Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Aero-engine RUL Prediction Method Based on Deep Equilibrium Network with Degradation Alignment and Domain Adaptation

Yun Liu
PLA Rocket Force University of Engineering
Qi Zhang
158297996980163.com

PLA Rocket Force University of Engineering
Lihao Yang
PLA Rocket Force University of Engineering
Zhengyang Zhang
PLA Rocket Force University of Engineering

Research Article

Keywords: Aero-engine, Remaining Useful Life prediction, Deep Equilibrium Model, Degradation Alignment, Domain Adaptation

Posted Date: September 10th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-10626383/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Aero-engine RUL Prediction Method Based on Deep Equilibrium Network with Degradation Alignment and Domain Adaptation

Yun Liu, Qi Zhang*, Lihao Yang, Zhengyang Zhang

PLA Rocket Force University of Engineering, Hongqing Road, Baqiao District, Xi'an, 710025, Shaanxi, China .

Contributing authors: 986417528@qq.cpm; 15829799698@163.com; yanglihao003@163.com; 3301162998@qq.com;

Abstract

Aiming at the problems of domain distribution shift, gradient vanishing, and insufficient generalization performance in aero-engine remaining useful life (RUL) prediction, this paper constructs DAI-DEQ and adaptive DAI-DEQ models by integrating degradation alignment, deep equilibrium structure, and adaptive modules, and proposes a novel aero-engine RUL prediction method based on a deep equilibrium network with collaborative degradation alignment and domain adaptation. First, multi-working condition clustering normalization, exponential moving average smoothing, and random forest feature selection are applied to preprocess the NASA C-MAPSS dataset. The Degradation Alignment Indicator (DAI) is embedded as an attention weighting coefficient into the deep equilibrium network built upon Anderson fixed-point iteration, which alleviates gradient vanishing and strengthens the extraction of degradation features. By combining the gradient reversal layer and domain discriminator, domain adaptation is realized, and Bayesian neural networks with Monte Carlo sampling are introduced to quantify prediction uncertainty. Experimental results show that in intra-domain prediction on the four standard C-MAPSS subsets, the RMSE values of the DA-DEQ model are 7.50 (FD001), 7.35 (FD002), 8.42 (FD003), and 7.41 (FD004), outperforming existing mainstream methods, including IDMFFN and GA-Transformer. In cross-domain transfer experiments, the RMSE of the adaptive DA-DEQ model drops from 49.85 to 12.90 (a decline of 74.1%) under the FD001→FD002 scenario, effectively alleviating domain shift. Ablation experiments confirm that removing the DAI module increases the maximum RMSE by up to 4.56 on FD002 and FD004, verifying its core effect. The experimental results demonstrate that the proposed method achieves remarkable superiority

in prediction accuracy, training stability, and generalization ability, and holds significant practical value for aero-engine health prognosis.

Keywords: Aero-engine, Remaining Useful Life prediction, Deep Equilibrium Model, Degradation Alignment, Domain Adaptation

1 Introduction

As the core power component of aircraft, aero-engines govern the safety and success of flight missions and are closely related to personal safety and aviation property security. As typical complex rotating machinery, aero-engines operate continuously under extreme conditions characterized by ultra-high temperatures, ultra-high pressures, and high rotational speeds. Irreversible performance degradation and structural damage inevitably occur in key components during long-term service, which may further cause unexpected unplanned downtime and even catastrophic flight accidents. Remaining Useful Life (RUL) prediction serves as a core module of Prognostics and Health Management (PHM)[1]. It can quantitatively evaluate equipment degradation degree and predict the residual service time of aero-engines, thereby promoting the transformation of maintenance strategy from traditional scheduled maintenance to Condition-Based Maintenance (CBM)[2]. RUL prediction technology can effectively avoid catastrophic failures, reduce redundant maintenance costs, and improve fleet operational efficiency and equipment utilization, demonstrating prominent engineering application value. Therefore, developing high-precision and strongly generalized RUL prediction models has always been a key research focus in the field of aero-engine health monitoring.

Benefiting from the rapid advancement of sensor monitoring technology and industrial big data acquisition systems, data-driven RUL prediction methods have gradually become the mainstream, replacing traditional physics-based models and statistical analysis methods that involve complicated modeling procedures and poor generalization. Empowered by the powerful nonlinear fitting and temporal feature mining capabilities of deep learning, a large number of advanced deep learning architectures have been developed to enhance single-condition RUL prediction performance in recent years. For temporal feature enhancement, multi-head sparse self-attention networks and knowledge-enhanced deep learning frameworks have been proposed to capture long-range degradation dependencies hidden in sensor sequences[3, 4]. To overcome the inherent limitations of recurrent neural networks in time series modeling, gated RNNs with differential weighted storage mechanisms[5] and dynamic self-learning neural networks have been designed to suppress gradient vanishing and strengthen dynamic degradation tracking ability[6, 7]. Meanwhile, multi-scale temporal convolutional networks, dual-attention lightweight LSTMs, and trend-augmented Transformers have been successively applied to extract hierarchical temporal degradation features, achieving satisfactory prediction accuracy under stable single-operating conditions[8–10]. Furthermore, a sensor-aware CapsNet was proposed to realize trustworthy multi-sensor feature fusion and further improve the reliability of deep degradation feature

extraction[11]. Nevertheless, all existing deep learning models rely on an ideal prerequisite that the training source domain and test target domain follow identical distributions. In actual engineering scenarios, aero-engines operate under time-varying conditions. Condition switching, environmental disturbances, and diverse fault modes trigger severe domain distribution shifts, leading to dramatic performance degradation when models trained on the source domain are directly deployed to unseen target domains, which severely restricts the practical cross-condition prediction application of existing methods.

To address the domain shift caused by inconsistent working conditions, transfer learning and domain adaptation technologies have been introduced into cross-domain RUL prediction to achieve effective degradation knowledge transfer from labeled source domains to unlabeled target domains. Current mainstream domain adaptation strategies are divided into four categories: adversarial domain adaptation, kernel space domain adaptation, contrastive learning-based domain adaptation, and ensemble supervised domain adaptation. However, each category faces unavoidable technical bottlenecks in practical degradation feature alignment. First, Wasserstein distance-based multi-scale adversarial domain adaptation narrows cross-domain distribution gaps via adversarial training but suffers from mode collapse and unstable training during iterative optimization[12]. Second, reproducing kernel Hilbert space (RKHS)-based domain adaptation[13] introduces kernel space constraints to enhance model interpretability for cross-domain alignment, yet it incurs massive computational overhead and limits real-time prediction efficiency[14]. Third, order-preserving kernel contrastive learning achieves fine-grained sample-level alignment while preserving temporal degradation order, but it ignores the inherent uncertainty distribution discrepancy between source and target degradation data[15]. Fourth, AdaBoost-integrated LSTM[16] supervised domain adaptation relies on partially labeled target domain samples to stabilize training, which cannot be deployed in fully unlabeled industrial target scenarios[17]. Apart from the above four mainstream frameworks, adaptive degradation feature alignment transfer learning has also been proposed to match cross-condition degradation trends, while still failing to balance domain alignment and discriminative degradation feature retention[18]. In addition, existing cross-domain prediction methods share several common limitations. The global adversarial alignment strategy easily causes excessive feature alignment and loses inherent degradation characteristics. The Degradation Alignment Index (DAI)[19] is only adopted as an independent evaluation indicator rather than being embedded into deep networks to guide degradation feature learning. Traditional deep networks constructed by layer stacking still suffer from gradient vanishing and unstable training. Moreover, most existing models only output deterministic prediction results without quantitative uncertainty analysis, which cannot support reliable decision-making in practical engineering applications. In response to the above limitations, recent studies have carried out supplementary explorations from three perspectives: model interpretability, prediction uncertainty quantification, and RUL prediction for specific devices. Liu et al. developed a novel interpretable deep autoencoder to construct adaptive health indicators and achieved RUL prediction via similarity-based methods, effectively solving the poor interpretability problem

of conventional deep learning prognostic models[20]. Chen et al. established a probabilistic inference framework dedicated to RUL prediction to realize the modeling, propagation, and quantitative evaluation of prediction uncertainty, further improving the credibility evaluation system of prognostic results[21]. Considering the frequency interference characteristics in actual operation, Li et al. adopted long short-term memory networks to predict the RUL of RF MEMS switches, broadening the application scope of time-series models in the prognostics of precision electromechanical equipment. Nevertheless, these studies still fail to address practical engineering challenges such as cross-condition domain shift, source-free domain adaptation, and prediction with incomplete degradation sequences[22].

Apart from inherent algorithm defects, current cross-domain RUL prediction methods confront three non-negligible engineering bottlenecks in practical industrial deployment, which are rarely solved simultaneously by existing research. First, core monitoring data of aero-engines belong to industrial confidential data, so original source domain data are completely inaccessible in real scenarios. Traditional domain adaptation methods relying on original source domain data for model training will fail directly, resulting in the source-free domain adaptation dilemma. Second, complete full-life degradation trajectories can hardly be collected by industrial sensors, and most available monitoring samples are incomplete truncated degradation sequences intercepted before equipment failure. Existing source-free adaptation methods will produce sharply amplified prediction errors when processing incomplete degradation samples[23]. Third, most existing models only extract single-modal time-series features and ignore implicit spatial texture degradation information contained in time-series signals. Single-modal feature extraction cannot fully characterize the whole-life degradation law of aero-engines, limiting the upper bound of cross-domain prediction performance. Although multimodal transfer learning frameworks have been developed to fuse temporal signals and recurrence plot images to compensate for single-modal feature deficiency, this research only focuses on conventional supervised cross-domain scenarios and cannot cope with source-free and incomplete degradation challenges[24]. In summary, previous studies only optimize one or two pain points separately, while a unified integrated cross-domain RUL prediction framework that simultaneously supports source-free adaptation, incomplete degradation trajectory adaptation, multimodal complementary feature fusion, and stable deep network training is still missing.

Aiming at the above challenges in cross-domain aero-engine RUL prediction, including severe domain distribution shift, insufficient degradation feature extraction, gradient vanishing of deep networks, lack of prediction uncertainty quantification, and poor adaptability to source-free scenarios and incomplete degradation samples, this paper proposes a novel RUL prediction method integrating a deep equilibrium model and adversarial domain adaptation. The proposed method adopts a deep equilibrium model[25] with implicit deep modeling to alleviate gradient vanishing and improve training stability compared with traditional stacked deep networks. Combined with improved adversarial domain adaptation, it realizes effective cross-domain feature alignment while retaining critical degradation features and avoiding excessive feature alignment. Meanwhile, degradation correlation indicators are embedded into the whole

network to guide discriminative degradation feature learning dynamically. Comparative experiments based on the public C-MAPSS[26] aero-engine dataset verify that the proposed method achieves superior prediction accuracy and cross-condition generalization performance compared with state-of-the-art methods. The overall research framework of this paper is illustrated in Fig. 1.

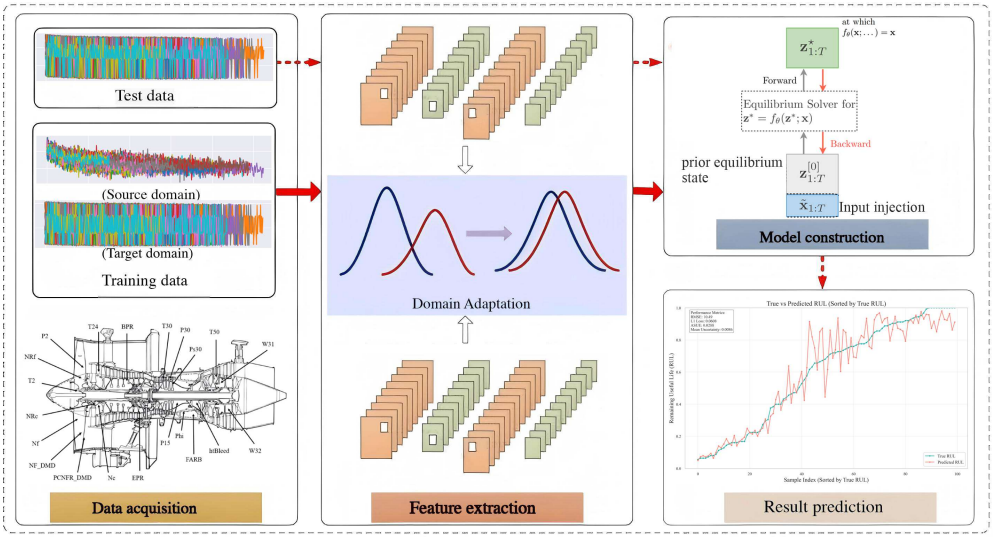


Fig. 1: Overall Research Framework

(1) A complete C-MAPSS data preprocessing pipeline is established. Standardization and clustering normalization strategies are separately designed for single-working-condition and multi-working-condition datasets. Exponential moving average is adopted to smooth sensor data, and redundant features are eliminated according to feature importance derived from random forest.

(2) The DAI is calculated via a fixed time window and integrated into the dual-input attention module as a correlation metric, which guides the deep equilibrium network to focus on highly correlated degradation features.

(3) An adversarial domain adaptation framework based on gradient reversal layer and domain discriminator is constructed to align feature distributions between the source domain and target domain, so as to mitigate the adverse effects of domain shift on model generalization ability.

(4) Anderson iteration is utilized to solve the fixed points of deep equilibrium networks. Combined with Bayesian neural networks and Monte Carlo sampling, RUL prediction and uncertainty quantification are accomplished, which enhances the reliability of prediction results.

The remainder of this paper is organized as follows. Section 2 reviews domestic and overseas research progress on aero-engine RUL prediction and cross-domain domain adaptation, and clarifies the research gaps and objectives. Section 3 elaborates the

basic theories and fundamental algorithms adopted in this research. Section 4 details the network architecture and training procedure of the proposed cross-domain RUL prediction framework. Section 5 presents comparative and ablation experiments to verify the effectiveness of each core module. Section 6 summarizes the research and discusses future directions.

2 Related Theories and Problem Description

2.1 Multi-working-condition Domain Distribution Shift Problem

Aero-engines undergo diverse working conditions, including takeoff, cruise, and landing, during actual flight missions. Significant differences exist in the physical responses of engines under various working conditions, causing the distribution of sensor monitoring data to shift dynamically with working condition changes. Such domain distribution shift violates the independent and identically distributed (i.i.d.) assumption in conventional machine learning, leading to inconsistent joint distributions between training and test datasets, which can be formulated as:

$$P_{\text{source}}(x, y) \neq P_{\text{target}}(x, y) \tag{1}$$

where x refers to the sensor observation sequence, and y is the corresponding RUL label. This distribution discrepancy can be further divided into marginal feature distribution difference:

$$P_s(x) \neq P_t(x) \tag{2}$$

and discrepancy in degradation conditional distribution:

$$P_s(y|x) \neq P_t(y|x) \tag{3}$$

The former is directly induced by working condition switching, and essential differences exist in sensor baseline values and fluctuation ranges under different operating conditions. The latter stems from the coupling effect between working conditions and degradation, and distinct sensor responses can be observed at the same degradation degree across various working conditions. Directly applying models trained on the source domain to the target domain usually results in a substantial drop in prediction accuracy.

Most existing domain adaptation methods adopt global feature alignment strategies, which may excessively eliminate key degradation features sensitive to operating conditions and cause feature homogenization, ultimately impairing the accuracy of degradation modeling. Meanwhile, most methods fail to introduce degradation correlation indicators to guide the domain alignment process, making it hard to strike an effective balance between distribution adaptation and degradation feature retention. Furthermore, they are not combined with deep equilibrium models, and thus cannot solve the gradient vanishing issue existing in traditional deep networks.

2.2 Two-stage Degradation Characteristics and Degradation Alignment Index

The degradation process of aero-engines presents distinct two-stage characteristics. The first is the slow degradation stage. In the early healthy service period of the engine, component wear and performance decay proceed at a mild rate, and only minor drift occurs in sensor signals, with noise and operating condition fluctuations dominating signal variations. The second is the accelerated degradation stage. As the engine nears the end of its service life, component performance deteriorates drastically, and sensor signals exhibit prominent trending variations, where degradation effects become the primary driver of signal changes.

Such staged features result in essential differences in the model's feature extraction demands across different degradation periods. In the slow degradation stage, the model is required to possess strong anti-interference capability to steadily extract long-term degradation trends while guaranteeing computational efficiency. In the accelerated degradation stage, it needs superior nonlinear representation capacity to precisely capture sharply varying degradation details.

To accurately characterize the consistency between sensor degradation trends and RUL attenuation trends, this paper constructs the Degradation Alignment Index (DAI). Its core idea is to distinguish the ongoing degradation stage by calculating the synchronization between sensor signal variation trends and RUL declining trends. Furthermore, the index is embedded into the subsequent deep equilibrium network as an attention weighting coefficient, guiding the model to focus on degradation features with high correlation. The specific calculation process and physical significance are as follows.

2.2.1 Definition of Basic Sequence for Sliding Window

Let the sliding window length be w. Taking the current time instant t as the window endpoint, consecutive time points, corresponding sensor measurements, and RUL values within the window are selected to construct the time series.

Time Node Sequence within the Window:

$$t = [t - w, t - w + 1, \ldots, t] \tag{4}$$

Sensor Measurement Sequence within the Window:

$$s = [s_{t-w}, s_{t-w+1}, \ldots, s_t] \tag{5}$$

RUL Sequence within the Window:

$$r = [r_{t-w}, r_{t-w+1}, \ldots, r_t] \tag{6}$$

To facilitate subsequent fitting and calculation, define the in-window sequence index as $i = 0, 1, \ldots, w$, which corresponds to the i-th temporal sampling point within the window.

2.2.2 Calculation of Trend Slope Based on LOWESS Robust Locally Weighted Regression

For time series within the sliding window, the LOWESS algorithm is employed to perform local linear fitting, which eliminates the global equal-weight fitting mode. Higher weights are assigned to recent time points and lower weights to remote ones through distance weighting. Meanwhile, a robust iterative mechanism is introduced to suppress outlier interference. The fitted slope can accurately reflect the instantaneous degradation trend inside the window. The variation slopes of sensor sequences and RUL sequences are presented in Fig. 2.

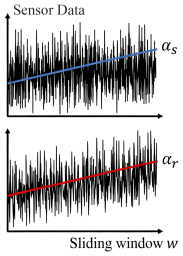


Fig. 2: Slope Variation Plot of Sensor Sequences and RUL Sequences

(a) Definition of the Local Weight Function

Taking any fitting center point within the window as the benchmark, the temporal distance weight is defined based on the tricubic kernel function to weaken the influence of remote time points and emphasize the recent degradation characteristics. The weight calculation formula is given as follows:

$$w_i(d) = \begin{cases} (1 - |d|^3)^3, & |d| \leq 1 \\ 0, & |d| > 1 \end{cases} \tag{7}$$

where d denotes the normalized distance of the time series index, $d = \frac{i - i_c}{w/2}$, and i_c represents the time series index of the window center, ensuring that the weight only acts on the local interval of the current sliding window.

(b) Slope calculation via robust weighted fitting

With the above kernel weights, local linear fitting based on weighted least squares is conducted for sensor sequences and RUL sequences respectively. The objective function of weighted sum of squared residuals is established, and the optimal trend slope is obtained by minimizing this objective function.

Weighted sum of squared residuals of sensor sequences:

$$J_s = \sum_{i=0}^{w} w_i \cdot \left(s_{t-w+i} - (k_s \cdot i + b_s) \right)^2 \tag{8}$$

Weighted residual sum of squares of RUL sequences:

$$J_r = \sum_{i=0}^{w} w_i \cdot \left(r_{t-w+i} - (k_r \cdot i + b_r)\right)^2 \tag{9}$$

Where k_s and b_s represent the slope and intercept of the fitted line for the sensor sequence, respectively; k_r and b_r represent the slope and intercept of the fitted line for the RUL sequence, respectively; w_i is the normalized kernel weight corresponding to each time series point.

By minimizing the objective function, the optimal variation slope α_s of the sensor signal and the optimal attenuation slope α_r of the RUL within the window are obtained. These slopes are noise-robust, eliminating the false trends caused by operating condition fluctuations and measurement noise.

2.2.3 Construction of Trend Variation Vector

Taking time span as the horizontal axis and the overall degradation variation of sequences as the vertical axis, a two-dimensional trend vector is constructed combined with the robust slope obtained by LOWESS fitting. This vector separately characterizes the actual variation trend of sensor signals and the real attenuation trend of RUL.

The sensor variation vector consists of window time span and total variation of sensor signals. Let the window time span be w. The total variation of sensor signals from the start to the end of the window is derived as $w\alpha_s$ according to the robust fitting slope. Accordingly, the sensor variation vector is expressed as follows:

$$v_s = (w, s_t - s_{t-w}) = (w, w\alpha_s) \tag{10}$$

The RUL decays continuously over time. The life attenuation amount between the start and end of the window calculated from the robust fitting slope is $w\alpha_r$. Since its variation direction is opposite to the temporal trend, the RUL attenuation vector is defined as follows:

$$v_r = (w, r_{t-w} - r_t) = (w, -w\alpha_r) \tag{11}$$

2.2.4 Derivation of Cosine of Vector Included Angle

The included angle between two groups of robust trend vectors can accurately reflect the trend synchronization between denoised sensor variation and RUL attenuation, as shown in Fig. 3.

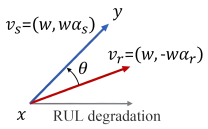


Fig. 3: Schematic Diagram of Trend Vector Included Angle

The cosine value of the included angle $\cos\theta$ between vectors $\boldsymbol{v_s}$ and $\boldsymbol{v_r}$ is calculated based on the vector dot product formula, and simplification is conducted by substituting the vector expressions:

$$\cos\theta = \frac{v_s \cdot v_r}{|v_s||v_r|} = \frac{w \cdot w + (w\alpha_s) \cdot (-w\alpha_r)}{w\sqrt{1+\alpha_s^2} \cdot w\sqrt{1+\alpha_r^2}} = \frac{1-\alpha_s\alpha_r}{\sqrt{1+\alpha_s^2}\sqrt{1+\alpha_r^2}} \tag{12}$$

The original value range of the included angle cosine $\cos\theta$ is $[-1, 1]$.

2.2.5 Index Normalization and Final Expression of DAI

To satisfy the requirements of engineering evaluation interval, the cosine value of denoised vector included angle is linearly mapped to the standard interval $[0, 1]$ for index normalization. Consequently, the calculation formula of DAI based on robust fitting is derived.

$$\mathrm{DAI} = \frac{\cos\theta + 1}{2} \tag{13}$$

The DAI indicator based on LOWESS robust fitting eliminates the interference of data noise and outliers. Its value can truly characterize the synchronization degree of degradation trends between sensor signals and RUL of the engine, and accurately reflect the actual degradation state of the engine. The specific physical meanings are described as follows:

(a) When DAI $\to 1$, the included angle θ between the two trend vectors approaches $0°$. The variation trend of denoised sensor signals is highly synchronized with the attenuation trend of RUL. The engine is in a real accelerated degradation stage with stable degradation characteristics and valid monitoring signals.

(b) When DAI $\to 0.5$, the included angle θ approaches $90°$. No stable degradation trend exists in sensor signals, and the life attenuation rate remains gentle. The engine operates in a slow and steady degradation stage.

(c) When DAI < 0.5, the two trend vectors present misaligned or opposite tendencies without valid degradation correlation. This phenomenon is not caused by equipment degradation, but mainly results from short-term operating condition fluctuations and random measurement noise. It can be concluded that no effective degradation trend occurs at present.

2.3 Deep Equilibrium Model

The Deep Equilibrium Network (DEQ), a new implicit deep learning framework, conducts feature modeling by solving fixed points of nonlinear transformations instead of adopting conventional layer-wise forward propagation. It enables unlimited deep feature representation with memory overhead unaffected by network depth. For aero-engine RUL prediction, which relies on mining degradation patterns from long time-series data, traditional networks consume excessive memory and computational resources when stacking layers to enhance feature extraction. By contrast, DEQ

reorganizes feature learning and adapts to the complex degradation mechanism of aero-engines through fixed-point iteration. It eliminates memory restrictions from network depth and reduces resource demands for high-precision RUL prediction.

For the input feature x, the layer-wise propagation process of traditional feedforward networks can be expressed as:

$$z_1 = f_\theta(z_0, x),\ z_2 = f_\theta(z_1, x),\ \ldots,\ z_T = f_\theta(z_{T-1}, x) \tag{14}$$

where $z_0 = 0$ denotes the initial state, f_θ represents the nonlinear transformation module with parameter θ, and T is the number of network layers. When $T \to \infty$, the network output will converge to a unique fixed point z^* if the transformation meets the convergence condition, which satisfies:

$$z^* = f_\theta(z^*, x) \tag{15}$$

This fixed point is taken as the final output feature of DEQ. There is no need to store intermediate results of layer-wise propagation, which greatly reduces memory consumption.

2.3.1 Fixed Point Solution for Forward Propagation

DEQ forward propagation depends on solving nonlinear fixed-point equations. Traditional iteration methods converge slowly, consume massive resources, and tend to diverge, which cannot meet the demands of aero-engine time-series modeling. We adopt Anderson acceleration iteration here. It achieves high convergence efficiency and strong stability, effectively reducing computation and memory overhead and fitting engine degradation rules well. By contrast, Newton iteration is computationally expensive due to Jacobian calculation, Gauss-Seidel iteration has poor convergence and stability, and basic iteration is low-efficiency.

(1) Initialize the iteration sequence $z_0 = 0$ and calculate the initial residual:

$$r^{(0)} = f_\theta\big(z^{(0)}, x\big) - z^{(0)} \tag{16}$$

Set the convergence threshold $\varepsilon = 10^{-6}$ and the history window length $m = 5$.

(2) For the k-th iteration, construct the residual matrix $\boldsymbol{R}_k = [\boldsymbol{r}_{k-m+1}, \ldots, \boldsymbol{r}_k]$, and solve the constrained least squares problem:

$$\min_{\boldsymbol{\gamma}} \|\boldsymbol{R}_k \boldsymbol{\gamma}\|_2, \quad \text{s.t.} \sum_{i=1}^{m} \gamma_i = 1 \tag{17}$$

(3) Update the iteration value:

$$\boldsymbol{z}_{k+1} = \sum_{i=1}^{m} \gamma_i \big(\boldsymbol{z}_{k-m+i} + \boldsymbol{r}_{k-m+i}\big) \tag{18}$$

and calculate the new residual:

$$\boldsymbol{r}_{k+1} = f_\theta(\boldsymbol{z}_{k+1}, \boldsymbol{x}) - \boldsymbol{z}_{k+1} \tag{19}$$

(4) Stop the iteration when $\|\boldsymbol{r}_{k+1}\|_2 < \varepsilon$, and set the fixed point $\boldsymbol{z}^* = \boldsymbol{z}_{k+1}$.

2.3.2 Derivation of Implicit Gradient for Back Propagation

Let the transformation function of DEQ be $f_\theta(\cdot, \boldsymbol{x})$, where θ denotes the learnable parameters of the model and $\boldsymbol{x}$ represents the input feature. If a latent state $\boldsymbol{z}^*$ satisfies the fixed-point equation $\boldsymbol{z}^* = f_\theta(\boldsymbol{z}^*, \boldsymbol{x})$, then $\boldsymbol{z}^*$ is regarded as the equilibrium state of the DEQ model.

Define the loss function of the model as $\mathcal{L}$. The gradient vector of the loss function with respect to the equilibrium state is expressed as $\boldsymbol{g} = \partial\mathcal{L}/\partial\boldsymbol{z}^*$, which serves as the fundamental gradient basis for model back propagation. Let $\boldsymbol{J}$ be the Jacobian matrix of the transformation function f_θ with respect to the latent state at the equilibrium point $\boldsymbol{z}^*$, namely $\boldsymbol{J} = \partial f_\theta(\boldsymbol{z}^*, \boldsymbol{x})/\partial\boldsymbol{z}^*$, and $\boldsymbol{I}$ represents the identity matrix with the same dimension.

For the equilibrium state satisfying $\boldsymbol{z}^* = f_\theta(\boldsymbol{z}^*, \boldsymbol{x})$, if the matrix $\boldsymbol{I} - \boldsymbol{J}$ is invertible, the gradient of the loss function with respect to model parameter θ conforms to the following formula:

$$\frac{\partial\mathcal{L}}{\partial\theta} = \boldsymbol{g}^\top \cdot (\boldsymbol{I} - \boldsymbol{J})^{-1} \cdot \frac{\partial f_\theta(\boldsymbol{z}^*, \boldsymbol{x})}{\partial\theta} \tag{20}$$

This gradient calculation strategy avoids storing intermediate results generated in forward iteration, and realizes the decoupling between memory consumption and network depth.

The detailed derivation process is presented as follows. First, take the partial derivative with respect to parameter θ on both sides of the fixed-point equilibrium equation. According to the derivation rule of composite functions, the following relation can be obtained:

$$\frac{\partial\boldsymbol{z}^*}{\partial\theta} = \frac{\partial f_\theta(\boldsymbol{z}^*, \boldsymbol{x})}{\partial\boldsymbol{z}^*} \cdot \frac{\partial\boldsymbol{z}^*}{\partial\theta} + \frac{\partial f_\theta(\boldsymbol{z}^*, \boldsymbol{x})}{\partial\theta} \tag{21}$$

Substitute the definition of the Jacobian matrix $\boldsymbol{J} = \partial f_\theta/\partial\boldsymbol{z}^*$ and rearrange the terms containing $\partial\boldsymbol{z}^*/\partial\theta$ to the left side of the equation:

$$(\boldsymbol{I} - \boldsymbol{J}) \cdot \frac{\partial\boldsymbol{z}^*}{\partial\theta} = \frac{\partial f_\theta}{\partial\theta} \tag{22}$$

Given the invertibility of matrix $\boldsymbol{I} - \boldsymbol{J}$, multiply both sides of the equation by $(\boldsymbol{I} - \boldsymbol{J})^{-1}$ from the left, and the gradient of equilibrium state concerning model parameters can be solved as:

$$\frac{\partial\boldsymbol{z}^*}{\partial\theta} = (\boldsymbol{I} - \boldsymbol{J})^{-1} \cdot \frac{\partial f_\theta}{\partial\theta} \tag{23}$$

Combined with the chain rule of neural network gradient and the gradient definition $\boldsymbol{g} = \partial\mathcal{L}/\partial\boldsymbol{z}^*$, the final gradient of loss function with respect to parameter θ is derived

as:

$$\frac{\partial \mathcal{L}}{\partial \theta} = \boldsymbol{g}^\top \cdot \frac{\partial \boldsymbol{z}^*}{\partial \theta} = \boldsymbol{g}^\top (\boldsymbol{I} - \boldsymbol{J})^{-1} \frac{\partial f_\theta}{\partial \theta} \tag{24}$$

Introduce an auxiliary row vector $\boldsymbol{u}^\top = \boldsymbol{g}^\top (\boldsymbol{I} - \boldsymbol{J})^{-1}$, and the matrix inversion operation can be converted into solving a linear system of equations:

$$\boldsymbol{u}^\top (\boldsymbol{I} - \boldsymbol{J}) = \boldsymbol{g}^\top \tag{25}$$

After obtaining vector $\boldsymbol{u}$ via the conjugate gradient method, the parameter gradient formula can be simplified into:

$$\frac{\partial \mathcal{L}}{\partial \theta} = \boldsymbol{u}^\top \cdot \frac{\partial f_\theta}{\partial \theta} \tag{26}$$

DEQ gradients are computed solely from equilibrium-state Jacobian and gradient information, requiring no intermediate feature storage. It eliminates gradient anomalies and separates memory usage from network depth, improving model stability and memory efficiency. Even so, conventional DEQ has three key limitations for cross-domain aero-engine life prediction: absence of degradation-aware design, lack of adversarial domain adaptation against distribution deviation, and no uncertainty quantification for prediction confidence. These flaws weaken its robustness and practical value in engineering.

3 Domain Adaptive DAI-DEQ

This paper proposes an aero-engine RUL prediction method integrating DEQ and adversarial domain adaptation. The model framework is shown in Fig. 4. The overall framework consists of five core modules: the Data Preprocessing and Feature Engineering Module, the Degradation Alignment Index (DAI) Calculation and Dual-Input Attention Module, the Deep Equilibrium Feature Extraction Module Based on Anderson Acceleration Iteration, the Adversarial Domain Adaptation Module, and the Bayesian RUL Prediction and Uncertainty Quantification Module.

First, data quality is improved through standardization preprocessing, multi-condition clustering normalization, feature selection, and time series smoothing. Second, the DAI is calculated and embedded into the dual-input attention mechanism to guide the model to focus on highly correlated degradation features. Subsequently, the deep equilibrium network based on Anderson acceleration iteration is adopted to complete deep feature extraction, fundamentally avoiding the gradient vanishing problem. Meanwhile, an adversarial domain adaptation framework based on the Gradient Reversal Layer (GRL)[27] is constructed to achieve distribution alignment between the source domain and the target domain. Finally, a Bayesian neural network[28] is employed for RUL prediction, and uncertainty quantification is realized via Monte Carlo sampling.

The principles and implementation details of each module are elaborated respectively in the following sections.

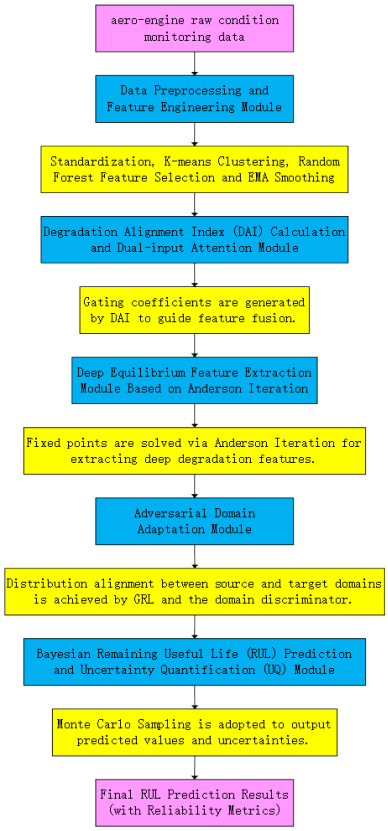


Fig. 4: Model Architecture Diagram

3.1 Data Preprocessing and Feature Engineering Module

3.1.1 K-means Clustering Partition

Working conditions cause significant discrepancies in engine operating parameters. Mixed modeling of multi-condition data will trigger misjudgment and gradient deviation, resulting in biased RUL prediction. K-means[29] is a distance-based clustering algorithm that generates compact clusters. Clustering divides data by working conditions, and subsequent separate data processing removes condition interference. It enables the model to learn real degradation characteristics and boosts the overall performance of RUL prediction. The relevant formulas are given as follows:

(1) Euclidean Distance

For a sample $\boldsymbol{x}_i = [x_{i1}, x_{i2}, \ldots, x_{in}]$ (where n is the dimension of operating condition parameters) and the centroid $\boldsymbol{c}_k = [c_{k1}, c_{k2}, \ldots, c_{kn}]$ of the k-th cluster, the distance

formula is expressed as:

$$d(\boldsymbol{x}_i, \boldsymbol{c}_k) = \sqrt{\sum_{j=1}^{n}(x_{ij} - c_{kj})^2} \tag{27}$$

(2) K-means Objective Function

The objective is to minimize the sum of squared distances from all samples to the centroids of their corresponding clusters:

$$J = \sum_{k=1}^{K} \sum_{\boldsymbol{x}_i \subset C_k} d^2(\boldsymbol{x}_i, \boldsymbol{c}_k) = \sum_{k=1}^{K} \sum_{\boldsymbol{x}_i \subset C_k} \sum_{j=1}^{n}(x_{ij} - c_{kj})^2 \tag{28}$$

where C_k denotes the set of samples in the k-th cluster, and K is the preset number of clusters.

(3) Centroid Update Formula

The cluster centroid is updated to the mean value of samples within the cluster after each iteration:

$$\boldsymbol{c}_k = \frac{1}{|C_k|} \sum_{\boldsymbol{x}_i \subset C_k} \boldsymbol{x}_i \tag{29}$$

where $|C_k|$ represents the number of samples in the k-th cluster.

3.1.2 Z-Score Normalization

As a mainstream preprocessing approach, Z-Score normalization[30] unifies data dimensions and scales. When applied by working conditions, it eliminates condition-related distribution shift and optimizes feature performance. It also stabilizes DEQ iteration, speeds up gradient optimization, and fits network modules including attention and normalization layers. Its core function is to map original data to a standard normal distribution. The formula and detailed explanations are given as follows:

For an arbitrary sample x_i in the dataset, the calculation formula of the normalized result z_i via Z-Score is:

$$z_i = \frac{x_i - \mu}{\sigma} \tag{30}$$

where z_i denotes the normalized result of the original sample x_i; x_i represents the raw data sample to be normalized; μ is the mean value of the entire dataset in this feature dimension; σ is the standard deviation of the entire dataset in this feature dimension, which is adopted to measure the dispersion degree of the data.

3.1.3 Feature Selection Based on Random Forest

High-dimensional multi-sensor data for aero-engine RUL prediction contains redundant features and noise, bringing higher computational overhead, overfitting risks, and lower prediction precision. Random Forest[31] (RF) is adopted to select effective features.

RF evaluates feature importance via the Gini coefficient to extract degradation-related core features, and its dual randomness improves noise resistance to adapt to complex engine data. It can directly process high-dimensional heterogeneous data without complicated preprocessing. The optimized features further improve model prediction performance. Feature importance calculation is the key procedure of RF-based feature selection. The calculation principle of feature importance for regression tasks is illustrated in Fig. 5.

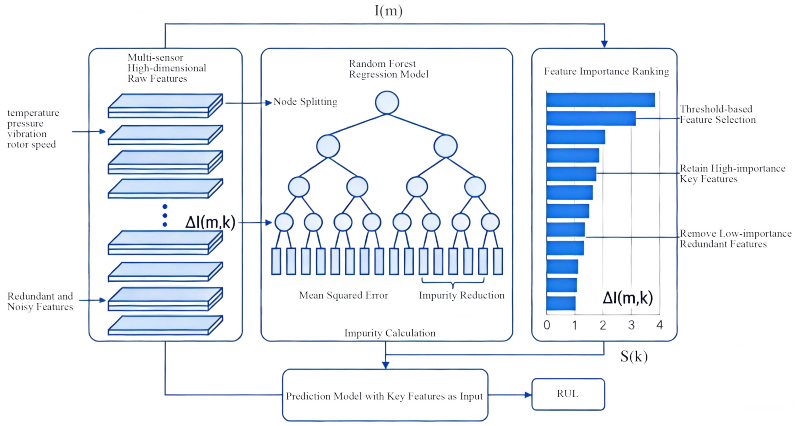


Fig. 5: Schematic Diagram of Feature Importance Calculation

(1) Node Impurity and Splitting Criterion of Regression Trees

A random forest consists of multiple decision trees. For regression trees, the *node impurity* is measured by the mean squared error (MSE):

$$I(m) = \frac{1}{N_m} \sum_{i \in m} \left(y_i - \bar{y}_m\right)^2 \tag{31}$$

where $I(m)$ denotes the impurity of node m, N_m is the number of samples contained in node m, i represents the index of a single sample within node m, y_i is the target variable value of the i-th sample in node m, and $\bar{y}_m$ is the mean value of the target variables for all samples in node m.

During the construction of a regression tree, the feature that yields the maximum reduction in node impurity is selected for node splitting. When node m is split into a left child node m_L and a right child node m_R via feature k, the reduction of node impurity is calculated as:

$$\Delta I(m,k) = I(m) - \frac{N_{m_L}}{N_m} I(m_L) - \frac{N_{m_R}}{N_m} I(m_R) \tag{32}$$

where $\Delta I(m,k)$ is the impurity reduction obtained by splitting node m with feature k; m_L and m_R denote the left and right child nodes generated after splitting node m

using feature k, respectively; N_{m_L} and N_{m_R} are the number of samples contained in the left child node m_L and right child node m_R, respectively.

(2) Feature Importance Calculation for Random Forest

For a single regression tree, the importance of feature k is defined as the sum of impurity reductions over all nodes split by this feature:

$$S(k, \text{tree}) = \sum_{m \in \{\text{split nodes of feature } k\}} \Delta I(m, k) \tag{33}$$

Given a random forest composed of T decision trees, the average importance of feature k across all trees is formulated as:

$$\bar{S}(k) = \frac{1}{T} \sum_{t=1}^{T} S(k, \text{tree}_t) \tag{34}$$

where $\bar{S}(k)$ is the average importance of feature k in the random forest, T is the total number of decision trees in the forest, t is the index of a single decision tree, and tree_t represents the t-th decision tree in the random forest.

The average importance values of all features are further normalized to obtain the final feature importance scores. A higher feature importance score indicates that the corresponding feature contributes more to reducing the prediction error of RUL and has a more significant impact on RUL prediction.

(3) Threshold Setting

Features with importance scores lower than the preset threshold are identified as low-importance features. The statistical characteristics of these low-importance features are further verified. The results show that the values of these features barely change and contribute little to the RUL prediction task. Finally, such redundant features are removed to reduce data redundancy and lower the computational complexity of the model.

3.1.4 Exponential Moving Average

In aero-engine RUL prediction, sensor parameters such as temperature, pressure, and rotational speed are vulnerable to extreme operating conditions, electromagnetic interference, and equipment noise, generating random fluctuations and abnormal data spikes. Direct use of noisy raw data will obscure real engine degradation trends, resulting in incorrect feature extraction and reduced prediction accuracy.

Exponential Moving Average[32] (EMA) is adopted to suppress sensor noise. It dynamically balances data timeliness and smoothness via exponential weighting: recent data are given higher weights to reflect real-time degradation status, while historical data get gradually reduced weights. This design retains valid long-term degradation information and avoids interference from outdated data.

In engine time-series processing, EMA removes data fluctuations and abnormal spikes caused by complex working conditions and external interference. It smooths degradation curves and highlights gradual performance degradation rules, helping the subsequent model capture subtle and long-term degradation characteristics accurately.

Furthermore, EMA denoised data provides stable inputs for subsequent working condition classification, feature selection, and RUL prediction. It eliminates noise interference throughout the modeling process, and effectively improves the robustness and generalization capability of the prediction model under complex engineering conditions. The calculation formula of EMA is expressed as:

$$s_{\text{smoothed},t} = \frac{\beta \cdot s_{\text{smoothed},t-1} + (1-\beta) \cdot s_t}{1-\beta^t} \tag{35}$$

where $\beta = 0.98$ is defined as the smoothing coefficient, which is adopted to suppress the interference of high-frequency sensor noise on the degradation trend.

3.1.5 Sliding Time Window for Sample Construction

RUL prediction of aero-engines is a core technology for guaranteeing the condition-based maintenance and safe operation of aero-equipment. Its primary task is to establish the mapping relationship between state features and remaining service life based on time-series degradation data collected by multiple sensors, which is essentially a typical time-series regression problem.

Restricted by the input dimension of deep learning models, the long-term time-series sensor data covering the full service life cycle of aero-engines cannot be directly adopted for model training. To address this issue, this paper introduces the Sliding Time Window[33] (STW) technique to convert the original long sequences into standardized supervised learning samples. The specific construction principle is illustrated in Fig. 6.

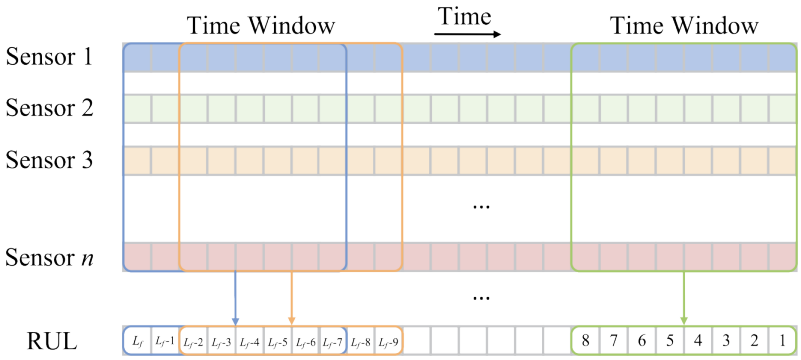


Fig. 6: Schematic Diagram of Sliding Time Window Construction

Aero-engines are equipped with n monitoring sensors to collect full-life operational data, forming a $T \times n$ raw time-series dataset, where T refers to the total operating cycles of a single engine. Each row represents the full-life monitoring sequence of one engine, and each column denotes sensor observations at a single time step.

A fixed-length sliding window with window size W and step size $s = 1$ is adopted to generate supervised samples. Each extracted $W \times n$ subsequence serves as model

input, while the corresponding RUL value at the last time step is taken as the sample label. Three colored windows correspond to three engine degradation stages: blue for healthy early stage with maximum RUL labels (L_f), orange for moderate degradation stage with medium RUL labels, and green for near-failure late stage with minimum RUL labels.

The sliding window divides long time-series data into abundant supervised samples with complete temporal correlation, matching the batch input requirement of deep learning models and capturing full-life degradation evolution of aero-engines. Besides, Piece-wise Linear RUL (PWLR) labeling strategy is employed for label construction. RUL maintains the maximum value L_f before the cycle threshold T_l in the healthy stage, and declines linearly until failure afterward. This strategy conforms to actual engine degradation patterns and eliminates useless RUL fluctuations in the healthy stage during training.

Compared with full-sequence input, the sliding time window has three core merits: first, it reduces input dimension and network computation cost to boost model training and inference efficiency; second, it augments time-series samples to enrich data diversity and relieve overfitting under limited data; third, it extracts local temporal degradation features to improve the timeliness and accuracy of RUL prediction.

3.2 Calculation of DAI and Dual-Input Attention Module

The DAI quantifies the synchronization degree between sensor degradation trends and monotonic RUL decay trends. To integrate such degradation prior knowledge into feature extraction, this paper designs a dual-input attention module guided by DAI. This module adaptively enhances features highly correlated with RUL and suppresses invalid noise by embedding DAI as dynamic attention constraints, and its implementation process is illustrated below.

First, the original DAI scalar value is mapped into a normalized attention gating coefficient via linear transformation and Sigmoid activation function, as presented in (36):

$$g = \sigma\big(W_{da} \cdot \mathrm{DAI} + b_{da}\big) \tag{36}$$

where W_{da} and b_{da} denote learnable weight and bias, respectively; σ refers to the Sigmoid function that restricts the gating coefficient g to the range $[0, 1]$. A larger g means a higher matching degree between sensor degradation and RUL attenuation, thus bringing larger attention weights to corresponding sensor features.

Different from standard attention structures, the proposed module takes two types of feature inputs: deep hidden features X_1 output by the deep equilibrium network (DEQ) and raw sensor measurement features X_2. The query, key, and value vectors are obtained through linear projection:

$$Q = W_Q X_1, \quad K = W_K X_2, \quad V = W_V X_2 \tag{37}$$

where Q built from DEQ hidden features captures global degradation information, while K and V are projected from original sensor features. Three learnable matrices W_Q, W_K, and W_V unify the feature dimension of three vectors for subsequent similarity calculation.

After acquiring standard attention weights, the DAI gating coefficient is adopted to correct attention distribution dynamically. The final attention map and output features are calculated by:

$$\text{Attn} = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right) \odot g \tag{38}$$

$$X_{\text{out}} = W_O\big(\text{Attn} \cdot V\big) + X_1 \tag{39}$$

In (38), $QK^\top$ computes cross-feature similarity, and $\sqrt{d_k}$ serves as a scaling factor to prevent gradient vanishing. The gating coefficient g modulates attention intensity by element-wise multiplication. In (39), residual connection is adopted to fuse weighted sensor features and original DEQ deep features, which avoids the loss of global degradation information. W_O is the output projection matrix for dimension matching.

In summary, the dual-input attention module combines equipment physical degradation rules with data-driven attention mechanism. It makes the network focus more on reliable degradation features matched with real RUL decay rules, so as to further improve the accuracy of RUL prediction.

3.3 Deep Equilibrium Feature Extraction Module Based on Anderson Iteration

In this work, the DEQ model is employed as the backbone feature extraction network. It replaces the conventional deep stacking architecture with fixed-point iteration, which fundamentally mitigates the gradient vanishing problem.

(1) Design of Deep Equilibrium Layer (DELayer)

The deep equilibrium layer constructs the nonlinear transformation function f_θ using one-dimensional convolution and group normalization:

$$x_0 = \text{ReLU}\big(\text{Conv}_{1\times1}(x)\big) \tag{40}$$

$$x_1 = \text{ReLU}\big(\text{Conv}_{1\times3}(x_0)\big) \tag{41}$$

$$x_2 = \text{ReLU}\big(\text{Conv}_{1\times3}(x_1)\big) \tag{42}$$

$$x_{\text{concat}} = \text{concat}(x_1, x_2) \tag{43}$$

$$x_{\text{norm}} = \text{GroupNorm}(x_{\text{concat}}) \tag{44}$$

$$f_\theta(z, x) = \text{GroupNorm}\Big(\text{ReLU}\big(\text{DualInputAttention}(z, x_{\text{norm}})\big)\Big) \tag{45}$$

where z denotes the latent fixed-point state and x is the network input. The transformation function f_θ participates in the fixed-point iteration of the DEQ model.

(2) Solution of Fixed-Point Iteration via Anderson Acceleration

In the forward propagation of DEQ, the Anderson accelerated iteration is adopted to solve the fixed-point equation:

$$z^* = f_\theta(z^*, x) \tag{46}$$

The detailed iteration procedures are presented as follows:

(a) Initialize $z^{(0)} = \mathbf{0}$, and construct the historical iteration matrices $\boldsymbol{X}$ and $\boldsymbol{F}$.
(b) Formulate the least-squares optimization objective:

$$\min_{\boldsymbol{\alpha}} \left\|\boldsymbol{G}\boldsymbol{\alpha}\right\|_2^2 + \lambda\left\|\boldsymbol{\alpha}\right\|_2^2 \tag{47}$$

(c) Perform iterative update:

$$z^{(k+1)} = \beta \sum \alpha_i \boldsymbol{F}_i + (1-\beta)\sum \alpha_i \boldsymbol{X}_i \tag{48}$$

(d) The iteration converges when the residual satisfies $\left\|z^{(k+1)} - f_\theta\left(z^{(k+1)}, x\right)\right\|_2 < 10^{-4}$, and the fixed point z^* is output.

(3) Implicit Gradient Back Propagation

The DEQ model leverages the implicit function theorem for back propagation, which eliminates the need to store intermediate results generated in the forward pass:

$$\begin{aligned} \left(\boldsymbol{I} - \frac{\partial f_\theta}{\partial z^*}\right)\frac{\partial z^*}{\partial \theta} &= \frac{\partial f_\theta}{\partial \theta} \\ \frac{\partial \mathcal{L}}{\partial \theta} &= \boldsymbol{g}^\top \left(\boldsymbol{I} - \boldsymbol{J}\right)^{-1}\frac{\partial f_\theta}{\partial \theta} \end{aligned} \tag{49}$$

where $\boldsymbol{J}$ represents the Jacobian matrix and $\boldsymbol{g} = \partial\mathcal{L}/\partial z^*$. The conjugate gradient method is utilized to solve the linear system, which realizes the decoupling between memory overhead and network depth.

3.4 Adversarial Domain Adaptation Module

To alleviate the distribution shift between the source domain and the target domain, this paper establishes an adversarial domain adaptation framework based on the Gradient Reversal Layer (GRL), as shown in Fig. 7.

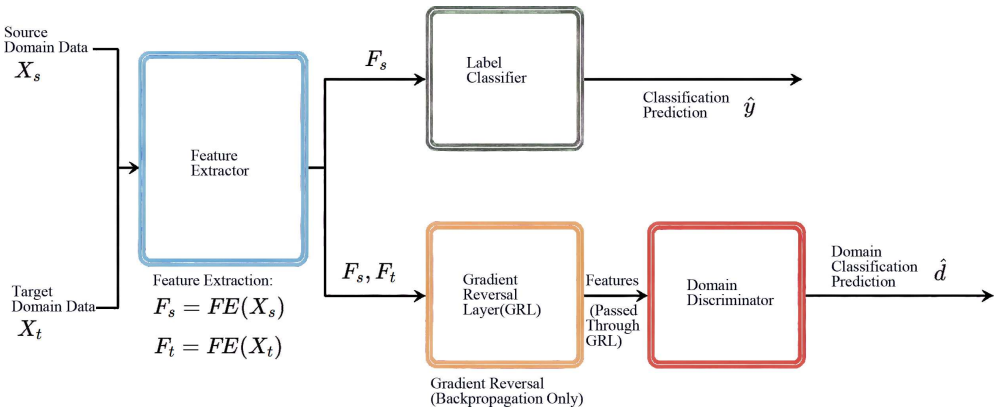


Fig. 7: Schematic Diagram of GRL-based Adversarial Domain Adaptation Framework

(1) Gradient Reversal Layer

The Gradient Reversal Layer performs identity mapping in the forward propagation, while reversing the gradient sign during back propagation:

$$\mathrm{GRL}(x) = x \tag{50}$$

$$\frac{\partial \mathcal{L}}{\partial x} = -\lambda \cdot \frac{\partial \mathcal{L}}{\partial \mathrm{GRL}(x)} \tag{51}$$

where λ denotes the gradient reversal coefficient. This mechanism realizes the adversarial game between the feature extractor and the domain discriminator.

(2) Domain Discriminator

The domain discriminator is implemented as a fully-connected network to distinguish the domain source of input features:

$$\begin{aligned} h_1 &= \mathrm{ReLU}\big(\mathrm{BN}(W_1 z + b_1)\big), \\ h_2 &= \mathrm{ReLU}\big(\mathrm{BN}(W_2 h_1 + b_2)\big), \\ D(z) &= \sigma(W_3 h_2 + b_3) \end{aligned} \tag{52}$$

where z represents the output feature of the DEQ module, and $D(z) \in [0, 1]$ is the predicted domain probability.

(3) Domain Adversarial Loss

The binary cross-entropy is adopted as the domain adversarial loss:

$$L_{\mathrm{domain}} = -\frac{1}{N} \sum_{i=1}^{N} \Big[d_i \log D(\tilde{z}_i) + (1 - d_i) \log\big(1 - D(\tilde{z}_i)\big) \Big] \tag{53}$$

where d_i is the domain label ($d_i = 0$ for the source domain and $d_i = 1$ for the target domain), and $\tilde{z}$ denotes the feature processed by the GRL.

3.5 Bayesian RUL Prediction and Uncertainty Quantification Module

This paper employs a Bayesian Neural Network (BNN) as the prediction head, and implements uncertainty quantification via Monte Carlo Dropout.

(1) Bayesian Prediction Network

The formulation of the prediction network is written as:

$$\begin{aligned} h_1 &= \mathrm{ReLU}\big(W_1 z_{\mathrm{pool}} + b_1\big), \\ h_2 &= \mathrm{ReLU}\big(W_2 h_1 + b_2\big), \\ \hat{y} &= W_3 h_2 + b_3. \end{aligned} \tag{54}$$

where z_{pool} is the adaptive pooling result of DEQ features. Dropout is activated in both training and testing phases.

(2) Monte Carlo Sampling and Uncertainty Calculation
During the testing phase, $T = 100$ stochastic forward samplings are performed:

$$\begin{aligned} \bar{\hat{y}} &= \frac{1}{T}\sum_{i=1}^{T}\hat{y}_i, \\ \sigma^2 &= \frac{1}{T}\sum_{i=1}^{T}\left(\hat{y}_i - \bar{\hat{y}}\right)^2. \end{aligned} \tag{55}$$

In the above equations, $\bar{\hat{y}}$ denotes the final predicted RUL value, and σ^2 represents the variance of prediction uncertainty, which provides a reliability metric for decision-making.

4 Experimental Analysis

4.1 Introduction to C-MAPSS Dataset

The experiments adopt the Commercial Modular Aero-Propulsion System Simulation (C-MAPSS) dataset developed and released by the National Aeronautics and Space Administration (NASA). As a widely recognized standard benchmark dataset for RUL prediction of aero-engines, it contains full-lifecycle degradation time-series data of engines under diverse operating conditions and multiple failure modes. Featuring complete data structure and realistic engineering scenarios, this dataset has become a mainstream benchmark to evaluate the performance of time-series modeling and life prediction algorithms worldwide.

The C-MAPSS dataset consists of four subsets with distinct operating characteristics, which are the primary cause of severe domain distribution shift between the source domain and target domain in cross-domain RUL prediction. Specifically, subsets FD001 and FD003 simulate single stable operating conditions, where the engines operate with constant load, rotational speed, and other operational parameters throughout the whole service life, and thus can be adopted as the source domain for cross-domain prediction. In contrast, subsets FD002 and FD004 cover six combinations of operating conditions, involving frequent load switching, rotational speed adjustment, and other variable operating scenarios. Their data distributions differ significantly from those of the single-condition subsets, so they are regarded as the target domain. The differences between the two types of operating conditions essentially lead to domain distribution shift. The detailed operating conditions are presented in Table 1.

The C-MAPSS dataset divides the entire operating process of aero-engines from takeoff to landing into six typical operating conditions according to flight mission profiles. It defines the corresponding flight scenarios, altitude ranges, Mach numbers, and throttle settings for each condition, which clearly illustrates the environmental conditions and power configurations of engines at different flight phases and intuitively reflects the prominent heterogeneity of data distribution under multiple operating conditions. Such operating-condition-driven distribution discrepancies further exacerbate the domain distribution shift in cross-domain prediction. Meanwhile, the dataset

Table 1: Operating conditions of aero-engine

Operating Condition	Flight Scenario	Altitude (ft)	Mach Number	Throttle Setting
1	Ground Takeoff	0	0–0.2	High
2	Climb Phase	5000–15000	0.3–0.6	Medium-High
3	Cruise Preparation	20000–30000	0.7–0.85	Medium
4	Cruise Phase	35000	0.8–0.84	Low
5	Descent Phase	15000–5000	0.5–0.7	Low
6	Landing Idle	0	0–0.1	Minimum

provides a typical cross-domain validation scenario for the adversarial domain adaptation framework proposed in this work, ensuring that the experimental setup conforms to the practical engineering requirements of cross-condition and cross-domain RUL prediction.

The raw data of the C-MAPSS dataset consists of 26 columns, including unit ID, operating cycles, operational parameters, and 21 sensor measurements. The detailed description of each sensor is presented in Table 2. Due to the complex scenarios with multiple operating conditions, the operational parameters collected by sensors, such as temperature, pressure, and rotational speed, exhibit obvious multimodal characteristics. The strong nonlinear coupling of parameter sequences under different conditions further enlarges the distribution discrepancy across domains and aggravates domain distribution shift. In addition, sensor data is susceptible to interference from environmental noise and operating condition switching, leading to severe temporal fluctuation and modal discrepancy. These issues greatly increase the difficulty of cross-domain feature alignment, which is the fundamental motivation for proposing the multi-condition clustering normalization preprocessing method and the adversarial domain adaptation framework in this study. The proposed methods are targeted to address the critical problem of domain distribution shift in cross-domain prediction.

The mixed data structure with multiple operating conditions results in highly diverse engine degradation patterns. Cross-domain RUL prediction thus faces multiple technical challenges, including severe parameter interference and implicit degradation rules. Moreover, the negative impacts of domain distribution shift become more prominent in cross-domain scenarios. Data heterogeneity among different domains will degrade the generalization performance of conventional models, which puts forward higher requirements on the generalization ability, robustness, and domain alignment capability of the prediction model. These inherent characteristics of the dataset also serve as the practical basis for designing a series of preprocessing strategies and evaluation metrics in this paper, including multi-condition clustering normalization, EMA smoothing, Random Forest-based redundant feature elimination, and the DAI. These approaches are adopted to mitigate domain distribution shift, realize effective feature alignment between the source domain and target domain, and ultimately improve the accuracy of cross-domain RUL prediction.

Table 2: Description of sensor signals

No.	Symbol	Description
S1	T2	Total temperature at fan inlet
S2	T24	Total temperature at low-pressure compressor inlet
S3	T30	Total temperature at high-pressure compressor inlet
S4	T50	Total temperature at low-pressure turbine outlet
S5	P2	Fan inlet static pressure
S6	P15	Bypass duct total pressure
S7	P30	High-pressure compressor outlet total pressure
S8	Nf	Physical fan rotational speed
S9	Nc	Physical core engine rotational speed
S10	Epr	Engine pressure ratio
S11	Ps30	Static pressure at high-pressure compressor outlet
S12	Phi	Fuel flow ratio
S13	NRf	Corrected fan rotational speed
S14	NRc	Corrected core engine rotational speed
S15	BPR	Bypass ratio
S16	FarB	Combustor fuel-air ratio
S17	htBleed	Bleed air enthalpy
S18	Nf_dmd	Demanded fan rotational speed
S19	PCNfR_dmd	Demanded corrected fan rotational speed
S20	W31	Cooling air flow rate of high-pressure compressor
S21	W32	Cooling air flow rate of low-pressure compressor

Nevertheless, the DAI relies on real RUL labels for calculation in the C-MAPSS dataset. For the test set, adopting authentic RUL labels to compute DAI will cause severe future data leakage. Although this operation can improve the prediction accuracy superficially, the optimized prediction results lack practical credibility. According to the inherent characteristics of the C-MAPSS dataset, the RUL value decreases by 1 sequentially with the advancement of time series. In the whole engine degradation stage, the slope of RUL variation remains fixed at -1. Therefore, adopting this fixed slope value for DAI calculation can effectively avoid future data leakage in cross-domain prediction tasks.

4.2 Evaluation Indicators

In this paper, three quantitative indicators including Root Mean Square Error (RMSE), official Score function, and Average Symbol Uncertainty Error (ASUE) are adopted to quantitatively evaluate the RUL prediction performance.

(1) RMSE

RMSE measures the overall deviation between predicted RUL values and real RUL values, which is the most critical evaluation indicator in RUL prediction tasks. A

smaller RMSE value means higher prediction accuracy. Its definition is expressed as:

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\hat{y}_i - y_i\right)^2} \tag{56}$$

(2) Score

The Score function is the official evaluation metric of the C-MAPSS dataset. It imposes a much heavier penalty on late prediction (predicted RUL is larger than the ground-truth RUL) than early prediction, which is more consistent with practical engineering application scenarios of aero-engines. Lower Score value indicates better prediction performance. The specific formula is defined as:

$$\mathrm{Score} = \sum_{i=1}^{N} \begin{cases} \exp\left(-\dfrac{\hat{y}_i - y_i}{13}\right) - 1, & \hat{y}_i - y_i < 0 \\ \exp\left(\dfrac{\hat{y}_i - y_i}{10}\right) - 1, & \hat{y}_i - y_i \geq 0 \end{cases} \tag{57}$$

(3) ASUE

Average Symbol Uncertainty Error (ASUE) is used to evaluate the uncertainty and deviation direction of prediction results. The closer the ASUE value is to 0, the smaller the systematic deviation of prediction results. Its calculation formula is shown below:

$$\mathrm{ASUE} = \frac{1}{N}\sum_{i=1}^{N}\frac{\hat{y}_i - y_i}{y_i} \tag{58}$$

4.3 Parameter Settings

The core parameter settings of the proposed model are listed in Table 3.

The hardware and software environment for simulation experiments is presented in Table 4.

4.4 Analysis and Comparison of RUL Prediction Results

4.4.1 Analysis of RUL Prediction Results with DAI Module on Single Dataset

First, the prediction framework integrated with DEQ and DAI modules is adopted for model training. The NASA C-MAPSS dataset preprocessed by multi-condition clustering normalization, exponential moving average smoothing, and redundant feature elimination is used as the experimental dataset. Anderson fixed-point iteration is adopted to realize end-to-end training of the deep equilibrium feature extraction network, which enables the model to sufficiently learn the inherent mapping relationship between engine degradation characteristics and RUL values. Afterwards, the test set

Table 3: Model parameter settings

Parameter Category	Parameter Name	Value
Basic Configuration	Random Seed	42
Training Hyperparameters	Maximum Epochs	35
	Learning Rate	1e-3
	Training Batch Size	256
	Validation Batch Size	128
	Test Batch Size	100
	StepLR Scheduler	step_size=10, gamma=0.5
Data Preprocessing	RUL Clipping Value	125
	Time Window Length	15
	Smoothing Coefficient	0.98
	Feature Screening Threshold	1e-3
	Training Set Division Ratio	0.8
	Monte Carlo Sampling Times	100
Model Structure	Initial Channel Number	15
	Hidden Unit Number	15
	Dropout Rate	0.4
	DEQ Convergence Threshold	1e-4
	Maximum DEQ Iterations	200
Domain Adaptation Module	Domain Loss Weight	0.1
	Gradient Reversal Coefficient	1.0
	Discriminator Hidden Dimension	64
	Discriminator Dropout Rate	0.3

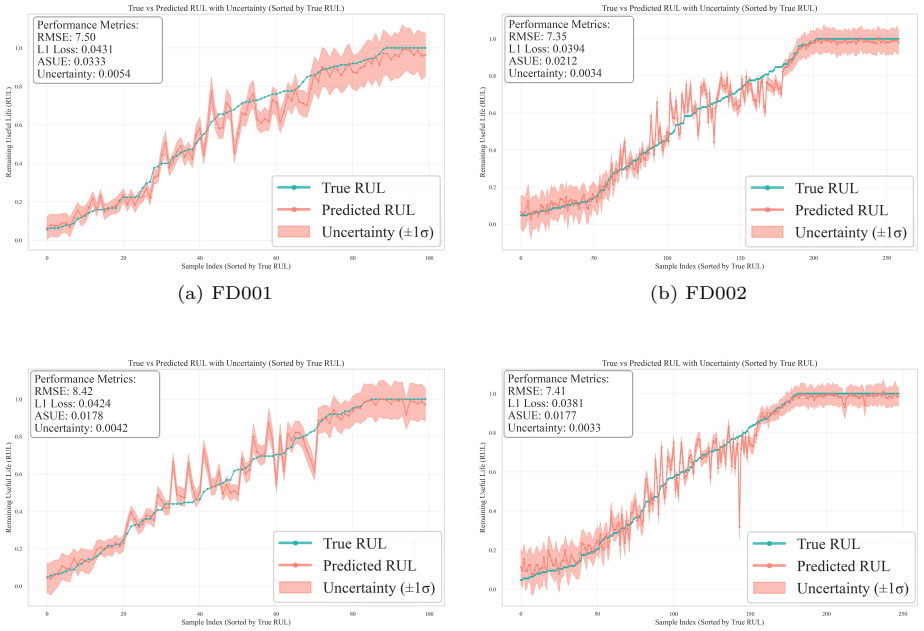
Table 4: Simulation experimental environment configuration

Item	Configuration
Operating System	Windows 11 Home Chinese Version 24H2
Processor	AMD Ryzen 9 7845HX
Memory	32.0 GB
Graphics Card	NVIDIA GeForce RTX 4070 Laptop GPU
Programming Language	Python
Deep Learning Framework	torch 2.3.0

of the C-MAPSS dataset is adopted to verify the prediction performance of the proposed framework, and the improvement effect of the introduced DAI module on RUL prediction accuracy is emphatically analyzed in subsequent experiments.

The prediction results are presented in Fig. 8 and Table 5.

In this figure, the horizontal axis represents the unit number of tested aero-engines. To clearly observe the fitting degree between predicted RUL values and ground-truth RUL values, all engine units are arranged from left to right in ascending order of predicted RUL values. The vertical axis denotes the number of RUL cycles, which directly reflects the remaining operating capacity of aero-engines. The green line refers to the ground-truth RUL values of engines, while the orange line represents the RUL prediction values output by the proposed model. The prediction accuracy and curve fitting performance of the model can be directly evaluated by comparing the changing trends of the two curves.



(a) FD001 (b) FD002

(c) FD003 (d) FD004

Fig. 8: Data distribution of four C-MAPSS subsets under different flight conditions

It can be clearly observed from the figure that the trend of the orange prediction curve is highly consistent with that of the green ground-truth curve. The two curves maintain a satisfactory fitting effect both in the early engine degradation stage and the late stage near the end of service life. This result demonstrates that the DAI module can effectively guide the model to focus on highly correlated degradation features, and the DEQ network is capable of fully mining inherent engine degradation patterns. Accordingly, the proposed method improves the RUL prediction accuracy under single dataset scenarios, which is consistent with the model design objectives illustrated in previous sections.

Table 5: RUL Prediction Results of DAI-DEQ Model on Single Dataset

Dataset	RMSE	L1 Loss	ASUE	Score	Uncertainty
FD001	7.50	0.0431	0.0333	74.60	0.0054
FD002	7.35	0.0394	0.0212	219.98	0.0034
FD003	8.42	0.0424	0.0178	120.67	0.0042
FD004	7.41	0.0381	0.0177	268.49	0.0033

It can be concluded from the results in Table 5 that the proposed method achieves excellent RUL prediction performance and strong robustness on all four subsets of

the C-MAPSS dataset. The RMSE values of all subsets range from 7.35 to 8.42, and all L1 Loss values are lower than 0.0431. Specifically, FD002 obtains the optimal prediction accuracy, while FD003 presents the largest prediction error due to its more complex degradation patterns. The ASUE value decreases with the increase of working condition complexity and fault complexity, and the ASUE of FD004 is only 0.0177, which proves the outstanding prediction reliability of the proposed model in critical service life intervals.

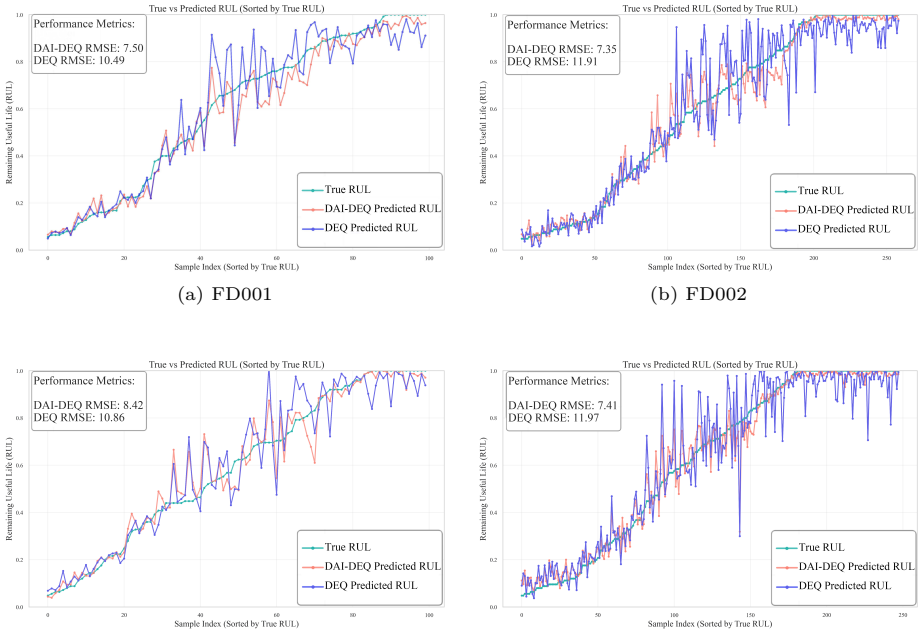
The PHM Score follows the order of FD004 ¿ FD002 ¿ FD003 ¿ FD001, which is consistent with the universal conclusion that datasets with diverse working conditions and compound faults bring higher prediction difficulty. Even under challenging experimental scenarios, the proposed model still maintains satisfactory generalization ability. In addition, the prediction uncertainty of all subsets remains at an extremely low level between 0.0033 and 0.0054. Notably, higher complexity of working conditions and faults corresponds to lower prediction uncertainty. This benefit stems from the DAI module, which realizes quantitative weighting for the synchronization between sensor degradation trends and RUL attenuation processes. It effectively highlights high-information degradation features and suppresses irrelevant noise interference. Overall, the comparative results verify the comprehensive advantages of the proposed method in prediction accuracy, reliability, and stability under diverse operating conditions and fault modes.

To further explore the core contribution of the DAI module, ablation experiments are conducted in this study. All experimental settings are consistent with the previous single-dataset experiments, except that the DAI module is completely removed from the original framework. Other experimental hyperparameters and data preprocessing strategies remain unchanged. The prediction performance discrepancy between the original DAI-DEQ model and the baseline model without the DAI module is compared comprehensively.

The prediction results of ablation experiments are presented in Fig. 9 and Table 6. The settings of horizontal axis, vertical axis, and arrangement rule of engine units are consistent with previous experiments. The green curve represents the ground-truth RUL values, the orange curve denotes the prediction results of the original DAI-DEQ model, and the newly added purple curve corresponds to the outputs of the baseline model without the DAI module. The comparison among three curves clearly reveals the improvement effect of the DAI module on model prediction accuracy.

Table 6: Comparative RUL Prediction Results of DAI-DEQ Model and DEQ Model on Single Dataset

Dataset	DAI-DEQ RMSE	DEQ RMSE
FD001	7.50	10.49
FD002	7.35	11.91
FD003	8.42	10.86
FD004	7.41	11.97



(a) FD001 (b) FD002

(c) FD003 (d) FD004

Fig. 9: RUL Prediction Comparison Results between DAI-DEQ Model and DEQ Model on Single Dataset

It can be clearly observed from the ablation experimental curves that the fitting degree between the baseline DEQ model without the DAI module (purple curve) and the ground-truth curve (green curve) decreases significantly. Its prediction error is much larger than that of the DAI-DEQ model, and such error becomes more prominent from the middle to the late stage of engine degradation. This result verifies the vital contribution of the DAI module. The DAI module can guide the model to focus on highly correlated degradation features and eliminate interference from irrelevant features, so as to guarantee stable prediction accuracy, which is consistent with the conclusions obtained in previous experiments.

Combined with the quantitative results of ablation experiments, the influence of the DAI module on model prediction performance is further analyzed by comparing the RMSE indicators between the DAI-DEQ model and the original DEQ baseline model without the DAI module, so as to verify the necessity of the DAI module. Overall, the RMSE values of the DEQ baseline model are significantly higher than those of the DAI-DEQ model across all four datasets, with the RMSE difference ranging from 2.44 to 4.56. This quantitative result is consistent with the curve analysis, proving that removing the DAI module will greatly increase model prediction error and highlighting the core status of the DAI module in the proposed framework.

Detailed analysis based on separate datasets is conducted as follows. The RMSE difference between the DEQ baseline model and the DAI-DEQ model reaches 4.56 on

both FD002 and FD004 datasets, which is the maximum difference among all test subsets. It indicates that the absence of the DAI module leads to severe performance degradation under scenarios with complex sensor features, further demonstrating the critical capacity of the DAI module in screening complex degradation features. For FD001 and FD003 datasets, the RMSE differences are 2.99 and 2.44, respectively. Although the gaps are relatively smaller, the DEQ baseline model still presents much higher prediction errors than the DAI-DEQ model. This reveals that prediction performance will decline inevitably even on datasets with relatively simple working conditions once the DAI module is removed. The relatively small error gap on FD003 is speculated to be caused by its complex working conditions and volatile degradation characteristics. Nevertheless, the DEQ model without the DAI module still cannot achieve the prediction performance of the complete DAI-DEQ model.

In conclusion, the graphical results and quantitative indicators of ablation experiments mutually validate the necessity and core function of the DAI module. After removing the DAI module, the DEQ baseline model suffers from increased prediction errors, fails to capture effective degradation features and suppress noise interference, resulting in deteriorated overall prediction performance. This conclusion confirms that the DAI module is an indispensable component for ensuring favorable prediction accuracy. The embedded DAI module enables the model to better capture engine degradation patterns and maintain stable prediction performance, which also lays a solid foundation for subsequent optimization of cross-domain prediction models.

4.4.2 Analysis of RUL Prediction Results of Adaptive DAI-DEQ Model in Cross-Domain Scenarios

For cross-domain prediction of the adaptive module, the unlabeled target domain prediction set and labeled source domain training set are input into the model for joint training. Leveraging the adversarial domain adaptation framework consisting of the Gradient Reversal Layer (GRL) and domain discriminator, we align domain feature distributions to alleviate cross-domain distribution shift and enhance the model's cross-domain prediction capability.

To investigate the performance of the adaptive module under various cross-domain conditions, we compare the DAI-DEQ model and the adaptive DAI-DEQ model using experimental data from multiple source-to-target domain cross-domain tasks with four evaluation metrics: RMSE, L1 loss, ASUE, and Score. This work analyzes the impact of data complexity discrepancy between source and target domains on the adaptive module, and identifies its applicable scenarios and the underlying reasons for adverse effects. The detailed experimental results are presented in Table 7.

Experimental Result Analysis

(1) Simple and clean source domain versus target domain with complex noise and diverse operating conditions: positive effects of the adaptive module

Taking FD001 (simple and clean dataset) as the source domain for cross-domain prediction on FD002, FD003, and FD004 (datasets with complex noise and diverse operating conditions), the adaptive DAI-DEQ model substantially outperforms the base DAI-DEQ model, demonstrating the effectiveness of the adaptive module.

Table 7: Cross-domain Prediction Comparison

Cross-domain Scenario	Model	RMSE	L1 Loss	ASUE	Score
FD001→FD002	DAI-DEQ	49.85	0.3260	0.0541	387456.91
	Adaptive DAI-DEQ	12.90	0.0752	0.0213	745.34
FD001→FD003	DAI-DEQ	18.53	0.0945	0.0732	658.77
	Adaptive DAI-DEQ	14.67	0.0888	0.0248	402.64
FD001→FD004	DAI-DEQ	52.26	0.3299	0.0413	291977.33
	Adaptive DAI-DEQ	14.26	0.0899	0.0409	883.25
FD002→FD001	DAI-DEQ	27.28	0.1615	0.0083	658.77
	Adaptive DAI-DEQ	24.79	0.1403	0.0501	25141.58
FD002→FD003	DAI-DEQ	47.84	0.2634	0.0038	291977.33
	Adaptive DAI-DEQ	36.97	0.2143	0.0562	124386.88
FD002→FD004	DAI-DEQ	21.91	0.1224	0.0625	3269.72
	Adaptive DAI-DEQ	16.76	0.0880	0.0476	2060.75
FD003→FD001	DAI-DEQ	9.85	0.0496	0.0241	149.52
	Adaptive DAI-DEQ	30.40	0.1789	0.0076	9610.60
FD003→FD002	DAI-DEQ	42.68	0.3028	7721.05	7721.05
	Adaptive DAI-DEQ	18.63	0.1051	0.0307	7347.86
FD003→FD004	DAI-DEQ	45.51	0.3259	0.2240	7838.46
	Adaptive DAI-DEQ	10.90	0.0596	0.0320	551.36
FD004→FD001	DAI-DEQ	14.32	0.0836	0.0100	495.82
	Adaptive DAI-DEQ	29.36	0.1684	0.0543	20355.86
FD004→FD002	DAI-DEQ	7.66	0.0410	0.0200	218.97
	Adaptive DAI-DEQ	31.21	0.1779	0.0225	100023.40
FD004→FD003	DAI-DEQ	15.27	0.0902	0.0069	490.67
	Adaptive DAI-DEQ	28.86	0.1524	0.0337	54101.09

For core error metrics RMSE and L1 Loss (lower values mean higher accuracy): in FD001 → FD002, RMSE decreases from 49.85 to 12.90 and L1 Loss from 0.3260 to 0.0752, with reductions of 74.1% and 76.9%. Both metrics drop by 72.7% in FD001 → FD004. In FD001 → FD003, RMSE and L1 Loss fall from 18.53 to 14.67 and from 0.0945 to 0.0888, respectively, and the adaptive model still retains performance advantages.

Auxiliary metrics ASUE (prediction stability) and Score (comprehensive performance) also confirm the superiority of the adaptive model. The results reveal that the adaptive module can extract core features from simple and clean source data, adapt to noisy and diversified operating conditions in the target domain, and thus lower prediction errors and improve accuracy.

(2) Complex source domain versus simple and clean target domain: adverse effects of the adaptive module

When complex datasets (FD002, FD003, FD004) are used as source domains and the simple clean dataset FD001 as the target domain, the adaptive DAI-DEQ model presents inferior performance compared with the base model, and the adaptive module induces significant adverse effects.

In FD003 → FD001, the model's RMSE and L1 Loss increase to 30.40 and 0.1789 (increases of 208.6% and 260.7%), and Score rises from 149.52 to 9610.60. The two metrics increase by nearly 100% in FD004 → FD001, and the RMSE rises by 307.4% in FD004 → FD002.

Even for FD004 → FD003 (both domains are complex and the source domain is slightly more complicated), the RMSE increases from 15.27 to 28.86, leading to performance degradation. It is proven that the adaptive module will degrade model

performance if the source domain is more complex than the target domain, especially when the target domain data is simple and clean.

(3) Mechanism analysis of adverse effects

The adverse effects mainly stem from the incorrect learning and transfer of impurities (noise, irrelevant operating conditions, etc.) contained in complex source data during adaptive training. These impurities pollute the pure degradation features of the target domain, hinder the model from learning effective target-domain characteristics, and reduce prediction accuracy.

The adaptive module is designed to learn cross-domain distribution differences and optimize parameters for target domain adaptation. It works well when source data has clear core features. Nevertheless, complex source data causes the module to treat impurities as valid features and accumulate useless information.

Such redundant information forces the model to fit simple target data with complex data patterns, introducing extra noise and interfering with the identification of key degradation features. Consequently, prediction errors rise and overall performance declines, forming the adverse effects of the adaptive module.

To conclude, the performance of the adaptive module depends heavily on the data complexity of source and target domains. It improves prediction accuracy when the source domain is simple and the target domain is complex, but degrades performance due to interference from source-domain impurities when the source domain is complex and the target domain is simple. This finding guides further model optimization, and the adaptive module can be selectively enabled or adjusted according to cross-domain complexity differences.

4.4.3 Comparison Experiments

To further verify the superiority of the proposed method, we adopt RMSE and Score as evaluation metrics and compare the experimental results of the proposed model with those of representative published methods under two settings: single-dataset prediction and cross-domain prediction.

To evaluate the basic fitting capability and prediction accuracy of the proposed model on datasets with single operating conditions, we select a series of state-of-the-art models in transfer learning, time series prediction, and deep learning as baseline methods, including IMDSSN, KGHM, SIGRNNDWI, Dynamic Self-Learning Neural Network, SACN, TATFA-Transformer, SA-MSTCN-GFA, and DA-LSTM. The comparison results are presented in Table 8 and Table 9.

According to the results, in terms of the RMSE metric, both the proposed DAI-DEQ and adaptive DAI-DEQ achieve the lowest prediction errors across all four datasets FD001–FD004, significantly outperforming all baseline methods. Specifically, the RMSE values of DAI-DEQ reach 7.44, 7.31, and 7.41 on FD001, FD002, and FD004 respectively, and 8.42 on FD003. These results are far better than those of the state-of-the-art SA-MSTCN-GFA (10.62, 15.60, 10.57, 15.46) and DA-LSTM. Although the adaptive DAI-DEQ obtains slightly higher RMSE than DAI-DEQ on FD002 and FD004, its RMSE remains below 8.05. The overall prediction accuracy is still much higher than that of existing deep learning methods. It demonstrates that the deep

Table 8: Comparison of RMSE for RUL prediction of different models on single datasets

Model	FD001	FD002	FD003	FD004
IMDSSN[3]	12.14	17.40	12.35	19.78
KGHM[4]	13.18	13.25	13.54	19.96
SIGRNNDWI[6]	13.10	13.32	14.34	15.12
Dynamic Self-Learning Neural Network[7]	11.23	12.89	13.43	15.10
SACN[11]	12.80	19.87	14.57	24.77
TATFA-Transformer[8]	12.21	15.07	11.23	18.81
SA-MSTCN-GFA[9]	10.62	15.60	10.57	15.46
DA-LSTM[10]	12.62	13.22	13.34	16.25
DAI-DEQ	6.97	7.35	8.42	7.41
Adaptive DAI-DEQ	7.35	8.05	8.00	8.10

Table 9: Comparison of Score for RUL prediction of different models on single datasets

Model	FD001	FD002	FD003	FD004
IMDSSN[3]	206.11	1775.15	229.54	2852.81
KGHM[4]	250.99	1131.03	333.44	3356.10
SIGRNNDWI[6]	229	1129	314	1568
Dynamic Self-Learning Neural Network[7]	204	1004	271	1558
SACN[11]	291.59	3184.75	418.89	8281.45
TATFA-Transformer[8]	261.50	1359.70	210.21	2506.35
SA-MSTCN-GFA[9]	165	1096	178	1115
DA-LSTM[10]	263	842	360	1372
DAI-DEQ	74.6	219.98	120.67	268.49
Adaptive DAI-DEQ	84.28	259.44	99.6	312.03

equilibrium structure and steady-state iteration mechanism can more accurately capture the temporal degradation features of engine sensor data and effectively improve the accuracy of single-point RUL prediction.

The official Score metric further verifies that the two proposed models obtain the lowest comprehensive deviation penalty scores. The Score of DAI-DEQ is only 74.6 on FD001 and 120.67 on FD003, which are substantially lower than those of the best baseline SA-MSTCN-GFA (165, 178) and DA-LSTM (263, 360). This indicates that the proposed methods can effectively suppress the accumulated errors caused by over-prediction and under-prediction, which is more consistent with practical engineering evaluation criteria. The adaptive DAI-DEQ also presents obvious advantages in Score, especially with a value of 99.6 on FD003, reflecting that the adaptive degradation alignment mechanism improves prediction stability.

In summary, compared with traditional recurrent networks, attention-based models, and graph temporal models, the proposed models achieve the lowest RMSE and Score simultaneously on single datasets. It proves that our models possess stronger representation capability and higher prediction reliability in nonlinear degradation fitting, temporal feature extraction, and RUL prediction, which establishes a solid performance foundation for subsequent cross-domain transfer prediction.

To validate the generalization ability and adaptability of the proposed models in cross-domain transfer prediction scenarios, we select eight representative models for

cross-domain RUL prediction as baselines, including CURA, DMFR, WD-MSADA, LSTM-AdaBoost, TLDFA, DCTLIP, and DANN-OPKC. These baselines cover mainstream approaches such as traditional transfer learning and time series prediction combined with transfer learning. Taking RMSE (prediction error) and the official CMAPSS Score (comprehensive deviation penalty) as core evaluation metrics, we conduct quantitative comparison experiments on 12 groups of cross-domain tasks ranging from FD001→FD002 to FD004→FD003. The results are listed in Table 10 and Table 11, where the symbol "-" means that the corresponding experiment is not reported in the original literature.

Table 10: Comparison of RMSE for RUL prediction of different models in cross-domain scenarios

Task	CURA[23]	DMFR[24]	WD-MSADA[12]	LSTM-AdaBoost[17]	TLDFA[18]	DCTLIP[14]	DANN-OPKC[15]	DAI-DEQ	Adaptive DAI-DEQ
FD001→FD002	32.24	31.4	18.60	18.44	-	-	14.43	49.85	12.90
FD001→FD003	39	25.4	19.13	19.16	35.99	28.05	16.83	18.53	14.67
FD001→FD004	37.35	39.3	21.93	22.30	42.66	-	17.99	52.26	14.26
FD002→FD001	43.29	46.6	15.09	15.91	-	-	11.79	27.28	24.79
FD002→FD003	38.65	43.9	19.13	19.41	34.58	-	15.71	47.84	36.97
FD002→FD004	42.02	23.1	23.38	21.42	37.26	31.54	17.09	21.91	16.76
FD003→FD001	26.67	36.3	17.64	17.95	20.30	29.71	17.27	9.85	30.40
FD003→FD002	22.56	39.1	16.38	19.03	37.46	-	16.49	42.68	18.63
FD003→FD004	30.52	39.0	19.49	19.61	-	-	16.01	45.51	10.90
FD004→FD001	33.87	46.1	19.15	24.51	27.13	-	15.94	14.32	29.36
FD004→FD002	35.27	34.0	20.30	20.91	19.93	31.95	16.79	7.66	31.21
FD004→FD003	35.35	47.9	14.86	20.51	-	-	12.67	15.27	28.86

Table 11: Comparison of Score for RUL prediction of different models in cross-domain scenarios

Task	CURA[23]	DMFR[24]	WD-MSADA[12]	LSTM-AdaBoost[17]	TLDFA[18]	DCTLIP[14]	DANN-OPKC[15]	DAI-DEQ	Adaptive DAI-DEQ
FD001→FD002	8581	3398	2043	1595.39	-	-	906	387456.91	745.34
FD001→FD003	3955	3485	1411	1667.71	5538.41	8645.80	790	658.77	402.64
FD001→FD004	13664	6490	3260	5038.59	16577.27	-	1518	291977.33	883.25
FD002→FD001	6551	58782	485	548.93	-	-	230	32680.75	25141.58
FD002→FD003	4568	51831	1394	1453.46	19925.92	-	398	138540.22	124386.88
FD002→FD004	20439	3025	3279	4959.30	12638.25	11645.93	1797	3269.72	2060.75
FD003→FD001	1782	10598	1191	756.29	1553.19	13442.76	373	149.52	9610.60
FD003→FD002	7350	9712	2156	2239.72	13987.47	-	1136	7721.05	7347.86
FD003→FD004	6360	8639	2594	2533.95	-	-	1526	7838.46	551.36
FD004→FD001	2826	25909	1427	2730.16	4007.69	-	387	495.82	20355.86
FD004→FD002	8660	19578	3963	2066.05	1290.95	21088.65	1265	218.97	100023.40
FD004→FD003	4123	2851	413	2045.40	-	-	131	490.67	54101.09

It can be observed from Table 10 and Table 11 that the overall performance of baseline models varies significantly. DANN-OPKC achieves the most stable performance across all scenarios, with the lowest RMSE and Score in most tasks, showing excellent cross-domain adaptation capability. WD-MSADA and LSTM-AdaBoost also perform steadily with minor fluctuations in prediction errors and comprehensive metrics, which are reliable traditional transfer learning models. In contrast, CURA, DMFR, TLDFA, and DCTLIP suffer from large performance fluctuations in different cross-domain tasks. Their prediction errors rise sharply under certain operating conditions, leading to inferior overall performance compared with the above three models.

The cross-domain prediction performance of the proposed DAI-DEQ and adaptive DAI-DEQ presents distinct characteristics. The DAI-DEQ only achieves optimal results in two specific tasks: FD003→FD001 and FD004→FD002, with RMSE values of 9.85 and 7.66, respectively. In most other cross-domain scenarios, its prediction errors increase dramatically, and the Score even reaches hundreds of thousands, which indicates that DAI-DEQ lacks universal cross-domain generalization ability and only works well in individual tasks with high domain matching degree. By contrast, the adaptive DAI-DEQ achieves performance improvement via the adaptive mechanism. It obtains the best RMSE and Score among all models in low-complexity cross-domain tasks with FD001 and FD003 as source domains. Nevertheless, its performance degrades obviously in noisy and variable operating condition scenarios where FD002 and FD004 serve as source domains, and its prediction accuracy and comprehensive performance are worse than mature baselines such as DANN-OPKC.

Overall, the adaptive mechanism effectively solves the problem of widespread failure of the original DAI-DEQ in cross-domain prediction. The model is upgraded from "applicable only in individual scenarios" to "optimal in partial scenarios", but it cannot maintain leading performance under all operating conditions. The overall performance ranking is: DANN-OPKC $>$ WD-MSADA $\approx$ LSTM-AdaBoost $>$ Adaptive DAI-DEQ $>$ CURA/DMFR/TLDFA/DCTLIP $>$ DAI-DEQ. The advantages of the proposed models are mainly reflected in cross-domain transfer tasks with low-complexity source domains. There is still room for improvement in robustness and generalization under complex operating conditions, and a certain gap remains compared with state-of-the-art domain adaptation methods.

5 Conclusion

To address the core issues in cross-domain RUL prediction of aero-engines, including domain distribution shift, insufficient representation of degradation features, unstable training of deep models, and lack of uncertainty quantification, this paper proposes an RUL prediction method integrating the DEQ and adversarial domain adaptation, constructing a complete framework covering data preprocessing, degradation feature guidance, deep feature extraction, domain distribution alignment, and uncertainty quantification. The method innovatively embeds the DAI into the dual-input attention module to quantify the synchronization between sensor degradation trends and RUL attenuation, guiding the model to focus on highly correlated degradation features. The DEQ network based on Anderson acceleration iteration is adopted to replace the traditional deep stacking structure, fundamentally alleviating the gradient vanishing problem and achieving the decoupling of memory consumption from network depth. Combined with the GRL and domain discriminator, an adversarial domain adaptation framework is constructed to effectively mitigate the distribution shift between the source and target domains. Finally, a Bayesian neural network and Monte Carlo sampling are introduced to realize RUL prediction and uncertainty quantification, improving the engineering reliability of prediction results.

Extensive experimental validations based on the NASA C-MAPSS dataset demonstrate that the proposed method exhibits excellent prediction performance in both

single-dataset and cross-domain scenarios. In single-dataset experiments, the RMSE of the DAI-DEQ model is maintained between 7.35 and 8.42, significantly outperforming existing mainstream deep learning models. Ablation experiments confirm that the DAI module can effectively reduce prediction bias and enhance the model's ability to mine critical degradation features. Cross-domain scenario experiments show that the adaptive DAI-DEQ model performs outstandingly in scenarios with simple source domains and complex target domains, such as a 74.1% reduction in RMSE in the FD001→FD002 scenario. However, it exhibits adverse effects in scenarios with complex source domains and simple target domains, mainly because noise and irrelevant operating condition information in the source domain are incorrectly transferred, interfering with the learning of core degradation features in the target domain. Comparative experiments further verify that the proposed method comprehensively outperforms existing baseline models in single-dataset prediction accuracy and achieves state-of-the-art performance in cross-domain scenarios with low-complexity source domains, demonstrating good practicality and advancement.

Although this study has effectively addressed some key issues in cross-domain aero-engine RUL prediction, there are still certain limitations: the robustness of the adaptive module in complex source domain scenarios needs to be improved, and stable optimal cross-domain prediction under all operating conditions has not yet been achieved; the DAI index only characterizes degradation synchronization based on linear fitting, with limited adaptability to nonlinear degradation trends. Future research can be optimized from three aspects: First, improve the adversarial domain adaptation framework by introducing a dynamic weight adjustment mechanism to suppress the transfer of impurity information from the source domain and enhance generalization ability in complex scenarios; Second, optimize the calculation method of the DAI index by combining nonlinear fitting methods to enhance its characterization ability for complex degradation trends; Third, simplify the model structure and optimize iteration efficiency to reduce the engineering deployment cost of the model, promoting the practical application of this method in aero-engine condition-based maintenance.

Declarations

- **Funding**:
  No funding was received for this work.
- **Competing interests**:
  The authors declare that they have no competing interests.
- **Ethics approval and consent to participate**:
  Not applicable.
- **Consent for publication**:
  Not applicable.
- **Data availability**:
  The C-MAPSS dataset that supports the findings of this study is publicly available from the NASA Prognostics Center of Excellence (PCoE) Data Set Repository at `https://www.nasa.gov/intelligent-systems-division/discovery-and-systems-health/pcoe/pcoe-data`
- **Materials availability**:

Not applicable.

- **Code availability**:
  The code generated during the current study is not publicly available but may be obtained from the corresponding author upon reasonable request.
- **Author contribution**:
  Not applicable.

References

[1] Xu, Q., Chen, Z., Wu, K., Wang, C., Wu, M., Li, X.: Kdnet-rul: A knowledge distillation framework to compress deep neural networks for machine remaining useful life prediction. IEEE Transactions on Industrial Electronics **69**(2), 2022–2032 (2022) https://doi.org/10.1109/TIE.2021.3057030

[2] Ahmad, R., Kamaruddin, S.: An overview of time-based and condition-based maintenance in industrial application. Computers & industrial engineering **63**(1), 135–149 (2012)

[3] Zhang, J., Li, X., Tian, J., Luo, H., Yin, S.: An integrated multi-head dual sparse self-attention network for remaining useful life prediction. Reliability Engineering & System Safety **233**, 109096 (2023)

[4] Li, Y., Chen, Y., Hu, Z., Zhang, H.: Remaining useful life prediction of aero-engine enabled by fusing knowledge and deep learning models. Reliability Engineering & System Safety **229**, 108869 (2023)

[5] Elloumi, S., Bahroun, S., Yahia, S.B., Kaddes, M.: Enhanced face recognition in crowded environments with 2d/3d features and parallel hybrid cnn-rnn architecture with stacked auto-encoder. Big Data and Cognitive Computing **9**(8) (2025) https://doi.org/10.3390/bdcc9080191

[6] Xiang, S., Li, P., Huang, Y., Luo, J., Qin, Y.: Single gated rnn with differential weighted information storage mechanism and its application to machine rul prediction. Reliability Engineering & System Safety **242**, 109741 (2024)

[7] Xiang, S., Zheng, X., Miao, J., Qin, Y., Li, P., Hou, J., Ilolov, M.: Dynamic self-learning neural network and its application for rotating equipment rul prediction. IEEE Internet of Things Journal **12**(9), 12257–12266 (2024)

[8] Zhang, Y., Su, C., Wu, J., Liu, H., Xie, M.: Trend-augmented and temporal-featured transformer network with multi-sensor signals for remaining useful life prediction. Reliability Engineering & System Safety **241**, 109662 (2024)

[9] Xu, Z., Zhang, Y., Miao, Q.: An attention-based multi-scale temporal convolutional network for remaining useful life prediction. Reliability Engineering & System Safety **250**, 110288 (2024)

[10] Shi, J., Zhong, J., Zhang, Y., Xiao, B., Xiao, L., Zheng, Y.: A dual attention lstm lightweight model based on exponential smoothing for remaining useful life prediction. Reliability Engineering & System Safety **243**, 109821 (2024)

[11] Li, D., Chen, J., Huang, R., Chen, Z., Li, W.: Sensor-aware capsnet: Towards trustworthy multisensory fusion for remaining useful life prediction. Journal of Manufacturing Systems **72**, 26–37 (2024)

[12] Shi, H., Huang, C., Zhang, X.: Wasserstein distance based multi-scale adversarial domain adaptation method for remaining useful life prediction. Applied Intelligence **53**, 3622–3637 (2023) https://doi.org/10.1007/s10489-022-03670-6

[13] Yuan, M., Cai, T.T.: A reproducing kernel hilbert space approach to functional linear regression (2010)

[14] Shu, Q., Zhang, F., Shen, L., Ng, H.K.T.: Rul prediction with cross-domain adaptation based on reproducing kernel hilbert space. IEEE Transactions on Reliability **74**(3), 3871–3883 (2025) https://doi.org/10.1109/TR.2024.3488792

[15] Zhu, Y., Zhang, F., Chen, W., Cheng, Z., Shen, L.: Order-preserving kernel contrastive learning with applications to cross-domain rul prediction. IEEE Transactions on Reliability **75**, 97–111 (2026) https://doi.org/10.1109/TR.2025.3635402

[16] Huang, C.-G., Huang, H.-Z., Li, Y.-F.: A bidirectional lstm prognostics method under multiple operational conditions. IEEE Transactions on Industrial Electronics **66**(11), 8792–8802 (2019) https://doi.org/10.1109/TIE.2019.2891463

[17] Seo, S., Hwang, J., Chung, M.: Supervised domain adaptation for remaining useful life prediction based on adaboost with long short-term memory. IEEE Access **12**, 96757–96768 (2024) https://doi.org/10.1109/ACCESS.2024.3426909

[18] Lyu, Y., Wen, Z., Chen, A.: A novel transfer learning approach based on deep degradation feature adaptive alignment for remaining useful life prediction with multi-condition data. Journal of Intelligent Manufacturing **36**, 619–637 (2025) https://doi.org/10.1007/s10845-023-02264-4

[19] Lyu, Y., Wen, Z., Chen, A.: A novel transfer learning approach based on deep degradation feature adaptive alignment for remaining useful life prediction with multi-condition data. Journal of Intelligent Manufacturing **36**(1), 619–637 (2025)

[20] Liu, L., Yu, Y., Guan, X., Jing, W., Wu, X.: A novel interpretable deep autoencoder for health indicators construction and similarity-based rul prediction. Measurement, 120571 (2026)

[21] Chen, W., Wu, W., Li, X., Li, H., Zhang, M.: A probabilistic inference framework for rul prediction: Modeling, propagation, and quantification of uncertainty.

Measurement **270**, 120851 (2026)

[22] Li, K., Gao, C., Fu, C., Sun, G., Huang, J., Du, N.: Rul prediction for rf mems switches based on lstm network considering frequency effects. Measurement, 121440 (2026)

[23] Hou, Y., Wang, Y., Wu, M., Kwoh, C.-K., Li, X., Chen, Z.: Source-free Calibrated Uncertainty for RUL Adaptation with incomplete degradation. Mechanical Systems and Signal Processing **233**, 112718 (2025) https://doi.org/10.1016/j.ymssp.2025.112718

[24] Li, J., Yang, Z.: Remaining useful life prediction of turbine engines using multimodal transfer learning. Machines **13**(9) (2025) https://doi.org/10.3390/machines13090789

[25] Bai, S., Koltun, V., Kolter, J.Z.: Multiscale deep equilibrium models. Advances in neural information processing systems **33**, 5238–5250 (2020)

[26] Vollert, S., Theissler, A.: Challenges of machine learning-based rul prognosis: A review on nasa's c-mapss data set. In: 2021 26th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), pp. 1–8 (2021). IEEE

[27] Osumi, K., Yamashita, T., Fujiyoshi, H.: Domain adaptation using a gradient reversal layer with instance weighting. In: 2019 16th International Conference on Machine Vision Applications (MVA), pp. 1–5 (2019). IEEE

[28] Arbel, J., Pitas, K., Vladimirova, M., Fortuin, V.: A primer on bayesian neural networks: review and debates. Statistical Science **41**(2), 316–353 (2026)

[29] Likas, A., Vlassis, N., Verbeek, J.J.: The global k-means clustering algorithm. Pattern recognition **36**(2), 451–461 (2003)

[30] Jain, A., Nandakumar, K., Ross, A.: Score normalization in multimodal biometric systems. Pattern recognition **38**(12), 2270–2285 (2005)

[31] Rigatti, S.J.: Random forest. Journal of insurance medicine **47**(1), 31–39 (2017)

[32] Lawrance, A., Lewis, P.: The exponential autoregressive-moving average earma (p, q) process. Journal of the Royal Statistical Society: Series B (Methodological) **42**(2), 150–161 (1980)

[33] Fang, T., Lahdelma, R.: Optimization of combined heat and power production with heat storage based on sliding time window method. Applied energy **162**, 723–732 (2016)