Remaining Useful Life Prediction across Datasets via Semi-Supervised Domain Adaptation

Gengyu Li, Takehisa Yairi

Department of Advanced Interdisciplinary Studies, The University of Tokyo, Tokyo, Japan
{li-gengyu, yairi} @g.ecc.u-tokyo.ac.jp

ABSTRACT

Remaining useful life (RUL) prediction has become a critical task, as accurate prediction enables effective maintenance planning and minimizes unplanned downtime. Complex systems frequently operate under diverse operating conditions, with distributional differences and limited RUL labels. To address these issues, domain adaptation (DA) methods have attracted increasing attention. Prevalent RUL prediction DA methods mainly adopt an unsupervised setting (UDA), relying solely on unlabeled target domain data. However, prediction performances of UDA methods are far inferior to Target-Only (TO) models trained on full labeled target data, and it fails to effectively transfer degradation-related knowledge for RUL prediction when source-target distribution differences are large. Additionally, existing research mainly focuses on in-dataset adaptation across operating conditions, lacking systematic exploration of cross-dataset RUL prediction. Therefore, this work proposes a semi-supervised domain adaptation (SSDA) workflow for RUL prediction between different datasets. We design an alignment pipeline to address feature discrepancies between datasets and sampling frequency differences. During SSDA training, the original RUL labels of the source domain are treated as noisy labels with respect to the target domain and are progressively modified throughout the adaptation process. We conduct two case studies, one on turbofan engines and the other on rolling bearings. The experimental results demonstrate that the proposed approach can effectively perform cross-dataset RUL prediction.

1. INTRODUCTION

The reliability of industrial systems, especially rotating machinery, has become important for operational safety and ensuring sustainable progress. For maintenance approaches, four general categories are commonly classified: corrective maintenance, preventive maintenance (PvM), condition-based maintenance (CBM), and predictive maintenance (PdM) (Çınar et al., 2020). Corrective maintenance is carried out when a system is failed and is aware of all its predefined sets of failures and damages (Merkt, 2019). PvM is scheduled at predefined intervals to prevent equipment failures before occurrence, whereas CBM is based on constant system monitoring when maintenance service is actually needed. PdM is actually a subclass of CBM, which uses condition monitoring (CM) data to forecast and evaluate the degradation of the items of interest or the entire system. Compared with other strategies, PdM enables timely interventions, reduces unnecessary downtime and maintenance costs. Furthermore, with the decreasing cost of CM sensors, seamless connectivity and real-time access to both streaming and historical heterogeneous data have become increasingly feasible for PdM, making it increasingly indispensable in modern industrial applications (Zonta et al., 2020).

Remaining useful life (RUL) prediction is considered as the key problem in PdM and has been widely studied (Gbore, Shafiq, Jain, & McGlinchey, 2025). Accurate RUL prediction helps assess health status of the system and schedule future maintenance. The prediction methods are mainly model-based (Gebraeel et al., 2023) and data-driven (Tsui, Chen, Zhou, Hai, & Wang, 2015). Model-based methods define mathematical models of system dynamics to detect and predict faults by analyzing the residuals between model outputs and sensor measurements (Lei et al., 2016). Although offering strong interpretability, model-based methods are limited by their dependence on expert knowledge and cannot make inferences beyond the scope of the models tested. Data-driven methods learn mappings from sensor measurements to outputs, with little or no reliance on the underlying physical principles or expert knowledge (L. Liu, Song, & Zhou, 2022). In recent years, the rise of artificial intelligence (AI) technologies, especially deep learning, has propelled data-driven methods to the mainstream, and they have been widely applied in prognostics and health management (PHM) studies.

However, data-driven methods still suffer from several limitations. First, data-driven approaches are highly dependent on data quality and quantity. This challenge is particularly

https://doi.org/10.36001/IJPHM.2026.v17i2.4804

evident in RUL prediction, where the problem is usually formulated as supervised learning. Representative degradation trajectories are often difficult to collect in large numbers, and labeling RUL is costly. Moreover, for systems such as turbofan engines, publicly available datasets remain very limited (Ferreira & Gonçalves, 2022). Given the scarcity of publicly available datasets, methods developed and validated on such limited datasets should be applied to practices with caution (Chi, Dong, Wang, Yu, & Leung, 2022). Second, industrial systems are often operated under various conditions and cycles, leading to significant distributional differences in sensor measurements. Moreover, when the system encounters different failure modes, trends in sensor readings may exhibit significant variations (Luo et al., 2020; Cheng, Wang, Wu, Zhu, & Lee, 2022). Learning models on data with large distribution differences or applying them to unknown failure modes often results in degraded performance.

Motivated by these challenges, domain adaptation (DA) has attracted increasing research attention (da Costa, Akçay, Zhang, & Kaymak, 2020). DA is a transfer learning (TL) method that aims to mitigate the adverse effects of domain shift. In this way, models trained on a source domain can be effectively transferred to another domain of interest. Depending on the availability of target domain labels, DA can generally be categorized into supervised domain adaptation (SDA), unsupervised domain adaptation (UDA), or semi-supervised domain adaptation (SSDA) as shown in Fig. 1. SDA is overly idealistic for industrial environments and is therefore seldom considered. Generally, the problem is formulated as UDA, assuming that labels of the target domain are not accessible (X. Li, Zhang, Li, & Hao, 2023; Huang, Zhu, Han, & Peng, 2021). UDA methods have demonstrated clear improvements in testing performance on the target domain. However, a considerable performance gap still remains between UDA and supervised training in the target domain (Target-Only, TO), which assumes that a sufficient amount of target domain label is available. This gap highlights the inherent difficulty of learning transferable representations without access to target labels and indicates that UDA has not yet achieved a sufficiently high performance. Meanwhile, SSDA allows access to part of the target domain labels, and existing studies on semi-supervised TL have demonstrated its potential for more practical problem formulation (Pan, Wang, Wen, & Li, 2023). However, SSDA has not yet been extensively studied in the PHM field. (G. Li & Yairi, 2025) provided an initial exploration of SSDA for RUL prediction under limited labeled data scenarios. Nevertheless, most existing DA studies in PHM have focused on adaptation within the same dataset or under relatively constrained settings. In contrast, detailed investigations into adaptation across entirely different datasets remain scarce.

To address the above research gaps, this work proposes a practical and extensible framework for SSDA in cross-dataset

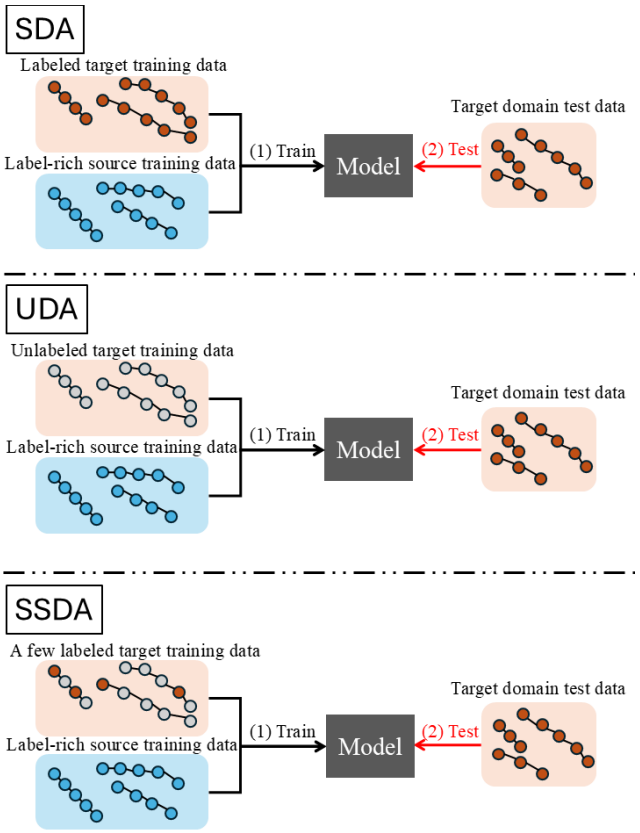

Figure 1. Problem settings of domain adaptation.

RUL prediction. We consider differences at both the sensor level and the sampling frequency level, and after aligning these discrepancies, we propose an SSDA approach to perform cross-dataset adaptation. Compared to previous work (G. Li & Yairi, 2025), this study introduces several substantial improvements. First, we provide a more rigorous problem formulation with an explicit mathematical definition of heterogeneous input spaces between different systems. Second, we systematically investigate the effect of the dynamic α scheduling strategy through ablation studies and comparisons with fixed-weight alternatives. Third, we extend the experimental evaluation by including quantitative comparisons with representative DA baselines under both UDA and SSDA settings. The contributions of this paper are listed as follows:

1. We address the challenging problem of cross-dataset RUL prediction under heterogeneous sensor configurations and sampling frequencies by developing a unified preprocessing and alignment framework.
2. We propose a novel and practical SSDA framework for cross-dataset RUL prediction, in which source domain RUL labels are progressively modified toward target domain predictions during the adaptation process.
3. We conduct experimental evaluations on multiple PHM benchmark datasets and to the best of our knowledge,

this work is the first to demonstrate the effectiveness of cross-domain RUL prediction between turbofan engine datasets, which are CMAPSS and N-CMAPSS (Saxena, Goebel, Simon, & Eklund, 2008; Arias Chao, Kulkarni, Goebel, & Fink, 2021).

The remainder of this paper is structured as follows. Section 2 provides a literature review on RUL prediction and DA. Section 3 explains the problem formulation of SSDA and the proposed methodology. Section 4 conducts two comprehensive case studies, section 5 is the conclusion and section 6 discusses the limitation and future work.

2. LITERATURE REVIEW

2.1. Domain Adaptation

As established previously, the DA categories include supervised (Motiian, Piccirilli, Adjeroh, & Doretto, 2017), unsupervised (Tzeng, Hoffman, Saenko, & Darrell, 2017), and semi-supervised (Saito, Kim, Sclaroff, Darrell, & Saenko, 2019). SDA relies on a strong assumption that sufficient labeled data are available in both the source and target domains. UDA is more attractive, as it does not require target data to be labeled. UDA can be classified into discrepancy-based (Long, Cao, Wang, & Jordan, 2015), reconstruction-based (Glorot, Bordes, & Bengio, 2011), and adversarial-based (Pei, Cao, Long, & Wang, 2018). Discrepancy-based methods aim to align the source and target domains by explicitly minimizing the distributional difference between their feature representations, such as approaches based on maximum mean discrepancy (MMD) (Tzeng, Hoffman, Zhang, Saenko, & Darrell, 2014). Reconstruction-based methods attempt to reconstruct the original input from both source and target domains, thereby encouraging the extraction of more common feature representations (Ghifary, Kleijn, Zhang, Balduzzi, & Li, 2016). Adversarial-based methods use the idea of adversarial learning to encourage feature representations that are indistinguishable between the source and target domains. (Ganin & Lempitsky, 2015) proposed a domain-adversarial neural network (DANN), which achieves UDA through a gradient reversal layer (GRL).

SSDA makes a less stringent assumption than SDA and a more relaxed one than UDA, thus considered as a more practical problem setting. (Yao, Pan, Ngo, Li, & Mei, 2015) proposed an SSDA framework based on subspace learning. Experiments showed that this method outperformed several state-of-the-art baselines. (Y.-C. Yu & Lin, 2023) proposed a source label adaptation framework for SSDA, which formulates DA as a noisy label learning (NLL) problem. These works suggest that, in SSDA, effectively leveraging the few labeled target data while simultaneously exploiting the unlabeled target data is crucial for improving adaptation performance.

2.2. RUL Prediction by Domain Adaptation

Recently, DA has been widely employed in RUL prediction to improve model performance and generalizability. Current approaches are mostly under UDA, including discrepancy-based methods (Zhuang, Jia, Ding, & Ding, 2021; Lu et al., 2024) and adversarial-based methods (Fu, Zhang, Lin, Zhao, & Zhong, 2021; Nejjar, Geissmann, Zhao, Taal, & Fink, 2024). (Zhuang et al., 2021) proposed a cross-domain RUL prediction method that combines a temporal convolutional network with residual self-attention for feature extraction, using MK-MMD and contrastive loss to align feature distributions. After the proposal of (Ganin & Lempitsky, 2015), DANN structure has become one of the most widely used architectures in the context of PHM. (C. Liu & Gryllias, 2020) proposed a bidirectional LSTM (Bi-LSTM) model trained in DANN to estimate the RUL of bearings. To better use extra physical information carried by different operation profiles, (Nejjar et al., 2024) introduced multiple discriminators into the original DANN to learn domain-invariant features. However, solely aligning target features with the source without constraints may have a limited impact within similar systems but can result in poor performance when applied across different systems.

So far, cross-system DA for RUL prediction has been mainly discussed in the context of rolling bearings (Lu et al., 2024; Zhao & Liu, 2022). Although UDA can be applied to bearing datasets, where the feature dimensionality and domain shift are relatively small, cross-system UDA still faces challenges when extended to high-dimensional datasets with substantial distribution discrepancies. In addition, SSDA has received limited attention in PHM applications. For example, (Berghout, Mouss, Bentrcia, & Benbouzid, 2021) proposed a semi-supervised deep transfer learning approach for bearing RUL prediction. However, the approach relies on manually designed components, such as predefined degradation functions and clustering-based stage partitioning. And the transfer is performed within a relatively homogeneous setting, which limits its applicability to cross-dataset scenarios with substantial domain gaps. Previous work (G. Li & Yairi, 2025) presented an initial attempt to apply SSDA to RUL prediction under limited labeled data conditions. However, the adaptation process in that study relied on manually specified hyperparameters and the training strategy lacked sufficient adaptivity for stable cross-domain transfer. Furthermore, the experimental validation was limited to a narrower problem setting and did not consider cross-dataset adaptation between substantially different benchmark datasets.

Motivated by all above limitations, this work proposes a more comprehensive SSDA framework for cross-system RUL prediction, with systematic consideration of cross-dataset discrepancies and validation on both turbofan engine and rolling bearing datasets. Unlike traditional pseudo-labeling and self-

training methods, which only generate pseudo-labels for unlabeled target data, the proposed approach corrects source domain labels simultaneously through a progressive refinement mechanism. This enables the framework to progressively adapt source domain supervision toward the heterogeneous target domain under cross-dataset settings.

3. PROBLEM FORMULATION AND METHODOLOGY

3.1. Problem Formulation

We first describe the data format and problem formulation of semi-supervised domain adaptation (SSDA) for RUL prediction. For a multivariate time-series sample collected from on-board sensors, the input of the i-th sample is denoted as $\mathbf{x}_i \in \mathbb{R}^{K\times T}$, where K denotes the number of sensor channels and T denotes the number of time steps. The corresponding remaining useful life (RUL) label is a scalar value denoted as $y_i \in \mathbb{R}$. We consider a fully labeled source domain D_S consisting of N_S samples, $D_S = \{(\mathbf{x}_i^s, y_i^s) \mid i = 1, \ldots, N_S\}$. For the target domain, only a limited number of labeled samples are available, $D_L = \{(\mathbf{x}_i^l, y_i^l) \mid i = 1, \ldots, N_L\}$, while the remaining target samples are unlabeled and denoted as $D_U = \{\mathbf{x}_i^u \mid i = 1, \ldots, N_U\}$.

In this study, the source and target domain datasets are assumed to comprise historical run-to-failure trajectories, which are used for model development and adaptation. The trained model is ultimately intended to be deployed on new in-service target domain units whose failure events have not yet occurred and whose RUL labels are therefore unavailable. Although RUL labels can be retrospectively derived once a run-to-failure trajectory is completed, establishing a reliable RUL ground-truth is not always a cost-free or fully automated process in practice. It typically requires additional engineering effort, such as post-failure inspection, confirmation of the exact failure mode and timing, and cross-referencing of maintenance or overhaul records, which may not be uniformly carried out across all historical units. Therefore, the labeled subset D_L represents historical target domain trajectories with reliable RUL labels, whereas D_U represents additional target domain trajectories treated as unlabeled data. In the benchmark experiments, this partially labeled setting is simulated by randomly selecting a proportion of the target domain data as labeled.

Since the source and target domains originate from different systems, their data distributions are generally different, i.e., $p_S(\mathbf{x}, y) \neq p_T(\mathbf{x}, y)$. Furthermore, in cross-system scenarios, input representations may differ significantly, as both the number of sensor channels K and the sampling rate (or temporal resolution) are system-dependent. This leads to heterogeneous input spaces across domains, which prevents direct feature alignment and makes knowledge transfer more difficult. As a result, existing DA RUL prediction methods often exhibit limited effectiveness in these scenarios. The objective is to learn a predictive function $f : \mathbb{R}^{K\times T} \rightarrow \mathbb{R}$ that can accurately estimate the RUL for target domain samples, leveraging D_S, D_L, and D_U. Therefore, the problem considered in this work is to perform SSDA for RUL prediction under cross-system settings with heterogeneous input spaces.

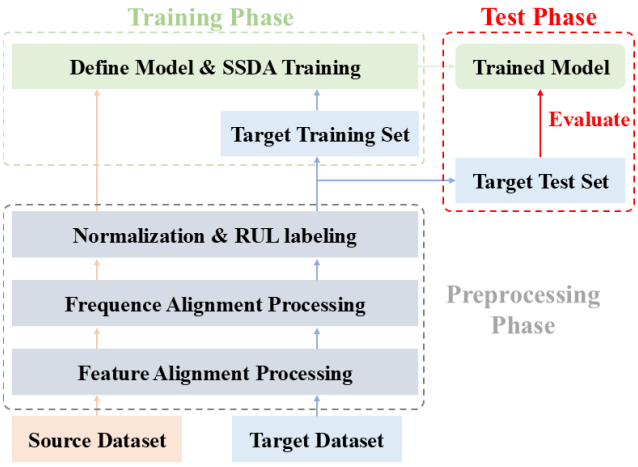

Figure 2. Proposed workflow in this work.

3.2. Methodology

3.2.1. Feature Alignment

Fig. 2 shows the overall workflow of the proposed SSDA-based RUL prediction framework. As a first step, we perform input feature-level alignment as illustrated in Fig. 3. In cross-dataset scenarios, although data are collected from similar systems, sensor configurations and recorded variables may differ. However, sensors measuring the same physical quantities are expected to reflect comparable degradation processes. Therefore, the model may fail to capture transferable patterns using heterogeneous feature sets across domains. To address this issue, we identify the set of commonly available sensors across datasets and retain only these shared features for subsequent modeling. Formally, let $\mathcal{F}_S$ and $\mathcal{F}_T$ denote the sensor sets of the source and target domains, respectively. We define the aligned feature set as $\mathcal{F}_{\text{align}} = \mathcal{F}_S \cap \mathcal{F}_T$, and restrict the input representation to this shared subset. This step ensures structural consistency between domains and serves as a prerequisite for subsequent steps.

3.2.2. Frequency Alignment

After feature alignment, we further address the mismatch in sampling frequencies between datasets. Sampling frequency is one of the most critical parameters in time-series data, as it often reflects the focus of the data collection process. In practice, datasets collected from different systems are often recorded at different sampling rates. This frequency mismatch can lead to difficulties in learning consistent tempo-

Figure 3. Feature alignment based on shared sensor intersection across different datasets.

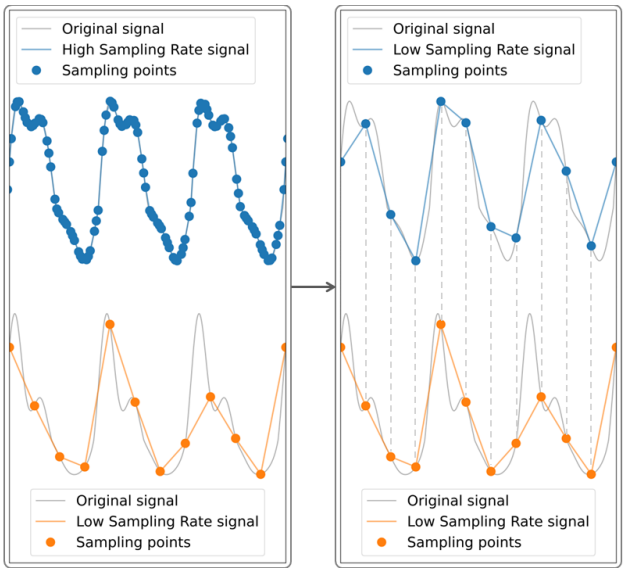

Figure 4. Frequency alignment by downsampling to a common temporal resolution.

ral patterns across datasets. Therefore, we perform frequency alignment so that the samples from different datasets exhibit comparable temporal trends. Denote Δt_S and Δt_T as the sampling intervals of the source and target domains, respectively. We define the aligned sampling interval as $\Delta t_{\text{align}} = \max(\Delta t_S, \Delta t_T)$, i.e., the coarser temporal resolution. We then adopt a downsampling strategy to the dataset with higher temporal resolution to match this common scale as shown in Fig. 4. Although this inevitably discards some high-frequency information, it offers several practical advantages. First, it ensures consistency in the effective temporal scale of the input sequences, facilitating stable model training. Second, it improves interpretability by focusing on degradation trends that are observable across systems rather than dataset-specific high-frequency fluctuations. Finally, downsampling reduces computational requirements, improving efficiency and enabling practical deployment.

3.2.3. SSDA Approach

The third step aims to achieve adaptation under cross-dataset conditions. As mentioned previously, RUL prediction across datasets is currently only demonstrated for bearings and is based on the UDA problem setting. This is mainly because, firstly, popular bearing datasets feature consistent sampling frequencies and sensor types, leading to a straightforward data preprocessing procedure. More significantly, the second reason lies in the relatively uniform distribution of bearing sensor data. The data from the acceleration sensor for healthy bearings is typically zero-centered with values that fluctuate symmetrically. In previous studies of UDA (Fu et al., 2021; Nejjar et al., 2024), the statistics of the source domain are used to normalize the target domain. However, this is not applicable in complex systems such as turbofan engines. In complex systems, healthy data distributions are typically non-zero-centered and have different means and variances. Directly applying source domain statistics for target domain normalization can amplify distributional discrepancies, thereby hindering DA approaches and even degrading prediction performance.

In SSDA, we consider three types of data: D_S, D_L, and D_U, with D_L being the most important. As stated previously, using source domain statistics to normalize D_L and D_U is inappropriate due to significant distribution discrepancies. Therefore, domain-specific normalization is adopted. Since the model is expected to learn the target domain mapping represented by D_L, i.e., $f_\theta(\mathbf{x}_i^l) \approx y_i^l$, the source domain supervision becomes biased after separate normalization. Therefore, source labels y_i^s can be regarded as noisy or biased with respect to target domain semantics.

To address this issue, we reinterpret the SSDA process as a progressive label refinement problem, where source domain labels are gradually adapted toward the target domain distribution during training. The proposed approach explicitly addresses cross-dataset discrepancies and introduces a progressive label correction mechanism to adapt source domain supervision toward target domain semantics. Specifically, we introduce a dynamic adaptation mechanism to modify source labels as follows:

$$\tilde{y}_i^s = (1-\alpha)\cdot y_i^s + \alpha \cdot \hat{y}_i^s \tag{1}$$

where $\hat{y}_i^s = f_\theta(\mathbf{x}_i^s)$ denotes the model predictions for the source samples, and $\tilde{y}_i^s$ is the corrected label. Meanwhile,

for the unlabeled target domain, pseudo-labels are generated as $\tilde{y}_i^u = f_\theta(\mathbf{x}_i^u)$. Learning from pseudo-labeled samples has been shown to improve target domain representation learning and facilitate cross-domain alignment, thereby reducing domain discrepancy (Zou, Yu, Liu, Kumar, & Wang, 2019; Y.-C. Yu & Lin, 2023). Consequently, the model progressively approximates the target domain conditional mapping, i.e., $f_\theta(\mathbf{x}) \approx \mathbb{E}[y \mid \mathbf{x}]$, where the expectation is implicitly defined by the data distribution of the target domain. For the dynamically adaptive coefficient α, we adopt the same setting as in (Ganin & Lempitsky, 2015):

$$\alpha = \frac{2}{1 + \exp(-\phi \cdot p)} - 1 \tag{2}$$

$$\gamma = \frac{\gamma_0}{(1 + \phi \cdot p)^\eta} \tag{3}$$

where p represents the normalized training progress, ϕ is a scaling parameter (typically set to 10), γ_0 is the initial learning rate, and γ denotes the learning rate at the current training step. As the training progresses (i.e., as p increases), γ gradually decays according to Eq. (3). This sigmoid-based scheduling is preferred to linear or step-function alternatives because it provides a smooth transition that avoids abrupt changes in the training objective. Fig. 5 illustrates an example of the dynamic scheduling of α over 50 training epochs. During the early training phases, α remains small, such that the corrected source labels are still dominated by the original source labels y_i^s. As the training process progresses and the model becomes more stable in D_L, the value of α gradually increases. This allows the model to gradually rely more on prediction-based labels $\hat{y}_i^s$, and better align the source supervision with characteristics of the target domain. Note that no explicit confidence thresholding is applied to the pseudo-labels, as the dynamic mechanism α serves as an implicit regularization.

It should also be noted that the proposed dynamic scheduling of the parameter α differs fundamentally from the gradient reversal layer (GRL) scheduling coefficient in (Ganin & Lempitsky, 2015). In DANN, the scheduling coefficient controls the strength of feature-level adversarial domain alignment. By contrast, the proposed α controls the interpolation between the original source labels y_i^s and the prediction-based labels $\hat{y}_i^s$, thereby progressively adapting the source domain supervision itself rather than enforcing feature-level domain invariance.

In each batch, three training losses are computed from the source, labeled target, and unlabeled target data: $\mathcal{L}_S$, $\mathcal{L}_L$, and $\mathcal{L}_U$, respectively. The overall objective is defined as:

$$\mathcal{L}_{\text{SSDA}} = \alpha\mathcal{L}_S + \mathcal{L}_L + \mathcal{L}_U \tag{4}$$

All three loss terms ($\mathcal{L}_S$, $\mathcal{L}_L$, and $\mathcal{L}_U$) are computed as mean losses over the samples within their respective mini-batches,

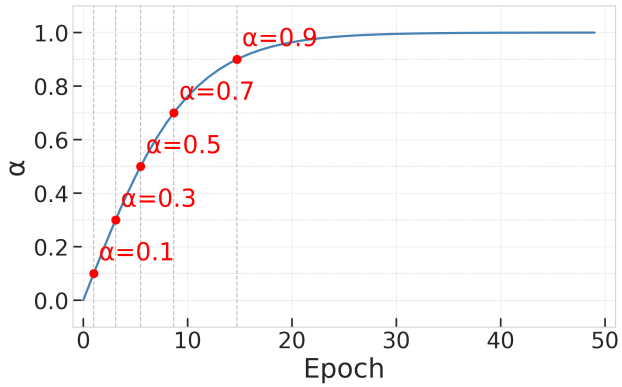

Figure 5. Dynamic scheduling of the adaptation parameter α over training epochs.

ensuring that the contribution of each term is normalized with respect to the number of samples in the corresponding mini-batch. The parameter α controls both the degree of source label correction and the weighting of source domain loss $\mathcal{L}_S$. In the early stages, a small α preserves the original source supervision, while a gradual increase in α encourages the model to rely more on target adaptive signals. Based on the above formulation, the complete training procedure of the proposed SSDA framework is summarized in Algorithm 1.

4. EXPERIMENT

We conducted two case studies by applying our proposed approach: turbofan engine and rolling bearing. The details are described in the following subsections.

4.1. Case Study I: Turbofan Engine

4.1.1. Datasets Description

In the first case study, we evaluate our proposed method on two publicly available turbofan engine datasets. A turbofan engine consists of five major components: the fan, low- and high-pressure compressor (LPC & HPC), high- and low-pressure turbine (HPT & LPT). Failure modes are typically associated with these components.

CMAPSS (Commercial Modular Aero-Propulsion System Simulation) is the most common dataset used to benchmark RUL estimation methods since its release (Saxena et al., 2008). It consists of four subsets and simulates two unknown failure modes. CMAPSS contains degradation information recorded from 21 sensors and three operating condition parameters. All training sets are treated as run-to-failure, whereas test sets are provided with specific RUL labels. In CMAPSS, each flight/cycle of the turbofan engine is represented as a single data instance. In other words, CMAPSS follows a per-cycle sampling scheme.

The second dataset, known as N-CMAPSS (New Commer-

Algorithm 1 Proposed SSDA approach for RUL prediction

Inputs: $D_S = \{(\mathbf{x}_i^s, y_i^s)\}_{i=1}^{N_S}$, $D_L = \{(\mathbf{x}_i^l, y_i^l)\}_{i=1}^{N_L}$, $D_U = \{(\mathbf{x}_i^u)\}_{i=1}^{N_U}$
Initialize: $f \leftarrow$ model, $\theta \leftarrow$ model parameters, $\gamma_0 \leftarrow$ initial learning rate, $\phi \leftarrow$ adaptation parameter, $E \leftarrow$ maximum epochs, $B \leftarrow$ batch size
for epoch = 1 to E **do**
  $p \leftarrow \text{epoch}/E$
  $\alpha \leftarrow \frac{2}{1+\exp(-\phi \cdot p)} - 1$
  $\gamma \leftarrow \frac{\gamma_0}{(1+\phi \cdot p)^{\eta}}$
  Update source labels via progressive correction:
  $\tilde{y}^s \leftarrow (1-\alpha) \cdot y^s + \alpha \cdot f_\theta(\mathbf{x}^s)$
  $\tilde{y}^u \leftarrow f_\theta(\mathbf{x}^u)$
  for each batch $(\mathbf{x}^s, \tilde{y}^s), (\mathbf{x}^l, y^l), (\mathbf{x}^u, \tilde{y}^u)$ **do**
    $\hat{y}^s \leftarrow f_\theta(\mathbf{x}^s)$
    $\mathcal{L}_S \leftarrow \text{Loss}(\hat{y}^s, \tilde{y}^s)$
    $\hat{y}^l \leftarrow f_\theta(\mathbf{x}^l)$
    $\mathcal{L}_L \leftarrow \text{Loss}(\hat{y}^l, y^l)$
    $\hat{y}^u \leftarrow f_\theta(\mathbf{x}^u)$
    $\mathcal{L}_U \leftarrow \text{Loss}(\hat{y}^u, \tilde{y}^u)$
    $\mathcal{L}_{\text{SSDA}} \leftarrow \alpha \mathcal{L}_S + \mathcal{L}_L + \mathcal{L}_U$
    $\theta \leftarrow \theta - \gamma \cdot \nabla_\theta \mathcal{L}_{\text{SSDA}}$
  end for
end for
return Trained model f_θ

cial Modular Aero-Propulsion System Simulation), contains eight subsets and simulates seven failure modes (Arias Chao et al., 2021). N-CMAPSS is developed using the same modeling strategy as CMAPSS but incorporates flight conditions recorded from real commercial jet operations. Each engine mainly contains recordings of four scenario descriptors, 38 sensors and four auxiliary variables. Both training and test sets are run-to-failure and provide RUL labels at each time step. In contrast to CMAPSS, failure labels are available for all time steps in N-CMAPSS. In addition, N-CMAPSS has a much higher sampling frequency of 1 Hz.

Fig. 6 shows a comparison of the kernel density estimation of four sensors and lists the mean and standard deviation of each feature from the FD001 and DS01. From this comparison, we can see that there are significant differences in the distribution of raw data between CMAPSS and N-CMAPSS. Given such differences, following the workflow proposed in (Fu et al., 2021) and (Nejjar et al., 2024) would lead to the failure of subsequent methods since normalization. Therefore, as described in section 3.2.3, we use respective statistical measures for normalization. In summary, the two datasets exhibit significant differences in sampling scheme, sensor configuration, and failure characteristics as summarized in Table 1, leading to a substantial domain gap. The specific subsets used in the following experiments are listed in Table 2 and Table 3.

Table 1. Comparison between CMAPSS and N-CMAPSS datasets.

	CMAPSS	**N-CMAPSS**
Sampling frequency	Per cycle	1 Hz
Sensor number	21	38
Failure modes	2	8
Failure annotation	No	Yes

Table 2. Overview of CMAPSS datasets.

Dataset	**Failure Modes**	**Operating Conditions**
FD001	1	1
FD002	1	6
FD003	2	1
FD004	2	6

Table 3. Overview of N-CMAPSS datasets.

Dataset	**Failure Modes**	**Failure Components**
DS01	1	HPT
DS03	1	LPT & HPT
DS05	1	HPC
DS07	1	LPT

4.1.2. Data Preprocessing

As described in section 3.2, preprocessing for different datasets often encounters challenges such as mismatched variables and inconsistent sampling frequencies.

First, we perform input feature alignment between the two datasets. This step involves directly extracting the intersection part of both datasets. After this process, 15 sensors are identified between CMAPSS and N-CMAPSS. We further observe a discrepancy in the pressure-related sensor variables provided by the two datasets. In CMAPSS, the recorded variables include P2 (fan inlet pressure) and epr (engine pressure ratio), while in N-CMAPSS the variables include P2 and P50 (total pressure at low-pressure turbine outlet). According to (Saxena et al., 2008), epr is defined as the ratio between P50 and P2, i.e., epr $=$ P50/P2. Therefore, the corresponding variables can be derived across the two datasets. To achieve consistent feature alignment across datasets, we adopt a sensor-oriented perspective and select P50 as the unified variable for subsequent analysis. As a result, a total of 16 common sensors are obtained, as summarized in Table 4.

We then proceed to frequency alignment. Prior to this step, N-CMAPSS data are first preprocessed following the procedure described in (Lövberg, 2021), to improve the visibility of degradation trends. Since the sampling frequency of N-CMAPSS is much higher than that of CMAPSS, we average the N-CMAPSS measurements within each cycle, thereby reducing its temporal resolution to match that of CMAPSS. Through this process, the two datasets are transformed to share a consistent feature set and sampling scheme, enabling

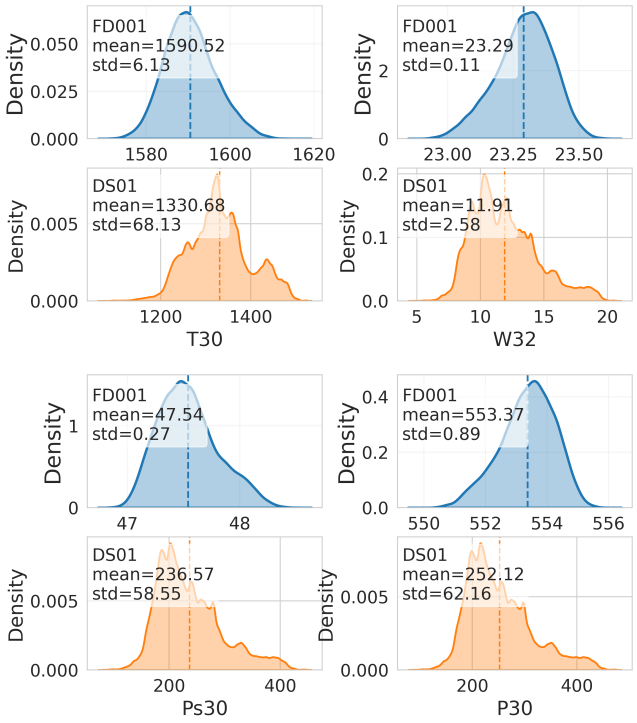

Figure 6. Kernel density estimations of features from CMAPSS and N-CMAPSS.

joint use in subsequent experiments.

After the above steps, all features of CMAPSS and N-CMAPSS are standardized using the Z-score normalization, respectively. Furthermore, for the FD002 and FD004 subsets of CMAPSS, we first perform clustering and then apply Z-score normalization separately within each cluster to account for heterogeneous operating conditions (Zheng, Ristovski, Farahat, & Gupta, 2017). Note that no further feature selection or sensitivity analysis is performed. For complex systems, our intention is to retain as many features as possible, as degradation patterns may be embedded in subtle and less intuitive signals. Eliminating features prematurely could risk discarding useful information, especially when feature relevance can vary under different operating conditions and failure modes. Therefore, all 16 sensors are used in the following experiments. Fig. 7 shows an example of the processed sensor data from CMAPSS and N-CMAPSS.

Before conducting experiments, we briefly describe the RUL labeling of CMAPSS and N-CMAPSS. In CMAPSS, training data are considered run-to-failure, and RUL is commonly defined using a piece-wise function (Fu et al., 2021). We adopt the same strategy with an empirically set threshold at 125. This threshold prevents excessively large early-stage RUL values from dominating the regression loss. In the proposed SSDA approach, such thresholding also helps stabilize cross-domain learning by reducing the distributional imbalance between early-stage and degradation-stage samples. In contrast, N-CMAPSS provides the ground-truth RUL label at every cycle. Therefore, we directly use the official RUL labels in N-CMAPSS for model training.

Table 4. Processed common sensors of CMAPSS and N-CMAPSS datasets.

#	Symbol	Description	Unit
1	alt	Altitude	ft
2	Mach	Flight Mach number	-
3	T2	Total temperature at fan inlet	°R
4	T24	Total temperature at LPC outlet	°R
5	T30	Total temperature at HPC outlet	°R
6	T50	Total temperature at LPT outlet	°R
7	P2	Total pressure at fan inlet	psia
8	P15	Total pressure in bypass-duct	psia
9	P30	Total pressure at HPC outlet	psia
10	P50	Total pressure at LPT outlet	psia
11	Ps30	Static pressure at HPC outlet	psia
12	Nf	Physical fan speed	rpm
13	Nc	Physical core speed	rpm
14	phi	Ratio of fuel flow to Ps30	-
15	W31	HPT coolant bleed	lbm/s
16	W32	LPT coolant bleed	lbm/s

4.1.3. Implementation Details

Two typical baseline models are selected for our experiments (Zheng et al., 2017; W. Yu, Kim, & Mechefske, 2019). The first model is a deep LSTM model, in which the feature extractor consists of two LSTM layers with 32 and 64 hidden units, respectively. On top of this extractor, a RUL regressor is implemented using two fully connected layers, each containing 8 hidden units. The second model is a Bidirectional Gated Recurrent Unit (BiGRU) model, which includes a single bidirectional GRU layer with 200 hidden units. Both models use ReLU as the activation function.

CMAPSS and N-CMAPSS are treated alternately as the source domain, and the trained models are transferred to the other dataset as the target domain. For the target domain, we randomly sample 10%, 20%, and 30% of the data as labeled instances, while the remaining data are treated as unlabeled. SSDA training is then performed under these different labeling ratios.

We employ a sliding window to segment the multivariate time series into fixed-length sequences. In our case, a window length of 10 is used with a step size of 1. The models are trained using the mean squared error (MSE) loss function and optimized with Adam, with an initial learning rate of 0.001. All experiments are conducted for 50 epochs with a batch size of 64. All reported results are obtained by averaging over five independent runs.

In addition to the experimental setup described above, we designed several sets of comparative experiments to further vali-

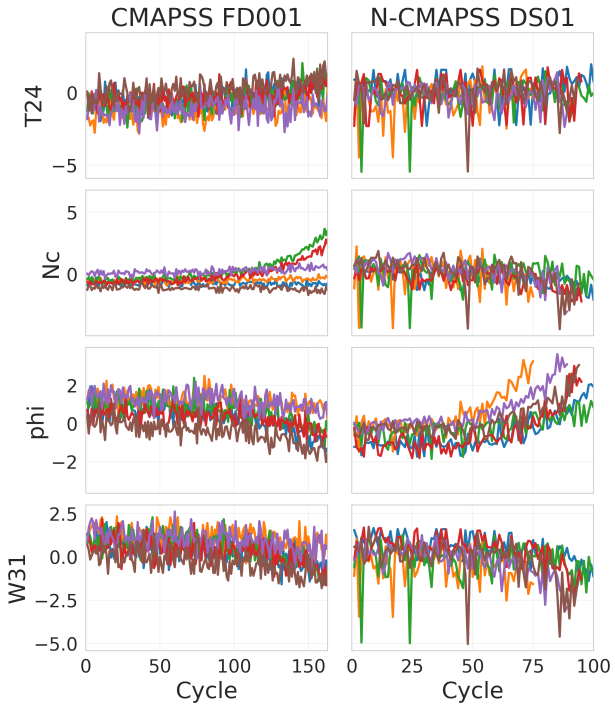

Figure 7. Example of processed CMAPSS and N-CMAPSS sensor data.

date the effectiveness of the proposed method. First, we compared the proposed dynamic adjustment strategy of α with the fixed-weight strategy. Second, we further compared our proposed method with other UDA methods, including deep CORAL (Sun & Saenko, 2016), DANN (Ganin & Lempitsky, 2015), and LAMA-Net (Joseph & Lalwani, 2025). To ensure a fair comparison, we formulate the optimization objectives of deep CORAL, DANN and LAMA-Net under both UDA and SSDA settings in a unified manner. For unsupervised deep CORAL, the training loss function is:

$$\mathcal{L}_{\text{Deep-CORAL}} = \mathcal{L}_S + \lambda \cdot \mathcal{L}_{\text{CORAL}} \tag{5}$$

where

$$\mathcal{L}_{\text{CORAL}} = \frac{1}{4d^2} \left\| \mathbf{C}_S - \mathbf{C}_T \right\|_F^2. \tag{6}$$

$\mathbf{C}_S$ and $\mathbf{C}_T$ denote the covariance matrices of the source and target features, respectively. In the semi-supervised setting, we select a form similar to (Yang & Lang, 2022), in which both source and labeled target data are used for supervision:

$$\mathcal{L}_{\text{SSDA-CORAL}} = \mathcal{L}_S + \mathcal{L}_L + \lambda \cdot \mathcal{L}_{\text{CORAL}}. \tag{7}$$

For DANN, the loss function (Ganin & Lempitsky, 2015) is:

$$\mathcal{L}_{\text{DANN}} = \mathcal{L}_S + \alpha \cdot \mathcal{L}_{\text{adv}} \tag{8}$$

where $\mathcal{L}_{\text{adv}}$ is the binary cross-entropy loss of the domain classifier, trained adversarially via a gradient reversal layer (GRL). Similarly, we attempt to extend DANN to a semi-supervised form, in which case the loss function becomes

$$\mathcal{L}_{\text{SSDA-DANN}} = \mathcal{L}_{\text{S}} + \mathcal{L}_{\text{L}} + \alpha \mathcal{L}_{\text{adv}}. \tag{9}$$

We further compare against LAMA-Net (Joseph & Lalwani, 2025), an encoder-decoder based DA method that combines latent alignment via MMD and manifold learning for turbofan engine RUL prediction. For unsupervised LAMA-Net, the training loss function is:

$$\mathcal{L}_{\text{LAMA-Net}} = \mathcal{L}_S + \lambda_m \mathcal{L}_{\text{MMD}} + \lambda_r \mathcal{L}_{\text{recon}} + \lambda_s \mathcal{L}_{\text{smooth}} \tag{10}$$

where $\mathcal{L}_{\text{MMD}}$ denotes the MMD loss computed at both the bottleneck layer and the linear projection layer, $\mathcal{L}_{\text{recon}}$ is the reconstruction loss from a GRU-based autoencoder, and $\mathcal{L}_{\text{smooth}}$ is a smoothness constraint defined for manifold learning. Similarly, we extend LAMA-Net to a semi-supervised form by incorporating labeled target supervision:

$$\mathcal{L}_{\text{SSDA-LAMA}} = \mathcal{L}_S + \mathcal{L}_L + \lambda_m \mathcal{L}_{\text{MMD}} + \lambda_r \mathcal{L}_{\text{recon}} + \lambda_s \mathcal{L}_{\text{smooth}} \tag{11}$$

We also include a fine-tuning baseline, in which the model is first pre-trained on the complete source domain dataset and then fine-tuned using 20% labeled target domain data, without any DA mechanism or unlabeled data.

4.1.4. Evaluation Metrics

The root mean square error (RMSE) and NASA score function (Saxena et al., 2008) are used as evaluation metrics for RUL estimation. The definitions of RMSE and the NASA score function for turbofan engines are given as follows:

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (\hat{y}_i - y_i)^2} \tag{12}$$

$$\text{NASA Score} = \begin{cases} \frac{1}{N} \sum_{i=1}^{N} \exp\left(-\frac{\hat{y}_i - y_i}{13}\right) - 1, & \hat{y}_i < y_i \\ \frac{1}{N} \sum_{i=1}^{N} \exp\left(\frac{\hat{y}_i - y_i}{10}\right) - 1, & \hat{y}_i \geq y_i \end{cases} \tag{13}$$

RMSE is a symmetric metric that assigns equal penalties to early and late predictions. In contrast, the NASA score function is an asymmetric metric that imposes heavier penalties on late predictions. Together, these two metrics provide a comprehensive evaluation of model performance, where smaller RMSE and NASA scores indicate more accurate RUL predictions. Note that for CMAPSS, the metrics are computed at the last time step of each sliding window, while for N-CMAPSS, they are calculated over all time steps.

4.1.5. Experiment Results

We compare and analyze the experimental results of the proposed SSDA approach on CMAPSS and N-CMAPSS, as shown

in Fig. 8 and Fig. 9. The results present the performance metrics of RMSE and NASA Score for baseline models under varying target domain label ratios ($0.1, 0.2, 0.3$), along with relative performance differences compared to the TO setting.

As discussed in the previous subsection, there is a substantial domain gap between CMAPSS and N-CMAPSS. Such discrepancies make cross-dataset RUL prediction particularly challenging and limit the effectiveness of conventional domain adaptation methods such as UDA, which typically rely on implicit distribution alignment assumptions. Despite this difficulty, even at a low labeled ratio of 0.1, both models achieve reasonably good predictive performance across tasks. This suggests that the proposed approach is capable of effectively bridging the domain gap and transferable degradation representations.

For the adaptation tasks from N-CMAPSS to CMAPSS, the results are particularly encouraging. In many tasks, the performance of both models surpasses that of the TO models, especially in terms of the NASA Score, where the improvements are substantial. This highlights the effectiveness of the proposed approach. In contrast, when transferring from CMAPSS to N-CMAPSS, the overall improvements are less pronounced. However, once the target domain labeled ratio reaches 0.3, the proposed model's performance is comparable to, and occasionally surpasses, the results achieved in the TO setting. A further comparison across tasks shows that FD003 is relatively more challenging in Fig. 8, as improvements are limited, while the transfer to DS05 consistently shows better results in Fig. 9.

It is also important to note that, in this study, the N-CMAPSS dataset is down-sampled by averaging the data within each flight cycle. Although trend-based preprocessing is applied, such down-sampling inevitably discards detailed information and reduces representativeness, which further increases the difficulty of the task. Interestingly, we observe that performance improvements are more substantial in target domains without down-sampling, as shown in Fig. 8. This indicates that the proposed alignment strategy is particularly effective in improving performance for lower frequency datasets. However, it should be noted that when the labeling rate in the target domain is low (e.g., 10%), the proposed method generally performs worse than TO. This suggests that when available supervisory information is insufficient, the model still struggles to fully learn the degradation patterns of the target domain.

Based on the experimental results above, we conducted further analysis of the proposed method using FD001 and DS01 as shown in Table 5. In these experiments, deep LSTM model is used and 20% of the target domain data are selected as labeled samples. First, as evidenced by Table 5, model's performance is quite sensitive to the value of α and the optimal α varies between different transfer tasks. This indicates that fixed-value strategies struggle to adapt to different task scenarios simultaneously. In contrast, the proposed dynamic α method achieved optimal performance in both tasks, validating its effectiveness in adaptively balancing different learning objectives.

To investigate the contribution of each component in the proposed framework, we conducted a loss ablation study as shown in Table 6. The results demonstrate that each component contributes to the final performance. Removing the pseudo-labeling term $\mathcal{L}_U$ leads to notable performance degradation, particularly when transferring from N-CMAPSS to CMAPSS. Further degradation occurs when the progressive label correction is removed while retaining all loss terms, highlighting the importance of the correction mechanism for cross-dataset adaptation. Notably, the consistent improvement brought about by $\mathcal{L}_U$ in both transfer directions highlights the positive contribution of unlabeled target data to the adaptation process. These results also suggest that the dynamic scheduling of α effectively moderates the influence of pseudo-labels during the early stages of training. Since α simultaneously governs both the progressive label correction in Eq. (1) and the weighting of $\mathcal{L}_S$ in Eq. (4), we further investigate the design choice of using α as the scaling factor for $\mathcal{L}_S$ by comparing it with three alternative scaling strategies, while keeping the progressive label correction in Eq. (1) unchanged. Specifically, three alternative strategies are evaluated: (1) using a fixed scaling factor of 0.5, (2) eliminating the α scaling (i.e., assigning $\mathcal{L}_S$ a fixed weight of 1), and (3) scaling $\mathcal{L}_S$ by α^2 instead of α. As shown in Table 6, none of the alternative scaling strategies consistently achieves comparable performance to the proposed formulation in both transfer directions. In particular, the relative ranking between α^2 scaling and the unscaled setting reverses between the two directions, indicating that the relative effectiveness of static scaling strategies depends on the transfer task. These results suggest that the use of the same dynamic α for both the correction of the source label and the weighting of the source domain loss provides a more effective and consistent weighting strategy than the alternative scaling schemes considered in this study.

In order to validate the necessity of domain-specific normalization, we compared the proposed normalization strategy with normalizing both domains using the statistics of the source domain as shown in Table 7. The result shows that applying source domain statistics results in substantial performance degradation in both transfer directions. This result empirically confirms the observation in Fig. 6, which shows that the significant distributional differences between CMAPSS and N-CMAPSS make source domain normalization inappropriate for cross-dataset adaptation. In contrast, domain-specific normalization ensures that each domain is standardized according to its own statistical characteristics, preserving the intrinsic degradation patterns and facilitating more effective

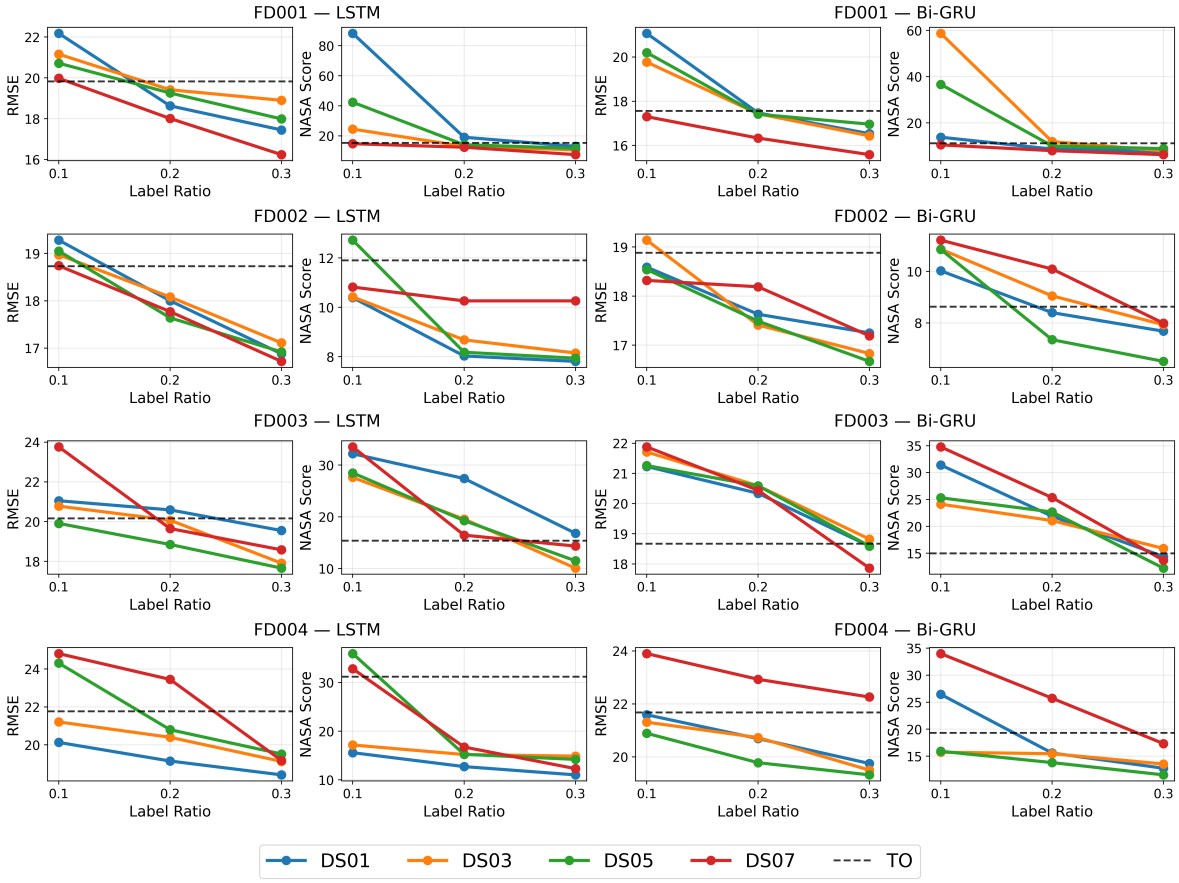

Figure 8. Experimental results when CMAPSS is the target domain.

cross-dataset knowledge transfer.

Fig. 10 shows the sensitivity of the proposed method to the length of the sliding window. All results are obtained by averaging over five independent runs, with error bars representing the standard deviation. As can be seen from the results, performance degrades significantly when the window size is small, as shorter sequences fail to capture sufficient degradation information. However, the performance remains relatively stable for larger window sizes. Based on these results, a window size of 10 is adopted for all experiments, as this length achieves the best overall performance while maintaining computational efficiency.

Table 8 shows that traditional UDA methods such as CORAL and DANN perform poorly in cross-dataset RUL prediction tasks, suggesting that relying solely on distribution alignment is insufficient to effectively mitigate discrepancies between different systems, even when using the same input features. LAMA-Net, a more recent UDA method for turbofan engine RUL prediction, also fails to achieve satisfactory performance under the UDA setting, further highlighting the limitations of unsupervised approaches in cross-dataset scenarios. After introducing 20% labeled target domain data, all SSDA baselines achieve substantial performance improvements over their UDA counterparts. As a practical transfer-learning baseline, the fine-tuning approach also achieves competitive performance compared with the other semi-supervised methods, and even outperforms them in several transfer tasks. More importantly, the proposed method consistently achieves the best performances in all four adaptation task pairs and both transfer directions. Although performance gain varies between different transfer tasks, the proposed method consistently outperforms both the fine-tuning baseline and the existing SSDA approaches. These results demonstrate that the proposed progressive label correction strategy is effective across a diverse range of cross-dataset adaptation scenarios, rather than being limited to a particular transfer pair.

4.2. Case Study II: Bearing

4.2.1. Dataset Description

In the second case study on rolling bearings, we also select two public datasets to validate the proposed approach.

We first select the IEEE PHM 2012 Data Challenge dataset recorded from the PRONOSTIA platform (Nectoux et al.,

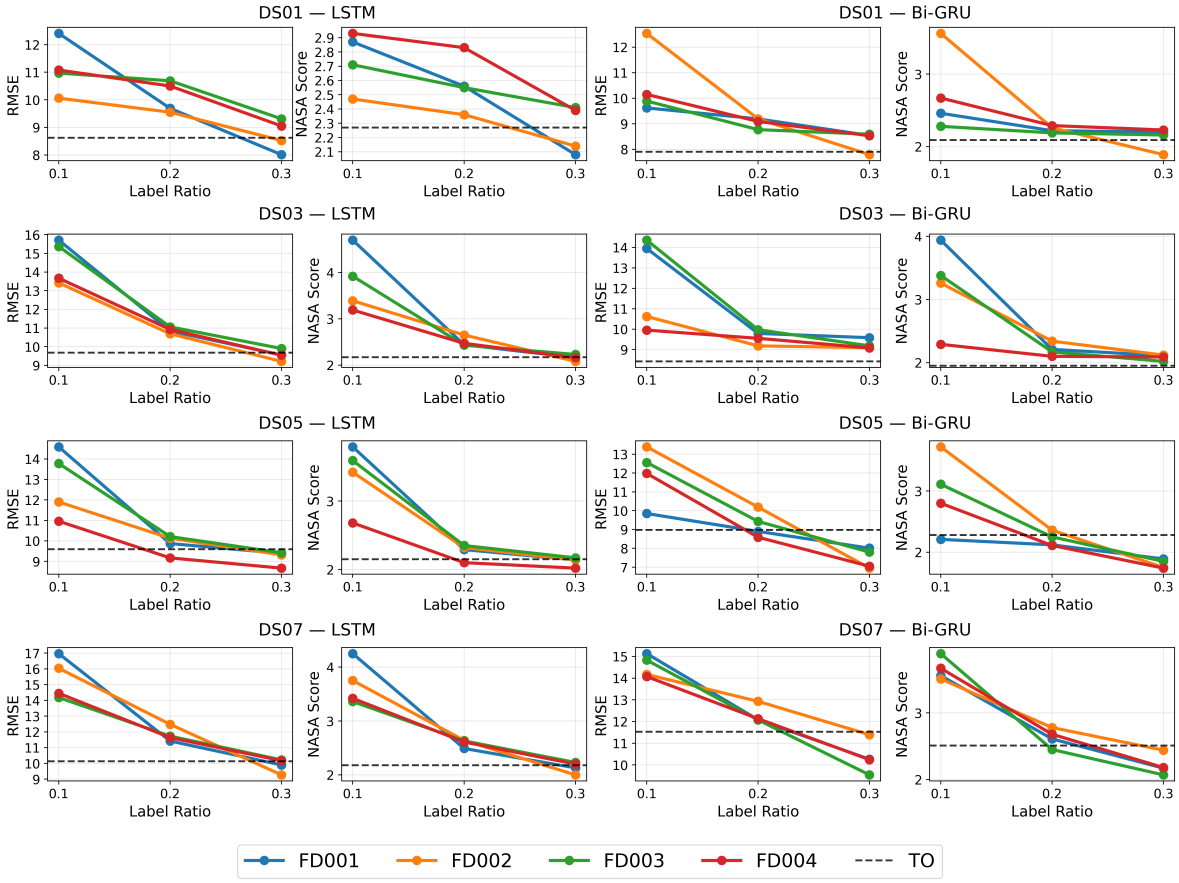

Figure 9. Experimental results when N-CMAPSS is the target domain.

Table 5. Comparison of α configuration strategies under two adaptation tasks (RMSE / NASA Score).

Setting	**DS01 → FD001**	**FD001 → DS01**
$\alpha = 0$	28.48/43.85	25.07/15.07
$\alpha = 0.1$	24.25/34.33	19.87/12.22
$\alpha = 0.3$	23.58/27.07	16.94/6.66
$\alpha = 0.5$	21.59/21.33	15.63/5.45
$\alpha = 0.7$	20.90/20.11	14.34/4.69
$\alpha = 0.9$	21.78/21.92	12.54/3.76
$\alpha = 1$	23.72/33.99	10.14/2.73
Proposed Method	**18.63/19.07**	**9.69/2.56**

Table 6. Loss ablation study under two adaptation tasks (RMSE / NASA Score). All settings use 20% target domain labels.

Setting	**DS01 → FD001**	**FD001 → DS01**
$\mathcal{L}_L$ only	36.22/56.05	34.29/26.70
w/o $\mathcal{L}_U$	27.71/42.35	14.50/4.81
w/o correction	22.59/22.83	13.34/4.11
Fixed scaling (0.5)	20.49/21.26	13.18/3.94
w/o $\mathcal{L}_S$ scaling	19.76/19.69	10.29/2.66
α^2 scaling	19.51/19.28	11.23/3.26
Proposed Method	**18.63/19.07**	**9.69/2.56**

2012). It provides experimental run-to-failure data collected from 17 rolling bearings. The recorded bearing lifetimes range from thirty minutes to several hours. All bearing data are divided into three subsets according to operating conditions. Horizontal and vertical vibration signals are measured by accelerometers mounted on the outer rings, and temperature signals are also available for certain bearings. The sampling frequency is 25.6 kHz, and 0.1 s of data are recorded every 10 seconds. Table 9 lists the bearings used in this study.

We choose XJTU-SY as the second dataset (Yaguo et al., 2019). It contains full-lifetime vibration signal data of 15 bearings. Similarly, the XJTU-SY dataset is divided into three subsets according to the operating conditions, as shown in Table 10. Compared with IEEE PHM 2012, XJTU-SY provides longer sampling durations and explicit fault location information. It records 1.28 seconds of data at a sampling frequency of 25.6 kHz, with each sample collected at one-minute intervals. Table 11 summarizes the differences between the two datasets, and Fig. 11 shows an example of raw sensor data .

Table 7. Effect of normalization strategy on adaptation performance (RMSE / NASA Score). Both settings use 20% target domain labels.

Normalization	DS01 → FD001	FD001 → DS01
Source statistics	41.10/60.75	21.68/10.39
Domain-specific	**18.63/19.07**	**9.69/2.56**

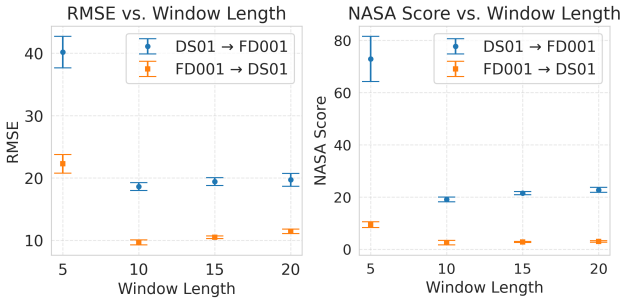

Figure 10. Sensitivity analysis of window length under different transfer tasks.

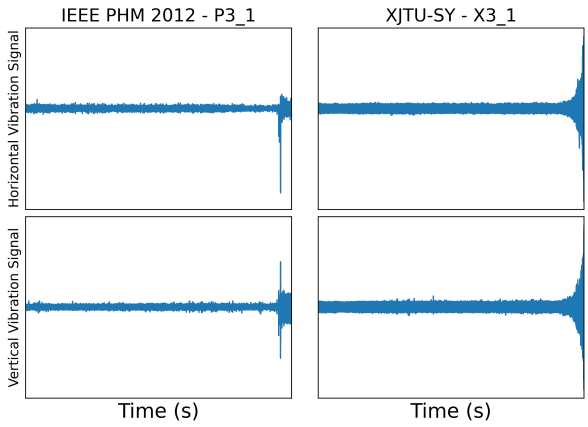

Figure 11. Example of IEEE PHM 2012 and XJTU-SY sensor data.

4.2.2. Data Preprocessing

Similarly to Case Study I, data preprocessing is performed for the bearing datasets. The two datasets used in this case share the same sampling frequency of 25.6 kHz, but differ in sampling duration and sampling interval. IEEE PHM 2012 has a sampling duration of 0.1 s with a sampling interval of 10 s, while XJTU-SY has a sampling duration of 1.28 second with a sampling interval of 1 minute. From the perspective of sampling duration, we align XJTU-SY with IEEE PHM 2012, as illustrated in Fig. 12. For each 1.28 s record in XJTU-SY, the data are segmented into sub-sequences of 0.1 s, and a sliding window with a length of 10 is then applied. In this way, each processed window from both datasets consistently represents the information corresponding to 10 minutes.

For input feature alignment, both datasets provide horizontal and vertical vibration signals, which can be directly used for processing. Vibration signals from bearings contain abundant information, and deep networks are capable of extracting features directly from raw signals for diagnostic tasks. Nevertheless, the application of feature engineering can often lead to improved inference performance and faster convergence. In particular, the marginal spectrum (MS) has been shown to achieve superior feature extraction and training performance compared to classical methods such as the Fast Fourier Transform (FFT) (Lu et al., 2024). Therefore, we adopt MS as the feature representation of the vibration signals. The MS is obtained by integrating the Hilbert spectrum over time, capturing the overall contribution of each frequency component and more accurately reflecting the time–frequency characteristics within a given period. We set the number of frequency bins to 2560 in the MS calculation which means that each 0.1 s raw segment is transformed into 2560 spectral features (Lu et al., 2024).

In both IEEE PHM 2012 and XJTU-SY datasets, all bearings operate under run-to-failure conditions. However, since the actual life cycle lengths vary across bearings, most studies on bearing RUL prediction adopt a normalization procedure for the true RUL values. Consequently, the RUL of each bearing is scaled to the range $[0, 1]$.

4.2.3. Implementation Details

We use the same DNN architectures as in Case Study I for the following experiments. Note that although the RUL values of bearings are normalized to the range $[0, 1]$, we still use ReLU as the activation function.

Similarly to Case Study I, IEEE PHM 2012 and XJTU-SY are alternately treated as the source and target domains, and the SSDA results are directly compared with those of the TO setting. The models are trained using the mean squared error (MSE) loss and optimized with the Adam optimizer, with an initial learning rate of 0.0005. Each task is trained for 30 epochs with a mini-batch size of 64, and all reported results are obtained by averaging over five independent runs.

We also compare the proposed method against Deep-CORAL, DANN, and their semi-supervised extensions SSDA-CORAL and SSDA-DANN, as well as a fine-tuning baseline. In addition, we include ERCDAN (Lu et al., 2024), a recent DA method specifically designed for rolling bearing RUL prediction, as an additional baseline. The UDA loss of ERCDAN is:

$$\mathcal{L}_{\text{ERCDAN}} = \mathcal{L}_S + \mathcal{L}_{\text{MK-MMD}} \cdot \lambda_{\text{att}} \tag{14}$$

where $\mathcal{L}_{\text{MK-MMD}}$ denotes the multi-kernel MMD loss for domain alignment, and λ_{att} is an exponential attenuation coefficient. Similarly, we extend ERCDAN to a semi-supervised

Table 8. Comparison of different DA methods across four cross-dataset adaptation task pairs (RMSE / NASA Score) using the deep LSTM model. Deep-CORAL, DANN and LAMA-Net are evaluated under the UDA setting (0% target labels); all other methods use 20% target domain labels.

Training Method	DS01→FD001	FD001→DS01	DS03→FD002	FD002→DS03	DS05→FD003	FD003→DS05	DS07→FD004	FD004→DS07
Deep-CORAL	75.85/1569	57.39/4192	61.63/700.2	55.70/1954	61.44/937.4	67.94/3262	59.33/547.0	68.49/6318
DANN	50.14/151.6	55.45/3246	59.30/584.9	60.89/3998	60.46/459.3	67.09/3248	61.38/971.6	64.27/6638
LAMA-Net	56.92/356.3	52.99/3286	56.11/535.0	51.07/1747	56.73/526.6	58.24/2857	52.97/462.1	52.35/1945
SSDA-CORAL	23.25/26.35	12.38/3.76	20.82/12.71	15.47/4.16	21.57/26.34	12.34/3.45	28.26/42.11	16.59/8.50
SSDA-DANN	20.77/19.66	11.12/3.11	23.10/21.83	12.29/3.44	25.52/34.38	11.34/2.75	24.20/36.49	14.19/3.80
SSDA-LAMA-Net	22.01/21.33	11.09/2.51	22.72/21.23	14.61/4.01	23.63/28.83	12.74/3.92	25.68/38.01	13.06/2.93
Fine-tuning	19.65/27.07	11.41/2.70	20.13/9.91	11.54/2.85	20.98/25.12	11.28/2.73	23.94/23.93	15.38/5.22
Proposed Method	**18.63/19.07**	**9.69/2.56**	**18.08/8.68**	**10.69/2.65**	**18.85/19.27**	**10.21/2.35**	**23.45/16.75**	**11.64/2.61**

Table 9. Descriptions of the IEEE PHM 2012 datasets.

Dataset	Training Bearings	Test Bearings
P1	P1_1, P1_2	P1_7
P2	P2_1, P2_2	P2_6
P3	P3_1, P3_2	P3_3

Table 10. Descriptions of the XJTU-SY datasets.

Dataset	Training Bearings	Test Bearings
X1	X1_1, X1_2, X1_3	X1_4, X1_5
X2	X2_1, X2_2, X2_3	X2_4, X2_5
X3	X3_1, X3_2, X3_3	X3_4, X3_5

Table 11. Comparison between IEEE PHM 2012 and XJTU-SY datasets.

	IEEE PHM 2012	XJTU-SY
Sampling frequency	25.6 kHz	25.6 kHz
Sampling interval	10 s	1 min
Duration per sample	0.1 s	1.28 s
Failure annotation	No	Yes

form:

$$\mathcal{L}_{\text{SSDA-ERCDAN}} = \mathcal{L}_S + \mathcal{L}_L + \mathcal{L}_{\text{MK-MMD}} \cdot \lambda_{\text{att}} \tag{15}$$

4.2.4. Evaluation Metrics

RUL prediction for bearings is formulated as a regression task. Therefore, RMSE is selected as one of the evaluation metrics. In addition, the IEEE PHM 2012 Data Challenge introduced an asymmetric score function specifically designed for this task, which is used as another evaluation criterion (Nectoux et al., 2012). The mathematical definition of this score function is given as follows:

$$Er_i = 100 \times \frac{y_i - \hat{y}_i}{y_i} \tag{16}$$

$$\mathrm{A}_i = \begin{cases} \exp\big(-\ln(0.5) \cdot (Er_i/5)\big), & \text{if } Er_i \leq 0 \\ \exp\big(+\ln(0.5) \cdot (Er_i/20)\big), & \text{if } Er_i > 0 \end{cases} \tag{17}$$

$$\text{Score} = \frac{1}{N} \sum_{i=1}^{N} (\mathrm{A}_i) \tag{18}$$

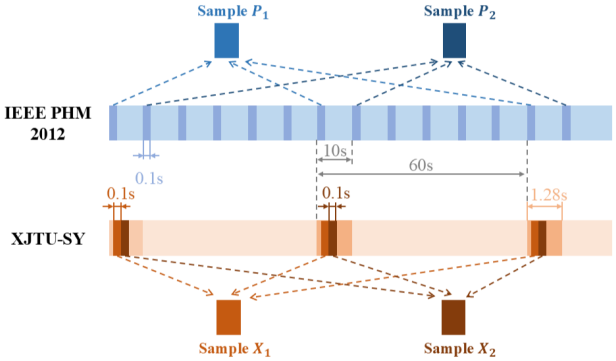

Figure 12. Data preprocessing for the bearing case study.

Similarly to the NASA score, this score function also assigns higher penalties to late predictions. The score ranges from 0 to 1, where values closer to 1 indicate better prediction performance.

When the target domain is the XJTU-SY dataset, two test sets are provided for each operating condition. In this case, RMSE and the score are first computed for each test bearing separately and then averaged. In contrast, when the target domain is the IEEE PHM 2012 dataset, both RMSE and the score are computed directly on the test bearing.

4.2.5. Experiment Results

Fig. 13 and Fig. 14 show the experimental results on the IEEE PHM 2012 and XJTU-SY datasets. The results report the prediction performance in terms of RMSE and Score under different target domain labeled ratios. From an overall perspective, RMSE consistently decreases while Score increases as the labeled ratio increases, indicating progressively improved RUL prediction accuracy. In most tasks, when the labeled ratio reaches 0.3, both models generally achieve performance that exceeds the TO setting. This trend is consistent with the observations in Case Study I, suggesting that the proposed SSDA framework remains effective across different types of systems.

As shown in Fig. 14, even under low labeled ratio conditions, the models trained with SSDA surpass the TO baseline in multiple tasks, with particularly notable improvements in

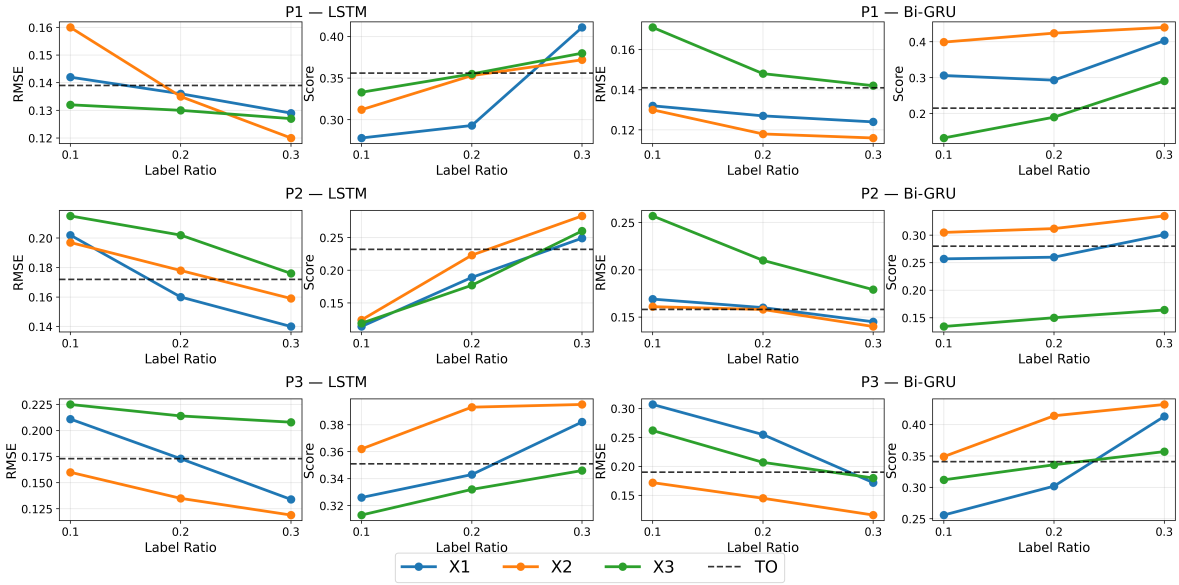

Figure 13. Experimental results when IEEE PHM 2012 is the target domain.

Score. This suggests that when XJTU-SY serves as the target domain, the proposed method can effectively capture underlying degradation patterns, even with limited labeled data. In contrast, when IEEE PHM 2012 is treated as the target domain (Fig. 13), the improvements are relatively limited, especially under low label ratio conditions. This asymmetry may be attributed to differences in data representativeness. XJTU-SY is a more recent dataset and provides more complete and informative degradation trajectories. As illustrated in Fig. 11, the degradation patterns in XJTU-SY are more pronounced and consistent compared to those of IEEE PHM 2012. This highlights that the effectiveness of cross-domain adaptation is influenced not only by the alignment strategy but also by the quality and representativeness of degradation trajectories in the target domain. It further suggests that learning from more informative target domain data can significantly enhance adaptation performance.

To further validate the effectiveness of the proposed method, Table 12 compares representative UDA methods and SSDA methods with 20% labeled target data. The proposed method consistently achieves the best RMSE performance in both directions. These results are consistent with the findings in Case Study I, suggesting that the proposed method provides additional benefits over existing DA approaches in cross-dataset adaptation settings.

It is also worth noting that the labeled target data are randomly selected, ensuring the generality and robustness of the reported results. In general, both adaptation directions validate the effectiveness of the proposed approach. These results further demonstrate that the proposed framework generalizes well across different types of industrial systems and

Table 12. Comparison of different DA methods under two adaptation tasks (RMSE / Score) using the BiGRU model. Deep-CORAL, DANN and ERCDAN are evaluated under the UDA setting (0% target labels); all other methods use 20% target domain labels.

Training Method	**X3 → P3**	**P3 → X3**
Deep-CORAL	0.212/0.224	0.324/0.253
DANN	0.227/0.215	0.255/0.277
ERCDAN	0.235/0.290	0.264/0.220
SSDA-CORAL	0.191/0.328	0.265/0.275
SSDA-DANN	0.202/0.299	0.224/0.233
SSDA-ERCDAN	0.180/**0.341**	0.209/0.307
Fine-tuning	0.198/0.307	0.214/0.322
Proposed Method	**0.165**/0.308	**0.207/0.336**

cross-dataset adaptation scenarios. Nevertheless, a common trend can be observed in both case studies: when the labeling rate in the target domain is low, the proposed method often performs worse than TO. This suggests that, under limited supervised information, the model's ability to capture target domain degradation patterns remains constrained.

5. CONCLUSION

In this study, we propose a novel semi-supervised domain adaptation (SSDA) framework designed to tackle the challenges of cross-dataset RUL prediction. Unlike most existing approaches that focus on unsupervised settings, our method establishes a more realistic and practical problem formulation by incorporating a limited number of labeled samples from the target domain. To address common discrepancies in feature space and sampling frequency between different datasets, we design a dedicated alignment pipeline. Then,

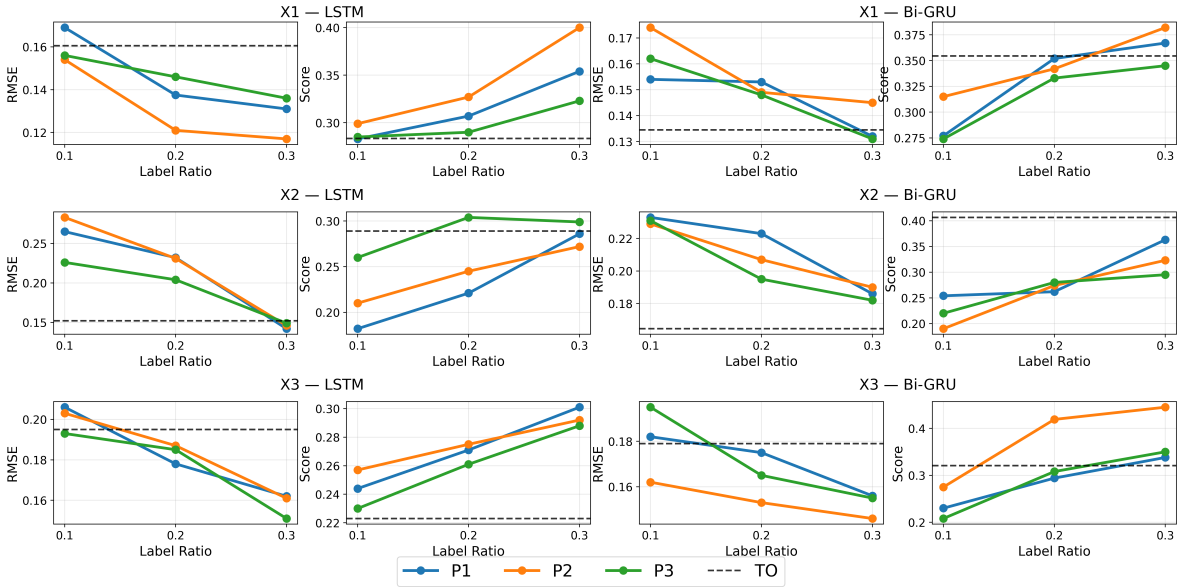

Figure 14. Experimental results when XJTU-SY is the target domain.

during SSDA training, the original RUL labels of the source domain are treated as noisy approximations of the target domain mapping and are progressively refined through a dynamically adaptive training strategy, which integrates pseudo-label generation and an adaptive weighting mechanism. Across both turbofan engine and bearing case studies, two consistent observations emerge. First, the proposed method can achieve reliable prediction even under very low labeling ratios, demonstrating its robustness and applicability. Second, prediction performance further improves as the proportion of labeled target data increases. Importantly, this study represents the first systematic investigation of cross-dataset RUL prediction, including the first successful realization of cross-dataset adaptation for turbofan engines. This work provides valuable insights for bridging the gap between academic research and practical PHM deployment.

6. FUTURE RECOMMENDATION

This study still has several limitations that warrant further investigation. First, from both case studies, it was observed that when transferring from a high-sampling-frequency dataset to a low-sampling-frequency dataset, models typically achieves a more significant performance improvement, whereas the results of transfer in the opposite direction are relatively poor. This phenomenon suggests that the downsampling strategy currently employed, while unifying the time scale, inevitably introduces the loss of fine-grained degradation information, making it difficult for low-frequency data to fully capture complex degradation processes. One promising approach to mitigating this issue is to apply statistical aggregation strategies. For example, representative statistical features could be extracted from unshared sensors and high-frequency signals within each cycle, rather than discarding or averaging them entirely. This approach could preserve complementary degradation information more effectively while maintaining cross-dataset compatibility. Secondly, although the proposed dynamic weight coefficient α demonstrates good performance, its scheduling is based solely on training progress and does not directly consider the model's prediction confidence in the target domain or the discrepancy between the source and target domains. As a result, during the early training phase, when target domain predictions are unreliable, increasing α may introduce noise into the source domain labels. An adaptive mechanism that adjusts α based on target domain validation performance could potentially mitigate this issue. However, implementing such a mechanism is non-trivial in our current setting, as all the labeled target data is used for training. Therefore, exploring more principled adaptive scheduling strategies remains an important direction for future work. Furthermore, when the target domain annotation ratio is low, the performance improvement of the method is relatively limited. While it still has certain advantages over traditional methods, its performance remains sensitive to the availability of labeled target data. This indicates that the current method still relies heavily on limited target domain supervision to correct source domain biases. Finally, the current framework relies to some extent on explicit preprocessing and alignment procedures, such as sensor feature filtering. While these steps help reduce domain-to-domain differences and improve model performance, they also make the method somewhat dependent on the prior design. In more complex or raw industrial scenarios, the integration of such alignment processes into the model itself—aiming for end-to-end and adaptive cross-domain modeling—remains a significant direction for future

research.

ACKNOWLEDGMENT

This research was partially supported by JSPS KAKENHI JP24K01110.

REFERENCES

Arias Chao, M., Kulkarni, C., Goebel, K., & Fink, O. (2021). Aircraft engine run-to-failure dataset under real flight conditions for prognostics and diagnostics. *Data*, *6*(1), 5.

Berghout, T., Mouss, L.-H., Bentrcia, T., & Benbouzid, M. (2021). A semi-supervised deep transfer learning approach for rolling-element bearing remaining useful life prediction. *IEEE Transactions on Energy Conversion*, *37*(2), 1200–1210.

Cheng, Y., Wang, C., Wu, J., Zhu, H., & Lee, C. K. (2022). Multi-dimensional recurrent neural network for remaining useful life prediction under variable operating conditions and multiple fault modes. *Applied Soft Computing*, *118*, 108507.

Chi, Y., Dong, Y., Wang, Z. J., Yu, F. R., & Leung, V. C. (2022). Knowledge-based fault diagnosis in industrial internet of things: a survey. *IEEE Internet of Things Journal*, *9*(15), 12886–12900.

Çınar, Z. M., Abdussalam Nuhu, A., Zeeshan, Q., Korhan, O., Asmael, M., & Safaei, B. (2020). Machine learning in predictive maintenance towards sustainable smart manufacturing in industry 4.0. *Sustainability*, *12*(19), 8211.

da Costa, P. R. d. O., Akçay, A., Zhang, Y., & Kaymak, U. (2020). Remaining useful lifetime prediction via deep domain adaptation. *Reliability Engineering & System Safety*, *195*, 106682.

Ferreira, C., & Gonçalves, G. (2022). Remaining useful life prediction and challenges: A literature review on the use of machine learning methods. *Journal of manufacturing systems*, *63*, 550–562.

Fu, S., Zhang, Y., Lin, L., Zhao, M., & Zhong, S.-s. (2021). Deep residual lstm with domain-invariance for remaining useful life prediction across domains. *Reliability Engineering & System Safety*, *216*, 108012.

Ganin, Y., & Lempitsky, V. (2015). Unsupervised domain adaptation by backpropagation. In *International conference on machine learning* (pp. 1180–1189).

Gbore, O. O., Shafiq, M., Jain, A. K., & McGlinchey, D. (2025). Towards reliable rul prediction: Impact of feature selection on degradation modelling. *International Journal of Prognostics and Health Management*, *16*(2).

Gebraeel, N., Lei, Y., Li, N., Si, X., Zio, E., et al. (2023). Prognostics and remaining useful life prediction of machinery: advances, opportunities and challenges. *Journal of Dynamics, Monitoring and Diagnostics*, 1–12.

Ghifary, M., Kleijn, W. B., Zhang, M., Balduzzi, D., & Li, W. (2016). Deep reconstruction-classification networks for unsupervised domain adaptation. In *European conference on computer vision* (pp. 597–613).

Glorot, X., Bordes, A., & Bengio, Y. (2011). Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proceedings of the 28th international conference on machine learning (icml-11)* (pp. 513–520).

Huang, C.-G., Zhu, J., Han, Y., & Peng, W. (2021). A novel bayesian deep dual network with unsupervised domain adaptation for transfer fault prognosis across different machines. *IEEE Sensors Journal*, *22*(8), 7855–7867.

Joseph, M., & Lalwani, V. (2025). Lama-net: Unsupervised domain adaptation via latent alignment and manifold learning for rul prediction. In *2025 5th international conference on ai-ml-systems (aimlsystems)* (pp. 8–18).

Lei, Y., Li, N., Gontarz, S., Lin, J., Radkowski, S., & Dybala, J. (2016). A model-based method for remaining useful life prediction of machinery. *IEEE Transactions on reliability*, *65*(3), 1314–1326.

Li, G., & Yairi, T. (2025). Semi-supervised domain adaptation with auxiliary task learning for rul prediction. In *2025 ieee international conference on prognostics and health management (icphm)* (pp. 1–8).

Li, X., Zhang, W., Li, X., & Hao, H. (2023). Partial domain adaptation in remaining useful life prediction with incomplete target data. *IEEE/ASME Transactions on Mechatronics*, *29*(3), 1903–1913.

Liu, C., & Gryllias, K. (2020). Unsupervised domain adaptation based remaining useful life prediction of rolling element bearings. In *Phm society european conference* (Vol. 5, pp. 10–10).

Liu, L., Song, X., & Zhou, Z. (2022). Aircraft engine remaining useful life estimation via a double attention-based data-driven architecture. *Reliability Engineering & System Safety*, *221*, 108330.

Long, M., Cao, Y., Wang, J., & Jordan, M. (2015). Learning transferable features with deep adaptation networks. In *International conference on machine learning* (pp. 97–105).

Lövberg, A. (2021). Remaining useful life prediction of aircraft engines with variable length input sequences. In *Annual conference of the phm society* (Vol. 13).

Lu, X., Jiang, Q., Shen, Y., Lin, X., Xu, F., & Zhu, Q. (2024). Enhanced residual convolutional domain adaptation network with cbam for rul prediction of cross-machine rolling bearing. *Reliability Engineering & System Safety*, *245*, 109976.

Luo, Q., Chang, Y., Chen, J., Jing, H., Lv, H., & Pan, T. (2020). Multiple degradation mode analysis via gated recurrent unit mode recognizer and life predic-

tors for complex equipment. *Computers in Industry*, *123*, 103332.

Merkt, O. (2019). On the use of predictive models for improving the quality of industrial maintenance: An analytical literature review of maintenance strategies. In *2019 federated conference on computer science and information systems (fedcsis)* (pp. 693–704).

Motiian, S., Piccirilli, M., Adjeroh, D. A., & Doretto, G. (2017). Unified deep supervised domain adaptation and generalization. In *Proceedings of the ieee international conference on computer vision* (pp. 5715–5725).

Nectoux, P., Gouriveau, R., Medjaher, K., Ramasso, E., Chebel-Morello, B., Zerhouni, N., & Varnier, C. (2012). Pronostia: An experimental platform for bearings accelerated degradation tests. In *Ieee international conference on prognostics and health management, phm'12*. (pp. 1–8).

Nejjar, I., Geissmann, F., Zhao, M., Taal, C., & Fink, O. (2024). Domain adaptation via alignment of operation profile for remaining useful lifetime prediction. *Reliability Engineering & System Safety*, *242*, 109718.

Pan, B., Wang, W., Wen, J., & Li, Y. (2023). Semi-supervised adversarial transfer networks for cross-domain intelligent fault diagnosis of rolling bearings. *Applied Sciences*, *13*(4), 2626.

Pei, Z., Cao, Z., Long, M., & Wang, J. (2018). Multi-adversarial domain adaptation. In *Proceedings of the aaai conference on artificial intelligence* (Vol. 32).

Saito, K., Kim, D., Sclaroff, S., Darrell, T., & Saenko, K. (2019). Semi-supervised domain adaptation via minimax entropy. In *Proceedings of the ieee/cvf international conference on computer vision* (pp. 8050–8058).

Saxena, A., Goebel, K., Simon, D., & Eklund, N. (2008). Damage propagation modeling for aircraft engine run-to-failure simulation. In *2008 international conference on prognostics and health management* (pp. 1–9).

Sun, B., & Saenko, K. (2016). Deep coral: Correlation alignment for deep domain adaptation. In *European conference on computer vision* (pp. 443–450).

Tsui, K. L., Chen, N., Zhou, Q., Hai, Y., & Wang, W. (2015). Prognostics and health management: A review on data driven approaches. *Mathematical Problems in Engineering*, *2015*(1), 793161.

Tzeng, E., Hoffman, J., Saenko, K., & Darrell, T. (2017). Adversarial discriminative domain adaptation. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 7167–7176).

Tzeng, E., Hoffman, J., Zhang, N., Saenko, K., & Darrell, T. (2014). Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*.

Yaguo, L., Tianyu, H., Biao, W., Naipeng, L., Tao, Y., & Jun, Y. (2019). Xjtu-sy rolling element bearing accelerated life test datasets: A tutorial. *Journal of Mechanical Engineering*, *55*(16), 1–6.

Yang, G., & Lang, H. (2022). Semisupervised heterogeneous domain adaptation via dynamic joint correlation alignment network for ship classification in sar imagery. *IEEE Geoscience and Remote Sensing Letters*, *19*, 1–5.

Yao, T., Pan, Y., Ngo, C.-W., Li, H., & Mei, T. (2015). Semi-supervised domain adaptation with subspace learning for visual recognition. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 2142–2150).

Yu, W., Kim, I. Y., & Mechefske, C. (2019). Remaining useful life estimation using a bidirectional recurrent neural network based autoencoder scheme. *Mechanical Systems and Signal Processing*, *129*, 764–780.

Yu, Y.-C., & Lin, H.-T. (2023, June). Semi-supervised domain adaptation with source label adaptation. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition (cvpr)* (p. 24100-24109).

Zhao, D., & Liu, F. (2022). Cross-condition and cross-platform remaining useful life estimation via adversarial-based domain adaptation. *Scientific Reports*, *12*(1), 878.

Zheng, S., Ristovski, K., Farahat, A., & Gupta, C. (2017). Long short-term memory network for remaining useful life estimation. In *2017 ieee international conference on prognostics and health management (icphm)* (pp. 88–95).

Zhuang, J., Jia, M., Ding, Y., & Ding, P. (2021). Temporal convolution-based transferable cross-domain adaptation approach for remaining useful life estimation under variable failure behaviors. *Reliability Engineering & System Safety*, *216*, 107946.

Zonta, T., Da Costa, C. A., da Rosa Righi, R., De Lima, M. J., Da Trindade, E. S., & Li, G. P. (2020). Predictive maintenance in the industry 4.0: A systematic literature review. *Computers & industrial engineering*, *150*, 106889.

Zou, Y., Yu, Z., Liu, X., Kumar, B., & Wang, J. (2019). Confidence regularized self-training. In *Proceedings of the ieee/cvf international conference on computer vision* (pp. 5982–5991).