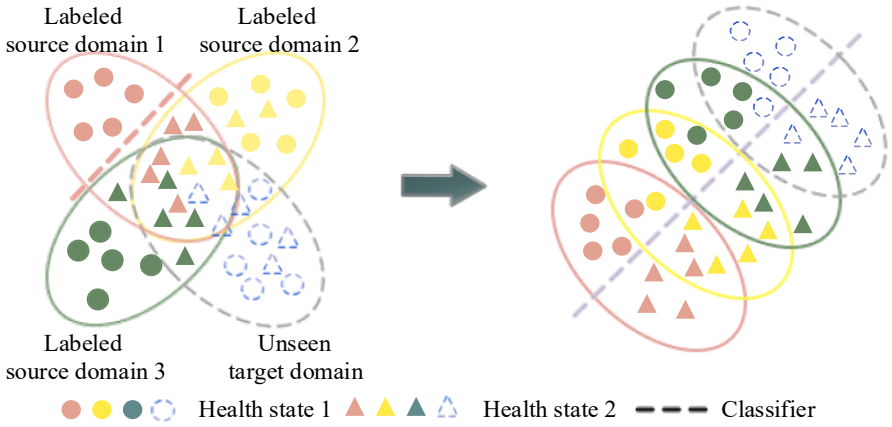


Fig. 4-1. IFD using domain generalization for the unseen target domain.

To make DG-based IFD more challenging, more constraints can be added. Two typical applications are: (1) DG combined with semi-supervised learning (Semi-SL) [68,70,30], and (2) DG combined with FSL [29]. The detailed differences between these methods are summarized in Table 4.1.

Table 4.1. Differences between domain adaptation and domain generalization.

Methods	Source domains	Labeled source data	Unlabeled source data	Target data
Domain generalization	One/Multiple	✓	✗	✗
Domain generalization + Semi-SL	One/Multiple	✓	✓	✗
Domain generalization + Limited training samples	One/Multiple	✓(few)	✗	✗

The goal of domain generalization is to discover domain-invariant features that generalize to unseen domains. Domain generalization combined with SSL focuses on improving generalization

using unlabeled source data, while domain generalization combined with FSL aims to maintain or boost the performance of the model when limited training samples are available. Some researchers also study how to better generalize domain-invariant knowledge from a single source domain [75]. The main contribution of single-domain generalization is that no mutual information can be learned by aligning known source domains.

4.3.2 Adversarial learning

Adversarial learning comes from the concept of generative adversarial networks (GANs) first proposed by Goodfellow [76]. GANs involve training a generator (*G*) and a discriminator (*D*) at the same time. The training of a GAN is a min-max game—the generator aims to generate extra data similar to the given data, to simulate a potential feature distribution while the purpose of the discriminator is to identify whether the sample was generated by the generator or is real. The loss function given an input *x* can be expressed as [76]

$$\min_{G}\max_{D}\mathcal{L}\left(D,G\right)=\mathbb{E}_{x\sim p_{\text{data}}(x)}\log D(x)+\mathbb{E}_{z\sim p_{z}(z)}\log\left(1-D\left(G\left(z\right)\right)\right) \tag{4-1}$$

where *x* represents a real sample drawn from the data distribution $p_{data}(x)$, while *z* denotes a latent variable sampled from a prior distribution $p(z)$, such as a Gaussian or uniform distribution. Inspired by GANs, adversarial learning is developed by defining a domain discriminator in transfer learning tasks. That is, the goal of the domain discriminator is to distinguish whether the sample comes from the source or the target domains. Once the domain discriminator cannot correctly distinguish samples from the source or the target domains, then these two domains can be considered to have similar distributions. So, transfer tasks could be performed with satisfactory accuracy. Different from the discriminator, the generator is usually replaced by a normal feature extractor, which aims to automatically extract features from the original signals. Additionally, the generator can be further used in data augmentation by artificially introducing more data in the model training stage.

4.3.3 Pseudo labeling strategy

When it comes to unsupervised transfer learning, y_t is not available for target domain data. So, the conditional distribution for the samples in the target domain cannot be acquired, to avoid modeling $P\left(y_t|x_t\right)$, an assumption is made following Bayes' theorem [61],

$$P\left(y_t|x_t\right)P\left(x_t\right)=P\left(x_t|y_t\right)P\left(y_t\right) \tag{4-2}$$

To simplify, the marginal distribution $P\left(x_t\right)$ in Eq. (4-2) is neglected. Therefore,

$$P\left(y_t|x_t\right)=P\left(x_t|y_t\right)P\left(y_t\right) \tag{4-3}$$

Then, the conditional distribution $P\left(y_t|x_t\right)$ can be approximated using $P\left(x_t|y_t\right)$ according to sufficient statistics. However, y_t is still unavailable. An alternative method involves training a

simple classifier first with (x_s, y_s) (e.g., a k-nearest neighbor (KNN) [59]).

To solve this problem, a pseudo label embedded in the conditional distribution alignment can be adopted to initialize the training process when performing intraclass transfers. Specifically, pseudo labels are initialized based on majority voting, by predicting the labels using the network trained with source domain data, as well as by obtaining $\tilde{y}_t$, as shown in Fig. 4-2, where $\tilde{y}_t$ is the output of the model with input target domain data. Notably, due to the existing unknown domain discrepancy, it is possible that target domain samples will be accurately predicted but will show lower confidence. Here, confidence refers to the model's certainty in its predictions (i.e., SoftMax probabilities for predicted class labels). Specifically, confidence that is larger than a threshold value of $\tau = 0.8$ can be considered with high certainty, and vice versa [70]. In transfer learning, confidence usually drops when the target domain distribution differs from the source domain because of the existence of unknown domain shifts [30].

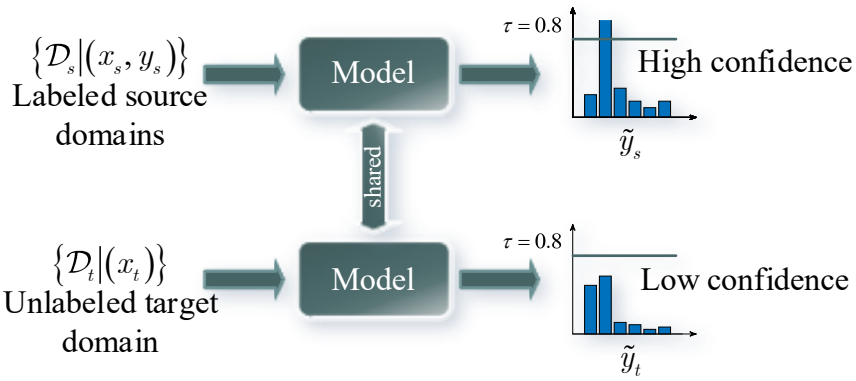


Fig. 4-2. Pseudo-class labeling strategy from labeled source domains to an unlabeled target domain.

Then the intra-class transfer is implemented to minimize the discrepancy when sharing the same labels. The expression for the MMD distance of the intraclass transfer can be given as [59]:

$$D(\mathcal{D}_s, \mathcal{D}_t) = \sum_{c=1}^{C} \left\| \frac{1}{n_1^{(c)}} \sum_{x_i \in \mathcal{D}_s^{(c)}} \phi(x_i) - \frac{1}{n_2^{(c)}} \sum_{x_j \in \mathcal{D}_t^{(c)}} \phi(x_j) \right\|_{\mathrm{H}}^2 \tag{4-4}$$

where $c \in \{1, 2, ..., N_c\}$ are the class labels, $\mathcal{D}_s^{(c)}$ and $\mathcal{D}_t^{(c)}$ indicate the samples in the source and target domains for label c, and $n_1^{(c)} = |\mathcal{D}_s^{(c)}|$ and $n_2^{(c)} = |\mathcal{D}_t^{(c)}|$ denote the number of samples with specific labels in the source and target domains, respectively. $\phi(\cdot)$ is the feature mapping function into the RKHS. By minimizing the distance, the samples with the same labels are aligned. However, the pseudo label in this section refers to pseudo class labels, which are generated based on the assumption that source and target domain data share similarities in their feature distribution.

Pseudo class labels will be unavailable when the target domain data is unseen during training. Inspired by this concept, pseudo domain labels are introduced, which facilitate the learning of feature diversity across different subdomains.

4.4 Pseudo domain label assignment strategy

4.4.1 Motivation

In this section, two experimental vibration signals are considered. The first signal is obtained from the bearing vibration dataset provided by Paderborn University (PU) [77]. Here, signals of a bearing with inner and outer race faults, operating at 900 rpm, 0.7 Nm torque load, and 1 kN radial force, are analyzed. The second experimental signal comes from the University of Ottawa (UO), where a bearing with an inner race fault operated under an increasing speed is selected. In Fig. 4-3, 600 samples are selected from each of the original signals and binned into groups of 100 samples assigned with the same class label.

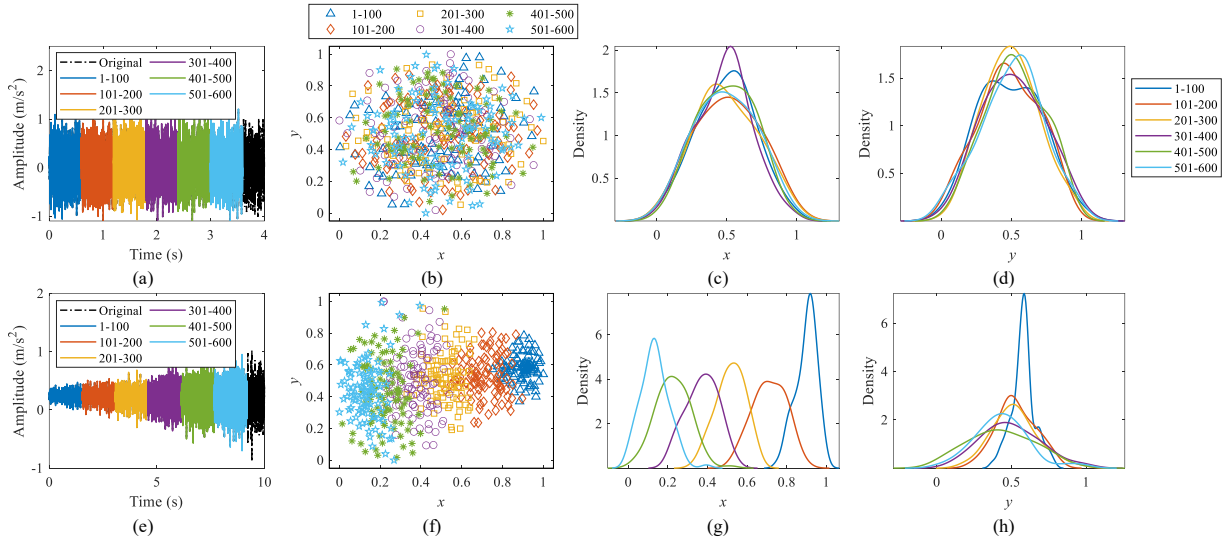


Fig. 4-3. Time domain signal waveform and corresponding feature distribution t-SNE results using PU and UO bearing vibration data for analysis: (a) PU bearing data, (b) feature visualization of (a) by t-SNE, (c) probability of (b) in the *x* direction, (d) probability of (b) in the *y* direction, (e) UO bearing data, (f) feature visualization of (e) by t-SNE, (g) probability of (f) in the *x* direction, and (h) probability of (f) in the *y* direction.

For the PU bearing signal, the signal waveform in the time domain is plotted in Fig. 4-3 (a) and the corresponding feature visualization result using t-SNE is given in Fig. 4-3 (b). Feature distribution results are normalized in the range of [0, 1]. To show differences in feature distributions between sequence samples, their probability density is plotted in Fig. 4-3 (c)-(d). The signal and feature distribution results of the UO dataset are shown in Fig. 4-3 (e)-(h). By comparing Fig. 4-3 (b)-(d) and (f)-(h), it can be found that samples from the PU dataset are randomly distributed in the whole feature space, sharing similar feature distributions versus time, while

dynamic changes versus time can be clearly observed for the UO bearing data due to the varying speed conditions. Notably, amplitude normalization (i.e., scaling the signal amplitude to a fixed range) can increase feature distribution consistency. However, since the proposed method aims to partition the original feature distribution by introducing pseudo domain labels, which of course, requires a higher level of diversified features in the raw signal, normalization is not adopted. Additionally, without strict amplitude normalization, the proposed method does not depend heavily on specific data preprocessing techniques.

While constant speed data is randomly distributed all around the feature space, variable speed samples are regularly distributed. Samples with relatively low speeds (nodes represented in blue, orange and yellow) are mostly located on the right-hand side, and a certain distribution with respect to time can be observed. Thus, dynamic feature distribution of signals collected under single working conditions should be considered to boost the model's capacity for domain generalization.

Fig. 4-4 exhibits the overall schematic of the proposed method by introducing pseudo domain labels, where samples from multiple source domains (denoted by different colors) are divided into two example pseudo domains. Since two subdomains are pre-determined, samples should be further categorized. It is easy to extend the original class labels by introducing pseudo domain labels. For instance, the original 2-category classification (circles and triangles) becomes a 4-category classification problem when 2 pseudo domains are used, as illustrated in Fig. 4-4. To improve the intra-class aggregation and inter-class separation, cross-entropy loss is helpful. However, it is worth noting that the new label contains both domain- and class-level information at the same time. Once supervised learning guided by newly generated samples is conducted, class 1 samples in pseudo domain 1 (in light yellow) become closer, which reflects intra-class aggregation. Similarly, class 1 samples in pseudo domain 2 are also aligned together. However, class 1 samples in pseudo domain 1 and pseudo domain 2 are pushed further apart, which reflects inter-class separation. Generally, the proposed method attempts to obtain generalized features for unseen target domain data. Fine-grained features at the domain- and class-level are learned by initializing a domain-class label using supervised learning to ensure intraclass aggregation and interclass separation (i.e., illustrated by gray arrows in Fig. 4-4).

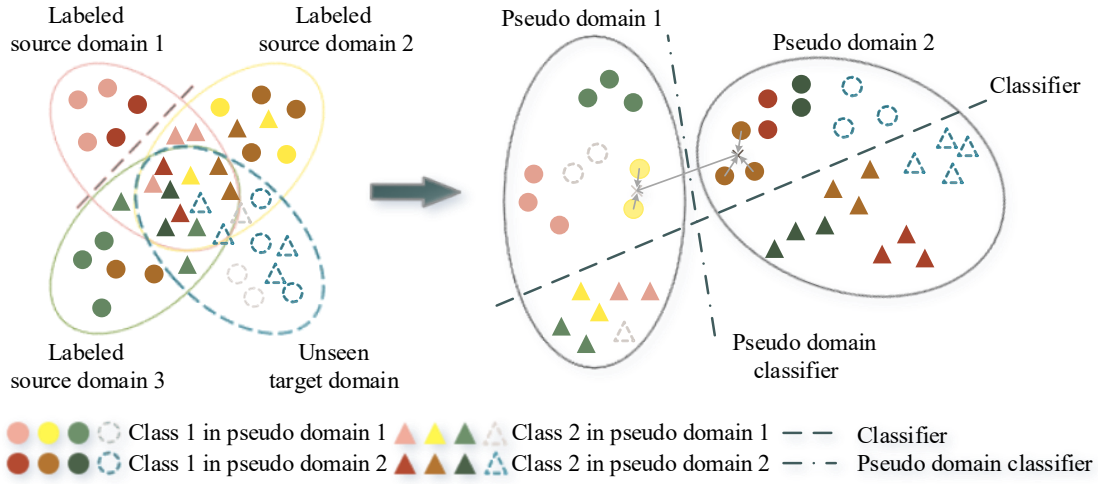


Fig. 4-4. Schematic diagram of the proposed method by introducing pseudo domain labels.

To realize this, a supervised learning approach is developed first to combine domain and class information, recorded as a newly generated pseudo domain-class label y_{pdc}, which can be expressed as:

$$y_{pdc} \in \mathbb{Z}^{+} \bigcap \left[1, N_c N_{pd}\right] \tag{4-5}$$

where y_{pdc} varies from 1 to $N_c N_{pd}$, N_c denotes the number of classes, N_{pd} is the newly introduced pseudo domain label. The training procedure can be conducted under the guidance of backward propagation using a cross-entropy loss function. Then, to separate the domain- and class-level information, another two discriminators should be designed to separate the domain and class labels [78]. To explore the domain-invariant features, a domain discriminator is widely used. When a domain discriminator cannot identify the actual domain labels of the test samples, it can be assumed that the samples share the same distribution so that invariant/shared features across multiple domains can be learned. Notably, to explore domain-specific features, class-invariant features are then required; that is, a class discriminator is then constructed to distinguish the domain labels of each test sample. This dual-adversarial learning plays a min-max game [76]. Class invariant features can be easily labeled since the labels of source domain data are available during model training. Then the class discriminator can guide the learning of domain-specific features. Specifically, domain-specific features are then assigned different pseudo domain labels.

Both discriminators collaborate through adversarial learning, where the pseudo domain discriminator extracts domain-invariant features that can also be used from multiple subdomains for unseen target domain data during inference. Meanwhile, the class discriminator focuses on extracting class invariant features that facilitate the distinction between different subdomains.

4.4.2 Overview of the proposed method

If the target domain data are unavailable, the conditional distributions in Section 4.3.3 cannot be acquired accordingly, thus pseudo-class labeling cannot contribute to the model training procedure. Towards better domain generalization without using target domain data, the original feature distribution collected in a single domain is separated into multiple subdomains by introducing pseudo domains. That is, the samples are assigned with new pseudo domain labels by exploring feature discrepancies. Cross-entropy loss based on pseudo domain-class labels y_{pdc} listed in Eq. (4-5) is calculated to guide the model to learn fine-grained features from the domain- and class-level simultaneously. The relationship between the newly generated y_{pdc} and the pseudo domain label can be expressed as: $y_{pdc} = y + N_c\left(N_{pd} - 1\right)$, where y is the original class label and N_{pd} denote the number of pseudo domains that are introduced when characterizing the dynamic feature distributions. From the above equation, it can be seen that $N_{pd} = 1$ is a special case, where only one pseudo domain is used, and the proposed method degenerates to a normal DANN without considering the class discriminator. The training of this part is carried out based on the newly generated pseudo domain class label to improve intra-class aggregation and inter-class separation, as illustrated by the fine-grained feature learning in Fig. 4-5, where a pseudo domain-class classifier C_{pdcls} is used to correctly predict the updated pseudo domain-class labels.

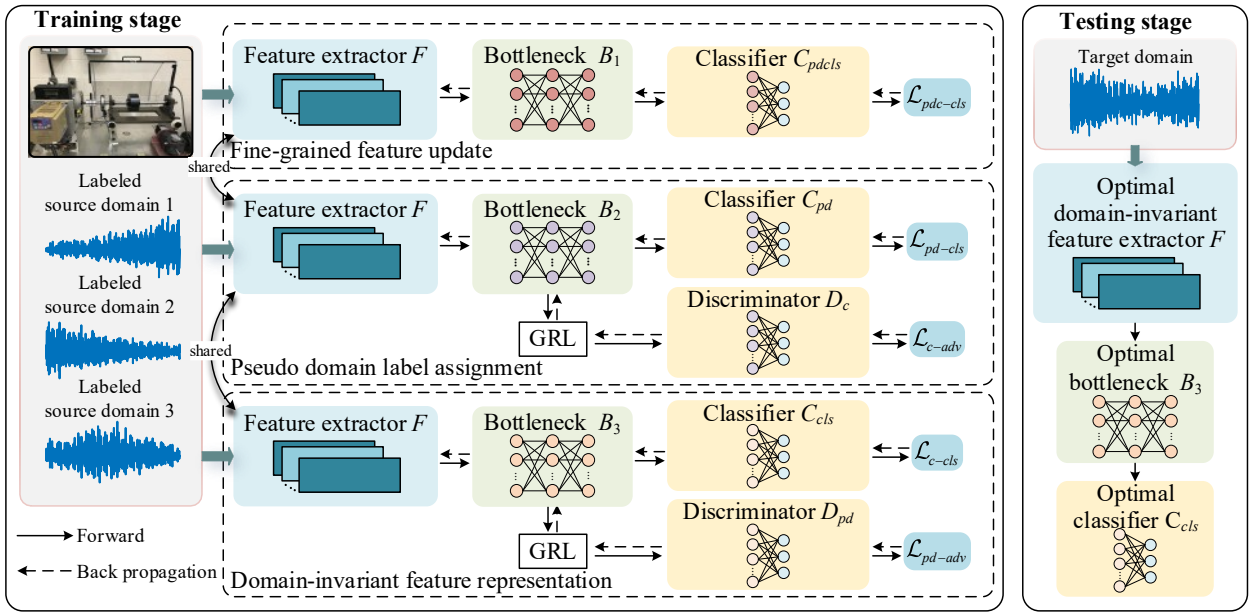


Fig. 4-5. Overview of the proposed method.

By defining F, B_1, and C_{pdcls} as the feature extractor (F), bottleneck (B), and pseudo domain-class classifier (C), respectively, the loss in fine-grained feature learning based on a cross-entropy loss ℓ can be recorded as:

$$\mathcal{L}_{pdc-cls}\left(\theta_F,\theta_{B_1},\theta_{C_{pdcls}}\right)=\min_{\theta_F,\theta_{B_1},\theta_{C_{pdcls}}}\ell\left(C_{pdcls}\left(B_1\left(F\left(\mathrm{x}\right)\right)\right),\mathrm{y}_{pdc}\right) \tag{4-6}$$

where θ_F denotes the parameters of the CNN-based feature extractor F, θ_{B_1} denotes the parameter of the fully connected layer B_1, and θ_{pdcls} denotes the parameters of the pseudo domain-class classifier C_{pdcls}. The used input at this stage is $\left(x,y_{pdc}\right)$ in multiple source domains.

However, introducing pseudo domain labels is a coarse estimation, and all the pseudo domain labels are initialized as 0, which are different from the true labels, indicating that high accuracy cannot always be guaranteed. Thus, iterative operations are preferred to provide a more stable estimation of the corresponding pseudo domain labels. For instance, the pseudo domain labels can be assigned based on the measured distance of all features sharing the same pseudo domain label. Before applying iteration to refine the domain center, a coarse domain center needs to be first determined by mapping all features to the domain space, expressed as:

$$\bar{f}_k=\frac{\sum_{i=1}^{n}\omega_i\cdot B_2\left(F\left(\mathrm{x}_i\right)\right)\cdot\mathbb{I}\left(\tilde{d}_{pd}=k\right)}{\sum_{i=1}^{n}\omega_i\cdot\mathbb{I}\left(\tilde{d}_{pd}=k\right)} \tag{4-7}$$

where $\bar{f}_k$ is the average feature vector for domain k, as the domain center can still be understood as a vector in a higher space, ω_i is the coefficient for the sample x_i, n is the total number of samples, B_2 is the bottleneck layer in the pseudo domain label assignment, $\mathbb{I}(\ )$ is an indicator function that is 1 if $\tilde{d}_{pd}=k$ and 0 otherwise, and $\tilde{d}_{pd}$ is the prediction of the pseudo domain classifier C_{pd} once it is fed with the extracted feature after the bottleneck B_2, which can be formulated by:

$$\tilde{d}_{pd}=C_{pd}\left(B_2\left(F\left(\mathrm{x}\right)\right)\right) \tag{4-8}$$

Then, the predicted pseudo label $\tilde{d}_{pd}$ based on random initialization can be updated by measuring the distance of each sample to the pseudo domain center, formulated as:

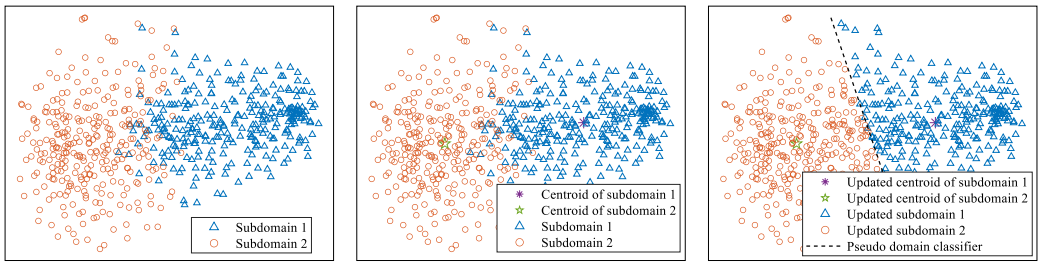
$$d_{pd,i}=\underset{k=1,2,3,\ldots,N_{pd}}{\arg\min}\ \mathcal{D}_{dist}\left(B_2\left(F\left(x_i\right)\right),\bar{f}_k\right) \tag{4-9}$$

where $d_{pd,i}$ is the new pseudo domain label for each sample based on distance, $\mathcal{D}_{dist}$ is the distance function (e.g., $\|\ \|_2$ for Euclidean distance). All samples are assigned to the pseudo domain label corresponding to the pseudo domain center that is closest to them. Then, one more iteration is applied to help update the new centroid and new pseudo domain labels for each sample, as the calculated pseudo domain center is still a coarse estimation based on random initialization of the model parameters. The refined centroid $\bar{f}_k'$ and pseudo domain label $d'_{pd,i}$ can be expressed as:

$$\bar{f}'_k = \frac{\sum_{i=1}^{n} \omega_i \cdot B_2\left(F\left(\mathrm{x}_i\right)\right) \cdot \mathbb{I}\left(d_{pd,i} = k\right)}{\sum_{i=1}^{n} \omega_i \cdot \mathbb{I}\left(d_{pd,i} = k\right)} \tag{4-10}$$

$$d'_{pd,i} = \underset{k=1,2,3,\ldots,N_{pd}}{\arg\min}\ \mathcal{D}_{dist}\left(B_2\left(F\left(\mathrm{x}_i\right)\right), \bar{f}'_k\right) \tag{4-11}$$

This iteration allows more precise estimations, from coarse to fine, and thus, further reduces the overlap of the resulting feature distributions. An example of the pseudo domain label update process is given in Fig. 4-6, where the UO bearing signal is borrowed for analysis.



(a) (b) (c)

Fig. 4-6. Flowchart of assignment and update of sample pseudo subdomain labels based on their distances to the centroid of each subdomain: (a) initial subdomain assignment by introducing pseudo domain labels, (b) centroids of subdomains, and (c) update pseudo domain labels.

Here, the original samples are divided into two subdomains. The centroid of each subdomain is calculated based on Eq. (4-7), as shown in Fig. 4-6 (b). It can be seen that overlap may exist between different subdomains, which results in a decision boundary between different subdomains that is not clear. Hence, one more iteration is required to help update the assigned pseudo domain labels so that a clear decision boundary and no overlap can be achieved (represented by a black dashed line), as illustrated in Fig. 4-6 (c).

To improve performance, adversarial learning also acts as a potential solution, as domain-level information is learned by constructing a class discriminator, as shown in the second stage (pseudo domain label assignment) in Fig. 4-5, where the gradient reversal layer (GRL) indicates adversarial learning is applied. The estimated pseudo domain labels help guide the network to learn domain-variant (domain-specific) features as class invariant features will guide the network to learn more diverse features among multiple pseudo domains. During the pseudo domain label assignment, bottleneck B_2 is used to project the learned features from a shared feature extractor. Since the pseudo domain label is designed to be learned, two parts make the final objective function in this stage. The first part is the pseudo domain classifier C_{pd}, which outputs the prediction of the pseudo domain label for each sample. While the second part is guided by adversarial learning by training a class discriminator to explore the domain-specific features. The

objective function for the pseudo domain label assignment can be recorded as:

$$\begin{aligned}&\mathcal{L}_{pd-cls}\left(\theta_F,\theta_{B_2},\theta_{C_{pd}}\right)+\mathcal{L}_{c-adv}\left(\theta_F,\theta_{B_2},\theta_{D_c}\right)\\&=\ell\left(C_{pd}\left(h_{b_2}\left(h_f\left(\mathrm{x}\right)\right)\right),d'_{pd}\right)+\ell\left(D_c\left(R_{\lambda_1}\left(B_2\left(F\left(\mathrm{x}\right)\right)\right)\right),y\right)\end{aligned} \tag{4-12}$$

where $\mathcal{L}_{pd-cls}$ means the cross-entropy loss using a pseudo domain classifier, $\mathcal{L}_{c-adv}$ denotes the adversarial loss of the class discriminator, R_{λ_1} denotes the GRL with a hyperparameter λ_1, which is widely used in adversarial learning.

Consequently, the pseudo domain discriminator in the domain-invariant feature learning step is trained to learn domain-invariant features among the newly introduced pseudo domains. Once a dynamic balance is achieved between these two branches of adversarial learning, original data is divided into several subdomains by maximizing the distribution discrepancies, while domain-invariant features are learned by reducing the feature distribution divergence among multiple pseudo domains.

The last stage in Fig. 4-5 is used to learn domain-invariant representations across multiple known source domains, which is widely studied in the literature. This stage can also directly output the predicted class labels of test signals since a classifier is directly trained. It also involves adversarial learning by training a pseudo domain discriminator to explore the domain-invariant features. The loss function can be written as:

$$\begin{aligned}&\mathcal{L}_{cls}\left(\theta_F,\theta_{B_3},\theta_{C_{cls}}\right)+\mathcal{L}_{pd-adv}\left(\theta_F,\theta_{B_3},\theta_{D_{pd}}\right)\\&=\ell\left(C_{cls}\left(B_3\left(F\left(x\right)\right)\right),y\right)+\ell\left(D_{pd}\left(R_{\lambda_2}\left(B_3\left(F\left(x\right)\right)\right)\right),d'_{pd}\right)\end{aligned} \tag{4-13}$$

where $\mathcal{L}_{cls}$ outputs the prediction of test samples, $\mathcal{L}_{pd-adv}$ represents the adversarial learning loss by training a pseudo domain discriminator. Once the pseudo domain discriminator cannot distinguish samples from different pseudo domains, it can then be concluded that invariant features across multiple subdomains can be learned. R_{λ_2} denotes the GRL with a hyperparameter λ_2.

Finally, to test the accuracy of the unseen target domain data, the last stage, which focuses on extracting domain-invariant features, is used for inference. Feature extractor F, bottleneck B_3, and classifier C_{cls} are used to predict the actual class labels of the input data.

The specific network structures used in the proposed method are listed in Table 4.2, where N_c and N_{pd} denote the number of classes and the number of pseudo domains, respectively. It can be seen that a basic CNN-based model is selected to automatically extract features. The structure of this model is not complex but efficient. The structure robustness using this CNN-based model, as well as a more advanced ResNet18, is provided in Section 4.6.6 [35].

Table 4.2. Network structures used in the proposed method.

Modules	Details
Module 1	Feature extractor F Conv(1, 16, 15), BN, ReLU Conv(16, 32, 3), BN, ReLU, Maxpool(2) Conv(32, 64, 3), BN, ReLU Conv(64, 128, 3), BN, ReLU, AdaptiveMaxpool(5), Flatten
Module 2	Bottlenecks B_1, B_2, B_3 Linear(640, 256), BN(256), ReLU, Dropout
Module 3	Pseudo domain-class classifier C_{pdcls} in fine-grained feature learning Linear(256, $N_c \times N_{pd}$)
Module 4	Pseudo domain classifier C_{pd} in pseudo domain label assignment Linear(256, N_{pd})
Module 5	Class discriminator D_c in pseudo domain label assignment Linear(256, 256), BN(256), ReLU, Linear(256, 256), ReLU, Linear(256, N_c)
Module 6	Class classifier C_{cls} in domain-invariant feature learning Linear(256, N_c)
Module 7	Pseudo domain discriminator D_{pd} in domain-invariant feature learning Linear(256, 256), BN(256), ReLU, Linear(256, 256), ReLU, Linear(256, N_{pd})

4.4.3 Training procedure

Based on the above-mentioned network architecture, the final optimization objective is:

$$\begin{aligned}
&\mathcal{L}_{\text{total}}\left(\theta_F,\theta_{B_1},\theta_{B_2},\theta_{B_3},\theta_{C_{pdcls}},\theta_{C_{pd}},\theta_{C_{cls}},\theta_{D_c},\theta_{D_{pd}}\right)\\
&=\mathcal{L}_{pdc-cls}\left(\theta_F,\theta_{B_1},\theta_{C_{pdcls}}\right)\\
&\quad+\mathcal{L}_{pd-cls}\left(\theta_F,\theta_{B_2},\theta_{C_{pd}}\right)+\lambda_1\mathcal{L}_{c-adv}\left(\theta_F,\theta_{B_2},\theta_{D_c}\right)\\
&\quad+\mathcal{L}_{c-cls}\left(\theta_F,\theta_{B_3},\theta_{C_{cls}}\right)+\lambda_2\mathcal{L}_{pd-adv}\left(\theta_F,\theta_{B_3},\theta_{D_{pd}}\right)
\end{aligned} \tag{4-14}$$

where λ_1 and λ_2 are coefficients used to balance the adversarial learning losses. The Adam optimizer is utilized to update the network parameters. At each training epoch, neural network parameters are updated as follows:

$$\theta_f^q \leftarrow \theta_f^{q-1}-\mu\left(\frac{\partial\mathcal{L}_{pdc-cls}}{\partial\theta_F^{q-1}}+\frac{\partial\mathcal{L}_{pd-cls}}{\partial\theta_F^{q-1}}+\lambda_1\frac{\partial\mathcal{L}_{c-adv}}{\partial\theta_F^{q-1}}+\frac{\partial\mathcal{L}_{c-cls}}{\partial\theta_F^{q-1}}+\lambda_2\frac{\partial\mathcal{L}_{pd-adv}}{\partial\theta_F^{q-1}}\right) \tag{4-15}$$

$$\theta_{B_1}^q \leftarrow \theta_{B_1}^{q-1}-\mu\cdot\frac{\partial\mathcal{L}_{pdc-cls}}{\partial\theta_{B_1}^{q-1}} \tag{4-16}$$

$$\theta_{B_2}^q \leftarrow \theta_{B_2}^{q-1}-\mu\left(\frac{\partial\mathcal{L}_{pd-cls}}{\partial\theta_{B_2}^{q-1}}+\lambda_1\frac{\partial\mathcal{L}_{c-adv}}{\partial\theta_{B_2}^{q-1}}\right) \tag{4-17}$$

$$\theta_{B_3}^{q} \leftarrow \theta_{B_3}^{q-1} - \mu\left(\frac{\partial \mathcal{L}_{c-cls}}{\partial \theta_{B_3}^{q-1}} + \lambda_2 \frac{\partial \mathcal{L}_{pd-adv}}{\partial \theta_{B_3}^{q-1}}\right) \tag{4-18}$$

$$\theta_{C_{pdcls}}^{q} \leftarrow \theta_{C_{pdcls}}^{q-1} - \mu \cdot \frac{\partial \mathcal{L}_{pdc-cls}}{\partial \theta_{C_{pdcls}}^{q-1}}, \theta_{C_{pd}}^{q} \leftarrow \theta_{C_{pd}}^{q-1} - \mu \cdot \frac{\partial \mathcal{L}_{pd-cls}}{\partial \theta_{C_{pd}}^{q-1}}, \theta_{C_{cls}}^{q} \leftarrow \theta_{C_{cls}}^{q-1} - \mu \cdot \frac{\partial \mathcal{L}_{c-cls}}{\partial \theta_{C_{cls}}^{q-1}} \tag{4-19}$$

$$\theta_{D_c}^{q} \leftarrow \theta_{D_c}^{q-1} - \mu \cdot \lambda_1 \cdot \frac{\partial \mathcal{L}_{c-adv}}{\partial \theta_{D_c}^{q-1}},\ \theta_{D_{pd}}^{q} \leftarrow \theta_{D_{pd}}^{q-1} - \mu \cdot \lambda_2 \cdot \frac{\partial \mathcal{L}_{pd-adv}}{\partial \theta_{D_{pd}}^{q-1}} \tag{4-20}$$

The training procedures are repeated until convergence occurs, or a maximum number of epochs is reached. Since the parameters of the feature extractors F are shared and copied, only the last few independent layers are optimized. Then inference is performed with the modules from the last step using F, B_3, C_{cls} to get the final test accuracy on the unseen target domain.

The training and inference procedure of the proposed method is summarized in **Algorithm 4.1**.

Algorithm 4.1: Proposed method by introducing pseudo domain labels.

Training stage

Model: Feature extractor *F*, bottlenecks B_1, B_2, B_3, pseudo domain-class classifier C_{pdcls}, pseudo domain classifier C_{pd}, class classifier C_{cls}, class discriminator D_c, and pseudo domain discriminator D_{pd}.

Input: Multiple (*m*) source domain datasets $\mathcal{D}_S == \left\{\mathcal{D}_S^1, \mathcal{D}_S^2, ..., \mathcal{D}_S^m\right\}$.

Initialization: The initialized parameters and other pre-setting hyperparameters.

1. **for** *epoch* = 1 to *epochs* **do**
2. Randomly select source domain samples and create a training stream for training.
3. Forward propagation to calculate the supervised loss and backward propagation to update F, B_1, C_{pdcls} following Eq. (4-6).
4. Find the centers of the multiple pseudo domains following Eqs. (4-7)-(4-8).
5. Refine pseudo domain label based on metrics following Eqs. (4-9)-(4-11).
6. Domain-specific feature learning by updating F, B_2, C_{pd}, D_c following Eq. (4-12).
7. Domain-invariant feature learning by updating F, B_3, C_{cls}, D_{pd} following Eq. (4-13).
8. **end for**

Return: F, B_3, C_{cls}.

Inference stage

Input: Unseen target domain dataset $\mathcal{D}_T$.

Model: Modules with optimal F, B_3, C_{cls}.

Output: Predicted labels of the unseen target domain samples in $\mathcal{D}_T$.

4.5 Experimental analyses

This section acts as a validation for pseudo domain label effectiveness. As first proposed in the conditional distribution alignment, the pseudo label strategy combined with iterations appears

to be powerful. However, sometimes, it is possible for some samples to obtain the wrong estimated labels even when the iterative procedure is applied. Therefore, adversarial learning is used to improve network performance when involving feature representation and feature selection.

4.5.1 Experimental setup

The UO bearing and PU bearing datasets collected under variable working conditions are used for analysis [51,77]. However, it must be noted that the testing samples in this section are not available during model training. This provides a unique challenge to the proposed algorithm.

4.5.1.1 UO bearing dataset

The UO bearing dataset provides data that is collected under time-varying rotational speeds. There are a total of five artificially made fault types: healthy (H), inner race fault (I), outer race fault (O), ball fault (B), and combined fault (C), which includes an inner race, an outer race, and a ball fault at the same time. Also, the time-varying speed conditions can be divided into four groups: (i) increasing speed, (ii) decreasing speed, (iii) increasing then decreasing speed, and (iv) decreasing then increasing speed. The test rig is shown in Fig. 4-7 [51]. The experiments are conducted using a SpectraQuest fault simulator (MFS-PK5M), which holds two ER16K ball bearings. In this case, the bearing on the right side has the fault. The shaft is powered by an AC drive. An accelerometer is mounted to record vibration data, and an encoder (EPC model 775) is used to measure the rotational speed of the motor.

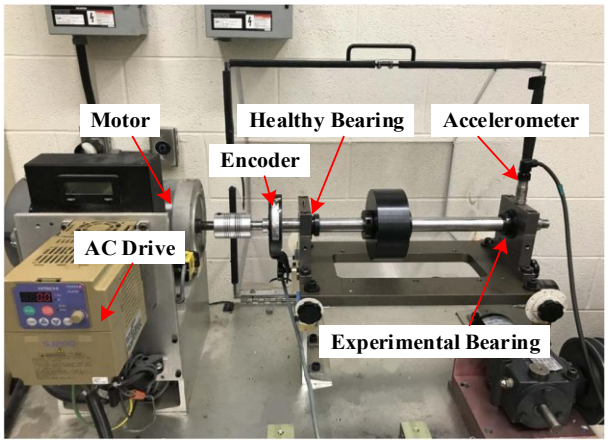


Fig. 4-7. UO bearing dataset test rig [51].

The detailed health states of the test bearings under different working conditions are listed in Table 4.3.

Table 4.3. Domains built from the UO bearing dataset.

Domains	Speeds	Health states	Sample number
A	increasing	Healthy (H)	5×600×4096
B	decreasing	Inner race fault (I)	5×600×4096
C	increasing then decreasing	Outer race fault (O)	5×600×4096
D	decreasing then increasing	Ball fault (B)	5×600×4096
		Combined faults (C)	

Four different domains containing varying speed conditions from the UO bearing dataset are provided in Fig. 4-8 for illustration. It can be clearly found that the speed is continuously time-varying, making this dataset challenging since the domain shift is variable (i.e., the test rig is operated under unseen and variable speeds). In this way, the UO bearing dataset is suitable for the proposed method as the dynamic feature distributions are designed to be learned through pseudo domain label assignment.

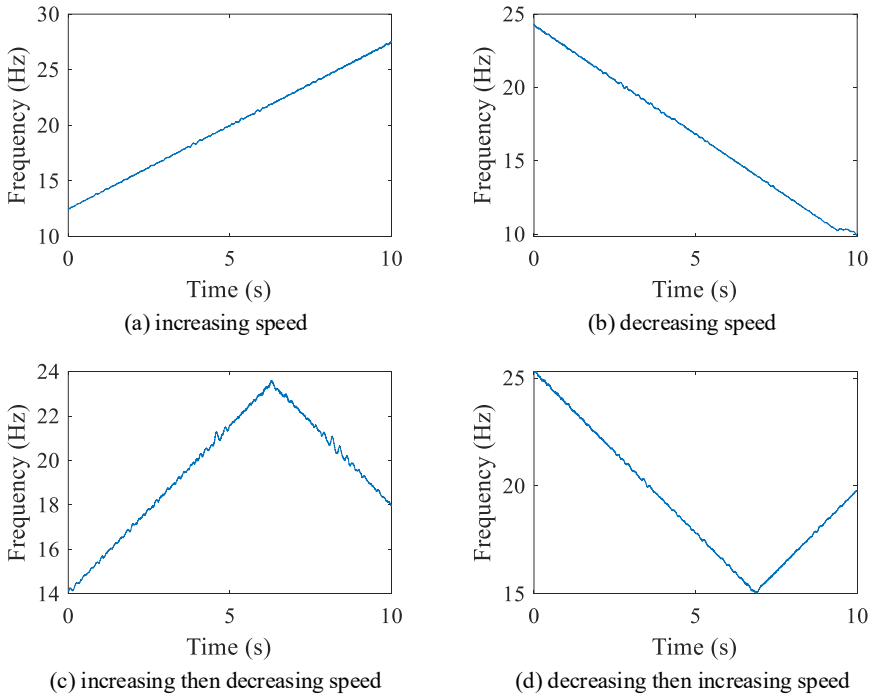


Fig. 4-8. Motor speeds for the UO bearing dataset.

Transfer tasks on the UO dataset are conducted between different speed conditions. For instance, TA means that data collected under conditions B, C, and D are involved in model training, thereby generalizing the knowledge to the unseen target domain under speed condition A. Speeds vary between 10 Hz and 30 Hz. The sampling frequency is set as 200 kHz. Under different speeds, training and testing samples are truncated using a sliding window with a length equal to 4096. The number of samples is balanced across different working conditions. 3000 samples are used for

each domain.

4.5.1.2 PU bearing dataset

The Paderborn experimental test rig is illustrated in Fig. 4-9. The test bearings are mounted in the bearing test module during experiments. The selected bearing's health state can be one of: normal, inner race fault, outer race fault, and combined faults, where inner race and outer race faults are present at the same time.

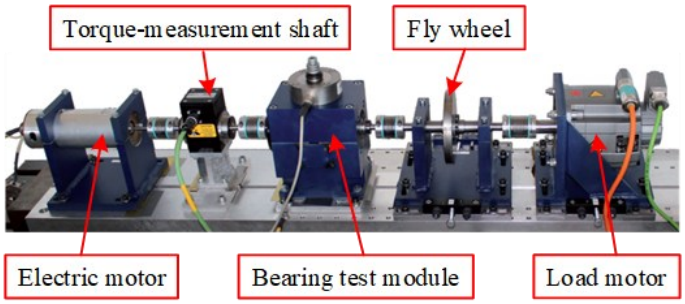


Fig. 4-9. Paderborn experimental test rig.

For the PU dataset, a signal with a length of 2048 is selected as a single sample, and the overlap between samples is set to 50 % to generate more data. Experiments are conducted under 4 different working conditions, as summarized in Table 4.4. The basic operational setup (domain PU0) consists of the test rig running at 1500 rpm with a torque load of M=0.7 Nm and a radial force on the bearing of F=1000 N. Three additional settings are used by reducing the parameters one by one: 900 rpm, M=0.1 Nm and F=400 N (set PU1, PU2, and PU3), respectively.

Table 4.4. Domains built on the PU bearing dataset.

Domains	Rotational speed (rpm)	Load torque (Nm)	Radial force (N)	Health states	Sample number
PU0	1500	0.7	1000	Healthy (H)	4×200×2048
PU1	900	0.7	1000	Inner (I)	4×200×2048
PU2	1500	0.1	1000	Outer (O)	4×200×2048
PU3	1500	0.7	400	Inner & Outer (IO)	4×200×2048

Four transfer tasks—T0, T1, T2, and T3 are conducted between different domains (e.g., T0 means that data collected under condition PU0 is unseen). The number of samples in each working condition is set as 800 (200 for each class). The vibration signals collected are inherently nonstationary, meaning their feature distributions vary even under constant-speed conditions. Therefore, the PU dataset is also relevant for evaluating the proposed method's ability to capture subdomain distributions.

4.5.2 Methods used for comparison

To better illustrate the improvement of the proposed method, a set of typical or updated methods is also used for comparison. The basic method considered (M1) is ERM plus a distance loss using MMD [50]. M2 is a classical method using adversarial learning, which is known as the DANN [73,79]. M3 is a recent domain generalization method using a triplet loss function and a data augmentation strategy [71].

As shown in Table 4.5, all methods are tested using the same network structure (i.e., feature extractor following the modules defined in Table 4.2) and the number of training epochs is set to be the same for a fair comparison.

Table 4.5. Methods used for comparison.

Methods	Description
M1	MMD [50]
M2	DANN [73,79]
M3	IEDGNet [71]
A1	Removal of $\mathcal{L}_{pd-cls}$ in pseudo domain label assignment
A2	Removal of $\mathcal{L}_{c-adv}$ in pseudo label assignment
A3	Proposed method

In particular, MMD (M1) can be used as a baseline to evaluate whether the proposed transfer task works when it involves distance learning. M2 can be considered as a simplified case if setting the number of pseudo domain labels equal to 1. Some other comparison methods are also conducted by removing parts of the final loss functions as an ablation study. A1 removes the pseudo domain classification loss and A2 removes the class discrimination loss when learning subdomain distributions, which are highly related to the proposed pseudo domain label strategy. The full version of the proposed method is recorded as A3. The hyperparameters used for all methods compared in this study are summarized in Table 4.6. The optimal values for the two trade-off parameters are set as $\lambda_1 = 0.1$ and $\lambda_2 = 1$ via grid searching from a range of {0.001, 0.01, 0.1, 1, 10}. In the case study, a relatively small λ_1 is preferred compared with λ_2 as λ_1 is used to guide the model to learn class-invariant features, while λ_2 helps guide the parameter update of the pseudo domain discriminator when learning domain-invariant features. The learning rate $\mu = 0.001$ is set by trial and error.

Table 4.6. Hyperparameter settings.

Hyperparameter	Value	Description
λ_1	0.1	Trade-off parameter for class discriminator
λ_2	1	Trade-off parameter for pseudo domain discriminator
μ	0.001	Learning rate
Batch size	32	Training samples per iteration
Max epoch	120	Total training iterations
Weight decay	0.005	L2 penalty to prevent overfitting

4.5.3 Accuracy results

4.5.3.1 Accuracy on the UO bearing dataset

Three independent trials are conducted for each method. The average accuracy results, as well as standard errors, are recorded in Table 4.7. Methods based on ERM and MMD involving metric learning are insufficient for cross-domain diagnosis with unseen working conditions in the target domain, as target domain data is not available. Thus, domain adaptation is only conducted between three known source domains. By including an adversarial learning strategy, M2 acts as a baseline and also a special case without introducing pseudo domain labels. The average accuracy of M2 is 92.82 %, which shows a little improvement for the unseen domain without the target data. Then for the domain generalization-based method, M3 (IEDGNet) obtains an unusually small accuracy in all tasks compared with M1 and M3. Notably, sometimes M3 is sensitive to the preset trade-off parameters, which are later discussed in Section 4.6.2. As shown in Table 4.7, A1 and A3 achieve remarkable improvement, especially for task TA since the other comparative methods all fail with an accuracy of almost 80 %. It can be seen that the average accuracy using the full version of the proposed method (A3) is highest, achieving 97.84 %, indicating that the health states of unseen target data can be accurately predicted by generalizing knowledge from the known source domains.

Table 4.7. UO bearing dataset accuracy results (%).

Methods	TA	TB	TC	TD	Average
M1	81.80±1.47	91.07±0.40	98.62±0.05	96.30±0.75	91.95±7.47
M2	81.88±1.05	94.13±0.47	99.93±0.04	95.35±0.37	92.82±7.71
M3	74.95±1.12	85.45±4.87	86.52±3.01	90.43±4.96	84.34±6.62
A1	90.24±9.75	98.25±0.85	99.08±1.18	97.84±0.93	96.36±4.11
A2	83.66±5.83	96.94±1.98	99.88±0.10	98.12±1.18	94.65±7.43
A3	**92.81±6.26**	**99.00±0.70**	**100±0**	**99.53±0.17**	**97.84±3.37**

This demonstrates that the proposed method generalizes well on unseen data. In this case study, the number of pseudo domains is set as $N_{pd} = 2$, that is, two pseudo domain labels are pre-determined when training the model. The trade-off parameters used here in the proposed method are both set to 0.1. The trade-off parameters used for IEDGNet (M3) are both set to 0.01 by

referring to [71].

4.5.3.2 Accuracy on the PU bearing dataset

Testing results on the PU bearing dataset are listed in Table 4.8. It can be seen that M1 only achieves 82.94 % accuracy by including metric learning. Specifically, M1 obtains an unusually small accuracy on the T0 task due to large and unseen domain shifts as speed, load torque, and radial forces are all different between the known source domains, which in turn makes the T0 task challenging. For task T1, three known source domains share the same rotation speed of 1500 rpm, while source domains in task T2 share the same load torque of 0.7 Nm, and source domains in task T3 share the same radial force of 1000 N. By comparing the accuracy results on tasks T0-T3 by all methods compared, it can be concluded that T0 is most challenging. M2 obtains an accuracy of almost 90 %, which is much better than M1. By considering the triplet loss, M3 can further improve the diagnosis accuracy to 92.33 %. Among all comparing methods, the proposed method (A3) achieves the highest accuracy on each task, with one of them being 97% and three being 100%, and the highest average accuracy at 99.30%, which indicates that unseen target data can be accurately predicted using the proposed method. A partial solution of the proposed method, by removing the pseudo domain classifier loss (A1), gets the second-highest accuracy of 95.29 %.

Table 4.8. PU bearing dataset accuracy results (%).

Methods	T0	T1	T2	T3	Average
M1	56.00±2.29	92.50±2.07	91.17±0.69	89.75±2.01	82.94±10.5
M2	70.29±1.14	99.88±0.13	88.62±1.80	99.62±0.13	89.60±13.90
M3	75.71±10.09	**100±0**	93.67±0.64	99.96±0.07	92.33±11.47
A1	81.79±2.96	99.92±0.14	99.46±0.39	**100±0**	95.29±9.00
A2	82.63±8.12	**100±0**	97.38±4.44	99.87±0.22	94.97±8.32
A3	**97.21±3.23**	**100±0**	**100±0**	**100±0**	**99.30±1.40**

To compare the results more vividly, a bar plot is illustrated in Fig. 4-10. It can also be seen that the T0 task is the most challenging, with the lowest average accuracy being less than 60 %. Only the proposed method (A3) achieves an accuracy above 95 %. While IEDGNet (M3) gets an accuracy of about 92 %, all the proposed methods (A1-A3) achieve an average accuracy of about 95 %. The results show that the proposed method also generalizes better than the comparison methods on the PU dataset after introducing the pseudo domain labels. This also indicates that even for the dataset collected under constant speed conditions, introducing pseudo domain labels can further help characterize the dynamic feature distributions.

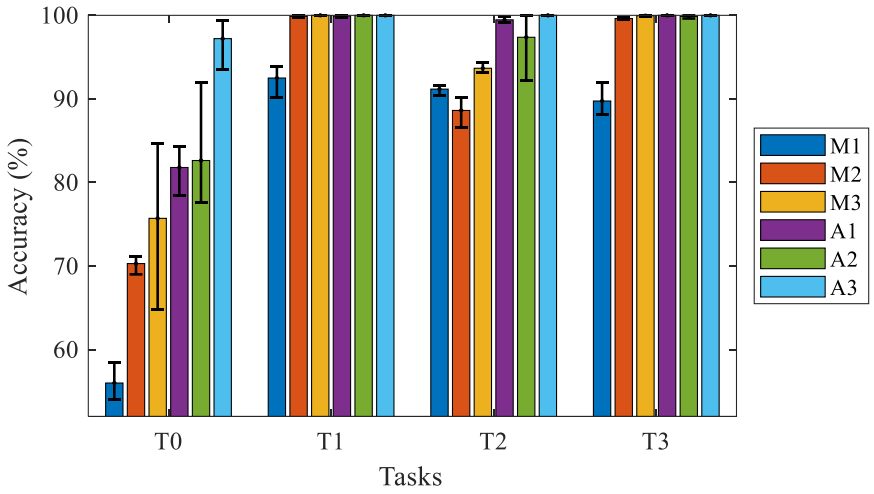
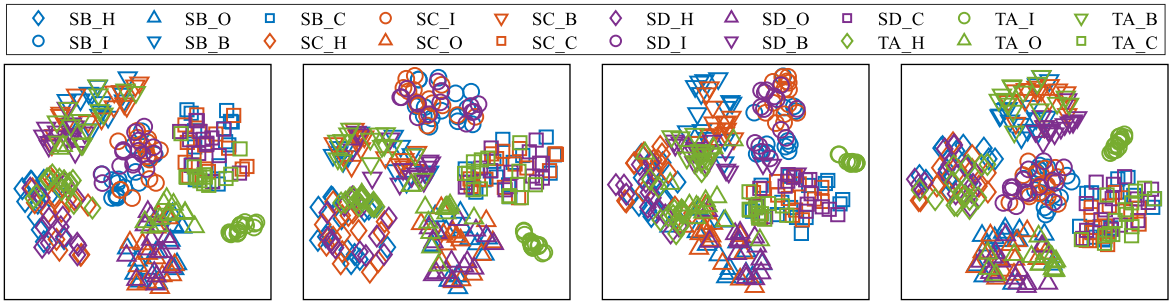


Fig. 4-10. Accuracy results using different methods with the PU dataset.

4.6 Discussion

4.6.1 Feature visualization of domain-invariant features

To study the cluster performance of the testing samples from the unseen target domain, feature visualization results are given in Fig. 4-11, where the UO-TA task is analyzed. Four different methods are used here for comparison. 30 pairs of samples in each class are drawn. Known samples from the same source domain are plotted using the same color, while different shapes represent different health states. For the UO-TA task, the data collected under speed condition A is unseen during training. By considering the samples represented by green circles, some feature distribution discrepancies between the known source and target domain data can be observed. M1-M3 results in Fig. 4-11 (a)-(c) wrongly classify some samples and the decision boundary between different health states is not clearly defined (especially in Fig. 4-11 (b) and (c), where the interclass separation cannot be clearly observed between the outer race fault, ball fault, and combined faults). For feature visualization results using the proposed method, the domain shift can still be observed, but the decision boundary for the target domain data is clear.

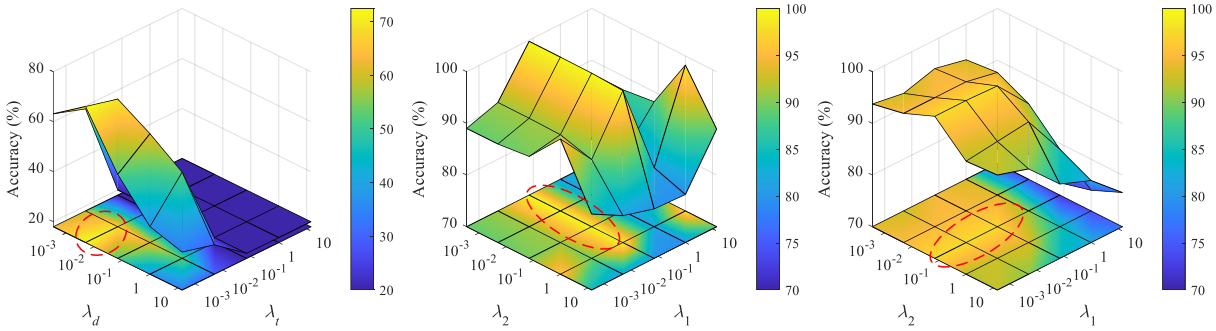


(a) M1 (b) M2 (c) M3 (d) A3

Fig. 4-11. Feature visualization results for the UO-TA task using different methods.

4.6.2 Hyperparameter analysis

As defined in Section 4.4, there are two trade-off parameters, λ_1 and λ_2, that are designed for the class and pseudo domain discriminators in the proposed method using an adversarial learning strategy. To investigate the effect of different parameter settings on experimental results, a series of comparison experiments is conducted by combining different values of λ_1 and λ_2, where both the parameters are selected from {0.001, 0.01, 0.1, 1, 10}. IEDGNet also has two hyperparameters (i.e., λ_t for triplet loss and λ_d for domain adversarial loss). These two parameters are also selected from {0.001, 0.01, 0.1, 1, 10}. The margin used for the triplet loss is set to 2 [71]. The UO-TA task is selected for further analysis. The accuracy results using IEDGNet and the proposed method are plotted in Fig. 4-12. It can be seen that the highest accuracy acquired by IEDGNet is almost 70 % while requiring a relatively small value of λ_t. If a larger value is used (e.g., 0.1, 1, and 10), the model will fail to diagnose specific fault types with an accuracy of almost 20 % (considering there are 5 different health states in total).



(a) UO-TA by IEDG (b) UO-TA by the proposed method (c) PU-T0 by the proposed method

Fig. 4-12. Accuracy results using different trade-off parameters.

Compared to IEDGNet, the accuracy performance of the proposed method in Fig. 4-12 (b) is relatively higher, ranging from 70 % to 100 %. When the trade-off parameter λ_2 is fixed, accuracy

experiences more significant fluctuations compared to the case when λ_1 is fixed. The comparison shows that the proposed method is more robust against variable hyperparameters and can achieve generally higher accuracies, especially when setting λ_1=0.1. These two parameters are also selected from a range of {0.001, 0.01, 0.1, 1, 10} by maximizing model robustness while maintaining considerable accuracy performance. To acquire a high accuracy performance on both datasets, $\lambda_1 = 0.1$ is preferred, as indicated by the red dashed circles in Fig. 4-12 (b)-(c). Once $\lambda_1 = 0.1$ is pre-determined, a flexible $\lambda_2 = 0.1$ can be selected for a more robust accuracy result on these two datasets at the same time. Therefore, the optimal combination of $\{\lambda_1 = 0.1, \lambda_2 = 0.1\}$ achieves the best trade-off between adversarial robustness and model stability via grid searching.

The learning rate μ controls the step size of the model parameter update, making it a vital hyperparameter in model training. Fig. 4-13 illustrates the diagnosis performance with different values of the learning rate μ for the proposed method. It can be found that the model's performance drops dramatically when $\mu \geq 0.1$. It may be because large values of the learning rate lead to large step sizes for model parameter updates, which may cause the model to be unable to converge during the training process. Then, among the three curves of $\mu < 0.1$, $\mu = 0.001$, provides the highest accuracy except for the UO-TA task, and is recommended in this article to guarantee mode performance.

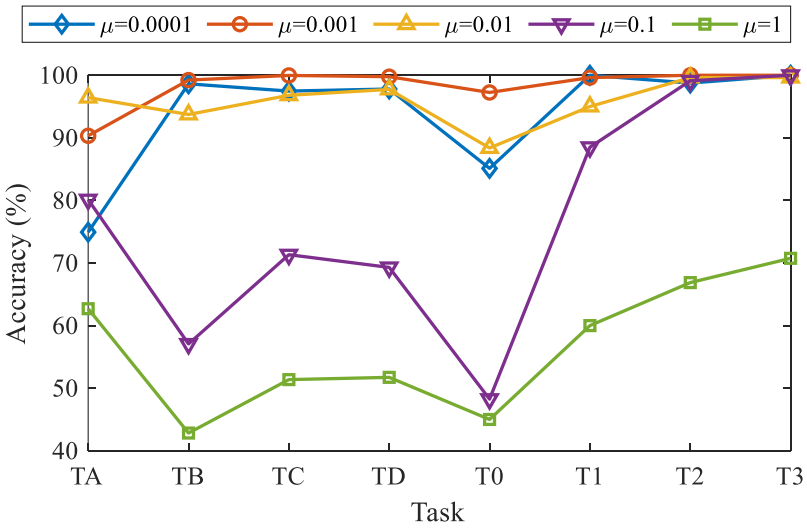


Fig. 4-13. The accuracy performance of different learning rates in the proposed method.

4.6.3 Number of pseudo domains

As mentioned, the proposed method will degenerate to a simplified version when the number of pseudo domains is set to 1 ($N_{pd} = 1$). That is, the collected samples will be assigned to the same pseudo domain label without considering the dynamic distributions within each working condition.

By increasing the number of pseudo domains, more diverse features can be extracted. On the contrary, increasing the number of pseudo domains to large values is not suggested either since the process becomes more time-consuming, as more subdomain centroids will be calculated. To find the optimal value for the number of pseudo domains, more experiments are conducted, where the number varies from 1 to 10, where setting the number to 1 can be considered as a comparison of the proposed method against other current studies. The accuracy versus the number of pseudo domains is shown in Fig. 4-14.

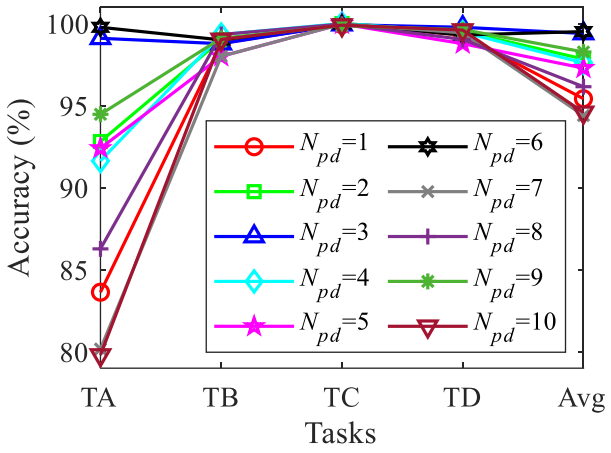


Fig. 4-14. Accuracy result versus number of pseudo domains on the UO dataset.

It can be found that among the four different tasks, the UO-TA task is the most sensitive to the number of pseudo domains. By comparing the results, it is suggested to set N_{pd}=6 for the UO bearing dataset, which leads to the highest average accuracy of 99.51 %. It can also be found that when compared with the accuracy results of N_{pd}=1 (the special case), more pseudo domains could further improve performance since more diverse features embedded into the multiple different subdomains can be explored, indicating the effectiveness of the pseudo domain strategy. The accuracy is 97.84 % when setting N_{pd}=2, while the accuracy is improved to 99.40 % when setting N_{pd}=3. However, it is also worth noting that setting too large a value for N_{pd} increases computational time since more subdomain centroids need to be calculated based on learned features. Furthermore, improved performance cannot always be guaranteed (i.e., setting N_{pd}=10 only leads to an average accuracy result of 94.58 %, which is less than 95 %). The optimal value for the number of pseudo domains deserves further study.

4.6.4 Pseudo domain feature distributions

To study the effectiveness of the proposed pseudo domain label during the training stage, the feature distributions across different subdomains are plotted, where the number of subdomains is set to 2, 5, and 8, respectively. That is, original feature distributions of given samples are further

divided into multiple latent subdomains, where the number of pseudo domains starts at 2 since a single pseudo domain is a special case. The corresponding feature distribution results obtained by using different numbers of latent domains are plotted in Fig. 4-15, where the last column represents the distributions of the unseen target domain samples. It can be seen that for these three different cases, most features can be well generalized from the known source domains to the unseen target domain. In Fig. 4-15 (a), if 2 subdomains are used, similar feature distributions as those shown in Fig. 4-4 can be obtained. However, for unseen target domain samples with an inner race fault, even though most features are well clustered, some features still act as outliers since target domain data is not involved in the model training stage. Similar results can be observed from feature visualization results in Section 4.6.1, where samples with an inner race fault act as outliers.

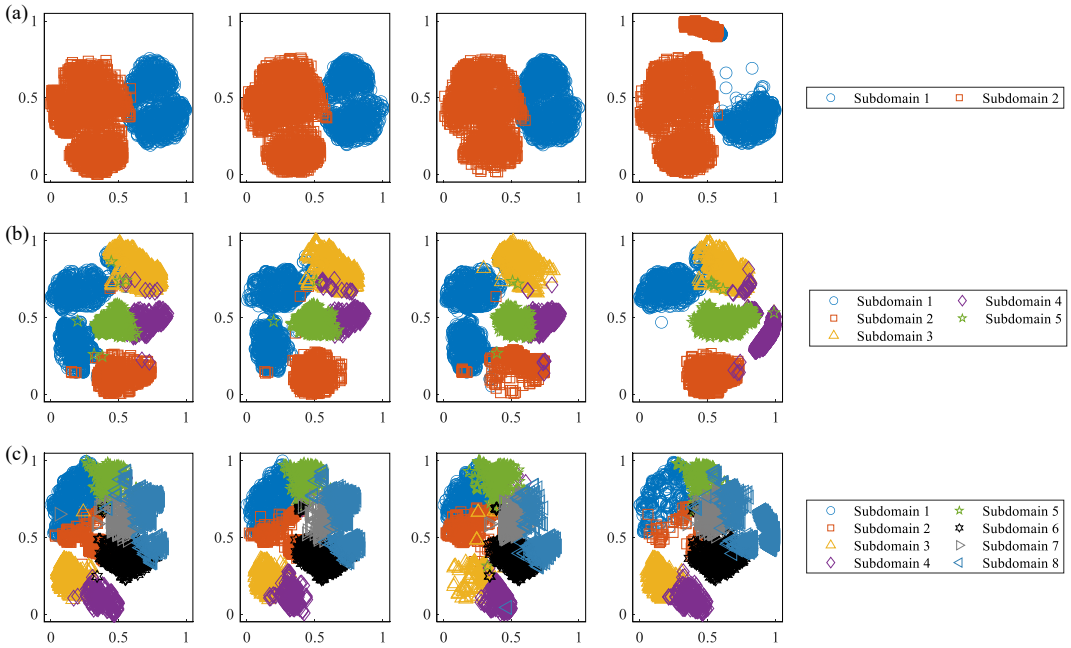
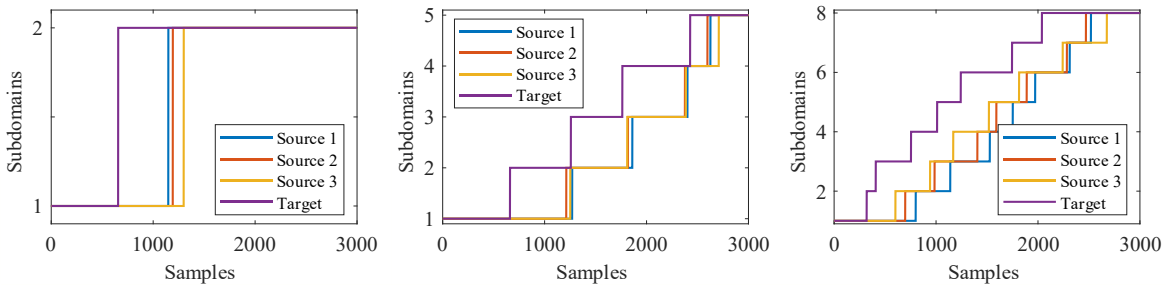


Fig. 4-15. Feature distributions in the latent subdomains of all samples from known source domains to the unseen target domain for the UO-TA task (left to right, feature visualization results from speeds B, C, D, and A): (a) 2 subdomains, (b) 5 subdomains, and (c) 8 subdomains.

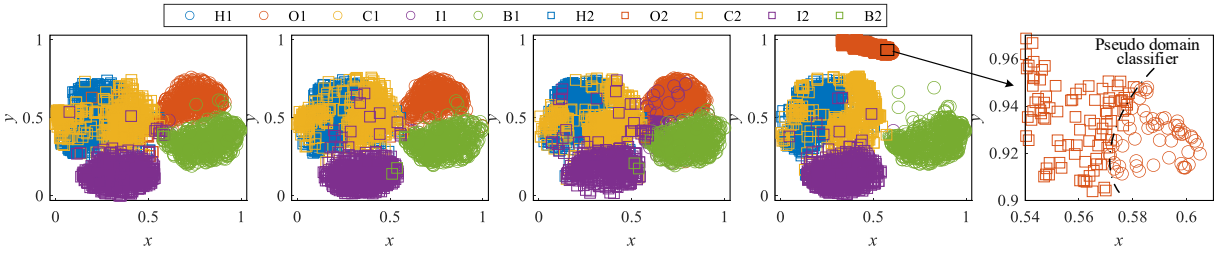
It is also worth noting that the contribution of each subdomain is not equal. Their percentages are dynamic since the working conditions are variable, as shown in Fig. 4-16. It can also be seen that the changing patterns of sample numbers in each subdomain of the known source domains share similarities. However, finding the optimal number of pseudo domains deserves further study to balance the diversity of the domain-specific features and computational complexity.



(a) 2 subdomains (b) 5 subdomains (c) 8 subdomains

Fig. 4-16. Number of samples assigned to each subdomain for the UO-TA task: (a) 2 subdomains, (b) 5 subdomains, and (c) 8 subdomains.

To further study the fine-grained features, detailed feature visualization results categorized by pseudo domain-class labels are provided, as illustrated in Fig. 4-17. Here. $N_{pd} = 2$, indicating that two subdomains are assumed for domain-specific feature learning. In Fig. 4-17, the samples assigned to pseudo domain 1 are all marked by circles, while samples that are divided into pseudo domain 2 are drawn with squares. It can then be concluded from Fig. 4-17 that most samples with a healthy state, an inner race fault and combined faults are aligned in pseudo domain 2, while most samples that indicate an outer race fault and a ball fault are divided into pseudo domain 1.



(a) source domain B (b) source domain C (c) source domain D (d) target domain A (e) enlarged area of (d)

Fig. 4-17. Fine-grained feature visualization results on the UO-TA task when $N_{pd} = 2$.

From Fig. 4-17 (a)-(c), it can also be clearly seen that among all 10 different classes, these three known source domains share similar feature distributions. For the unseen target domain A, the result in Fig. 4-17 (d) shows that a cluster which appears to be outliers can be observed from the middle top part of the feature map, and its corresponding labels are outer race faults in pseudo domain 2. Furthermore, it can be found that the right edge of this extra cluster is much closer to samples sharing a label of O1 (an outer race fault in pseudo domain 1). By zooming in the local area marked by the box in Fig. 4-17 (d), the corresponding feature distribution is plotted in Fig. 4-17 (e), where a pseudo domain classifier (represented by a dashed line) between two pseudo domains can be observed. This indicates that part of the samples with O1 labels in the target

domain are covered by transferring prior knowledge trained on multiple source domains.

4.6.5 Application of limited training samples

In real industrial applications, it is sometimes difficult to collect and label a large number of samples. That is, the number of collected known samples is limited. To simulate this case, different data size experiments are conducted to study the performance of the proposed method. These include data sizes of 32, 160, 320, 640, 960, 1280, and 1600, when using a batch size equal to 32. The size of the test data remains unchanged. The accuracy results using MMD (M1), DANN (M2), IEDGNet (M3), and the proposed method (A3) are listed in Table 4.9, where the UO-TB task is analyzed here since almost all comparison methods obtain an accuracy of over 90 %, showing that all methods with full access to training data can obtain a satisfactory performance. The proposed method is capable of achieving adequate results (i.e., greater than 95% accuracy) with far less data.

Table 4.9. Accuracy results (%) on the UO-TB task by using limited training data.

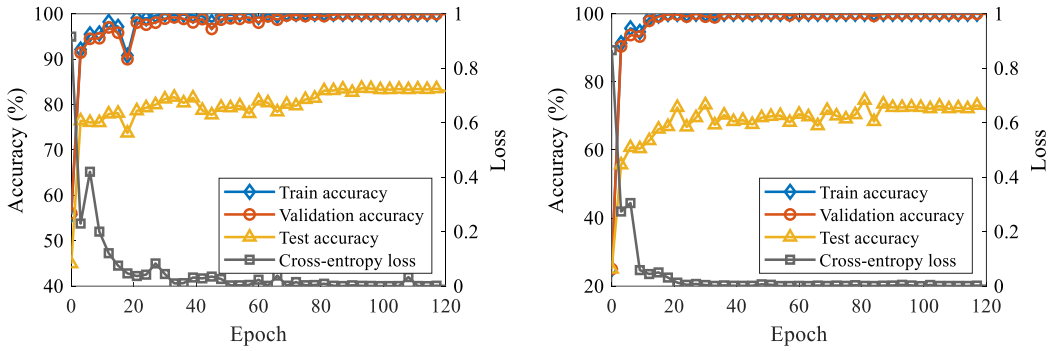
Training size / Method	32	160	320	640	960	1280	1600
M1	47.58±7.24	73.54±0.48	78.85±2.24	82.96±1.80	84.72±3.51	86.46±2.09	87.64±2.32
M2	56.72±1.47	84.94±2.43	89.59±2.15	92.51±0.63	93.14±0.15	92.93±0.93	94.39±0.10
M3	48.06±2.69	64.83±3.37	62.67±6.61	74.53±4.29	83.94±4.16	81.00±8.37	74.90±6.39
A3	**79.01±3.08**	**94.68±1.30**	**96.56±1.57**	**97.20±1.32**	**98.92±0.37**	**98.11±0.85**	**99.01±0.19**

It can also be seen from Table 4.9 that by increasing the number of training samples from 32 to 960, all methods increase their diagnosis precision. However, using more than 960 samples during training leads to overfitting of the model, which can be observed when the accuracy on the test sample decreases (for M2, M3, and A3). It can be found that the proposed method performs better than the compared methods when limited training samples are available. M2 achieves the second-highest accuracy. However, the proposed method obtains an accuracy of 94.68 % after 5 steps per epoch, while M2 requires 10 times more data, showing that the proposed method can still generalize well to unseen target domains with limited training data.

4.6.6 Architecture robustness

In Section 4.4.3, a table that lists all the network structures used in this study is provided, where a CNN-based model is selected as a backbone. To further extend the generalization capacity based on the proposed method, a more advanced version of the network structure can be developed (i.e., replacing the original basic CNN-based feature extractor with a well-established ResNet18 structure) [35]. It is widely acknowledged that ResNet is efficient by involving shortcut connections that enable network parameter updates through deeper layers. First, to study the model convergence, the UO-TA and PU-T0 tasks are analyzed. Training history versus epoch is provided

in Fig. 4-18 (a)-(d), where source domain data is further divided into two parts using a train split ratio of 0.8 / 0.2 (0.8 for training and 0.2 for validation). It can be clearly seen from Fig. 4-18 (a)-(b) that the CNN-based model stabilizes after about 60 epochs on the UO-TA task and a faster convergence is observed on the PU-T0 task after 30 epochs. By comparing the CNN and ResNet18, it can be concluded that the well-constructed ResNet18 achieves faster convergence and smaller fluctuations than the CNN. Given these results, setting the maximum epoch to 120 (referring to Table 4.6) ensures sufficient training while maintaining computational efficiency.



(a) UO-TA task using a CNN-based model

(b) PU-T0 task using a CNN-based model

(c) UO-TA task using a ResNet18-based model

(d) PU-T0 task using a ResNet18-based model

Fig. 4-18. Accuracy and loss versus training epoch.

Considering both CNN-based and ResNet18-based models both converge after a certain number of epochs, and the PU dataset has fewer test samples, the accuracy results of all compared methods on the PU bearing dataset are then provided in Table 4.10 to verify the effectiveness of the proposed method under different neural network architectures. It can be found from Table 4.10 that higher accuracies are obtained by all methods compared, especially when compared with the accuracy results listed in Table 4.8, indicating that the proposed method's effectiveness still holds under different network structures.

Table 4.10. Accuracy results (%) on the PU dataset based on ResNet18.

Methods	T0	T1	T2	T3	Average
M1	73.92±5.60	99.37±0.67	99.92±0.07	99.54±0.69	93.19±12.85
M2	79.50±8.13	**100±0**	92.75±4.99	**100±0**	93.06±9.67
M3	81.66±10.22	**100±0**	99.87±0.22	**100±0**	95.38±9.15
A1	99.21±0.83	**100±0**	**100±0**	**100±0**	99.80±0.40
A2	**99.96±0.07**	**100±0**	**100±0**	**100±0**	**99.99±0.02**
A3	99.92±0.14	**100±0**	**100±0**	**100±0**	99.97±0.04

4.7 Conclusion

In this chapter, a new method for domain generalization-based IFD using a pseudo domain labeling strategy is proposed, where the original feature distributions are further divided into several subdomains by assigning samples pseudo domain labels. This improves IFD performance when the target domain is completely unseen. Training for the proposed method involved three steps: (1) fine-grained feature learning from both domain- and class-levels, (2) exploring domain-specific features by assigning samples with pseudo domain labels, and (3) learning domain-invariant features across multiple pseudo domains following adversarial learning. The proposed method first combines information from both the domain- and class-levels, which helps learn fine-grained features following intraclass aggregation and interclass separation. This step enables the model to capture as many diverse features as possible. Then, another two branches employ adversarial learning strategies to separate the domain and class information. By constructing a class discriminator, domain-specific features existing in each subdomain can be explored, and the diversity of the automatically extracted features can be ensured. At the same time, features shared among multiple subdomains can also help in the process of learning domain-invariant features by exploring their similarities. Experiments on 8 different tasks from two publicly available bearing datasets have verified the effectiveness of the proposed method for IFD by exploring feature generalization to unseen target domains. By including both constant-speed and variable-speed datasets, a comprehensive evaluation of the proposed method is provided, ensuring that the proposed method is not only applicable under variable speed conditions but also maintains strong generalization performance in simpler, constant speed scenarios. Also, by limiting access to training data, the proposed method has been proven to be more effective than some current methods, showing that the proposed method is valuable when performing IFD and has the potential to deal with tasks where less labeled data is available.

By aligning feature distribution among multiple subdomains at the same time, the dynamic distributions embedded in a single domain related to variable working conditions can be studied. However, it is worth noting that the optimal number of pseudo domains (e.g., N_{pd}=2) is manually determined in this chapter. Studies of different values are discussed in Section 4.6.3. Considering

that larger numbers may ensure more diversity in the learned features at the cost of more computation time, a modest range in the number may be preferred (e.g., 2-10). For instance, when N_{pd}=6, the accuracy performance of the proposed method with the UO dataset could be further improved when compared with N_{pd}=2. It is also worth noting that if the number of pseudo domains is set to 1 (i.e., N_{pd}=1), the proposed method degenerates to a DANN method without the contribution of a class discriminator. To better perform IFD using the pseudo domain label assignment strategy, determining the number of optimal pseudo domains (N_{pd}) deserves further study.

Chapter 5 Domain interference suppression for reliable fault diagnosis under unseen operating conditions

This chapter addresses objectives 1, 2, and 3. In this chapter, a new IFD method is proposed from the perspective of domain-specificity removal. The effectiveness of the proposed method is verified using unseen target data and a limited training sample scenario.

The contents of this chapter have been published in *Mechanical Systems and Signal Processing*.

Zehui Hua, Juanjuan Shi, Patrick Dumond, Domain interference suppression for reliable fault diagnosis under unseen operating conditions, *Mechanical Systems and Signal Processing*, 256, 2026, 114457.

Authorship contribution statement:

Zehui Hua: writing of the original draft, algorithm development and implementation, writing review and editing, methodology, funding acquisition, conceptualization;

Juanjuan Shi: writing review and editing, validation, funding acquisition;

Patrick Dumond: writing review and editing, supervision, project administration.

5.1 Abstract

Fault diagnosis under varying operating conditions remains challenging due to the entanglement of fault-related information and condition-induced variations in measured signals. In practical industrial scenarios, continuously varying operating conditions such as speed changes can induce significant intra-class variability within the same operating state, which in turn leads to unstable feature representations and compromised diagnostic reliability under unseen conditions. Motivated by the observation that fault-related and condition-specific information are entangled in measured signals but separable at the representation level, this chapter models them as coexisting factors within a shared representation space, where condition-induced variations act as structured interference. Accordingly, a domain interference suppression (DIS) framework is proposed to mitigate condition-related interference while preserving fault-discriminative representations. To stabilize the learning of condition-sensitive variations and prevent representation collapse, a teacher-student learning strategy is introduced, in which the teacher network provides structured guidance during training, enabling the student network to learn more consistent feature representations without increasing inference complexity. The proposed DIS framework is evaluated on two public bearing fault diagnosis datasets under continuous speed variation and unseen operating conditions. Experimental results demonstrate that suppressing domain interference leads to more stable and reliable diagnostic performance compared with state-of-the-art domain generalization methods.

5.2 Introduction

Fault diagnosis of rotating machinery under variable working conditions remains a long-standing challenge in reliability and condition monitoring [80]. In practical industrial environments, operating conditions such as rotational speed and load often vary continuously over time, introducing substantial nonstationarity into collected vibration signals. These condition-induced variations can significantly distort fault-related signatures, leading to degraded diagnostic performance and poor generalization when models are trained under limited conditions but deployed in unseen scenarios [81]. Due to the fast development of science and technology, big data-based fault diagnosis methods have become interesting for both researchers and engineers [82]. Big data-based fault diagnosis utilizes machine learning architectures to train diagnostic models, known as IFD [2,4]. Once a model is well-trained, it can provide accurate and real-time fault diagnosis results. Combined with transfer learning, models have the potential to provide promising results for new data collected under variable working conditions [3].

However, collected vibration data are often highly sensitive to actual working conditions, leading to pronounced distribution discrepancies [24]. Such domain shifts prevent newly observed

feature distributions from being adequately covered by models trained under limited conditions, resulting in significant performance degradation. To address the unknown domain shifts, DA has been widely adopted in fault diagnosis by leveraging transfer learning to reduce distribution discrepancies between the source and target domains [83]. By aligning feature distribution across domains, DA-based methods can mitigate certain condition-dependent variations and improve accuracy performance [84]. Nevertheless, most existing DA methods implicitly assume that fault-related information can be isolated from condition-related variations in the learned representation. This assumption is particularly restrictive in practical scenarios involving continuously varying operating conditions, where fault characteristics and condition effects are inherently entangled within a shared feature space. As a result, residual domain interference may persist even after distribution alignment, limiting the robustness of fault diagnosis. From a methodological perspective, metric learning and adversarial learning are two commonly used strategies for domain alignment. Metric learning guides the convergence of the network training process by minimizing the measured distance between samples from different domains in the learned feature space. [49]. For adversarial learning, the feature extractor acts as a generator to extract features, while the domain discriminator is used to help the model distinguish between the sources of features. Once the domain discriminator fails to distinguish the sources, it can be considered that domain-invariant features are learned [26]. While these approaches have demonstrated effectiveness in scenarios involving a limited number of discrete domains, their extension to multi-source settings and continuously varying conditions remains challenging [85]. In particular, how to regulate the contribution of different source domains, rather than relying on global alignment or averaging, is still an open issue for achieving robust generalization [86].

DA-based IFD methods rely on the availability of target domain data during training and therefore cannot be applied to scenarios where target domain data is completely unseen [26]. To overcome this limitation, DG methods have been developed to enable models trained on multiple source domains to generalize to unknown target domains, and have attracted increasing attention in recent years [67]. Although DG methods are effective when the target domain is unseen, the model's capacity for generalization still needs to be improved, especially when combined with semi-supervised and limited training samples [87]. Most existing DG approaches in fault diagnosis focus on directly distilling domain-invariant features, aiming to eliminate the influence of domain-specific variations induced by different working conditions. While encouraging results have been reported, this paradigm implicitly treats domain-specific information as undesirable noise. In practice, however, domain-specific and domain-invariant characteristics are inherently coupled within vibration signals collected under variable operating conditions. Neglecting the distinct yet informative patterns associated with individual domains may lead to over-suppressed

representations, thereby weakening fault-discriminative structures and limiting generalization to unseen conditions [68]. Several recent studies have attempted to address domain variability from different perspectives. For instance, Hua et al. introduced pseudo domain labels to characterize dynamic feature distributions and facilitate fine-grained alignment across latent subdomains by emphasizing domain-specific features [88]. Liu et al. proposed to remove operational condition information from mixed features in order to obtain cross-domain-invariant representations [89]. Xie et al. exploited domain-specific information as auxiliary supervision and employed data generation strategies to transfer labeled source domain samples toward the target domain [90]. Song et al. adopted dual feature extractors to separately model domain-invariant and domain-specific features for semi-supervised fault diagnosis [68]. Wang et al. proposed an adversarial learning-based method to learn feature representations and reduce the influence of distant source domains, where domain-invariant features are studied through a domain-specific adversarial learning process [34]. Despite these efforts, most existing DG methods ultimately aim to enhance domain-invariant representations by suppressing or isolating domain-specific variations. Such strategies overlook the fact that domain-specific information is not merely a nuisance factor but a structured and working-condition-sensitive component that can interfere with fault discrimination if left unregulated. This observation indicates that, rather than completely removing domain-specific features, it is essential to suppress their interfering effects on fault-related representations to achieve robust generalization under variable operating conditions.

To effectively perform DG-based IFD, DIS is proposed to perform effective generalization fault diagnosis. The main contribution of the proposed method can be summarized as follows:

(1) Conceptual perspective on domain-specificity in DG-based IFD. This work investigates the role of domain-specific information induced by variable working conditions in DG-based IFD. Domain-specific features are associated with operating conditions, whereas domain-invariant features correspond to the health states of key rotating components. This perspective highlights that domain-specific information is not noise to be discarded, but a condition-sensitive factor whose interference should be properly regulated to achieve robust fault diagnosis.

(2) A two-stage learning framework for domain interference suppression. A two-step training strategy is proposed to regulate domain-specific interference. First, a teacher network is trained via contrastive learning to capture condition-sensitive characteristics under variable operating conditions. These representations are then transferred to a student network through KD, guiding the student to regulate the influence of domain-specific factors during feature extraction. This design stabilizes training and facilitates effective interference regulation without increasing inference complexity.

(3) Feature-level interference regulation via dissimilarity. To further suppress domain interference, feature-level regulation is introduced by maximizing the dissimilarity between fault-discriminative and condition-sensitive representations using cosine similarity. By discouraging overlap between these representations, the proposed method enhances the robustness of transferable features and improves generalization performance under unseen operating conditions, including scenarios with limited training samples. The effectiveness of the proposed DIS framework is validated on two public bearing fault datasets.

The remainder of this chapter is laid out as follows. Section 5.3 introduces the background of DG-based IFD, contrastive learning, and KD, followed by a discussion of domain-invariant and condition-sensitive features that coexist in vibration signals under variable operating conditions. Section 5.4 presents the proposed DIS framework and details the learning and regulation of different domain-specific and domain-invariant feature representations, as well as the overall training procedure. Section 5.5 reports comparative experimental results on two public bearing datasets under variable working conditions, including feature visualization. Section 5.6 provides in-depth discussions, including ablation studies, a sensitivity analysis, domain-specific feature interpretations, performance under limited training sample settings, statistical validation, and a computational efficiency analysis. Finally, conclusions are drawn in Section 5.7

5.3 Preliminary knowledge

5.3.1 Intelligent fault diagnosis using domain generalization

DG-based IFD aims to train a generalized model to make it apply to new data that is not involved in training. It is possible that the collected monitoring data shows different feature distributions, such as time-varying speeds and loads. Assuming there are k accessible source domains, recorded as $\left\{\mathcal{D}_S^i\right\}_{i=1}^k$, the model will be trained using all fully labeled source domains and tested on the unseen target domain $\mathcal{D}_T$. Unseen here means that the target domain data is not available, thus data in the target domain does not contribute to model training. Domain shifts exist due to variable working conditions between different source domains. In this case, feature distributions vary between each source and can be recorded as $P\left(\mathcal{X}_S^1\right) \neq P\left(\mathcal{X}_S^2\right) \neq \cdots \neq P\left(\mathcal{X}_S^k\right)$. In the label space, source and target domains do not have category shifts (i.e., no emerging or extra fault types).

DG provides notable advantages over DA by enabling fault diagnosis models to generalize to unseen operating conditions without requiring access to target domain data during training. By focusing on learning transferable fault-related representations, DG-based methods are better suited for practical scenarios where operating conditions vary unpredictably.

The DG-based method can be further extended to more challenging settings by considering scenarios with scarce data, such as limited availability of labeled source domain samples or restricted access to training data. In such cases, the key objective is to maintain satisfactory diagnostic performance despite reduced data volume. Ensuring robustness under limited-data conditions is essential for real-world applications, where collecting sufficient labeled vibration data is often costly or impractical.

5.3.2 Contrastive learning

Contrastive learning is a branch of self-supervised learning that is primarily used in representation learning. The aim is to learn feature representations of samples by contrasting positive (similar or related) pairs against negative (dissimilar or unrelated) pairs [91]. By encouraging consistency among related samples while separating unrelated ones in the embedding space, contrastive learning enables effective extraction of structured representations without heavy reliance on labeled data.

In fault diagnosis under variable operating conditions, contrastive learning provides a natural mechanism for modeling condition-sensitive characteristics. Samples collected under similar operating conditions can be treated as positive pairs, while those from different conditions form negative pairs, allowing the model to capture variations induced by working conditions. This property makes contrastive learning particularly suitable for learning domain-specific representations that reflect condition differences, which can subsequently be leveraged to regulate their interfering influence on fault discriminative features.

5.3.3 Knowledge distillation

Motivated by knowledge transfer, KD is a two-step model training technique that distills knowledge from a larger teacher network to a student network without losing important information, which of course makes model inference more efficient. After the teacher network is well trained, the student network is trained guided by a distillation loss so that its inferences are similar to those of the teacher network. This allows the transfer of knowledge between different models, which also leads to faster model convergence.

The concept of KD was popularized by Hinton et al. by introducing the idea of using soft outputs to smooth the probability distribution, expressed as [92]:

$$p_i = \frac{\exp\left(z_i/T\right)}{\sum_j \exp\left(z_i/T\right)} \tag{5-1}$$

where z_i is the logits of the original output of the model, and T is the temperature to smooth the probability. A special case is T=1, where the SoftMax function will behave as usual. Training of

the teacher network is guided by a cross-entropy loss based on the true label y and the teacher's prediction $\hat{y}_t$. Similarly, training of the student network is then guided by both cross-entropy loss and KD loss, which can be expressed as:

$$\mathcal{L} = \ell_{CE}\left(y, \hat{y}_s\right) + \ell_{KD}\left(p_t, p_s\right) \tag{5-2}$$

where ℓ_{CE} is the classifier loss based on the true label y and the student's prediction $\hat{y}_s$, ℓ_{KD} is the distillation loss between the teacher's and the student's predictions (p_t , p_s). ℓ_{KD} encourages the student to learn the teacher's performance, which ensures that enhanced features can be learned during KD.

5.3.4 Domain-invariant versus domain-specific features

For DG-based IFD, domain-invariant features refer to transferable representations that are shared across different domains and remain effective for fault identification regardless of domain shifts. These features capture fault-related features that are consistent between source and target domains, enabling a trained model to correctly predict the health state of unseen samples under varying operating conditions.

In contrast, domain-specific features correspond to representations that are sensitive to particular operating conditions and vary across domains. Such features are generally non-transferable and can degrade model generalization when they dominate the learned representation. In practical fault diagnosis scenarios, domain-specific information often reflects operating condition variations associated with individual domains. It is worth noting that domain-invariant and specific information typically coexist within the same vibration signal rather than being cleanly separable. While domain-invariant features support robust fault diagnosis, domain-specific features encode condition-related variations that may interfere with the learning of transferable representations. Therefore, instead of removing domain-specific information, the proposed DIS strategy focuses on regulating its influence, encouraging the model to emphasize fault discriminative features while mitigating condition-induced interference.

Under variable operating conditions, such interference is further intensified by continuously changing working conditions, such as rotational speed variations. In raw vibration measurements, fault-related characteristics (in blue) and condition-induced variations (in orange) are inherently entangled, making it difficult to directly extract representations that are both discriminative and generalizable, as shown in Fig. 5-1 (a). From a representation learning perspective, both factors are jointly embedded into a shared latent space by a single feature extractor, as shown in Fig. 5-1 (b), where different geometric structures can emerge depending on how the learned representation is viewed.

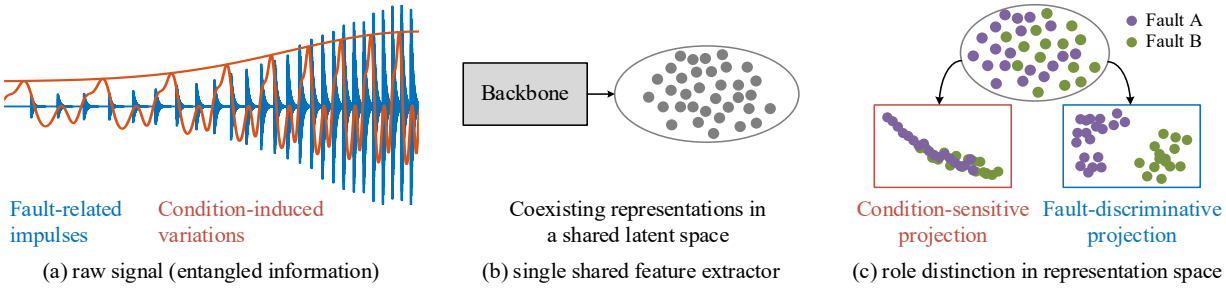


Fig. 5-1. Illustration of coexisting representations under variable working conditions.

In Fig. 5-1 (c), a condition-sensitive projection organizes samples along a continuous manifold reflecting condition variations, while fault classes remain entangled, whereas a fault discriminative projection yields compact and separable clusters corresponding to different fault types. This geometric distinction motivates regulating condition-sensitive interference to facilitate robust invariant fault discrimination. Based on this perspective, the next section introduces a domain interference suppression framework that models and regulates condition-sensitive representations to facilitate robust domain-invariant feature learning.

5.4 Proposed domain interference suppression framework

To enhance the generalization capability of DG-based IFD, it is essential to model condition-sensitive variations that are embedded in vibration signals collected under variable operating conditions. As discussed in Section 2, fault-related and condition-specific information typically coexist within a shared representation space rather than being cleanly separable. Therefore, instead of removing domain-specific information, the proposed DIS framework focuses on learning condition-sensitive representations and regulating their interfering influence on fault-discriminative features.

5.4.1 Condition-sensitive representation learning

In practical fault diagnosis scenarios, operating conditions such as rotational speed and load vary across domains and induce domain-specific variations in vibration signals. To capture these condition-sensitive characteristics, domain labels available in the source domains are exploited to guide representation learning. Specifically, a contrastive learning objective is adopted in the teacher network to encourage samples from the same domain to be mapped closer together while pushing samples from different domains apart. This process enables the model to learn representations that emphasize domain-related variations without relying on target-domain data. Given two samples x_i and x_j with domain labels d_{x_i} and d_{x_j}, the contrastive loss function is defined as:

$$\mathcal{L}_{CL} = -\sum_{i=1}^{N} \log \frac{\sum_{j \in \mathcal{P}(i)} \exp\left(sim\left(F\left(x_i\right), F\left(x_j\right)\right)\right)}{\sum_{j \in \mathcal{P}(i) \cup \mathcal{N}(i)} \exp\left(sim\left(F\left(x_i\right), F\left(x_j\right)\right)\right)} \tag{5-3}$$

$$\begin{aligned} \mathcal{P}(i) &= \left\{ j \middle| d_{x_j} = d_{x_i}, j \neq i \right\} \\ \mathcal{N}(i) &= \left\{ j \middle| d_{x_j} \neq d_{x_i} \right\} \end{aligned} \tag{5-4}$$

where F denotes the feature extractor, $\mathcal{P}(i)$ denotes positive pairs and $\mathcal{N}(i)$ denotes negative pairs. $sim(\cdot)$ is a similarity function (a contrastive objective with cosine similarity and temperature scaling $\tau = 0.07$ is used here), recorded as:

$$sim\left(\mathbf{z}_i, \mathbf{z}_j\right) = \frac{\mathbf{z}_i^{\mathrm{T}} \mathbf{z}_j}{\tau}, \left\|\mathbf{z}\right\|_2 = 1 \tag{5-5}$$

This objective encourages the teacher network to focus on condition-sensitive representations that characterize domain variations.

5.4.2 Domain-specific feature learning via knowledge distillation

To enable effective domain generalization, condition-sensitive and fault-discriminative representations must be jointly modeled and balanced within a shared feature space. While condition-sensitive representations capture domain-related variations, fault diagnosis ultimately relies on transferable fault-discriminative features. Therefore, domain-specific knowledge learned in the auxiliary branch must be appropriately transferred and regulated in the main branch.

As introduced in Section 5.4.1, condition-sensitive representations are first learned in the teacher network using domain-supervised contrastive learning. To transfer this knowledge, a KD strategy is adopted, where the student network is guided to learn condition-sensitive representations consistent with those of the teacher. Specifically, a MSE loss is adopted to enforce instance-level consistency rather than distribution-level matching, ensuring that the student learns domain-specific features with similar geometric structure, which is defined as:

$$\mathcal{L}_{mse} = \left\| f_{spec,tea} - f_{spec,stu} \right\|_2^2 \tag{5-6}$$

where $f_{spec,tea}$ and $f_{spec,stu}$ respectively denote the condition-induced specific features in the teacher and student network, respectively. In the student network, condition-sensitive representations are further reinforced through adversarial learning. A domain classifier is trained to predict domain labels from the corresponding feature representations, ensuring that domain-related variations are effectively captured. The associated cross-entropy loss is defined as follows:

$$\mathcal{L}_{d-cls} = \ell\left(h_{d-cls}\left(h_{b,stu}\left(h_{f,stu}\left(x\right)\right)\right), d_x\right) \tag{5-7}$$

where x is the training sample, d_x is its corresponding domain label, $h_{f,stu}$, $h_{b,stu}$, and $h_{d\text{-}cls}$ denote the feature extractor, bottleneck, and domain classifier layers in the student network, respectively.

In addition, class adversarial learning is incorporated to prevent fault-related information from dominating condition-sensitive representations. This mechanism can be regarded as an extension of DANN, where adversarial learning is applied with respect to class labels rather than domain labels [34]. The resulting objectives are written as:

$$\mathcal{L}_{c-adv} = \ell\left(h_{c-adv}\left(R_{\lambda_1}\left(h_{b,stu}\left(h_{f,stu}\left(x\right)\right)\right)\right), y_x\right) \tag{5-8}$$

where y_x and d_x denote class and domain labels of the sample x, respectively. λ_1 is the trade-off parameter of the class discriminator $h_{c\text{-}adv}$ after applying a gradient reserve layer (GRL). Through the combined use of contrastive learning, KD, and adversarial regulation, the proposed framework enables modeling and controlled influence of condition-sensitive representations, laying the foundation for subsequent domain-invariant feature learning. The overall process of learning and regulating condition-sensitive representations is summarized in Fig. 5-2. First, domain-specific information induced by variable operating conditions is modeled as a condition-sensitive factor whose interference is geometrically regulated rather than removed or clearly separated, consistent with the proposed conceptual perspective. Second, a two-stage KD framework is adopted, where the teacher network provides a domain-pure reference to stabilize the learning of condition-sensitive representations in the student network. Third, feature-level interference regulation is achieved by discouraging excessive alignment between condition-sensitive and fault-discriminative features, enabling the student network to mitigate domain interference while preserving fault discrimination capability.

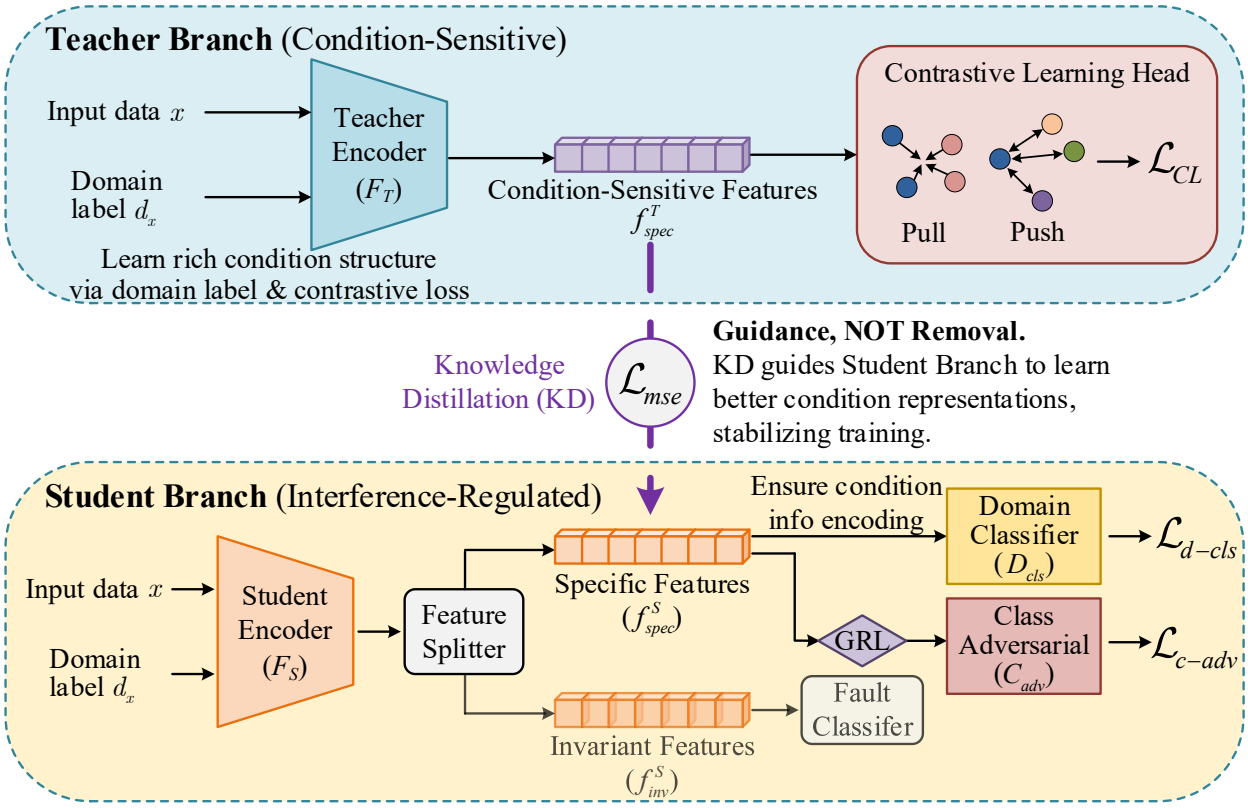


Fig. 5-2. Domain-specific feature learning guided by domain labels.

5.4.3 Domain-invariant feature learning

As previously mentioned, domain-invariant features are expected to capture fault-related characteristics that are transferable across different domains, enabling robust fault diagnosis under unseen operating conditions. So, at this point, adversarial learning is adopted to reduce the sensitivity of learned representations to domain variations. Specifically, a preliminary framework based on DANN is employed, where the feature extractor $h_{f,stu}$, bottleneck $h_{b,stu}$, classifier h_{cls} and domain discriminator h_{d-adv} jointly participate in training. During inference, the domain discriminator is removed, and only the feature extractor, bottleneck, and classifier are retained for fault diagnosis on unseen target domain data. The optimization objectives for the classifier and domain discriminator are given as:

$$\mathcal{L}_{cls} = \ell\left(h_{cls}\left(h_{b,stu}\left(h_{f,stu}\left(x\right)\right)\right), y_x\right) \tag{5-9}$$

$$\mathcal{L}_{d-adv} = \ell\left(h_{d-adv}\left(R_{\lambda_1}\left(h_{b,stu}\left(h_{f,stu}\left(x\right)\right)\right)\right), d_x\right) \tag{5-10}$$

where R_{λ_1} denotes the GRL with hyperparameter λ_1 (the same parameter value is shared here since they are both evolved in adversarial learning). Also, it is worth noting that the representations learned after the bottleneck layer are treated as a shared feature embedding, within which different

components may play different roles during training. For implementation convenience, the bottleneck features are partitioned into two designated subspaces: one primarily associated with fault-discriminative learning and the other associated with condition-sensitive modeling. Specifically, the domain-invariant features learned in the student network are recorded as:

$$f_{inv,stu} = h_{b,stu}\left(h_{f,stu}\left(x\right)\right)_{\lfloor n/2\rfloor+1,\lfloor n/2\rfloor+2,\cdots,n} \tag{5-11}$$

while the condition-sensitive (domain-specific) features are defined as

$$f_{spec,stu} = h_{b,stu}\left(h_{f,stu}\left(x\right)\right)_{1,2,\cdots,\lfloor n/2\rfloor} \tag{5-12}$$

where $\lfloor\rfloor$ represents the round-down operation, and n is the dimensionality of the bottleneck feature. This partition does not imply clean separability of the underlying factors; instead, it provides a structural means to apply different learning objectives and regulation constraints within a shared representation space.

To suppress the interfering influence of condition-sensitive representations on fault-discriminative learning, a cosine similarity-based regulation term is introduced to reduce the feature redundancy between $f_{inv,stu}$ and $f_{spec,stu}$. The loss function guided by cosine similarity can be written as

$$\mathcal{L}_{cos} = \sum_{i=1}^{k}\sum_{j=1}^{n_s}\frac{h_{f_{inv,stu}}\left(x_{i,j}^{s}\right)^{T}\cdot h_{f_{spec,stu}}\left(x_{i,j}^{s}\right)}{\left\|h_{f_{inv,stu}}\left(x_{i,j}^{s}\right)\right\|\left\|h_{f_{spec,stu}}\left(x_{i,j}^{s}\right)\right\|} \tag{5-13}$$

By penalizing high similarity between the two representations, this regulation discourages excessive overlap and encourages feature diversity, thereby limiting the dominance of condition-related variations in fault discrimination. Notably, the objective is not to enforce strict feature separation or opposite alignment, but to regulate the interaction between domain-invariant and domain-specific representations, ensuring robust generalization under variable operating conditions via feature partition and redundancy reduction.

5.4.4 Overview of the proposed method

The whole flowchart of the proposed DIS method that leverages domain-invariant features while removing domain-specific features is illustrated in Fig. 5-3.

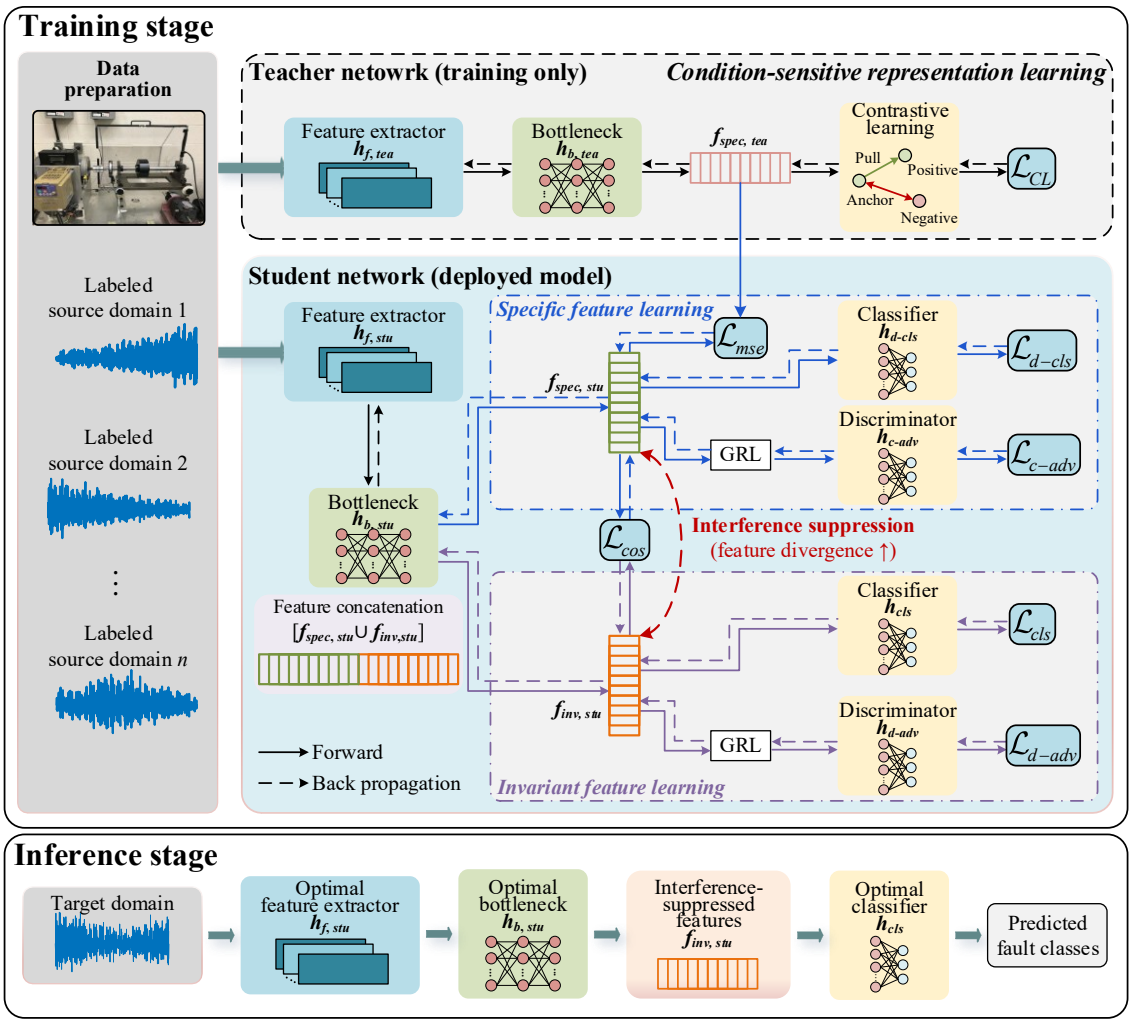


Fig. 5-3. Detailed network of the proposed DIS method for IFD.

First, condition-sensitive (domain-specific) features are learned in the teacher network under the guidance of a domain-supervised contrastive, as illustrated in Fig. 5-3. After the teacher network is optimized with respect to the feature extractor and bottleneck layers, domain-specific characteristics induced by operating-condition variations are captured. The learned condition-sensitive knowledge is then transferred to the student network through KD, providing structured guidance for recognizing domain-related variations during training. Specifically, the domain-specific features learned in the student network are aligned with those of the teacher network by minimizing the MSE loss, which stabilizes training and facilitates more effective learning compared with random initialization.

In the student branch, condition-sensitive representations are further regulated through adversarial learning. A domain classifier is trained to reinforce the encoding of domain-related variations, while a class adversarial objective is employed to prevent fault-discriminative

information from dominating condition-sensitive representations. This design enables domain-specific characteristics to be consistently modeled in the student network, either with or without feature alignment via KD.

To achieve robust IFD across different domains, including unseen target domains, transferable fault-discriminative representations are learned by incorporating a domain adversarial objective. A domain discriminator is employed to suppress domain-related sensitivity in the fault-relevant features, thereby enhancing their invariance and generalization capability. Through the coordinated use of contrastive learning, knowledge distillation, and adversarial regulation, the proposed DIS framework models condition-sensitive variations and regulates their influence on fault-discriminative learning. The specific settings utilized in the proposed DIS method are listed in Table 3.1, where N_c and N_d denote the number of classes and the number of known source domains. The settings listed here are consistent with the designed modules in Fig. 5-3.

Table 5.1. Detailed network settings for the proposed DIS method.

Modules	Layers	Settings
Feature extractors $h_{f,tea}, h_{f,stu}$	1-Conv	Conv(1, 16, 64), BN, ReLU, Maxpool(2)
	2-Conv	Conv(16, 32, 16), BN, ReLU, Maxpool(2)
	3-Conv	Conv(32, 64, 3), BN, ReLU, Maxpool(2)
	4-Conv	Conv(64, 64, 3), BN, ReLU, AdaptiveMaxpool(4), Flatten
Bottleneck $h_{b,tea}$	1-FC	Linear(256, 128), ReLU, Dropout
Bottleneck $h_{b,stu}$	1-FC	Linear(256, 256)
Domain classifier h_{d-cls}	1-FC	Linear(128, N_d)
Class discriminator h_{c-adv}	1-FC	Linear(128, 256), BN
	2-FC	Linear(256, 256), BN
	3-FC	Linear(256, N_c)
Classifier h_{cls}	1-FC	Linear(128, N_c)
Domain discriminator h_{d-adv}	1-FC	Linear(128, 256), BN
	2-FC	Linear(256, 256), BN
	3-FC	Linear(256, N_d)

The final optimization objectives of the proposed DIS model can be summarized as follows:

$$\mathcal{L}_{DSR,tea}\left(\theta_{h_{f,tea}},\theta_{h_{b,tea}}\right)=\mathcal{L}_{CL}\left(\theta_{h_{f,tea}},\theta_{h_{b,tea}}\right) \tag{5-14}$$

$$\begin{aligned}&\mathcal{L}_{DSR,stu}\left(\theta_{h_{f,stu}},\theta_{h_{b,stu}},\theta_{h_{d-cls}},\theta_{h_{c-adv}},\theta_{h_{cls}},\theta_{h_{d-adv}}\right)\\&=\mathcal{L}_{mse}\left(\theta_{h_{f,stu}},\theta_{h_{b,stu}}\right)+\mathcal{L}_{d-cls}\left(\theta_{h_{f,stu}},\theta_{h_{b,stu}},\theta_{h_{d-cls}}\right)+\mathcal{L}_{cls}\left(\theta_{h_{f,stu}},\theta_{h_{b,stu}},\theta_{h_{cls}}\right)\\&+\lambda_1\left(\mathcal{L}_{c-adv}\left(\theta_{h_{f,stu}},\theta_{h_{b,stu}},\theta_{h_{c-adv}}\right)+\mathcal{L}_{d-adv}\left(\theta_{h_{f,stu}},\theta_{h_{b,stu}},\theta_{h_{d-adv}}\right)\right)+\lambda_2\mathcal{L}_{cos}\left(\theta_{h_{f,stu}},\theta_{h_{b,stu}}\right)\end{aligned} \tag{5-15}$$

where λ_1 and λ_2 are two hyperparameters used to balance the losses. Eq. (5-14) refers to the training of the teacher network and Eq. (5-15) guides the training of the student network following a 2-step knowledge distillation. The Adam optimizer is utilized to update the network parameters. The parameters of the neural network in the student network are updated as follows:

$$\theta_{h_{f,stu}}^{q} \leftarrow \theta_{h_{f,stu}}^{q-1} - \mu\left(\frac{\partial \mathcal{L}_{mse}}{\partial \theta_{h_{f,stu}}^{q-1}} + \frac{\partial \mathcal{L}_{d-cls}}{\partial \theta_{h_{f,stu}}^{q-1}} + \frac{\partial \mathcal{L}_{cls}}{\partial \theta_{h_{f,stu}}^{q-1}} + \lambda_1\left(\frac{\partial \mathcal{L}_{c-adv}}{\partial \theta_{h_{f,stu}}^{q-1}} + \frac{\partial \mathcal{L}_{d-adv}}{\partial \theta_{h_{f,stu}}^{q-1}}\right) + \lambda_2 \frac{\partial \mathcal{L}_{cos}}{\partial \theta_{h_{f,stu}}^{q-1}}\right) \tag{5-16}$$

$$\theta_{h_{b,stu}}^{q} \leftarrow \theta_{h_{b,stu}}^{q-1} - \mu\left(\frac{\partial \mathcal{L}_{mse}}{\partial \theta_{h_{b,stu}}^{q-1}} + \frac{\partial \mathcal{L}_{d-cls}}{\partial \theta_{h_{b,stu}}^{q-1}} + \frac{\partial \mathcal{L}_{cls}}{\partial \theta_{h_{b,stu}}^{q-1}} + \lambda_1\left(\frac{\partial \mathcal{L}_{c-adv}}{\partial \theta_{h_{b,stu}}^{q-1}} + \frac{\partial \mathcal{L}_{d-adv}}{\partial \theta_{h_{b,stu}}^{q-1}}\right) + \lambda_2 \frac{\partial \mathcal{L}_{cos}}{\partial \theta_{h_{b,stu}}^{q-1}}\right) \tag{5-17}$$

$$\theta_{h_{d-cls}}^{q} \leftarrow \theta_{h_{d-cls}}^{q-1} - \mu\left(\frac{\partial \mathcal{L}_{d-cls}}{\partial \theta_{h_{d-cls}}^{q-1}}\right), \theta_{h_{c-adv}}^{q} \leftarrow \theta_{h_{c-adv}}^{q-1} - \mu\left(\lambda_1\left(\frac{\partial \mathcal{L}_{c-adv}}{\partial \theta_{h_{c-adv}}^{q-1}}\right)\right) \tag{5-18}$$

$$\theta_{h_{cls}}^{q} \leftarrow \theta_{h_{cls}}^{q-1} - \mu\left(\frac{\partial \mathcal{L}_{cls}}{\partial \theta_{h_{cls}}^{q-1}}\right), \theta_{h_{d-adv}}^{q} \leftarrow \theta_{h_{d-adv}}^{q-1} - \mu\left(\lambda_1\left(\frac{\partial \mathcal{L}_{d-adv}}{\partial \theta_{h_{d-adv}}^{q-1}}\right)\right) \tag{5-19}$$

where μ is the learning rate. Pseudo code for training and inference of the proposed DIS method are detailed in Algorithm 5.1.

Algorithm 5.1: Training procedure of the proposed DIS method

Input: Multiple source domain datasets $\mathcal{D}_S = \left\{\mathcal{D}_S^1, \mathcal{D}_S^2, ..., \mathcal{D}_S^k\right\}$.

Initialization: The initialized parameters of the teacher network $\theta_{f,tea}$ and $\theta_{b,tea}$, and the student network $\theta_{f,stu}$, $\theta_{b,stu}$, $\theta_{h_{cls}}$, $\theta_{h_{d-cls}}$, $\theta_{h_{c-adv}}$ and $\theta_{h_{d-adv}}$. Setting learning rate μ, number of maximum epochs (N_{epoch}), batch size, manually selected trade-off parameters λ_1 and λ_2.

Training stage:

Teacher network: condition-sensitive representation learning

for *epoch* = 1 to N_{epoch} **do**

1. Randomly sample mini-batches from $\mathcal{D}_S$.
2. Forward propagation to extract features h_b^{tea}, compute the domain supervised contrastive loss following Eqs. (5-3)-(5-5).
3. Update $\theta_{f,tea}$ and $\theta_{b,tea}$.

end for

Return: $h_{f,tea}, h_{b,tea}$.

Student network: interference-regulated representation learning

for *epoch* = 1 to N_{epoch} **do**

4. Randomly sample mini-batches from $\mathcal{D}_S$.
5. Forward propagation to extract shared features by the feature extractor $h_{f,stu}$ and bottleneck $h_{b,stu}$.
6. Designate condition-sensitive features $f_{spec,stu}$ and fault-discriminative features $f_{inv,stu}$ following Eqs. (5-11) and (5-12).
7. Align condition-sensitive representations $f_{spec,tea}$ and $f_{spec,stu}$ via KD by minimizing $\mathcal{L}_{mse}$ following Eq. (5-6).

8. Enhance condition-sensitive modeling using domain classification and class-adversarial learning following Eq. (5-7);
9. Learn transferable fault-discriminative representations $f_{inv,stu}$ via domain adversarial learning following Eqs. (5-9) and (5-10).
10. Regulate the interaction between the domain-specific and invariant features by minimizing the cosine similarity loss $\mathcal{L}_{cos}$ following Eq. (5-13).
11. Update $h_{f,stu}, h_{b,stu}, h_{d-cls}, h_{c-adv}, h_{cls}, h_{d-adv}$ via backward propagation.

end for

Return: $h_{f,stu}, h_{b,stu}, h_{cls}$.

Inference stage:

Input: Unseen target domain dataset $\mathcal{D}_T$.

Model: Trained student network with optimal $h_{f,stu}, h_{b,stu}, h_{cls}$ without the domain discriminator $\theta_{h_{d-adv}}$.

Output: Predicted health states of the unseen target domain samples.

Note: The detailed settings of the modules listed in Algorithm 5.1 are illustrated in Fig. 5-3 and summarized in Table 5.1.

5.5 Experimental study

In this section, experiments are conducted on two bearing datasets to demonstrate the effectiveness of the proposed DIS method. It is worth noting that testing samples are not available during training.

5.5.1 Dataset description

5.5.1.1 UO bearing dataset

Fig. 5-4 presents the UO bearing test bench [51]. A SpectraQuest fault simulator (MFS-PK5M) is used to conduct the experiments. Here, the test bearing is on the right side. There is an accelerometer mounted close to the test bearing to record vibration data. Five artificially created health states are tested, including healthy, ball fault, inner race fault, outer race fault, and combined faults.

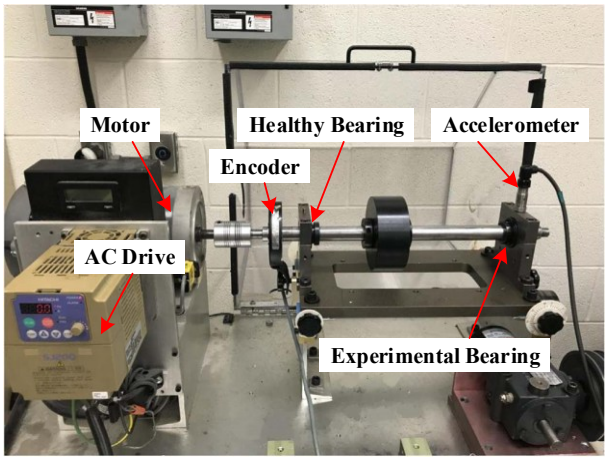


Fig. 5-4. UO bearing dataset test rig [51].

Data collection is operated under four different time-varying speed conditions. And each speed condition is considered as a single domain. Detailed domain settings of the UO bearing dataset are listed in Table 5.2.

Table 5.2. Domains of the UO bearing dataset.

Domain	Speed condition	Health states	Sample number
A	Increasing	Healthy (H)	5×600×4096
B	Decreasing	Ball fault (B)	5×600×4096
C	Increasing then decreasing	Inner race fault (I) Outer race fault (O)	5×600×4096
D	Decreasing then increasing	Combined faults (C)	5×600×4096

The target domain data does not contribute to the training, which makes it a domain generalization problem from known source domains to the unseen target domain. Four tasks can be built for the UO bearing dataset, including: TA, TB, TC and TD, as shown in Table 5.3. Here, three known source domains are used to form a known source domain dataset and then the model is trained, while the unknown target domain is unseen during the training procedure. The average accuracy of the three independent trials is taken as the diagnostic result of each task to avoid contingency in the experiment.

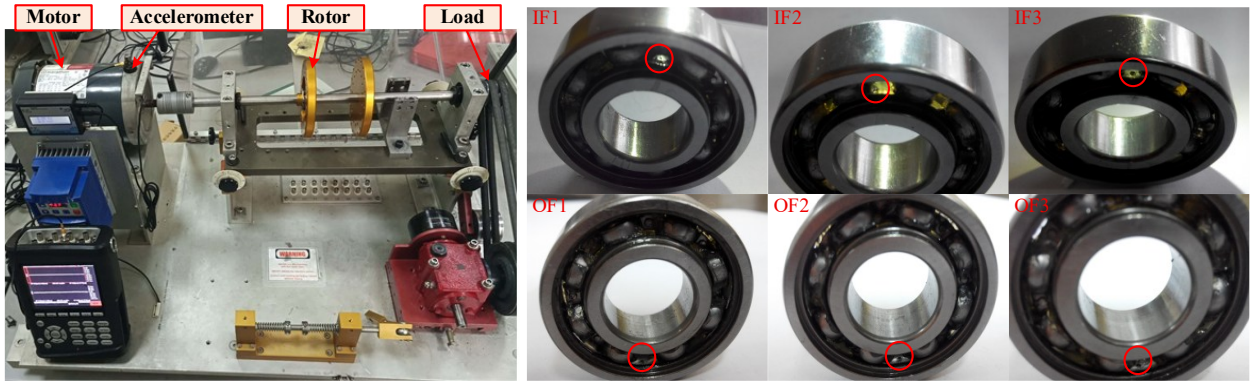
Table 5.3. Fault diagnosis tasks built on the UO bearing dataset.

Task	Known source domains	Unknown target domain
TA	B, C, D	A
TB	A, C, D	B
TC	A, B, D	C
TD	A, B, C	D

5.5.1.2 SQV bearing dataset

Fig. 5-5 (a) shows the test bench of the SQV bearing dataset from Xi'an Jiaotong University

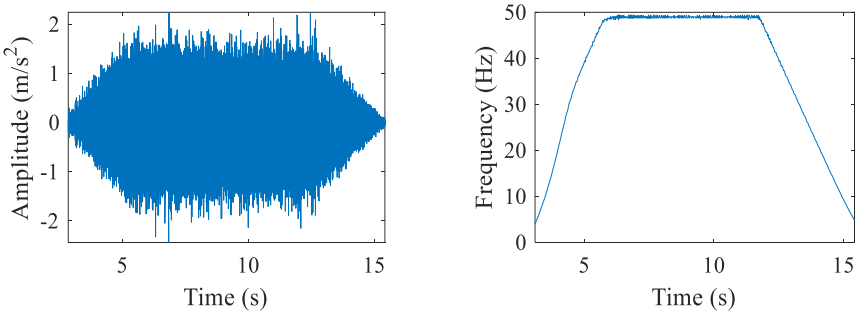
[52]. There are a total of 7 different health states in the SQV dataset, including healthy (H), three inner race faults (IF1, IF2, and IF3) and three outer race faults (OF1, OF2, and OF3). Fig. 5-5 (b) gives the images of the different faulty bearings, where the local faults are marked by red circles. The SQV dataset is collected using a sampling frequency of 25.6 kHz.



(a) (b)

Fig. 5-5. SQV bearing dataset from XJTU [52]: (a) test rig, (b) illustrations of fault simulation bearings.

An example of how the rotation speed changes is plotted in Fig. 5-6. It can be found that the amplitude is almost proportional to the rotation speed versus time. However, to ensure that the collected vibration signals are relatively stable without suffering from too much randomness during low speeds, preprocessing is used here when preparing the dataset. To be specific, by measuring the rotation speed, some signal segments typically within the first and the last few seconds are filtered out, so that the rotation speed (frequency) plotted in Fig. 5-6 (b) does not start from 0 Hz.



(a) signal waveform (b) rotation speed versus time

Fig. 5-6. SQV bearing data example: (a) filtered signal waveform, (b) corresponding rotation speed.

Data collection was repeated six times with different time lengths (varying time during increasing, constant and decreasing speeds). Detailed settings for the SQV dataset are summarized

in Table 5.5. 100 samples of size 3200 are generated for each health state, and an overlap of 512 data points is used here to make sure the selected samples focus on the time-varying speeds. Similar to the UO bearing dataset, another six fault diagnostic tasks recorded from T1 to T6 are conducted on the SQV dataset. For instance, in the T1 task, domain 1 (trial 1) acts as the unseen target domain and is not involved during the model training procedure.

Table 5.4. Domains created from the SQV bearing dataset.

Domains	Speeds	Health states	Sample number
[1, 2, 3, 4, 5, 6]	Increasing, then stable and finally decreasing	H, IF1, IF2, IF3, OF1, OF2, OF3	7×100×3200

5.5.2 Comparison methods

To verify the effectiveness of the proposed DIS method, five DG-based fault diagnosis methods are used for comparison, as listed in Table 5.5. These include metric learning strategies such as MMD (M1) and CORAL (M2) [49] [53]. Since adversarial learning is embedded in the proposed method, thus DANN (M3) is compared as a baseline study [45]. IEDGNet (M4) is a classical DG method that uses a triplet loss function and a data augmentation strategy [71]. DIFE (M5) is a recently published DG method that only focuses on characterizing domain-invariant features without considering domain-specific features [92]. M6 is the complete version of the proposed DIS method. To reduce randomness and ensure the reliability of results, each diagnostic task was repeated 3 times, and the average result was recorded for later analysis. For a fair comparison, all comparison methods share a similar network structure and experimental settings.

Table 5.5. Methods used for comparison.

Methods	Description
M1	Metric learning (MMD) [49]
M2	Metric learning (CORAL) [53]
M3	Domain adversarial learning (DANN) [73]
M4	Triplet loss and data augmentation (IEDGNet) [71]
M5	Intra- and inter-domain-invariant features only (DIFE) [92]
M6	Proposed DIS method

The hyperparameter settings used in the proposed DIS method are summarized in Table 5.6. The listed parameters are used for all compared methods to ensure a fair comparison. The selection of λ_1, λ_2 in the proposed DIS method is discussed later in Section 5.6.2, where a grid searching strategy is conducted to find optimal values.

Table 5.6. Hyperparameter settings.

Hyperparameter	Learning rate μ	Batch size	Weight decay	Max epoch	λ_1, λ_2
Value	0.01	32	0.0005	120	{0.1, 0.01}

5.5.3 Accuracy results

5.5.3.1 Accuracy results on the UO dataset

The average accuracies as well as their standard errors on the UO dataset are recorded in Table 5.7. The average accuracy using the proposed method (M6) is highest, achieving 98.96 % by obtaining three highest and one second-best (by only a very small margin) accuracies, showing that the proposed DIS method generalizes well on unseen target domain data. Here, the UO-TA task is the most challenging one since all other compared methods only get about 80 % or less accuracy. By exploring and then removing domain-specific features, it can be concluded that the proposed DIS method can generalize well on unseen target data so that improved accuracy can be acquired, even compared to state-of-the-art methods.

Table 5.7. Accuracy results obtained on the UO bearing dataset (%).

Methods	TA	TB	TC	TD	Average
M1	79.94±0.10	80.46±5.63	91.22±5.52	90.37±2.29	85.48±6.16
M2	79.94±0.06	84.71±2.39	88.89±5.38	90.04±5.65	85.90±4.59
M3	79.83±0.09	94.13±0.47	99.53±0.70	95.35±0.37	92.21±8.57
M4	74.95±1.12	85.45±4.87	86.52±3.01	90.43±4.96	84.34±6.62
M5	86.23±0.88	**99.28±0.64**	**100±0**	98.87±0.15	96.10±6.59
M6	**98.57±0.31**	99.23±0.52	**100±0**	**99.02±0.17**	**98.96±0.77**

5.5.3.2 Accuracy results on the SQV dataset

Another three independent trials are conducted when testing the SQV bearing dataset. Testing results on the SQV dataset are listed in Table 5.8. After comparing the average accuracies, it can be found that the proposed DIS method (M6) achieves the highest accuracy of 90.59 %. The second-best method is M5 with an accuracy of 89.77 %, providing similar results. By comparing, it can be concluded that the proposed method can provide similar or better accuracy results when inferring unseen target domain data, showing the effectiveness of the proposed DIS method for IFD when the target domain is unseen.

Table 5.8. Accuracy results obtained on the SQV bearing dataset (%).

Methods	T1	T2	T3	T4	T5	T6	Average
M1	85.33±2.44	88.52±5.16	85.43±2.75	79.24±5.11	80.14±6.53	78.81±5.99	82.91±4.04
M2	83.52±3.76	83.00±3.47	82.10±4.37	77.33±6.54	77.67±6.86	75.93±5.77	79.93±3.31
M3	88.29±2.44	87.43±1.50	86.71±2.81	80.19±4.65	83.05±4.87	77.95±5.58	83.94±4.23
M4	89.00±1.85	87.24±1.95	86.71±1.49	79.10±5.26	79.90±4.71	77.43±4.60	83.23±4.97
M5	92.95±0.30	91.95±1.43	91.90±1.37	88.19±2.42	87.43±2.68	86.19±2.00	89.77±2.84
M6	**93.05±1.07**	**93.47±0.30**	**91.90±1.04**	**88.71±1.76**	**89.19±1.95**	**87.24±0.72**	**90.59±2.56**

Confusion matrices from the SQV-T6 task are provided in Fig. 5-7 to figure out which kind of target samples are misidentified, where M1-M6 are all included for a detailed comparison. It can be found that IF3 is the most challenging task, and its number is the smallest among all 7 categories. The proposed DIS method obtains the highest accuracy when inferring samples with labels of IF2, OF1, and OF2, which also leads to the highest average accuracy (90.59 %).

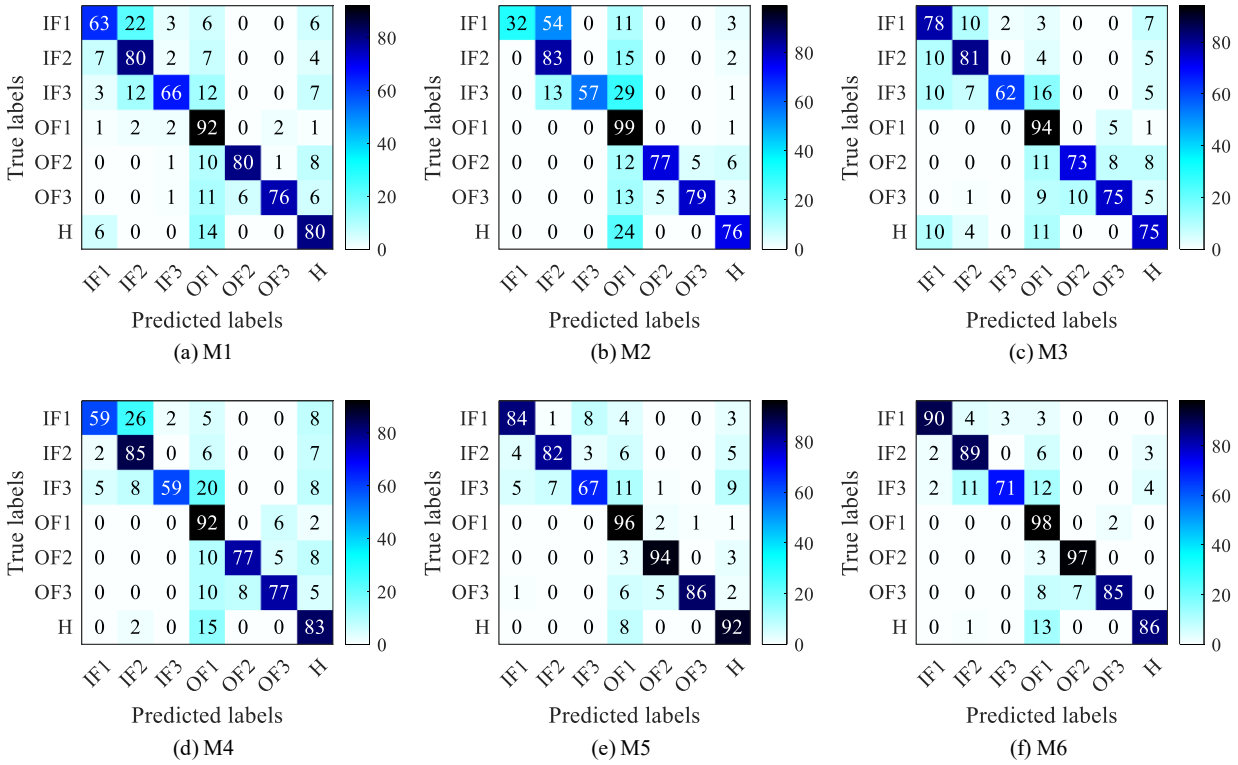


Fig. 5-7. Confusion matrices on the SQV-T6 task.

5.5.4 Feature visualization

Feature visualization results help in understanding how the data forms clusters after training, rather than just focusing on the accuracy results. The SQV-T6 task is analyzed by plotting all feature distribution results by using t-SNE. In the compared methods, the learned features after the bottleneck layers are used. However, since the fault diagnosis task is more important, only domain-

invariant features in the student network are analyzed for the proposed M6 method. All methods mentioned in Section 5.5.2 are implemented for comparison. The corresponding results are shown in Fig. 5-8 (a)-(f). In the feature visualization results provided by using M1-M4, it can be seen that clusters corresponding to IF1 and IF2 are mixed together, which leads to misidentified fault types, as illustrated by samples in diamonds and circles. In Fig. 5-8 (e), M5 provides a relatively clear decision boundary. However, it fails to accurately distinguish samples between IF1, IF2, IF3, and OF1. This observation is also consistent with the given confusion matrix presented in Fig. 5-7, where noticeable misclassifications occur among these fault categories. In contrast, it can be found that the proposed M6 method provides the clearest inter-class separation. A well-defined separation can be observed in the central region of the feature space, with only a small number of IF3 samples exhibiting partial overlap with other fault types, as shown in Fig. 5-8 (f).

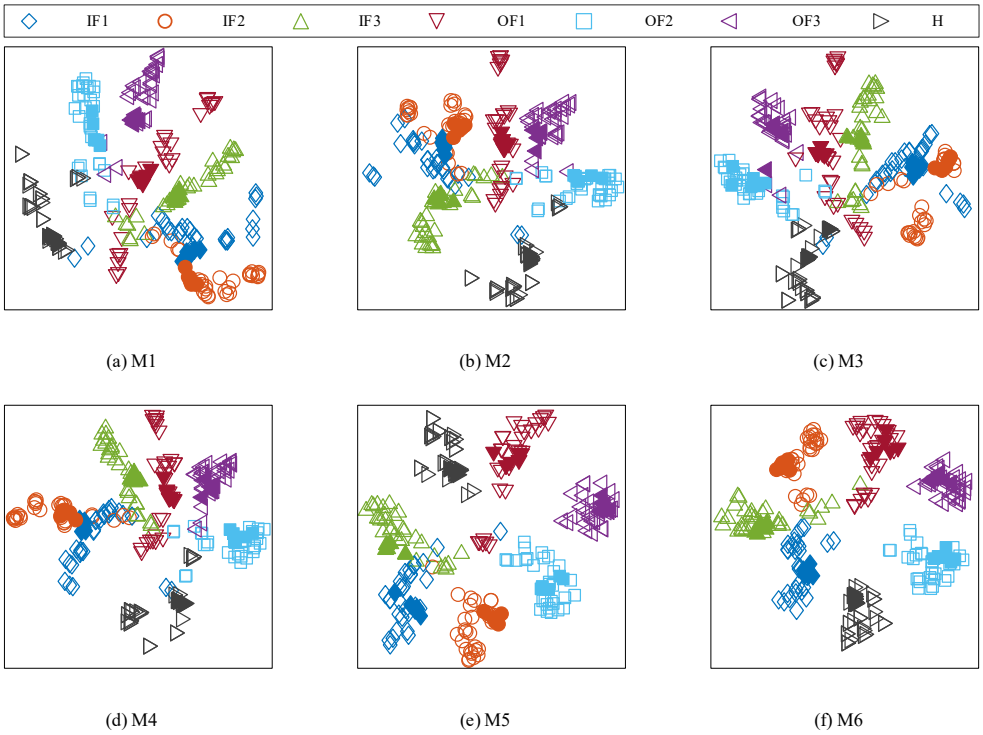


Fig. 5-8. Feature visualization results for the SQV-T6 task (samples from the unseen target domain are plotted using solid markers).

5.6 Discussion

5.6.1 Ablation study

To verify the effectiveness of the objective functions used in the proposed DIS method, ablation experiments are implemented. The final training objective consists of six individual loss terms that jointly regulate feature learning in both the teacher and student networks, referring to Eq. (5-15). To isolate the contribution of the key components, a total of 4 representative settings

are evaluated, as summarized in Table 5.9.

In the first ablation setting (A1), the domain-specific features learning mechanism in the teacher network is disabled (i.e., by removing $\mathcal{L}_{mse}$ only). As a result, the teacher network no longer provides informative domain-specific guidance, and KF does not contribute to the optimization of the student network. Under this setting, both domain-specific and invariant features in the student network are learned via a step-by-step procedure after parameter random initialization instead, without supervision from the teacher. In the second configuration (A2), the student network is not designed to learn domain-specific features directly, while domain-specific features are only learned in the teacher network. Although the divergence between invariant and specific features is still considered, the lack of domain-specific feature representation in the student network limits the effectiveness of feature decoupling. In the third ablation setting (A3), the cosine-similarity-based suppression term is removed. In this case, dissimilarities between the learned invariant and specific features are no longer enforced, which may lead to potential geometric overlap and residual domain interference between extracted specific and invariant features. Then, to investigate the effect of different independent metrics, an additional ablation study setting is tested, denoted as A4, in which the cosine-based disentanglement constraint is replaced by the Hilbert-Schmidt independence criterion (HSIC) [93]. In this setting, the domain-invariant and domain-specific representations are regulated by minimizing their statistical dependence at the batch level. Notably, the HSIC regulation is applied only to the student network and does not involve the teacher network. In addition, a new ablation setting, denoted as A5, is introduced to evaluate the effectiveness of HSIC under the complete DIS framework. Compared with A4, A5 retains the complete KD architecture and all other loss components but replaces the cosine-based regulation constraint with HSIC. The full DIS model, denoted as M6, integrates all proposed components and serves as the reference configuration.

Table 5.9. Ablation experimental settings.

Methods	$\mathcal{L}_{cls}$	$\mathcal{L}_{mse}$	$\mathcal{L}_{d-cls}$	$\mathcal{L}_{c-adv}$	$\mathcal{L}_{d-adv}$	$\mathcal{L}_{cos}$	**Note**
A1	✓	✗	✓	✓	✓	✓	w/o teacher network
A2	✓	✓	✗	✗	✓	✓	w/o student network
A3	✓	✓	✓	✓	✓	✗	w/o regulation
A4	✓	✗	✓	✓	✓	✓	w/o teacher network, HSIC-based regulation
A5	✓	✓	✓	✓	✓	✓	HSIC-based regulation
M6	✓	✓	✓	✓	✓	✓	Proposed DIS method

Table 5.10 summarizes the classification accuracy results of all ablation settings on the SQV dataset. Overall, A1-A4 achieve performance improvements compared with the baseline methods (e.g., M1-M4) in Table 5.10, indicating that introducing domain-specific learning, adversarial

learning, or alternative independence regulation is beneficial to some extent. However, within the ablation study, these four settings exhibit comparable performance and constitute the lowest performance tier, suggesting that partially removing or independently modifying individual components leads to limited gains.

In contrast, A5 consistently achieves the second-best performance across tasks, demonstrating that introducing HSIC-based independence regulation into the complete teacher-student framework can noticeably improve generalization performance. Nevertheless, M6 consistently outperforms all ablated variants and achieves the highest accuracy on all tasks, confirming that the proposed cosine-based feature decoupling strategy is more effective than HSIC when jointly optimized with the full DIS framework.

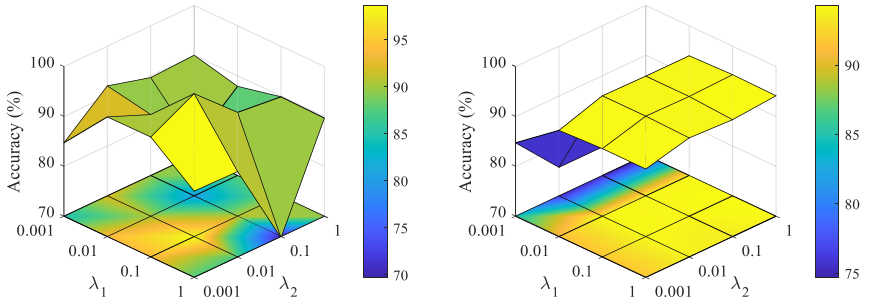
Table 5.10. Accuracy results of the ablation studies on the SQV dataset.

Methods	T1	T2	T3	T4	T5	T6	Average
A1	92.24±1.53	91.53±1.08	91.67±1.07	87.62±3.08	84.95±1.46	85.86±1.55	88.98±3.23
A2	92.86±1.36	91.62±1.41	90.76±1.32	88.05±1.32	86.71±4.52	85.48±2.37	89.25±2.93
A3	92.95±0.92	92.14±1.74	92.00±0.71	86.76±2.45	85.62±1.86	86.05±1.80	89.25±3.44
A4	92.76±1.21	92.00±0.50	91.52±0.59	85.76±1.90	86.90±1.44	85.81±1.35	89.13±3.30
A5	92.91±1.22	92.47±1.89	91.71±0.90	88.05±2.28	86.90±2.79	85.71±2.34	89.63±3.11
M6	**93.05±1.07**	**93.47±0.30**	**91.90±1.04**	**88.71±1.76**	**89.19±1.95**	**87.24±0.72**	**90.59±2.56**

5.6.2 Sensitivity analysis

The proposed framework involves two hyperparameters, λ_1 and λ_2, which regulate different but complementary mechanisms for mitigating domain interference. Specifically, λ_1 is a hyperparameter used in adversarial learning to encourage domain confusion, while λ_2 is a weighted cosine-similarity-based constraint that directly suppresses the correlation between domain-invariant and domain-specific features.

A sensitivity analysis over a wide logarithmic range {0.001, 0.01, 0.1, 1} is conducted to evaluate the robustness of the proposed method, which helps clarify the relative contributions of adversarial alignment and cosine-based domain suppression to the overall generalization performance. Fig. 5-9 shows the diagnostic accuracy under different parameter combinations on the UO-TA and SQV-T1 tasks. It can be found that the proposed method exhibits sensitivity on the UO-TA task, where the optimal accuracy is obtained at $\{\lambda_1=0.1, \lambda_2=0.01\}$. This behavior may suggest that UO-TA involves stronger domain shifts, as reflected by the relatively low accuracies (approximately 80 % reported by most compared methods), thereby making the balance between adversarial alignment and feature disentanglement more critical.



(a) UO-TA (b) SQV-T1

Fig. 5-9. Sensitivity analysis based on the used trade-off parameters on the two bearing datasets.

In contrast, performance on the SQV-T1 task remains relatively stable across a wide range of parameter settings, indicating that the proposed method is less sensitive to hyperparameter variations on this task. This robustness implies that learned representations are already aligned well and effectively disentangled, such that moderate changes in regulation do not significantly affect the final performance.

5.6.3 Limited training samples

In this subsection, the performance of the proposed method is further tested by using limited samples. The UO-TC task is analyzed for comparison. Test accuracies are recorded based on the model's prediction of all 3000 samples in the unseen target domain. Also, to avoid randomness, the random seed is set to 0 to ensure that all methods adopt the same data stream.

For the limited sample numbers, a variety of different percentages are used, and their test accuracy results are listed in Table 5.11. As expected, accuracy increases when more samples are involved during training, indicating that sufficient data samples ensure model convergence, and vice versa.

Table 5.11. Accuracies obtained on the UO-TC task with limited training samples.

Methods	**1 %**	**2 %**	**5 %**	**10 %**	**20 %**	**50 %**	**100 %**
M1	44.43	55.03	78.00	90.57	91.93	96.13	96.29
M2	40.33	40.67	44.43	57.10	73.03	88.07	90.33
M3	74.67	91.37	96.07	98.87	99.73	99.87	99.93
M4	38.70	42.00	90.83	94.83	93.97	97.33	98.03
M5	93.27	**99.87**	99.93	99.93	**100**	**100**	**100**
M6	**96.87**	99.77	**100**	**100**	**100**	**100**	**100**

By comparing M6 with other methods, it can be concluded that the proposed DIS method achieves the highest accuracy under all percentages of training samples in almost all cases (only exception occurs when using 2 %: 99.87 % when using M5 versus 99.77 % when using the proposed DIS method). This indicates that the proposed DIS method outperforms other

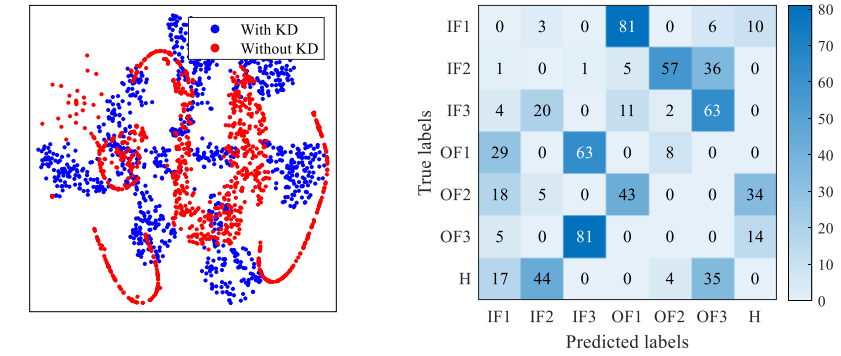
comparison methods even when only a small number of training samples are used (i.e., 1 % and 2 %).

5.6.4 Domain-specific feature interpretation

5.6.4.1 Domain-specific features

In the proposed method, domain-specific features are not treated as noise to be completely discarded. Rather, they are regarded as structured interference factors that coexist with domain-invariant fault representations in a shared latent space. Accordingly, the objective is not to eliminate domain-specific features, but to prevent them from contaminating the fault-discriminative subspace. This consideration motivates the need for a mechanism that regulates the interaction between domain-invariant and domain-specific representations at the feature level.

It is also worth noting that the teacher network in the KD framework is intentionally trained without domain interference suppression, and thus, may still encode entangled domain-specific and invariant information. Accordingly, the teacher does not serve as a perfect target representation, but rather as a weak structural prior during early training. All disentanglement constraints are imposed exclusively on the student network, which is deployed during the inference stage. As illustrated in Fig. 5-10 (a), without KD, the specific representations in the student network degenerate into low-dimensional manifolds, indicating a form of representational collapse. In contrast, KD stabilizes the early training dynamics and preserves high-dimensional, structured representations in the student, despite the teacher itself not being regulated.



(a) effect of KD on the learned specific features (b) predicted classes based on the learned specific features

Fig. 5-10. Domain-specific features analysis.

By design, domain-specific features learned in the student network are expected to be distinct from domain-invariant features and should not encode useful fault-discriminative information when performing IFD. To verify this assumption, learned domain-specific features are fed into the classifier h_{cls} for health state prediction. The resulting confusion matrix is given in Fig. 5-10 (b), where an accuracy of 0 % is obtained. This result confirms that the domain-specific representations

do not contain useful fault-discriminative information, indicating the effectiveness of the proposed disentanglement strategy.

5.6.4.2 Cosine similarity

To quantitatively characterize the interaction between domain-invariant and specific features, cosine similarity is adopted as a geometric measure of their alignment in the shared latent space. Cosine similarity reflects the angular relationship between two feature vectors: a cosine similarity close to zero indicates near-orthogonality, suggesting that the two representations encode largely independent information, whereas large positive or negative values imply strong coupling.

Fig. 5-11 illustrates the distribution of cosine similarity under different configurations on the SQV-T1 task. When aligning multiple source domains, the cosine similarity among domain-invariant features yields a high mean value of 0.6737, which indicates that domain-invariant features learned across multiple known source domains are moderately to highly correlated. Furthermore, when measuring the similarity between invariant features extracted from the known source domains and those from the unseen target domain, the resulting distribution (shown in green) exhibits a mean value of 0.4837, despite the presence of domain shift between the known source domains and the unseen target domain. These results indicate that the learned invariant features capture domain-shared fault-related characteristics. In contrast, the cosine similarity between domain-invariant and domain-specific features is shown by the blue and purple distributions in Fig. 5-11 (a). In this case, the mean value μ is approximately 0, indicating effective geometric decoupling between the two types of representations. To demonstrate the effectiveness of the proposed regulation between domain-invariant and domain-specific features, the cosine similarity distribution of unseen target domain samples is shown in yellow, with a mean cosine similarity of -0.5894, suggesting a moderate-to-strong decorrelation between the extracted feature representations. To further evaluate the effectiveness of the proposed domain interference suppression across multiple domains, domain-wise cosine similarities between the learned invariant features and specific features are calculated (i.e., invariant features in T1 against specific features in T1, invariant features in T1 against specific features in S2, etc.). Results are illustrated in Fig. 5-11 (b). It can be observed that most cosine values approach 0, with a scale range of $[-0.1, 0.1]$, indicating that learned invariant features are geometrically decoupled from learned specific features.

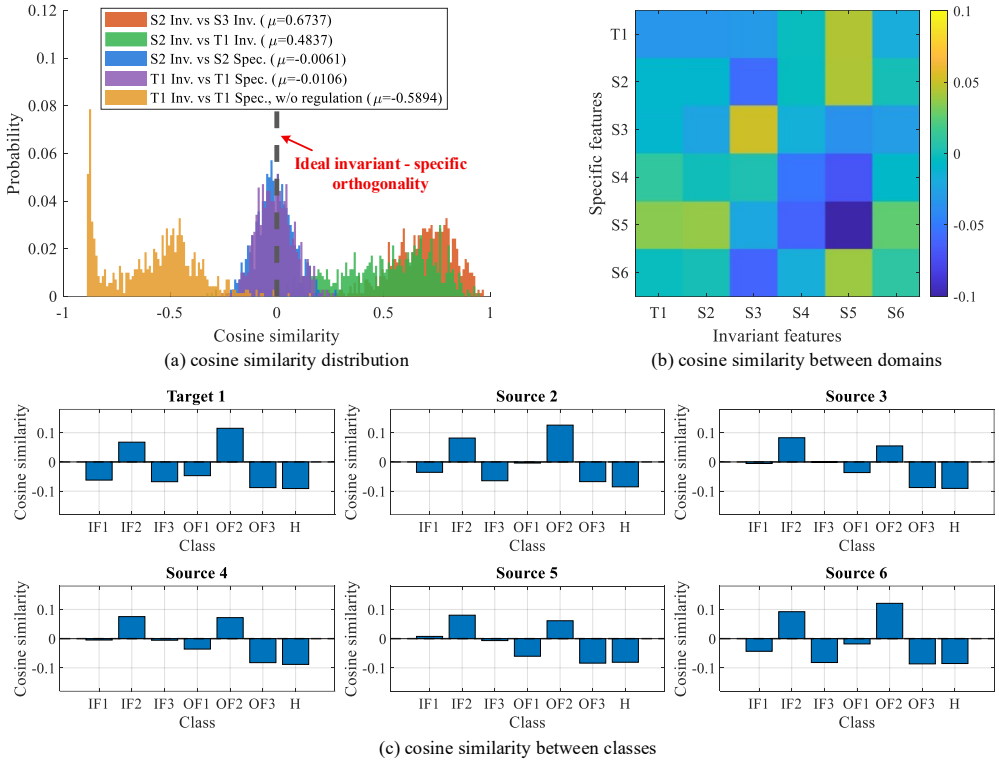


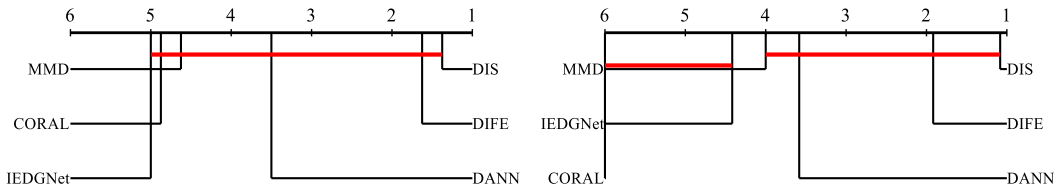
Fig. 5-11. Cosine similarity measured between the learned features on the SQV-T1 task.

At the same time, class-wise cosine similarities are reported in Fig. 5-11 (c), where the values are measured between the decoupled invariant and specific features in each class. It can be found that after introducing the proposed cosine-based suppression term, similarity distributions consistently concentrate around zero across different fault categories and across both source and target domains, illustrated by the narrow scale of [−0.1, 0.12]. This consistent near-orthogonality demonstrates that the proposed constraint effectively enforces geometric separation between domain-invariant and domain-specific features, thereby suppressing domain-induced interference in the learned representation. Additionally, by examining the changing pattern of the calculated cosine similarity calculated for different classes under different operating conditions, it can be seen that the proposed cosine-based decoupling constraint generalizes consistently to the unseen target domain at the representation level, maintaining similar geometric separation between invariant and specific features across both source and target domains.

5.6.5 Statistical validation

While the cosine similarity study provides representation level evidence of effective domain interference suppression, it remains necessary to examine whether such geometric decoupling leads to consistent performance across different tasks and operating conditions. In terms of diagnostic accuracy, the Friedman test statistic for the UO dataset is 15.58 with a corresponding

p-value of 8.15×10^{-3}; the Friedman test statistic for the SQV dataset is 27.16 with a p-value of 5.30×10^{-5}. These two p-values are lower than the threshold of 0.05, indicating there are significant differences between the proposed DIS method and the five comparison methods. Two critical difference diagrams are also plotted to examine the diagnostic performance on all 10 diagnostic tasks, as shown in Fig. 5-12 [94]. The smaller values obtained by using the proposed DIS method indicate superior model performance.



(a) UO (b) SQV

Fig. 5-12. Critical difference diagrams of the post hoc Friedman test.

5.6.6 Computational efficiency analysis

Table 5.12 summarizes the computational time and space complexities of different methods evaluated on the UO and SQV datasets. The reported floating-point operations (FLOPs) correspond to a single forward inference pass for each dataset. All compared methods adopt the same backbone architecture, ensuring a fair comparison in terms of computational cost.

The fault diagnosis task on the UO bearing dataset employs a longer input sequence length (4096) than that on the SQV dataset (3200), which leads to slightly higher FLOPs. As shown in Table 5.12, the FLOPs of all methods are highly comparable across both datasets, while the parameter counts differ marginally due to the adoption of a two-branch design following the KD strategy.

Table 5.12. Computational complexity comparison in terms of FLOPs and parameter counts.

Methods	FLOPs on the UO dataset (M)	FLOPs on the SQV dataset (M)	Parameters (k)
M1-M4	33.73	26.18	95.77
M5	33.79	26.25	162.07
M6	33.79	26.25	161.18

Note: All methods share the same backbone network, resulting in almost identical FLOPs.

5.7 Conclusion

In this chapter, a generalized fault diagnosis method is proposed from a DIS perspective to effectively extract transferable features across varying operating conditions. DIS models domain-specific information and regulates its influence to enhance domain-invariant fault representations. The proposed method adopts a two-stage teacher-student learning strategy, in which condition-

sensitive characteristics are first captured in a teacher network and subsequently transferred to a student network through KD. Feature-level interference is further suppressed by enforcing geometric decoupling between domain-specific and domain-invariant features using a cosine similarity-based regulation mechanism. The collaboration between KD and cosine-based feature regulation effectively suppresses domain-induced interference, thereby facilitating robust feature representations of domain-invariant fault characteristics. During inference, only domain-invariant features are involved, ensuring stable and reliable fault identification under seen conditions. Extensive experiments conducted on two public bearing datasets demonstrate that the proposed DIS method consistently outperforms state-of-the-art domain generalization approaches. In addition, DIS exhibits strong robustness under limited training sample settings, highlighting its practical applicability to real industrial scenarios with variable operating conditions.

Chapter 6 SSTFA Net: A deep learning-based self-supervised time-frequency analysis tool

This chapter addresses objective 4, where a self-supervised deep learning-based TFA framework is proposed to perform adaptive TF representation learning.

The content of this chapter will be submitted for publication.

6.1 Abstract

TFA is a powerful tool for revealing time-varying frequency characteristics in nonstationary signals. Unfortunately, many advanced methods that incorporate energy enhancement and TF reassignment still suffer from heavy TF smearing, especially when signal components are closely located or intersect. To address this, a deep learning-based end-to-end framework is proposed, termed self-supervised TFA (SSTFA). Instead of requiring ground-truth IF labels, SSTFA is trained using IFs estimated via an improved IF estimator, which integrates a chirp-rate operator and a group-delay estimator, and applies multiple stepwise reassignment operations to obtain clear TFRs and accurate IF estimations. There are two main modules in the proposed SSTFA framework: (1) the TF convolution module, and (2) the energy enhancement module. The TF convolution module is used to generate a series of coarse TFRs by transforming the original one-dimensional time series data to two-dimensional TF planes, and the energy enhancement module guides the network to learn feature representations around the frequency components of interest. Thus, sharper and clearer results closer to the ideal TFR can be generated. Finally, the performance of the proposed SSTFA model in representing time-varying features is verified by analyzing both simulated and real-world data.

6.2 Introduction

TFA has attracted attention from researchers and engineers worldwide, providing critical insights into how frequency components of a signal evolve over time [95]. However, classical TFA methods such as the STFT and the continuous wavelet transform (CWT) are fundamentally constrained by the Heisenberg uncertainty principle, which means that achieving high resolution in the time domain necessarily degrades resolution in the frequency domain, and vice versa [96]. In terms of TFA, the Heisenberg uncertainty principle states that the TF resolution of the resulting TFR of the signals is affected by adjusting the window length of a signal when creating a TFR. There is a special case when the frequency of a signal is constant, where the frequency trajectory is parallel to the time axis. Thus, more attention should be focused on the frequency domain to identify the exact frequency of interest. For instance, after mapping the information to the frequency domain by applying the FFT, the dominant frequency components that exist in the time domain signal can be clearly identified by searching for local peaks. However, for a signal with a time-varying speed, the local average amplitudes within a short time window will be recorded at each time instant, which will be a frequency range rather than a local peak [17]. To reveal the changing patterns of the frequency trajectories, more advanced TFA methods must be developed. However, for signals with rapidly changing or closely spaced frequency components, the resulting TFR will suffer from heavy TF smearing, as well as low energy concentration levels, which make it hard to identify their specific changing patterns over time [19].

To overcome this, post-processing acts as a potential solution by further reassigning the TF energy along the frequency of interest, which leads to a sharper TF ridge and better readability [17]. The key idea of such post-processing techniques is to find the true IF of the signal, which is also recorded as the IF estimator, making the energy reassignment completely self-supervised without prior knowledge of the true IFs [97,98]. Such an operation helps post-processing techniques to further improve the energy concentration level. It is also worth noting that energy reassignment is performed along the frequency axis, which retains the ability to reconstruct the signal, allowing the signal components of interest to be extracted and recovered from the resulting TFR as the transform is invertible. Such reconstruction is useful for signal decomposition, especially when the signal is contaminated by noise or interference. Selective reconstruction of the dominant components can effectively improve the SNR [99]. However, these methods also have their own limitations, as the TF energy caused by noise and interference would also be reassigned to the estimated IF, which causes lower TF amplitudes. These techniques are a two-step method that depends entirely on the original TFR generated by the STFT or the CWT. That is, reassignment only changes the energy concentration level based on the TFR via the STFT, while the resolution remains the same. If the analyzed signal is contaminated by heavy noise, the corresponding results after reassignment would suffer from heavy frequency oscillations and would fail to sharpen the TF ridges, as subsequent processing is still based on a TFR of poor quality, and the resolution is not further improved. Based on the above analysis, it can be found that post-processing techniques such as the SST and the SET that make the readability of varying frequency signals better have their own limitations. These are also two-step methods that are highly related to being able to find a reasonable window function before generating a TFR via the STFT or the CWT [18].

To better perform TFA even when the signal has IFs that are closely adjacent or intersect, more advanced methods should be proposed to deal with the low energy concentration level and cross-term interference in the resulting TFRs. A better TFR could be generated by prior knowledge or extra parameters. A typical example is the parameterized TFA, by modifying the changing pattern of the kernels, frequency components with the same changing pattern can be enhanced, and vice versa [100–102]. Similar to parametric methods, frequency modulation also leads to a higher TF concentration level compared to the original STFT. Then, combined with reassignment operations, both energy enhancement and sharp ridges can be obtained. This kind of method can also be considered as a hybrid method, as both modulation and reassignment are integrated together for TFA [103]. But it is worth noting that frequency modulation is also based on the assumption that the frequency modulation (FM) law of the analyzed signal can be acknowledged in advance. If an inaccurate IF is used, it will lead to the same TF smearing problem and even the wrong IF information. Requiring prior knowledge limits the use of many TFA methods, which are

clearly unsuitable for real-world applications.

With the development of recent advances in deep learning, convolutional neural networks have been widely studied. Some researchers have also studied how to use deep learning as a tool to perform TFA. For instance, Razzaq et al. utilized deep neural networks to estimate the instantaneous frequency, where the frequency components have linear changing patterns [104]. In seismic data analysis, some applications of TFA focus on isolating the source earthquake signal from the noisy components on the TFR plane [105]. There are also some applications for using deep learning to provide super-resolution TFRs, where a U-net was used to extract more detailed features in the TF plane [106]. However, these methods focus on learning the linear representations of the frequency components, which is a simplified case because real frequency components may have some unpredictable time-varying patterns. There have also been other attempts to modify the kernels. By defining a similar kernel function as the STFT or CWT, Chen et al. proposed an interpretable TFconv layer to extract fault-related TF information [107]. Additionally, Pan et al. and Zhao et al. trained an encoder-decoder network for TFA, where the true IFs are involved during the model training procedure [108,109].

Inspired by these studies, a new framework is presented in this chapter to further improve model performance when performing TFA and to avoid requiring the use of any useful prior knowledge. This framework is aimed at providing adaptive TFA results, where TFA is automatically implemented by a deep learning-based TFA model. Unlike the two steps needed in classical post-processing methods, the proposed method can provide an end-to-end analysis by involving two modules to transform the original one-dimensional time domain signal into a highly concentrated two-dimensional TFR. They are (1) a TF convolution module, and (2) an energy enhancement module. The first module serves as a learnable layer to generate a series of coarse TFRs across multiple channels using a complex exponential kernel function. By doing so, the original one-dimensional time domain data are transformed into a two-dimensional TF plane. During training, the weights of this layer are guided to form multiple groups of complete basis functions. Unlike the STFT (that is restricted to a fixed Fourier basis), the proposed SSTFA model learns a diverse set of kernels (that can also be understood as channels), which are widely used in deep learning since deep learning can learn deeper features using more channels. This allows the module to extract a much richer set of TF features from the input signal, producing a set of coarse TFRs that contain more information than the original STFT. The energy enhancement module takes the coarse TFRs generated in the first module and performs SST-like TF energy reassignment to produce a final TFR with sharpened TF ridges closer to ideal TFRs. To realize this, a residual encoder-decoder network is constructed, allowing the model to learn frequency representations through deep layers. The input will be the coarse TFRs generated through the TF convolution

module, usually with a pre-determined size in the frequency domain. Finally, by automatically adjusting the weights of each component, a series of multi-component signals is generated so that numerous data as well as their precise IF labels can be used for model training. This helps improve the model's performance, enabling it to identify weak components and generalize well to new data. The main contributions of the proposed method can be summarized as follows:

(1) An improved IF estimator is discussed, which achieves enhanced energy concentration in the TF ridges after performing multiple energy reassignment operations. The improved IF estimator integrates both a chirp-rate operator and a group delay estimator to progressively refine IF estimation results in a stepwise manner. Compared with the conventional SST or MSST, this design leads to sharper TF ridges without TF smearing problems when the frequency is rapidly changing.

(2) A self-supervised deep learning-based TFA model (SSTFA) is introduced, enabling adaptive and data-driven TFA without prior knowledge and human intervention. The model uses a TF convolution module to map the time series data to a joint TF plane. Subsequently, an encoder-decoder structure is applied and the TF features are learned through deep layers.

(3) A multi-component weighting strategy for weak feature enhancement based on the superposition theorem is proposed, which not only enables accurate IF estimation for closely located frequencies but also achieves weak feature enhancement. By automatically adjusting the amplitude weights of each component, a diverse set of multi-component signals can be generated. This enables large-scale training data to be formed in a controllable manner, supporting accurate IF estimation and effective enhancement of weak or low-energy features.

The remainder of this chapter is laid out as follows. In Section 6.3.1, an introduction to TFA is first given. Then, an example of the improved IF estimator is illustrated to show superiority when it comes to rapidly changing frequencies. The IF estimation is also mathematically verified after employing multiple reassignment operations. Then, the model is analyzed in detail using flowcharts and by listing key network structures. To verify the effectiveness of the proposed SSTFA model, both simulated and real-world data are analyzed in Section 6.4. Conclusions are finally given in Section 6.5.

6.3 Preliminary knowledge and the proposed methodology

6.3.1 Brief introduction to the STFT and the SST

The TFR obtained by applying the STFT of a specific signal is generated by a sliding window, and the corresponding Fourier spectrum is recorded as the spectral amplitude for each time slice. The expression of the TFR generated by the STFT can be written as:

$$S(u,\xi)=\int_{-\infty}^{+\infty}x(\tau)g(\tau-u)e^{-i\xi(\tau-u)}\mathrm{d}\tau \tag{6-1}$$

where $x(t)$ is the simulated signal with a time-varying frequency, $g(\ )$ denotes the window to truncate the analyzed signal (usually a Gaussian window is used), i is the imaginary unit which means that $i=\sqrt{-1}$, τ and ξ are the time and frequency indices. The corresponding TFR $S(u,\xi)$ has a specific amplitude at each TF point, so that the TF energy is reflected by the different colors. As a special case of the linear chirplet transform (LCT), the STFT can be considered as employing a chirp rate equal to 0, which is parallel to the time axis. According to this, it can be found that the STFT leads to higher energy concentrations when the frequency is relatively constant, as a 0 chirp rate will match the changing patterns of the frequency at those time instants, and vice versa, a wrongly estimated chirp rate will even degrade the TFR, causing blurring problems. The chirp rate must match the changing pattern of the analyzed frequency to improve the readability, as is widely used in parameterized TFA.

The SST is used as a post-processing technique based on the STFT results, and can be formulated as:

$$T(u,\omega)=\int_{-\infty}^{+\infty} S^{g}(u,\xi)\cdot\delta\left(\omega-\tilde{\omega}(u,\xi)\right)\mathrm{d}\xi \tag{6-2}$$

where $S^{g}(u,\xi)$ shares the same expression with Eq. (6-1), using a specific window $g(t)$, and the TF energy is reassigned by the IF estimator, recorded as $\tilde{\omega}(u,\xi)$, which can be expressed as:

$$\tilde{\omega}(u,\xi)=\frac{\partial_{u}S^{g}(u,\xi)}{i2\pi S^{g}(u,\xi)}=\xi-\frac{S^{g'}(u,\xi)}{i2\pi S^{g}(u,\xi)} \tag{6-3}$$

where $S^{g'}(u,\xi)$ denotes the resulting representation using a specific window $g'(t)$. After applying energy redistribution, the TF energy becomes more condensed along the TF ridges, leading to a sharper result. However, the SST also fails to improve the energy concentration level when the frequency is rapidly changing, since the TFR created using the STFT has already suffered from heavy TF smearing. An example of this issue is given in Fig. 6-1, where a signal with two varying IFs is analyzed, formulated as:

$$x(t)=\sum_{k=1}^{2}\exp\left(1i\cdot 2\pi\int_{0}^{t}f_{k}(\tau)\mathrm{d}\tau\right),\begin{cases}f_{1}(t)=600-300\cos(4\pi t)\\ f_{2}(t)=150+80\sin(2\pi t)\end{cases} \tag{6-4}$$

The simulated signal has a duration of 1 s and the corresponding two IFs are plotted in Fig. 6-1 (a). It can be found that the frequency component related to $f_{1}(t)$ has a stronger FM law than $f_{2}(t)$. The signal lasts for 1 s, and the sampling frequency is set to 2000 Hz. The signal waveform of $x(t)$ is presented in Fig. 6-1 (b) for illustrative purposes. The TFRs generated by the STFT and the SST, following Eqs. (6-1) and (6-2), are shown in Fig. 6-1 (c) and (d), respectively. It can be found that the SST result leads to a higher TF energy by taking a look at the color bar range. A TF slice comparison is also given to show the difference, as plotted in Fig. 6-1 (f).

However, both these two methods fail to improve the TF energy for $f_1(t)$, as it shows strong FM. The IF estimator guiding the TF redistribution is also drawn, as shown in Fig. 6-1 (e), where the color denotes the frequencies of the TF ridge.

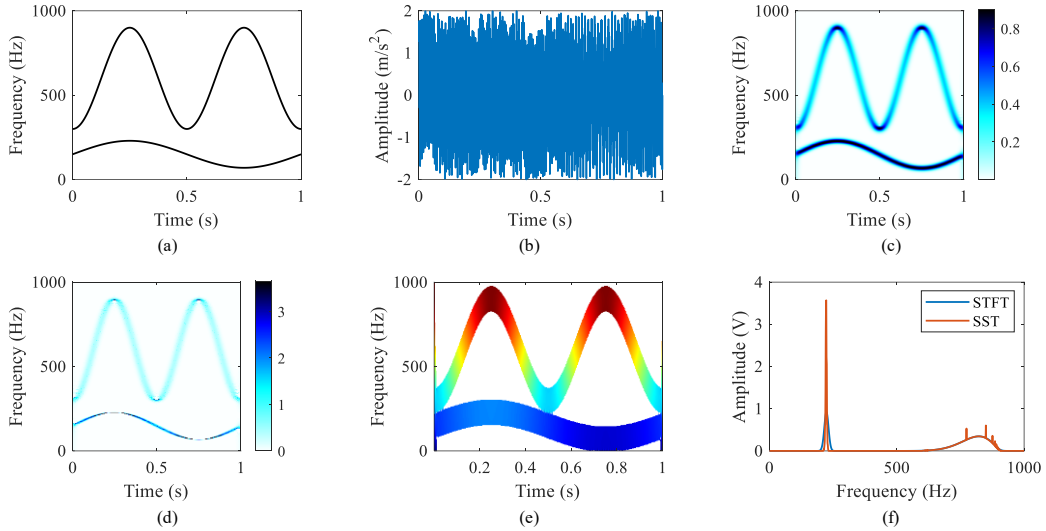


Fig. 6-1. Sample signal analysis: (a) true IF, (b) original signal waveform, (c) TFR using the STFT, (d) TFR using the SST, (e) IF estimator, and (f) spectral slice comparison at time instant $t = 0.37$ s.

6.3.2 IF estimation

6.3.2.1 Existing IF estimator

To improve results when the IF is rapidly changing, the matching SST further introduces a chirp-rate operator $\tilde{c}(u,\xi)$ and a group delay estimator, recorded as $\tilde{t}(u,\xi)$ [20,99]. These additions can be expressed respectively by:

$$\tilde{c}(u,\xi) = \frac{\partial_u \tilde{\omega}(u,\xi)}{\partial_u \tilde{t}(u,\xi)} \tag{6-5}$$

$$\tilde{t}(u,\xi) = u - \frac{\partial_\xi S(u,\xi)}{i2\pi S(u,\xi)} \tag{6-6}$$

Then an improved IF estimator brings together associated TF points such that:

$$\tilde{\omega}_x(u,\xi) = \tilde{\omega}(u,\xi) + \tilde{c}(u,\xi)\left(u - \tilde{t}(u,\xi)\right) \tag{6-7}$$

Combining the defined operators, the new IF estimator can be further derived as:

$$\tilde{\omega}_x(u,\xi) = \xi - \frac{\left(S^{g''}(u,\xi)S^{g}(u,\xi) - S^{g'}(u,\xi)^2\right)S^{tg}(u,\xi)}{i\left(S^{tg}(u,\xi)S^{g'}(u,\xi) - S^{tg'}(u,\xi)S^{g}(u,\xi)\right)S^{g}(u,\xi)} \tag{6-8}$$

Here, several TFRs are introduced by applying different window functions. For instance, the original TFR using the STFT with a Gaussian window $g(t)$ can be written as $S^{g}(u,\xi)$. Similarly, $S^{g''}(u,\xi)$ denotes the result using a window function of the second derivative of $g(t)$, and $S^{tg'}(u,\xi)$ is acquired by using a window of the first derivative of $g(t)$ multiplied by t. Then the resulting TFR guided by the improved IF estimator can be formulated as:

$$T_x(u,\omega)=\int_{-\infty}^{+\infty} S^{g}(u,\xi)\cdot\delta\left(\omega-\tilde{\omega}_x(u,\xi)\right)\mathrm{d}\xi \tag{6-9}$$

The new TFR is constructed according to the IF of the chirp-rate-like signal and an improved IF estimator compared to that of the original SST, as given in Eq. (6-3). The main contribution provided by calculating the improved IF estimator is to match the changing pattern of the time-varying frequency trajectories, which further alleviates the TF smearing problem.

6.3.2.2 Improved IF estimator

Another method, known as the MSST, includes multiple energy redistribution operations that are applied to generate the final TFR [19]. The idea behind this method is that the TF energy can be gradually reallocated to the true IFs in a stepwise manner. Assuming an FM signal, this can be formulated as [20]:

$$x(t)=Ae^{i\left(\varphi(t)+\varphi'(t)(t-u)+0.5\varphi''(t)(t-u)^2\right)} \tag{6-10}$$

where $\varphi(t)$ denotes the instantaneous phase, $\varphi'(t)$ means the IF, and $\varphi''(t)$ means the chirp-rate.

The IF estimator using Eqs. (6-7) and (6-8) can be written as:

$$\tilde{\omega}_x(u,\xi)=\varphi'(u)+\frac{\varphi''(u)^2}{1+\varphi''(u)^2}\left(\xi-\varphi'(u)\right) \tag{6-11}$$

which indicates that the estimated IF is a biased estimation, and the error can be determined such that

$$\left|\tilde{\omega}_x(u,\xi)-\varphi'(u)\right|=\frac{\varphi''(u)^2}{1+\varphi''(u)^2}\left(\xi-\varphi'(u)\right)<\xi-\varphi'(u) \tag{6-12}$$

which shows that $\tilde{\omega}_x(u,\xi)$ is much closer to the true IF so that iteration can be applied to create an improved IF estimator in a stepwise manner. After N iterations, the IF estimator can be recorded as:

$$\tilde{\omega}_x^{N}(u,\xi)=\varphi'(u)+\left(\frac{\varphi''(u)^2}{1+\varphi''(u)^2}\right)^{N}\left(\xi-\varphi'(u)\right) \tag{6-13}$$

and the corresponding TFR will be

$$T_x^N(u,\omega)=\int_{-\infty}^{+\infty} S^g(u,\xi)\cdot\delta\left(\omega-\tilde{\omega}_x^N(u,\xi)\right)\mathrm{d}\xi \tag{6-14}$$

A comparison of the two IF estimators formulated by Eqs. (6-3) and (6-13) is plotted in Fig. 6-2. A local zoom from 0.34 to 0.42 s is provided for a closer comparison, where the frequency is presented by the different colors. It can be found that the IF estimated by the STFT is distributed in an inclined fashion relative to the time axis, making the TF distributions of the same frequency component dispersed when the IF shows rapidly changing laws. Then, considering that the reassignment operation is applied only along the frequency axis, which leads to diffusion problems, as the true TF energy is reassigned to different TF points. Simultaneously, it can be found that the improved IF estimator here is distributed perpendicular to the time axis, so it can be guaranteed that the TF energy is gathered relative to the true IFs since the reassignment is only applied along the frequency axis.

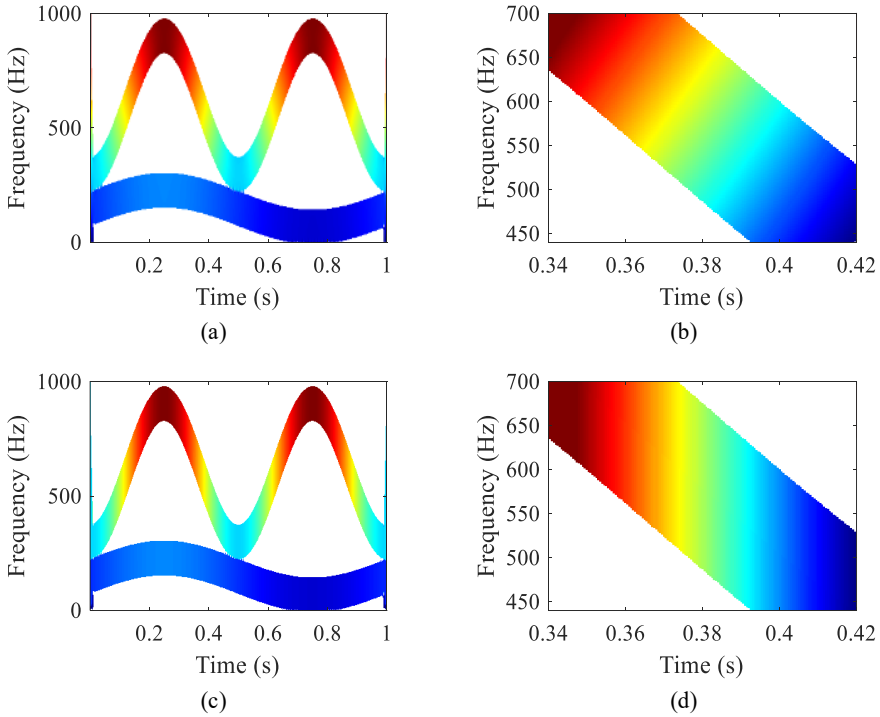


Fig. 6-2. IF estimator comparison: (a) original IF estimator in the SST, (b) local zoom of (a), (c) improved IF estimator, and (d) local zoom of (c).

The IF extractor from the improved IF estimator can be written as:

$$IF(u,\xi)=1\left(\left|\xi-\tilde{\omega}_x^N(u,\xi)\right|\le\Delta f\right) \tag{6-15}$$

where $IF(u,\xi)=1$ when the estimated frequency is equal to the true IF. Δf is the frequency resolution when performing a discrete Fourier transform during code implementation. Note that zero padding would cause the estimated TFR to have sidebands in the TF plane. A clearer version

of the TF amplitude-aware IF extractor of Eq. (6-15) can be further written as:

$$IF(u,\omega)=1\left(S_x^g(u,\omega)>\tau_{TFR},\ \omega-\tilde{\omega}_x^N(u,\xi)<\Delta\omega\right) \tag{6-16}$$

where τ_{TFR} is the threshold setting of the TF amplitude so that the resulting IF extractor can avoid the boundary effect, which is usually caused by zero padding at the beginning and the end of the analyzed signal. Considering that $\Delta\omega=1$ only retains one TF point when the estimated IF matches the true IF exactly, while a larger $\Delta\omega$ value means that several TF points around the true IF are retained, which leads to broader TF ridges.

After introducing the improved IF estimator here, the TF amplitude-aware IF extracting operator in Eq. (6-16) is plotted, which is used to guide the method in retaining only the TF distributions around the true IFs, as shown below in Fig. 6-3, with two local zooms for both $f_1(t)$ and $f_2(t)$ provided. It can be seen that the IF extractor guided by the improved IF estimator matches the true IFs well, showing the potential of substituting the true IFs from the analyzed signal with estimated IF extractors so that prior knowledge of the true IFs can be avoided, thus providing the solution for nonparametric TFA.

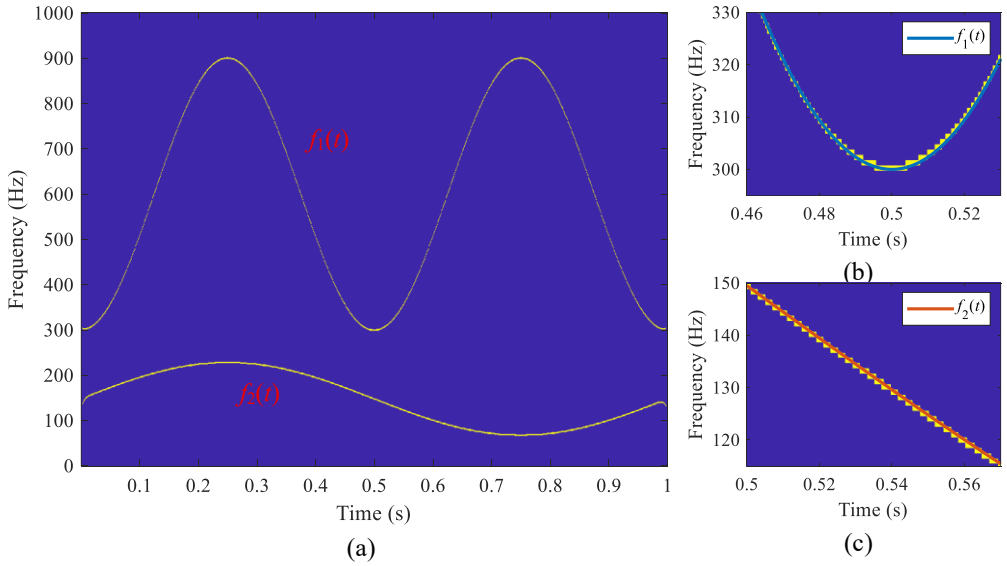


Fig. 6-3. IF estimation guided by the improved IF estimator: (a) IF extractor, (b) local zoom of $f_1(t)$, and (c) local zoom of $f_2(t)$.

6.3.3 Overview of the proposed method

For the proposed SSTFA model, the input is 1-D time series data, which can be easily generated through simulation. During dataset preparation, increasing the complexity of the self-generated training data is essential for improving the model's generalization capability to unseen signals. For signals where there is only an individual frequency component, it is easier to reveal

the IF by implementing TFA techniques. In contrast, multi-component signals present greater challenges and can generally be categorized into two cases: (1) intersecting components and (2) non-intersecting components. Intersection refers to the scenario in which two frequency components share the same frequency value at one or more time instants, whereas non-intersecting components remain separable throughout the TF plane. Representative examples of both cases are provided later for evaluation.

As an end-to-end TFA method, the proposed method aims to provide the resulting TFR directly without prior knowledge of target frequency information, which makes the training procedure self-supervised. To achieve accurate IF estimation, an IF extractor is incorporated and guided by the improved IF estimator introduced earlier in Section 6.3.2.2, which serves as the self-supervised loss function. Energy enhancement is realized through encoder-decoder modules that progressively refine the TF representations.

A flowchart of the proposed method is plotted in Fig. 6-4, which consists of two modules: (1) the TF convolution module and (2) the energy enhancement module. The TF convolution module transforms the original time domain input to two-dimensional TFRs, where kernels with adaptive parameter values are learned by the model. The kernels act as sliding window functions in the STFT, which aim to reveal coarse TF information. As window length is an important setting when performing TFA, a series of kernels is employed to simulate different window types used to generate a more diverse set of TF features, as different frequency components require different TF resolutions.

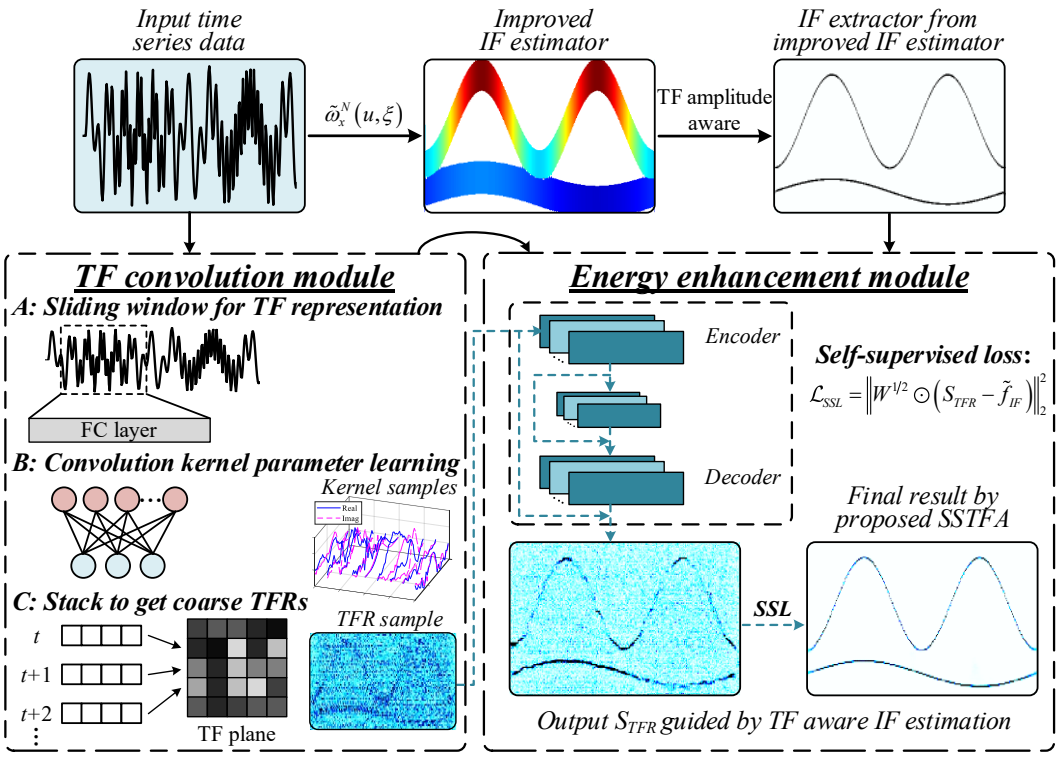


Fig. 6-4. Flowchart of the proposed SSTFA method.

In the second energy enhancement module, the model aims to improve the coarse TFRs so that they become closer to the ideal TFR, where the energy in the TFR only concentrates around the true IFs. This operation also acts as a common reassignment when it comes to post-processing techniques, as sharper TF ridges are desired. To realize this, an encoder-decoder module is designed. By stacking multiple layers with shortcut connections, following the same idea developed in ResNet, the model can learn deep features from the feature maps [35].

Additionally, by considering the efficiency of model training, TF resolution needs to be adjusted. It is obvious that a higher resolution in both the time and frequency domains can lead to a clearer TFR as more TF points are included when plotting the results. However, increasing the TF resolution also greatly increases the size of each sample, thereby increasing computation time. To balance representation quality and training efficiency, a resolution of 256×256 is adopted in this work, which is commonly used in image-based deep learning tasks. This configuration corresponds to 256 time bins and 256 frequency bins in the resulting TFRs.

The pseudo labels are inherently sparse, as illustrated in Fig. 6-3, with nonzero amplitudes only in the TF region of the target frequency trajectories. The representation learning procedure, guided by a large amount of example data, can generate a TFR that is cleaner and more robust, even if the true IFs are not involved in the training procedure.

Nonetheless, there is still a challenge with multi-component signal analysis, as it is hard to get accurately estimated IFs directly, especially when there are cross terms between two adjacent frequency components. To simulate more complicated cases, multi-components are defined by adding mono-component signals with different amplitudes together using the superposition theorem. Amplitude here could be used to simulate an amplitude-modulated (AM) signal so that the trained model can still be used to reveal the frequency components with lower amplitudes. By defining large-scale multi-component signals, the proposed method can potentially generalize well to new signals, further boosting model performance when performing TFA. To enhance weak features in the proposed SSTFA model, global scaling is used. In this case, the weight matrix of a signal with *k* components can be written as:

$$\begin{aligned} W\left(u,\xi\right) &= \sum_{k=1}^{N} w_k IF_k\left(u,\xi\right), \\ w_k &= 20\log_{10}\left(\frac{A_x}{a_k+\varepsilon}+1\right), \\ A_x &= \max_{k=1,\ldots,N}\left(a_k\right) \end{aligned} \tag{6-17}$$

where the weighting strategy is recorded by the matrix $W(\ )$, w_k denotes the amplitude, and $IF_k\left(u,\xi\right)$ denotes the IF estimation for each mono-component, ε is added here to avoid a denominator equal to 0 (i.e., $\varepsilon = 10^{-12}$). It can be found that w_k shares the expression of SNR. Here, the weight represents the signal-to-signal ratio used to enhance relatively weak features from the multi-component signals. Finally, for the loss function, two-dimensional regularization between the output and the estimated IFs is calculated, expressed as:

$$\mathcal{L}_{SSL} = \sum_{k=1}^{N} w_k \left\| IF_k\left(u,\xi\right) \odot \left(S_{TFR}\left(u,\xi\right) - IF_k\left(u,\xi\right)\right)\right\|_2^2 \tag{6-18}$$

where $\odot$ denotes point-by-point multiplication, and the final output by the model is recorded as $S_{TFR}\left(u,\xi\right)$. It can be seen that the same TFR size is obtained, and the resulting TFR is formed to learn the target frequency components using the IF estimation. As the $IF_k\left(u,\xi\right)$ represents the ideal IF estimation for each *k* component, it is desirable to obtain a clear ridge as a final result.

6.3.4 Algorithm implementation

To obtain the improved IF estimator, a pseudo-code is given in Algorithm 6.1.

Algorithm 6.1. The improved IF estimator in Eq. (6-13)

Step 1: *Initialization and calculation*

Determine the window function g, g', tg, tg', g'' and generate the corresponding TFRs $S_x^g, S_x^{g'}, S_x^{tg}, S_x^{tg'}, S_x^{g''}$

Step 2*: Improved IF estimators with N iterations*

Calculate and initialize $\tilde{\omega}_x^1(u,x) \leftarrow \tilde{\omega}_x(u,\xi)$ according to Eqs. (6-3) and (6-13).

for iter=1:*N*

  for *nt*=1:*t* bins

    for *nf*=1:*f* bins

      $\xi \leftarrow \tilde{\omega}_x^{iter}[nt,nf]$

      $\tilde{\omega}_x^{iter+1}[nt,nf] \leftarrow \tilde{\omega}_x^{iter}[nt,\xi]$

    end *for*

  end *for*

end *for*

Return: Improved IF estimator $\tilde{\omega}_x^N[u,\xi]$.

Network structure details, including encoder-decoder network settings, are provided in Table 6.1.

Table 6.1. Network settings used for the energy concentration module.

Modules	Settings	Output size
Input	/	1×2×256
TF convolution	ComplexConv1d(1, 4096, 31)	16×256×256
Energy enhancement	Encoder: Conv2d(16, 32, 3×3), ReLU	32×128×128
	[Conv2d(32, 32, 3×3), ReLU]×19	32×128×128
	Decoder: [ConvT2d(32, 32, 3×3), ReLU]×19	32×128×128
	ConvT2d(32, 16, 3×3), ReLU	16×256×256
Output	ConvT2d(16, 1, 3×1)	1×256×256

6.4 Signal analysis

In this section, the proposed method is verified using both simulated and real-world signals.

6.4.1 Multi-component signal analysis

As previously discussed, it would be easy to analyze a mono-component signal, since its estimated IF would be more accurate. However, in the real world, it is common for the signal being analyzed to have multiple components at the same time, especially when frequency components are closely located and intersect with each other. To demonstrate the necessity of multi-component TFA, a specific multi-component signal is defined by modifying the previously defined signal in Eq. (6-4) as follows:

$$x(t)=\sum_{k=1}^{3}\exp\left(1i\cdot 2\pi\int_0^t\left(a_k f_k(\tau)\right)\mathrm{d}\tau\right),\quad a_1=0.8,a_2=0.8,a_3=1.1 \qquad \begin{cases} f_1(t)=160-60\cos(4\pi t) \\ f_2(t)=40+20\sin(2\pi t) \\ f_3(t)=220-160\sin(\pi t) \end{cases} \tag{6-19}$$

where the analyzed signal has three frequency components. The true IFs of these three components are given in Fig. 6-5 (a). It can be seen that $f_1(t)$ and $f_3(t)$ intersect at two time instants, and $f_2(t)$ and $f_3(t)$ are closely located around $t=0.4$ s. The resulting TFR using the STFT is shown in Fig. 6-5 (b). The highest TF energy is lower than 1. However, the maximum amplitude should be at least 1.1 for $f_3(t)$. It can be seen that $f_1(t)$ and $f_2(t)$ share the same amplitude, due to faster frequency changing laws, $f_1(t)$ is revealed with a lower TF energy than $f_2(t)$, represented by the light blue curves during increasing and decreasing time instants. The spread reflects the inherent uncertainty of IF estimations caused by finite window lengths and boundary constraints. After applying reassignment operations, TFRs using the SST, SET, and MSST are given in Fig. 6-5 (c)-(e), respectively. It can be found that TFRs created by these methods suffer from frequency oscillations between $f_2(t)$ and $f_3(t)$ around $t=0.4$ s, and discontinuities appear at the frequency crossings, especially for $f_1(t)$ with a lower amplitude. TF ambiguity can also be observed around the frequency crossing area, which will, of course, lead to inaccurate IF estimations. The TFR obtained using the proposed SSTFA model is shown in Fig. 6-5 (f), where three frequency trajectories can be clearly observed, indicating the effectiveness of the proposed method in revealing time-varying frequency components in the analyzed signal.

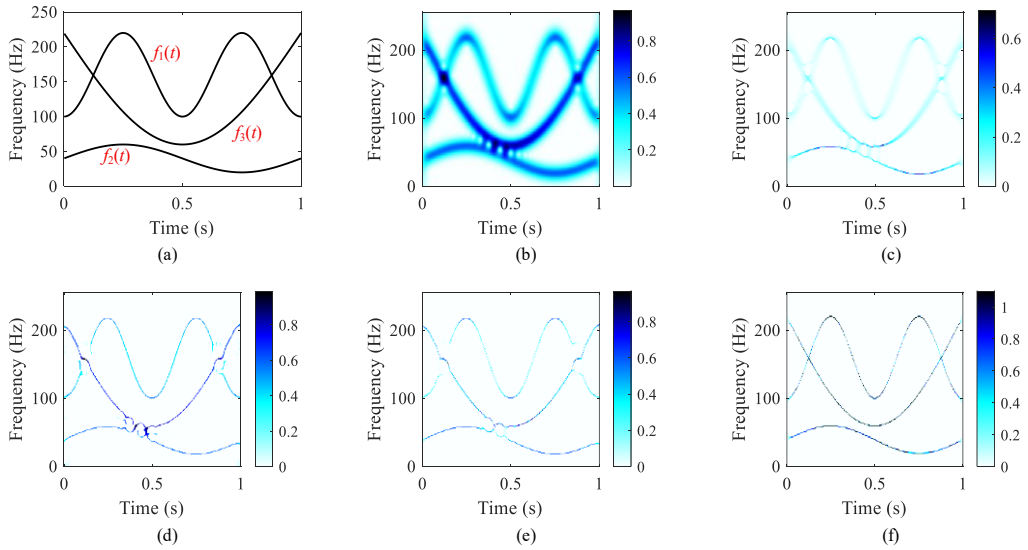


Fig. 6-5. Analysis for a simulated multi-component signal: (a) true IFs for reference, (b)-(f) TFRs using the STFT, SST, SET, MSST, and proposed SSTFA model.

For multi-component signal analysis, conventional IF estimators often suffer from severe cross-term interference, particularly when frequency components are closely spaced or intersect.

In the proposed approach, pseudo-labels are first generated independently for each mono-component signal, thereby avoiding cross-term contamination. These clean pseudo-labels are then used to guide the learning process, enabling the model to produce improved TFRs.

To further address IF estimation errors in multi-component signals, the superposition theorem is exploited. By superimposing TF-amplitude-aware IF estimations obtained from each mono-component signal, an accurate IF estimation for the synthesized multi-component signal is achieved. An example of the IF extractor using the theorem of superposition is provided in Fig. 6-6.

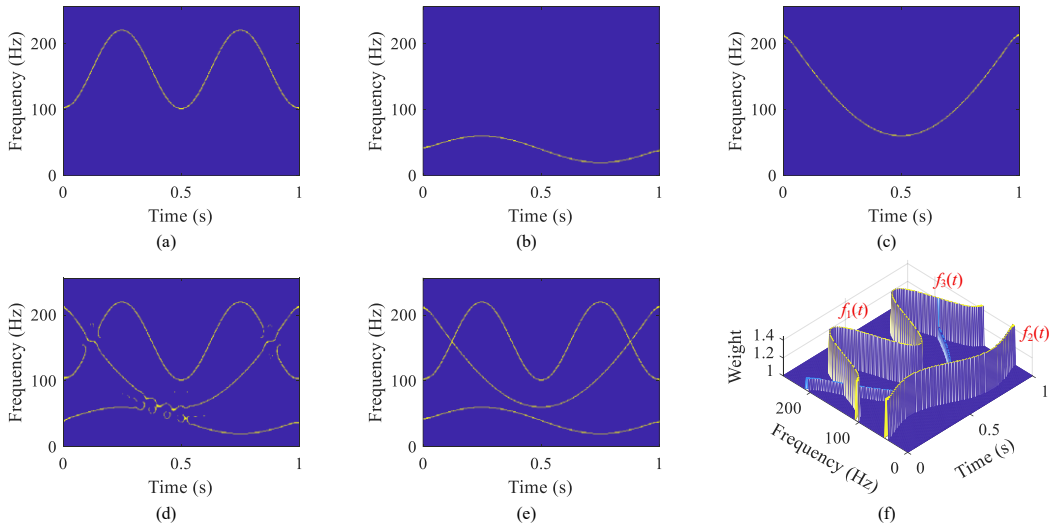


Fig. 6-6. The IF extractors and weights for multi-component signal analysis: (a)-(c) IF extractor for $f_1(t)$, $f_2(t)$, and $f_3(t)$, (d) original IF extractor directly from the multi-component signal without TF energy awareness, (e) IF extractor using the superposition theorem, (f) weighting factor (assuming the global maximum amplitude is 1.3, and amplitudes for $f_1(t)$, $f_2(t)$, and $f_3(t)$ are 0.8, 0.8, and 1.1, respectively).

The IF estimation accuracy for mono signal components retains a high precision, as shown in Fig. 6-6 (a)-(c). In Fig. 6-6 (d), the IF extractor directly derived from the SET suffers from cross-term interferences in the TF region, where frequencies are closely located and intersected, which may lead to TF information loss when used as guidance to train the TFA model. Moreover, frequency components suffer from discontinuity, leading to lower readability and accuracy of the estimated IFs when dealing with multi-component signals. Then the theorem of superposition is adopted here by adding all individual IF extractors after taking proper actions like suppressing sidebands caused by zero padding and eliminating outliers caused by mode overlap via normalization of the new IF extractor for the multi-component signal, as plotted in Fig. 6-6 (e), where three distinct components can be clearly observed. This new IF extractor serves as supervision to guide model training. To balance the energy level of each frequency component, the

weight factor is also plotted here, as shown in Fig. 6-6 (f), where the global maximum is assumed to be 1.3 for illustrative purposes. It can be seen that the weight shares the same variation pattern as the IF extractor, and weights corresponding to weak components are somewhat larger (i.e., weights for f_1(t) and f_2(t) with an amplitude of 0.8 are larger than that of f_3(t) with an amplitude of 1.1), which enables the model to reveal weaker features integrated with the IF extractor.

After generating a series of simulated multi-component signals by randomly combining mono-component signals, the ratio of each frequency component in the simulated signal can be recovered by calculating the weighting factor, since it reflects the TF energy concentration level of each component. In this way, a series of simulated data can be easily generated, thereby ensuring that the model has as many simulated signals and accurate IF estimations as the pseudo labels used during training. This also helps improve the model's performance on new signals, enabling the model to adaptively reveal the frequency's changing patterns in the analyzed signal without any prior knowledge of the true IFs.

Remark 1: Noise robustness for TFA

To enhance model performance against noise, Gaussian white noise can also be added into the newly generated multi-component signals, by defining a signal-to-noise ratio, expressed as:

$$SNR = 10\log_{10}\left(\frac{P_x}{P_{noise}}\right) \tag{6-20}$$

where P_x and P_{noise} denote the power of the signal and noise. Introducing noise leads to additional interferences in the resulting TFRs, making model training more challenging and helping the model generalize to real-world noisy conditions. During model training, the SNR of the noisy signal is set to -5 dB. Note that while a lower amplitude can be adopted so that the training model can be more robust against noise, the resulting TFRs become more sensitive to noise, which may reveal undesired frequency components caused by external noise.

Remark 2: Model training setup

In the training stage, 30,000 pairs of simulated signal samples and their pseudo labels are generated by randomly combining 3,000 single-component signals, and the model is trained for 400 epochs. A maximum of 6 frequency modulation components can exist simultaneously in each training sample. The combination is totally random, thus enabling frequency crossings and closely-located frequency components that cannot be clearly represented by using existing TFA methods. It is also worth noting that increasing the number of training samples or epochs may further improve model performance at the cost of additional computation time. The learning rate is initially set to 0.001. The loss versus epoch is provided in Fig. 6-7. Both the training and validation

losses decrease rapidly during the early training stage and then tend to stabilize. After approximately 200 epochs, the training loss continues to decrease, while the validation loss shows only a slight increasing trend. This indicates that a wide range of epochs, for example from 200 to 400 epochs, may provide usable models for the present task. In this study, the training loss is also an important reference because the objective is to learn a signal representation model from simulated samples rather than to perform a conventional classification task with a fixed validation set. Since both the training and validation sets contain only a limited number of simulated signals, the validation loss should not be interpreted as a complete measure of generalization to all possible nonstationary signals. Empirically, the model trained for 200 epochs already produces acceptable time-frequency representations. Training for 400 epochs can further reduce the loss on the training samples and may improve some local details in the resulting representations. However, whether these improvements consistently hold for unseen signals requires further analysis. Therefore, the number of training epochs is selected by considering both the loss curves and the quality of the obtained time-frequency representations.

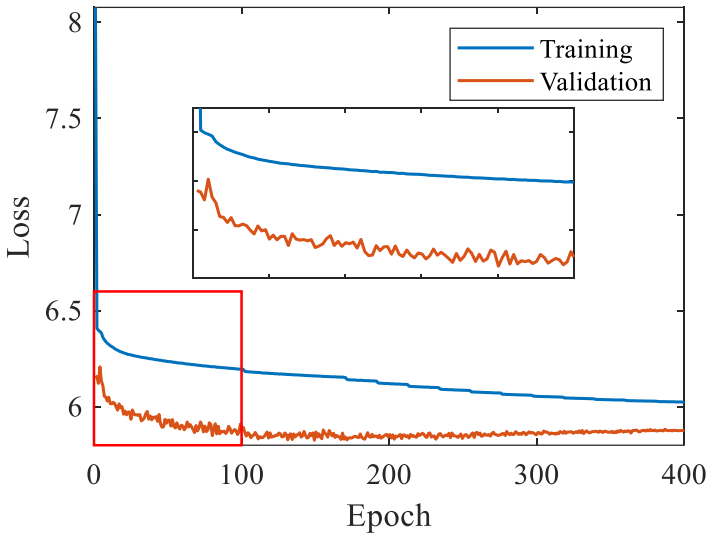


Fig. 6-7. Loss versus epoch.

Remark 3: Runtime comparison

To know the computational efficiency of each method compared, the runtime of each algorithm is recorded. Each algorithm is implemented 1000 times. The average time is then calculated and recorded, as shown in Table 6.2. All methods compared were executed in MATLAB R2022a on a workstation with a 12th Gen Intel Core i7-12700K, 32 GB RAM, and an RTX 3080 Ti. The proposed SSTFA model was implemented in Python (PyTorch 1.13.0, CUDA 11.7) and tested on both GPU and CPU.

Table 6.2. Computation time comparison.

Methods	Runtime
STFT	0.492 ms
SST	6.178 ms
SET	4.894 ms
MSST (20 iterations)	14.635 ms
Proposed SSTFA model (CPU)	324.892 ms
Proposed SSTFA model (GPU)	5.565 ms

It can be seen that, except for the MSST, the computation time of all comparison methods is less than 10 ms. This is because the MSST performs iterative reassignment operations to enhance the level of TF energy. While 20 iterations were used, the computation time is not directly 20 times that of performing the SST alone, since the IF estimator is first calculated after 20 iterations, and the reassignment operation is performed only once to get the same result. The proposed SSTFA method clearly demonstrates different performances on CPU and GPU. When executed on a CPU, the proposed SSTFA model has a higher runtime of 324.892 ms, which is expected because the designed model involves many convolutional operations through deep layers. However, when accelerated on the GPU, the runtime is significantly reduced to 5.565 ms, which is nearly 58× faster than the CPU version and comparable to (or even faster than) some classical reassignment-based methods such as the SST and the MSST. These results confirm that the proposed SSTFA model is computationally efficient when deployed in a GPU-accelerated environment.

To further show the effectiveness of the proposed method, a noisy version of simulated data is also analyzed, with an SNR set to 5 dB. The noisy signal waveform is plotted in Fig. 6-8 (a). In this case, more oscillation phenomena can be observed compared to the noise-free signal. The TFRs obtained by the STFT, SST, SET, MSST, and the proposed SSTFA model are displayed in Fig. 6-8 (b)-(f), respectively. By comparing the results, it can be found that all methods compared suffer from heavy TF smearing problems around $t = 0.5$ s, where the three components are closely located. The proposed method shows some frequency oscillations when characterizing $f_2(t)$, while the changing pattern for $f_1(t)$ and $f_3(t)$ can be clearly observed, especially around $t = 0.1$ s and $t = 0.9$ s, which shows the robustness of the proposed SSTFA model when analyzing noisy signals. It can also be found that TFRs created using the STFT, SST, SET, and MSST share almost the same TF amplitude range. However, the TF amplitude by the proposed SSTFA model is enlarged by applying the weighting factor, which may lead to inaccurate signal reconstruction from the resulting TFRs.

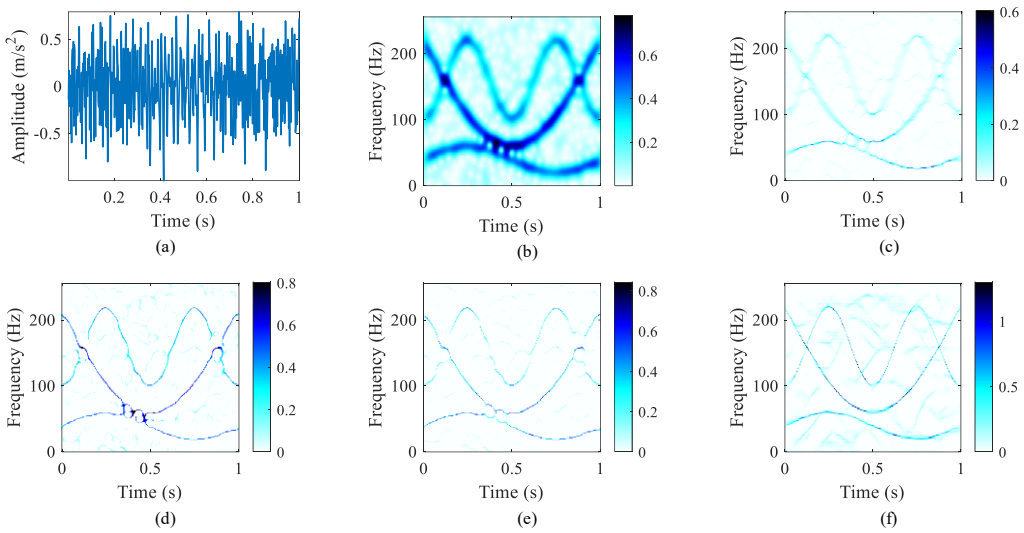


Fig. 6-8. Noisy multi-component signal analysis: (a) noisy signal waveform, (b)-(f) TFRs created using the STFT, SST, SET, MSST, and the proposed method.

6.4.2 Bearing vibration signal analysis

In this section, the SSTFA model is validated by analyzing a vibration signal from a bearing with an outer race fault. The bearing data is provided by the Case Western Reserve University (CWRU). The experimental setup is shown in Fig. 6-9 (a), where the bearing in the motor has an outer race fault.

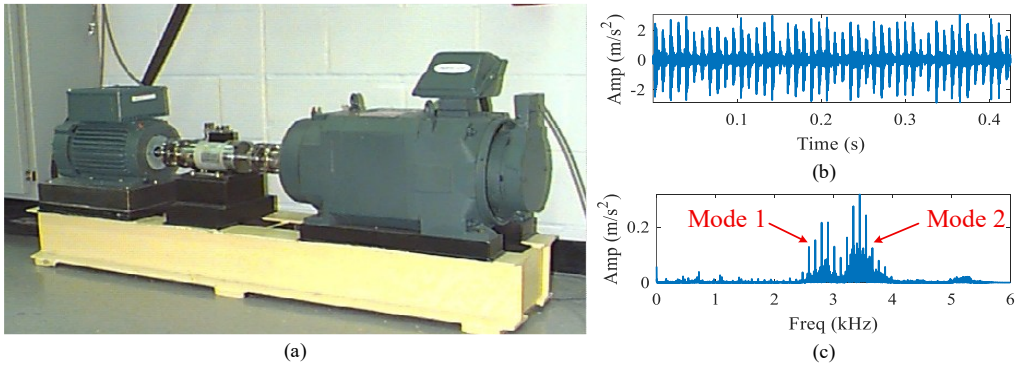


Fig. 6-9. CWRU bearing data: (a) experimental test rig, (b) original time domain signal, and (c) Fourier spectrum of (b).

The signal is collected by the accelerometer placed at the drive end. The sampling frequency is set to 12 kHz. The signal waveform and its corresponding Fourier spectrum are given in Figs. 6-9 (b)-(c). It can be found that the analyzed signal shows strong AM features. The spectrum indicates that the main frequency components are in the frequency range of [2.5, 4] kHz. It can be found that there are two different modes (besides 3 kHz), as marked in red in Fig. 6-9 (c). From the amplitude, it can be concluded that mode 2, which has a higher frequency component, also has a higher energy level. To reveal more details of these two modes in this frequency range, different

TFA methods are applied. From top to bottom, Fig. 6-10 plots the analyzed signal segment (5 fault-related impulses are selected), as well as the TFRs generated by the STFT, SST, MSST, and the proposed SSTFA model with reassignment operations, where the reassignment is conducted in the same way as the MSST (MSST obtains TFRs using the STFT, while the result in Fig. 6-10 is obtained using the trained SSTFA model).

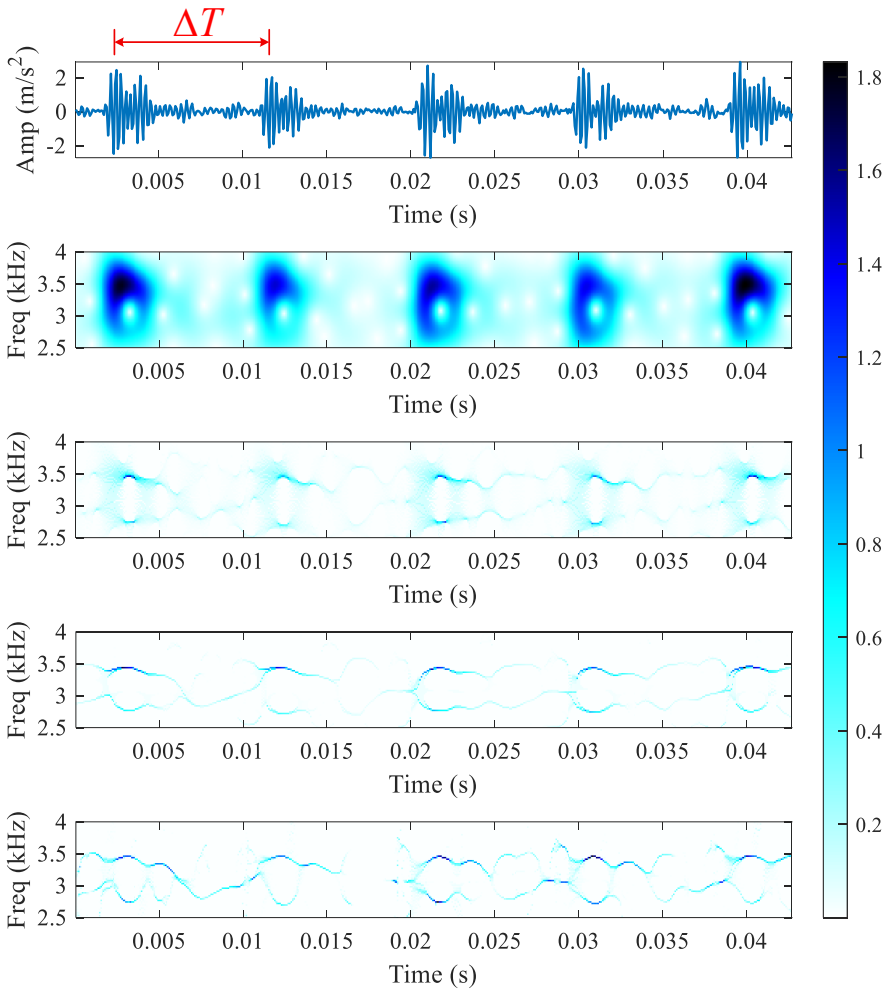


Fig. 6-10. Analyzed signal segment (top) and TFRs by different methods (from top to bottom): STFT, SST, MSST, and proposed SSTFA model with reassignment.

Five fault-related impulses with the same time interval ΔT can be observed. The STFT result suffers from heavy TF smearing, and the TF features are not continuous, especially when the signal has lower amplitudes. For the SST result, the TF energy concentration level is not condensed enough, which happens especially at the impulses. After employing multiple TF energy reassignments, the MSST leads to a better TFR with both condensed TF energy and fewer smearing problems. However, due to a low amplitude, the result is still discontinuous between impulses for

mode 2, especially during impulse intervals, when compared to the result obtained by the proposed method.

Once all TFRs are obtained, the changing patterns of modes 1 and 2 detected from the TFR by the proposed model are plotted. To make a detailed comparison, local TFRs with detected TF ridges are plotted at the same time, as shown in Fig. 6-11.

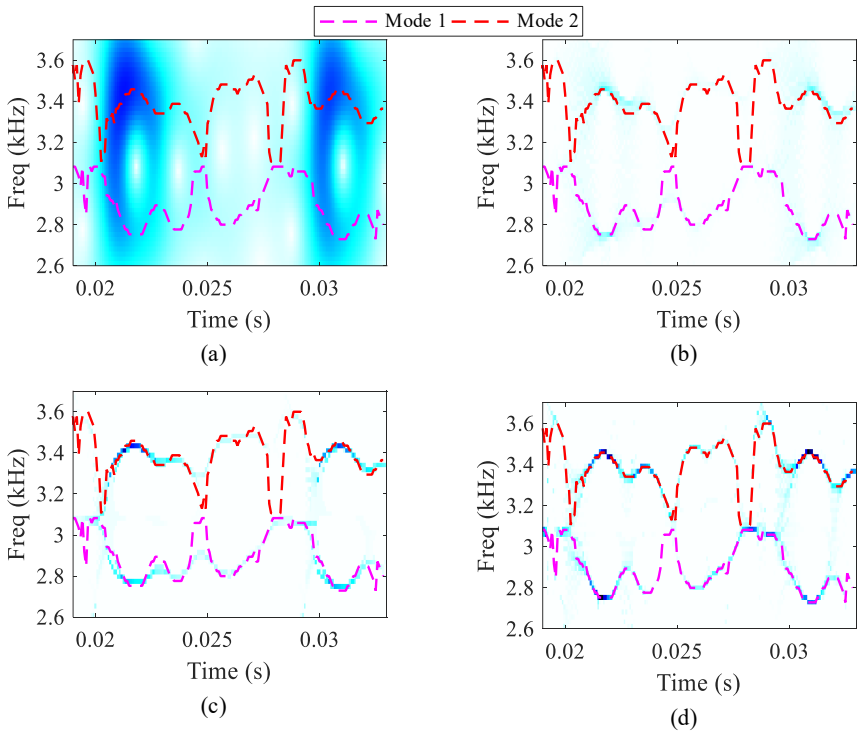


Fig. 6-11. Local zoom with detected modes: (a) STFT, (b) SST, (c) MSST, and (d) proposed method.

Time-varying frequency trajectories of mode 1 and mode 2 are drawn in magenta and red, respectively. It can be seen that the extracted TF ridges of these two modes generally match TFRs for all compared methods, indicating that the proposed SSTFA model can provide powerful insight while ensuring accurate TF feature representations for bearing vibration data analysis, paving the way for further analyses like ridge detection and signal reconstruction.

6.4.3 Bat echolocation signal analysis

To further examine the effectiveness of the trained SSTFA model, a bat echolocation signal that has a multi-component FM signal with IFs experiencing a run-down period is analyzed. The signal has 400 sampling points with a 7 μs (7×10^{-6} s) sampling period, which means that the sampling frequency is almost equal to 142.86 kHz ($1/\left(7\times10^{-6}\right)$ Hz). As a classical example of TFA, the analysis results can help provide powerful insight into the performance of the TFA method itself. The signal waveform is plotted in Fig. 6-12 (a). The Fourier spectrum is provided

in Fig. 6-12 (b), where local peaks can be observed within a wide range of frequencies. By performing TFA, the one-dimensional time domain signal is transformed to a two-dimensional TF plane, so that time-varying features are revealed. TFRs created using the STFT, SST, SET, and proposed SSTFA model are displayed in Fig. 6-12 (c)-(f), respectively. It can be seen that the proposed model extracts more features from the analyzed bat signal. The trained SSTFA model outperforms other comparison methods, effectively recognizing time-varying features below 20 kHz, and can better reveal the frequency aliasing near the upper frequency limit.

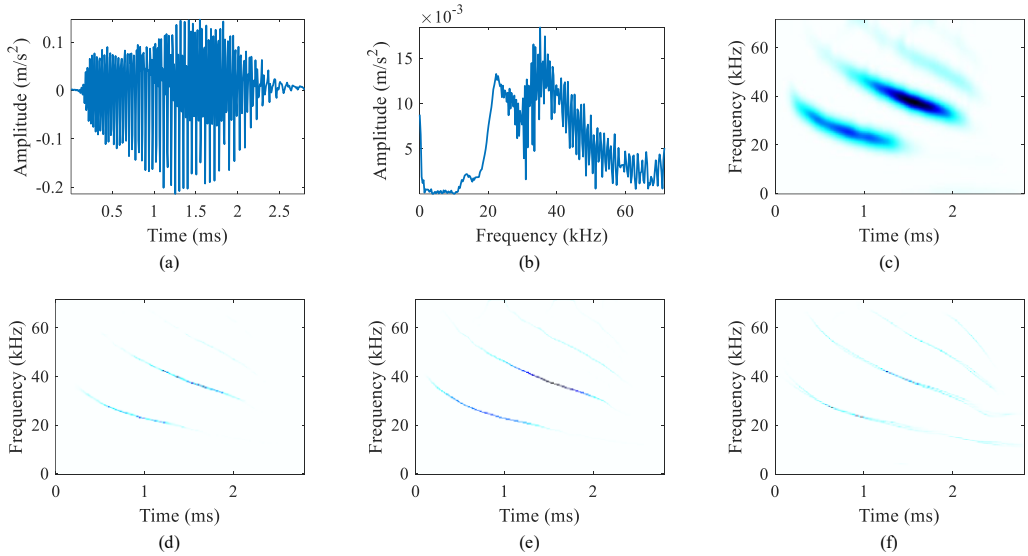


Fig. 6-12. Bat signal analysis: (a) original signal waveform, (b) Fourier spectrum, (c)-(f) TFRs using the STFT, SST, SET, and proposed SSTFA model.

The results based on two real-world signals show that the trained model generalizes well for analyzing complex real-world signals. Furthermore, comparing the TF feature representations generated by the SSTFA model with those of the STFT model reveals that its superior performance in revealing weak time-varying features can be further improved through reassignment operations.

6.5 Conclusion

In this chapter, a new end-to-end TFA tool is proposed, which is used for adaptive TFA without prior knowledge of the true IFs. By investigating the signal processing technique, an improved IF estimator is first established, which can lead to a more accurate IF estimation in a stepwise manner with multiple reassignment operations. This procedure is mathematically verified. Based on the improved IF estimator, the TFR only shows TF energy around the true IFs, which later acts as a pseudo label to train the SSTFA model. Here, the label is not externally obtained. Thus, the training procedure is self-supervised. Then, to generate a clear TFR, two modules are employed in the SSTFA network. They are: (1) TF convolution module, and (2) energy

enhancement module. In this case, the TF convolution module aims to transform the time series data into a two-dimensional TF plane, and the energy enhancement module guides the network to gather the TF energy around the true IFs through designed encoders and decoders. For multi-component analysis, the superposition theorem is used to ensure accurate estimation of multiple frequency components, and a weighting strategy is adopted to enhance weak components, thereby making the training model effective. By combining mono-component signals into multi-component signals, numerous data samples can be generated, thereby improving the generalization performance of the training model on new data. Finally, the effectiveness of the SSTFA model is verified by analyzing both simulated and real-world data.

In TFA, signal reconstruction remains important because transforms must be invertible, which means that the original time domain signal for each component can be restored by searching for local peaks using ridge detection algorithms. Signal reconstruction is used when it comes to increasing the signal-to-noise ratio and further improving the readability of a TFR. However, for multi-component signals, reliable reconstruction still depends heavily on precise TF ridge detection, which becomes highly challenging in the presence of mode overlap or frequency intersections. Moreover, even though the proposed SSTFA model leads to concentrated TF energy, the resulting TFR may exhibit inaccurate TF amplitude estimates at certain time instants. How to better realize signal reconstruction under such conditions should be further studied.

Chapter 7 Contributions, conclusions, comparisons, and future work

7.1 Contributions

The goal of this thesis was to address several problems associated with IFD under variable working conditions, particularly in scenarios where the target domain remains unseen, and there is no sufficient training data available. Traditional signal processing-based fault diagnosis methods can achieve accurate results, but they typically rely heavily on expert knowledge and manual feature engineering. To improve automation and intelligence, deep learning-based IFD strategies have been developed, framing fault diagnosis as an end-to-end classification task in which health states are directly mapped to their corresponding labels. This strategy reduces reliance on domain expertise and enables the model to autonomously learn discriminative features from collected vibration data. When both the training and testing data are fully accessible, achieving high diagnostic accuracy with an IFD model is relatively straightforward. However, when the model encounters unseen data or unknown distribution shifts, its performance often degrades significantly. This sensitivity to distributional changes severely limits the practicality of conventional models, as they are unable to adapt to new working conditions. Consequently, improving a model's robustness and ability to generalize becomes essential for real-world applications.

To train a more generalized model, methods were investigated and new methods were proposed. The contributions of this thesis are listed as follows:

(1) Domain-invariant features are further divided into mutually invariant and internally invariant features in Chapter 3 . This refined categorization provides a more comprehensive understanding of how features behave across and within different working conditions, offering a new perspective for constructing robust DG-based IFD models.

(2) The dynamic feature distribution in an individual working condition is considered by introducing pseudo domain labels in Chapter 4 . These pseudo labels allow the model to capture fine-grained domain variations and effectively learn representations that remain stable even when the operating environment changes continuously.

(3) Domain-related variations introduced by variable working conditions are systematically studied in Chapter 5 . By eliminating these domain-discriminative components, the model can focus on learning domain-invariant representations, thereby improving generalization to unseen target domains and enhancing the interpretability of the learned features.

(4) A self-supervised TFA framework is proposed to realize adaptive TFA by incorporating a TF convolution module and an energy enhancement module in Chapter 6 . The resulting SSTFA model can effectively capture time-varying characteristics of a signal without relying on the true IFs, making it inherently IF-free. This adaptive TFA mechanism not

only enhances the clarity and accuracy of the generated TFRs but also enables automatic preparation of training data, thereby eliminating human intervention and reducing the need for expert knowledge during feature characterization.

The effectiveness of the proposed methods was validated using experimentally acquired signals.

7.2 Conclusions

The research completed for intelligent fault diagnosis and health state recognition of rotating machinery under variable working conditions is summarized in the following.

(1) Internally invariant and mutually invariant features across different working conditions when performing IFD

To better study how domain-invariant features work across different working conditions when performing IFD and to train a generalized model when the target domain is unseen, domain-invariant features are divided into internally invariant and mutually invariant features in Chapter 3 . Internally invariant features only exist in the individual domain and do not change with other domains, which is realized by applying the FFT. On the other hand, mutually invariant features can be acquired by using existing algorithms (i.e., CORAL and adversarial learning) to align different domains. Then, since different domain-invariant features are unified, their divergence can be maximized to guarantee their difference. The novelty of the proposed method is introduced by the diversity of the domain-invariant features, wherein the model's ability to generalize is boosted by considering both internally- and mutually invariant features. Two experimental analyses show the effectiveness of the proposed method.

(2) A pseudo domain label for characterizing dynamic feature distributions within each working condition

A new algorithm using pseudo domain labels is proposed to explore subdomain distributions within each subdomain at the domain level in Chapter 4 . The idea behind the proposed method is that the domain shifts caused by variable working conditions, like varying speeds, should also be considered since data may show a dynamic distribution of temporal features that are not limited to spatial distributions. That is, the original domain distribution can be further divided into several latent subdomains by introducing pseudo domain labels, which enables the proposed method to learn domain-specific features. By exploring features at the class and domain levels, the domain generalization capabilities of the model can be improved, thereby further increasing the accuracy of results. Experiments on two public bearing datasets show that the proposed method outperforms state-of-the-art methods.

(3) Perform effective generalizable fault diagnosis based on a domain interference suppression perspective

A two-step knowledge distillation training strategy is adopted based on a domain interference suppression perspective by regulating domain-specific features in Chapter 5 . The first step is designed to learn only domain-specific features, guided by contrastive learning, to capture unique characteristics embedded within each domain. Then, the second step is to remove specific learned features from the transferable features using adversarial learning. By aligning learned domain-specific features during knowledge distillation, the diversity of domain-specific features can be guaranteed. Additionally, by regulating domain-specific information from the extracted domain-invariant features, only transferable features are retained for inference. After geometrically encouraging orthogonality of the extracted domain-invariant and domain-specific features, feature decoupling can be satisfied. Hence, the model generalizes well to unseen operating conditions. Two studies on experimental datasets show that the proposed method provides satisfactory accuracy performance.

(4) A self-supervised and IF-free TFA framework

A deep learning-based end-to-end model named self-supervised TFA (SSTFA) is proposed in Chapter 6 , where the training label is the IF estimation result derived from the improved IF estimator, thus avoiding the usage of prior knowledge of true IFs. By incorporating a TF convolution module and an energy enhancement module, the proposed SSTFA model first generates a series of coarse TFRs by transforming the original one-dimensional time series data to a two-dimensional TF plane. Then, the energy enhancement module is used to guide the network to learn feature representations around the interested frequency components, thus a sharpened and clearer result closer to the ideal TFR is generated. The performance of the proposed SSTFA model in representing time-varying features is verified by analyzing both simulated and real-world data.

7.3 Comparisons

A comparison of the methods proposed for IFD under unseen working conditions is summarized in Table 7.1, including their advantages and disadvantages. They can be selected for applications according to the characteristics listed in the table. The methods proposed in Chapters 3-5 all focus on improving IFD under unseen working conditions. Although they share the same general objective, their assumptions, feature learning mechanisms, and practical limitations are different. Therefore, their differences should be discussed not only in terms of diagnostic performance, but also in terms of what type of domain discrepancy each method is designed to address.

Table 7.1. Comparison of methods proposed for IFD under unseen working conditions.

Methods	Main focus	Feature learning mechanism	Advantages	Limitations	Suitable scenarios
DIFE in Chapter 3	Diversity of domain-invariant features	Learns internally invariant and mutually invariant features, with KD for feature enhancement	Uses complementary invariant representations; improves transferable feature learning; relatively clear feature design	Domain-specific features are not directly modeled; extra computational cost is introduced by dual branches and KD	Cases with multiple source domains and moderate working condition discrepancies
Latent subdomain assignment in Chapter 4	Dynamic distribution modeling within each working condition	Introduces pseudo domain labels to divide each original domain into several latent subdomains	Considers both domain-invariant and domain-specific information; supports finer feature learning; better describes dynamic feature distributions	Requires centroid calculation and pseudo label assignment, with the optimal number of subdomains difficult to determine	Cases where working conditions vary continuously or contain fine-grained distribution changes
DIS in Chapter 5	Suppression of domain-specific interference	Learns domain-specific information first and then reduces its influence on transferable features through KD, adversarial learning, and feature separation constraints	Provides a more direct way to reduce domain-specific interference; encourages separation between transferable and domain-specific representations	Training is more complex; performance depends on the robustness of both teacher and student branches; additional computational cost is required	Cases with relatively strong domain discrepancies and sufficient source domain data for complex training

The DIFE method proposed in Chapter 3 focuses on the use of diverse domain-invariant features. Instead of relying on a single form of invariant representation, this method separates domain-invariant features into internally invariant features and mutually invariant features. Internally invariant features are extracted from each working condition, while mutually invariant features are obtained by reducing the discrepancy among different source domains. This design allows the model to use complementary information from both individual domain stability and cross-domain consistency. KD is further used to enhance feature learning. Therefore, DIFE is suitable when the main objective is to improve the diversity of transferable representations under multiple available source working conditions.

However, DIFE does not directly model domain-specific information. This may be sufficient when the discrepancy among source domains is moderate, but it can be less effective when the working conditions contain complex operating variations. In addition, the use of dual feature branches and KD introduces extra computational cost compared with a conventional single-branch diagnostic model.

The latent subdomain assignment method proposed in Chapter 4 further considers the dynamic feature distributions within each working condition. In many practical cases, one working condition may not be adequately described by a fixed distribution, especially when operating variables such as speed or load change continuously. To address this issue, pseudo domain labels are introduced to divide the original domain into several latent subdomains. In this way, the model can learn not only class-related features, but also finer domain-related variations inside each working condition.

Compared with DIFE, the method proposed in Chapter 4 extends the DANN framework by introducing pseudo domain labels, which allows more flexible domain partitioning beyond the original domain annotations. The results obtained under both constant speed and variable speed conditions suggest the necessity of analyzing dynamic distributions within a single working condition. Through adversarial alignment of these latent subdomains, the model is encouraged to learn fine-grained domain-invariant representations. Nevertheless, the estimation of latent subdomain centroids during training inevitably introduces additional computational cost, and the optimal number of latent subdomains is difficult to determine accurately.

The DIS method proposed in Chapter 5 studies the problem from the perspective of domain interference suppression. Different from Chapter 3, which emphasizes the diversity of invariant features, and Chapter 4, which characterizes latent domain structures, DIS directly considers the possible negative effect of domain-specific features on generalization. The basic assumption is that not all learned features are useful for unseen working conditions. Some features may be strongly

related to the source domains and may interfere with transferable fault representations. Therefore, DIS first learns domain-specific information and then suppresses its influence on the final transferable representation.

The main strength of DIS is that it provides a clearer feature separation strategy. By using KD, adversarial learning, and feature redundancy reduction constraints, the model attempts to separate domain-invariant features from domain-specific features. This is useful when the domain discrepancy is relatively strong and when domain-related components may dominate the learned representation. However, the method also has higher training complexity. Its performance depends on the stability of both the teacher and the student branches. If either branch fails to learn reliable representations, the suppression of domain-specific information may become less effective. Therefore, DIS is more suitable for cases where sufficient source domain data are available to support a relatively complex training strategy.

Overall, the three methods address different aspects of DG-based IFD. DIFE mainly improves the diversity of transferable features. Latent subdomain assignment focuses on dynamic feature distributions inside each working condition. DIS further suppresses domain-specific interference to obtain cleaner transferable representations. These methods are not simply stronger or weaker versions of each other. Their suitability depends on the distribution characteristics of the available source domains, the complexity of working condition changes, and the acceptable training cost. Specifically, DIFE can be considered when the working condition discrepancy is moderate, and the main requirement is to enhance invariant feature learning. Latent subdomain assignment is more appropriate when the operating conditions show continuous or fine-grained variations. DIS is more suitable when domain-specific components have a strong influence on the learned features and need to be reduced before inference.

Chapter 6 is different from the methods presented in Chapters 3 to 5. The first three methods are designed for fault classification under unseen working conditions, while Chapter 6 focuses on TFA for nonstationary signals. The proposed SSTFA method is not a DG-based classifier itself. Instead, it provides a self-supervised way to generate energy concentrated TFRs without using true IF labels. Therefore, it can serve as a signal representation method or a preprocessing tool for downstream diagnostic tasks. When combined with fault classification models, SSTFA may help construct more informative input representations, especially for signals with time-varying frequency characteristics. However, its contribution should be distinguished from the DG methods in Chapters 3 to 5, because its main objective is representation construction rather than direct cross-domain fault classification.

7.4 Future work

Potential future work is listed as follows:

(1) Single domain DG-based IFD:
This further limits the availability of the training dataset. Single domain DG indicates that only a single source domain is available.

(2) Feature distribution discrepancies across different sensors:
It is widely acknowledged that variable working conditions will result in different feature distributions. At the same time, data collected by using different sensors may also show distribution differences, thus making the integration of information across multiple sensors challenging.

(3) Adaptive TFA towards ideal TFRs:
To generate TFA results closer to ideal TFRs, model robustness against noise and interferences should be explored further.

References

[1] Y. Lei, B. Yang, X. Jiang, F. Jia, N. Li, A.K. Nandi, Applications of machine learning to machine fault diagnosis: A review and roadmap, Mech. Syst. Signal Process. 138 (2020) 106587. https://doi.org/10.1016/j.ymssp.2019.106587.

[2] W. Li, R. Huang, J. Li, Y. Liao, Z. Chen, G. He, R. Yan, K. Gryllias, A perspective survey on deep transfer learning for fault diagnosis in industrial scenarios: Theories, applications and challenges, Mech. Syst. Signal Process. 167 (2022) 108487. https://doi.org/10.1016/j.ymssp.2021.108487.

[3] S. Luo, X. Huang, Y. Wang, R. Luo, Q. Zhou, Transfer learning based on improved stacked autoencoder for bearing fault diagnosis, Knowl.-Based Syst. 256 (2022) 109846. https://doi.org/10.1016/j.knosys.2022.109846.

[4] R.K. Mishra, A. Choudhary, S. Fatima, A.R. Mohanty, B.K. Panigrahi, A systematic review on advancement and challenges in multi-fault diagnosis of rotating machines, Eng. Appl. Artif. Intell. 156 (2025) 111306. https://doi.org/10.1016/j.engappai.2025.111306.

[5] Z. Zhao, Q. Zhang, X. Yu, C. Sun, S. Wang, R. Yan, X. Chen, Applications of unsupervised deep transfer learning to intelligent fault diagnosis: A survey and comparative study, IEEE Trans. Instrum. Meas. 70 (2021) 1–28. https://doi.org/10.1109/TIM.2021.3116309.

[6] Q. Qian, J. Zhou, Y. Qin, Relationship transfer domain generalization network for rotating machinery fault diagnosis under different working conditions, IEEE Trans. Ind. Inform. 19 (2023) 9898–9908. https://doi.org/10.1109/TII.2022.3232842.

[7] Z. Feng, M. Liang, F. Chu, Recent advances in time-frequency analysis methods for machinery fault diagnosis: A review with application examples, Mech. Syst. Signal Process. 38 (2013) 165–205. https://doi.org/10.1016/j.ymssp.2013.01.017.

[8] T. Wang, F. Chu, Q. Han, Fault diagnosis for wind turbine planetary ring gear via a meshing resonance based filtering algorithm, ISA Trans. 67 (2017) 173–182. https://doi.org/10.1016/j.isatra.2016.11.008.

[9] T. Wang, F. Chu, Bearing fault diagnosis under time-varying rotational speed via the fault characteristic order (FCO) index based demodulation and the stepwise resampling in the fault phase angle (FPA) domain, ISA Trans. (2019). https://doi.org/10.1016/j.isatra.2019.04.020.

[10] T. Wang, M. Liang, J. Li, W. Cheng, Rolling element bearing fault diagnosis via fault characteristic order (FCO) analysis, Mech. Syst. Signal Process. 45 (2014) 139–153. https://doi.org/10.1016/j.ymssp.2013.11.011.

[11] L. Saidi, J. Ben Ali, F. Fnaiech, Si-spectrum based-EMD applied to the non-stationary vibration signals for bearing faults diagnosis, ISA Trans. 53 (2014) 1650–1660. https://doi.org/10.1016/j.isatra.2014.06.002.

[12] Z. Feng, M. Zuo, J. Qu, T. Tian, Z. Liu, Joint amplitude and frequency demodulation analysis based on local mean decomposition for fault diagnosis of planetary gearboxes,

Mech. Syst. Signal Process. 40 (2013) 56–75. https://doi.org/10.1016/j.ymssp.2013.05.016.

[13] Y. Berrouche, G. Vashishtha, S. Chauhan, R. Zimroz, Local damage detection in rolling element bearings based on a single ensemble empirical mode decomposition, Knowl.-Based Syst. 301 (2024) 112265. https://doi.org/10.1016/j.knosys.2024.112265.

[14] N. Rehman, H. Aftab, Multivariate Variational Mode Decomposition, IEEE Trans. Signal Process. 67 (2019) 6039–6052. https://doi.org/10.1109/TSP.2019.2951223.

[15] J. Chen, Z. Li, J. Pan, G. Chen, Y. Zi, J. Yuan, B. Chen, Z. He, Wavelet transform based on inner product in fault diagnosis of rotating machinery: A review, Mech. Syst. Signal Process. 70 (2016) 1–35. https://doi.org/10.1016/j.ymssp.2015.08.023.

[16] I. Daubechies, J. Lu, H.-T. Wu, Synchrosqueezed wavelet transforms: An empirical mode decomposition-like tool, Appl. Comput. Harmon. Anal. 30 (2011) 243–261. https://doi.org/10.1016/j.acha.2010.08.002.

[17] F. Auger, P. Flandrin, Y.-T. Lin, S. McLaughlin, S. Meignen, T. Oberlin, H.-T. Wu, Time-frequency reassignment and synchrosqueezing: An overview, IEEE Signal Process. Mag. 30 (2013) 32–41. https://doi.org/10.1109/MSP.2013.2265316.

[18] G. Yu, M. Yu, C. Xu, Synchroextracting transform, IEEE Trans. Ind. Electron. 64 (2017) 8042–8054. https://doi.org/10.1109/TIE.2017.2696503.

[19] G. Yu, Z. Wang, P. Zhao, Multisynchrosqueezing Transform, IEEE Trans. Ind. Electron. 66 (2019) 5441–5455. https://doi.org/10.1109/TIE.2018.2868296.

[20] Z. Hua, J. Shi, Y. Luo, W. Huang, J. Wang, Z. Zhu, Iterative matching synchrosqueezing transform and application to rotating machinery fault diagnosis under nonstationary conditions, Measurement 173 (2021) 108592. https://doi.org/10.1016/j.measurement.2020.108592.

[21] S. Lv, Y. Lv, R. Yuan, H. Li, High-order synchroextracting transform for characterizing signals with strong AM-FM features and its application in mechanical fault diagnosis, Mech. Syst. Signal Process. 172 (2022) 108959. https://doi.org/10.1016/j.ymssp.2022.108959.

[22] X. Li, W. Zhang, Q. Ding, J.-Q. Sun, Multi-Layer domain adaptation method for rolling bearing fault diagnosis, Signal Process. 157 (2019) 180–197. https://doi.org/10.1016/j.sigpro.2018.12.005.

[23] S. Schwendemann, Z. Amjad, A. Sikora, Bearing fault diagnosis with intermediate domain based layered maximum mean discrepancy: A new transfer learning approach, Eng. Appl. Artif. Intell. 105 (2021) 104415. https://doi.org/10.1016/j.engappai.2021.104415.

[24] T. Han, C. Liu, W. Yang, D. Jiang, Deep transfer network with joint distribution adaptation: A new intelligent fault diagnosis framework for industry application, ISA Trans. 97 (2020) 269–281. https://doi.org/10.1016/j.isatra.2019.08.012.

[25] P. Ma, H. Zhang, W. Fan, C. Wang, A diagnosis framework based on domain adaptation for bearing fault diagnosis across diverse domains, ISA Trans. 99 (2020) 465–478. https://doi.org/10.1016/j.isatra.2019.08.040.

[26] Y. Feng, J. Chen, Z. Yang, X. Song, Y. Chang, S. He, E. Xu, Z. Zhou, Similarity-based meta-learning network with adversarial domain adaptation for cross-domain fault identification, Knowl.-Based Syst. 217 (2021) 106829. https://doi.org/10.1016/j.knosys.2021.106829.

[27] Y. Xu, L. Chen, F. Zhang, S. Wang, J. Shi, C. Shen, Multi-source alignment domain adaptation with similarity measurement for cross-domain bearing fault diagnosis, Meas. Sci. Technol. 34 (2023) 055006. https://doi.org/10.1088/1361-6501/acb6e2.

[28] X. Zhang, W. Huang, C. Ding, J. Wang, C. Shen, J. Shi, Cross-Supervised multisource prototypical network: A novel domain adaptation method for multi-source few-shot fault diagnosis, Adv. Eng. Inform. 61 (2024) 102538. https://doi.org/10.1016/j.aei.2024.102538.

[29] Z. Ren, Y. Zhu, K. Yan, K. Chen, W. Kang, Y. Yue, D. Gao, A novel model with the ability of few-shot learning and quick updating for intelligent fault diagnosis, Mech. Syst. Signal Process. 138 (2020) 106608. https://doi.org/10.1016/j.ymssp.2019.106608.

[30] D. Gao, Y. Zhu, K. Yan, H. Fu, Z. Ren, W. Kang, C. Guedes Soares, Joint learning system based on semi–pseudo–label reliability assessment for weak–fault diagnosis with few labels, Mech. Syst. Signal Process. 189 (2023) 110089. https://doi.org/10.1016/j.ymssp.2022.110089.

[31] Z. Wu, Z. Xu, W. Fan, F. Poulhaon, P. Michaud, P. Joyot, Semi-supervised multi-label feature selection algorithm for online monitoring of laser metal deposition manufacturing quality, Measurement 219 (2023) 113301. https://doi.org/10.1016/j.measurement.2023.113301.

[32] H. Ren, J. Wang, W. Huang, X. Jiang, Z. Zhu, Domain-invariant feature fusion networks for semi-supervised generalization fault diagnosis, Eng. Appl. Artif. Intell. 126 (2023) 107117. https://doi.org/10.1016/j.engappai.2023.107117.

[33] Q. Li, L. Chen, L. Kong, D. Wang, M. Xia, C. Shen, Cross-domain augmentation diagnosis: An adversarial domain-augmented generalization method for fault diagnosis under unseen working conditions, Reliab. Eng. Syst. Saf. 234 (2023) 109171. https://doi.org/10.1016/j.ress.2023.109171.

[34] Z. Wang, X. Zhang, Z. Li, F. Chen, Domain generalization based on domain-specific adversarial learning, Appl. Intell. 54 (2024) 4878–4889. https://doi.org/10.1007/s10489-024-05423-z.

[35] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: 2016 IEEE Conf. Comput. Vis. Pattern Recognit. CVPR, 2016: pp. 770–778. https://doi.org/10.1109/CVPR.2016.90.

[36] Z. Hua, P. Dumond, A structural re-parameterization network for bearing fault diagnosis under variable working conditions, in: American Society of Mechanical Engineers Digital Collection, 2024. https://doi.org/10.1115/DETC2024-143171.

[37] Q. Li, L. Chen, L. Kong, D. Wang, M. Xia, C. Shen, Cross-domain augmentation diagnosis: An adversarial domain-augmented generalization method for fault diagnosis under unseen

working conditions, Reliab. Eng. Syst. Saf. 234 (2023) 109171. https://doi.org/10.1016/j.ress.2023.109171.

[38] W. Cheng, X. Liu, J. Xing, X. Chen, B. Ding, R. Zhang, K. Zhou, Q. Huang, AFARN: Domain adaptation for intelligent cross-domain bearing fault diagnosis in nuclear circulating water pump, IEEE Trans. Ind. Inform. 19 (2023) 3229–3239. https://doi.org/10.1109/TII.2022.3177459.

[39] X. Yu, Y. Wang, Z. Liang, H. Shao, K. Yu, W. Yu, An adaptive domain adaptation method for rolling bearings' fault diagnosis fusing deep convolution and self-attention networks, IEEE Trans. Instrum. Meas. 72 (2023) 1–14. https://doi.org/10.1109/TIM.2023.3246494.

[40] T. Li, Z. Zhao, C. Sun, R. Yan, X. Chen, Domain adversarial graph convolutional network for fault diagnosis under variable working conditions, IEEE Trans. Instrum. Meas. 70 (2021) 1–10. https://doi.org/10.1109/TIM.2021.3075016.

[41] J. Zhu, N. Chen, C. Shen, New multiple source domain adaptation fault diagnosis method between different rotating machines, IEEE Trans. Ind. Inform. 17 (2021) 4788–4797. https://doi.org/10.1109/TII.2020.3021406.

[42] Z. Wu, H. Jiang, H. Zhu, X. Wang, A knowledge dynamic matching unit-guided multi-source domain adaptation network with attention mechanism for rolling bearing fault diagnosis, Mech. Syst. Signal Process. 189 (2023) 110098. https://doi.org/10.1016/j.ymssp.2023.110098.

[43] X. Cong, Y. Song, Y. Li, L. Jia, Federated domain generalization with global robust model aggregation strategy for bearing fault diagnosis, Meas. Sci. Technol. 34 (2023) 115116. https://doi.org/10.1088/1361-6501/ace841.

[44] Z. Fan, Q. Xu, C. Jiang, S.X. Ding, Deep mixed domain generalization network for intelligent fault diagnosis under unseen conditions, IEEE Trans. Ind. Electron. 71 (2024) 965–974. https://doi.org/10.1109/TIE.2023.3243293.

[45] H. Wu, J. Li, Q. Zhang, J. Tao, Z. Meng, Intelligent fault diagnosis of rolling bearings under varying operating conditions based on domain-adversarial neural network and attention mechanism, ISA Trans. 130 (2022) 477–489. https://doi.org/10.1016/j.isatra.2022.04.026.

[46] J. Zhong, Y. Huang, Time-frequency representation based on an adaptive short-time Fourier transform, IEEE Trans. Signal Process. 58 (2010) 5118–5128. https://doi.org/10.1109/tsp.2010.2053028.

[47] G. Hinton, O. Vinyals, J. Dean, Distilling the knowledge in a neural network, (2015). https://doi.org/10.48550/arXiv.1503.02531.

[48] R. Li, S. Li, K. Xu, X. Li, J. Lu, M. Zeng, M. Li, J. Du, Adversarial domain adaptation of asymmetric mapping with CORAL alignment for intelligent fault diagnosis, Meas. Sci. Technol. 33 (2022) 055101. https://doi.org/10.1088/1361-6501/ac3d47.

[49] X. Wang, C. Shen, M. Xia, D. Wang, J. Zhu, Z. Zhu, Multi-scale deep intra-class transfer learning for bearing fault diagnosis, Reliab. Eng. Syst. Saf. 202 (2020) 107050. https://doi.org/10.1016/j.ress.2020.107050.

[50] Y. Ma, J. Yang, L. Li, Gradient aligned domain generalization with a mutual teaching teacher-student network for intelligent fault diagnosis, Reliab. Eng. Syst. Saf. 239 (2023) 109516. https://doi.org/10.1016/j.ress.2023.109516.

[51] H. Huang, N. Baddour, Bearing vibration data collected under time-varying rotational speed conditions, Data Brief 21 (2018) 1745–1749. https://doi.org/10.1016/j.dib.2018.11.019.

[52] Z. Shi, J. Chen, Y. Zi, Z. Zhou, A novel multitask adversarial network via redundant lifting for multicomponent intelligent fault detection under sharp speed variation, IEEE Trans. Instrum. Meas. 70 (2021) 1–10. https://doi.org/10.1109/TIM.2021.3055821.

[53] R. Li, S. Li, K. Xu, J. Lu, G. Teng, J. Du, Deep domain adaptation with adversarial idea and coral alignment for transfer fault diagnosis of rolling bearing, Meas. Sci. Technol. 32 (2021) 094009. https://doi.org/10.1088/1361-6501/abe163.

[54] J. Shi, Z. Hua, P. Dumond, Z. Zhu, W. Huang, C. Shen, Refined matching linear chirplet transform for exhibiting time-frequency features of nonstationary vibration and acoustic signals, Measurement 187 (2022) 110298. https://doi.org/10.1016/j.measurement.2021.110298.

[55] H. Pu, K. Zhang, Y. An, Restricted sparse networks for rolling bearing fault diagnosis, IEEE Trans. Ind. Inform. 19 (2023) 11139–11149. https://doi.org/10.1109/TII.2023.3243929.

[56] D. Zhao, S. Liu, H. Du, L. Wang, Z. Miao, Deep branch attention network and extreme multi-scale entropy based single vibration signal-driven variable speed fault diagnosis scheme for rolling bearing, Adv. Eng. Inform. 55 (2023) 101844. https://doi.org/10.1016/j.aei.2022.101844.

[57] Y. Li, L. Zou, L. Jiang, X. Zhou, Fault diagnosis of rotating machinery based on combination of deep belief network and one-dimensional convolutional neural network, IEEE Access 7 (2019) 165710–165723. https://doi.org/10.1109/ACCESS.2019.2953490.

[58] G. Vashishtha, S. Chauhan, S. Kumar, R. Kumar, R. Zimroz, A. Kumar, Intelligent fault diagnosis of worm gearbox based on adaptive CNN using amended gorilla troop optimization with quantum gate mutation strategy, Knowl.-Based Syst. 280 (2023) 110984. https://doi.org/10.1016/j.knosys.2023.110984.

[59] X. Wang, C. Shen, M. Xia, D. Wang, J. Zhu, Z. Zhu, Multi-scale deep intra-class transfer learning for bearing fault diagnosis, Reliab. Eng. Syst. Saf. 202 (2020) 107050. https://doi.org/10.1016/j.ress.2020.107050.

[60] I.M. Kamal, H. Bae, Semi-supervised binary classification with latent distance learning, Adv. Eng. Inform. 61 (2024) 102441. https://doi.org/10.1016/j.aei.2024.102441.

[61] C. Shen, X. Wang, D. Wang, Y. Li, J. Zhu, M. Gong, Dynamic joint distribution alignment network for bearing fault diagnosis under variable working conditions, IEEE Trans. Instrum. Meas. 70 (2021) 1–13. https://doi.org/10.1109/TIM.2021.3055786.

[62] M. Shi, C. Ding, R. Wang, C. Shen, W. Huang, Z. Zhu, Graph embedding deep broad learning system for data imbalance fault diagnosis of rotating machinery, Reliab. Eng. Syst. Saf. 240 (2023) 109601. https://doi.org/10.1016/j.ress.2023.109601.

[63] H. Chen, J. Wei, H. Huang, L. Wen, Y. Yuan, J. Wu, Novel imbalanced fault diagnosis method based on generative adversarial networks with balancing serial CNN and Transformer (BCTGAN), Expert Syst. Appl. 258 (2024) 125171. https://doi.org/10.1016/j.eswa.2024.125171.

[64] W. Li, X. Zhong, H. Shao, B. Cai, X. Yang, Multi-mode data augmentation and fault diagnosis of rotating machinery using modified ACGAN designed with new framework, Adv. Eng. Inform. 52 (2022) 101552. https://doi.org/10.1016/j.aei.2022.101552.

[65] Y. Xiao, H. Shao, J. Wang, B. Cai, B. Liu, Domain-augmented meta ensemble learning for mechanical fault diagnosis from heterogeneous source domains to unseen target domains, Expert Syst. Appl. 259 (2025) 125345. https://doi.org/10.1016/j.eswa.2024.125345.

[66] X. Ren, S. Wang, W. Zhao, X. Kong, M. Fan, H. Shao, K. Zhao, Universal federated domain adaptation for gearbox fault diagnosis: A robust framework for credible pseudo-label generation, Adv. Eng. Inform. 65 (2025) 103233. https://doi.org/10.1016/j.aei.2025.103233.

[67] Y. Xiao, H. Shao, S. Yan, J. Wang, Y. Peng, B. Liu, Domain generalization for rotating machinery fault diagnosis: A survey, Adv. Eng. Inform. 64 (2025) 103063. https://doi.org/10.1016/j.aei.2024.103063.

[68] Q. Song, X. Jiang, J. Liu, J. Shi, Z. Zhu, Contrast-Assisted domain-specificity-removal network for semi-supervised generalization fault diagnosis, IEEE Trans. Neural Netw. Learn. Syst. (2024) 1–14. https://doi.org/10.1109/TNNLS.2024.3383467.

[69] Y. Shi, A. Deng, M. Deng, M. Xu, Y. Liu, X. Ding, W. Bian, Domain augmentation generalization network for real-time fault diagnosis under unseen working conditions, Reliab. Eng. Syst. Saf. 235 (2023) 109188. https://doi.org/10.1016/j.ress.2023.109188.

[70] D. Gao, K. Huang, Y. Zhu, L. Zhu, K. Yan, Z. Ren, C. Guedes Soares, Semi-supervised small sample fault diagnosis under a wide range of speed variation conditions based on uncertainty analysis, Reliab. Eng. Syst. Saf. 242 (2024) 109746. https://doi.org/10.1016/j.ress.2023.109746.

[71] T. Han, Y.-F. Li, M. Qian, A Hybrid Generalization Network for Intelligent Fault Diagnosis of Rotating Machinery Under Unseen Working Conditions, IEEE Trans. Instrum. Meas. 70 (2021) 1–11. https://doi.org/10.1109/TIM.2021.3088489.

[72] J. Shi, M. Liang, Intelligent bearing fault signature extraction via iterative oscillatory behavior based signal decomposition (IOBSD), Expert Syst. Appl. 45 (2016) 40–55. https://doi.org/10.1016/j.eswa.2015.09.039.

[73] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, V. Lempitsky, Domain-adversarial training of neural networks, J. Mach. Learn. Res. 17 (2016) 2096–2130. https://dl.acm.org/doi/10.5555/2946645.2946704.

[74] J. Sun, J. Wen, C. Yuan, Z. Liu, Q. Xiao, Bearing fault diagnosis based on multiple transformation domain Fusion and improved residual dense networks, IEEE Sens. J. 22 (2022) 1541–1551. https://doi.org/10.1109/JSEN.2021.3131722.

[75] G. Zhang, X. Kong, Q. Wang, J. Du, J. Wang, H. Ma, Single domain generalization method

based on anti-causal learning for rotating machinery fault diagnosis, Reliab. Eng. Syst. Saf. 250 (2024) 110252. https://doi.org/10.1016/j.ress.2024.110252.

[76] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial networks, Commun ACM 63 (2020) 139–144. https://doi.org/10.1145/3422622.

[77] C. Lessmeier, J.K. Kimotho, D. Zimmer, W. Sextro, Condition Monitoring of Bearing Damage in Electromechanical Drive Systems by Using Motor Current Signals of Electric Motors: A Benchmark Data Set for Data-Driven Classification, PHM Soc. Eur. Conf. 3 (2016). https://doi.org/10.36001/phme.2016.v3i1.1577.

[78] V.K. Kurmi, V.K. Subramanian, V.P. Namboodiri, Informative discriminator for domain adaptation, Image Vis. Comput. 111 (2021) 104180. https://doi.org/10.1016/j.imavis.2021.104180.

[79] H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, Domain-adversarial neural networks, (2015). https://doi.org/10.48550/arXiv.1412.4446.

[80] L. Zhang, W. Cheng, S. Zhang, J. Xing, Z. Nie, X. Chen, D. Lan, Y. Liu, Y. Yang, Z. Pang, How large AI model empowers time-series forecasting for the operation and maintenance of industrial automation system?, IEEE Trans. Ind. Inform. 21 (2025) 8201–8213. https://doi.org/10.1109/TII.2025.3575118.

[81] B. Li, Q. Li, T. Du, D. Liu, Q. Yang, T. Chen, J. Xiong, B. Peng, J. Ren, J. Zhao, Research, application, and challenges of causal inference in industrial fault diagnosis: A survey, Eng. Appl. Artif. Intell. 158 (2025) 111376. https://doi.org/10.1016/j.engappai.2025.111376.

[82] L. Alzubaidi, J. Zhang, A.J. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, J. Santamaría, M.A. Fadhel, M. Al-Amidie, L. Farhan, Review of deep learning: concepts, CNN architectures, challenges, applications, future directions, J. Big Data 8 (2021) 53. https://doi.org/10.1186/s40537-021-00444-8.

[83] C. Yang, J. Liu, Y. Hu, B. Wu, T. Shi, Dynamic graph-driven rotating machine fault diagnosis: An adaptively updating cross-domain relationship information, IEEE Trans. Ind. Inform. 20 (2024) 14479–14488. https://doi.org/10.1109/TII.2024.3454065.

[84] Q. Jiang, X. Lin, X. Lu, Y. Shen, Q. Zhu, Q. Zhang, Self-supervised learning-based dual-classifier domain adaptation model for rolling bearings cross-domain fault diagnosis, Knowl.-Based Syst. 284 (2024) 111229. https://doi.org/10.1016/j.knosys.2023.111229.

[85] C. Wang, H. Jie, J. Yang, T. Gao, Z. Zhao, Y. Chang, K.Y. See, A multi-source domain feature-decision dual fusion adversarial transfer network for cross-domain anti-noise mechanical fault diagnosis in sustainable city, Inf. Fusion 115 (2025) 102739. https://doi.org/10.1016/j.inffus.2024.102739.

[86] K. Zhao, F. Jia, H. Shao, A novel conditional weighting transfer Wasserstein auto-encoder for rolling bearing fault diagnosis with multi-source domains, Knowl.-Based Syst. 262 (2023) 110203. https://doi.org/10.1016/j.knosys.2022.110203.

[87] J. Wei, Q. Wang, G. Zhang, H. Ma, Y. Wang, Domain knowledge guided pseudo-label

generation framework for semi-supervised domain generalization fault diagnosis, Adv. Eng. Inform. 67 (2025) 103540. https://doi.org/10.1016/j.aei.2025.103540.

[88] Z. Hua, J. Shi, P. Dumond, Latent subdomain assignment based on pseudo domain labels for fault diagnosis of unseen data, Adv. Eng. Inform. 67 (2025) 103526. https://doi.org/10.1016/j.aei.2025.103526.

[89] Z. Liu, H. Zheng, H. Liu, G. Duan, J. Tan, A novel domain feature disentanglement method for multi-target cross-domain mechanical fault diagnosis, ISA Trans. (2025). https://doi.org/10.1016/j.isatra.2025.01.012.

[90] S. Xie, P. Xia, H. Zhang, Domain adaptation with domain-specific information and feature disentanglement for bearing fault diagnosis, Meas. Sci. Technol. 35 (2024) 056101. https://doi.org/10.1088/1361-6501/ad20c3.

[91] J. Xu, Y. Zhao, W. Bao, C. Hao, Fault diagnosis of motor bearing in complex scenarios based on Mamba and Indicative Contrastive Learning, Eng. Appl. Artif. Intell. 146 (2025) 110216. https://doi.org/10.1016/j.engappai.2025.110216.

[92] Z. Hua, J. Shi, P. Dumond, Domain-invariant feature exploration for intelligent fault diagnosis under unseen and time-varying working conditions, Mech. Syst. Signal Process. 224 (2025) 112193. https://doi.org/10.1016/j.ymssp.2024.112193.

[93] T. Wang, X. Dai, Y. Liu, Learning with Hilbert–Schmidt independence criterion: A review and new perspectives, Knowl.-Based Syst. 234 (2021) 107567. https://doi.org/10.1016/j.knosys.2021.107567.

[94] Y. Chen, J. Shi, C. Shen, H. Yang, Z. Hua, W. Huang, Z. Zhu, Time-frequency aware feature disentanglement learning for intelligent bearing fault diagnosis under variable speed conditions, Expert Syst. Appl. 303 (2026) 130664. https://doi.org/10.1016/j.eswa.2025.130664.

[95] E. Sejdic, I. Djurovic, J. Jiang, Time-frequency feature representation using energy concentration: An overview of recent advances, Digit. Signal Process. 19 (2009) 153–183. https://doi.org/10.1016/j.dsp.2007.12.004.

[96] F. Auger, P. Flandrin, Improving the readability of time-frequency and time-scale representations by the reassignment method, IEEE Trans. Signal Process. 43 (1995) 1068–1089. https://doi.org/10.1109/78.382394.

[97] J. Shi, M. Liang, D.-S. Necsulescu, Y. Guan, Generalized stepwise demodulation transform and synchrosqueezing for time-frequency analysis and bearing fault diagnosis, J. Sound Vib. 368 (2016) 202–222. https://doi.org/10.1016/j.jsv.2016.01.015.

[98] J. Shi, Z. Hua, P. Dumond, Z. Zhu, W. Huang, C. Shen, Refined matching linear chirplet transform for exhibiting time-frequency features of nonstationary vibration and acoustic signals, Measurement 187 (2022) 110298. https://doi.org/10.1016/j.measurement.2021.110298.

[99] S. Wang, X. Chen, G. Cai, B. Chen, X. Li, Z. He, Matching demodulation transform and synchrosqueezing in time-frequency analysis, IEEE Trans. Signal Process. 62 (2014) 69–84.

https://doi.org/10.1109/tsp.2013.2276393.

[100] G. Yu, Y. Zhou, General linear chirplet transform, Mech. Syst. Signal Process. 70 (2016) 958–973. https://doi.org/10.1016/j.ymssp.2015.09.004.

[101] J. Zheng, H. Pan, S. Yang, J. Cheng, Adaptive parameterless empirical wavelet transform based time-frequency analysis method and its application to rotor rubbing fault diagnosis, Signal Process. 130 (2017) 305–314. https://doi.org/10.1016/j.sigpro.2016.07.023.

[102] P. Zhou, X. Dong, S. Chen, Z. Peng, W. Zhang, Parameterized model based Short-time chirp component decomposition, Signal Process. 145 (2018) 146–154. https://doi.org/10.1016/j.sigpro.2017.12.007.

[103] C. Li, V. Sanchez, G. Zurita, M.C. Lozada, D. Cabrera, Rolling element bearing defect detection using the generalized synchrosqueezing transform guided by time–frequency ridge enhancement, ISA Trans. 60 (2016) 274–284. https://doi.org/10.1016/j.isatra.2015.10.014.

[104] H.S. Razzaq, Z.M. Hussain, Instantaneous frequency estimation of FM Signals under Gaussian and symmetric α-stable noise: Deep learning versus time–frequency analysis, Information 14 (2023) 18. https://doi.org/10.3390/info14010018.

[105] B.M. Krishna, S.V.V. Satyanarayana, K. Baboji, SeismoNet: A deep learning approach for time–frequency analysis of seismic data, IEEE Trans. Geosci. Remote Sens. 63 (2025) 1–14. https://doi.org/10.1109/TGRS.2025.3599523.

[106] J. Qian, S. Huang, L. Wang, G. Bi, X. Yang, Super-Resolution ISAR imaging for maneuvering target based on deep-learning-assisted time–frequency analysis, IEEE Trans. Geosci. Remote Sens. 60 (2022) 1–14. https://doi.org/10.1109/TGRS.2021.3050189.

[107] Q. Chen, X. Dong, G. Tu, D. Wang, C. Cheng, B. Zhao, Z. Peng, TFN: An interpretable neural network with time-frequency transform embedded for intelligent fault diagnosis, Mech. Syst. Signal Process. 207 (2024) 110952. https://doi.org/10.1016/j.ymssp.2023.110952.

[108] P. Pan, Y. Zhang, Z. Deng, S. Fan, X. Huang, TFA-Net: A Deep Learning-Based Time-Frequency Analysis Tool, IEEE Trans. Neural Netw. Learn. Syst. 34 (2023) 9274–9286. https://doi.org/10.1109/TNNLS.2022.3157723.

[109] D. Zhao, D. Shao, L. Cui, CTNet: A data-driven time-frequency technique for wind turbines fault diagnosis under time-varying speeds, ISA Trans. 154 (2024) 335–351. https://doi.org/10.1016/j.isatra.2024.08.029.