Target-specific Adaptation and Consistent Degradation Alignment for Cross-Domain Remaining Useful Life Prediction

Yubo Hou ⬤, Mohamed Ragab ⬤, Min Wu ⬤, *Senior Member, IEEE,* Chee-Keong Kwoh ⬤, Xiaoli Li, *Fellow, IEEE,* Zhenghua Chen*, *Senior Member, IEEE*

Abstract—Accurate prediction of the Remaining Useful Life (RUL) in machinery can significantly diminish maintenance costs, enhance equipment up-time, and mitigate adverse outcomes. Data-driven RUL prediction techniques have demonstrated commendable performance. However, their efficacy often relies on the assumption that training and testing data are drawn from the same distribution or domain, which does not hold in real industrial settings. To mitigate this domain discrepancy issue, prior adversarial domain adaptation methods focused on deriving domain-invariant features. Nevertheless, they overlook target-specific information and inconsistency characteristics pertinent to the degradation stages, resulting in suboptimal performance. To tackle these issues, we propose a novel domain adaptation approach for cross-domain RUL prediction named TACDA. Specifically, we propose a target domain reconstruction strategy within the adversarial adaptation process, thereby retaining target-specific information while learning domain-invariant features. Furthermore, we develop a novel clustering and pairing strategy for consistent alignment between similar degradation stages. Through extensive experiments, our results demonstrate the remarkable performance of our proposed TACDA method, surpassing state-of-the-art approaches with regard to two different evaluation metrics. Our code is available at https://github.com/keyplay/TACDA.

Note to Practitioners—This paper introduces TACDA, a deep neural network crafted for domain adaptation, aimed at the task of predicting the remaining useful life (RUL) of machinery using time series data from multiple sensors under different operating conditions. In particular, it utilizes limited labeled data in one operating condition (source domain) and unlabeled data in another condition (target domain) to train a model. This well-trained model is capable of predicting the RUL in the target domain based on sensor data. Unlike existing methods, TACDA can maintain information specific to the target domain and align the data within the same degradation stage for better adaptation performance. A current limitation of TACDA is its need for simultaneous access to data from both the source and target domains for training. Future research will explore source-free domain adaptation, which relies solely on unlabeled data from the target domain and a model initially trained in the source domain.

Index Terms—Domain adaptation, remaining useful life prediction, target-specific information, degradation stage.

Yubo Hou, Mohamed Ragab, Min Wu, Xiaoli Li and Zhenghua Chen are with Institute for Infocomm Research (I²R), Agency for Science, Technology and Research (A*STAR), Singapore. (Email: yubo002@e.ntu.edu.sg, mohamedr002@e.ntu.edu.sg, wumin@i2r.a-star.edu.sg, xlli@i2r.a-star.edu.sg, chen0832@e.ntu.edu.sg)

Yubo Hou, Chee-Keong Kwoh and Xiaoli Li are with School of Computer Science and Engineering, Nanyang Technological University, Singapore. (Email: asckkwoh@ntu.edu.sg)

*Corresponding author: Zhenghua Chen

I. INTRODUCTION

Prognostic and Health Management (PHM) of industrial systems and equipment plays a crucial role across diverse industries, including manufacturing, aerospace, and energy. PHM helps detect potential failures and forecast the lifespan of critical components, ultimately improving operational safety and efficiency. In aerospace, for instance, a significant percentage of aircraft breakdowns are due to turbofan engine failures, underscoring the urgent need for advanced PHM solutions that can prevent catastrophic incidents and reduce costly unscheduled maintenance. Similarly, in energy production, unplanned downtime due to equipment failure can lead to substantial financial losses and disrupt energy supply, emphasizing the value of PHM in ensuring consistent operations. By leveraging PHM, organizations can enhance equipment reliability, reduce maintenance costs, and optimize performance, ultimately minimizing the risk of unexpected failures [1]. Within the domain of PHM, accurately predicting the Remaining Useful Life (RUL) of machinery is pivotal, as it enables proactive maintenance decisions. Various approaches have been proposed for RUL prediction, which can generally be classified into three categories: model-based approaches, data-driven approaches, and hybrid approaches. Specifically, model-based approaches focus on describing the degradation behavior of a system using a mathematical or physics model, which requires a strong theoretical understanding [2], [3]. As the mechanical structure becomes increasingly complex, predicting RUL with model-based approaches can be extremely challenging. Meanwhile, with the increasing availability of data from deployed sensors, data-driven approaches become popular for RUL prediction [4], [5], [6], [7]. Hybrid approaches utilize both the pre-established model of model-based approaches and prediction ability of data-driven approaches based on historical data, which can improve the accuracy of prognostic result. For example, [8] integrates an unscented Kalman filter with complete ensemble empirical mode decomposition and a relevance vector machine, while [9] couple a particle filter with a relevance vector machine.

Despite the promise of data-driven approaches, their success is mainly attributed to two main assumptions, i.e., sufficient labeled data and independent and identically distributed (IID) data [10]. However, these assumptions are often impractical in real industrial settings for the subsequent reasons. First, the acquisition of annotated data (failures) incurs significant costs.
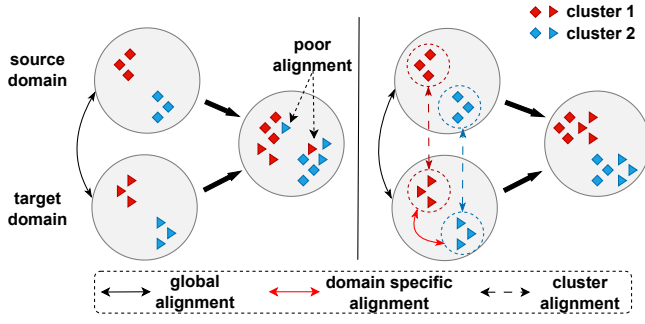
Fig. 1: Comparisons between previous UDA methods and ours. Left: only pursuing global alignment results in poor alignment. Right: considering target domain-specific information and inconsistency characteristics of clusters reduces the feature discrepancy.

Additionally, the degradation process of machines can span prolonged periods, further constraining the accessibility of faulty data [11]. Secondly, considering the dynamics of a real environment, a model is usually trained under one operating condition and tested under different operating conditions, leading to a significant performance deterioration due to the domain shift problem [12].

Given the aforementioned challenges, accurately predicting the RUL for machines under various working conditions with limited labeled data poses significant difficulties. Unsupervised domain adaptation (UDA) is a promising technique that facilitates knowledge transfer from a labeled source domain to a distinct yet related unlabeled target domain [12]. Recently, there has been an increased focus on UDA for RUL prediction task. Predominantly, the existing methods endeavor to identify domain-invariant features to mitigate the distribution shift problem, either through adversarial training [13], [14], [10] or by minimizing the statistical distance between domains [15], [16], [17]. Despite the promise of these methods in addressing the domain shift problem with scarce target labels, they still exhibit the following limitations.

- **Lack of target-specific information**: Solely pursuing domain-invariant features does not assure optimal performance in the target domain [18], [19]. Enforcing target domain features to globally resemble the source domain features without appropriate constraints may lead to the removal of target-specific information.
- **Inconsistent characteristics for degradation stages**: RUL degradation can be categorized into distinct stages based on degradation speed [20]. Each stage possesses unique characteristics. Existing methods bring the two domains closer globally, which may blend samples from different degradation stages [21], [22]. For instance, early-stage degradation in the target domain might be aligned with late-stage degradation in the source domain, resulting in misalignment.

As presented in Fig. 1 (left), the above-mentioned limitations could potentially hinder the overall model performance. In response to these challenges, we propose a novel target-specific adaptation and consistent degradation alignment

(TACDA) approach for machine RUL prediction across different working conditions. Specifically, to tackle the lack of target-specific information issue, we design an auto-encoder based domain alignment module with a soft dynamic time warping (soft-DTW) loss to preserve target domain information for time series sensory data. To address the ignore of inconsistency characteristics for the degradation stage problem, we not only make domain alignment on global data but also make degradation alignment between source and target domains. Through extensive experimentation, we have thoroughly evaluated the performance of our proposed TACDA method in accurately predicting the RUL of machines across diverse operating conditions.

The main contributions of this study are listed as follows.
- We propose an innovative approach called TACDA, which considers both the target-specific adaptation and consistent degradation alignment for RUL prediction.
- We leverage the auto-encoder to preserve target-specific information, while learning invariant representations. In particular, we employ soft-DTW instead of mean squared error as reconstruction loss within the auto-encoder to preserve target-specific information.
- We propose a clustering and pairing strategy, which clusters the data into different degradation stages and then aligns the data within the same degradation stage for better adaptation performance.

II. RELATED WORKS

A. Deep Learning for RUL Prediction

Deep learning for RUL prediction is broadly categorized into two types: feed-forward neural networks and recurrent neural networks (RNNs). In [23], convolutional neural networks (CNNs) was utilized to extract features at multi-scale for detecting fault growth and predicting machine RUL. In [24], a CNN network was proposed with a joint loss function to simultaneously identify faults and provide predictions for RUL. Another study [25] employed a Deep Belief Network (DBN) for feature extraction and RUL prediction.

For dynamic systems, RNNs offer sequential modeling capabilities. Among RNN architectures, long short-term memory (LSTM) networks have gained significant popularity due to their ability to capture long-term dependencies and overcome the vanishing gradient problem associated with traditional RNNs. In [26], a bidirectional LSTM (BiLSTM) methodology was introduced, which is enhanced with auxiliary features to forecast the RUL across diverse operational conditions. The bidirectional nature of BiLSTM allows it to capture dependencies from both past and future time steps. In [27], an attention-based LSTM approach was developed for RUL prediction. This method dynamically selects important features using attention mechanisms, resulting in accurate RUL predictions. However, these methods cannot work well with insufficient labeled data and non independent and identically distributed data in real world application.

B. Unsupervised Domain Adaptation for RUL Prediction

Unsupervised Domain Adaptation (UDA) for RUL prediction aims to reduce labeling costs by training neural networks

to transfer knowledge from a labeled source domain to an unlabeled target domain. Existing UDA methods focus on minimizing domain discrepancy to achieve high performance on the target domain. These methods can be categorized into two branches: metric-based and adversarial-based methods.

Metric-based methods align the feature representations of source and target domains by leveraging statistical distances. Deep Domain Confusion (DDC) [15], AdvSKM [28] and FDMTCN [29] utilize the maximum mean discrepancy (MMD) to mitigate domain differences. To address distributional shifts, TCNN [16] incorporates a multi-kernel MMD approach for better alignment between source and target domains. Correlation alignment (CORAL) [30] aims to reduce the covariance shift between feature distributions across domains. Additionally, in [17], a hybrid strategy combining the contractive denoising autoencoder and transfer component analysis (TCA) was implemented.

Adversarial-based methods use domain discriminator networks to encourage the feature extractor to learn domain-invariant representations. The domain adversarial neural network (DANN) [10], [31] employs a gradient reversal strategy to facilitate adversarial training between the domain classifier and the feature extractor. Similarly, the adversarial domain adaptation for remaining useful life prediction (ADARUL) [13] adopts a traditional GAN loss with label flipping to achieve domain-invariant feature learning. Adversarial regressive domain adaptation approach [14] used discriminator and bi-regressors to align distributions for infrared thermography-based RUL prediction. In [32], a transformer-based method was proposed which leverages feature-level and semantic-level discriminators for distribution alignment. In [33], a method was proposed that integrates clustering for sensor selection and ensemble technology for transfer learning. However, these methods do not incorporate target-specific information during the process of learning domain-invariant features, which harms the performance in target domain. In [7], a method was proposed that leverages an adversarial framework that separately aligns the marginal distributions of different operation phases between source and target domains.

III. METHODOLOGY

A. Problem Formulation

We denote a source domain with N_S labeled samples $\{X_S^i, y_S^i\}_{i=1}^{N_S}$ and a target domain with N_T unlabeled samples $\{X_T^i\}_{i=1}^{N_T}$, where $X_S^i \in \mathbb{R}^{M \times L}$ and $X_T^i \in \mathbb{R}^{M \times L}$ are both multivariate time series data consisting of M sensors and L time steps. y_S^i is the RUL label. We aim to transfer knowledge from labeled source domain to unlabeled target domain and then improve the performance of RUL prediction on the target. Table I provides a summary of the notations employed in this paper.

B. Overview

The overall structure of our proposed TACDA method is shown in Fig. 2. TACDA consists of three main components: source encoder E_S, target encoder E_T, and shared predictor R. First, we pretrain the source encoder E_S and predictor

TABLE I: List of notations.

Notation	Definition
X_S/X_T	source/target data
y_S	source RUL label
N_S/N_T	number of source/target samples
f_S/f_T	source/target features
E_S/E_T	source/target encoder
D_T	target decoder
D	domain discriminator
R	predictor
M	number of sensors
L	sequence length

R to learn the RUL distribution from the source domain. Then, in the adaptation phase, we train the target encoder E_T. This adaptation phase is divided into two stages. In the first stage, both source and target data are fed to the target-specific adaptation (TA) module to preserve target-specific information while learning invariant representations. To ensure consistent degradation alignment, in the second stage, source and target data are clustered, respectively. Then, data groups from source and target domains are paired by similar status. In the end, multiple paired source and target data are fed into the TA module successively for fine-tuning the model parameters from the first stage. Once training is complete, we use the well-trained target encoder E_T and predictor R to estimate RUL in the target domain. The following sections will introduce each module in detail.

C. Pretraining on Source Domain

During this phase, the objective is to train a model using the labeled source domain data to have the knowledge about the RUL prediction task. Specifically, a source encoder E_S and RUL predictor R are trained on source data X_S. The source encoder extracts features from source data: $f_S = E_S(X_S)$. The RUL predictor estimates RUL value according to the extracted features. The source encoder and the RUL predictor are optimized using the mean square error loss:

$$L_{mse} = \frac{1}{N_S} \sum_{i=1}^{N_S} (y_i - \widehat{y_i})^2, \tag{1}$$

where $\widehat{y_i} = R(E_S(X_S))$ is the estimated RUL value, y_i is the true RUL label, and N_S represents the number of samples in source domain.

D. Target-specific Adaptation

Given the pretrained source model, the focus lies in achieving adaptation upon the unlabeled target data. To do so, the target-specific adaptation module is proposed, containing adversarial adaptation for domain invariant feature and the decoder for preserving target-specific information. The target-specific adaptation module, depicted in Fig. 2 (left), comprises a trained source encoder E_S, a target encoder E_T, a target decoder D_T and a domain discriminator D. The E_T is initialized with the weights of E_S which was pre-trained on labeled source data $\{X_S, y_S\}$. The source domain feature f_S and target domain features f_T are extracted from source data X_S and target X_T via E_S and E_T, respectively. Then extracted
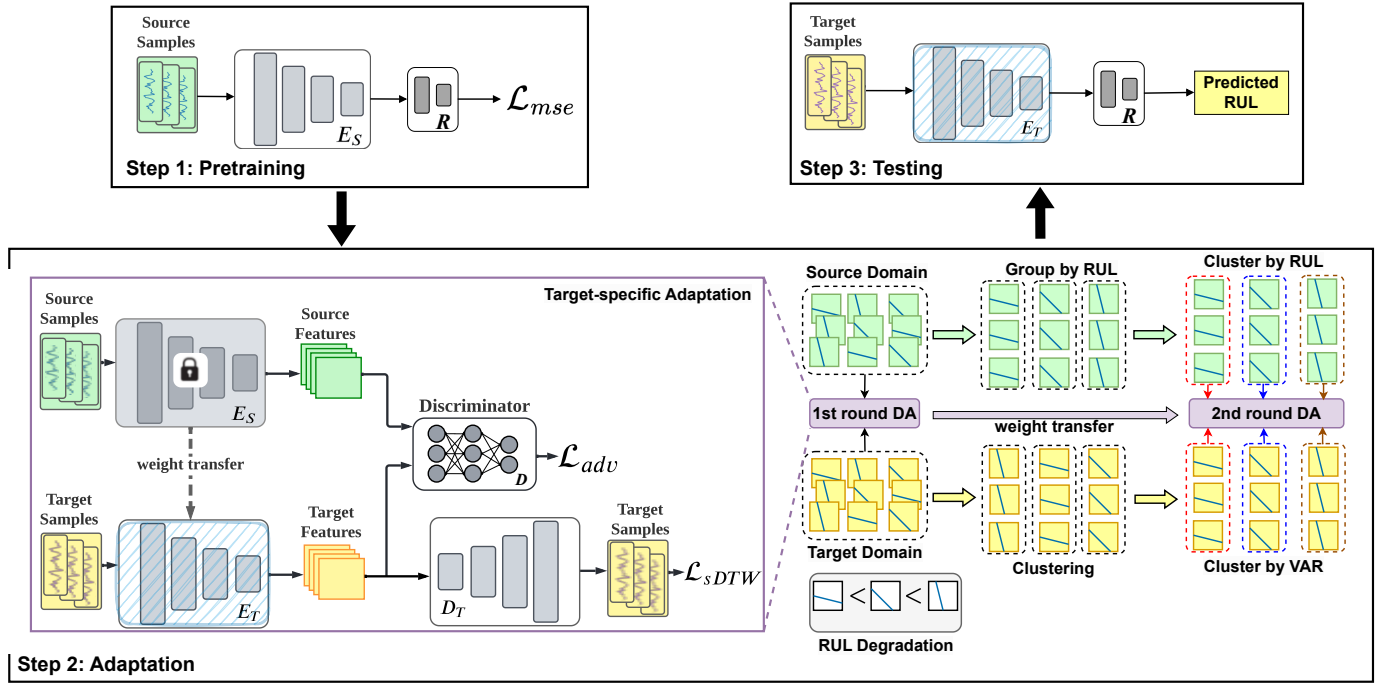
Fig. 2: Our proposed TACDA approach. Target-specific Adaptation (TA) learns the target encoder E_T, the target decoder D_T and discriminator D via optimizing the adversarial loss $\mathcal{L}_{adv}$ and the soft-DTW loss $\mathcal{L}_{sDTW}$. Specifically, TA is applied in two rounds: 1st Round DA: All source and target data are fed into TA for global domain alignment. 2nd Round DA: Clustered and paired source-target data (grouped by degradation stages, e.g., stage 1, 2, 3) are iteratively aligned in a loop (stage 1 → stage 2 → stage 3) for stage alignment. After the 2nd round DA, the optimized E_T is directly transferred to Step 3.

features f_S and f_T are fed into the domain discriminator D to minimize their discrepancy. Simultaneously, the target features f_T are passed through the decoder D_T to get reconstructed target data X'_T. The difference between X_T and X'_T is computed to preserve target-specific information during alignment. The formalized procedure of the target-specific adaptation module in our approach is outlined in Algorithm 1. The detailed procedures are described in the subsequent paragraphs.

1) Adversarial Adaptation: The training procedure of adversarial adaptation between the domain discriminator D and the target encoder E_T can be defined as follows:

$$\min_{E_T} \max_{D} \mathcal{L}_{adv} = \mathbb{E}_{X_S \sim P_S} \left[logD(E_S(X_S)) \right]$$
$$+ \mathbb{E}_{X_T \sim P_T} \left[log(1 - D(E_T(X_T))) \right]. \quad (2)$$

The target encoder E_T is trained to minimize $\mathcal{L}_{adv}$, while the discriminator D is updated in an adversarial manner to maximize $\mathcal{L}_{adv}$. Consequently, the trained target encoder E_T becomes capable of extracting features f_T that have minimum disparity compared to the source features f_S.

2) Target-specific Decoder: Adversarial domain adaptation demonstrates efficacy in learning target domain features that exhibit invariance with respect to the source features. Nevertheless, in its pursuit to minimize the adversarial loss, it runs the risk of removing task-specific information from the target features, thereby potentially compromising performance in the target domain. Consequently, it is imperative to preserve target-specific information throughout the process of domain adaptation. To achieve this objective, we develop an auto-
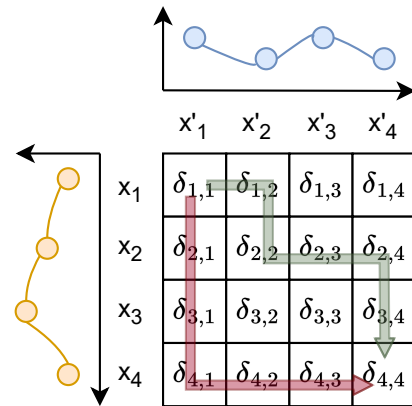

Fig. 3: Two warping paths (red and green, as well as the top-left and bottom-right entries) between two time series of length 4. The cost of a warping path is determined by the cumulative sum of difference δ visited along the path.

encoder based on soft-DTW loss [34], which is robust to the shifts and dilatation across the time dimension compared with conventional MSE loss.

Soft-DTW measures the similarity between X_T and X'_T by considering all possible warping path mapping the elements of X_T to X'_T. Fig. 3 shows an example of warping path between two time series of length 4. Soft-DTW can be expressed as follows:

$$\mathcal{L}_{sDTW} = -\gamma log \sum_{A \in \mathcal{A}_{L,L}} e^{-\langle A, \triangle(X_T, X_T') \rangle / \gamma}, \qquad (3)$$

where γ is a smoothing parameter, $\triangle(X_T, X_T') = [\delta(x_i, x_j')]_{ij} \in \mathbb{R}^{L \times L}$, $\delta(x_i, x_j')$ is the quadratic Euclidean distance and $\langle A, \triangle(X_T, X_T') \rangle$ is inner product. The set $\mathcal{A}_{L,L} \subset \{0,1\}^{L \times L}$ represents the collection of (binary) warping paths on an $L \times L$ matrix. These paths connect the upper-left matrix entry $(1,1)$ to the lower-right one (L, L) using exclusively $\downarrow$, $\rightarrow$, and $\searrow$ moves.

Algorithm 1: Target-specific Adaptation (TA)

Input: Source domain: $\{X_S, y_S\}$, Target domain: X_T
Output: Trained target encoder E_T, Trained target
       decoder D_T, Trained discriminator D
1   $E_S \leftarrow$ Pretrained source encoder
2   $E_T \leftarrow$ Initialized with E_S parameters
3   $D_T \leftarrow$ Target decoder
4   $D \leftarrow$ Domain discriminator
5   **for** *number of iterations* **do**
6      $f_S \leftarrow E_S(X_S)$ // Extract source
           features
7      $f_T \leftarrow E_T(X_T)$ // Extract target
           features
8   Feed f_S and f_T to D
9   Calculate adversarial loss $\mathcal{L}_{adv}$ using Eq. (2)
10   Update D by $\mathcal{L}_{adv}$
11     $X_T' \leftarrow D_T(f_T)$ // Reconstruct target
           sample
12   Calculate soft-DTW loss $\mathcal{L}_{sDTW}$ using Eq. (3)
13   Update D_T by $\mathcal{L}_{sDTW}$
14   Update E_T by $\mathcal{L}_{adv} + \lambda \mathcal{L}_{sDTW}$
15 **return** E_T, D_T, D

E. Consistent Degradation Alignment

The target-specific adaptation module aligns the distributions of source and target data on a global scale. However, it overlooks discrepancies in characteristics across different degradation stages, leading to suboptimal alignment. To address this issue, we propose a consistent degradation alignment module to ensure alignment between analogous degradation stages during the adaptation phase. Initially, we classify the degradation of both source and target domains into three stages based on their degradation speed: sluggish, accelerated, and terminal.

1) Identifying Source Degradation Stages: For the source domain data, we partition each engine's lifecycle into three degradation stages using available labels. To define precise stage boundaries, we compute a health index (HI) through a linear combination of critical sensor readings, following methodologies in [11], [20]. As illustrated in Fig. 4, which plots the HI trajectory of a representative engine against operational cycles, this index quantifies progressive degradation by synthesizing multi-sensor signals. The HI trajectory is analyzed in conjunction with RUL labels to identify distinct
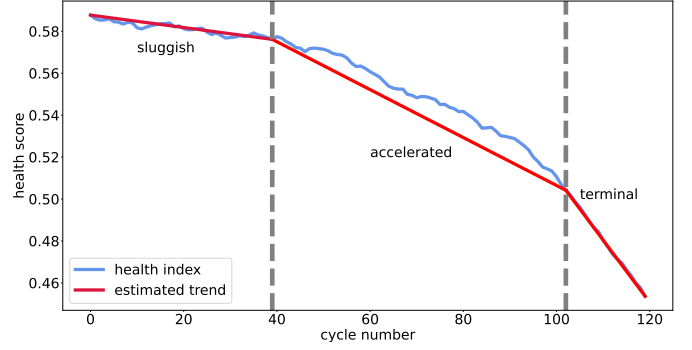

Fig. 4: Three degradation stages categorized by the health index.
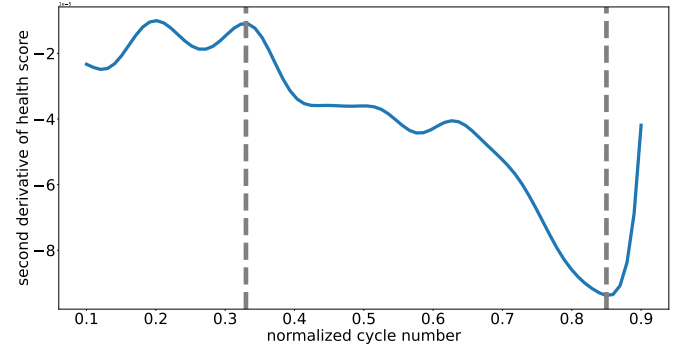

Fig. 5: Second derivative of health index.

degradation phases. The second derivative of the averaged HI reflects how the rate of health degradation (first derivative) changes over normalized cycle numbers.

As outlined in Algorithm 2, we begin by discretizing the normalized RUL values $y_i \in [0,1]$ into K equally spaced bins. Each bin k accumulates the health-index observations h_i whose corresponding y_i falls into that bin, and we track the count of samples per bin to enable averaging. After iterating over all N samples, we obtain an average health index $\overline{h}_k$ at each bin. To reduce noise and highlight meaningful curvature in the degradation profile, we apply a Gaussian smoothing filter to the sequence $\{\overline{h}_k\}_{k=0}^{K}$. Finally, we approximate the second derivative of HI. In our experiments, we set $K = 100$.

As shown in Fig. 5, we observe varying degradation rates across the lifecycle: initial slow decline transitions to accelerated deterioration before toward failure. Based on this pattern, we empirically define three lifecycle intervals:

- Sluggish stage (0%–33%): The second derivative maintains negative values (close to zero), indicating stable system dynamics. This results in a slow decrease of the HI.
- Accelerated stage (33%–85%): The second derivative shows persistent decrease (becoming more negative). This drives accelerated downward trends in the first derivative, leading to decelerating decay patterns in the HI curve.
- Terminal stage (85%–100%): The second derivative begins increasing (less negative) while remaining in negative territory. This causes HI decay to approach linear

---

Algorithm 2: Compute Second Derivative of the Averaged Health Index

---

Input: RUL label: $y \in [0, 1]$, Health index: h
Output: Second Derivative of HI: $\overline{h}''$

1 Initialize bins $b_1, \ldots, b_K \leftarrow 0$
2 Initialize counts $c_1, \ldots, c_K \leftarrow 0$
3 for $i \leftarrow 1$ **to** N **do**
4 $k \leftarrow \lfloor y_i \times K \rfloor$
5 $b_k \leftarrow b_k + h_i$
6 $c_k \leftarrow c_k + 1$
7 for $k \leftarrow 1$ **to** K **do**
8 $\overline{h}_k \leftarrow b_k / c_k$
9 Apply Gaussian smoothing to the sequence $\{\overline{h}_k\}_{k=0}^{K}$
10 $\overline{h}'' = (\overline{h}(k) - \overline{h}(k-1)) - (\overline{h}(k-1) - \overline{h}(k-2))$
11 return $\overline{h}''$

---

behavior with highest degradation rate.

2) Identifying Target Degradation Stages: In the target domain, classifying into the three degradation stages becomes a challenge due to the absence of labels. To address this, we propose an unsupervised clustering approach based on the k-means algorithm for the target domain data. To consider the temporal sequences in the data, soft dynamic time warping is integrated as a distance metric in the k-means algorithm. After the clustering process, it can be challenging to ascertain the specific degradation stage (e.g., sluggish, accelerated, or terminal) of each target cluster in the absence of labels. We posit that there is a correlation between degradation speed and the variance of sensor readings. Specifically, a faster degradation speed should correspond to higher variance values, and conversely, slower degradation should link to lower variance values. In our study, we begin by computing the variance for each sensor within its respective cluster and sum up variance of all sensors to get the variance of its respective cluster. This is represented by:

$$VAR_c^m = \frac{1}{N_c L} \sum_{i=1}^{N_c L} (x_i - \mu_c)^2, \tag{4}$$

where N_c is the number of samples in a cluster c, L is the sequence length and μ_c is the mean value of all element in the cluster. Subsequently, we determine the overall variance for each cluster as:

$$VAR_c = \sum_{m=1}^{M} VAR_c^m. \tag{5}$$

Finally, we categorize the clusters based on overall variance values. As such, the cluster with the highest variance corresponds to the terminal stage, while the cluster with the lowest variance indicates the sluggish stage.

Due to the increase in computational cost for this clustering approach, we conduct a theoretical analysis of the computational complexity. The theoretical time complexity of the K-means using soft-DTW as the metric is dominated by two key components. First, the pairwise soft-DTW computation between two M dimensional time series of length L requires $O(M \cdot L^2)$ operations, as it involves filling a $L \times L$ dynamic programming matrix with M dimensional distance calculations at each cell. During the cluster assignment step, all N samples are compared to each of the K cluster centers across I iterations. This results in a total complexity of $O(I \cdot N \cdot K \cdot M \cdot L^2)$ for distance calculations. Assuming the center update process has negligible complexity compared to distance computations, the simplified overall time complexity becomes $O(I \cdot N \cdot K \cdot M \cdot L^2)$.

F. Overall Objective

In the TACDA algorithm (Algorithm 3), the primary objective is to fine-tune the target encoder, E_T, for the RUL estimation of equipment. As indicated in Line 1, the algorithm begins with the pretraining of the source encoder E_S and the predictor R using the source domain data (X_S, y_S). After pretraining, the algorithm starts with a domain adaptation utilizing both source and target domains. Following this, data from both domains are clustered into degradation stages—the source domain based on its inherent labels (Line 3) and the target domain using a combination of K-means and variance (Line 4). A subsequent refinement occurs through a second round of domain adaptation for each degradation stage (Lines 5-6). The culmination of the process is the extraction of a trained E_T, which, when coupled with an RUL predictor R, can proficiently estimate the RUL for the target domain.

Algorithm 3: Our Proposed TACDA

Input: Source domain: $\{X_S, y_S\}$, Target domain: X_T
Output: Trained target encoder E_T

1 $E_S, R \leftarrow pretrain(X_S, y_S)$ using Eq. (1)
2 $E_T, D_T, D \leftarrow TA(X_S, X_T, E_S)$ // Algorithm 1 for 1st round DA
3 $X_{S1}, X_{S2}, X_{S3} \leftarrow$ clustering X_S by degradation stage based on y_S
4 $X_{T1}, X_{T2}, X_{T3} \leftarrow$ clustering X_T by degradation stage based on K-means and variance
5 for c *in number of stages* **do**
6 $E_T, D_T, D \leftarrow TA(X_{Sc}, X_{Tc}, E_S)$ // 2nd round DA
7 return E_T

IV. EXPERIMENTS

A. Preparation of Data

We employ the C-MAPSS benchmark dataset to evaluate the performance of our proposed method. The C-MAPSS dataset includes operational data from four different turbofan engines. As presented in Table II, each engine has a unique set of operational conditions and fault modes, consisting of 21 sensors deployed in multiple locations to monitor the health of an engine. We pre-process the datasets following the data preparation [19]. The pre-processed data has selected 14 sensors and label is remaining useful life circle of engines.

TABLE II: Details of C-MAPSS dataset.

Sub-dataset	FD001	FD002	FD003	FD004
# Engine units for training	100	260	100	249
# Engine units for testing	100	259	100	248
# Operating conditions	1	6	1	6
# Fault modes	1	1	2	2

B. Experimental Setting

All experiments run five times and the average results are shown to prevent the effect of random initialization. Additionally, we set batch size as 256, optimizer as Adam, learning rate as 5e-5 for the target encoder and the domain discriminator. Since the decoder is trained from scratch during the adaptation process, we opt a higher learning rate of 5e-3 specifically for the decoder. And we set the smoothing parameter γ for soft-DTW as 0.1. Furthermore, we built and trained our model based on Pytorch and NVIDIA GeForce RTX A4000 GPU. We adopt root mean square error (RMSE) and Score [19]. The lower the two indicators are, the better the model is.

The RMSE metric is defined as follows:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \widehat{y_i})^2}, \qquad (6)$$

where $\widehat{y_i}$ and y_i represent the estimated RUL and true RUL respectively.

The RMSE metric assigns equal importance to both early and late RUL predictions. However, in prognostics applications, late RUL predictions have more detrimental consequences for the systems. In order to address this concern, the Score metric is employed, which imposes a more severe penalty for late RUL predictions. The Score metric is expressed as follows:

$$Score_i = \begin{cases} e^{-\frac{\widehat{y_i}-y_i}{13}} - 1; \widehat{y_i} < y_i, \\ e^{\frac{\widehat{y_i}-y_i}{10}} - 1; \widehat{y_i} > y_i, \end{cases} \qquad (7)$$

$$Score = \sum_{i=1}^{N} Score_i. \qquad (8)$$

C. Comparison with State-of-the-Art Methods

We compare proposed TACDA with SOTA UDA methods, including DDC [35], Coral [30], ADARUL [13], CLUDA [36], SDAT [37], CADA [19], DARE-GRAM [38] and WIDAN [39]. For fair comparisons, we adopt the same feature extractor (5 layers BiLSTM and 32 hidden dimensions) and predictor for these methods, each of which runs five times to show average results. Further, the results of source only (Source) are also compared.

Table III shows the RMSE and Score results respectively in 12 cross-domain scenarios for RUL prediction. From the results, we observe that TACDA achieves the best averaged performance across all scenarios with regards to both RMSE and Score, whose improvement is more than 10% and 32% for RMSE and Score compared with the second best method. Specifically, TACDA outperforms all the SOTA methods in the 7 scenarios for RMSE and in the 9 scenarios for Score.

For example, TACDA outperforms the second-best method by 55% in FD003→FD001, by 39% in FD002→FD004, and by 32% in the FD003→FD002. Furthermore, TACDA attains the second-best performance in 4 scenarios with regards to RMSE and 1 scenario Score. Our proposed TACDA achieves the superior performance with the ability of preserving target domain specific information.

D. Ablation Study

1) Component Contribution Study: To validate the contribution of key components, we perform the ablation study for our proposed TACDA. We derive four variants of TACDA, namely, "Source", "w/o C", "w/o D_T" and "w/o C-D_T". Specifically, the variant "Source" denotes the unadapted version of our model. "w/o C" is the variant without using the 2nd round DA (i.e., no fine-tuning with the clustered data) for consistent degradation alignment. The variant "w/o D_T" refers to our adversarial adaptation without using the target decoder D_T for target-specific adaptation. The last variant "w/o C-D_T" uses the adversarial loss only, i.e., it has neither fine-tuning step or target decoder D_T. It is worth noting that "w/o C-D_T" is the same as ADARUL [13].

Table IV shows presents the comparative outcomes between TACDA and its variants. Our observation reveals that the "Source" exhibits the poorest performance, implying the significant disparity between the source and target domain. The proposed TACDA suppresses the "w/o C", "w/o D_T" and "w/o C-D_T" in around 35%, 39% and 72%, regarding the average improvements in terms of Score. These results demonstrate the effectiveness of the decoder, as well as the clustering and pairing strategy for the 2nd round DA for RUL prediction.

2) Reconstruction Loss Study: To verify the effectiveness of soft-DTW loss for domain information preservation, we perform comparison between MSE and soft-DTW loss in our proposed TACDA without degradation alignment. As shown in Fig. 6, soft-DTW outperforms MSE in 9 scenarios in terms of RMSE and in 10 scenarios in terms of Score. The rationale underlying this phenomenon resides in the capacity of soft-DTW to effectively capture temporal sequence patterns.

E. Soft-DTW Contribution Analysis

We conducted a sensitivity analysis of the hyperparameter λ to investigate the impact of soft-DTW on the proposed method. Several experiments were carried out using various values of λ, ranging from 0.001 to 1 with intervals of 10×. Given that the initial magnitude of the soft-DTW surpasses the adversarial loss by a factor of 1000 during the training, it is advisable to constrain the weight parameter λ to a value not exceeding 1. Fig. 7 and Fig. 8 show the performance of TACDA under different values of λ after the 1st and 2nd rounds of DA. The results indicate that the proposed method demonstrates stability across most cross-domain scenarios in terms of RMSE. However, we observed that the performance of TACDA tends to degrade when λ is either too small or too large in terms of Score. For instance, in the FD004 → FD001 scenario, TACDA exhibits the worst performance when λ is set to 1 during the 1st round of DA, and when λ is set

TABLE III: Comparison of the proposed TACDA against benchmark approaches (Up: RMSE; Down: Score). Note that F1 is short for FD001, and F1→F2 refers to the scenario where FD001 is the source domain and FD002 is the target domain. Bold indicates the best result, and underline indicates the second-best result.

Methods	F1→F2	F1→F3	F1→F4	F2→F1	F2→F3	F2→F4	F3→F1	F3→F2	F3→F4	F4→F1	F4→F2	F4→F3	Avg.
Source	20.62	54.12	36.80	15.00	34.91	37.65	38.31	46.10	31.44	37.66	32.98	19.25	33.74
Coral	20.29	38.53	34.04	13.55	32.91	33.53	34.24	36.99	30.68	38.68	31.94	16.37	30.15
DDC	20.08	39.26	34.14	13.53	33.40	33.47	30.31	30.60	24.19	37.08	31.78	16.06	28.66
ADARUL	20.26	**21.73**	31.16	13.31	28.98	33.08	24.33	24.02	21.68	26.41	25.66	15.02	23.80
CLUDA	44.36	39.17	44.71	40.07	39.17	44.71	40.09	44.45	44.72	40.14	44.38	39.19	42.10
SDAT	20.03	39.18	34.85	17.94	35.92	31.74	23.90	32.11	23.16	19.81	20.53	17.22	26.37
CADA	20.42	38.61	31.29	13.44	33.58	35.26	20.39	20.85	22.03	20.26	18.62	14.34	24.09
WIDAN	20.90	37.78	30.29	14.78	35.91	30.05	21.10	32.89	22.79	25.54	31.78	16.78	26.66
DARE-GRAM	**15.70**	36.32	**30.03**	17.44	30.46	31.47	21.23	22.19	24.56	20.95	24.50	15.97	24.24
TACDA	19.39	23.16	30.38	**13.11**	**21.86**	**26.82**	**18.90**	**19.72**	**20.46**	**19.06**	19.54	15.51	**20.66**
Source	5454	26074	20782	536	4892	19796	4991	72309	40770	99469	62301	2461	29986
Coral	3312	8887	14481	315	4762	12986	17364	105354	33777	114609	63330	1246	31702
DDC	4062	12966	16324	351	5156	14969	20088	58604	31221	90087	60334	1323	26290
ADARUL	3624	1753	11788	330	3815	16077	6250	31462	7062	14645	35174	754	11061
CLUDA	22351	5869	18270	6000	5963	18655	5797	30091	20506	6602	29771	7100	14748
SDAT	2343	5490	19432	762	10000	10587	2460	27984	4399	1241	**2793**	844	7361
CADA	4156	10540	12050	321	5484	19766	1856	16760	4134	1755	4949	913	6890
WIDAN	2787	6388	10121	353	4863	6983	5733	7591	6540	2950	7142	554	5167
DARE-GRAM	4438	8765	**8267**	740	4057	9819	2139	7053	3675	1569	2944	**566**	4503
TACDA	**2183**	**1678**	10563	**298**	**1901**	**5948**	**834**	**4770**	**2422**	**966**	4564	799	**3077**

TABLE IV: Ablation study for the proposed TACDA (Up: RMSE; Down: Score). Note that C denotes the clustering and pairing strategy for consistent degradation alignment, while D_T is the target decoder.

Methods	F1→F2	F1→F3	F1→F4	F2→F1	F2→F3	F2→F4	F3→F1	F3→F2	F3→F4	F4→F1	F4→F2	F4→F3	Avg.
Source	20.62	54.12	36.80	15.00	34.91	37.65	38.31	46.10	31.44	37.66	32.98	19.25	33.74
w/o C	**19.12**	23.40	31.07	13.75	27.47	31.00	21.21	20.93	22.05	21.48	20.11	16.16	22.31
w/o D_T	20.33	**21.03**	31.12	13.25	26.32	**26.36**	21.70	22.46	21.85	27.67	23.83	17.04	22.75
w/o C-D_T	20.26	21.73	31.16	13.31	28.98	33.08	24.33	24.02	21.68	26.41	25.66	**15.02**	23.80
TACDA	19.39	23.16	**30.38**	**13.11**	**21.86**	26.82	**18.90**	**19.72**	**20.46**	**19.06**	**19.54**	15.51	**20.66**
Source	5454	26074	20782	536	4892	19796	4991	72309	40770	99469	62301	2461	29986
w/o C	2288	1786	10767	366	2865	10558	2275	9635	6428	1952	7205	1050	4765
w/o D_T	3353	**1634**	11627	301	4808	6268	1876	9070	3824	2798	13715	1122	5033
w/o C-D_T	3624	1753	11788	330	3815	16077	6250	31462	7062	14645	35174	**754**	11061
TACDA	**2183**	1678	**10563**	**298**	**1901**	**5948**	**834**	**4770**	**2422**	**966**	**4564**	799	**3077**

to 0.001 during the 2nd round. Similarly, in the FD002 → FD003 scenario, TACDA performs poorly when λ is 0.001 in the 1st round, and when λ is 1 in the 2nd round. To achieve satisfactory performance, we recommend setting the value of λ as 0.1. The potential reason for the consistency observed in RMSE while encountering variability in the Score metric could be attributed to the temporal influence exerted on the Score metric during the utilization of soft-DTW temporal alignment.

To highlight the role of soft-DTW as a reconstruction loss in preserving target-specific information, we analyzed its progression over training epochs during both the 1st and 2nd rounds of DA. As illustrated in Fig. 9, the soft-DTW loss exhibits a sharp decline in the initial epochs of the 1st round of DA, followed by a gradual stabilization as training converges. Since soft-DTW serves as a reconstruction loss, this behavior underscores the effectiveness of the proposed approach in maintaining target-specific information while facilitating cross-domain alignment.

In the 2nd round of DA, during the initial stages, the soft-DTW loss values in the sluggish stage and accelerated stage are higher than those in the moderate stage, indicating that the information from the sluggish and accelerated stages was not

well preserved during the 1st round of DA. Through the 2nd round of DA, the soft-DTW loss continues to decrease across data from various degradation stages, further demonstrating its ability to capture and preserve distinct target-specific characteristics at different stages.

F. Insufficient Degradation Data Analysis

In practical real-world applications, acquiring data in the failure condition from target domain for training purposes poses challenges, which is prone to negative adaptation if match the whole source domain to the target domain. To examine the impact of insufficient degradation data for our proposed TACDA, we conducted an analysis. Specifically, we performed experiments by excluding data with accelerated degradation stage, from the dataset during training for domain adaptation. The results, depicted in Fig. 10, highlight the stability of the proposed method across various cross-domain scenarios. The rationale for this lies in the fact that degradation alignment exclusively aligns the distributions of source and target data in the shared degradation stage, thereby mitigating the influence stemming from inadequate degradation data
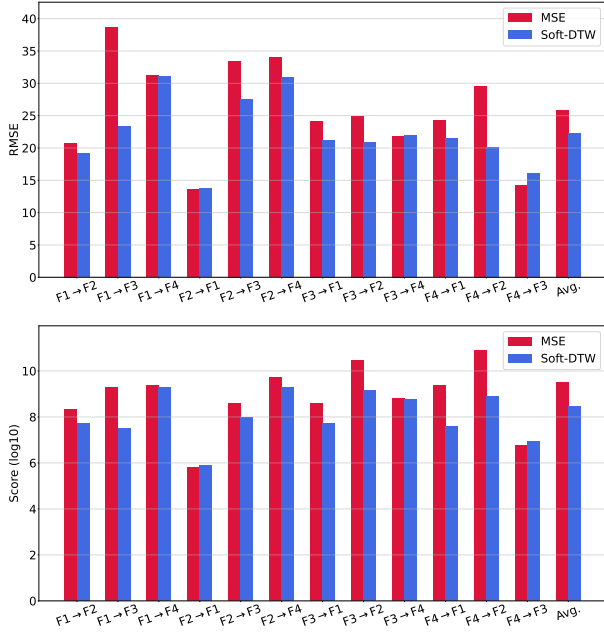
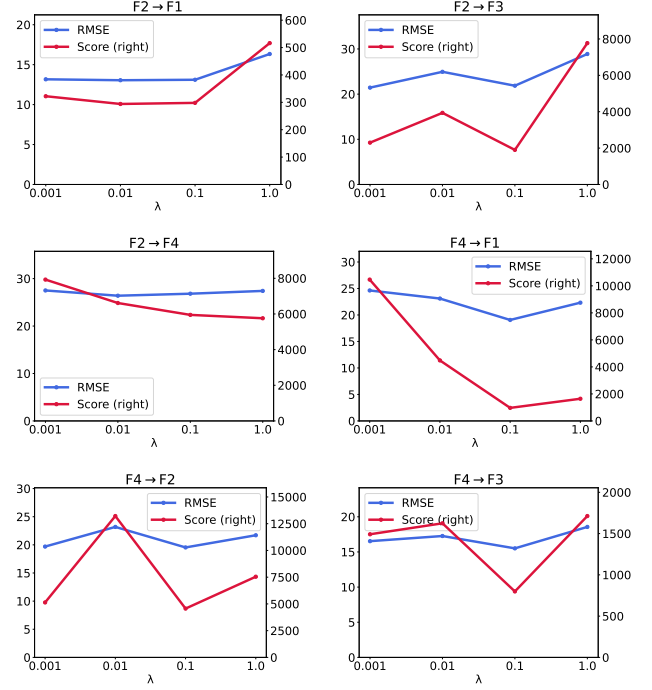Fig. 6: Comparison of using MSE and soft-DTW as reconstruction loss.

Fig. 8: The sensitivity analysis for different λ in the 2nd round DA.
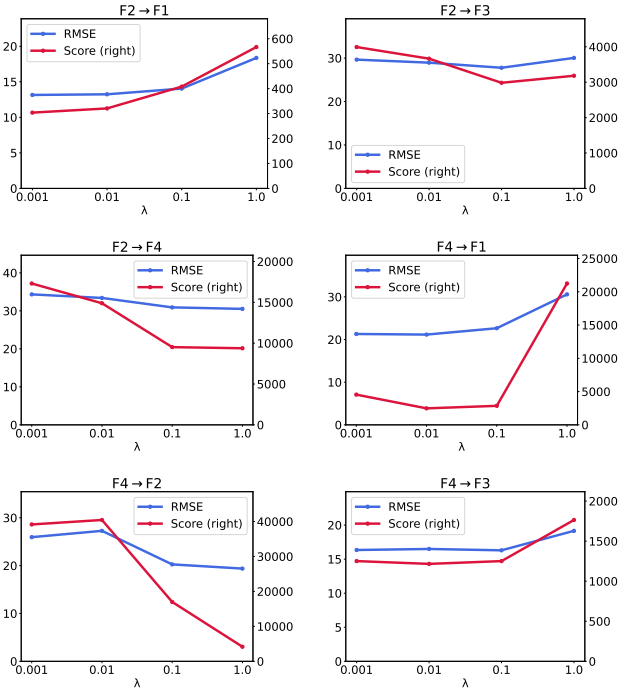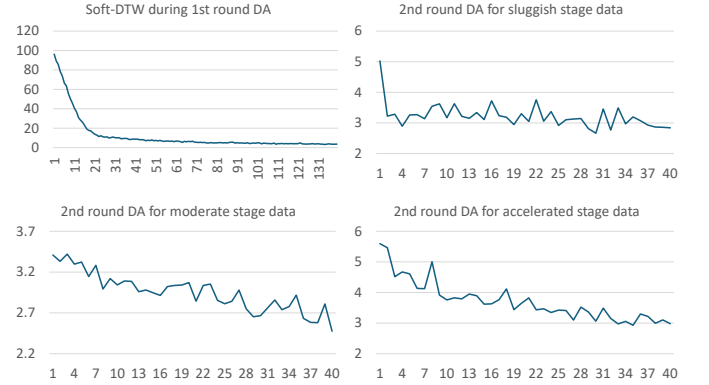

Fig. 9: The changes of soft-DTW loss in the 1st and 2nd round DA.

within the target domain, such as poor alignment due to align data of accelerated stage to the sluggish stage.

G. Analysis of Time Utilization and Resource Consumption

We conducted additional experiments to analyze the temporal efficiency and resource consumption of our proposed method under four scenarios, and compared with the second-best method, i.e., DARE-DRAM. To ensure fair comparison, both TACDA and DARE-DRAM are trained for 180 epochs. As demonstrated in Table IV, our method trains faster in two scenarios and is marginally slower in the other two. And TACDA requires a 10% more GPU memory over DARE-DRAM. This increased memory usage is primarily due to the additional parameters and computations introduced by the decoder and discriminator, which allows the model to

Fig. 7: The sensitivity analysis for different λ in the 1st round DA.

REFERENCES

[1] M. Kordestani, M. E. Orchard, K. Khorasan, and M. Saif, "An overview of the state-of-the-art in aircraft prognostic and health management strategies," *IEEE Transactions on Instrumentation and Measurement*, 2023.

[2] D. Wang, Z. Peng, and L. Xi, "Theoretical and experimental investigations on spectral lp/lq norm ratio and spectral gini index for rotating machine health monitoring," *IEEE Transactions on Automation Science and Engineering*, vol. 18, no. 3, pp. 1074–1086, 2020.

[3] K. Liu, N. Z. Gebraeel, and J. Shi, "A data-level fusion model for developing composite health indices for degradation modeling and prognostic analysis," *IEEE Transactions on Automation Science and Engineering*, vol. 10, no. 3, pp. 652–664, 2013.

[4] Y. Wang, M. Wu, R. Jin, X. Li, L. Xie, and Z. Chen, "Local–global correlation fusion-based graph neural network for remaining useful life prediction," *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

[5] S. Behera, R. Misra, and A. Sillitti, "Gan-based multi-task learning approach for prognostics and health management of iiot," *IEEE Transactions on Automation Science and Engineering*, 2023.

[6] D. Wang, X. Xian, and C. Song, "Joint learning of failure mode recognition and prognostics for degradation processes," *IEEE Transactions on Automation Science and Engineering*, vol. 21, no. 2, pp. 1421–1433, 2023.

[7] I. Nejjar, F. Geissmann, M. Zhao, C. Taal, and O. Fink, "Domain adaptation via alignment of operation profile for remaining useful lifetime prediction," *Reliability Engineering & System Safety*, vol. 242, p. 109718, 2024.

[8] Y. Chang, H. Fang, and Y. Zhang, "A new hybrid method for the prediction of the remaining useful life of a lithium-ion battery," *Applied energy*, vol. 206, pp. 1564–1578, 2017.

[9] Y. Chang and H. Fang, "A hybrid prognostic method for system degradation based on particle filter and relevance vector machine," *Reliability Engineering & System Safety*, vol. 186, pp. 51–63, 2019.

[10] P. R. d. O. da Costa, A. Akçay, Y. Zhang, and U. Kaymak, "Remaining useful lifetime prediction via deep domain adaptation," *Reliability Engineering & System Safety*, vol. 195, p. 106682, 2020.

[11] Y. Lei, N. Li, L. Guo, N. Li, T. Yan, and J. Lin, "Machinery health prognostics: A systematic review from data acquisition to rul prediction," *Mechanical systems and signal processing*, vol. 104, pp. 799–834, 2018.

[12] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[13] M. Ragab, Z. Chen, M. Wu, C. K. Kwoh, and X. Li, "Adversarial transfer learning for machine remaining useful life prediction," in *2020 IEEE international conference on prognostics and health management (ICPHM)*. IEEE, 2020, pp. 1–7.

[14] Y. Jiang, T. Xia, D. Wang, X. Fang, and L. Xi, "Adversarial regressive domain adaptation approach for infrared thermography-based unsupervised remaining useful life prediction," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 10, pp. 7219–7229, 2022.

[15] X. Jia, M. Zhao, Y. Di, Q. Yang, and J. Lee, "Assessment of data suitability for machine prognosis using maximum mean discrepancy," *IEEE transactions on industrial electronics*, vol. 65, no. 7, pp. 5872–5881, 2017.

[16] H. Cheng, X. Kong, G. Chen, Q. Wang, and R. Wang, "Transferable convolutional neural network based remaining useful life prediction of bearing under multiple failure behaviors," *Measurement*, vol. 168, p. 108286, 2021.

[17] W. Mao, J. He, and M. J. Zuo, "Predicting remaining useful life of rolling bearings based on deep feature representation and transfer learning," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 4, pp. 1594–1608, 2019.

[18] P. Stojanov, Z. Li, M. Gong, R. Cai, J. Carbonell, and K. Zhang, "Domain adaptation with invariant representation learning: What transformations to learn?" *Advances in Neural Information Processing Systems*, vol. 34, pp. 24 791–24 803, 2021.

[19] M. Ragab, Z. Chen, M. Wu, C. S. Foo, C. K. Kwoh, R. Yan, and X. Li, "Contrastive adversarial domain adaptation for machine remaining useful life prediction," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 8, pp. 5239–5249, 2020.

[20] J. Liu, F. Lei, C. Pan, D. Hu, and H. Zuo, "Prediction of remaining useful life of multi-stage aero-engine based on clustering and lstm fusion," *Reliability Engineering & System Safety*, vol. 214, p. 107807, 2021.
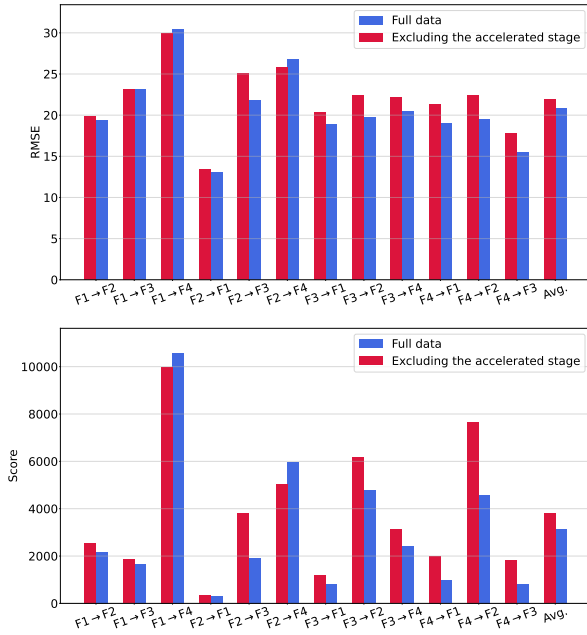

Fig. 10: The insufficient degradation data analysis for excluding the accelerated degradation stage from the training dataset.

preserve target-specific information during adaption. These results demonstrate that TACDA delivers superior performance without significantly higher computational costs, maintaining practical deployability while advancing state-of-the-art results.

TABLE IV: Comparison of Time Utilization and Resource Consumption

Method		TACDA	DARE-GRAM
Time (second)	F1→F2	979.4	784.8
	F1→F3	571.8	482.9
	F2→F3	575.0	1011.6
	F2→F4	1127.5	1141.2
GPU Memory (MiB)		1792	1629

V. CONCLUSION

In this paper, we introduced a novel approach called TACDA for the automatic discovery of domain-invariant features, while simultaneously preserving target-specific information and aligning the data within the same degradation stage in RUL prediction. The TACDA method was developed based on an adversarial auto-encoder domain adaptation framework, incorporating a soft-DTW loss and a data selection mechanism utilizing soft-DTW based K-means clustering and variance. To evaluate the effectiveness of the proposed TACDA method, extensive experiments were conducted, including a comprehensive comparison with various state-of-the-art domain adaptation techniques. The experimental results demonstrated that the TACDA method outperforms all other state-of-the-art methods significantly. Additionally, an ablation study was conducted to demonstrate the effectiveness of the key components in the domain adaptation process.

[21] S. Li, S. Song, G. Huang, Z. Ding, and C. Wu, "Domain invariant and class discriminative feature learning for visual domain adaptation," *IEEE transactions on image processing*, vol. 27, no. 9, pp. 4260–4273, 2018.

[22] C. Chen, W. Xie, W. Huang, Y. Rong, X. Ding, Y. Huang, T. Xu, and J. Huang, "Progressive feature alignment for unsupervised domain adaptation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 627–636.

[23] J. Zhu, N. Chen, and W. Peng, "Estimation of bearing remaining useful life based on multiscale convolutional neural network," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 4, pp. 3208–3216, 2018.

[24] R. Liu, B. Yang, and A. G. Hauptmann, "Simultaneous bearing fault recognition and remaining useful life prediction using joint-loss convolutional neural network," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 1, pp. 87–96, 2019.

[25] J. Deutsch and D. He, "Using deep learning-based approach to predict remaining useful life of rotating components," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 48, no. 1, pp. 11–20, 2017.

[26] C.-G. Huang, H.-Z. Huang, and Y.-F. Li, "A bidirectional lstm prognostics method under multiple operational conditions," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 11, pp. 8792–8802, 2019.

[27] Z. Chen, M. Wu, R. Zhao, F. Guretno, R. Yan, and X. Li, "Machine remaining useful life prediction via an attention-based deep learning approach," *IEEE Transactions on Industrial Electronics*, vol. 68, no. 3, pp. 2521–2531, 2020.

[28] Q. Liu and H. Xue, "Adversarial spectral kernel matching for unsupervised time series domain adaptation." in *IJCAI*, 2021, pp. 2744–2750.

[29] J. Yan, Z.-S. Ye, S. He, and Z. He, "A feature disentanglement and unsupervised domain adaptation of remaining useful life prediction for sensor-equipped machines," *Reliability Engineering & System Safety*, vol. 242, p. 109736, 2024.

[30] B. Sun, J. Feng, and K. Saenko, "Correlation alignment for unsupervised domain adaptation," *Domain adaptation in computer vision applications*, pp. 153–171, 2017.

[31] Y. Deng, S. Du, D. Wang, Y. Shao, and D. Huang, "A calibration-based hybrid transfer learning framework for rul prediction of rolling bearing across different machines," *IEEE Transactions on Instrumentation and Measurement*, vol. 72, pp. 1–15, 2023.

[32] X. Li, J. Li, L. Zuo, L. Zhu, and H. T. Shen, "Domain adaptive remaining useful life prediction with transformer," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–13, 2022.

[33] J. Chen, D. Li, R. Huang, Z. Chen, and W. Li, "Aero-engine remaining useful life prediction method with self-adaptive multimodal data fusion and cluster-ensemble transfer regression," *Reliability Engineering & System Safety*, vol. 234, p. 109151, 2023.

[34] M. Cuturi and M. Blondel, "Soft-dtw: a differentiable loss function for time-series," in *International conference on machine learning*. PMLR, 2017, pp. 894–903.

[35] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell, "Deep domain confusion: Maximizing for domain invariance," *ArXiv*, vol. abs/1412.3474, 2014.

[36] M. Vayyat, J. Kasi, A. Bhattacharya, S. Ahmed, and R. Tallamraju, "Cluda: Contrastive learning in unsupervised domain adaptation for semantic segmentation," *arXiv preprint arXiv:2208.14227*, 2022.

[37] H. Rangwani, S. K. Aithal, M. Mishra, A. Jain, and V. B. Radhakrishnan, "A closer look at smoothness in domain adversarial training," in *International Conference on Machine Learning*. PMLR, 2022, pp. 18 378–18 399.

[38] I. Nejjar, Q. Wang, and O. Fink, "Dare-gram: Unsupervised domain adaptation regression by aligning inverse gram matrices," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 11 744–11 754.

[39] C. He, H. Shi, X. Liu, and J. Li, "Interpretable physics-informed domain adaptation paradigm for cross-machine transfer diagnosis," *Knowledge-Based Systems*, vol. 288, p. 111499, 2024.