

# Tensor Representation-based Transferability Analytics and Selective Transfer Learning of Prognostic Knowledge for Remaining Useful Life Prediction Across Machines

Wentao Mao<sup>a,b</sup>, Wen Zhang<sup>a</sup>, Ke Feng<sup>c,\*</sup>, Michael Beer<sup>d,e,f</sup> and Chunsheng Yang<sup>g</sup>

<sup>a</sup>School of Computer and Information Engineering, Henan Normal University, Xinxiang, China, Xinxiang, 453007, China

<sup>b</sup>Engineering Lab of Intelligence Business & Internet of Things of Henan Province, Xinxiang, 453007, China

<sup>c</sup>Department of Industrial Systems Engineering and Management, National University of Singapore, 117576, Singapore

<sup>d</sup>Institute for Risk and Reliability, Leibniz University Hannover, Callinstr. 34, Hannover, 30167, Germany

<sup>e</sup>Institute for Risk and Uncertainty, University of Liverpool, Peach Street, Liverpool, L69 7ZF, United Kingdom

<sup>f</sup>Department of Civil Engineering, Tsinghua University, Beijing, 100190, China

<sup>g</sup>Institute of Artificial Intelligence, Guangzhou University, Guangzhou, 510006, China

---

## ARTICLE INFO

### Keywords:

Remaining useful life prediction  
Transfer learning  
Transferability analytics  
LSTM  
Tensor decomposition

## ABSTRACT

In recent years, deep transfer learning techniques have been successfully applied to solve RUL prediction across different working conditions. However, for RUL prediction across different machines in which the data distribution and fault evolution characteristics vary largely, the extraction and transition of prognostic knowledge become more challenging. Even if fault mode information can assist in the knowledge transfer, model bias will inevitably exist on the target machine with mixed or unknown faults. To address this issue from a transferability perspective, this paper proposes a novel selective transfer learning approach for RUL prediction across machines. First, the paper utilizes the tensor representation to construct the meta-degradation trend of each fault mode and evaluates the transferability of source domain data from fault mode and degradation characteristics through a new cross-machine transfer degree indicator (*M-TDI*). Second, a Long Short-Term Memory (LSTM)-based selective transfer strategy is proposed using the *M-TDIs*. The paper designs a training algorithm with an alternating optimization scheme to seek the optimal tensor decomposition and knowledge transfer effect. Theoretical analysis proves that the proposed approach significantly reduces the upper bound of prediction error. Furthermore, experimental results on three benchmark datasets prove the effectiveness of the proposed approach.

---


## 1. Introduction

The evaluation of a machine's failure time, known as remaining useful life (RUL), has been a noticeable research topic in the field of prognostics and health management [1]. In the past decade, with the rapid development of machine learning, various algorithms have been employed for model construction, including shallow models like support vector machine (SVM) [2], Gaussian process regression (GPR) [1], as well as deep neural networks such as deep belief net (DBN) [3], long short-term memory (LSTM) [4], and convolutional neural network (CNN) [5].

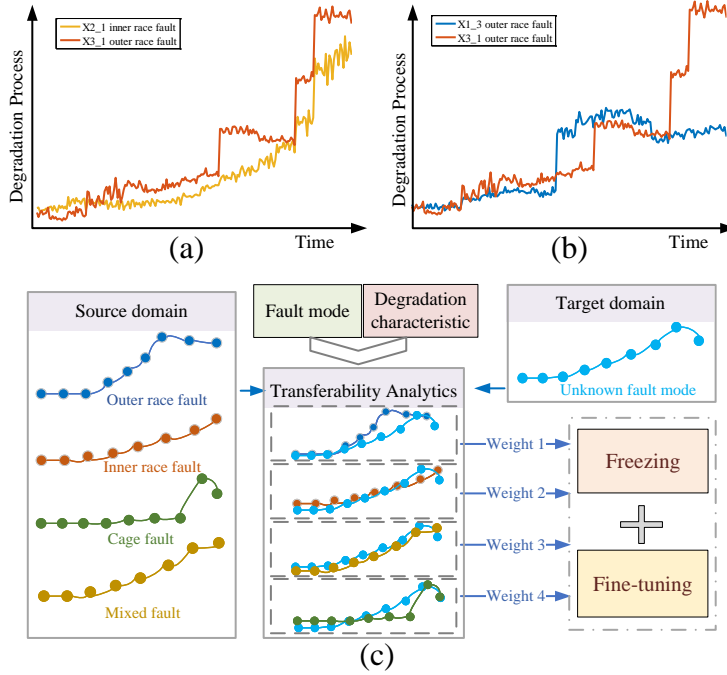
Recently, transfer learning techniques have been introduced to address the RUL prediction problem across different working conditions, also known as RUL transfer prediction [6; 7]. For a detailed survey, please refer to Section 2. Aiming to evaluate how much prognostic knowledge from source domain data is beneficial for target domain tasks, transferability is becoming a crucial issue in RUL transfer prediction. However, different from fault diagnosis, transferability analytics for RUL prediction have unique requirements. Specifically, temporal degradation characteristics are more significant than discriminative characteristics, and the negative impact of randomness in the degradation process should be reduced when extracting prognostic knowledge. Existing methods, such as those presented in [5; 7], assume that machines with the same manufacturing specifications have similar degradation processes when operating under different working conditions. This assumption enables the transfer of prognostic knowledge. However, RUL prediction across different machines or platforms, which is more practically significant,

---

\*Corresponding author

 ke.feng@outlook.com.au (K. Feng)

ORCID(s):



**Figure 1:** Schematic diagram of RUL transfer prediction across different machines, where (a) and (b) are the degradation processes with similar shapes but different fault modes, and with the same fault mode but dissimilar shapes, respectively; (c) is the sketch map of selective knowledge transfer proposed in this paper. Here the XJTU-SY rolling bearing dataset is chosen as an example.

poses greater challenges due to the greater discrepancy in degradation characteristics compared to those across working conditions. For instance, for complex machines like a shield or high-speed train, it is not easy to collect actual run-to-failure data. Manufacturers usually collect whole-life data on test machines in a factory or laboratory to estimate performance. Nevertheless, since test machines are a simplification of real-world machines in terms of boundary conditions and models, their degradation processes may differ significantly in terms of data scale and degradation characteristics. As a result, evaluating the transferability across different machines is becoming a significant concern.

Although some previous works [8; 9] attempted to address this problem by obtaining degradation characteristics (such as geometric shape and tendency) from monitoring data, they failed to consider the intrinsic degradation mechanism, such as fault mode information. As depicted in Figure 1(a), different fault modes may have similar degradation trends, but their essential prognostic knowledge would differ. In addition, this study [10] that utilized fault mode information to improve RUL prediction, but it is not appropriate for RUL prediction across machines where the target machine may have mixed or unknown fault modes. Moreover, due to the drift of working conditions, the same fault mode may result in degradation processes with noticeably different shapes, as demonstrated in Figure 1(b). Hence, using only fault mode as a transferability measure can lead to negative transfer, causing a deviation in knowledge transfer.

Based on the aforementioned analysis, the main challenges of RUL transfer prediction across machines can be summarized as follows: 1) How to evaluate the transferability of data with large distribution divergence; and 2) How to build a knowledge transfer channel when the target machine’s fault mode is unknown. To tackle these challenges, the transferability is evaluated in this paper from two aspects: degradation characteristic and fault mode. Then a selective transfer learning model is constructed to transfer the prognostic knowledge using the transferability analysis, as illustrated in Figure 1(c).

Specifically, the technique of tensor representation is introduced to facilitate the transferability analysis. Tensor tucker decomposition is able to extract the intrinsic information from the original data, which helps to represent the common degradation trend of each fault mode, named by meta-degradation trend. The geometry and tendency similarity between the degradation sequences of the two domains can be also calculated to evaluate transferability. The

proposed selective transfer learning model uses weighted initialization and adaptive freezing to adaptively transfer prognostic knowledge. Theoretical analysis shows that the proposed model can significantly reduce the upper bound of prediction error on the target task.

To validate our approach, rolling element bearings are taken as the test object and set a series of cross-machine prediction tasks using three benchmark datasets: the XJTU-SY bearing dataset, the IEEE PHM Challenge 2012 bearing dataset (PHM for short), and the University of New South Wales (UNSW for short) bearing dataset. These three datasets are all from run-to-failure experiments, in which the fault modes in XJTU-SY and UNSW are provided and the fault mode in PHM is unknown. The experimental results demonstrate the rationality of the proposed transferability metric and the effectiveness of the proposed approach. The implementation code of our approach can be found on GitHub (<https://github.com/unikz22/Selective-transfer-learning-based-on-tensor-representation>).

The main novelty and contributions of this research work can be summarized as follows:

- The paper introduces a novel metric, *M-TDI*, for assessing the transferability of RUL predictions based on tensor representations. Unlike existing methods, *M-TDI* offers a dynamic evaluation of the relevance of prognostic knowledge in the context of alternate optimization. Notably, this metric demonstrates robust performance in handling degradation randomness. To the best of our knowledge, this is the pioneering work in exploring transferability analytics within the realm of RUL prediction.
- This paper introduces an innovative method for predicting RUL transfer across different machines. Leveraging a transferability evaluation, the proposed approach can effectively transfer knowledge despite significant disparities in data distribution. Remarkably, this method performs admirably even when confronted with situations where the target machine’s fault mode is absent, underscoring its substantial potential for practical application and deployment.
- This paper establishes an upper limit on prediction errors for the proposed approach. This upper limit is a theoretical guarantee that the approach will enhance the reliability of RUL transfer learning. Furthermore, this bound provides robust support for the rationale behind regression transfer learning with fine-tuning. To the best of our knowledge, this study represents the initial endeavor to provide a theoretical analysis of the reliability of RUL transfer learning.

The remaining sections of this paper are organized as follows. In Section 2, a thorough analysis of related works on RUL transfer prediction is presented. Section 3 introduces the proposed approach, including the tensor representation and transferability analysis. The selective transfer learning model with weighted initialization and adaptive freezing is also described. Section 4 presents the experimental results and comparisons to demonstrate the effectiveness of the proposed approach. Finally, Section 5 concludes this paper and outlines potential future research directions.

## 2. Preliminary works

Deep transfer learning has been proven promising for RUL prediction across different working conditions, by learning and transferring prognostic knowledge from one distribution (called source domain) to a related but different task (called target domain). With end-to-end modeling capability, this approach has been explored in various transfer strategies, such as domain-adversarial training [11], feature adaptation [12], and parameter fine-tuning [13], to address the domain shift problem between different working conditions. However, these methods do not apply to the cross-machine scenario since a large distribution divergence can hinder the extraction of domain-invariant feature representation. In the last year, the RUL transfer prediction across different machines has started to gain attention and new research is emerging. Zhu et al. [14] designed an active querying-based transfer learning strategy on Bayesian deep learning framework to relieve data distribution discrepancy from different machines. Deng et al. [15] designed a calibrated-based hybrid transfer learning framework by combining physical model parameters into adversarial learning to transfer the most informative knowledge across different machines. By setting the cross-machine task in an online scenario, Mao et al. [16] proposed a self-supervised deep regression adaptation for online RUL prediction. This paper first built a pre-training model with adversarial training and then extracted tendency information from sequentially-collected online data to reduce domain shift. Although these works all introduced extra information to facilitate the domain adaptation with large distribution divergence, the used information is too less comprehensive to support reliable transfer learning. More importantly, these works are just with algorithmic study, without a theoretical guarantee on the

transfer effect. Degradation knowledge from different dimensions or scales is demanded for solving RUL prediction across different machines.

The degradation mechanism can provide valuable information for RUL prediction. For example, Xia et al. [10] integrated fault mode information into a convolutional LSTM ensemble network for RUL prediction, which allowed for weakly-supervised domain adaptation and the learning of degradation patterns for different fault modes. However, this approach assumes that the fault mode of the target domain is known and related to the source domain, which may not be the case in practical applications. Furthermore, the approach does not refine the prognostic knowledge during prediction. In another study, Liu et al. [17] proposed a multi-task learning method for fault mode identification and RUL prediction. However, the fault mode information was only used to reduce the risk of overfitting in RUL prediction, and was not coupled with the extraction of fault knowledge. The role of fault mode information in RUL transfer prediction, particularly in cross-machine scenarios, requires further investigation.

Various metrics and techniques have been introduced to evaluate transferability in recent years. Dong et al. [18] pioneered this area by developing the knowledge aggregation-induced transferability perception adaptation network (KATPAN) to determine where and how to transfer. To achieve better marginal and conditional distribution alignment between different domains, Hu et al. [19] theoretically analyzed unbiased transferability learning. Yang et al. [20] designed an optimal transport-embedded joint distribution similarity measure to assess the transferability of fault diagnosis across machines. Besides these studies that are mostly on classification problems, the transferability analysis of regression problems is also receiving attention. Mansour et al. [21] proved theoretically a series of adaptation bounds for support vector machines and ridge regression based on the empirical discrepancy. Nguyen et al. [22] proposed two MSE-based transferability estimators to evaluate the transferability between regression tasks. These works mostly focus on the discrepancy between two regression tasks and their generalization bounds for the expected loss. Nevertheless, current studies generally neglect consideration of regression characteristics, e.g., temporal information, and cannot directly apply to the degradation modeling on rotating machines which focuses more on the monotonicity and tendency characteristics. Moreover, transferability, representing the volume of domain knowledge to be transferred, is supposed to be dynamic with different feature representations or models. However, current methods conduct transferability analysis only at a relatively static level, lacking in-depth combination with a specific model.

To the best of our knowledge, there have been very few works in RUL prediction that have adopted a similar concept of transferability. In terms of feature selection, Cao et al. [23] utilized dynamic time wrapping (DTW) and Wasserstein distance to select transferable temporal features and improve RUL prediction accuracy. He et al. [24] developed an online RUL prediction method that is transferable to sequentially-collected data blocks by minimizing the marginal and conditional probability distribution. However, the term "transferable" is mainly used to describe the concept drift phenomenon in online prediction, rather than evaluating the degradation of knowledge to be transferred. Furthermore, these works did not consider prior information on fault mode.

Another similar concept to transferability is interpretability which focuses on how the model approaches the data and how it functions [8]. Interpretability can be analyzed on neurons, network structures, and samples by means of attention mechanisms, saliency evaluation, rule/concept analysis, etc., which can also be utilized in transferability analysis. In general, interpretability pays more attention on which elements affect learning process (including transfer learning) and why the prediction results can be obtained. With an easily-overlooked difference, transferability dedicates to estimating what volume of domain knowledge can be learned from a source task and how it works for a target task. The work in this paper is just exploring how to analyze and utilize transferability in the process of transfer learning, especially with a large data distribution discrepancy.

Tensor decomposition [25] is a type of higher-order principal component analysis (PCA) that decomposes a higher-order tensor into a core tensor and a series of factor matrices. Currently, tensor decomposition is mainly applied to classification problems. For instance, Hu et al. [26] employed tensor representation to align subspaces and developed a CNN model for cross-domain fault diagnosis. However, these studies use tensor decomposition only as a data preprocessing technique, and the factor matrices are usually initialized randomly. To our best knowledge, there has been no prior application of tensor representation to RUL prediction.

To summarize, the current studies on RUL prediction suffer from several limitations. Firstly, most methods are unable to handle the significant distribution discrepancy across different machines, and they also lack fault mode information in the target domain. Secondly, the current transferability analysis methods mainly focus on feature similarity, distribution alignment, and parameter significance, but they do not sufficiently represent domain knowledge to be transferred for a specific transfer learning task. Finally, transferability is evaluated independently without joint analysis with the prediction task, which can lead to biased knowledge transfer and reduced reliability of transfer

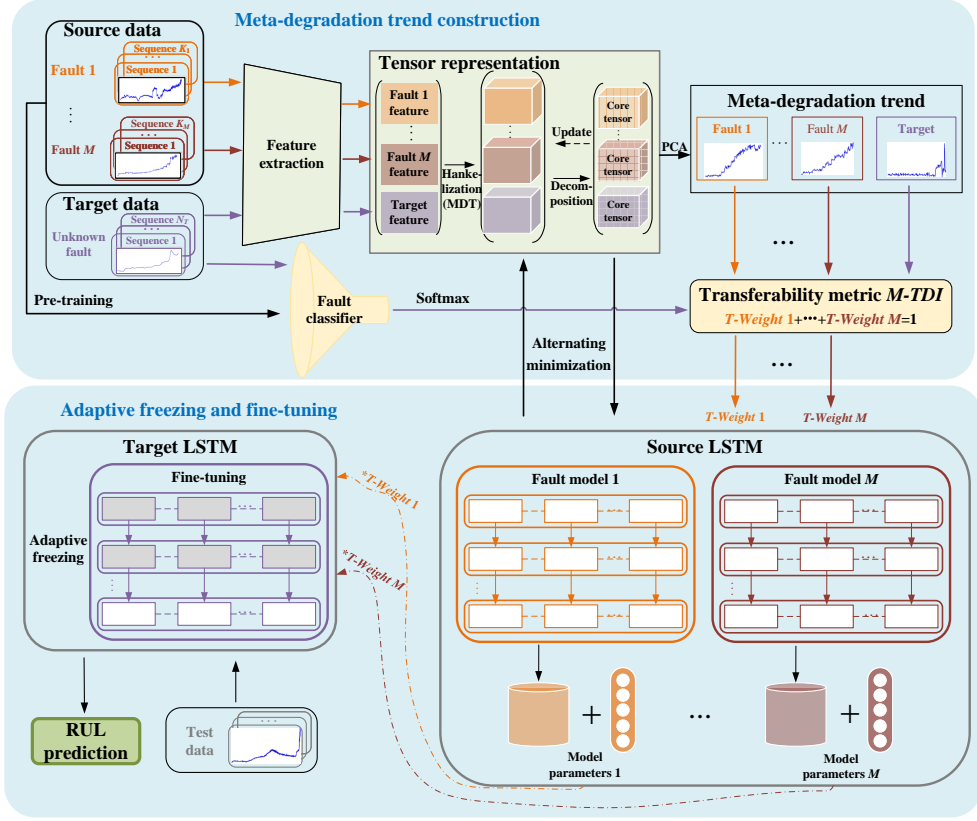


Figure 2: Flowchart of the proposed approach.

learning. This paper tries to address these limitations from theoretical and practical aspects. From the theoretical aspect, this paper utilizes tensor optimization to seek essential degradation information and effective transfer channels, which can set up a new research paradigm of transferability analysis, i.e., transforming the static transferability metric to a dynamic one and then optimizing it with a prediction task. From the practical aspect, this paper designs a selective transfer strategy, which helps to evaluate RUL values with large data distribution discrepancy even if the fault mode information in the target task is unavailable.

### 3. Proposed approach

This section presents a new RUL transfer prediction approach across different machines, as shown in Figure 2. This approach is composed of three parts: (1) Tensor representation-based meta-degradation trend extraction; (2) Construction of transferability metric based on fault mode and degradation characteristic; (3) Selective transfer learning network with an alternating optimization-based training algorithm. The detailed implementation is introduced as follows.

#### 3.1. Problem description

Assume that the source domain  $D^S$  contains  $N_S$  machines with the degradation data  $\{X_i^S, Y_i^{S_{Class}}, Y_i^{S_{RUL}}\}_{i=1}^{N_S}$ , where the superscript  $S$  indicates the source domain,  $Y_i^{S_{Class}}$  and  $Y_i^{S_{RUL}}$  are the fault mode labels and RUL values of the  $i$ -th source machine, respectively. The target domain  $D^T$  has  $N_T$  machines for training with the degradation data  $\{X_i^T, Y_i^{T_{RUL}}\}_{i=1}^{N_T}$ , where the superscript  $T$  indicates the target domain,  $Y_i^{T_{RUL}}$  are the RUL values of the  $i$ -th target machine. There are test data  $X^{T_{test}}$  in the target domain.

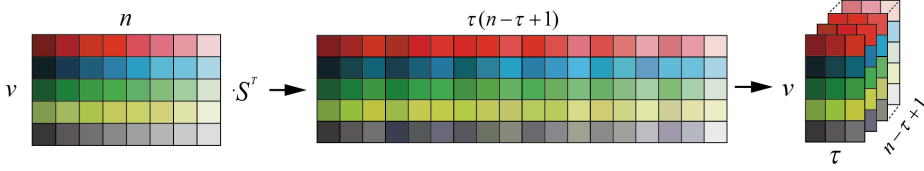


Figure 3: Illustration of tensorization along sample direction using MDT.

$D^S$  consists of the sample space  $\chi^S = \{\chi^{S_1}, \dots, \chi^{S_M}\}$  of total  $M$  fault modes under the source machine and its marginal probability distribution  $P(\chi^S) = \{P(\chi^{S_1}), \dots, P(\chi^{S_M})\}$ , i.e.,  $D^S = \{\chi^{S_m}, P(\chi^{S_m})\}_{m=1}^M$ .  $D^T$  consists of the sample space  $\chi^T$  under the target machine and its marginal probability distribution  $P(\chi^T)$ , i.e.,  $D^T = \{\chi^T, P(\chi^T)\}$ . The machine types in the source domain  $D^S$  and target domain  $D^T$  are the same, but the machines' models and sizes are different. The data collected from different machines have different distribution characteristics, i.e.,  $P(\chi^S) \neq P(\chi^T)$ .

Since the target domain data is on a small scale, it is not easy to directly build the mapping relationship from  $\chi^T$  to the RUL label space  $\gamma^T$ , i.e.,  $f_{Target} : \chi^T \mapsto \gamma^T$ . The RUL transfer prediction problem to be solved aims to improve the prediction performance of  $f_{Target}$  by using the nonlinear mapping  $f_{Source} : \chi^S \mapsto \gamma^S$  from  $\chi^S$  to the RUL label space  $\gamma^S$  in the source domain.

### 3.2. Tensor representation-based meta-degradation trend

To utilize the fault mode information, an effective way is to extract the common degradation trend of each fault mode, i.e., the meta-degradation trend. According to the degradation mechanism of typical rotating machines [27], the same fault mode is believed to contain identical prognostic knowledge for the machines with the same manufacturing specifications and under identical working conditions. Tensor decomposition is introduced in this section to extract the meta-degradation trend of each fault mode in the source domain.

Since the target domain data has no fault label, the feature extraction should run in an unsupervised learning mode. In this paper, a deep autoencoder (DAE) is used to extract features. Certainly, the other unsupervised learning methods can also work. Let the encoder and decoder of DAE consist of  $L$  layers, respectively, and the network parameters of the encoder and decoder are expressed as  $\{\theta_{en}, \theta_{de}\}$ . Minimizing the loss function  $\ell_{AE} = \|X - \tilde{X}\|_F^2$  can make the reconstructed data  $X = \left\{ \left\{ X_i^S \right\}_{i=1}^{N_S}, \left\{ X_i^T \right\}_{i=1}^{N_T} \right\} \subseteq \mathbb{R}^{n \times d}$ . Then a deep feature set can be obtained as

$$\widehat{X} = f_{en}(X; \theta_{en}) = \left\{ \left\{ \widehat{X}_i^S \right\}_{i=1}^{N_S}, \left\{ \widehat{X}_i^T \right\}_{i=1}^{N_T} \right\} \subseteq \mathbb{R}^{n \times v}.$$

To represent the temporal information in the degradation sequence, the multi-way delay embedding transform(MDT) [28] is used to transform  $\widehat{X}$  into a high-order block Hankel tensor  $\mathbb{X} = \left\{ \left\{ \mathcal{X}_i^S \right\}_{i=1}^{N_S}, \left\{ \mathcal{X}_i^T \right\}_{i=1}^{N_T} \right\} \subseteq \mathbb{R}^{(n-\tau+1) \times v \times \tau}$  with smooth characteristics along the time dimension, as shown in Eq. (1). The MDT operation can also be illustrated in Figure 3 for a better understanding.

$$\mathbb{X} = \mathcal{H}_\tau(\widehat{X}) = \text{Fold}\left(\widehat{X} \times_1 S_1 \times \dots \times_H S_H\right) \quad (1)$$

where  $S$  is the duplication matrix,  $H$  is the order of  $\widehat{X}$ ,  $\tau$  is the embedded dimension. The core tensor  $G = \left\{ \left\{ G_i^S \right\}_{i=1}^{N_S}, \left\{ G_i^T \right\}_{i=1}^{N_T} \right\} \subseteq \mathbb{R}^{(n-\tau+1) \times v \times \tau}$  can be obtained by Tucker decomposition [28]:

$$G = \mathbb{X} \times_1 U^{(1)\top} \times_2 U^{(2)\top} \times \dots \times_{C-1} U^{(C-1)\top} \quad (2)$$

s.t.  $(U^{(c)})^\top U^{(c)} = I, c = 1, \dots, C-1$

where  $C$  is the order of  $\mathbb{X}$ ,  $\zeta$  is the feature dimension of  $G$ ,  $\{U^{(c)}\}_{c=1}^{C-1}$  are the projection matrices that usually have orthonormal columns,  $C - 1$  indicates no tensor expansion at the last dimension, i.e., the time dimension.

To calculate the meta-degradation trend,  $G$  needs to be transformed back to the original sample space. Here the inverse MDT is adopted to get  $\widehat{G} = \left\{ \left\{ \widehat{G}_i^S \right\}_{i=1}^{N_S}, \left\{ \widehat{G}_i^T \right\}_{i=1}^{N_T} \right\} \subseteq \mathbb{R}^{n \times \zeta}$ :

$$\widehat{G} = \mathcal{H}_\tau^{-1}(G) = \text{Unfold}(G) \times_1 S_1^\dagger \cdots \times_H S_H^\dagger \quad (3)$$

where  $\dagger$  is the Moore-Penrose pseudo-inverse. With the obtained  $\widehat{G}$ , the meta-degradation trend  $Meta = \{Meta^{S,m}\}_{m=1}^M$  for the total  $M$  fault modes and the target domain can be obtained, where  $Meta^{S,m}$  is the meta-degradation trend of the  $m$ -th fault mode in the source domain and can be solved by using PCA:

$$Meta^{S,m} = \left\{ PCA \left( \left[ PCA \left( \widehat{G}_i^S \right) \right]_{i=1, \dots, K_m} \right) \right\} \quad (4)$$

where  $K_1 + \cdots + K_M = N_S$ ,  $[\cdot]$  represents the sequence concatenation, and  $PCA(\cdot)$  calculate the sequence of the first principal component. Eq. (4) determines the first principal component of each degradation sequence from the source domain, and then re-calculates the first principal component of all the obtained sequences that all belong to the  $m$ -th fault mode, i.e., the meta-degradation trend of the  $m$ -th fault mode. The calculation process for  $Meta^T$  is identical to Eq (4).

It is worth noting that  $\{U^{(c)}\}_{c=1}^{C-1}$  in Eq. (2) is randomly initialized. To seek the optimal  $G$  that can retain the essential information from  $\mathbb{X}$  to the greatest extent,  $\{U^{(c)}\}_{c=1}^{C-1}$  should be optimized by minimizing the loss

$$\ell_{Tucker} = \frac{1}{2} \sum_{i=1}^{N_S+N_T} \left\| \mathcal{X}_i^{S,T} - \widehat{\mathcal{X}}_i^{S,T} \right\|_F^2, \text{ where } \widehat{\mathbb{X}} = \left\{ \left\{ \widehat{\mathcal{X}}_i^S \right\}_{i=1}^{N_S}, \left\{ \widehat{\mathcal{X}}_i^T \right\}_{i=1}^{N_T} \right\} \subseteq \mathbb{R}^{(n-\tau+1) \times \nu \times \tau} \text{ is calculated by:}$$

$$\begin{aligned} \widehat{\mathbb{X}} &\approx G \times_1 U^{(1)} \times_2 U^{(2)} \cdots \times_{C-1} U^{(C-1)} \\ &= \llbracket G; U^{(1)}, U^{(2)}, \dots, U^{(C-1)} \rrbracket \end{aligned} \quad (5)$$

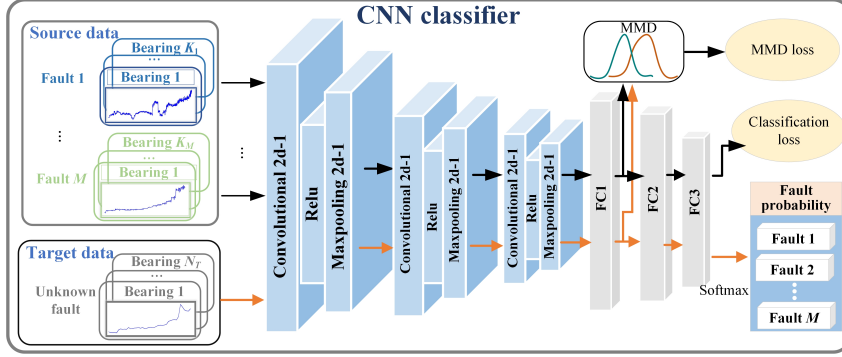
The optimization process will be elaborated in Section 3.4. Compared to direct feature extraction from original data, the meta-degradation trend with tensor representation can reduce the disturbance by noise and degradation randomness and express more accurately the degradation mechanism information under different fault modes.

### 3.3. Construction of transferability metric

Based on the meta-degradation trend of each fault mode, transferability is further studied. In an ideal scenario, the meta-degradation trend of all fault modes can be directly weighed to evaluate the transferability. But if the fault mode in the target domain is unknown, the weight of each fault mode in the source domain is unresolvable. To integrate the fault mode information, it is necessary to determine the fault probability of the target domain data. Moreover, as indicated by Figure 1, some characteristics of the data, such as geometric shape and tendency, should also be considered.

Specifically, a CNN classifier, as a popular technique in fault diagnosis, is first built to identify the fault probability of the target domain data, as shown in Figure 4. With the concern about data distribution discrepancy, the MMD distance is added after the first fully-connected layer of the CNN classifier to realize domain adaptation. Certainly, the CNN classifier, which only provides a rough fault probability for the target domain data, is not the most important issue in the proposed methodology. Degradation characteristic information is going to be further incorporated to guarantee the effect of knowledge transfer. The detailed structure of the CNN model will not be discussed. Here the CNN classifier can be built directly using raw signals instead of core tensors. The role of tensor representation is to extract the meta-degradation trend and calculate the transferability metric. The classification model training will be computationally expensive once applying core tensors to train the fault classifier. Please note that the CNN model can be replaced by any other classification model with probability output.

The degradation characteristic, in terms of geometric shape and tendency, is further considered. Then a transferability metric, named cross-machine transferrable degree indicator ( $M-TDI$ ), is constructed to indicate the similarity



**Figure 4:** Sketch map of CNN-based fault classifier for calculating the fault probability of target domain data. One can modify the network structure following specific requirements. The degradation characteristic, in terms of geometric shape and tendency, is further considered.

degree of prognostic knowledge between the source domain data and the target domain data, as follows:

$$M - TDI(m) = \frac{\frac{1}{N_T} \sum_{i=1}^{N_T} \text{Softmax}(X_i^T)}{\text{DTW}(Meta^T, Meta^{S,m}) \times \frac{|\text{MIC}(Meta^T) - \text{MIC}(Meta^{S,m})|}{\text{MIC}(Meta^T) * \text{MIC}(Meta^{S,m})}} \quad (6)$$

where  $X_i^T$  is the  $i$ -th degradation sequence in the target domain,  $\text{Softmax}(X_i^T)$  is the fault probability of  $X_i^T$  which comes from the Softmax layer,  $\text{DTW}(\cdot, \cdot)$  indicates the dynamic time warping (DTW) distance between the two sequences with unequal length,  $\text{MIC}(\cdot)$  indicates the maximal information coefficient (MIC) value of a sequence that is used to represent the correlation between the sequence and its own degradation time. The detailed calculations of DTW and MIC can be found in [23] and [29], respectively.

In Eq. (6), a larger DTW value indicates that the two meta-degradation trends are more geometrically divergent, i.e., with smaller geometric similarity. The MIC value can measure the variation of tendency information in a meta-degradation trend. The larger the MIC value is, the better the sequence's tendency will be.  $\frac{|\text{MIC}(Meta^T) - \text{MIC}(Meta^{S,m})|}{\text{MIC}(Meta^T) * \text{MIC}(Meta^{S,m})}$  measures the difference between the two meta-degradation trends' MIC value and the MIC value of each meta-degradation trend itself. Specifically, if the MIC value of one meta-degradation trend is smaller, which indicates the degradation trend is not significant, the similarity of two meta-degradation trends should decrease, i.e., the tendency similarity will be smaller. Obviously, the denominator of  $M-TDI$  is able to measure the geometric similarity and tendency similarity between two meta-degradation trends (called degradation characteristic in this paper). Therefore,  $M-TDI$  comprehensively considers the fault probability and degradation characteristic and can accurately evaluate the transferrable degree of the source domain data.

To apply  $M-TDI$  practically, a transfer weight is further built for the  $m$ -th fault mode in the source domain:  $T - Weight_m = \frac{M - TDI(m)}{\sum_{m=1}^M M - TDI(m)}$ . A larger value of  $T - Weight_m$  indicates a greater significance of the prognostic knowledge of the  $m$ -th fault mode's data to the target domain.

### 3.4. Selective transfer learning approach

#### 3.4.1. Transfer strategy based on M-TDI

This section presents a new  $M-TDI$ -based transfer strategy to transfer the prognostic knowledge selectively. Since the target domain has a small amount of training data, freezing and fine-tuning are adopted as the baseline transfer strategy. To extract temporal information from degradation sequences, a three-layer stacked LSTM network is utilized as the prototype prediction model. In the past decade, LSTM has been widely used as the backbone network to extract temporal features for RUL prediction [4; 10; 24]. LSTM has several merits in capturing long dependency and alleviating gradient vanishing or exploding. Obeying the designed transfer strategy, the other temporal neural networks can also be applied. Please note that the LSTM network runs on the degradation sequence in the core tensor instead of raw



signals. The network parameters  $\theta_{LSTM}$  can be optimized by minimizing the empirical loss  $\ell_{LSTM} = \left\| Y - \widehat{Y} \right\|_F^2$ , where  $\widehat{Y} = f_{LSTM}(G; \theta_{LSTM})$  is the predicted RUL value. The transfer strategy contains two parts:

1) Weighted initialization: One LSTM network is separately trained for each fault mode in the source domain. Then there is a total of  $M$  LSTM networks. The LSTM network of the target domain is set with an identical structure to the network of the source domain, and its parameter  $\theta_{LSTM}^T$  is initialized by:

$$\theta_{LSTM}^T = \sum_{m=1}^M \left( \theta_{LSTM}^{S,m} \times T - Weight_m \right) \quad (7)$$

where  $\{T - Weight_m\}_{m=1, \dots, M}$  and  $\{\theta_{LSTM}^{S,m}\}_{m=1, \dots, M}$  are the transfer weights and LSTM parameters of the total  $M$  fault modes in the source domain. Here the value of  $M$  is usually not too large since the proposed approach mainly involves the major fault modes that cause machines to fail. Also, the prognostic knowledge to transfer has no need to be fine-grained. Otherwise, the knowledge might be less representative. If too many fault modes are considered, one can first adaptively allocate all degradation sequences to  $M$  clusters in order to avoid extra training burden.

2) Adaptive freezing: After initialization, the first few layers of the target LSTM network should be frozen to keep the source domain's knowledge. Then the remaining parameters are fine-tuned using the target domain data. The number of frozen layers determines how much the prognostic knowledge from the source domain is retained. It is critical to determine the frozen layers utilizing the transferability analysis.

An adaptive freezing strategy is designed to determine the frozen layers by estimating the contribution discrepancy between the two fault modes that are respectively with the maximum value and the minimum value of  $T-Weight$ . Here a simple threshold  $1/M$  is introduced for the determination. For instance, if the value of  $span = \max_{m=1, \dots, M} \{T - Weight_m\} - \min_{m=1, \dots, M} \{T - Weight_m\}$  is greater than  $1/M$ , the largest contribution discrepancy would go beyond the average level. In this case, it means that there is a fault mode whose data have better transferability than the others. More layers are required to be frozen to transfer the knowledge of this fault mode. On the contrary, if the value of  $span$  is less than the threshold  $1/M$ , all of the fault modes in the source domain have a close contribution to the target domain task, which indicates the target domain data do not particularly belong to any fault mode. Fewer layers need to be frozen in order to learn more prognostic knowledge from the target domain itself. Following this idea, the number of frozen layers can be determined by:

$$Frozen\_layer = \begin{cases} \left\{ 0, 1, \dots, \left\lfloor \frac{N}{2} \right\rfloor \right\}, & \text{if } span < \frac{1}{M} \\ \left\{ \left\lfloor \frac{N}{2} \right\rfloor + 1, \dots, N \right\}, & \text{if } span \geq \frac{1}{M} \end{cases} \quad (8)$$

where  $\lfloor \cdot \rfloor$  represents the down-rounding operation, and  $N$  is the total layer number. Certainly, the threshold  $1/M$  is just roughly defined and can be further optimized according to task requirements, e.g., the amount of target domain data, and the degradation similarity between two domains. Moreover, Eq. (8) merely provides a proper set of the number of frozen layers. One can run cross-validation or add a validation set to determine the most suitable layers to freeze.

Therefore, the overall loss of the selective transfer learning approach can be expressed as:

$$L = \beta_1 \sum_{1, \dots, N_S, 1, \dots, N_T} \ell_{AE} + \beta_2 \sum_{1, \dots, M} \ell_{LSTM}^{S,m} + \beta_3 \ell_{LSTM}^T + \sum_{1, \dots, N_S, 1, \dots, N_T} \ell_{Tucker} \quad (9)$$

where  $\beta_1$ ,  $\beta_2$  and  $\beta_3$  are the regularization parameters. From Eq. (9), the solution covers optimizing the prediction model and feature extraction. Minimizing Eq. (9) can not only determine the optimal core tensor representation (i.e., the meta-degradation trend) but also obtain the optimal effect of prognostic knowledge transfer.

### 3.4.2. Training algorithm

The optimization variables in Eq. (9) include not only the DAE parameters  $\theta_{de}$ ,  $\theta_{en}$ , LSTM parameters  $\{\theta_{LSTM}^{S,m}\}_{m=1 \dots M}$  and  $\theta_{LSTM}^T$ , but also the tensor factor matrix  $\{U^{(c)}\}_{c=1}^{C-1}$ . The optimal  $\theta_{de}$ ,  $\theta_{en}$ ,  $\{\theta_{LSTM}^{S,m}\}_{m=1 \dots M}$  and  $\theta_{LSTM}^T$  can be found by using a stochastic gradient descent (SGD) method, while the solution of  $\{U^{(c)}\}_{c=1}^{C-1}$  cannot be solved in the same way. An alternately minimizing scheme is adopted to train the network: Fix  $\{U^{(c)}\}_{c=1}^{C-1}$ , then

update  $\theta_{de}, \theta_{en}, \left\{ \theta_{LSTM}^{S,m} \right\}_{m=1 \dots M}$  and  $\theta_{LSTM}^T$ ; Then fix  $\theta_{de}, \theta_{en}, \left\{ \theta_{LSTM}^{S,m} \right\}_{m=1 \dots M}$  and  $\theta_{LSTM}^T$  update  $\left\{ U^{(c)} \right\}_{c=1}^{C-1}$ . These two steps run alternately until reaching convergence. The optimization process is as follows.

1) Update  $\theta_{de}, \theta_{en}, \left\{ \theta_{LSTM}^{S,m} \right\}_{m=1 \dots M}$  and  $\theta_{LSTM}^T$ : The DAE feature extractor and the LSTM network can be jointly optimized. For simplicity, the solution is expressed as solving Eq. (10):

$$\begin{aligned}
\min J &= \beta_1 \ell_{AE} + \beta_2 \sum_{m=1}^M \ell_{LSTM}^{S,m} + \beta_3 \ell_{LSTM}^T \\
&= \sum_{i=1}^{N_S, N_T} \left\| X_i^{S,T} - f_{AE} \left( X_i^{S,T}; \theta_{en}, \theta_{de} \right) \right\|_F^2 \\
&\quad + \sum_{m=1}^M \sum_i^{K_m} \left\| Y_i^{S, RUL} - f_{LSTM} \left( G_i^S; \theta_{LSTM}^{S,m} \right) \right\|_F^2 \\
&\quad + \sum_{i=1}^{N_T} \left\| Y_i^{T, RUL} - f_{LSTM} \left( G_i^T; \theta_{LSTM}^T \right) \right\|_F^2
\end{aligned} \tag{10}$$

By combining Eq. (1) and (2), the core tensor can be obtained by:

$$\begin{aligned}
G &= \text{Fold} \left( \widehat{X} \times_1 S_1 \times \dots \times_H S_H \right) \times_1 U^{(1)T} \times_2 U^{(2)T} \times \dots \times_{C-1} U^{(C-1)T} \\
&= \text{Fold} \left( f_{en} \left( X; \theta_{en} \right) \times_1 S_1 \times \dots \times_H S_H \right) \times_1 U^{(1)T} \times_2 U^{(2)T} \times \dots \times_{C-1} U^{(C-1)T}
\end{aligned} \tag{11}$$

Since  $\left\{ U^{(c)} \right\}_{c=1}^{C-1}$  can be regarded as a constant variable,  $\theta_{de}, \theta_{en}, \left\{ \theta_{LSTM}^{S,m} \right\}_{m=1 \dots M}$  and  $\theta_{LSTM}^T$  can be updated by:

$$\theta_{de} \leftarrow \theta_{de} - \alpha \frac{\partial J}{\partial f_{AE}} \frac{\partial f_{AE}}{\partial \theta_{de}} \tag{12}$$

$$\theta_{en} \leftarrow \theta_{en} - \alpha \left( \frac{\partial J}{\partial f_{AE}} \frac{\partial f_{AE}}{\partial \theta_{en}} + \frac{\partial J}{\partial f_{LSTM}} \frac{\partial f_{LSTM}}{\partial f_{en}} \frac{\partial f_{en}}{\partial \theta_{en}} \right) \tag{13}$$

$$\theta_{LSTM}^{S,m} \leftarrow \theta_{LSTM}^{S,m} - \alpha \frac{\partial J}{\partial f_{LSTM}} \frac{\partial f_{LSTM}}{\partial \theta_{LSTM}^{S,m}} \tag{14}$$

$$\theta_{LSTM}^T \leftarrow \theta_{LSTM}^T - \alpha \frac{\partial J}{\partial f_{LSTM}} \frac{\partial f_{LSTM}}{\partial \theta_{LSTM}^T} \tag{15}$$

where  $\alpha$  is the learning rate. It should be noted that  $\theta_{LSTM}^T$  is not all updated because the first few layers are frozen according to Eq. (8).

2) Update  $\left\{ U^{(c)} \right\}_{c=1}^{C-1}$ : The optimal  $\left\{ U^{(c)} \right\}_{c=1}^{C-1}$  can be found by minimizing  $\ell_{Tucker} = \frac{1}{2} \sum_{i=1}^{N_S+N_T} \left\| \mathcal{X}_i^{S,T} - \widehat{\mathcal{X}}_i^{S,T} \right\|_F^2$ .

This problem can be reformulated by unfolding each tensor variable as:

$$\ell_{Tucker} = \sum_{i=1}^{N_S+N_T} \left\| \mathcal{X}_i^{S,T} - \left[ \left[ G_i^{S,T}; U^{(1)}, U^{(2)}, \dots, U^{(C-1)} \right] \right] \right\|_F^2 \tag{16}$$

Then minimizing  $\ell_{Tuc\ ker}$  equals the following problem:

$$\begin{aligned}
& \min_{\{U^{(c)}\}} \left\| \mathcal{X}_i^{S,T} - \left[ \left[ G_i^{S,T}; U^{(1)}, U^{(2)}, \dots, U^{(C-1)} \right] \right] \right\|_F^2 \\
&= \left\| \mathcal{X}_i^{S,T} \right\|_F^2 - 2 \left\langle \mathcal{X}_i^{S,T}, \left[ \left[ G_i^{S,T}; U^{(1)}, U^{(2)}, \dots, U^{(C-1)} \right] \right] \right\rangle \\
&\quad + \left\| \left[ \left[ G_i^{S,T}; U^{(1)}, U^{(2)}, \dots, U^{(C-1)} \right] \right] \right\|_F^2 \\
&= \left\| \mathcal{X}_i^{S,T} \right\|_F^2 - \left\| \mathcal{X}_i^{S,T} \times_1 U^{(1)} \times_2 U^{(2)} \dots \times_{C-1} U^{(C-1)} \right\|_F^2
\end{aligned} \tag{17}$$

Since  $U^{(c)}$  is orthogonal, Eq. (16) can be re-written as the following problem:

$$\begin{aligned}
& \max_{U^{(c)}} \left\| \mathcal{X}_i^{S,T^{(c)}} \times_1 U^{(1)} \times_2 U^{(2)} \dots \times_{C-1} U^{(C-1)} \right\| \\
&= \max_{U^{(c)}} \left\| U^{(c)} W \right\|
\end{aligned} \tag{18}$$

where  $W = \mathcal{X}_i^{S,T^{(c)}} \left( U^{(C-1)T} \otimes \dots \otimes U^{(c+1)T} \otimes U^{(c-1)T} \otimes \dots \otimes U^{(1)T} \right)$  represents the  $c$ -th dimension expansion of  $\mathcal{X}_i^{S,T}$ . The optimal  $U^{(c)}$  can be determined using an alternating least squares algorithm [25], i.e., realize the singular value decomposition (SVD) in all directions of  $\mathcal{X}_i^{S,T}$  and update  $U^{(c)}$  iteratively.

With the obtained model parameters, the test data are fed into the target LSTM network to obtain the predicted RUL value. By clarifying the transferable degree of the source domain data, the prognostic knowledge can be selectively transferred to improve the effect of cross-machine transfer prediction.

### 3.5. Analysis of upper bound of prediction error for target model

Without loss of generality and for the ease of formula derivation, the LSTM model can be replaced by an RNN model with linear activation, as shown in Eq. (19). LSTM can be viewed as a nonlinear extension of the RNN model, so the following analysis and conclusion also apply to the proposed method.

$$f(x_t) = Vx_t + Ws_{t-1}, \quad s_{t-1} = Vx_{t-1} + Ws_{t-2} \tag{19}$$

**Definition 1 (Linear decoupling)** [30]. For a linear model, let  $X \in \mathbb{R}^{n \times d}$  be a row matrix from the input space  $D$  and denote the empirical covariance matrix  $\frac{1}{n} X^T X$  by  $\bar{\Sigma}$ . Define  $P_{\parallel}$  to be the projection matrix into the row space of  $X$ , and  $P_{\perp}$  to be the projection matrix into its orthogonal complement, i.e.:

$$P_{\parallel} \triangleq X^T (X X^T)^{-1} X, \quad P_{\perp} \triangleq I - P_{\parallel} \tag{20}$$

**Definition 2 (The optimal model)**. Set  $\epsilon > 0$  is a constant. By minimizing the loss  $L$ , s.t.  $L \leq \epsilon$ , the optimal model parameters  $\theta^*$  for the prediction task can be obtained.

From Definition 2, there exists a constant  $\epsilon_S > 0$ , then the optimal source model parameters  $\theta_S^* = \{ \theta_{S,1}^*, \dots, \theta_{S,M}^* \}$  can be get by minimizing the loss  $\ell_{LSTM}^{S,m}$  in Eq. (9) subject to  $\ell_{LSTM}^{S,m} \leq \epsilon_S$ . From Definition 1, the optimal parameters can be expressed as  $\{ V_S^*; W_S^* \} = \{ V_{S,1}^*, \dots, V_{S,M}^*; W_{S,1}^*, \dots, W_{S,M}^* \}$ , where  $M$  is the total number of fault modes. Meanwhile, by Eq. (6), the transfer weight  $T - Weight = [T - Weight_1, \dots, T - Weight_M]$  can be calculated for all  $m$  fault modes in the source domain.

**Definition 3 (Fine-tuning for the target model)**. Let  $\{ V_T, W_T \} \in \mathbb{R}^d$  be the ground-truth parameters of the target task, and  $y \in \mathbb{R}^n$  be the real label of the target data  $X_T = \{ x_t \}_{t=1}^n$ , i.e.,  $y_i = V_T x_t + W_T s_{t-1}$ . For a linear RNN model, the optimal parameters of the target task with fine-tuning can be expressed as:

$$V_T^* = P_{\perp} V_S + P_{\parallel} V_T, \quad W_T^* = Q_{\perp} W_S + Q_{\parallel} W_T \tag{21}$$

Denote by  $\mathcal{V}$  and  $\mathcal{W}$  the target model parameters. These parameters can be initialized as  $\mathcal{V}^0 = T - Weight \times V_S^*$  and  $\mathcal{W}^0 = T - Weight \times W_S^*$ . Then there is the following definition and proposition.

**Definition 4** (Population loss). With the target model parameters  $\mathcal{V}$  and  $\mathcal{W}$  the population loss can be defined as:

$$R(\mathcal{V}, \mathcal{W}) \triangleq E_{x \sim D} \left[ \left( (x_t^T V_T + s_{t-1}^T W_T) - (x_t^T \mathcal{V} + s_{t-1}^T \mathcal{W}) \right)^2 \right] \quad (22)$$

**Proposition** (Upper bound of prediction error for target model). For  $1 \leq m \leq d$ ,  $\lambda_m > 0$  and all  $\delta \geq 1$  with the probability at least  $1 - e^{-\delta}$ . Assume the number of frozen layer is  $k$ , the population loss  $R(\mathcal{V}, \mathcal{W})$  of the proposed method has the upper bound:

$$\begin{aligned} R(\mathcal{V}, \mathcal{W}) \leq & 4 \sum_t \left\| x_t^T (V_T - \mathcal{V}^0) \right\|_{l \leq k}^2 \\ & + 8g(\lambda, \delta, n)^3 \frac{\left\| P_{\leq m} (V_T - \mathcal{V}^0) \right\|_{l > k}^2}{\lambda_m^2} \\ & + 8g(\lambda, \delta, n) \left\| P_{> m} (V_T - \mathcal{V}^0) \right\|_{l > k}^2 \end{aligned} \quad (23)$$

Please refer to Section Appendix for more derivation details.

Eq. (23) indicates that the upper bound of prediction error is composed of two crucial parts. The first is the function  $g(\lambda, \delta, n)$  that captures the extent of the covariance  $\sum$ , which indicates the upper bound depends on the number of training samples. The second is the bias between  $\mathcal{V}^0$  and  $V_T$ . A smaller bias will raise a lower bound, and vice versa. Optimizing the source tasks and target tasks together in Eq. (9) can definitely reduce the bias. Moreover, since the transfer weights obtained from Eq. (6) can evaluate the transferability between every source task and the target task, the weighted initialization for  $\mathcal{V}^0$  can also reduce the bias. Since the last two terms on the right side of Eq. (23) are updated via the fine-tuning process, this upper bound can provide a reliability analysis for the adaptive freezing strategy shown in Eq. (8). The upper bound of prediction error for the target model can be reduced by means of the proposed approach.

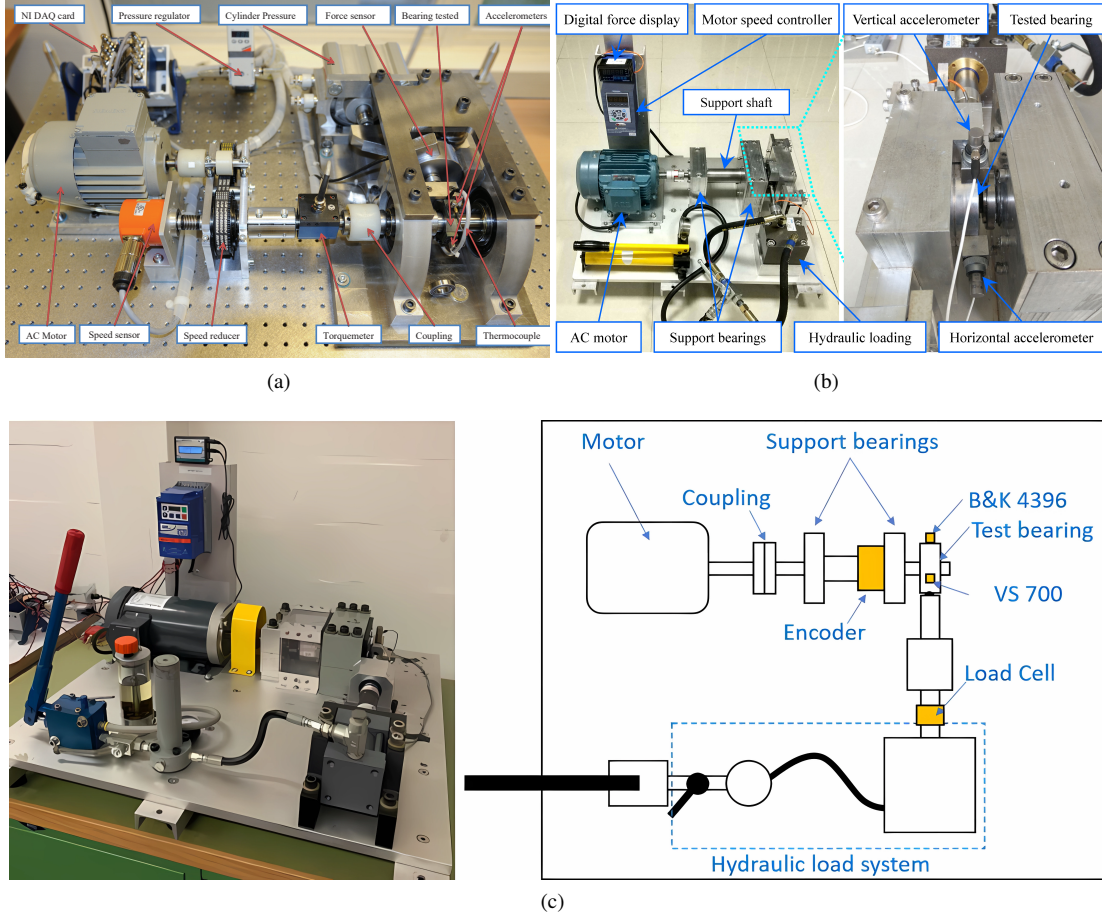
## 4. Experimental results

Rolling bearing is chosen as the validation subject to conduct experimental verification on three benchmark datasets: PHM, XJTU-SY, and UNSW. The programming environment is Matlab2014a and Python3.7, with computer configuration i7-4790 processor and 16G memory.

### 4.1. Experimental Setup

Figure 5 shows the test platforms of the three datasets for run-to-failure experiments of rolling bearings. The PHM dataset contains three working conditions with the motor speed 1800rpm, 1650rpm, and 1500rpm and the load 4000N, 4200N, and 5000N, respectively. The sampling frequency is 25.6k Hz, and the sampling interval is 10 seconds. The XJTU-SY dataset also contains three working conditions with the motor speed 2100 rpm, 2250 rpm, 2400 rpm and the load 12kN, 11kN, and 10kN, respectively. The sampling frequency is 25.6k Hz, and the sampling interval is 1 min. The UNSW dataset contains four bearings with run-to-failure experiments, named Test 1-4. The motor speed is 6000rpm. In Test 1 and Tests 3-4, a 10.5kN radial load is applied on the test bearing, and in Test 2, the load is 7kN. The sampling frequency is 51.2k Hz. The sampling interval is set to 50000 cycles in the initial stage and decreased to 20000 cycles when the spall started to grow.

The fault modes in the PHM dataset are unknown, while the fault modes in the XJTU-SY and UNSW datasets have been given. Therefore, two sets of transfer prediction tasks across machines are built by setting the XJTU-SY dataset as the source domain while the PHM and UNSW dataset as the target domain respectively, as listed in Table 1. Two bearings (B2\_1 and B2\_2) in the PHM dataset are randomly selected as the test bearing alternately to build two specific tasks. It is worth noting that only the fast degradation data are utilized for model training. The target bearings B2\_2 and B2\_6 in Task 1 have 55 and 18 samples in the fast degradation state, respectively. In Task 2, the target bearing B2\_1 has 41 samples for the model training. Although the bearings B2\_1, B2\_2, and B2\_6 might have different distributions, they still run under the same working condition, and their degradation characteristics have rather smaller divergence. Compared to the other bearings under the second working condition, the bearings B2\_1 and B2\_2 both have a much longer degradation process. So setting these two bearings as the target bearing in Task1 and Task2, respectively, is expected to raise a better visualization effect for illustration.



**Figure 5:** Testbeds used in this paper with (a) PRONOSTIA platform [31] for the PHM dataset, (b) XJTU-SY platform [32] and (c) UNSW dataset platform [33]

The location of early fault occurrence can be firstly determined by using the state assessment method in the reference [34]. Then the fast degradation data can be selected for training. Following the references [15; 34], we linearly normalize the period between the early fault location and ending point for each bearing to a ratio range from 1 to 0, serving as the RUL labels. The ending point with complete failure can be determined using different thresholds. For instance, the termination condition in the XJTU-SY dataset is set to  $10Ah$  [32], where  $Ah$  is the highest vibration amplitude in the normal state. For all data, the Hilbert-Huang transform (HHT) [35] is first run on the raw signal to calculate the marginal spectrum:  $H(w) = \int H(w, t)dt$ . The HHT feature dimension is set to 2558. The HHT features are first fed into the DAE encoder, whose structure is set [2558, 1024, 512, 128, 50]. Then tensor Tucker decomposition, shown in Eq. (2), is conducted on the 50-dimensional hidden features to get the core tensor that is set with 3-order and 25-dimensional features. The core tensor is further fed into the LSTM network, whose hidden neuron number is set to 100, and the learning rate is set to 0.001. For the test bearings listed in Table 1, we also determine the starting point of the degradation state and run HHT on the raw signals in the degradation period to obtain the test samples. The test samples with HHT features are then fed into the trained prognostic model to predict their RUL values.

## 4.2. Validation results

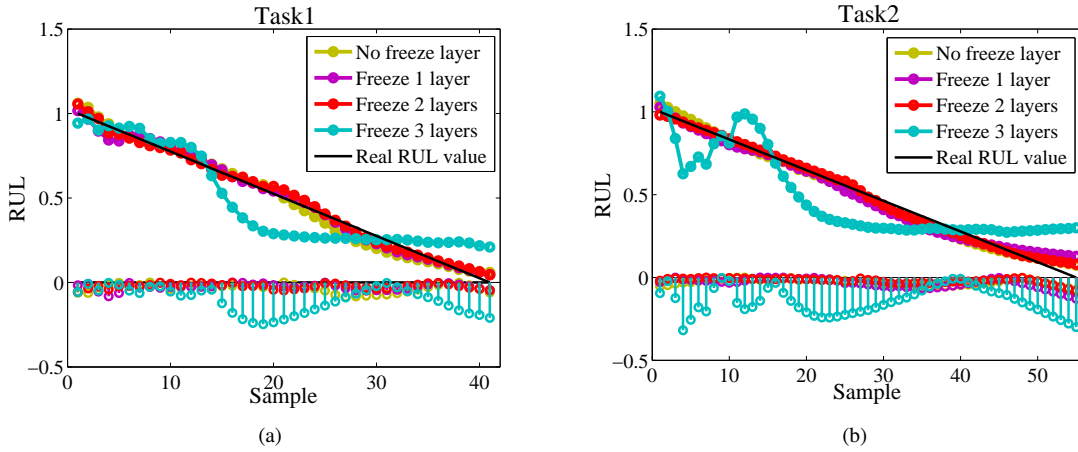
Task 1 and Task 2 are mainly used for validation and performance evaluation. Figure 6 and Figure 7 show the RUL prediction results and the box diagram of loss drop with freezing different layers. According to the  $T\text{-Weight}$  values (see Figure 8), it is suggested to freeze two or more layers for the two tasks by Eq. (8). From Figure 6, the results of freezing two layers are better than that of freezing different layers. From Figure 7, the convergence speed of freezing

**Table 1**

Task settings of RUL transfer prediction across different datasets. The first number in the bearing name indicates working condition, while the second number indicates index.

Source domain (XJTU-SY)				Target domain			
Fault mode	Bearing name			Specific task	Data	Bearing name	
Outer race	X1_1	X1_2	X1_3	Task1 (PHM)	Training	B2_2	B2_6
	X2_2	X2_4	X2_5		Test	B2_1	
	X3_1	X3_5		Task2 (PHM)	Training	B2_1	B2_6
Inner race	X2_1	X3_3	X3_4	Test		B2_2	
	Cage	X1_4	X2_3	Task3 (UNSW)	Training		Test1
Mixed	X1_5	X3_2	Test				Test2

two layers is also faster than the others. It indicates that freezing appropriate layers can improve the transfer prediction performance. The results also demonstrate the adaptive freezing strategy is effective.



**Figure 6:** RUL prediction results with freezing different layers on (a) Task1 and (b) Task2.

Eq. (6) is composed of two parts (numerator and denominator). Two new metrics (named fault probability and degradation characteristic) are designed by separately using the numerator and denominator. Figure 8 shows the values of the three metrics, and Figure 9 shows the corresponding feature distributions. When only fault mode is used, the target domain data are categorized to outer race fault (78.11% for Task1 and 79.08% for Task2). But if the degradation characteristic is used, the target domain data are more similar to the cage fault data in the source domain in terms of geometric shape and tendency. On the contrary, the proposed metric  $M-TDI$ , as reflected by  $T-Weight$ , can leverage fault mode information and degradation characteristic, then provides a comprehensive evaluation of transferability. The numerical results can be precisely reflected by Figure 9. When only using fault probability, the feature distribution of the target domain data is close to the distribution of the outer race fault data from the source domain, as visualized by the purple line leaning on the blue line (outer race fault). In Figure 9(b) and (d), the purple line turns to lean on the green line (cage fault), indicating the target domain data has better similarity with the cage fault data in degradation characteristics. The contradictory effect is the same as the results in Figure 8. Regardless of which metric is used, the prediction results are not satisfactory.  $M-TDI$  ( $T-Weight$ ) can make a tradeoff between these two metrics and facilitate the transfer of prognostic knowledge.

Besides Eq. (6), the proposed approach also includes tensor optimization. The following ablation experiments are built: 1) Without tensor optimization; 2) Without the metric of degradation characteristic; 3) Without the metric of fault probability; 4) Without  $M-TDI$ , i.e., merely fine-tune the prediction model using the target domain data. The results of the ablation experiment are shown in Figure 10. The results without tensor optimization significantly decrease, which indicates that high-quality feature representation can support knowledge transfer. However the purple

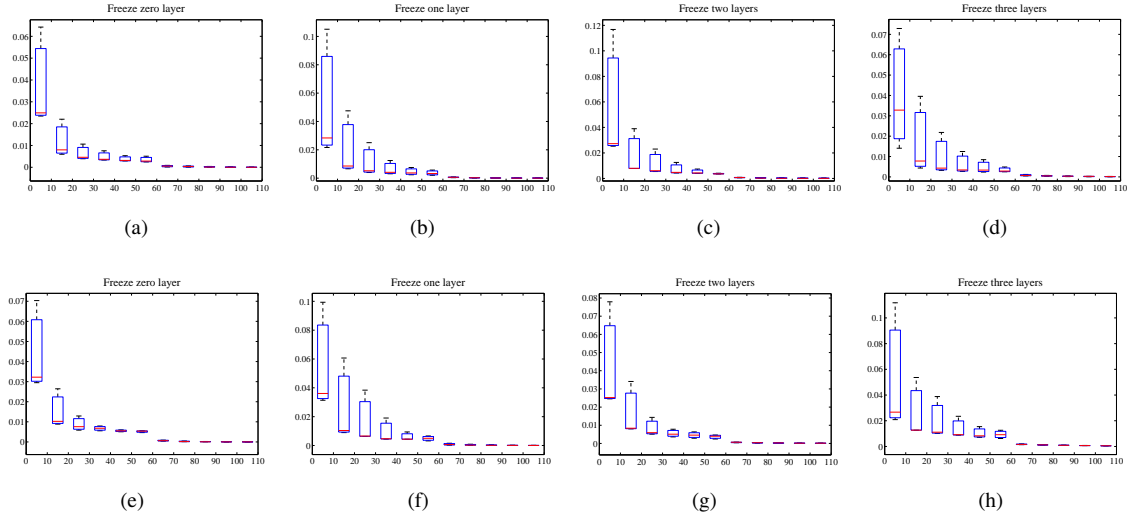


Figure 7: Box diagram of loss drops with freezing different layers, where (a)-(d) are on Task1, (e)-(h) are on Task2.

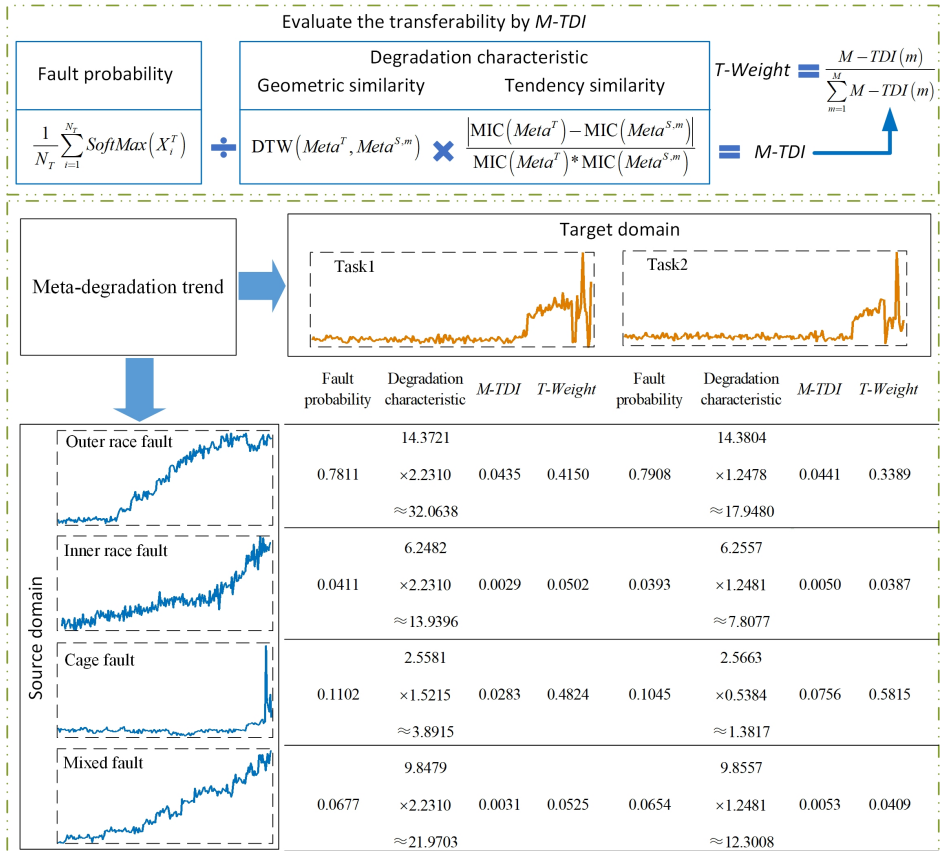


Figure 8: Detailed calculation of  $T\text{-Weight}$  on Task1 and Task2. Since the difference between the maximum value and the minimum value of  $T\text{-Weight}$  is greater than  $1/4$ , two or more layers are suggested to freeze according to Eq. (8).

line shows the worst prediction effect, indicating the necessity of an appropriate transfer strategy. Also, the transfer without degradation characteristic (dark green line) does not achieve good performance. It means that only transferring fault mode information will make the knowledge deviate. The transfer without fault probability has a similar effect. In contrast, the proposed approach can exploit the transferability from the two aspects and achieve a favorable transfer effect.

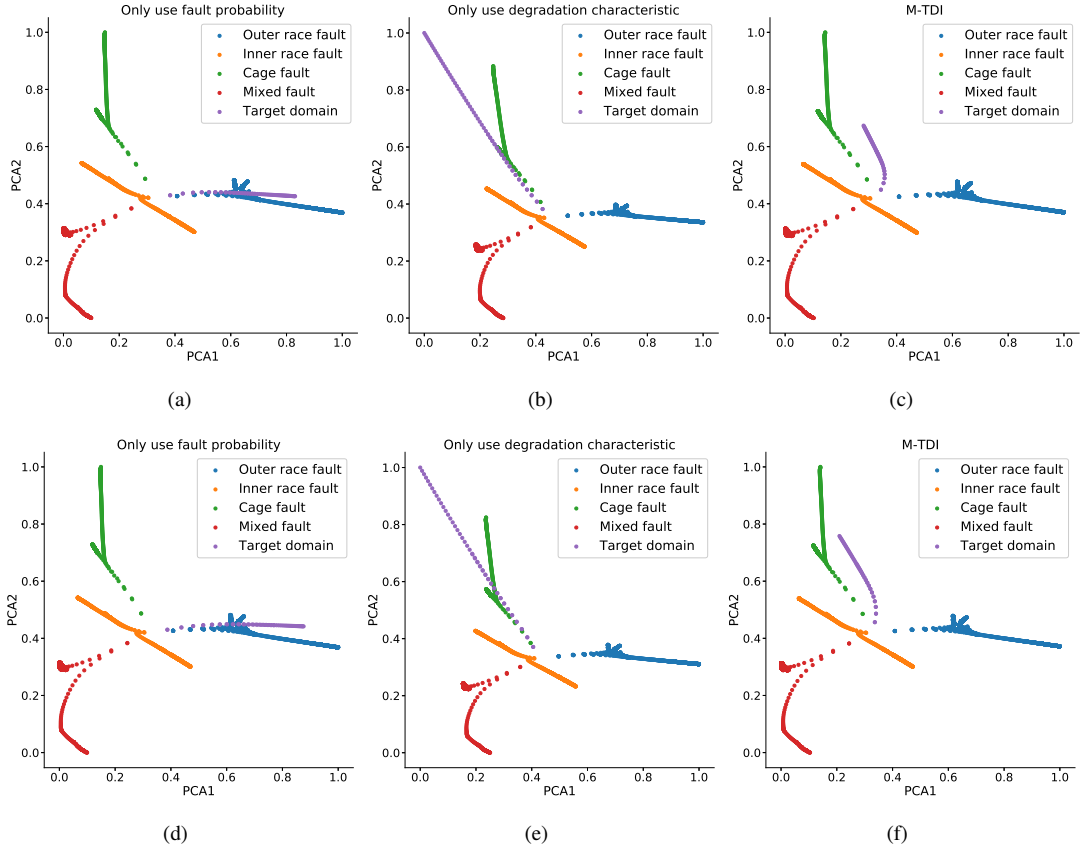
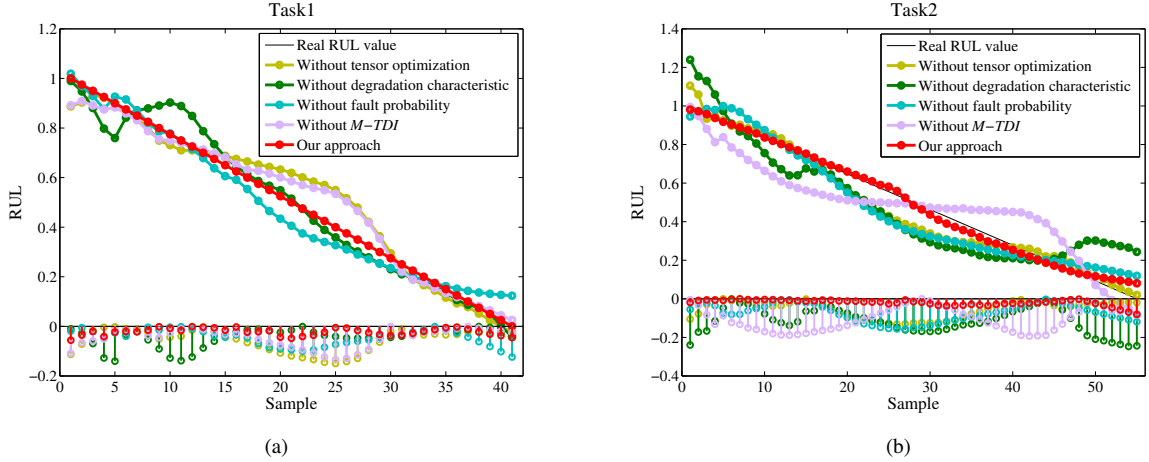


Figure 9: Feature distributions corresponding to different metrics, where (a)-(c) are on Task1, (d)-(f) are on Task2.

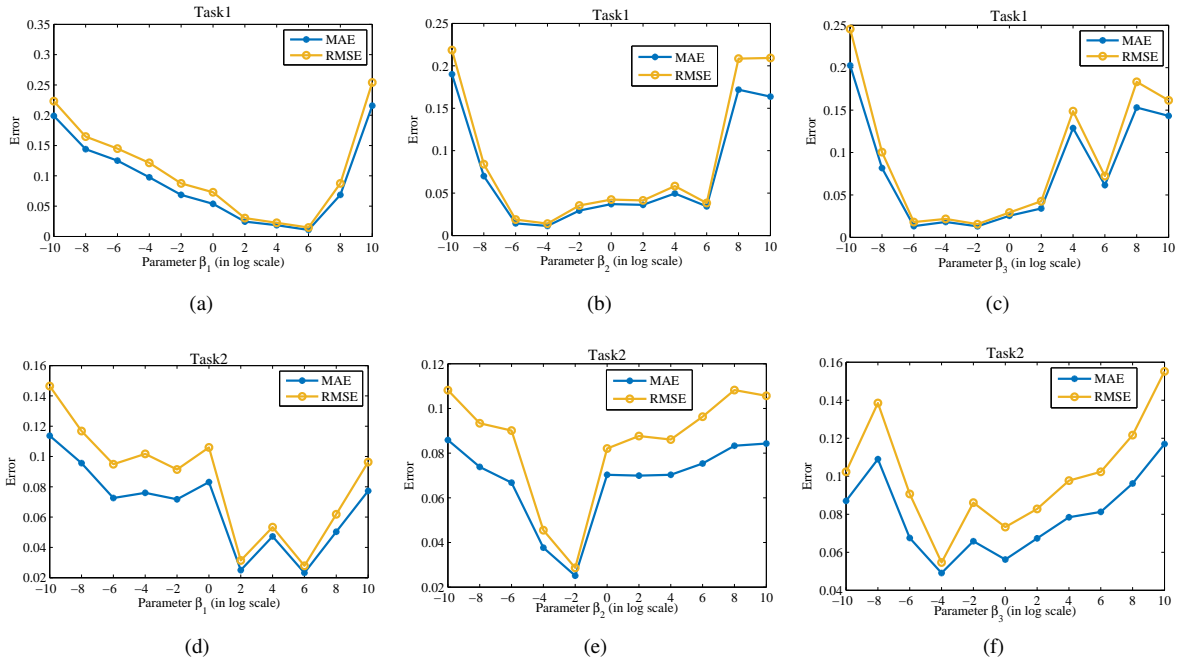
The sensitivity of the three key hyper-parameters  $\beta_1$ ,  $\beta_2$  and  $\beta_3$  in Eq. (9) is further checked.  $\beta_1$  is set on  $\ell_{AE}$ , while  $\beta_2$  and  $\beta_3$  is set on  $\ell_{LSTM}^{S,m}$  and  $\ell_{LSTM}^T$  respectively. These three parameters are used to control the tradeoff between the DAE reconstruction error, the source domain/target domain LSTM prediction error, and the tensor reconstruction error. Specifically, these three parameters are set by  $\beta_1 = 1$ ,  $\beta_2 = 1$ , and  $\beta_3 = 1$  in turn, and test the prediction performance with different values of the other parameter. The mean value of 30 repeated trials is calculated as the final results, as shown in Figure 11. Please note that the curves in Figure 11 are not smooth enough, which is caused by the noise. A smaller value of  $\beta_1$  will deteriorate the quality of initial DAE features, while a much larger one may cause overfitting. When  $\beta_2$  and  $\beta_3$  become larger, the model tends to seek a lower training error of the LSTM networks, relatively neglecting the adequacy of tensor representation, and vice versa. The ratio of  $\beta_2$  and  $\beta_3$  can also affect the significance of the prognostic knowledge from the source domain and target domain. These three hyper-parameters can be determined via cross-validation. In our experiment, the parameters are set  $\beta_1$ ,  $\beta_2$  and  $\beta_3$  to 1, 0.01 and 0.1, respectively.

The sensitivity of the three key hyper-parameters: learning rate  $\alpha$ , the hidden feature dimension of DAE and LSTM is also evaluated. Specifically, two of the three parameters are fixed in turn, then the mean value of 30 repeated trials is calculated as the final results, as shown in Figure 12. From the subfigures (a) and (d), the model is insensitive to the learning rate  $\alpha$  except for the too-large or small values. Moreover, the hidden feature dimension of DAE has a certain influence on the prediction results. The too-small dimension of DAE may ignore essential information, while the too-large dimension would contain redundant information, both of which are harmful to the representation of the





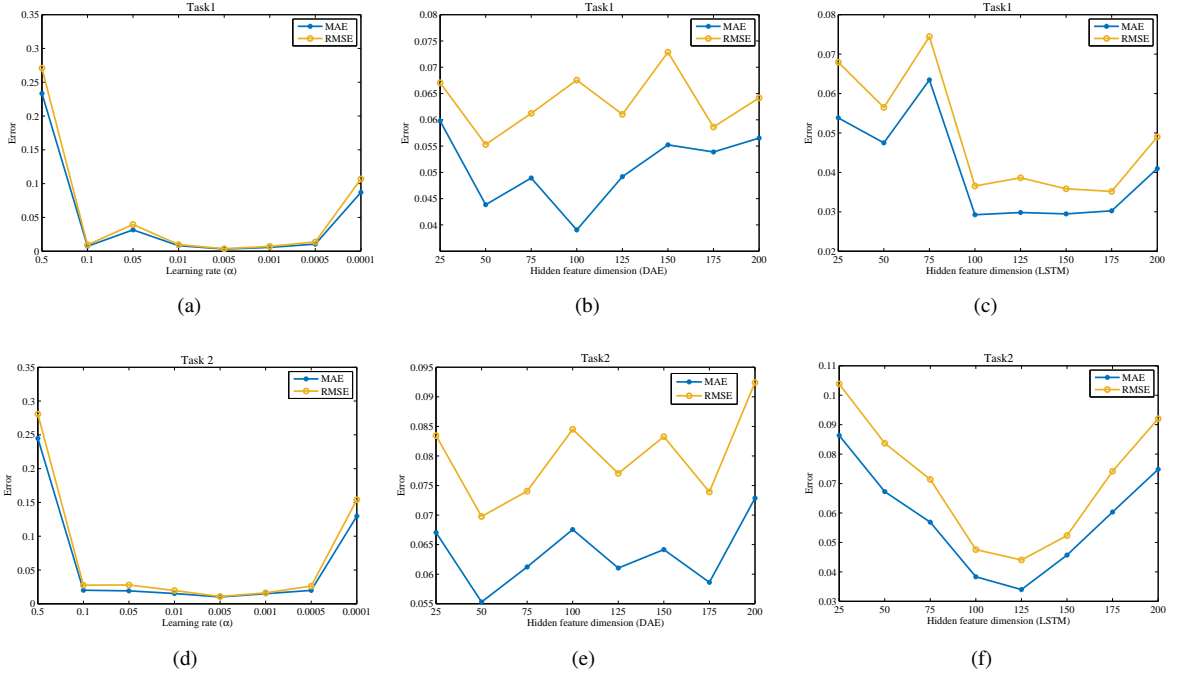
**Figure 10:** RUL prediction results of ablation experiments, where (a) is on Task1 and (b) is on Task2.



**Figure 11:** Sensitivity evaluation of different hyper-parameters settings in Eq. (9) in terms of the MAE and RMSE, where (a)-(c) are on Task1, (d)-(f) are on Task2.

meta-degradation trend via tensor decomposition. Due to random initialization, Figure 12(b) and (e) inevitably have some fluctuations. For the hidden feature dimension of LSTM, a larger value will also raise redundant information, while a smaller value may block extracting the prognostic information. According to Figure 12, the parameters are set  $\alpha = 0.001$ , the hidden feature dimension of DAE and LSTM, to be 50 and 100, respectively, in our experiment.

To evaluate the effect of DAE model's bias, the following two experiments in the revised manuscript are designed with: 1) different imbalance ratios of training samples, and 2) separated DAEs models in the two domains. Here only the DAE part is modified, while the remaining parts of the proposed method remain unchanged. The experimental settings are listed in Table 2. The prediction results and corresponding DAE features are shown in Fig. 13 and Fig. 14 respectively. The DAE model in the proposed method is less susceptible to the different sample ratios. The



**Figure 12:** Sensitivity evaluation of different network settings on prediction error in terms of the MAE and RMSE, where (a)-(c) are on Task1, (d)-(f) are on Task2.

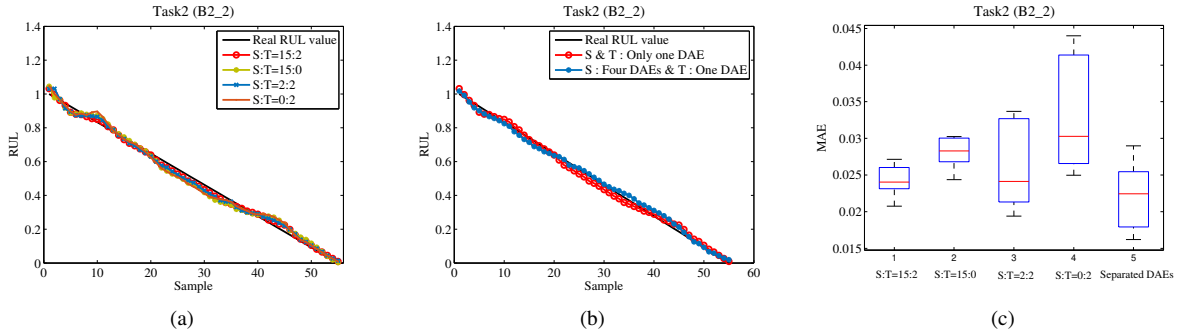
**Table 2**

Experimental settings on Task 2 with different ratios of bearing samples and separated DAE models. Trial 1 is just with the same setting used in this paper.

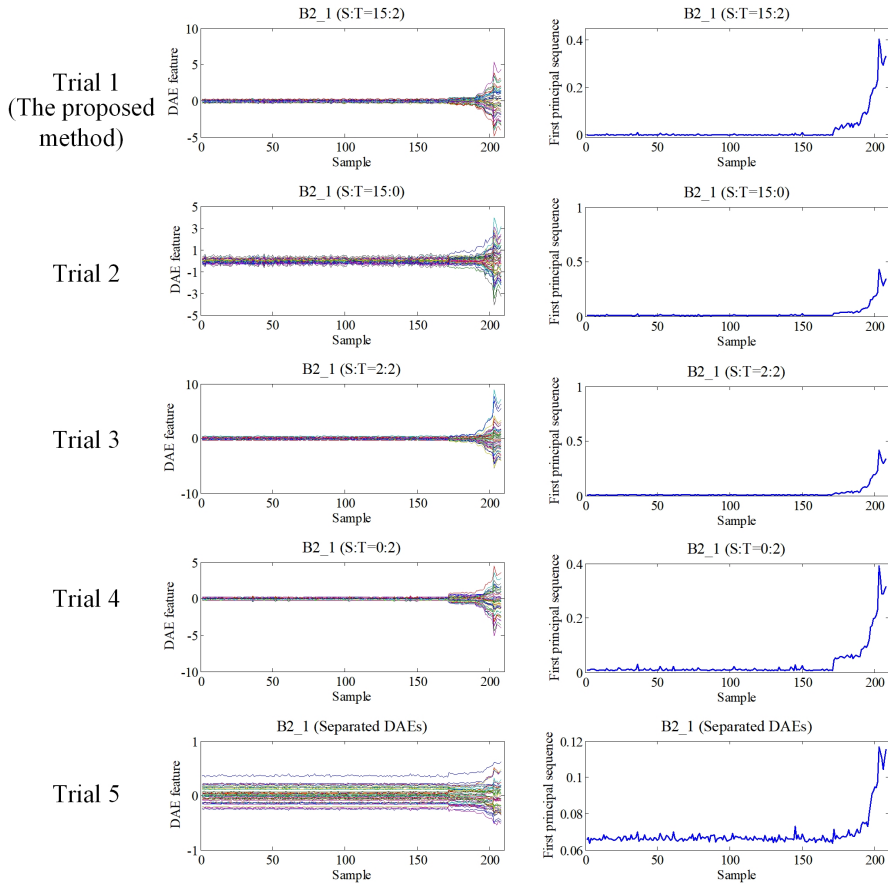
Trial index	Source domain	Target domain	Test bearing	Ratio (source domain(S):target domain (T))
1 (The proposed method)	XJTU (X1_1-X3_5)	PHM (B2_1,B2_6)	PHM(B2_2)	Bearing number S:T=15:2
2	XJTU (X1_1-X3_5)	-		Bearing number S:T=15:0
3	XJTU (X1_1,X1_5)	PHM (B2_1,B2_6)		Bearing number S:T=2:2
4	-	PHM (B2_1,B2_6)		Bearing number S:T=0:2
5	XJTU (X1_1-X3_5)	PHM (B2_1,B2_6)		Separated DAEs S: 4 DAEs & T:1 DAE

prediction results with different sample ratios are similar, while the feature sequences on the target bearing B2\_1 show relatively identical degradation trends. These results indicate that the DAE model is devoted to dimensionality reduction. Moreover, the model becomes more unstable when the sample size used for training is smaller. Once only the two target bearings (B2\_1 and B2\_6) are used for the DAE training, the MAE error deviation becomes maximum (see Figure 13(c)). In contrast, no matter of S:T=15:2 or S:T=15:0, the error deviation is much smaller. This phenomenon indicates that the DAE model in the proposed method is sensitive to the whole volume of training data more than the imbalance of training samples.

Also from Figure 13, we observe that running with separated DAEs can bring similar, but more unstable, prediction results than only using one DAE. More DAE models to be optimized will probably increase model complexity.



**Figure 13:** Prediction results on Task2 with (a) different ratios of bearing sample, (b) separated DAEs and (c) error box diagram.



**Figure 14:** DAE features (left column) and their first principal sequence (right column) of the target bearing B2\_1 from the PHM dataset. Here PCA is used for one-dimensional visualization.

Meanwhile, training separated DAEs definitely raises extra computational costs. Since the DAE model is just for dimensionality reduction, the proposed method prefers to adopt one DAE model, instead of separated DAEs, that are trained directly using all samples from the source domain and target domain. We believe this design will not deteriorate the feature extraction and prediction performance in the target domain, while the extraction of meta-degradation trends will also not be influenced negatively.

**Table 3**

Description of 14 classic methods for comparison.

Number	Method name	Method type	Implementation
1	Deutsch's method [3]	Deep learning with no transfer learning	DBN
2	Zraibi's method [4]		LSTM
3	Zhu's method [5]		Multiscale-CNN
4	KMM [36]	Transfer learning with shallow model	Kernel mean matching
5	SA [37]		Subspace alignment
6	GFK [38]		Geodesic flow kernel
7	TCA [39]		Kernel with MMD
8	Zhang's method [6]	Deep transfer learning	Bi-LSTM with fine-tuning
9	Cheng's method [40]		LSTM with CNN and MMD features
10	Zhu's method [7]		Multilayer perceptron with MMD
11	Sun's method [12]		Sparse DAE with K-L divergence
12	Mao's method [8]		LSTM with interpretability-based fine-tuning
13	Peng's method [41]		Multi-source transfer learning with fine-tuning
14	Hu's method [9]		Weighted DANN with fine-tuning

### 4.3. Comparative experiments

For comparison, 14 typical RUL prediction methods are chosen, as shown in Table 3. Methods 8-14 can be regarded as state-of-the-art (SOTA) RUL transfer prediction methods. Specifically, Methods 8-11 and 13-14 were designed for the prediction across working conditions. Method 12 is an interpretability analytics work for RUL transfer prediction, but it still tested the RUL transfer prediction across machines and provided the results. For the methods with shallow models (Methods 4-7), DAE is first adopted to extract 25-dimensional features, and the grid search with cross-validation is utilized to determine the optimal parameters. The average values of 50 repeated tests are used as the final results, as shown in Figure 15. The corresponding numerical comparisons are shown in Figure 16 and Table 4. Here, the root

mean square error ( $RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}$ ), mean absolute error ( $MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$ ) and Score  $= \frac{1}{n} \sum_{i=1}^n (A_i)$

are selected for performance evaluation, where  $A_i = \begin{cases} \exp^{-\ln(0.5) \cdot (Er_i/5)} & \text{if } Er_i \leq 0 \\ \exp^{+\ln(0.5) \cdot (Er_i/20)} & \text{if } Er_i > 0 \end{cases}$  and  $Er_i = 100 \times \frac{y_i - \hat{y}_i}{y_i}$ .

From Table 4, the proposed approach has the smallest RMSE & MAE and the highest Score.

For Task 1 and Task 2, Methods 4-7 get much worse results than the proposed approach due to the features with less representative capability. For data with large distribution differences, it is not easy to map the data into the same feature space to transfer the prognostic knowledge. Although Methods 1-3 adopt deep learning techniques that can better reflect the degradation characteristics, they are still inferior to the proposed approach. An interesting phenomenon is found that the prediction effect of these three methods is close to the effect of Methods 8 and 10-11. It indicates that only fine-tuning or domain adaptation cannot significantly improve the prediction performance once the feature representation is good enough. A more reasonable transfer strategy is required, verified by Method 9 that adopts a strategy of subspace adaptation plus fine-tuning. For Methods 12-14, their results are superior to Method 8 and Methods 10-11, which indicates that selective transfer of degradation knowledge is more beneficial to the target task with large data distribution discrepancy. However, their prediction performance is still worse than ours because these methods do not make a comprehensive information extraction and representation. Based on the sample saliency-based interpretability analysis, Method 12 achieved the effective transfer of prognostic knowledge and improved the transfer effect. But its results are still worse than ours because this method did not integrate fault mode information which is critical for the RUL prediction across machines.

Despite the UNSW dataset being collected at the laboratory, it mainly regards bearing fault (spall) severity assessment under simulated real-world operating conditions. Meanwhile, its degradation processes appear more randomness and noise interference than the other datasets. The prediction performances by all methods on Task 3 deteriorate markedly. Only two methods in all 14 methods, i.e., Mao's method (Method 12) and Peng's method (Method 13) can achieve certain prediction effects. The other 11 methods all fail to work, either resulting in large fluctuations (Methods 4, 9-10) or causing clearly distorted prediction values (Methods 1-3, 5-7, 8, 11, 14). Especially for the

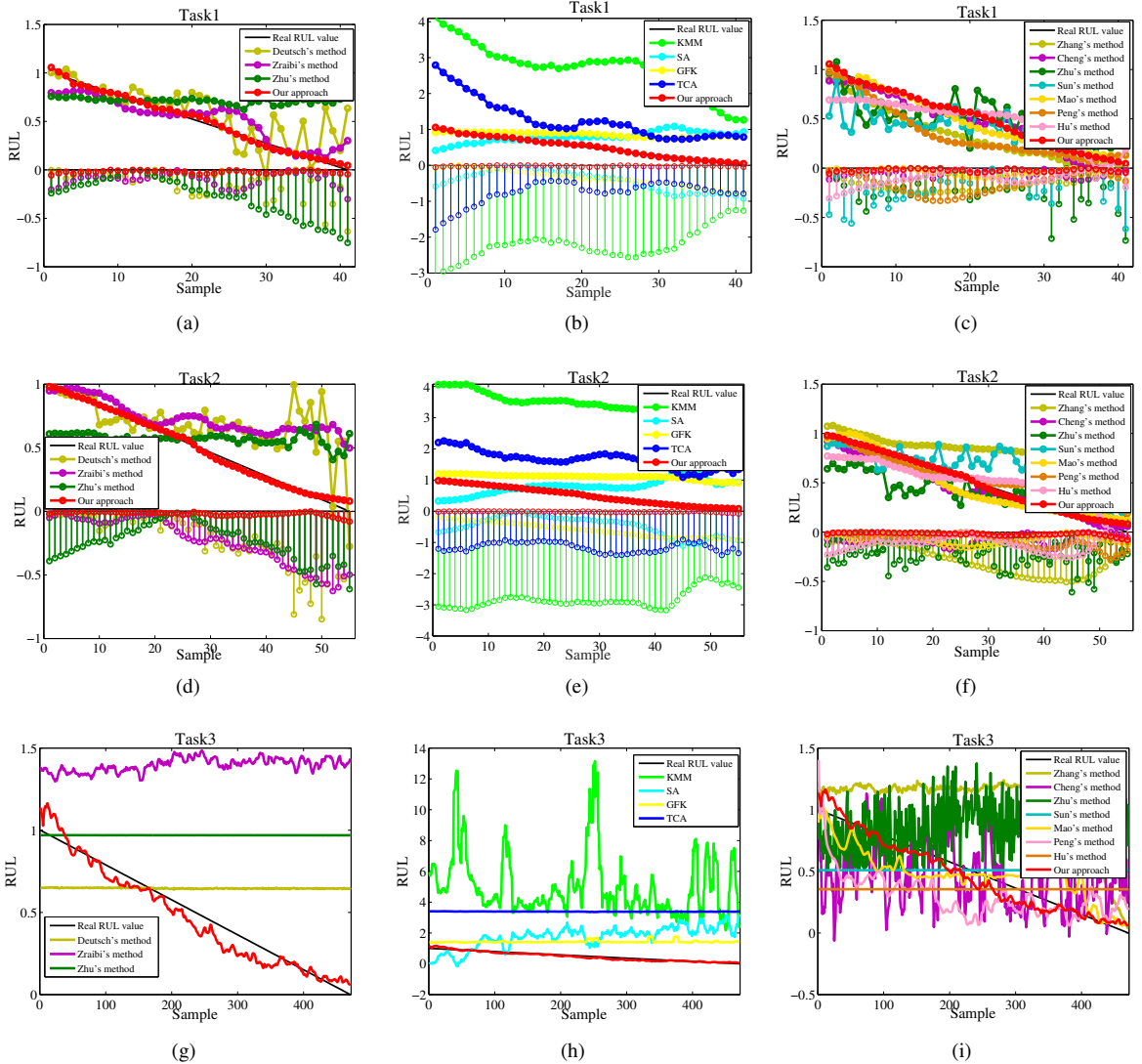


Figure 15: RUL prediction results of the total 13 methods, where (a)-(c) are on Task1, (d)-(f) are on Task2, (g)-(i) are on Task3.

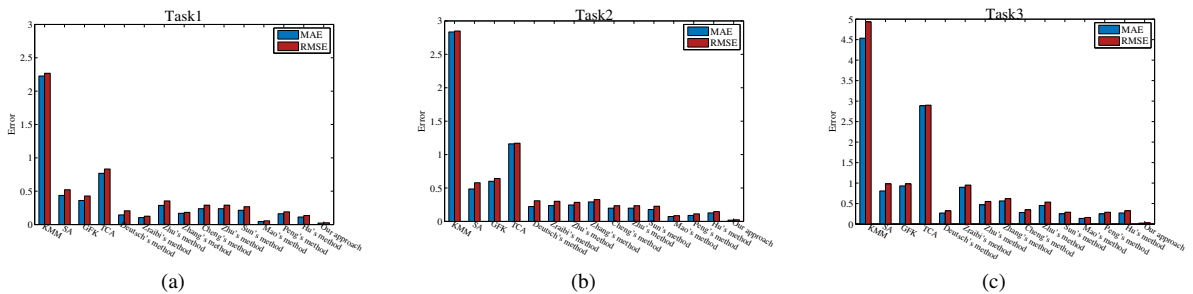


Figure 16: RMSE and MAE of total 15 methods, where (a) is on Task1, (b) is on Task2, (c) is on Task3

**Table 4**

Numerical results of prediction effect by all 15 methods. Lower RMSE and MAE, as well as higher Score, indicates better prediction performance.

Method	Task1			Task2			Task3		
	MAE	RMSE	Score	MAE	RMSE	Score	MAE	RMSE	Score
1	0.1446	0.2059	0.3308	0.2225	0.3090	0.2880	0.2702	0.3229	0.2318
2	0.1041	0.1252	0.3983	0.2370	0.3014	0.1861	0.8982	0.9517	4.094e-4
3	0.2870	0.3540	0.2316	0.2460	0.2855	0.2367	0.4707	0.5516	.0916
4	2.2261	2.2676	2.83e-18	2.8347	2.8487	1.58e-20	4.5353	4.9337	1.4142e-23
5	0.4371	0.5217	0.1539	0.4854	0.5792	0.1181	1.3989	1.5671	0.0444
6	0.3613	0.4283	0.1330	0.6006	0.6417	0.0042	0.9308	0.9813	2.0893-e4
7	0.7680	0.8324	7.78e-06	1.1607	1.1707	8.64e-09	2.8874	2.9014	1.0139e-16
8	0.1684	0.1825	0.2794	0.2916	0.3274	0.0549	0.5648	0.6200	0.0064
9	0.0644	0.0717	0.4925	0.0748	0.0901	0.5222	0.2814	0.3484	0.2012
10	0.2377	0.2910	0.2674	0.1992	0.2358	0.2693	0.4520	0.5354	0.1461
11	0.2139	0.2677	0.2706	0.1791	0.2276	0.2712	0.2505	0.2893	0.2307
12	0.0457	0.0561	0.6146	0.0737	0.0868	0.4672	0.1378	0.1612	0.3223
13	0.1619	0.1905	0.3227	0.0895	0.1134	0.4379	0.2490	0.2883	0.2313
14	0.1127	0.1352	0.3205	0.1278	0.1466	0.3307	0.2715	0.3236	0.2038
Ours	<b>0.0195</b>	<b>0.0253</b>	<b>0.6510</b>	<b>0.0224</b>	<b>0.0266</b>	<b>0.7201</b>	<b>0.0593</b>	<b>0.0717</b>	<b>0.5471</b>

latter, some of the methods remain essentially unchanged or in small fluctuation, which is completely meaningless. This comparative effect roots from the large data distribution divergence and less feature extraction on the complex degradation data. The results by Methods 8 and 10-11 indicate that only fine-tuning or domain adaptation cannot tackle the prediction task with large distribution divergence. Despite performing well on the PHM dataset, Method 14 still gets a worse prediction result, which is really surprising to us. Since Method 14 utilizes the discriminator output to realize weighted prediction and fine-tuning, its transfer strategy is unable to support the knowledge transfer for a complex degradation process. The results by Methods 12-13 are superior to the other deep transfer learning methods due to the selective or ensemble transfer strategies. However, their results are still worse than ours because they do not integrate fault mode information which is critical for the RUL prediction across machines. On the contrary, the proposed approach not only analyses the transferability of source domain data but also uses the transferability metric to facilitate the knowledge transfer. Targeted knowledge transfer can then be achieved to improve the transfer learning effect. Reasonable transferability analytics is believed to play a critical role in the RUL transfer prediction across machines.

The observed differences in performance could also be attributed to the dependence on data scale. Deep learning methods typically require a large amount of data to extract degradation knowledge from noise and other random interferences in the degradation process. On the contrary, the proposed knowledge transfer mechanism with transferability analytics can significantly improve the transfer efficiency and reduce the demand for high data volume due to the following merits: 1) Core tensor is employed to represent explicitly the degradation knowledge, decreasing the random interference in the degradation process; 2) Using core tensor as the input of LSTM can also improve the initial feature quality and avoid noise disturbance; 3) The designed alternating optimization scheme facilitates to find the optimal tensor representation and knowledge transfer effect, avoiding feature deterioration and reducing the overfitting risk on small-scale data; 4) The *M-TDI*-based transfer strategy is able to efficiently transfer the prognostic knowledge via selective freezing and fine-tuning, which further avoids the demand for large-scale data. In this experiment, the training data from one bearing are just from dozens to a thousand samples. On this data scale, our approach gets a much lower prediction error than the other deep learning methods.

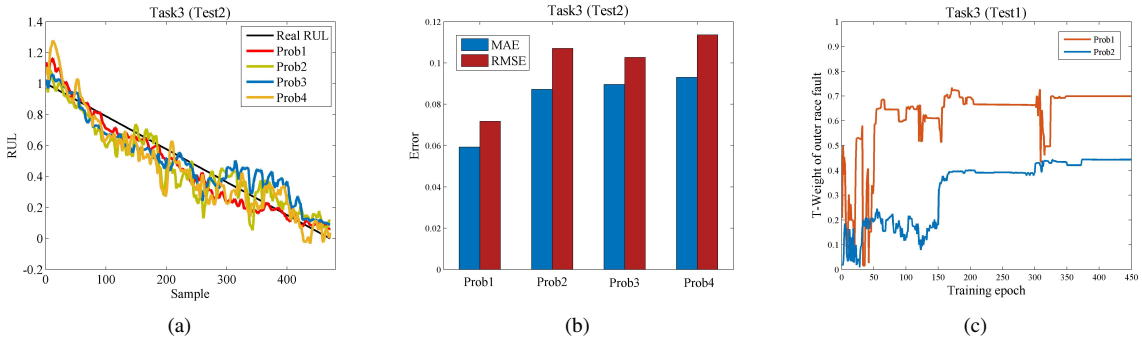
We further check the reliability of the proposed approach with incorrect fault probability. In the three tasks listed in Table 1, only Task 3 (UNSW) has known fault modes in the target domain. Therefore, we employ Task 3 for the verification. Through the CNN classifier, we can get the fault probability value of the target domain data: [0.8313 (outer race fault), 0.0293 (inner race fault), 0.0545 (cage fault), 0.0849 (mixed fault)], in which the classification result keeps line with the ground-truth information. We further modify the probability value to incorrect ones, as listed in

**Table 5**

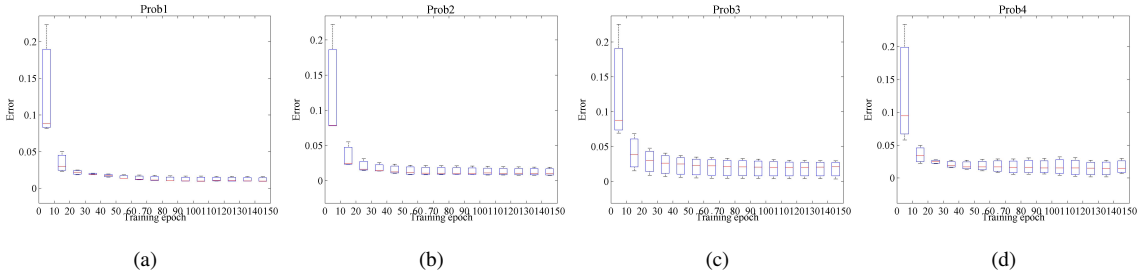
Settings of different fault probability on Task3. The first trial Prob1 is just with the same setting used in this paper. The ground-truth fault mode is the outer race fault.

Trial index	Fault probability ([outer race fault, inner race fault, cage fault, mixed fault])	Remark
Prob1	[0.8313, 0.0293, 0.0545, 0.0849]	Probability output of CNN classifier
Prob2	[0.0100, 0.9700, 0.0100, 0.0100]	Manually setting
Prob3	[0.0100, 0.0100, 0.9700, 0.0100]	Manually setting
Prob4	[0.0100, 0.0100, 0.0100, 0.9700]	Manually setting

Table 5. The prediction results and convergence performance are shown in Figures 17-18. Taking outer race fault as an example, Figure 17 also illustrates the change of *T-Weight*.



**Figure 17:** RUL prediction performance on Task3 with different values of fault probability, where (a) and (b) are the predicted RUL values and corresponding prediction errors respectively on the test bearing Test2, (c) is the *T-Weight* value of outer race fault on the training bearing Test1.



**Figure 18:** Box diagram of loss drops on Task3 with different values of fault probability.

Not surprisingly, the prediction results with incorrect fault probability are more biased than the proposed approach, with much unstable convergence during the training process. However, compared to Table 4, their prediction errors are still lower than the other comparative methods (Methods 1-14). We also observe a very interesting phenomenon from Fig. 17(c). Even if the initial probability value of outer race fault is set very small, the corresponding *T-Weight* value can still climb up to a much higher value along with the training. This result effectively proves the self-healing ability, or say reliability, of the proposed approach. Despite of incorrect fault probability, the proposed approach can adaptively recognize the useful prognostic knowledge via introducing degradation characteristic information and achieve effective knowledge transfer through the alternating optimization scheme. As a result, the transfer effect, as well as the RUL prediction performance, can be guaranteed as much as possible.

## 5. Conclusions

In this paper, a new selective transfer learning approach is proposed for RUL prediction across machines, which utilizes tensor representation-based transferability analytics. The proposed approach demonstrates the ability to reduce the negative influence of degradation randomness and noise disturbance through the use of tensor representation. Additionally, the approach leverages core tensors to determine the transferability of source domain data based on fault mode information and degradation characteristics. By quantifying the transferable degree of the source domain data, more prognostic knowledge can be transferred and the transfer learning effect can be improved. Notably, the proposed approach does not require the availability of fault mode information in the target domain, indicating better deployment capability.

The approach presented in this paper operates on the assumption that the source domain data contains explicit fault information. However, it is worth noting that this assumption is not always necessary. In cases where the fault categories in the source domain are unknown, one can use clustering or time-frequency signal analysis methods to roughly identify potential fault modes. It should be emphasized that the main idea of the proposed approach is to selectively extract and transfer degradation knowledge from the source domain data, and it does not rely on specific information about fault modes.

The introduction of transferability analytics can realize a targeted transition of prognostic knowledge, which improves the reliability of the transfer process and has been theoretically proven. The proposed approach can serve as a transfer learning framework since *M-TDI* and the modeling algorithm can be replaced by new similarity metrics and regression algorithms according to practical requirements. This framework is universally applicable and can also solve RUL prediction across different working conditions. In future work, the RUL transfer prediction in an online scenario will be studied. In practical applications, it is preferable to extract prognostic knowledge using online data collected sequentially.

## Acknowledgement

This work was supported by the National Natural Science Foundation of China [Grant No. U1704158, 61963026, and 61963026], the Henan Province Technologies Research and Development Project of China [Grant No. 212102210103].

## References

- [1] Zio E. Prognostics and Health Management (PHM): Where are we and where do we (need to) go in theory and practice. *Reliability Engineering & System Safety* 2022; 218:108119. <https://doi.org/10.1016/j.res.2021.108119>
- [2] Huang H, Wang H, Li Y, Zhang L, Liu Z. Support vector machine based estimation of remaining useful life: current research status and future trends. *Journal of Mechanical Science and Technology* 2015;29:151-163. <https://doi.org/10.1007/s12206-014-1222-z>
- [3] Deutsch J, He D. Using deep learning-based approach to predict remaining useful life of rotating components. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 2018;48(1):11-20. <https://doi.org/10.1109/TSMC.2017.2697842>
- [4] Zraibi B, Mansouri M, Loukili SE. Comparing deep learning methods to predict the remaining useful life of lithium-ion batteries. *Materials Today: Proceedings* 2022;62:6298-6304. <https://doi.org/10.1016/j.matpr.2022.04.082>
- [5] Zhu J, Chen N, Peng W. Estimation of bearing remaining useful life based on multiscale convolutional neural network. *IEEE Transactions on Industrial Electronics* 2019;66(4):3208-3216. <https://doi.org/10.1109/TIE.2018.2844856>
- [6] Zhang A, Wang H, Li S, et al. Transfer learning with deep recurrent neural networks for remaining useful life estimation. *Applied Sciences* 2018; 8(12):2416. <https://doi.org/10.3390/app8122416>
- [7] Zhu J, Chen N, Shen C. A new data-driven transferable remaining useful life prediction approach for bearing under different working conditions. *Mechanical Systems and Signal Processing* 2020;139:106602. <https://doi.org/10.1016/j.ymssp.2019.106602>
- [8] Mao W, Liu J, Chen J, Liang X. An interpretable deep transfer learning-based remaining useful life prediction approach for bearings with selective degradation knowledge fusion. *IEEE Transactions on Instrumentation and Measurement* 2022;71:1-16. <https://doi.org/10.1109/TIM.2022.3159010>
- [9] Hu T, Guo Y, Gu L, et al. Remaining useful life estimation of bearings under different working conditions via Wasserstein distance-based weighted domain adaptation. *Reliability Engineering & System Safety* 2022; 224: 108526. <https://doi.org/10.1016/j.res.2022.108526>
- [10] Xia P, Huang Y, Li P, Liu C, Shi L. Fault knowledge transfer assisted ensemble method for remaining useful life prediction. *IEEE Transactions on Industrial Informatics* 2022;18(3):1758-1769. <https://doi.org/10.1109/TII.2021.3081595>
- [11] Zhuang J, Jia M, Zhao X. An adversarial transfer network with supervised metric for remaining useful life prediction of rolling bearing under multiple working conditions. *Reliability Engineering & System Safety* 2022;225:108599. <https://doi.org/10.1016/j.res.2022.108599>
- [12] Sun C, Ma M, Zhao Z, Tian S, Yan R, Chen X. Deep Transfer Learning Based on Sparse Autoencoder for Remaining Useful Life Prediction of Tool in Manufacturing. *IEEE transactions on industrial informatics* 2019;15(4):2416-2425. <https://doi.org/10.1109/TII.2018.2881543>
- [13] Zhao K, Hu J, Shao H, et al. Federated multi-source domain adversarial adaptation framework for machinery fault diagnosis with data privacy. *Reliability Engineering & System Safety* 2023; 236:109246. <https://doi.org/10.1016/j.res.2023.109246>



- [14] Zhu R, Peng W, Wang D, Huang C. Bayesian transfer learning with active querying for intelligent cross-machine fault prognosis under limited data. *Mechanical Systems and Signal Processing* 2023; 183: 109628. <https://doi.org/10.1016/j.ymssp.2022.109628>
- [15] Deng Y, Du S, Wang D, Shao Y, Huang D. A calibration-based hybrid transfer learning framework for RUL prediction of rolling bearing across different machines. *IEEE Transactions on Instrumentation and Measurement* 2023; 72:1-15. <https://doi.org/10.1109/TIM.2023.3260283>
- [16] Mao W, Liu K, Zhang Y, Liang X, Wang Z. Self-Supervised Deep Tensor Domain-Adversarial Regression Adaptation for On-line Remaining Useful Life Prediction Across Machines. *IEEE Transactions on Instrumentation and Measurement* 2023;72:1-16. <https://doi.org/10.1109/TIM.2023.3265109>
- [17] Liu R, Yang B, Hauptmann AG. Simultaneous bearing fault recognition and remaining useful life prediction using joint-loss convolutional neural network. *IEEE Transactions on Industrial Informatics* 2020;16(1):87-96. <https://doi.org/10.1109/TII.2019.2915536>
- [18] Dong J, Cong Y, Sun G, Fang Z, Ding Z. Where and how to transfer: knowledge aggregation-induced transferability perception for unsupervised domain adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 2021. <https://doi.org/10.1109/TPAMI.2021.3128560>
- [19] Hu J, Zhong H, Yang F, Gong S, Wu G, Yan J. Learning Unbiased Transferability for Domain Adaptation by Uncertainty Modeling. *Computer Vision–ECCV 2022: 17th European Conference, 2022*, 223-241. <https://doi.org/10.1007/978-3-031-19821-2-13>
- [20] Yang B, Lei Y, Xu S, Lee CG. An optimal transport-embedded similarity measure for diagnostic knowledge transferability analytics across machines. *IEEE Transactions on Industrial Electronics* 2022;69(7):7372-7382. <https://doi.org/10.1109/TIE.2021.3095804>
- [21] Mansour Y, Mehryar M, Afshin R. Domain adaptation: Learning bounds and algorithms. *arXiv preprint arXiv:0902.3430* 2009.
- [22] Nguyen CN, Ho LST, Dinh VC, et al. Transferability Between Regression Tasks. *NeurIPS 2022 Workshop on Distribution Shifts: Connecting Methods and Applications*.
- [23] Cao Y, Jia M, Ding P, Ding Y. Transfer learning for remaining useful life prediction of multi-conditions bearings based on bidirectional-GRU network. *Measurement* 2021;178:109287. <https://doi.org/10.1016/j.measurement.2021.109287>
- [24] He R, Tian Z, Zuo M. A transferable neural network method for remaining useful life prediction. *Mechanical Systems and Signal Processing* 2023;183:109608. <https://doi.org/10.1016/j.ymssp.2022.109608>
- [25] Comon P, Luciani X, De Almeida A. Tensor decompositions, alternating least squares and other tales. *Journal of Chemometrics: A Journal of the Chemometrics Society* 2009;23(7-8)393-405. <https://doi.org/10.1002/cem.1236>
- [26] Hu C, Wang Y, Gu J. Cross-domain intelligent fault classification of bearings based on tensor-aligned invariant subspace learning and two-dimensional convolutional neural networks. *Knowledge-Based Systems* 2020;209:106214. <https://doi.org/10.1016/j.knosys.2020.106214>
- [27] Cerrada M, SÁnchez R V, Li C, et al. A review on data-driven fault severity assessment in rolling bearings. *Mechanical Systems and Signal Processing* 2018; 99: 169-196. <https://doi.org/10.1016/j.ymssp.2017.06.012>
- [28] Shi Q, Yin J, Cai J, Cichocki A, Yokota T, et al. Block hankel tensor arima for multiple short time series forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence* 2020;34(4):5758-5766. <https://doi.org/10.1609/aaai.v34i04.6032>
- [29] Kinney JB, Atwal GS. Equitability, mutual information, and the maximal information coefficient. *Proceedings of the National Academy of Sciences* 2014;111(9):3354-3359. <https://doi.org/10.1073/pnas.1309933111>
- [30] Shachaf G, Brutzkus A, Globerson A. A Theoretical Analysis of Fine-tuning with Linear Teachers. *Advances in Neural Information Processing Systems* 2021;34:15382-15394.
- [31] Nectoux P, Gouriveau R, Medjaher K, et al. PRONOSTIA: An experimental platform for bearings accelerated life test. *IEEE International Conference on Prognostics and Health Management, PHM'12*. 2012, p. 1-8.
- [32] Wang B, Lei Y, Li N, Li N. A hybrid prognostics approach for estimating remaining useful life of rolling element bearings. *IEEE Transactions on Reliability* 2020;69(1):401-412. <https://doi.org/10.1109/TR.2018.2882682>
- [33] Zhang H, Borghesani P, Randall RB, Peng Z. A benchmark of measurement approaches to track the natural evolution of spall severity in rolling element bearings. *Mechanical Systems and Signal Processing* 2022;166:108466. <https://doi.org/10.1016/j.ymssp.2021.108466>
- [34] Mao W, He J, Zuo M. Predicting remaining useful life of rolling bearings based on deep feature representation and transfer learning. *IEEE Transactions on Instrumentation and Measurement* 2019;69(4):1594-1608. <https://doi.org/10.1109/TIM.2019.2917735>
- [35] Huang NE, Wu M, Qu W, Long SR, Shen SSP. Applications of Hilbert–Huang transform to non-stationary financial time series analysis. *Applied stochastic models in business and industry* 2023;19(3):245-268. <https://doi.org/10.1002/asmb.501>
- [36] Huang J, Bretton A, Borgwardt K, et al. Correcting sample selection bias by unlabeled data. *Advances in neural information processing systems* 2006, vol. 19.
- [37] Fernando B, Habrard A, Sebban M, Tuytelaars T. Unsupervised visual domain adaptation using subspace alignment. *Proceedings of the IEEE international conference on computer vision*, 2013, pp. 2960-2967.
- [38] Gong B, Shi Y, Sha F, Grauman K. Geodesic flow kernel for unsupervised domain adaptation. *IEEE conference on computer vision and pattern recognition*, 2012, pp. 2066-2073. <https://doi.org/10.1109/CVPR.2012.6247911>
- [39] Pan SJ, Tsang IW, Kwok JT, Yang Q. Domain adaptation via transfer component analysis. *IEEE transactions on neural networks* 2020;22(2):199-210. <https://doi.org/10.1109/TNN.2010.2091281>
- [40] Cheng H, Kong X, Chen G, et al. Transferable convolutional neural network based remaining useful life prediction of bearing under multiple failure behaviors. *Measurement* 2021;168:108286. <https://doi.org/10.1016/j.measurement.2020.108286>
- [41] Peng C, Tao Y, Chen Z, Zhang Y, Sun X. Multi-source transfer learning guided ensemble LSTM for building multi-load forecasting. *Expert Systems with Applications* 2022; 202:117194. <https://doi.org/10.1016/j.eswa.2022.117194>
- [42] Koltchinskii V, Lounici K. Concentration inequalities and moment bounds for sample covariance operators. *Bernoulli* 2017;23(1):110-133. <https://www.jstor.org/stable/44075465>

## Appendix

Our approach utilizes freezing and fine-tuning to update the target model. From Eq. (22), the upper bound of  $R(\mathcal{V}, \mathcal{W})$  can be derived in an inductive mode. For RNN-kind model,  $s_0 = 0$ . For the  $t$ -th sample, assume there exists a constant  $\varepsilon_t$ , then the following inequation exists:

$$\left| (x_t^T \mathcal{V}_T + s_{t-1}^T \mathcal{W}_T) - (x_t^T \mathcal{V} + s_{t-1}^T \mathcal{W}) \right| \leq \varepsilon_t \quad (24)$$

Eq. (24) can be rewritten as:

$$\begin{aligned} & \left| (x_t^T \mathcal{V}_T + s_{t-1}^T \mathcal{W}_T) - (x_t^T \mathcal{V} + s_{t-1}^T \mathcal{W}) \right| \\ &= \left| x_t^T (\mathcal{V}_T - \mathcal{V}) + s_{t-1}^T (\mathcal{W}_T - \mathcal{W}) \right| \\ &\leq \left| x_t^T (\mathcal{V}_T - \mathcal{V}) \right| + \left| s_{t-1}^T (\mathcal{W}_T - \mathcal{W}) \right| \leq \varepsilon_t \end{aligned} \quad (25)$$

From Eq. (25),  $s_{t-1} \leq \frac{\varepsilon_t - |x_t^T (\mathcal{V}_T - \mathcal{V})|}{|\mathcal{W}_T - \mathcal{W}|}$  can be obtained. On the other hand, by substituting Eq. (24) into the definition of  $R(\mathcal{V}, \mathcal{W})$ ,  $R(\mathcal{V}, \mathcal{W})$  can be rewritten as:

$$R(\mathcal{V}, \mathcal{W}) \stackrel{\Delta}{=} E_{x \sim D} \left[ \left( (x_t^T \mathcal{V}_T + s_{t-1}^T \mathcal{W}_T) - (x_t^T \mathcal{V} + s_{t-1}^T \mathcal{W}) \right)^2 \right] \leq E_{x \sim D} (\varepsilon_t^2) \quad (26)$$

By taking  $s_{t-1}$  into Eq. (26), the upper bound of  $R(\mathcal{V}, \mathcal{W})$  can be derived by:

$$\begin{aligned} R(\mathcal{V}, \mathcal{W}) &= E_{x \sim D} \left[ \left( (x_t^T \mathcal{V}_T + s_{t-1}^T \mathcal{W}_T) - (x_t^T \mathcal{V} + s_{t-1}^T \mathcal{W}) \right)^2 \right] \\ &= E_{x \sim D} \left[ \left( x_t^T (\mathcal{V}_T - \mathcal{V}) + s_{t-1}^T (\mathcal{W}_T - \mathcal{W}) \right)^2 \right] \\ &\leq E_{x \sim D} \left[ \left( x_t^T (\mathcal{V}_T - \mathcal{V}) + (\varepsilon_t - |x_t^T (\mathcal{V}_T - \mathcal{V})|) \right)^2 \right] \\ &\leq 2E_{x \sim D} \left[ \left( x_t^T (\mathcal{V}_T - \mathcal{V}) \right)^2 + (\varepsilon_t - |x_t^T (\mathcal{V}_T - \mathcal{V})|)^2 \right] \end{aligned} \quad (27)$$

From Eqs. (26) and (27), the following inequation can be obtained:

$$2E_{x \sim D} \left[ \left( x_t^T (\mathcal{V}_T - \mathcal{V}) \right)^2 + (\varepsilon_t - |x_t^T (\mathcal{V}_T - \mathcal{V})|)^2 \right] \leq E_{x \sim D} (\varepsilon_t^2) \quad (28)$$

Eq. (28) can be rewritten as:

$$E_{x \sim D} \left[ (\varepsilon_t - 2x_t^T (\mathcal{V}_T - \mathcal{V}))^2 \right] \leq 0 \quad (29)$$

Obviously, for a square variable, its expectation cannot be less than zero. Therefore,  $\varepsilon_t = 2x_t^T (\mathcal{V}_T - \mathcal{V})$ . By substituting  $\varepsilon_t = 2x_t^T (\mathcal{V}_T - \mathcal{V})$  into Eq. (26),  $R(\mathcal{V}, \mathcal{W}) \leq 4E_{x \sim D} \left[ \left( x_t^T (\mathcal{V}_T - \mathcal{V}) \right)^2 \right]$  can be obtained. Since the value of  $V$  is not deterministic, the right part of this inequation cannot be calculated directly. The upper bound of  $R(\mathcal{V}, \mathcal{W})$  can then be transformed into seeking the upper bound of  $4E_{x \sim D} \left[ \left( x_t^T (\mathcal{V}_T - \mathcal{V}) \right)^2 \right]$ , i.e.,

$$\begin{aligned} E_{x \sim D} \left[ \left( x_t^T (\mathcal{V}_T - \mathcal{V}) \right)^2 \right] &= E_{x \sim D} \left[ \left( x_t^T (\mathcal{V}_T - P_{\perp} \mathcal{V}^0 - P_{\parallel} \mathcal{V}_T) \right)^2 \right] \\ &= E_{x \sim D} \left[ \left( x_t^T P_{\perp} (\mathcal{V}_T - \mathcal{V}^0) \right)^2 \right] \\ &= (\mathcal{V}_T - \mathcal{V}^0)^T P_{\perp}^T E_{x \sim D} [x_t x_t^T] P_{\perp} (\mathcal{V}_T - \mathcal{V}^0) \\ &= (\mathcal{V}_T - \mathcal{V}^0)^T P_{\perp}^T \sum P_{\perp} (\mathcal{V}_T - \mathcal{V}^0) \\ &= \left\| \sum^{0.5} P_{\perp} (\mathcal{V}_T - \mathcal{V}^0) \right\|^2 \end{aligned} \quad (30)$$

For all  $k$  frozen layers, Eq. (30) can be expressed as:

$$E_{x \sim D} \left[ (x_t^T (V - \mathcal{V}))^2 \right] = \sum_t \left\| x_t^T (V_T - \mathcal{V}^0) \right\|_{l \leq k}^2 + \left\| \sum^{0.5} P_{\perp} (V_T - \mathcal{V}^0) \right\|_{l > k}^2 \quad (31)$$

In Eq. (31),  $\sum_t \left\| x_t^T (V_T - \mathcal{V}^0) \right\|_{l \leq k}^2$  is a constant at the beginning of fine-tuning. According to the theorem from [30], there exists a constant  $c > 0$ , and for  $1 \leq m \leq d$ ,  $\lambda_m > 0$ , all  $\delta \geq 1$ ,  $\left\| \sum^{0.5} P_{\perp} (V_T - \mathcal{V}^0) \right\|_{l > k}^2$  has an upper bound:

$$\begin{aligned} \left\| \sum^{0.5} P_{\perp} (V_T - \mathcal{V}^0) \right\|_{l > k}^2 &\leq \frac{2 \left\| \Sigma - \tilde{\Sigma} \right\|^3}{\lambda_m^2} \left\| P_{\leq m} (V_T - \mathcal{V}^0) \right\|_{l > k}^2 \\ &\quad + 2 \left\| \Sigma - \tilde{\Sigma} \right\|^2 \left\| P_{> m} (V_T - \mathcal{V}^0) \right\|_{l > k}^2 \end{aligned} \quad (32)$$

Moreover, the theorem from [42] provides a high probability bound for  $\left\| \Sigma - \tilde{\Sigma} \right\|$ :

$$\left\| \Sigma - \tilde{\Sigma} \right\| \leq g(\lambda, \delta, n) = c \lambda_1 \max \left\{ \sqrt{\frac{\sum_t \lambda_t}{n \lambda_1}}, \frac{\sum_t \lambda}{n \lambda_1}, \sqrt{\frac{\delta}{n}}, \frac{\delta}{n} \right\} \quad (33)$$

Hence, Eq. (23) can be obtained by substituting Eq. (33) into Eq. (32).