The Journal of Supercomputing (2025) 81:573
https://doi.org/10.1007/s11227-024-06889-x

Leveraging working-condition-related features for enhanced cross-domain remaining useful life prediction of aircraft engines

Zhiyao Zhang[1,2] · Jiting Cheng[3] · Pengpeng Chen[4] · Shuang Gao[5] · Xiaohui Chen[2] · Enrico Zio[6,7]

Accepted: 23 December 2024

Abstract
Accurately predicting the remaining useful life (RUL) of aircraft engines across varying working conditions is challenging, particularly due to the absence of labeled data in the target domain. Traditional approaches in unsupervised domain adaptation (UDA) have largely focused on working-condition-invariant (domain-invariant) features, often overlooking the valuable contributions that working-condition-related (domain-related) features can provide. To address this limitation, we propose a novel UDA framework that explores the potential contributions of working-condition-related features and leverages the synergy between working-condition-invariant and working-condition-related features for cross-domain RUL prediction. Our approach involves the development of two specialized feature generators, one for working-condition-invariant features and another for working-condition-related features. We employ orthogonality constraints to optimize the expression of both feature types and ensure their independence within the feature space, which effectively reduces interference between them. Combined with a domain discriminator and a domain classifier, this setup allows our RUL predictor to harness the combined strengths of both feature types under varying working conditions. The effectiveness of our approach is validated through extensive experiments across twelve distinct working-condition scenarios, demonstrating a significant improvement in prediction accuracy, with a 9.3% reduction in root-mean-square error (RMSE).

Keywords Remaining useful life prediction · Domain adaption · Prognostics and health management · Working-condition shift · Aircraft engines

Extended author information available on the last page of the article

Published online: 03 March 2025

1 Introduction

Predicting the remaining useful life (RUL) of aircraft engines is pivotal for the efficacy of prognostics and health management (PHM) systems, with engine malfunctions potentially leading to substantial economic losses and heightened safety risks [1–5]. Aircraft engines are subjected to time-varying operating conditions and diverse fault modes, resulting in heterogeneous condition monitoring data distributions [6, 7]. Changes in flight altitude, air density, and temperature impact engine performance and fault modes. For instance, higher altitudes reduce efficiency and alter component stress, which can manifest as distinct wear patterns and behaviors across various operational scenarios. This situation can be depicted by Fig. 1, where the sensor signal distributions vary significantly across different working conditions, highlighting the substantial impact of the working environment on the sensor readings. For instance, the distribution under working condition A shows the highest peak, whereas working condition D exhibits a lower and more dispersed distribution. Such variations in the data distributions are critical for developing machine learning models, especially for tasks involving cross-domain learning and adaptation. The inherent variability in condition monitoring data significantly undermines the efficacy of traditional RUL prediction methodologies, leading to compromised accuracy amid shifts in working conditions. These methodologies often assume an identical distribution (i.i.d.) of training and test data, which is a presumption rarely valid for the RUL prediction task of aircraft engines. Consequently, the shift toward dynamic, non-identical working conditions urgently necessitates the development of innovative prediction models that can effectively handle diverse conditions and deliver reliable predictions.

The RUL prediction of aircraft engines across various working conditions can be considered a quintessential unsupervised domain adaptation (UDA) problem. This characterization stems from the fact that the RUL models must generalize across diverse engine operating environments (each resembling a distinct domain) without the benefit of labeled training data in many of these domains. In addressing this UDA problem, researchers have employed a variety of strategies. In the early works, approaches focused on re-weighting or selecting samples from the source domain or seeking an explicit feature space transformation from the source distribution to the target distribution, effectively mitigating the data shift problem within the shallow learning regime [8–11]. These shallow UDA methods have proven capable of bridging the gap between differing working conditions by

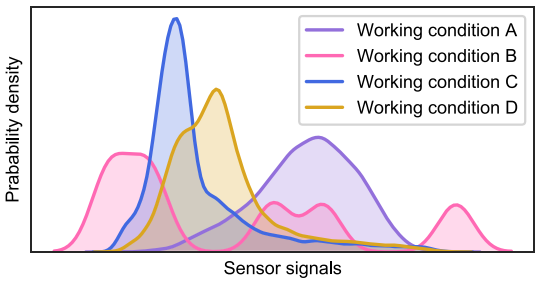

Fig. 1 The probability density distributions of condition monitoring data under four different working conditions

adapting the data distribution, thus facilitating a more accurate RUL prediction under varied working conditions. Although these shallow approaches provide a solid foundation for domain adaptation, they often rely on domain expertise and manual feature engineering to achieve the desired adaptability. More recently, leveraging deep learning techniques marks a significant shift in UDA strategies for RUL prediction. Deep UDA methods, utilizing deep neural networks, aim to automatically learn the domain-invariant feature that is robust across different operational states. By adopting an end-to-end learning approach, these methods mainly focus on extracting features that remain consistent across domains, and some of them also attempt to retain domain-specific information that is crucial for accurate RUL prediction. The emergence of deep UDA methods indicates a promising direction for future research in RUL prediction across varying working conditions.

Among the deep UDA methods, statistics-matching-based methods and adversarial-learning-based methods are the most prevalent [12]. The former, i.e, statistics-matching-based UDA methods, minimize domain discrepancy in a latent feature space through distribution dissimilarity/similarity metrics, such as maximum mean discrepancy (MMD) and correlation alignment (CORAL) and Wasserstein distance, to measure the distributions discrepancy [13–20]. Thus, statistics-matching-based UDA methods usually involve minimum optimization problems. The latter, i.e., adversarial-learning-based UDA methods, attempt to adaptively learn a measure of divergence, instead of the choose of a divergence measure. These methods learn domain-invariant representations to reduce the domain discrepancy via adversarial learning and min–max optimization [21–23]. More recently, some researchers have attempted to integrate adversarial-learning-based UDA methods with statistics-matching-based methods to enhance prediction efficiency [24–27].

Despite the good performance reported by these deep UDA methods, there is a limitation that hinders improved performance. Some UDA methods remove important domain-related information, and the domain-invariant feature generated cannot guarantee accurate cross-domain RUL prediction, as working conditions can impact the degradation process of aircraft engines. To improve prediction accuracy, researchers have explored extracting target-related features [28–30]. These methods attempt to preserve target-related mutual information while learning domain-invariant features via the integration of parameters-based and adversarial-learning-based UDA methods [28], the introduction of contrastive learning [29] and representation disentanglement learning [30]. However, in these studies, the target-related mutual information can be contaminated by the noise or the domain-invariant information during training without constraints. Besides, domain characteristics of target-related features are considered, but the source-specific features are ignored, which can also benefit the cross-domain RUL predictions [31].

In this research, we focus on predicting the RUL of aircraft engines across various working conditions, with a particular emphasis on incorporating working-condition-related features such as flight altitude and Mach number, which significantly influence engine degradation. By integrating these working-condition-specific insights into our RUL prediction model, we aim to improve the accuracy and generalization capabilities of the predictions, especially under shifting working conditions.

Motivated by the concept of feature disentanglement, we introduce a novel UDA method, named UDATrans, that builds upon the traditional DANN. Our approach enhances the model by including a working-condition-related feature generator and a domain classifier in conjunction with the traditional working-condition-invariant feature generator and domain discriminator. This setup allows for the independent extraction and optimization of both types of features, applying orthogonality constraints to reduce their mutual interference effectively. The domain discriminator drives adversarial learning to refine the use of working-condition-invariant features, whereas the domain classifier focuses on harnessing the potential of working-condition-related features. Consequently, the RUL predictor utilizes the synergy between these two feature types to provide robust RUL estimates under dynamically changing working conditions.

The main contributions of this work are fourfold:

- Considering working-condition-related factors such as altitude and Mach number that significantly impact the remaining useful life (RUL) of aircraft engines, this paper innovatively explores the potential contributions of working-condition-related features in scenarios involving cross-working conditions. Unlike traditional studies that often overlook or discard such working-condition-specific information and rely solely on working-condition-invariant (domain-invariant) features for RUL prediction, our approach integrates both working-condition-invariant and working-condition-related features. This integration not only aims to enhance the adaptability and accuracy of the predictive model across diverse working conditions but also addresses the gap in current methodologies by utilizing information that was previously neglected.
- To investigate the potential contributions of working-condition-related features to RUL predictions for aircraft engines, we employed orthogonality constraints to disentangle working-condition-invariant and working-condition-related features. This approach optimizes the expression of both types of features, ensuring their independence within the feature space and thereby reducing interference between them. Moreover, by integrating the use of discriminators and classifiers, we explored the interaction mechanisms between working-condition-invariant features and working-condition-related features. This study provides a new perspective on how these features collaboratively influence RUL predictions across varying working conditions.
- In the experimental study conducted on the C-MAPSS dataset, we investigated the potential contributions of domain-related (working-condition-related) features and explored the interaction mechanisms between working-condition-invariant features and working-condition-related features in cross-domain RUL predictions. Our UDATrans method achieved the best overall performance, outperforming state-of-the-art methods across various cross-working-condition scenarios. Besides, the experiment results demonstrated that in the majority of working-condition shift scenarios, the fault-mode shift problem poses a more significant challenge compared to the operating condition shift problem.

The remainder of the paper is organized as follows. Section 2 and Sect. 3 present the related work and problem formulation. The framework of the proposed method (UDATrans) is described in Sect. 4. Experiments are conducted in Sect. 5. We conclude the paper in Sect. 6.

2 Related work

The work related to this paper generally concerns two aspects, i.e., the UDA methods to handle the problem of working-condition shift and the deep learning algorithms for the RUL representation.

2.1 The UDA methods to handle the problem of working-condition shift

The UDA methods transfer the knowledge from the labeled source domain to the unlabeled target domain under the shifted working condition. These UDA methods can be roughly classified into two categories, i.e., statistics-matching-based methods and adversarial-learning-based methods. Statistics-matching-based UDA methods align feature distributions across domains by minimizing the distribution discrepancy and task risks via distribution dissimilarity/similarity metrics, such as maximum mean discrepancy (MMD). For instance, Li et al. [14] designed a shrinkage attention module to improve the sparsity of attention weights, which enables the model to pay more attention to critical features and overlook irrelevant information. Their study introduces the representation of subspace distance loss and the bases mismatch penalization loss. Jin et al. [15] constructed a domain adaptive residual network for the cross-domain RUL prediction of aircraft engines, in which multi-layer multi-kernel MMD (MKMMD) is introduced to reduce the distribution discrepancy of different domains. Likewise, He et al. [13] introduced joint MMD (JMMD) to simultaneously measure the distance of marginal and conditional probability distributions in different domains. Similar improved MMD methods have been proposed in [16–20]. However, the disadvantages of these methods lie in that they may not fully utilize the unlabeled target data about the domain-related information, which can lead to limited performance improvement in RUL prediction.

Adversarial-learning-based UDA methods learn domain-invariant representations to reduce the differences in the distribution of data from the source domain and target domain at the feature/representation level, with the guidance of a domain discriminator as an adaptively learned divergence measure [8]. Inspired by generative adversarial networks (GANs), adversarial-learning-based methods train the model to confuse the domain classifier to minimize the domain discrepancy. The feature representation model can extract discriminative and invariant features to the change of domains through implicit adversarial metric functions. Usually, the feature representations are simultaneously sent to a domain discriminator and a task classifier in the adversarial learning [8]. For instance, da Costa et al. [32] have attempted to solve the cross-domain RUL prediction problem through long short-term memory (LSTM) and adversarial training. Zhang et al. [33] assumed that the target domain

may contain multiple unknown operating modes, and then, proposed a multi-domain decoupling adversarial-learning model based on three-dimensional metrics to achieve fine adaptation to distinguishable structures in the target domain. Nevertheless, these adversarial-learning-based methods may lose useful target-related information, limiting the performance of the cross-domain RUL prediction task. To this end, Ragab et al. [28] integrated parameters-based and adversarial-learning-based methods, which captured the domain-invariant feature while preserving the target-related feature with a contrastive loss-based approach. Further, Zhuang et al. [29] considered that the local semantics of degenerate features and mutual information from the target domain may be discarded when aligning the distribution of the source domain and target domain, and they introduced contrastive learning into the adversarial-learning-based UDA network.

More recently, some researchers have attempted to integrate adversarial training-based UDA methods with other UDA methods to enhance prediction accuracy. For instance, Fu et al. [24] integrated statistical-matching-based UDA with domain adversarial learning in a deep residual LSTM network based on multi-kernel MMD to extract domain-invariant features for improving the RUL prediction performance in scenarios of working-condition shift of aircraft engines. Ye and Yu [25] proposed a selective adversarial DA model through MMD-based selective interaction and adversarial learning. Similarly, Shi et al. [26] presented a Wasserstein distance-based deep multi-scale CNN domain adversarial network for the RUL prediction of aircraft engines, which not only avoids the problem of gradient vanishing but also enhances the robustness of the adaptive RUL prediction. However, the noise or domain-invariant information can contaminate the target-related mutual information during training without constraints. Therefore, to consider the working-condition-related information (not only in the target domain but also in the source domain), we present a UDA method based on the domain-invariant feature and domain-related feature for the RUL prediction task of aircraft engines under scenarios of working-condition shift.

2.2 The deep learning algorithms for the RUL representation

Recurrent neural networks (RNNs), LSTM, and gated recurrent unit (GRU) are the most widely used deep learning algorithms for RUL representation of aircraft engines, due to their excellent capability in capturing the temporal relationship among time series of sensory data [18, 34]. Furthermore, the attention mechanism is applied to these recurrent models to improve the deep feature learning capability [35–37]. However, in these recurrent models, the condition monitoring sensory data are processed sequentially, resulting in accumulated errors and high computation costs.

The Transformer [38] employs the self-attention mechanism instead of recurrent networks, showing excellent temporal feature extraction capabilities and allowing for parallel computation. Several Transformer-based approaches have been proposed in the field of RUL prediction of aircraft engines [39–43]. However, these Transformer-based RUL prediction models have shown a limitation in

capturing the long-term dependencies among the multi-sensory data of aircraft engines. To this end, Informer [44] proposed the *ProbSparse* self-attention to enable the model to deal with long-range dependencies with less cost than Transformer. Therefore, in our study, we select the Informer—a successful variant of Transformer, to be the base deep learning network of UDATrans.

3 Problem formulation

We use $P(X_S)$ and $P(X_T)$ to denote the data distributions of source domain data $\boldsymbol{x}_{\boldsymbol{n}}^{(\boldsymbol{S})}$ and target domain data $\boldsymbol{x}_{\boldsymbol{n}}^{(\boldsymbol{T})}$, respectively. Therefore, predicting the RUL of aircraft engines under scenarios of working-condition shifts essentially addresses a UDA problem due to data distribution shifts: $P(X_S) \neq P(X_T)$. We assume the source domain $D_S = \{(\boldsymbol{x}_{\boldsymbol{n}}^{(\boldsymbol{S})}, \boldsymbol{y}_{\boldsymbol{n}}^{(\boldsymbol{S})})\}_{n=1}^{N_S}$ and the target domain $D_T = \{\boldsymbol{x}_{\boldsymbol{n}}^{(\boldsymbol{T})}\}_{n=1}^{N_T}$. N_S and N_T represent the counts of aircraft engine units in the source and target domains, respectively, with n serving as the index for these units. $\boldsymbol{x}_{\boldsymbol{n}}^{(\boldsymbol{S})}$ and $\boldsymbol{x}_{\boldsymbol{n}}^{(\boldsymbol{T})}$ represent the high-dimensional condition monitoring data from sensor signals of aircraft engine unit n, belonging to feature spaces $\mathcal{X}_S$ and $\mathcal{X}_T$ of the source (D_S) and target (D_T) domains, respectively. $\boldsymbol{y}_{\boldsymbol{n}}^{(\boldsymbol{S})}$ is the associated RUL labels for $\boldsymbol{x}_{\boldsymbol{n}}^{(\boldsymbol{S})}$. Within the source domain D_S, $\boldsymbol{x}_{\boldsymbol{n}}^{(\boldsymbol{S})} = \{\boldsymbol{x}_{\boldsymbol{n},\boldsymbol{i}}^{(\boldsymbol{S})}\}_{i=1}^{T_n^{(S)}} \in \mathbb{R}^{T_n^{(S)} \times M}$, where $T_n^{(S)}$ indicates the trajectory length of aircraft engine unit n in D_S, and M represents the count of sensor measurements in the condition monitoring system. The RUL labels for $\boldsymbol{x}_{\boldsymbol{n}}^{(\boldsymbol{S})}$ are denoted by $\boldsymbol{y}_{\boldsymbol{n}}^{(\boldsymbol{S})} = \{y_{n,i}^{(S)}\}_{i=1}^{T_n^{(S)}} \in \mathbb{R}^{T_n^{(S)}}$. $y_{n,i}^{(S)}$ is the the RUL label for $\boldsymbol{x}_{\boldsymbol{n},\boldsymbol{i}}^{(\boldsymbol{S})}$. In the target domain D_T, $\boldsymbol{x}_{\boldsymbol{n}}^{(\boldsymbol{T})} = \{\boldsymbol{x}_{\boldsymbol{n},\boldsymbol{i}}^{(\boldsymbol{T})}\}_{i=1}^{T_n^{(T)}} \in \mathbb{R}^{T_n^{(T)} \times M}$, where $T_n^{(T)}$ is the length of trajectories of aircraft engine unit n. Note that RUL labels $\boldsymbol{y}_{\boldsymbol{n}}^{(\boldsymbol{T})}$ for aircraft engine n of D_T are unavailable.

For simplification, we omit the index of aircraft engine units (n) in the following paragraph since we do not emphasize the individual differences in aircraft engines. Then, we use $\boldsymbol{x}_{\boldsymbol{i}}^{(\boldsymbol{S})}$ and $\boldsymbol{x}_{\boldsymbol{i}}^{(\boldsymbol{T})}$ to denote the condition monitoring sensor signals in the source and target domains, respectively. $y_i^{(S)}$ and $y_i^{(T)}$ are the corresponding RULs for $\boldsymbol{x}_{\boldsymbol{i}}^{(\boldsymbol{S})}$ and $\boldsymbol{x}_{\boldsymbol{i}}^{(\boldsymbol{T})}$.

The ultimate goal of this study is the RUL prediction for the target domain data. Motivated by shared-private theory [45] and disentanglement representation learning [30], we construct a cross-domain RUL prediction model based on the working-condition-invariant feature and the domain-related feature ($\boldsymbol{f}_{\boldsymbol{s},\boldsymbol{i}}$ and $\boldsymbol{f}_{\boldsymbol{p},\boldsymbol{i}}$). We use working-condition-invariant generator $\mathcal{S}$ and working-condition-related feature generator $\mathcal{P}$ to generate the $\boldsymbol{f}_{\boldsymbol{s},\boldsymbol{i}}$ and $\boldsymbol{f}_{\boldsymbol{p},\boldsymbol{i}}$, respectively. Then, our cross-domain RUL prediction model based on the working-condition-invariant feature and the domain-related feature can be denoted as follows:$\hat{y}_i^{(T)} = \mathcal{Y}(\boldsymbol{f}_{\boldsymbol{s},\boldsymbol{i}}, \boldsymbol{f}_{\boldsymbol{p},\boldsymbol{i}}) = \mathcal{Y}\Big(\mathcal{S}(\boldsymbol{x}_{\boldsymbol{i}}^{(\boldsymbol{T})}), \mathcal{P}(\boldsymbol{x}_{\boldsymbol{i}}^{(\boldsymbol{T})})\Big)$. More details of our method are given in Sect. 4. The main symbols used in this paper are summarized in Table 1.

Table 1 Main symbols and their description

Symbols	Description
n	The index of aircraft engine units
i	The index of time (cycles)
M	The total number of sensors
D_S/D_T	The source/target domain
$\boldsymbol{x_n}^{(S)}/\boldsymbol{x_n}^{(T)}$	The sensory time series the n-th aircraft engines in the source/target domain
$y_n^{(S)}/y_n^{(T)}$	The RUL labels for aircraft engine unit n in the source/target domain
N_S/N_T	The total number of the aircraft engine units in the source/target domain
$\mathcal{X}_S/\mathcal{X}_T$	The feature space of source/target data (condition monitoring sensor signals)
$\boldsymbol{x}_{\boldsymbol{n,i}}^{(\boldsymbol{S})}/\boldsymbol{x}_{\boldsymbol{n,i}}^{(\boldsymbol{T})}$	The sensory time series of aircraft engine unit n at time i in the source/target domain
$T_n^{(S)}/T_n^{(T)}$	The length of trajectories of the n-th aircraft engine in the source/target domain
$P(X_S)/P(X_T)$	The marginal distribution of source/target data
$\boldsymbol{x_i}$	The sensory time series
$\boldsymbol{x}_{\boldsymbol{i}}^{(\boldsymbol{S})}/\boldsymbol{x}_{\boldsymbol{i}}^{(\boldsymbol{T})}$	The sensory time series in the source/target domain
$\mathcal{S}/\mathcal{P}$	The working-condition-invariant/working-condition-related feature generator
$\boldsymbol{f_{s,i}}/\boldsymbol{f_{p,i}}$	The working-condition-invariant/working-condition-related feature
θ_s/θ_p	The parameters in $\mathcal{S}/\mathcal{P}$
$\mathcal{L}_{diff}$	The orthogonality loss
$\hat{\theta}_s/\hat{\theta}_p$	The optimal parameter value of θ_s/θ_p
$\mathcal{D}$	The domain discriminator
$d_i/\hat{d}_i$	The ground-truth/discriminated domain label
$\hat{d}_i^{(p)}$	The probability that the instance i is predicted to be of the source domain by $\mathcal{D}$
$\theta_d/\hat{\theta}_d$	The parameters in $\mathcal{D}$/the optimal parameter value of θ_d
$\mathcal{L}_{adv}$	The adversarial loss
$y_t/\hat{y}_t$	The ground-truth/predicted RULs
$\mathcal{Y}$	The RULs predictor
$\theta_y/\hat{\theta}_y$	The parameters in $\mathcal{Y}$/the optimal parameter value of θ_y
$\mathcal{L}_{rul}$	The RULs regression loss
$\mathcal{C}$	The domain classifier
$\hat{c}_i$	The classified domain label by $\mathcal{C}$
$\hat{c}_i^{(p)}$	The probability that the instance i is predicted to be of the source domain by $\mathcal{C}$
$\theta_c/\hat{\theta}_c$	The parameters in $\mathcal{C}$/the optimal parameter value of θ_c
$\mathcal{L}_{ce}$	The domain classification loss
$\mathcal{L}_{all}$	The overall loss of UDATrans
$\alpha_{\text{rul}}/\alpha_{\text{ce}}/\alpha_{\text{adv}}/\alpha_{\text{diff}}$	The trade-off parameters for $\mathcal{L}_{\text{rul}}/\mathcal{L}_{\text{ce}}/\mathcal{L}_{\text{adv}}/\mathcal{L}_{\text{diff}}$
y_{max}	The maximum RUL

4 Proposed method

Inspired by the shared-private latent feature spaces and disentangled representation learning [30, 45], we introduce a UDA methodology, termed UDATrans, leveraging

both the working-condition-invariant feature and the working-condition-related feature. This section details the UDATrans framework (refer to Fig. 2), which is structured around five principal components: two feature generators, a domain discriminator, an RUL predictor, and a domain classifier. Each component serves a distinct function, collaboratively enhancing the model's learning efficacy.

- *The working-condition-invariant feature generator and the working-condition-related feature generator*: Performing the generation of the working-condition-invariant (domain-invariant) feature and the working-condition-related (domain-related) feature, respectively.
- *Domain discriminator*: Implementing adversarial training with the working-condition-invariant feature generator and making the generated feature robust across different working conditions.
- *RUL predictor*: Performing the RUL prediction via the synergy of the working-condition-invariant feature and the working-condition-related feature.
- *Domain classifier*: Discerning the origin of working-condition-related features, either from the source or the target domain, thereby ensuring that the features accurately reflect distinct working conditions.

4.1 The working-condition-invariant feature generator and the working-condition-related feature generator

The sensory time series of aircraft engines hide working-conditions-related (domain-related) features and working-conditions-invariant (domain-invariant) features. Most previous studies only use the working-condition-invariant feature for the RUL prediction of aircraft engines under cross-domain scenarios. In our study, not only the working-condition-invariant feature but also the working-condition-related feature are used for the RUL representation under scenarios of working-condition shift. We introduce two separate feature generators, i.e., working-condition-invariant feature generator $\mathcal{S}$ and working-condition-related feature generator $\mathcal{P}$, in the framework of UDATrans. Generators $\mathcal{S}$ and $\mathcal{P}$ perform their respective tasks without interfering with each other. $\mathcal{S}$ is used to extract the working-condition-invariant feature f_s (i.e., the domain-related

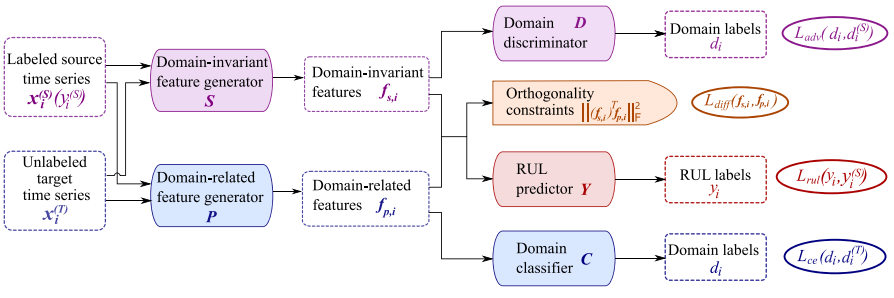

Fig. 2 The framework of the UDATrans

features). On the countray, $\mathcal{P}$ is used to capture the working-condition-related feature f_p (i.e., the domain-related features). Formally, for any sensory data $\boldsymbol{x_i} \in \{\mathcal{X}_S, \mathcal{X}_T\}$, we can get the working-condition-invariant feature $\boldsymbol{f_{s,i}}$ and the domain-related feature $\boldsymbol{f_{p,i}}$ through their feature representation $\mathcal{S}(\boldsymbol{x_i})$ and $\mathcal{P}(\boldsymbol{x_i})$, respectively. Equations 1 and 2 describe the $\mathcal{S}$ and $\mathcal{P}$, where θ_s and θ_p are the parameters of $\mathcal{S}$ and $\mathcal{P}$, respectively:

$$\boldsymbol{f_{s,i}} = \mathcal{S}(\boldsymbol{x_i};\theta_s), \boldsymbol{x_i} \in \mathcal{X}_S \cup \mathcal{X}_T, \tag{1}$$

$$\boldsymbol{f_{p,i}} = \mathcal{P}(\boldsymbol{x_i};\theta_p), \boldsymbol{x_i} \in \mathcal{X}_S \cup \mathcal{X}_T. \tag{2}$$

In most previous UDA approaches for working-condition shift problems of aircraft engines, the extracted working-condition-invariant feature is prone to be contaminated by the working-condition-related feature or the noise brought by the source and target domains, thereby suffering from feature contamination and feature redundancy. To solve this problem, we employ orthogonality constraints [45] to alleviate the interference of the feature space of the working-condition-invariant feature generator $\mathcal{S}$ and working-condition-related feature generator $\mathcal{P}$. Accordingly, our framework introduces the orthogonality loss $\mathcal{L}_{diff}$:

$$\begin{aligned} \mathcal{L}_{diff} &= \frac{1}{N_S + N_T} \sum_{i=1}^{N_S+N_T} \|\boldsymbol{f_{s,i}}^T \cdot \boldsymbol{f_{p,i}}\|_F^2 \\ &= \frac{1}{N_S + N_T} \sum_{i=1}^{N_S+N_T} \|\mathcal{S}(\boldsymbol{x_i})^T \cdot \mathcal{P}(\boldsymbol{x_i})\|_F^2, \boldsymbol{x_i} \in \mathcal{X}_S \cup \mathcal{X}_T, \end{aligned} \tag{3}$$

where $\|\cdot\|_F^2$ is the squared Frobenius norm [46]. $\mathcal{L}_{diff}$ penalizes redundant latent representations and encourages $\mathcal{S}$ and $\mathcal{P}$ to encode different aspects of the inputs $\boldsymbol{x_i}$ to optimize the parameters (θ_s and θ_p) of $\mathcal{S}$ and $\mathcal{P}$. With the introduction of orthogonality constraints, the working-condition-invariant and working-condition-related feature spaces are inherently disjoint. In this way, the extracted features $\boldsymbol{f_s}$ and $\boldsymbol{f_p}$ can be close to uncontaminated with the potential to perform the subsequent tasks. The parameters θ_s and θ_p in this stage are optimized to achieve:

$$(\hat{\theta}_s, \hat{\theta}_p) = \arg\min_{\theta_s,\theta_p} \mathcal{L}_{diff}\Big(\mathcal{S}(\boldsymbol{x_i};\theta_s), \mathcal{P}(\boldsymbol{x_i};\theta_p)\Big), \boldsymbol{x_i} \in \mathcal{X}_S \cup \mathcal{X}_T, \tag{4}$$

where $\hat{\theta}_s$ and $\hat{\theta}_p$ are optimal parameter values of θ_s and θ_p, respectively.

For the type of neural network for the working-condition-invariant and working-condition-related feature generators ($\mathcal{S}$ and $\mathcal{P}$), Transformer-based networks have been considered. We select the Informer [44], a successful variant of the Transformer [38]. The Informer can handle the high-dimensional and temporal dependence characteristics among the multi-sensor time series. Figure 3 shows the structure of the Informer-based feature generator.

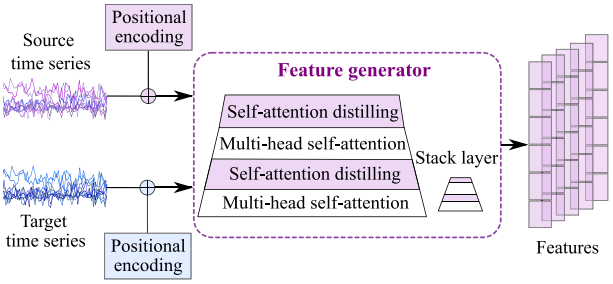

Fig. 3 The structure of the Informer-based feature generator

4.2 Domain discriminator

In recent years, adversarial learning has been widely used in prognostics since the DANN model was proposed [28, 32, 40]. The main idea in adversarial learning is similar to GANs, as the feature generator attempts to extract the features that could confuse the discriminator. In UDATrans, the working-condition-invariant feature generator $\mathcal{S}$ is working adversarial to a learnable domain discriminator $\mathcal{D}$, preventing it from accurately classifying the types of domains (source domain or target domain). The adversarial training encourages working-condition-invariant feature space to be pure and ensures the working-condition-invariant representation not be contaminated by the working-condition-related information and noise. This adversarial interaction enhances the ability of the model to generalize the working-condition-invariant features across different operational environments, making them more robust and useful for accurate RUL prediction.

The domain label of $\boldsymbol{x_i}$ is defined as follows:

$$d_i = \begin{cases} 0, & \boldsymbol{x_i} \in \mathcal{X}_S, \\ 1, & \boldsymbol{x_i} \in \mathcal{X}_T. \end{cases} \tag{5}$$

We use θ_d to denote the parameters of $\mathcal{D}$. Then, the domain discriminator $\mathcal{D}$ could be described as follows:

$$\hat{d}_i^{(p)} = \mathcal{D}(\boldsymbol{f_{s,i}};\theta_d) = \mathcal{D}\Big(\mathcal{S}(\boldsymbol{x_i};\theta_s);\theta_d\Big),\ \boldsymbol{x_i} \in \mathcal{X}_S \cup \mathcal{X}_T, \tag{6}$$

where $\hat{d}_i^{(p)}$ is the probability that the instance i is predicted to be of the source domain by $\mathcal{D}$. Then, the predicted domain label $\hat{d}_i$ is calculated as follows:

$$\hat{d}_i = \begin{cases} 0, & \hat{d}_i^{(p)} \geq 0.5, \\ 1, & \text{otherwise}. \end{cases} \tag{7}$$

We adopt the adversarial loss $\mathcal{L}_{adv}$ to prevent the working-condition-related feature from creeping into the working-condition-invariant feature space. The adversarial loss is used to train a model to produce the working-condition-invariant feature such that a classifier cannot reliably discriminate the true domain label based on these features. We define the ground-truth labels of source and target domains as 1 and 0,

respectively. The domain discriminator $\mathcal{D}$ is trained to correctly identify whether the generated features are from the source or target domains. Hence, the adversarial loss for $\mathcal{D}$ can be calculated as follows:

$$\mathcal{L}_{adv} = \min_{\mathcal{S}} \max_{\mathcal{D}} \mathbb{E}_{\boldsymbol{x_i} \sim \mathcal{X}_S}[\log \mathcal{D}(\mathcal{S}(\boldsymbol{x_i}))] + \mathbb{E}_{\boldsymbol{x_i} \sim \mathcal{X}_T}[\log(1 - \mathcal{D}(\mathcal{S}(\boldsymbol{x_i}))]. \tag{8}$$

Use $\hat{\theta}_d$ to denote the optimal value of θ_d. Then, the parameters in this stage are optimized to achieve:

$$\begin{aligned}(\hat{\theta}_s, \hat{\theta}_d) = \arg \Bigg\{ & \max_{\theta_s, \theta_d} \mathcal{L}_{\text{adv}} \Big(\mathcal{D}\Big(\mathcal{S}(\boldsymbol{x_i}; \theta_s); \theta_d \Big), d_i \Big), \ \boldsymbol{x_i} \in \mathcal{X}_S, \\ & \min_{\theta_s, \theta_d} \mathcal{L}_{\text{adv}} \Big(\mathcal{D}\Big(\mathcal{S}(\boldsymbol{x_i}; \theta_s); \theta_d \Big), d_i \Big), \ \boldsymbol{x_i} \in \mathcal{X}_T \Bigg\}.\end{aligned} \tag{9}$$

The gradient reversal layer (GRL) [45] is adopted into the adversarial training architecture, and the parameters can be simultaneously updated in one step during training with back-propagation training and min–max optimization. During the back-propagation process, the GRL is inserted between the domain discriminator $\mathcal{D}$ and the working-condition-invariant feature extractors $\mathcal{S}$ to make the reverse training targets of $\mathcal{D}$ and $\mathcal{S}$, i.e., $\mathcal{S}$ is optimized to extract working-condition-invariant feature (the working-condition-invariant feature of source and target domain: $\boldsymbol{f_{s,i}}$) from the sensory data of two domains, but $\mathcal{D}$ is optimized to discriminate as much as possible their domain of membership. GRL can be implemented by multiplying the gradient by -1 without introducing other hyper-parameters, as shown in Fig. 4. When the discriminator can no longer discriminate the exact domain of a sample, the working-condition-invariant feature space aligning the two distributions is considered to be successfully constructed. This means that we use $-\alpha_{\text{adv}} \frac{\partial \mathcal{L}_{\text{adv}}}{\partial \theta_s}$ for updating θ_s in $\mathcal{S}$ and $\alpha_{\text{adv}} \frac{\partial \mathcal{L}_{\text{adv}}}{\partial \theta_d}$ for updating θ_d in $\mathcal{D}$.

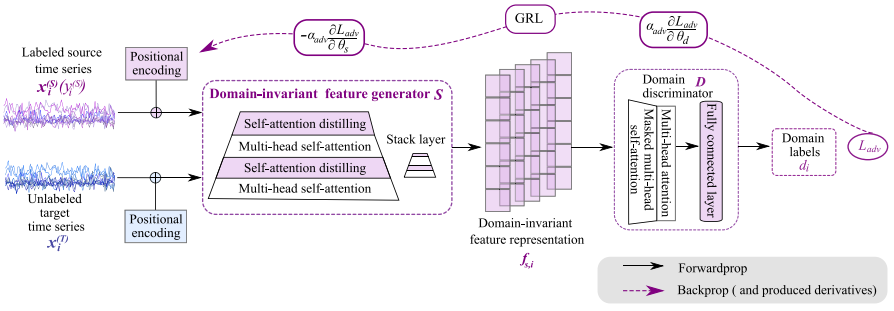

Fig. 4 The adversarial learning in our proposed UDATrans

4.3 RUL predictor

We introduce a predictor after the working-condition-invariant and working-condition-related feature generators to predict the RUL:

$$\hat{y}_i = \mathcal{Y}\Big(\mathcal{S}(\boldsymbol{x_i};\theta_s), \mathcal{P}(\boldsymbol{x_i};\theta_p);\theta_y\Big), \boldsymbol{x_i} \in \mathcal{X}_S. \tag{10}$$

The parameters of the predictor $\mathcal{Y}$ are trained to minimize the RUL prediction errors of the predicted and true RULs. As often done, we adopt the mean square error (MSE) to construct the loss $\mathcal{L}_{rul}$ related to RUL prediction errors:

$$\mathcal{L}_{\text{rul}} = \frac{1}{N_S}\sum_{i=1}^{N_S}(y_i - \hat{y}_i)^2 = \frac{1}{N_S}\sum_{i=1}^{N_S}\left(y_i - \mathcal{Y}\Big(\mathcal{S}(\boldsymbol{x_i}), \mathcal{P}(\boldsymbol{x_i})\Big)\right)^2, \boldsymbol{x_i} \in \mathcal{X}_S. \tag{11}$$

Hence, the parameters in this stage are optimized to achieve:

$$(\hat{\theta}_y, \hat{\theta}_s, \hat{\theta}_p) = \arg\min_{\theta_y,\theta_s,\theta_p} \mathcal{L}_{\text{rul}}\left(\mathcal{Y}\Big(\mathcal{S}(\boldsymbol{x_i};\theta_s), \mathcal{P}(\boldsymbol{x_i};\theta_p);\theta_y\Big), y_i\right), \boldsymbol{x_i} \in \mathcal{X}_S, \tag{12}$$

where $\hat{\theta}_y$ is the optimal parameter value of θ_y.

4.4 Domain classifier

The primary task of the domain classifier $\mathcal{C}$ is to utilize the working-condition-related features extracted by the working-condition-related feature generator $\mathcal{P}$ to distinguish whether these features originate from the source or the target domain. This task is critical for guiding the learning of working-condition-related features and ensuring that these features effectively capture domain-specific variations. The domain classifier complements the orthogonality constraints and the domain discriminator by focusing specifically on working-condition-related features, thereby creating a balanced feature representation space. This ensures that the features accurately reflect the unique aspects of varying working conditions, thereby enhancing the model's ability to generalize across distinct working conditions. Additionally, by working in tandem with the orthogonality constraints and the domain discriminator, the domain classifier supports the effective separation and utilization of working-condition-invariant and working-condition-related features. This separation is crucial for improving model adaptability and performance under complex cross-working-condition scenarios.

It should be noted that the accuracy of the domain classifier is not the primary focus of its design. Instead, its role lies in guiding the working-condition-related feature generator $\mathcal{P}$ to extract meaningful domain-specific features. Even if the classification accuracy is suboptimal, the extracted features can still contribute significantly to the overall model performance by accurately capturing working-condition-related variations.

Hereby, the functionality of the domain classifier $\mathcal{C}$ is formulated as follows:

$$\hat{c}_i^{(p)} = \mathcal{C}\Big(\mathcal{P}(\boldsymbol{x_i};\theta_p);\theta_c\Big), \boldsymbol{x_i} \in \mathcal{X}_S \cup \mathcal{X}_T, \tag{13}$$

where $\hat{c}_i^{(p)}$ is the probability that the instance i is predicted to be of the source domain by $\mathcal{C}$, and θ_c depicts the parameters of the classifier $\mathcal{C}$. Then, the predicted domain label $\hat{c}_i$ is calculated as follows:

$$\hat{c}_i = \begin{cases} 0, & \hat{c}_i^{(p)} \geq 0.5, \\ 1, & \text{otherwise}. \end{cases} \tag{14}$$

We employ cross-entropy loss to refine the feature extraction process of the domain classifier $\mathcal{C}$. Unlike adversarial loss used in the domain discriminator, which aims to make working-condition-invariant features indistinguishable across domains, cross-entropy loss in the domain classifier ensures that the working-condition-related features remain distinct and representative of their respective domains. This design choice allows the domain classifier to directly enhance the feature generator $\mathcal{P}$, leading to more robust and domain-specific feature representations. We use $\mathcal{L}_{ce}$ to denote the domain classification loss of $\mathcal{C}$:

$$\mathcal{L}_{ce} = \frac{1}{N_S + N_T} \sum_{i=1}^{N_S+N_T} -\left[d_i \log\left(\mathcal{C}\Big(\mathcal{P}(\boldsymbol{x_i})\Big)\right) + (1-d_i)\log\left(1-\mathcal{C}\Big(\mathcal{P}(\boldsymbol{x_i})\Big)\right)\right]. \tag{15}$$

Hence, the parameters in this stage are optimized to achieve:

$$(\hat{\theta}_p, \hat{\theta}_c) = \arg\min_{\theta_p,\theta_c} \mathcal{L}_{ce}\left(\mathcal{C}\Big(\mathcal{P}(\boldsymbol{x_i};\theta_p);\theta_c\Big), d_i\right), \boldsymbol{x_i} \in \mathcal{X}_S \cup \mathcal{X}_T, \tag{16}$$

where $\hat{\theta}_c$ is the optimal parameter value of θ_c.

4.5 General network optimization

Combining the orthogonality loss $\mathcal{L}_{diff}$ (Eq. 3), the adversarial loss $\mathcal{L}_{adv}$ (Eq. 8), the loss of RUL prediction errors $\mathcal{L}_{rul}$ (Eq. 11) and the domain classification loss $\mathcal{L}_{ce}$ (Eq. 15) can get the total loss of UDATrans $\mathcal{L}_{all}$:

$$\mathcal{L}_{all} = \alpha_{diff}\mathcal{L}_{\text{diff}} + \alpha_{\text{adv}}\mathcal{L}_{adv} + \alpha_{\text{rul}}\mathcal{L}_{\text{rul}} + \alpha_{ce}\mathcal{L}_{ce}, \tag{17}$$

where α_{diff}, α_{adv}, α_{rul} and α_{ce} are the trade-off parameters with values greater than zero.

Therefore, the loss of the UDATrans can be represented as follows:

$$\begin{aligned} \mathcal{L}_{all}(\theta_s, \theta_p, \theta_d, \theta_y, \theta_c) = {} & \alpha_{\text{diff}}\mathcal{L}_{\text{diff}}(\theta_s, \theta_p) + \alpha_{\text{adv}}\mathcal{L}_{\text{adv}}(\theta_s, \theta_d) \\ & + \alpha_{\text{rul}}\mathcal{L}_{\text{rul}}(\theta_s, \theta_p, \theta_y) + \alpha_{ce}\mathcal{L}_{ce}(\theta_p, \theta_c). \end{aligned} \tag{18}$$

In summary, Eqs. 4, 9, 12 and 15 can be integrated in this stage, and by using the GRL, the network parameters can be updated in one training step as follows:

$$
\begin{aligned}
\theta_s &\leftarrow \theta_s - \eta f_{\text{Adam},s}(\alpha_{\text{diff}}\frac{\partial \mathcal{L}_{\text{diff}}}{\partial \theta_s} - \alpha_{\text{adv}}\frac{\partial \mathcal{L}_{\text{adv}}}{\partial \theta_s} + \alpha_{\text{rul}}\frac{\partial \mathcal{L}_{\text{rul}}}{\partial \theta_s}),\\
\theta_p &\leftarrow \theta_p - \eta f_{\text{Adam},p}(\alpha_{\text{diff}}\frac{\partial \mathcal{L}_{\text{diff}}}{\partial \theta_p} + \alpha_{\text{rul}}\frac{\partial \mathcal{L}_{\text{rul}}}{\partial \theta_p} + \alpha_{ce}\frac{\partial \mathcal{L}_{ce}}{\partial \theta_p}),\\
\theta_d &\leftarrow \theta_d - \eta f_{\text{Adam},d}(\alpha_{\text{adv}}\frac{\partial \mathcal{L}_{\text{adv}}}{\partial \theta_d}),\\
\theta_y &\leftarrow \theta_y - \eta f_{\text{Adam},y}(\alpha_{\text{rul}}\frac{\partial \mathcal{L}_{\text{rul}}}{\partial \theta_y}),\\
\theta_c &\leftarrow \theta_c - \eta f_{\text{Adam},c}(\alpha_{ce}\frac{\partial \mathcal{L}_{ce}}{\partial \theta_c})
\end{aligned}
\tag{19}
$$

where η is the learning rate. $f_{\text{Adam},s}$, $f_{\text{Adam},p}$, $f_{\text{Adam},d}$, $f_{\text{Adam},y}$ and $f_{\text{Adam},c}$ denote the computation functions in the Adam optimization package [47] for the parameters θ_s, θ_p, θ_d, θ_y and θ_c, respectively. α_{diff}, α_{adv}, α_{rul} and α_{ce} denote the penalty coefficients for the losses $\mathcal{L}_{\text{diff}}$, $\mathcal{L}_{\text{adv}}$, $\mathcal{L}_{\text{rul}}$ and $\mathcal{L}_{ce}$, respectively.

5 Experiments

5.1 Datasets description

The simulated turbofan engine data C-MAPSS [48] is widely used as a benchmark in the PHM study of the aircraft engine. A commercial aircraft gas turbine engine is the object of the C-MAPSS dataset, and Fig. 5 shows a schematic of this kind of aircraft engine and the sensors in the turbofan. Table 2 describes the C-MAPSS data in detail. In our study, the fault mode and the operating condition jointly describe the working condition of the aircraft engine; thus, the working condition of four sub-datasets in C-MAPSS varies from each other. Here, we use FD001→FD003 to depict that the source working condition is FD001 and the target working condition is FD003, i.e., the working-condition shift scenario of that from FD001 to FD003. In this scenario, the condition monitoring data in FD001 are considered the source domain, whereas the condition monitoring data in FD003 are considered the

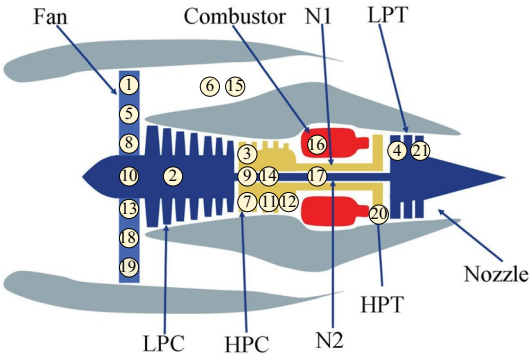

Fig. 5 The sensors in the turbofan of C-MAPSS [48]

Table 2 Description of C-MAPSS [48]

Dataset	FD001	FD002	FD003	FD004
Training trajectories	100	260	100	249
Test trajectories	100	259	100	248
Maximum lifespan (cycles)	362	378	525	543
Average lifespan (cycles)	206	206	247	245
Minimum lifespan (cycles)	128	128	145	128
Operating conditions	1	6	1	6
Fault modes	1	1	2	2

target domain. In this way, we construct 12 working-condition shift scenarios by the C-MAPSS dataset. The piece-wise linear degradation is used to label the RUL values, and the maximum RUL y_{max} is set to be 125 cycles [49, 50].

Figure 6 illustrates the data distribution of ten different sensor measurements in the sub-datasets of C-MAPSS. The sensor signal values shown in this figure are standardized using z-score normalization to transform the multi-sensor signals into a dimensionless standardized distribution. This process ensures that differences in the magnitudes and units of the sensor signals do not affect the model's ability to learn features across different sensors. The distribution divergence of different sub-datasets caused by different working conditions is the reason for the deterioration performance of the deep learning models for the cross-working-condition RUL predictions. Considering the focus on the distribution characteristics rather than the physical units of the signals, Fig. 6 presents the standardized signal distributions to reflect the relative changes and characteristics of each sensor, which are critical for improving the model's generalization ability under varying working conditions. The model trained in one dataset often does not perform well in a different dataset; thereby, we must tackle the data shift problem in the RUL predictions of aircraft engines under the working-condition shift scenarios.

From Fig. 6, we can observe that, (1) the data distribution of most of the sensor measurements is affected by the operating condition, even though some sensor

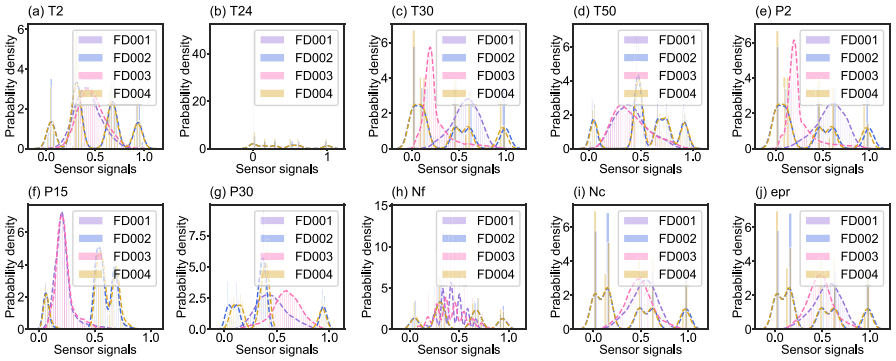

Fig. 6 The condition monitoring data distribution of four sub-datasets in C-MAPSS

measurements, such as sensor T30 and sensor NRc, are hardly affected. (2) The condition monitoring data in FD001 and FD003 are subject to similar distributions in most cases. Likewise, condition monitoring data in FD002 and FD004 show a similar divergence among the distributions for most sensor measurements. On the other hand, the data distribution of FD001/FD003 and FD002/FD004 is totally inconsistent due to the operating condition shift. (3) Influenced by the time-varying operating conditions, one can observe multiple bell curves for the distribution of condition monitoring data of FD002 and FD004. (4) Compared with the bell curves in sensor T2 (Fig. 6a) and sensor T30 (Fig. 6c), the data distribution in sensor T24 (Fig. 6c) is interesting since there are no bell curves. This indicates that sensor T24 does not have the potential capability to describe the degradation of the aircraft engine. On this basis, this kind of sensor measurement is excluded from the training process of the RUL prediction model. Concretely, 14 sensor measurements, i.e., T24, T30, T50, P30, Nf, Nc, Ps30, Phi, NRf, NRc, BPR, htBleed, W31, and W32 are considered in the following experiment.

5.2 Experimental setup

We conducted experiments on a computer with NVIDIA TITAN XP GPUs via Python 3.8 with Pytorch 1.9.0. Each experiment was repeated ten times, with average results reported.

Performance metrics: Consistent with prior studies, this research employs the root-mean-square error (RMSE) (Eq. 21) and the scoring function (Eq. 20) [48] for evaluating cross-domain RUL prediction performance. We use e_i to denote the prediction error between the predicted RUL value $\hat{y}_i$ and the corresponding ground truth y_i.

$$\text{Score} = \begin{cases} \sum_{i=1}^{N_T} (\exp(-\frac{e_i}{13}) - 1), & e_i \leq 0, \\ \sum_{i=1}^{N_T} (\exp(\frac{e_i}{10}) - 1), & \text{otherwise} \end{cases} \tag{20}$$

$$\text{RMSE} = \sqrt{\frac{1}{N_T} \sum_{i=1}^{N_T} (e_i)^2} \tag{21}$$

Parameter setting details: For the feature generators $\mathcal{S}$ and $\mathcal{P}$, the Informer [44], a successful variant of Transformer [38], was chosen. The Informer is capable of handling the high-dimensional and temporal dependence characteristics among the condition monitoring time series. There are a three-layer stack and a one-layer stack (1/4 input) in the two feature generators (encoders) $\mathcal{S}$ and $\mathcal{P}$. The predictor $\mathcal{Y}$ comprises a two-layer stack decoder and three fully connected layers (FCLs) with 128, 64, and 1 neurons in the hidden layers, respectively. Likewise, both the domain discriminator $\mathcal{D}$ and domain classifier $\mathcal{C}$ are constructed by three FCLs with the same configuration of the predictor $\mathcal{Y}$. The time window size is 36, and the stride step is 1. The learning rate η starts from 0.0005, decaying two times every epoch. The batch

size is 64. The input data are standard normalized. The dropout rate is set to 0.05. The parameter settings of α_{rul}, α_{ce}, α_{adv} and α_{diff} are shown in Table 3.

5.3 Performance results

Figure 7 shows the predicted RUL results of the target dataset of each cross-domain task. Besides, instances are rearranged in descending order by RUL values to observe the estimation results better. We can observe that: (1) In general, the predicted RULs closely follow the ground truth, indicating the excellent performance of the UDATrans in scenarios of the working-condition shift. (2) The proposed UDATrans demonstrates particularly excellent performance when the RUL of aircraft engines is short, attributed to distinct degradation features that depict impending failure. (3) Most cross-domain scenarios yield early predictions during the aircraft engine's health stage (when $y_i = 125$), in contrast with the irregular early and late predictions after the health stage. This is because the degradation characteristics are not remarkable in the aircraft engines' health stage. Introducing the state of health assessment task or first prediction time determination into the UDA model may alleviate this situation. (4) As the divergence between the working conditions of the source and target domains increases, the prediction performance deteriorates. For instance, the performance of UDATrans of FD001→FD004 (1 OC & 1 FM→6 OCs & 2 FMs) is inferior to FD001→FD002 (1 OC & 1 FM →6 OCs & 1 FM). Likewise, the performance of UDATrans of FD004→FD001 (6 OCs & 2 FMs → 1 OC & 1 FM) is inferior to FD002→FD001 (6 OCs & 1 FM→1 OC & 1 FM). (5) Across the 12 working-condition shift scenarios, fault-mode shifts pose a greater challenge than operating condition shifts. For example, the performance of UDATrans of the fault-mode shift scenario FD003→FD001 (1 OC & 2 FMs → 1 OC & 1FM) is inferior to the operating condition shift scenario FD002→FD001 (6 OCs & 1 FM → 1 OC & 1 FM). Likewise, the performance of UDATrans of FD002→FD004 (6 OCs & 1 FM→ 6 OCs & 2 FMs) is inferior to FD003→FD004 (1 OC & 2 FMs → 6 OCs & 2 FMs).

Table 3 Parameter settings of α_{rul}, α_{ce}, α_{adv} and α_{diff}

Case	α_{rul}	α_{ce}	α_{adv}	α_{diff}
FD001→FD002	0.001	2	6	0.1
FD001→FD003	0.001	0.5	0.3	0.2
FD001→FD004	0.001	1	3	0.05
FD002→FD001	0.001	0.5	1	0.2
FD002→FD003	0.001	0.02	1	0.02
FD002→FD004	0.001	2	3	1
FD003→FD001	0.001	2	6	0.5
FD003→FD002	0.001	2	6	0.1
FD003→FD004	0.001	0.5	6	1
FD004→FD001	0.001	2	4	1
FD004→FD002	0.001	2	3	1
FD004→FD003	0.001	2	3	1

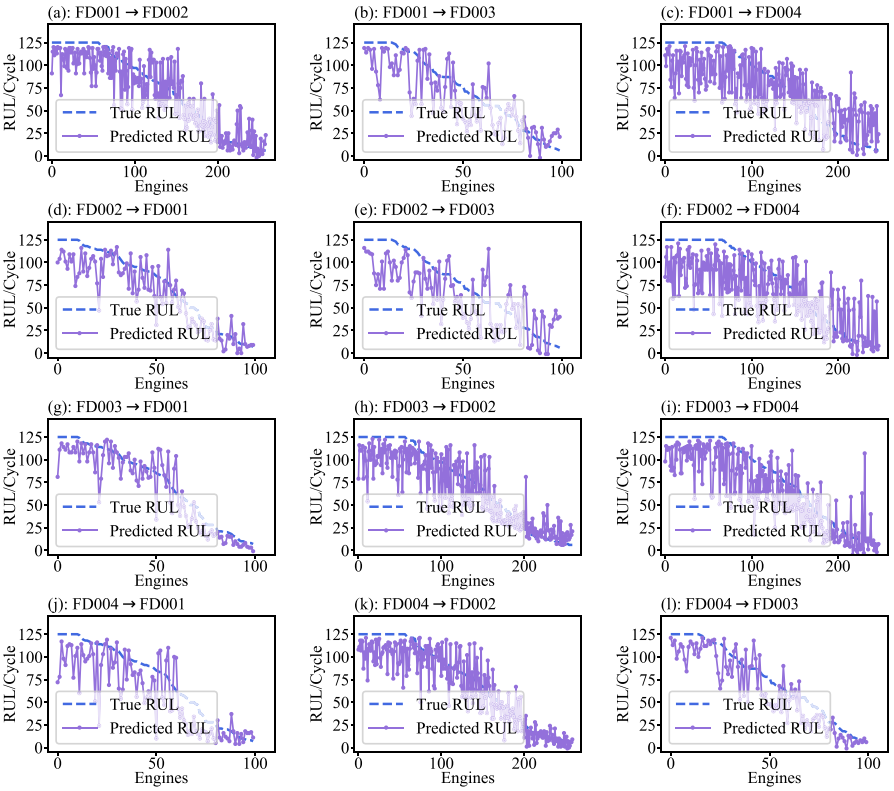

Fig. 7 The ground truth (dashed-blue) and predicted RULs (solid-violet)

5.4 Comparison performance

In this section, the UDATrans is compared with five state-of-the-art UDA methods to show the excellent cross-domain prognostic performance, which are DAN (domain adaptive residual network) [15], JAN (joint distribution adaptation network with adversarial learning) [16], DDC (bidirectional-GRU transfer learning network) [51], DANN (deep adversarial neural networks for data alignments) [52], CADA (contrastive adversarial domain adaptation) [28], DIDRLSTM (deep residual LSTM network) [24] and ADDA (cross-domain adversarial network) [53], respectively. Each experiment runs 10 times and takes the average result as the final result. Tables 4 and 5 summarize the results of our approach UDATrans alongside results from seven state-of-the-art methods that appeared over the last 3 years. The best results are highlighted in boldface, and the second-best results are underlined. The different performances of UDATrans on RMSE and Score are due to the imbalance penalties for delayed and advanced forecasting by the Score function (Eq. 20).

From Tables 4 and 5, we can conclude that: **(1)** The UDATrans achieves the best performance (21.72) on RMSE by decreasing 9.3% on average, compared with the

Table 4 RUL prediction performance comparison on RMSE of UDATrans with seven state-of-the-art methods appeared in the last 3 years

Source	Target	DAN [15]	JAN [16]	DDC [51]	DANN [52]	CADA [28]	DIDRLSTM [24]	ADDA [53]	UDATrans
FD001	FD002	35.9	35.7	32.6	32.4	19.52	29.4	49.60	**19.33**
	FD003	24.4	25.4	21.5	23.7	39.58	**20.9**	35.95	24.77
	FD004	38.7	38.8	36.8	36.6	31.23	34.4	50.76	**28.69**
FD002	FD001	29.9	20.1	20.3	21.6	**13.88**	16.1	36.71	16.06
	FD003	24.8	23.7	22.1	24.7	33.53	**21.1**	40.03	23.50
	FD004	37.3	38	37.1	37.5	33.71	34	31.68	**30.66**
FD003	FD001	19.2	23.5	22.5	23.9	19.54	18.8	24.96	**18.07**
	FD002	34.9	35.4	33.2	34.4	**19.33**	29.4	44.17	22.31
	FD004	33.8	33	32.7	33.1	**20.61**	32.3	44.34	24.86
FD004	FD001	23.8	24.8	22.9	22.3	20.1	21.3	27.72	**18.04**
	FD002	31.4	32.2	31.2	32.3	18.5	28.9	23.01	**18.11**
	FD003	19	19.8	19.7	19.7	**14.49**	16.9	32.41	16.29
Average		29.43	29.2	27.72	28.52	23.95	25.29	36.79	**21.72**

Table 5 RUL prediction performance comparison on Score of UDATrans with seven state-of-the-art methods appeared in the last 3 years

Source	Target	DAN [15]	JAN [16]	DDC [51]	DANN [52]	CADA [28]	DIDRLSTM [24]	ADDA [53]	UDATrans
FD001	FD002	11,675	15,684	7293	6764	2122	4822	54,009	**1913**
	FD003	650	648	**457**	656	8415	470	4210	1156
	FD004	31,083	11,865	8124	8240	11,577	**5538**	44,055	6645
FD002	FD001	412	573	483	531	351	**213**	12,272	412
	FD003	653	567	**383**	684	5213	474	7374	1312
	FD004	8977	10,204	7034	7759	15,106	**5471**	19,086	7663
FD003	FD001	737	628	813	777	1451	**442**	4644	624
	FD002	14,969	8287	6988	8996	5257	**4127**	33,101	5503
	FD004	5451	4681	4259	4497	**3219**	4123	25,577	5230
FD004	FD001	754	916	784	615	840	**553**	38,800	764
	FD002	4787	5736	5485	5814	4460	3483	8588	**1310**
	FD003	323	472	466	436	682	**249**	10,711	584
Average		6706	5022	3547	3814	4891	**2497**	21,868	2760

second-best approach CADA (23.95) [28]. **(2)** The UDATrans gains great performance on RMSE in the working-condition shift scenarios where the source datasets are FD001 (the simplest working condition: 1 OC & 1 FM) and FD004 (the most complex working condition: 6 OCs & 2 FMs), respectively. In other scenarios, the performance of UDATrans is also competitive. This demonstrates the good generalization ability of UDATrans. **(3)** The fault-mode shift problem is more challenging than the operating condition shift problem in the majority of working-condition shift scenarios. For example, the performance of FD003→FD001 (1 OC & 2 FMs → 1 OC & 1 FM) is inferior to that of FD002→FD001 (6 OCs & 1 FM → 1 OC & 1 FM), 18.07 vs 16.06 on RMSE and 624 vs 412 on Score. FM and OC are the fault mode and operating condition for short, respectively. Likewise, FD002→FD004 (6 OCs & 1 FM → 6 OCs & 2 FMs) vs FD003→FD004 (1 OC & 2 FMs → 6 OCs & 2 FMs) is 30.66 vs 24.86 on RMSE and 7663 vs 5230 on Score; FD003→FD002 (1 OC & 2 FMs → 6 OCs & 1 FM) vs FD001→FD002 (1 OC & 1 FM → 6 OCs & 1 FM) is 22.31 vs 19.33 on RMSE and 5503 vs 1913 on Score.

To provide a comprehensive analysis, it is necessary to discuss the limitations of the compared methods alongside those of UDATrans. Traditional UDA methods such as DANN and ADDA focus primarily on domain-invariant feature learning, which limits their effectiveness in scenarios involving complex fault-mode shifts, such as FD003→FD001 and FD003→FD004. Similarly, methods like CADA perform relatively well under operating condition shifts but struggle when both operating conditions and fault modes vary significantly, as observed in FD002→FD004. This is likely due to their lack of explicit modeling for working-condition-related features, which are crucial for capturing domain-specific variations. Additionally, models such as DIDRLSTM and DDC lack mechanisms to disentangle working-condition-invariant and working-condition-related features, leading to feature interference and degraded performance in multi-faceted shift scenarios.

5.5 Parameter sensitivity analysis

We perform the parameter sensitivity analysis on key hyper-parameters α_{rul}, α_{ce}, α_{adv} and α_{diff} in the total loss $\mathcal{L}_{\text{all}}$ (Eq. 18) of the UDATrans. These hyper-parameters determine the trade-offs among orthogonality loss, adversarial loss, RUL prediction loss and domain classification loss, in the optimization procedure of UDATrans. To tune these parameters, our current approach primarily relies on an empirical method augmented by trial and error. This process involves iteratively adjusting parameters based on model performance feedback from validation datasets. Although this method enabled us to navigate through parameter settings effectively within a constrained experimental scope, it is inherently limited by not being systematic, potentially missing more optimal settings across the comprehensive hyper-parameter space.

We take the scenarios FD003→FD004 to explore the sensitivity of these four hyper-parameters of UDATrans. Since the four losses controlled by these hyper-parameters are of different magnitudes, there is no comparability between the values of the four hyper-parameters. During this analysis, each parameter was varied one at

a time while the other three parameters were held constant at their optimal values to isolate the effect of each parameter on the model's performance.

The results of the sensitive study for the key hyper-parameters α_{diff}, α_{adv}, α_{rul} and α_{ce} are illustrated in Fig. 8. We can observe that: (1) The four key hyper-parameters greatly impact the prediction performance, and the best RUL prediction performance is yielded when $\alpha_{rul} = 0.001$, $\alpha_{ce} = 0.5$, $\alpha_{\text{adv}} = 6$ and $\alpha_{\text{diff}} = 1$. (2) Blindly increasing the regression loss weight of the RUL prediction task (α_{rul}) does not produce the desired prediction results. Conversely, when the α_{rul} is relatively large, the training process will be significantly dominated by the RUL prediction task of the source domain, resulting in impaired RUL prediction performance on the target domain.

5.6 Ablation study

We conducted an ablation experiment to study the contribution of each block or strategy in UDATrans, including the orthogonality constraints $\mathcal{L}_{\text{diff}}$, the working-condition-related feature generator $\mathcal{P}$, the domain discriminator $\mathcal{D}$ and the domain classifier $\mathcal{C}$. Figure 9 summarizes the performance of each case.

- Case 1: UDATrans without orthogonality constraints $\mathcal{L}_{\text{diff}}$.
- Case 2: UDATrans without the domain discriminator $\mathcal{D}$ and corresponding without the adversarial loss $\mathcal{L}_{\text{adv}}$.
- Case 3: UDATrans without the domain classifier $\mathcal{C}$.
- Case 4: UDATrans without the working-condition-related feature generator $\mathcal{P}$ and corresponding without the domain classifier $\mathcal{C}$ and orthogonality constraints $\mathcal{L}_{\text{diff}}$. This case can be considered as a standard adversarial-learning-based UDA model with only the working-condition-invariant feature, i.e., the DANN.

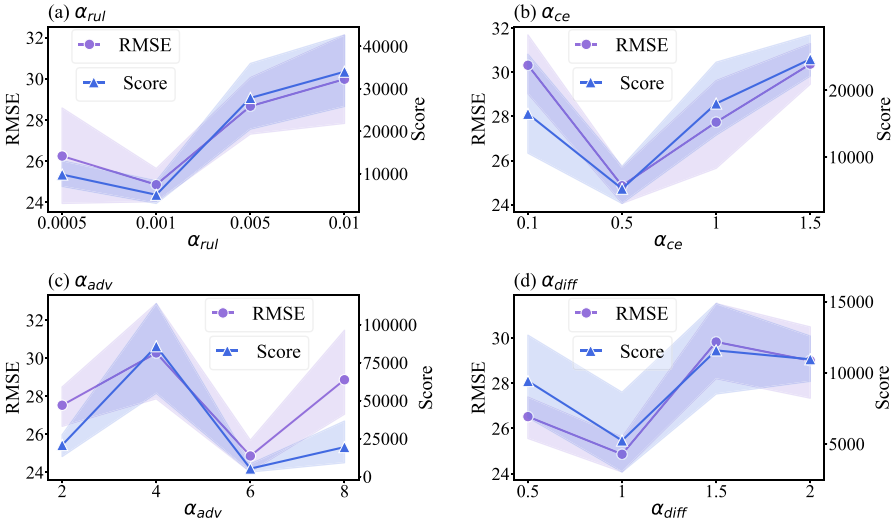

Fig. 8 The results of the sensitive study for the key hyper-parameters α_{rul}, α_{ce}, α_{adv} and α_{diff}

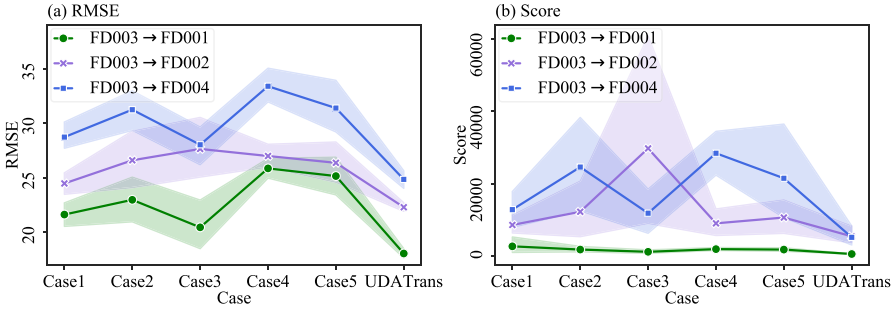

Fig. 9 The performance of the ablation study

- Case 5: Case 4 without domain discriminator $\mathcal{D}$ and the adversarial loss $\mathcal{L}_{\text{adv}}$. In essence, Case 5 can be considered as a Transformer-based RUL prediction model under working-condition shift scenarios with no domain adaption strategy and ignoring the data shift problem between the training dataset and the target dataset.

From Fig. 9, we can observe that (1) each block or strategy contributes to the improvement of RUL prediction performance under scenarios of working-condition shift. These findings underscore the importance of integrating various feature processing strategies to enhance predictive accuracy in complex environments. (2) Among all the five cases, Case 4 has yielded the highest RMSE in most working-condition shift scenarios, demonstrating the efficiency of the working-condition-invariant feature and the working-condition-related feature-based UDA methods. The working-condition-related feature generator can capture the feature-related different working conditions, whereas the domain classifier ($\mathcal{C}$) can encourage this learning process. The potential contribution of the working-condition-related feature in the cross-domain RUL prediction task is verified through the comparisons of Case 4 with other cases and UDATrans. **(3)** It should be noted that the performance of Case 3 is inconsistent among all the working-condition shift scenarios, with competitive performance in FD003→FD001 and FD003→FD004 and the most inferior performance in FD003→FD002. This inconsistency highlights the critical role of the domain classifier $\mathcal{C}$ in managing mixed working-condition shifts, which involve both fault-mode shifts and operating condition variations. **(4)** While the performance of Case 5 is acceptable in some scenarios despite not utilizing specific strategies, it is important to note that this does not suggest that a RUL prediction model trained on the source domain can be directly applied to the target domain without adjustments, as the performance stability of this case varies. **(5)** Although the contributions of the orthogonality constraints ($\mathcal{L}_{\text{diff}}$) and the domain discriminator ($\mathcal{D}$) are relatively smaller, they remain crucial for enhancing the overall balance of the model and preventing overfitting, thereby improving the RUL prediction performance.

6 Conclusion

In this work, we study the RUL prediction of aircraft engines in scenarios of working-condition shifts using a Transformer-based unsupervised domain adaptation model, UDATrans. Unlike the previous studies that rely solely on the working-condition-invariant features, our approach utilizes both the working-condition-invariant and working-condition-related features for cross-domain RUL prediction. In UDATrans, we independently extract the working-condition-invariant and working-condition-related features, further investigating their interaction mechanisms for cross-domain RUL prediction. The working-condition-invariant feature effectively captures the general relationship between degradation features and RUL, while the working-condition-related feature assesses the impact of varying working conditions on engine degradation. We also introduce orthogonality constraints to reduce feature space interference between the two feature generators. This, along with the domain discriminator and classifier, improves the disentanglement of the working-condition-invariant and working-condition-related features, thus enabling the RUL predictor to better utilize these two features for RUL predictions. Our experiments on the C-MAPSS dataset demonstrate superior generalization and robustness of the UDATrans, achieving a 24.32% RMSE reduction over state-of-the-art methods and a 9.3% improvement over the next best approach.

Future work will focus on exploring data distribution shifts, such as divergences in conditional probability distributions across different entities, and discuss systematic approaches to hyper-parameter optimization to enhance the robustness and adaptability of unsupervised domain adaptation models. Additionally, we will consider extending the research to out-of-domain scenarios, evaluating the effectiveness of domain adaptation techniques in environments not covered by the initial training data.

Acknowledgements This work was supported by the National Natural Science Foundation of China (No. 52405220 & No.52035002), the Basic Research Expenses of Central Universities—Science and Technology Innovation Project (No. 2682024CX066), the National Key R&D Program of China (No. 2022YFB3303600), the Fundamental Research Funds for the Central Universities (No. 2022CDJKYJH048) and the Aeronautical Science Foundation of China (No. 2022Z071020002).

Author contributions Zhiyao Zhang helped in conceptualization, methodology, software, validation, investigation and writing—original draft. Jiting Cheng helped in conceptualization, methodology and writing—review & editing. Pengpeng Chen helped in conceptualization, methodology and writing—review & editing. Shuang Gao helped in methodology and writing—review & editing. Xiaohui Chen helped in funding acquisition, methodology, supervision and formal analysis. Enrico Zio worked in supervision, methodology and writing—review & editing.

Data availability Data are openly available in a public repository at https://drive.google.com/drive/folders/1x8z8sJ7Gsxh1YFn7a7RtVnip406Qg8jF.

Declarations

Conflict of interest The authors declare no conflict of interest.

References

1. Yang J, Tang S, Fang P, Wang F, Sun X, Si X (2024) Remaining useful life prediction of implicit linear wiener degradation process based on multi-source information. Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability 238(1):93–111. https://doi.org/10.1177/1748006X22113260
2. Dash BM, Prakash O, Samantaray AK (2023) Failure prognosis of the components with unlike degradation trends: A data-driven approach. Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability 237(6):1132–1149. https://doi.org/10.1177/1748006X22111930
3. Lima AL, Aranha VM, Nascimento EGS (2023) Predictive maintenance applied to mission critical supercomputing environments: remaining useful life estimation of a hydraulic cooling system using deep learning. J Supercomput 79:4660–4684. https://doi.org/10.1007/s11227-022-04833-5
4. Wen X, Chung S-H, Ma H-L, Khan WA (2024) Airline crew scheduling with sustainability enhancement by data analytics under circular economy. Ann Operations Res 342(1):959–985. https://doi.org/10.1007/s10479-023-05312-7
5. Khan WA, Chung S-H, Eltoukhy AE, Khurshid F (2024) A novel parallel series data-driven model for IATA-coded flight delays prediction and features analysis. J Air Transp Manag 114:102488. https://doi.org/10.1016/j.jairtraman.2023.102488
6. Zhang Z, Chen P, Xing C, Liu B, Wang R, Li L, Chen X, Zio E (2023) A data augmentation boosted dual informer framework for the performance degradation prediction of aero-engines. IEEE Sens J 23(11):12018–12030. https://doi.org/10.1109/JSEN.2023.3269030
7. Khan WA, Masoud M, Eltoukhy AE, Ullah M (2024) Stacked encoded cascade error feedback deep extreme learning machine network for manufacturing order completion time. J Intell Manuf. https://doi.org/10.1007/s10845-023-02303-0
8. Ganin Y, Lempitsky V (2015) Unsupervised domain adaptation by backpropagation. In: International Conference on Machine Learning, pp. 1180–1189. https://doi.org/10.48550/arXiv.1409.7495
9. Jiang J, Shu Y, Wang J, Long M (2022) Transferability in deep learning: a survey. arXiv e-prints https://doi.org/10.48550/arXiv.2201.05867
10. Liu G, Shen W, Gao L, Kusiak A (2023) Automated broad transfer learning for cross-domain fault diagnosis. J Manuf Syst 66:27–41. https://doi.org/10.1016/j.jmsy.2022.11.003
11. Liu Q, Zhang Z, Guo P, Wang Y, Liang J (2024) Enhancing aircraft engine remaining useful life prediction via multiscale deep transfer learning with limited data. J Comput Des Eng 11(1):343–355. https://doi.org/10.1093/jcde/qwae018
12. Liu X, Yoo C, Xing F, Oh H, El Fakhri G, Kang J-W, Woo J et al (2022) Deep unsupervised domain adaptation: a review of recent advances and perspectives. APSIPA Trans Signal Inf Process 10(1561/116):00000192
13. He R, Tian Z, Zuo M (2023) A transferable neural network method for remaining useful life prediction. Mech Syst Signal Process 183:109608. https://doi.org/10.1016/j.ymssp.2022.109608
14. Li W, Shang Z, Gao M, Qian S, Feng Z (2022) Remaining useful life prediction based on transfer multi-stage shrinkage attention temporal convolutional network under variable working conditions. Reliab Eng Syst Saf 226:108722. https://doi.org/10.1016/j.ress.2022.108722
15. Jin Y, Qin C, Liu J, Lin K, Liu C (2020) A novel domain adaptive residual network for automatic atrial fibrillation detection. Knowl-Based Syst 203:106122. https://doi.org/10.1016/j.knosys.2020.106122
16. Zhao K, Jiang H, Wang K, Pei Z (2021) Joint distribution adaptation network with adversarial learning for rolling bearing fault diagnosis. Knowl-Based Syst 222:106974. https://doi.org/10.1016/j.knosys.2021.106974
17. Xiao Y, Shao H, Han S, Huo Z, Wan J (2022) Novel joint transfer network for unsupervised bearing fault diagnosis from simulation domain to experimental domain. IEEE Trans Mech 27(6):5254–5263. https://doi.org/10.1109/TMECH.2022.3177174
18. Zhang J, Li X, Tian J, Jiang Y, Luo H, Yin S (2023) A variational local weighted deep sub-domain adaptation network for remaining useful life prediction facing cross-domain condition. Reliab Eng Syst Saf 231:108986. https://doi.org/10.1016/j.ress.2022.108986
19. Cao H, Shao H, Zhong X, Deng Q, Yang X, Xuan J (2022) Unsupervised domain-share CNN for machine fault transfer diagnosis from steady speeds to time-varying speeds. J Manuf Syst 62:186–198. https://doi.org/10.1016/j.jmsy.2021.11.016

20. Su K, Liu J, Xiong H (2022) A multi-level adaptation scheme for hierarchical bearing fault diagnosis under variable working conditions. J Manuf Syst 64:251–260. https://doi.org/10.1016/j.jmsy.2022.06.009
21. He R, Tian Z, Zuo M (2023) Machine prognostics under varying operating conditions based on state-space and neural network modeling. Mech Syst Signal Process 182:109598. https://doi.org/10.1016/j.ymssp.2022.109598
22. Zhuang J, Cao Y, Jia M, Zhao X, Peng Q (2023) Remaining useful life prediction of bearings using multi-source adversarial online regression under online unknown conditions. Expert Syst Appl 227:120276. https://doi.org/10.1016/j.eswa.2023.120276
23. Li K, Li Z, Jia X, Liu L, Chen M (2024) A domain adversarial graph convolutional network for intelligent monitoring of tool wear in machine tools. Comput Ind Eng 187:109795. https://doi.org/10.1016/j.cie.2023.109795
24. 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. Reliab Eng Syst Saf 216:108012. https://doi.org/10.1016/j.ress.2021.108012
25. Ye Z, Yu J (2022) A selective adversarial adaptation network for remaining useful life prediction of machines under different working conditions. IEEE Syst J. https://doi.org/10.1109/JSYST.2022.3183134
26. Shi H, Huang C, Zhang X, Zhao J, Li S (2023) Wasserstein distance based multi-scale adversarial domain adaptation method for remaining useful life prediction. Appl Intell 53(3):3622–3637. https://doi.org/10.1007/s10489-022-03670-6
27. Ding Y, Jia M, Cao Y, Ding P, Zhao X, Lee C-G (2023) Domain generalization via adversarial out-domain augmentation for remaining useful life prediction of bearings under unseen conditions. Knowl-Based Syst 261:110199. https://doi.org/10.1016/j.knosys.2022.110199
28. Ragab M, Chen Z, Wu M, Foo CS, Kwoh CK, Yan R, Li X (2021) Contrastive adversarial domain adaptation for machine remaining useful life prediction. IEEE Trans Ind Inf 17(8):5239–5249. https://doi.org/10.1109/TII.2020.3032690
29. Zhuang J, Jia M, Zhao X (2022) An adversarial transfer network with supervised metric for remaining useful life prediction of rolling bearing under multiple working conditions. Reliab Eng Syst Saf 225:108599. https://doi.org/10.1016/j.ress.2022.108599
30. Zhou H, Lei Z, Zio E, Wen G, Liu Z, Su Y, Chen X (2023) Conditional feature disentanglement learning for anomaly detection in machines operating under time-varying conditions. Mech Syst Signal Process 191:110139. https://doi.org/10.1016/j.ymssp.2023.110139
31. Hu T, Guo Y, Gu L, Zhou Y, Zhang Z, Zhou Z (2022) Remaining useful life prediction of bearings under different working conditions using a deep feature disentanglement based transfer learning method. Reliab Eng Syst Saf 219:108265. https://doi.org/10.1016/j.ress.2021.108265
32. Costa PRdO, Akçay A, Zhang Y, Kaymak U (2020) Remaining useful lifetime prediction via deep domain adaptation. Reliab Eng Syst Saf 195:106682. https://doi.org/10.1016/j.ress.2019.106682
33. Zhang K, Chen J, He S, Li F, Feng Y, Zhou Z (2022) Triplet metric driven multi-head GNN augmented with decoupling adversarial learning for intelligent fault diagnosis of machines under varying working condition. J Manuf Syst 62:1–16. https://doi.org/10.1016/j.jmsy.2021.10.014
34. Forouzandeh Shahraki A, Al-Dahidi S, Rahim Taleqani A, Yadav OP (2023) Using lstm neural network to predict remaining useful life of electrolytic capacitors in dynamic operating conditions. Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability 237(1):16–28. https://doi.org/10.1177/1748006X2210875
35. Zhang J, Jiang Y, Wu S, Li X, Luo H, Yin S (2022) Prediction of remaining useful life based on bidirectional gated recurrent unit with temporal self-attention mechanism. Reliab Eng Syst Saf 221:108297. https://doi.org/10.1016/j.ress.2021.108297
36. Dong S, Xiao J, Hu X, Fang N, Liu L, Yao J (2023) Deep transfer learning based on Bi-LSTM and attention for remaining useful life prediction of rolling bearing. Reliab Eng Syst Saf 230:108914. https://doi.org/10.1016/j.ress.2022.108914
37. Wang Z, Liu T, Wu X, Liu C (2023) A diagnosis method for imbalanced bearing data based on improved smote model combined with CNN-AM. J Comput Des Eng 10(5):1930–1940. https://doi.org/10.1093/jcde/qwad081
38. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I (2017) Attention is all you need. Advances in Neural Information Processing Systems, pp. 5998–6008. https://doi.org/10.48550/arXiv.1706.03762

39. Jose S, Ngouna RH, Nguyen KT, Medjaher K (2022) Solving time alignment issue of multimodal data for accurate prognostics with CNN-Transformer-LSTM network. In: 2022 8th International Conference on Control, Decision and Information Technologies (CoDIT), vol. 1, pp. 280–285. https://doi.org/10.1109/CoDIT55151.2022.9804090 . IEEE
40. Zhang Z, Song W, Li Q (2022) Dual-aspect self-attention based on transformer for remaining useful life prediction. IEEE Trans Instrum Meas 71:1–11. https://doi.org/10.1109/TIM.2022.3160561
41. Xu D, Xiao X, Liu J, Sui S (2023) Spatio-temporal degradation modeling and remaining useful life prediction under multiple operating conditions based on attention mechanism and deep learning. Reliab Eng Syst Saf 229:108886. https://doi.org/10.1016/j.ress.2022.108886
42. Zhang J, Li X, Tian J, Luo H, Yin S (2023) An integrated multi-head dual sparse self-attention network for remaining useful life prediction. Reliab Eng Syst Saf 233:109096. https://doi.org/10.1016/j.ress.2023.109096
43. Peng G, Qi L, Shui Y, Jianyu X, Xiang T, Chao G (2023) A transformer with layer-cross decoding for remaining useful life prediction. J Supercomput 79:11558–11584. https://doi.org/10.1007/s11227-023-05126-1
44. Zhou H, Zhang S, Peng J, Zhang S, Li J, Xiong H, Zhang W (2021) Informer: Beyond efficient transformer for long sequence time-series forecasting. In: Proceedings of AAAI, pp. 11106–11115. https://ojs.aaai.org/index.php/AAAI/article/view/17325
45. Liu P, Qiu X, Huang X (2017) Adversarial multi-task learning for text classification. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 1–10. ACL, https://doi.org/10.18653/v1/P17-1001
46. Bousmalis K, Trigeorgis G, Silberman N, Krishnan D, Erhan D (2016) Domain separation networks. Advan Neural Inf Process Syst. https://doi.org/10.48550/arXiv.1608.06019
47. Kingma DP, Ba J (2014) Adam: a method for stochastic optimization. CoRR 1412:6980. https://doi.org/10.48550/arXiv.1412.6980
48. Frederick DK, Decastro JA, Litt JS (2007) User's guide for the commercial modular aero-propulsion system simulation (C-MAPSS).
49. Zhang Z, Chen X, Zio E (2022) A framework for predicting the remaining useful life of machinery working under time-varying operational conditions. Appl Soft Comput 126:109164. https://doi.org/10.1016/j.asoc.2022.109164
50. Zhang Z, Chen X, Zio E, Longxiao L (2023) Multi-task learning boosted predictions of the remaining useful life of aero-engines under scenarios of working-condition shift. Reliab Eng Syst Saf 237:109350. https://doi.org/10.1016/j.ress.2023.109350
51. Cao Y, Jia M, Ding P, Ding Y (2021) Transfer learning for remaining useful life prediction of multi-conditions bearings based on bidirectional-GRU network. Measurement 178(5):109287. https://doi.org/10.1016/j.measurement.2021.109287
52. Li X, Zhang W, Ma H, Luo Z, Li X (2020) Data alignments in machinery remaining useful life prediction using deep adversarial neural networks. Knowl-Based Syst 197:105843. https://doi.org/10.1016/j.knosys.2020.105843
53. Duan Y, Xiao J, Li H, Zhang J (2022) Cross-domain remaining useful life prediction based on adversarial training. Machines 10(6):438. https://doi.org/10.3390/machines10060438

Authors and Affiliations

Zhiyao Zhang[1,2] · Jiting Cheng[3] · Pengpeng Chen[4] · Shuang Gao[5] · Xiaohui Chen[2] · Enrico Zio[6,7]

✉ Xiaohui Chen
chenxiaohui@cqu.edu.cn

Zhiyao Zhang
zhiyaozhang@swjtu.edu.cn

Jiting Cheng
jitingcheng2024@163.com

Pengpeng Chen
cplwting@163.com

Shuang Gao
sgao8@qnlm.ac

Enrico Zio
enrico.zio@mines-paristech.fr

1 School of Mechanical Engineering, Southwest Jiaotong University, Chengdu, China

2 State Key Laboratory of Mechanical Transmission, Chongqing University, Chongqing, China

3 Nus School of Computing, National University of Singapore, Kent Ridge, Singapore

4 China's Aviation System Engineering Research Institute, Beijing, China

5 Laoshan Laboratory (Qingdao), Qingdao, China

6 Department of Energy, Politecnico di Milano, Milano, Italy

7 MINES Paris, PSL University, CRC, Sophia Antipolis, Paris, France