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.

Continuous Degradation Representation Learning for Remaining Useful Life Prediction

Wei Li
North University of China
Ruitao Ning
North University of China
Shilin He
North University of China
Bo Li
North University of China
Zhidong Zhang
zdzhang@nuc.edu.cn

North University of China

Article

Keywords:

Posted Date: July 14th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-10212340/v1

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

Additional Declarations: No competing interests reported.

Version of Record: A version of this preprint was published at Scientific Reports on October 13th, 2026.
See the published version at https://doi.org/10.1038/s41598-026-75502-3.

Continuous Degradation Representation Learning for Remaining Useful Life Prediction

Wei Li[1], **Ruitao Ning**[1], **Shilin He**[1], **Bo Li**[1], **and Zhidong Zhang**[1,*]

[1]Key Laboratory of Extreme Environment Optoelectronic Dynamic Testing Technology and Instrument, North University of China, Taiyuan, 030051, Shanxi, China
[*]zdzhang@nuc.edu.cn

ABSTRACT

Predicting the Remaining Useful Life (RUL) of aero-engines is crucial for aviation safety and maintenance scheduling. Existing deep learning approaches, however, often learn fragmented or discontinuous degradation representations from multivariate sensor data, limiting their ability to accurately model degradation trajectories and estimate RUL. To address this issue, we propose CoDeR, a continuous degradation representation learning framework for aero-engine RUL prediction. CoDeR is designed to learn temporally coherent degradation representations that better capture the progressive degradation process. Specifically, a Dual-axis Coupled Transformer Encoder (DCTE) is developed to extract degradation features by jointly modeling temporal dynamics and sensor-wise dependencies. In addition, a Time-Aware Contrastive Learning (TACL) module is introduced to incorporate temporal priors into contrastive learning, enabling the feature space to preserve degradation continuity by explicitly regulating relationships among samples at different degradation stages. By jointly optimizing degradation representation learning and RUL regression,CoDeR learns more discriminative and continuous degradation representations, leading to improved prediction accuracy.Experimental results on representative aero-engine benchmark datasets demonstrate the effectiveness and superiority of the proposed framework.

1 Introduction

Operational safety and efficiency are critical in modern industrial systems, especially in high-risk sectors where unexpected failures may extend maintenance cycles, increase operating costs, or lead to serious accidents[1]. Predictive Maintenance (PdM) has therefore become an important strategy for balancing reliability and economic efficiency[2]. By using real-time sensor data, PdM systems can monitor equipment conditions and detect early signs of degradation before faults become severe[3].

Remaining Useful Life (RUL) prediction is a core task in predictive maintenance (PdM). It estimates the remaining safe operating window of a system from historical and real-time time-series data, thereby supporting timely maintenance decisions[4]. With the rapid growth of sensor data and computing power, deep learning (DL) has become the dominant paradigm for RUL prediction. Early studies often relied on traditional machine learning methods, such as support vector machines (SVMs) and shallow neural networks, which typically combined handcrafted features with regression models and were mainly applied in data-limited engineering scenarios. In contrast, deep models can learn features directly from raw sensor signals. Convolutional neural networks (CNNs) improve prediction performance through local pattern extraction and sparse connectivity[5,6], whereas recurrent neural networks (RNNs) and their variants, such as long short-term memory (LSTM) networks and gated recurrent units (GRUs), are better suited for modeling temporal dynamics in degradation sequences[7–9]. Attention mechanisms further enhance performance by emphasizing informative time steps[10,11], and dual-weighting strategies across temporal and feature dimensions have also been introduced to improve feature discrimination[12]. Although these methods strengthen feature extraction, they are still limited in capturing the coupled temporal and structural evolution of degradation signals.

To better capture the complex dependencies in multivariate degradation signals, recent studies have explored parallel modeling strategies to learn complementary information from different dimensions. As a representative work in multivariate time-series analysis, Crossformer[13]explicitly models both temporal dependencies and cross-variable correlations through a two-stage attention mechanism, demonstrating the effectiveness of dual-axis representation learning. Building upon this idea, several studies have extended parallel spatiotemporal learning to RUL prediction. PSTFormer[14] employs parallel temporal and spatial attention modules to simultaneously capture degradation dynamics and sensor-wise dependencies, followed by task-guided feature fusion. Similarly, MSMT-FDA[15]jointly models spatial correlations and multiscale temporal degradation patterns while incorporating frequency-domain attention to enhance degradation representation. In addition, DAST[16] adopts dual-attention branches to separately extract temporal and sensor-level features before feature fusion for RUL estimation. These studies demonstrate that parallel modeling across temporal and structural dimensions can improve the representation capability of multivariate time-series data and yield more informative degradation features for RUL prediction. Nevertheless,

most existing methods still rely on separate temporal-spatial feature extraction followed by feature fusion, which may limit their ability to capture fine-grained interactions between degradation evolution and sensor correlations under complex operating conditions.Therefore, developing a unified modeling framework that effectively couples these two dimensions is crucial for learning more informative degradation features and accurately characterizing degradation trajectories.

Meanwhile, accurate RUL prediction depends not only on extracting informative degradation features but also on how these features are organized in the representation space. Even with powerful backbone networks, samples from different degradation stages may still exhibit weak ordering or overlapping distributions, resulting in representations that are insufficiently aligned with the continuous degradation process. Therefore, learning degradation-aware and well-structured representations has become an important challenge beyond backbone design.

To address this issue, recent studies have increasingly explored contrastive learning as a representation regularization strategy. By explicitly constraining relationships among samples in the feature space, contrastive learning can promote more structured and discriminative degradation representations. Representative methods such as SimCLR[17], MoCo[18], and CP[19] learn discriminative feature spaces through positive-negative pair construction. In supervised settings, contrastive objectives have also been integrated with regression tasks to improve feature structure and generalization[20,21]. In degradation analysis, contrastive learning has been employed to enhance feature separability and robustness. For example, Liu et al.[22] combined time-frequency enhancement with reconstruction loss for multiscale representation learning, while Abdelfattah et al.[23] improved inter-class separability through multi-layer contrastive regularization.Since RUL prediction is inherently an ordered regression problem, several studies have further incorporated degradation-order information into contrastive objectives. Zha et al.[24] transformed label ordering into feature-space constraints, Zhu et al.[25] introduced parity decomposition for robust order comparison, and Li et al.[26] improved cross-condition generalization through a label-distance-based multi-triplet strategy. These studies demonstrate the potential of contrastive learning for structured degradation representation learning. However, most existing methods still rely on discrete positive-negative pair construction, which imposes uniform separation among samples and may fail to capture the continuous and progressive nature of degradation trajectories.

Motivated by this limitation, we propose CoDeR, a continuous degradation representation learning framework for RUL prediction. CoDeR aims to learn structured degradation representations that make the degradation process easier to model and the downstream RUL regression task easier to solve. The framework contains two key modules. First, a Dual-axis Coupled Transformer Encoder (DCTE) is used to extract informative sequence features from multivariate sensor data. Second, a Time-Aware Contrastive Learning (TACL) module is introduced to optimize the representation space by incorporating temporal relationships among degradation samples. In this way, CoDeR learns structured degradation representations that are more suitable for RUL regression. Experiments on the C-MAPSS[27] and N-CMAPSS[28] benchmarks show that the proposed method achieves competitive prediction accuracy and yields more ordered degradation representations.

The main contributions of this work are summarized as follows:

- We propose CoDeR, a continuous degradation representation learning framework for aero-engine RUL prediction, aiming to improve the continuity of degradation representations learned from multivariate sensor data.
- We design a DCTE to model dependencies across temporal and feature dimensions in multivariate sensor sequences, enabling more informative degradation feature extraction.
- We introduce a TACL module that incorporates temporal progression priors into contrastive representation learning, encouraging degradation representations with improved temporal continuity in the feature space.
- Experiments on the C-MAPSS and N-CMAPSS benchmark datasets demonstrate that the proposed framework achieves competitive RUL prediction performance while improving the continuity of learned degradation representations.

2 Methodology

The proposed framework, illustrated in Fig. 1 using aircraft engines as a representative case, comprises three stages: data preparation, joint learning, and RUL prediction. During data preparation, multi-channel sensor sequences are segmented via a sliding window to generate training and test sets. The joint learning stage simultaneously performs degradation representation learning and RUL regression. Finally, in the RUL prediction stage, the trained feature extractor and regression model process the test data to estimate the remaining life. We define the training dataset as $D = \{(x_i, y_i)\}_{i=1}^{N}$, where $x_i \in \mathbb{R}^{w \times m}$ represents the i-th multi-channel window observation, y_i is the corresponding RUL label, and N denotes the total number of samples. Here, m and w signify the feature dimension and window length, respectively. The DCTE extracts a d-dimensional feature vector $\mathbf{z}_i$ from x_i, which the regression model then maps to y_i. Ultimately, this framework aims to optimize the feature extractor and regression model for superior generalization on unseen data. Table 1 summarizes the principal notations used in this paper.

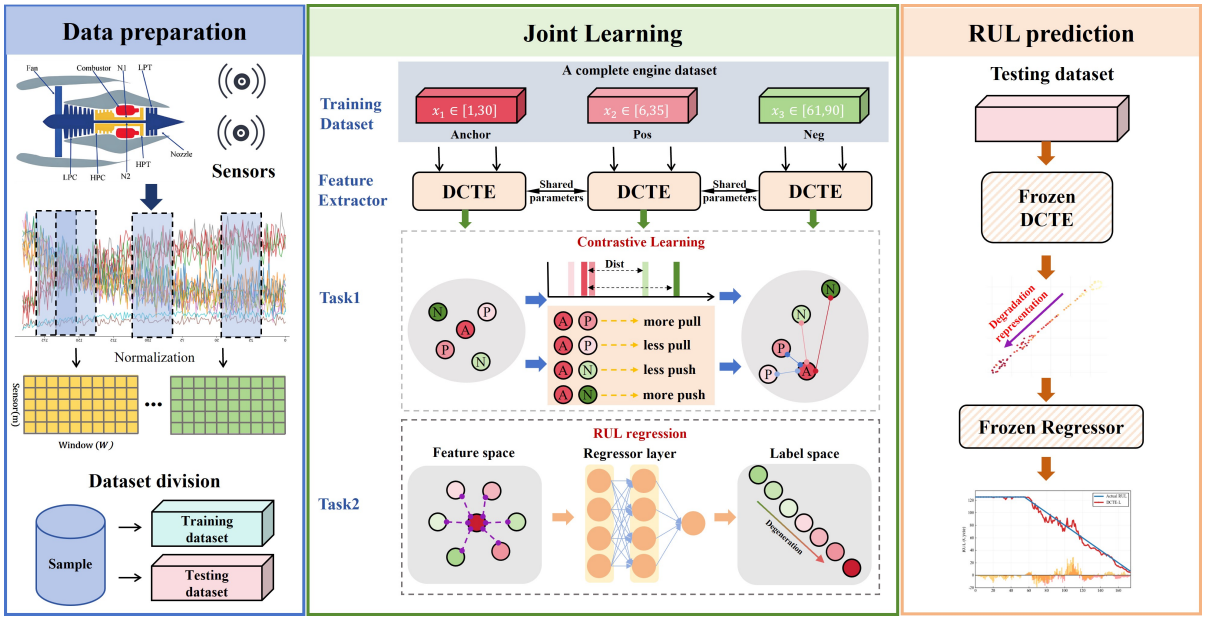


Figure 1. Overall workflow of the proposed method.

Table 1. Notations used in this paper.

Notation	Description
$\mathscr{E}:\mathbb{R}^m \to \mathbb{R}^H$	Linear embedding
$P_t \in \mathbb{R}^{w\times H}$	Temporal position encoding
$P_s \in \mathbb{R}^{H\times w}$	Sensor-axis position encoding
$\mathrm{Enc}_t(\cdot)$	Temporal Transformer encoder
$\mathrm{Enc}_s(\cdot)$	Spatial Transformer encoder
$\mathscr{P}_{t\to s}(\cdot)$	Temporal-to-spatial linear mapping
$\mathscr{P}_{s\to t}(\cdot)$	Spatial-to-temporal linear mapping
$\mathrm{CrossAttn}(\cdot)$	Cross-axis attention operation
$\mathrm{Flatten}(\cdot)$	Flatten operation
$\mathscr{L}(\cdot)$	Regression layer

2.1 Network Architecture

As shown in Fig. 2, the designed DCTE model maps the observed sequence of a failure instance $x \in \mathbb{R}^{w\times m}$ to a scalar output $\hat{y}$, while simultaneously capturing temporal and channel information. The model first projects the original input into a temporal view through a linear embedding operator $\mathscr{E}(\cdot)$:

$$X = \mathscr{E}(x) \in \mathbb{R}^{w\times H} \tag{1}$$

where w is the time window length and H is the hidden dimension.

To prevent self-attention operations from blurring sequential information, the model explicitly introduces positional embeddings along both axes. Specifically, on the temporal axis, a position encoding $P_t \in \mathbb{R}^{w\times H}$ similar to that in Transformer is adopted to obtain the initial temporal representation:

$$T^{(0)} = X + P_t \tag{2}$$

On the spatial axis, a sequential position embedding $P_s \in \mathbb{R}^{H\times w}$ is used to encode the channel sequence structure by first transposing the input to obtain a spatial representation:

$$S^{(0)} = X^\top + P_s \tag{3}$$

The above parallel positional injection mechanism ensures that the sequential order of the time dimension and sensor dimension is preserved and participates in representation learning during subsequent attention computations. The temporal and spatial

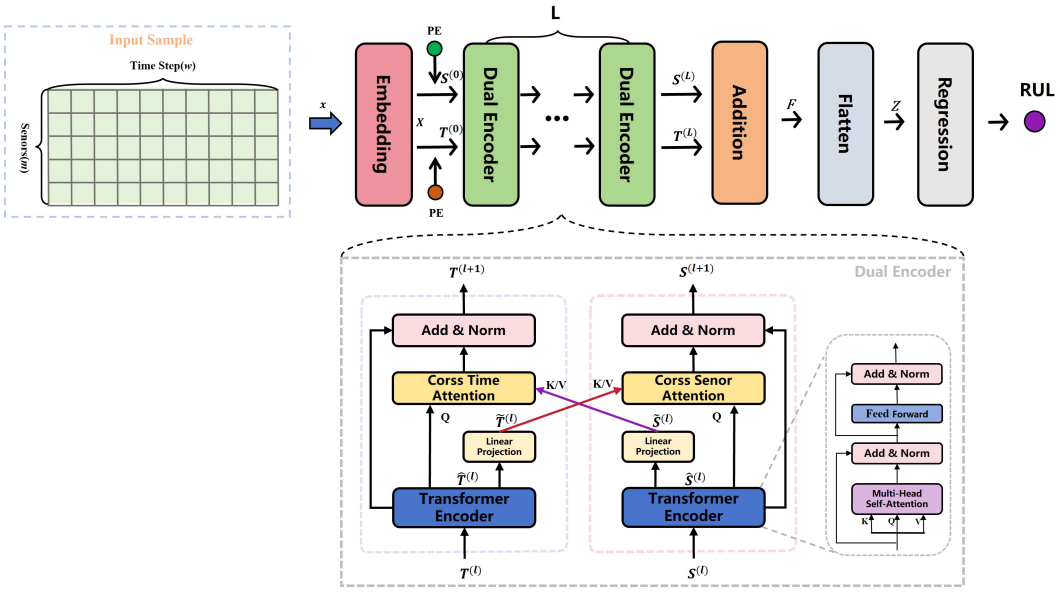


Figure 2. The architectural design of the DCTE.

representations are subsequently modeled within their respective dimensions using a pair of symmetric Transformer Encoders, enabling intra-axis modeling and thus achieving semantic interpretation: the temporal axis emphasizes the evolutionary dynamics of system degradation; the spatial axis reveals the redundant and coupled structures among channels. Specifically, let $T^{(l)}$ and $S^{(l)}$ denote the temporal and sensor representations at layer l, respectively. The intra-axis encoders generate:

$$\hat{T}^{(l)} = \mathrm{Enc}_t(T^{(l)}) \tag{4}$$

$$\hat{S}^{(l)} = \mathrm{Enc}_s(S^{(l)}) \tag{5}$$

where Enc_t and Enc_s are standard Transformer Encoder units[29].

Relying solely on intra-axis modeling cannot fully capture the complementarity between temporal and spatial information. To address this, bidirectional cross-axis attention is introduced after intra-axis encoding at each layer. This process first applies learnable linear mappings to project one representation into the embedding space of the other axis.

Projecting the spatial representation into the temporal domain and updating the temporal representation:

$$\tilde{S}^{(l)} = \mathscr{P}_{s\to t}(\hat{S}^{(l)}) \tag{6}$$

$$T^{(l+1)} = \mathrm{Norm}\left(\hat{T}^{(l)} + \mathrm{CrossAttn}(\hat{T}^{(l)}, \tilde{S}^{(l)})\right) \tag{7}$$

Projecting the temporal representation into the spatial domain and updating the sensor representation:

$$\tilde{T}^{(l)} = \mathscr{P}_{t\to s}(\hat{T}^{(l)}) \tag{8}$$

$$S^{(l+1)} = \mathrm{Norm}\left(\hat{S}^{(l)} + \mathrm{CrossAttn}(\hat{S}^{(l)}, \tilde{T}^{(l)})\right) \tag{9}$$

where $P_{s\to t} \in \mathbb{R}^{w\times H}$ and $P_{t\to s} \in \mathbb{R}^{H\times w}$ are linear mappings. Through bidirectional interaction, the model dynamically exchanges information at each layer, forming a joint representation.

After L layers, the final joint output is obtained:

$$T^{(L)}, S^{(L)} = \mathrm{DualEnc}_L(T^{(0)}, S^{(0)}) \tag{10}$$

To construct the final joint representation for RUL estimation, the spatial-domain output is fused with the temporal-domain output:

$$F = T^{(L)} + (S^{(L)})^{\top} \tag{11}$$

where $F \in \mathbb{R}^{w\times H}$ is the fused representation. Subsequently, F is flattened into a one-dimensional feature vector:

$$\mathbf{z} = \mathrm{Flatten}(F) \in \mathbb{R}^{w\cdot H} \tag{12}$$

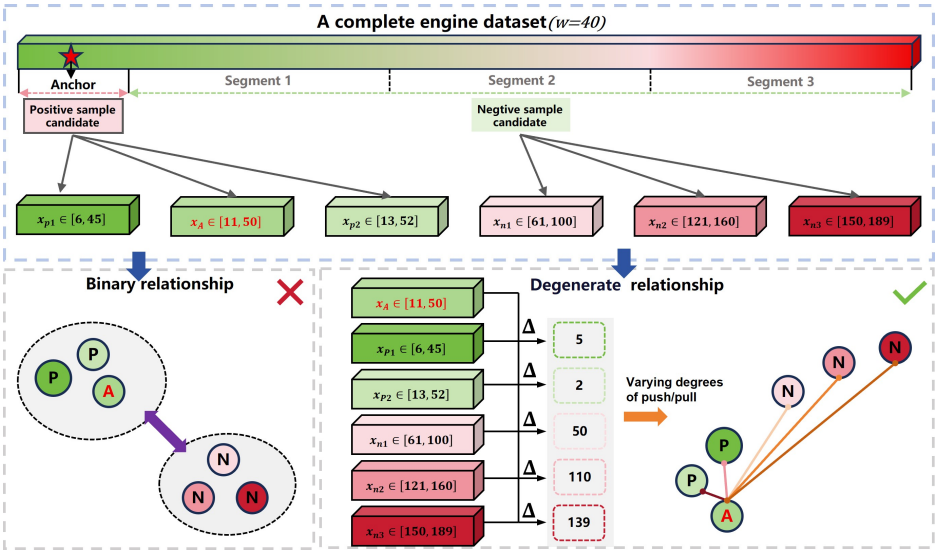


Figure 3. Illustration of sampling and contrastive learning for modeling continuous degradation trajectories.

For the downstream RUL estimation task, the model applies a linear regression:

$$\hat{y} = \mathscr{L}(\mathbf{z}) \tag{13}$$

2.2 Time-Aware Contrastive Learning

To mitigate the fragmented degradation representations induced by discrete modeling, we propose a TACL module that captures the temporal continuity of degradation evolution. At the sampling stage, TACL jointly models local temporal dependencies and the global degradation distribution through neighborhood-based positive sampling and interval-aware negative sampling. At the optimization stage, temporal distances are incorporated into the contrastive objective to adaptively modulate attraction and repulsion between samples, thereby encouraging more consistent degradation representations learned by contrastive learning.

2.2.1 Sampling Strategy

As shown in Fig. 3, the temporal index of samples is used as reference information to construct positive and negative sample sets. Given a degradation sequence $D = [x_1, x_2, \ldots, x_n]$ and its index set $I = \{1, \ldots, n\}$, continuous samples are generated using a sliding window of length w. For any anchor index $a \in I$, let the half-window width be $r = \lfloor w/2 \rfloor$, and define its local neighborhood as:

$$B_a = \{i \in I \mid |i - a| \leq r\}. \tag{14}$$

Positive sample indices are drawn from within the anchor's neighborhood. Let $P_a = B_a \setminus \{a\}$, and uniformly sample P indices without replacement from P_a, denoted as $\{p_1, \ldots, p_P\}$. The corresponding positive sample set is $X_P = \{x_{p_i}\}_{i=1}^{P}$.

Negative sample indices are drawn from the remaining sequence $N_a = I \setminus B_a$. We divide N_a into K consecutive sub-intervals:

$$N_a = S_1 \cup S_2 \cup \cdots \cup S_K, \quad S_1 < S_2 < \cdots < S_K. \tag{15}$$

One index is randomly sampled from each non-empty sub-interval S_k, resulting in the negative sample set $X_N = \{x_{n_k}\}_{k=1}^{K}$.

2.2.2 Loss Function

To incorporate temporal priors, we employ a time-distance-weighted contrastive objective. Let $\mathbf{z} = f(x)$ be the embedding extracted by the DCTE. We compute the similarity between an anchor $\mathbf{z}$ and a sample $\mathbf{z}_k$ (positive or negative) using the cosine similarity on the L_2-normalized representation space:

$$s(\mathbf{z}, \mathbf{z}_k) = \frac{\mathbf{z}^\top \mathbf{z}_k}{|\mathbf{z}|_2 |\mathbf{z}_k|_2} \tag{16}$$

Specifically, the similarity for the i-th positive sample and j-th negative sample are denoted as s_i^+ and s_j^-, respectively.

To capture relative positional relationships, we define positive weights α_i and negative weights β_j:

$$\alpha_i = 1 - \frac{|a - p_i|}{n}, \quad \beta_j = \frac{|a - n_j|}{n}, \tag{17}$$

where n is the sequence length. The loss is defined as:

$$\mathscr{L}_{\text{TACL}} = -\log \frac{\sum_i \exp(\alpha_i s_i^+ / T)}{\sum_i \exp(\alpha_i s_i^+ / T) + \sum_j \exp(\beta_j s_j^- / T)}, \tag{18}$$

where T is the temperature parameter. The gradient with respect to $\mathbf{z}$ is:

$$\frac{\partial \mathscr{L}_{\text{TACL}}}{\partial \mathbf{z}} = \sum_i \frac{\alpha_i}{T}(q_i^+ - p_i^+)\mathbf{z}_i^+ + \sum_j \frac{\beta_j}{T} r_j^- \mathbf{z}_j^-, \tag{19}$$

where the intermediate variables are defined as:

$$p_i^+ = \frac{\exp(\alpha_i s_i^+ / T)}{\sum_k \exp(\alpha_k s_k^+ / T)}, \tag{20a}$$

$$q_i^+ = \frac{\exp(\alpha_i s_i^+ / T)}{\sum_k \exp(\alpha_k s_k^+ / T) + \sum_l \exp(\beta_l s_l^- / T)}, \tag{20b}$$

$$r_j^- = \frac{\exp(\beta_j s_j^- / T)}{\sum_k \exp(\alpha_k s_k^+ / T) + \sum_l \exp(\beta_l s_l^- / T)}. \tag{20c}$$

From this, it follows that $\partial \mathscr{L} / \partial s_i^+ < 0$ and $\partial \mathscr{L} / \partial s_j^- > 0$. The temporal-aware weights α_i and β_j function as adaptive scaling factors to modulate the intensity of attraction and repulsion.

Specifically, the gradient magnitude of the positive term increases with α_i, ensuring that temporally proximal samples aggregate toward the anchor at an accelerated rate. Conversely, the negative term gradient is strengthened by β_j, facilitating the repulsion of long-span samples. Since the attraction and repulsion forces vary continuously with temporal distance, neighboring degradation states are constrained to preserve local consistency in the representation space, while distant states remain distinguishable. The resulting embeddings therefore exhibit a smooth transition along the degradation process, which is consistent with the underlying continuous evolution of degradation.

2.3 Joint Training Process

This paper integrates RUL regression supervision with temporal-aware contrastive constraints through joint optimization, enabling the learned representations to better reflect the continuous degradation process while preserving prediction accuracy.

Let the model's RUL prediction for the i-th sample be $\hat{y}_i$ and its corresponding ground-truth label be y_i. The regression loss is formulated using the mean squared error (MSE):

$$\mathscr{L}_{\text{MSE}} = \frac{1}{|B|} \sum_{i \in B} (\hat{y}_i - y_i)^2, \tag{21}$$

where B denotes a mini-batch.

To explicitly model the relative proximity relationships among degradation states along the temporal dimension, the continuity constraint loss $\mathscr{L}_{\text{TACL}}^{(i)}$ is introduced, where $\mathscr{L}_{\text{TACL}}^{(i)}$ denotes the contrastive loss computed with the i-th sample as the anchor. This term maps the continuity of physical time into the learned feature space. The final multi-objective optimization function is expressed as:

$$\mathscr{L} = \mathscr{L}_{\text{MSE}} + \frac{\mu}{|B|} \sum_{i \in B} \mathscr{L}_{\text{TACL}}^{(i)}, \tag{22}$$

where $\mu \geq 0$ is a balancing coefficient that controls the trade-off between regression accuracy and temporal consistency of degradation representations.

3 Experiments

To evaluate the proposed method, experiments are conducted primarily on the C-MAPSS benchmark, which serves as the main testbed for RUL prediction. The N-CMAPSS dataset is further used as an external generalization benchmark to assess the robustness of the proposed approach under more realistic operating conditions. The experimental environment is based on Ubuntu, Python 3.11, and PyTorch 2.3.1. All computations are performed on an NVIDIA RTX 4090 GPU with 24 GB VRAM.

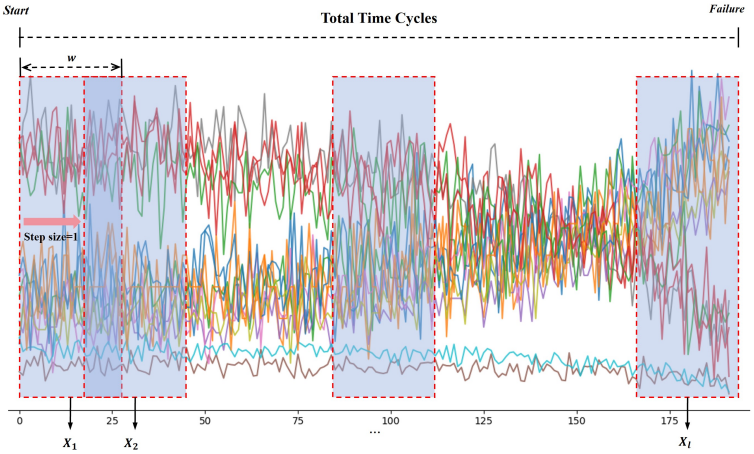


Figure 4. Illustration of splitting data with the time window method.

3.1 Evaluation Metrics

RUL prediction is treated as a regression problem. This work adopts the root mean squared error (RMSE) and Score as evaluation metrics. Let N denote the total number of samples, y_i be the ground-truth RUL of the i-th sample, and $\hat{y}_i$ be its predicted value.

The RMSE is defined as:

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}. \tag{23}$$

The Score metric introduces an asymmetric penalty mechanism, assigning higher weights to overestimation than underestimation. It is defined as:

$$\text{Score} = \sum_{i=1}^{N} \begin{cases} \exp\left(-\frac{y_i - \hat{y}_i}{13}\right) - 1, & \text{if } y_i - \hat{y}_i < 0 \\ \exp\left(\frac{y_i - \hat{y}_i}{10}\right) - 1, & \text{if } y_i - \hat{y}_i \geq 0. \end{cases} \tag{24}$$

3.2 Experiments on C-MAPSS

3.2.1 Data Description

The C-MAPSS dataset was generated using the NASA aerospace propulsion system simulation software. It comprises four sub-datasets (FD001–FD004) consisting of sensor monitoring data from multiple engines. Their basic information is summarized in Table 2.

Table 2. Description of the aircraft engine dataset C-MAPSS

Dataset	FD001	FD002	FD003	FD004
Training engines	100	260	100	249
Test engines	100	259	100	248
Op. conditions	1	6	1	6
Fault modes	1	1	2	2

Each time step includes readings from 21 sensors and operating condition parameters. The training set covers the complete lifecycle, while the test set is truncated at a random point before failure.

3.2.2 Data Preprocessing

Data from the C-MAPSS dataset were preprocessed prior to modeling. Specifically, for all sub-datasets FD001–FD004, sensors with invariant readings were excluded. Following established protocols[30], features indexed 1, 5, 6, 10, 16, 18, and 19 were removed from all datasets.

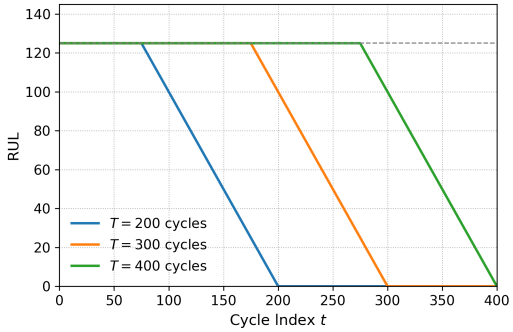


Figure 5. Piecewise RUL function for labeling.

A sliding window method is used to generate samples. For a sequence of length T, window length w, and m sensor channels, $(T-w+1)$ samples are obtained. The label is assigned as the RUL value at the end of each window (see Fig. 4). Input features are normalized using Min-Max scaling:

$$\bar{x}_i = \frac{x_i - x_{\min}}{x_{\max} - x_{\min}}, \tag{25}$$

where $x_{\min}$ and $x_{\max}$ denote the minimum and maximum values computed from the training set, respectively.

To mitigate the impact of the early health stage, a piecewise linear RUL function is employed (see Fig. 5). In this study, the constant interval is set to $N_H = 125$ cycles. The RUL label y is computed as:

$$y = \begin{cases} N_H, & \text{if } T_{\text{total}} - t \geq N_H \\ T_{\text{total}} - t, & \text{if } T_{\text{total}} - t < N_H. \end{cases} \tag{26}$$

3.2.3 Hyperparameter Sensitivity and Configuration

To determine the architecture of DCTE, grid search is conducted on the embedding dimension $H \in \{16, 32, 64, 128, 256\}$ and the stacking depth $L \in \{1, 2, 4, 6, 8\}$. As shown in Fig. 6, model performance is more sensitive to the stacking depth. The RMSE decreases significantly when L increases from 1 to 4, while performance gradually saturates for deeper structures. Therefore, $L = 4$ is selected in subsequent experiments. For the embedding dimension, FD001 and FD003 achieve the best performance with $H = 32$, whereas FD002 and FD004 require a larger hidden dimension ($H = 64$) due to their increased operational complexity. The sensitivity of the TACL module is further analyzed on FD001 with respect to the temperature coefficient T and the number of sub-regions K. As shown in Fig. 7, a very small temperature leads to overly sharp distributions, while a large temperature weakens feature compactness. The best trade-off is obtained at $T = 0.2$. Similarly, $K = 5$ provides the best balance between structural discrimination and noise robustness. The final hyperparameter configuration is summarized in Table 3.

Table 3. Experimental hyperparameter configuration

Parameter	Value	Description
Batch size	256	Input batch size
Learning rate	10^{-3}	Learning rate for optimizer
w	40	Window size
Optimizer	AdamW	Optimization algorithm
Epochs	100	Number of training epochs
T	0.2	Temperature coefficient
P	2	Positive sample size
K	5	Number of sub-regions
μ	0.5	Balancing coefficient
H	32 / 64	Embedding dimension
L	4	Stack depth

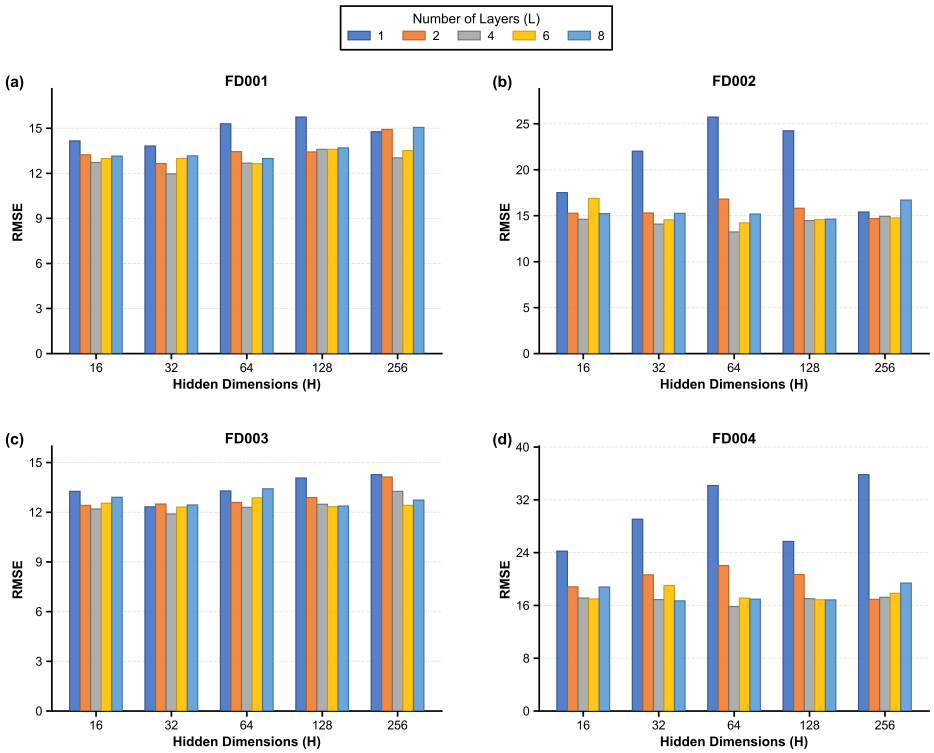


Figure 6. Effect of embedding dimensionality H and stacking layers L on RMSE across FD001–FD004. Panels (a)–(d) correspond to FD001–FD004.

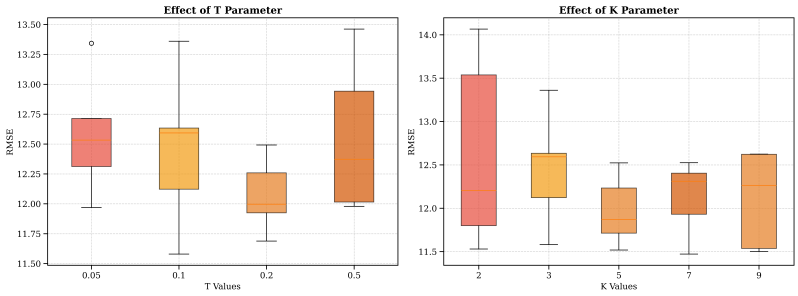


Figure 7. RMSE across different values of temperature coefficient T and number of subregions K.

3.2.4 Feature Representation Analysis

To investigate whether TACL improves the organization of learned representations, t-SNE visualization is performed on the extracted features. Fig. 8 presents the feature distributions of late-stage samples, while Fig. 9 illustrates the evolution of feature embeddings over the full degradation process. Compared with the DCTE baseline, the proposed CoDeR framework yields more compact and structured feature distributions. In contrast, the embeddings produced by DCTE show scattered regions and local overlaps, whereas CoDeR produces smoother transitions across degradation stages. These results suggest that TACL helps organize degradation-related samples in a more coherent manner, which is beneficial for downstream RUL regression. In addition, the learned feature embeddings exhibit clearer progression trends along the degradation process, with samples from adjacent degradation stages distributed more coherently.

3.2.5 RUL Prediction Results

Following the optimal configurations, 10 independent trials are conducted for both DCTE and CoDeR to reduce the influence of random initialization. The quantitative results are summarized in Table 4. CoDeR consistently outperforms the DCTE baseline across all C-MAPSS subsets. Compared with DCTE, the proposed framework achieves lower RMSE and Score values together with smaller standard deviations, indicating improved prediction robustness. Fig. 10 illustrates the prediction results on the test engines. Most predicted trajectories remain close to the ground truth, and the prediction errors are relatively concentrated. In

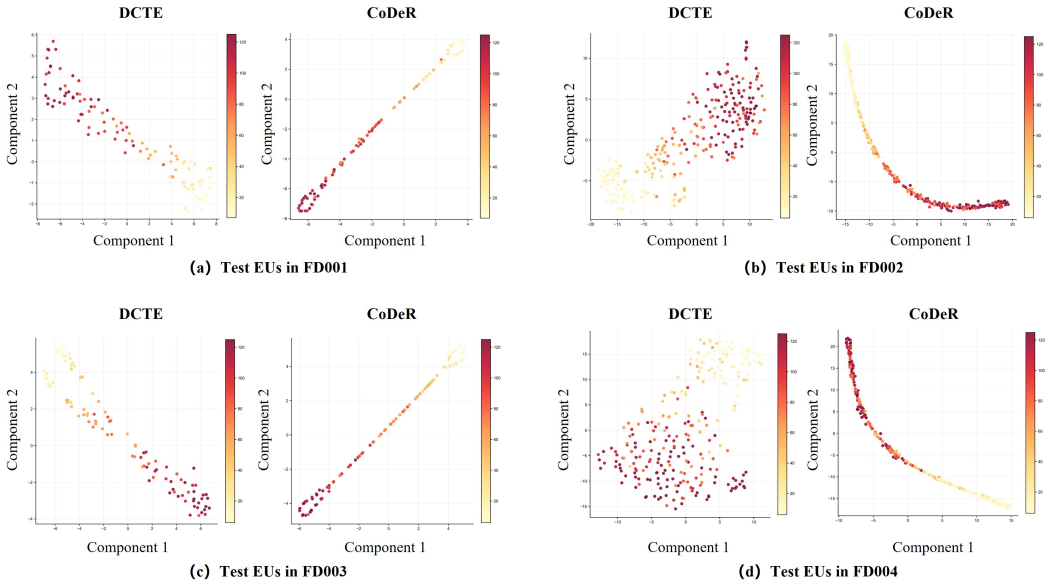


Figure 8. T-SNE visualization of end-stage engine samples for DCTE and CoDeR. Colors transition from light (high RUL) to dark (low RUL).

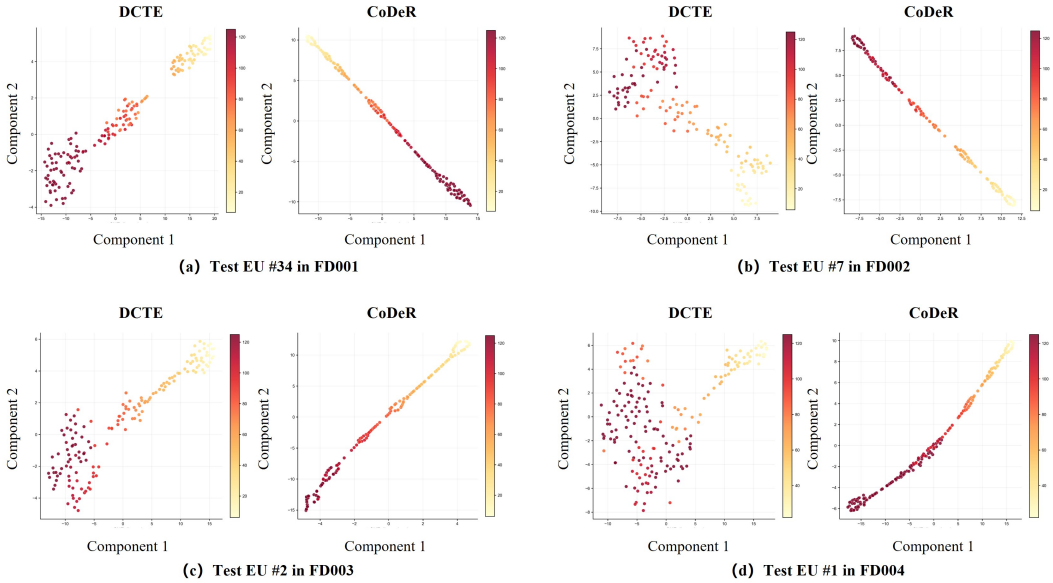


Figure 9. T-SNE plots of full lifecycle feature trajectories for single engines using DCTE and CoDeR.

particular, CoDeR produces more stable predictions during the degradation stage, especially under the multi-condition settings of FD002 and FD004. Representative prediction trajectories are further shown in Fig. 11. Compared with DCTE, CoDeR generates smoother regression curves with fewer abrupt fluctuations. These results suggest that the more organized latent representations learned by TACL contribute to more reliable RUL estimation.

3.2.6 Attention Pattern Analysis

To better understand the feature weighting behavior of DCTE, the learned attention maps are visualized in Fig. 12. Specifically, self-attention weights from the final encoder layer are projected back to the input space through the embedding matrix to generate the spatiotemporal attribution maps. According to the C-MAPSS sensor definitions, the highly weighted channels mainly correspond to pressure-related variables (s_7, s_{11}), fan/core speed measurements (s_8, s_9, s_{13}, s_{14}), and the fuel-flow-to-P_{s30} ratio (s_{12}), while s_{17} (bleed enthalpy) receives relatively low attention. In particular, s_{12} and s_{14} exhibit the strongest responses,

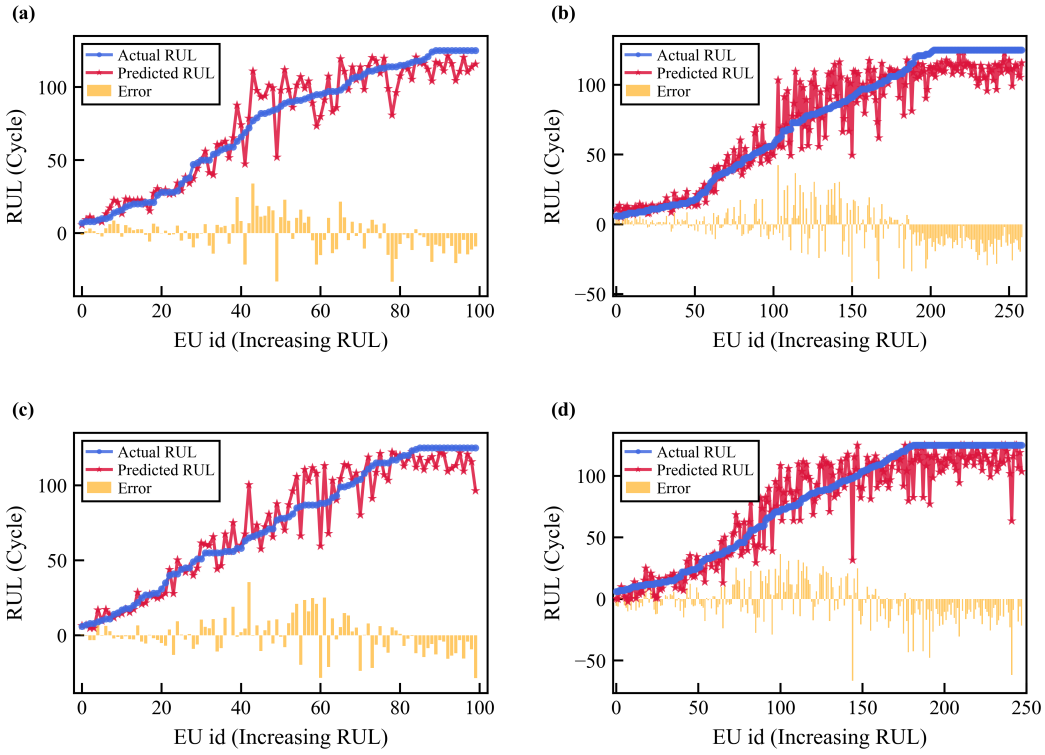


Figure 10. Comparison between the predicted RUL and ground-truth RUL at the final point of test EUs: (a) FD001, (b) FD002, (c) FD003, (d) FD004.

Table 4. Performance comparison between DCTE and CoDeR

Method	Metric	FD001	FD002	FD003	FD004
DCTE	RMSE	12.59 ± 0.42	14.03 ± 0.34	12.17 ± 0.22	16.71 ± 0.41
	Score	236 ± 29	857 ± 98	226 ± 15	1291 ± 163
CoDeR	RMSE	$\mathbf{11.87 \pm 0.32}$	$\mathbf{13.52 \pm 0.21}$	$\mathbf{11.53 \pm 0.10}$	$\mathbf{15.68 \pm 0.55}$
	Score	$\mathbf{204 \pm 18}$	$\mathbf{703 \pm 57}$	$\mathbf{205 \pm 15}$	$\mathbf{1023 \pm 116}$

followed by s_7, s_8, s_9, and s_{13}. Moreover, different temporal segments receive different attention levels during the degradation process. These observations indicate that DCTE learns meaningful spatiotemporal feature selection patterns from multivariate sensor sequences.

3.2.7 Comparative Experiments

The proposed framework is compared with several representative deep learning and Transformer-based RUL prediction methods, as well as recent advanced paradigms, and the results are summarized in Table 5. Overall, CoDeR achieves the best average RMSE and Score among all compared methods, demonstrating its effectiveness for aero-engine RUL prediction on the C-MAPSS benchmark.

Specifically, CoDeR achieves the best performance on FD002 and FD004 in terms of both RMSE and Score. Since these two subsets involve multiple operating conditions and more complex degradation dynamics, the results demonstrate the capability of the proposed framework to learn more discriminative degradation representations under challenging working environments.

Compared with conventional CNN- and RNN-based approaches (e.g., DCNN and LSTM), CoDeR delivers a substantial performance gain, accompanied by significantly lower prediction errors. More importantly, when compared with recent state-of-the-art methods—including advanced Transformer variants, multi-axis spatiotemporal architectures, and self-supervised contrastive learning frameworks—the proposed approach consistently exhibits superior overall performance. Combined with the feature visualization results, these findings suggest that CoDeR can more effectively structure degradation representations in the latent feature space, thereby facilitating more accurate and reliable RUL prediction.

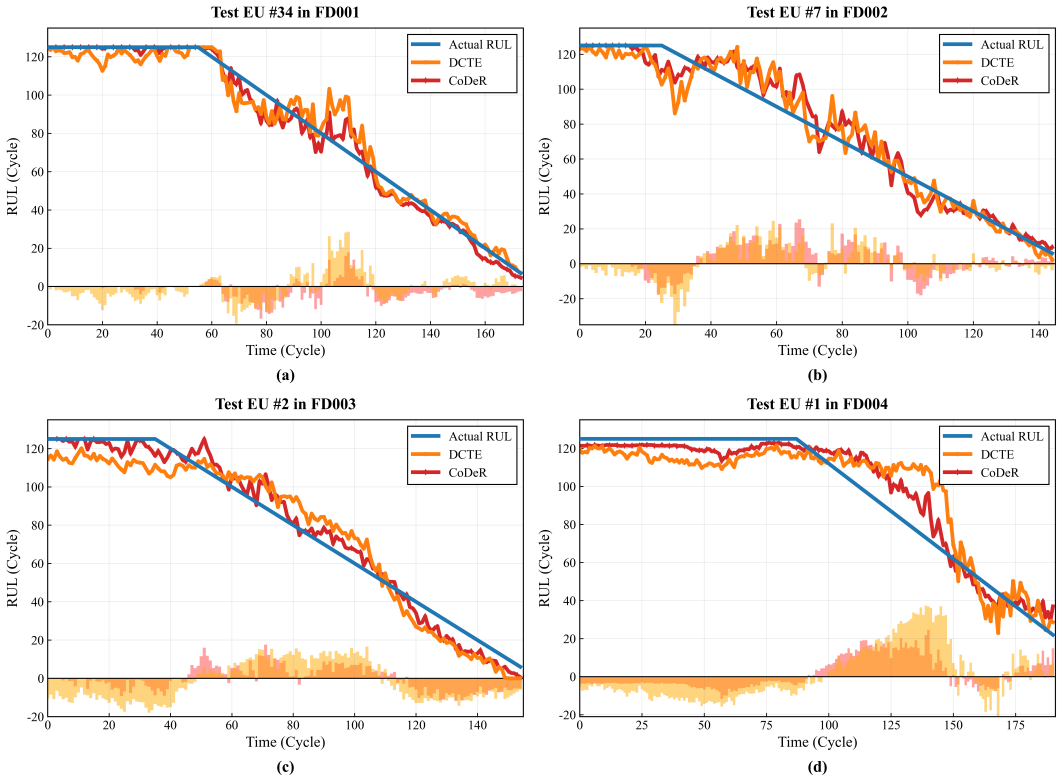


Figure 11. RUL prediction trajectories and error histograms for representative EUs: (a) EU 34 in FD001, (b) EU 7 in FD002, (c) EU 2 in FD003, (d) EU 1 in FD004.

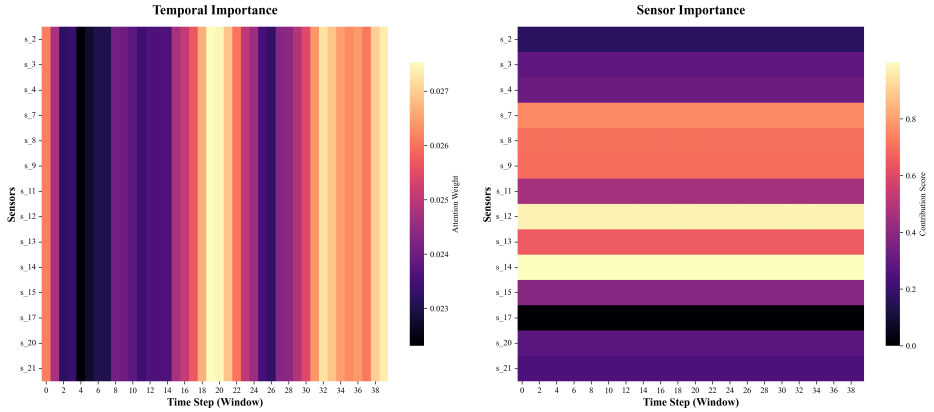


Figure 12. Self-Attention mechanism analysis for Test unit #34 in FD001 dataset

3.2.8 Ablation Study

To evaluate the contribution of different components, ablation studies are conducted on both DCTE and TACL. For DCTE, three variants are considered: (1) removing the cross-attention interaction, (2) removing the spatial branch, and (3) removing the temporal branch. The results in Table 6 show that all components contribute positively to prediction performance. Among them, the temporal branch has the largest impact, especially on FD002 and FD004, indicating the importance of temporal dependency modeling under complex operating conditions.

To further validate the effectiveness of TACL, the proposed objective is replaced with the standard InfoNCE loss while retaining the same neighborhood-based positive sampling and interval-aware negative sampling strategy. The results in Table 7 show that the vanilla InfoNCE formulation leads to noticeable performance degradation, especially under multi-condition settings. Moreover, the t-SNE visualization in Fig. 13 reveals scattered and mixed feature distributions, indicating insufficient

Table 5. Performance comparison with state-of-the-art methods on C-MAPSS

Method	FD001		FD002		FD003		FD004		Avg	
	RMSE	Score	RMSE	Score	RMSE	Score	RMSE	Score	RMSE	Score
CoDeR (Ours)	11.87	204	**13.52**	**703**	11.53	205	**15.68**	**1023**	**13.15**	**533.8**
MACPTD[31]	11.08	197.91	14.34	923.69	11.24	**156.03**	18.27	1326.01	13.73	650.91
MSMT-FDA[15]	11.88	249.78	15.55	1051.54	**10.47**	259.46	15.93	1645.38	13.46	801.54
DCDM[32]	**10.75**	166.01	15.08	1639.14	11.26	198.89	16.24	1752.19	13.33	939.06
STG-TF[33]	11.17	**165.7**	14.92	909.9	11.41	215.8	17.32	1438.2	13.71	682.40
HMDAM[34]	10.82	170.07	15.33	1030.42	11.21	239.47	17.48	1738.19	13.71	794.54
Temporal Adjacent[35]	11.57	192	16.45	986	11.24	191	18.63	2043	14.47	852.3
ATCN[36]	11.48	194.3	15.82	1210.6	11.34	249.2	17.80	1934.9	14.11	897.2
TATFA[37]	12.21	261.5	15.07	1359.7	11.23	210.2	18.81	2506.4	14.33	1084.4
MSIDSN[38]	11.74	205	18.26	2046	12.04	196	22.48	2910	16.13	1339.3
DAST[16]	12.88	279	14.80	1100	12.51	399	17.43	1796	14.41	893.5
DCNN[6]	12.61	273	22.36	10412	12.64	284	23.31	1246	17.73	3053.8
LSTM[30]	16.14	338	24.49	4450	16.18	852	28.17	5550	21.25	2797.5

temporal continuity in the learned representations. In contrast, CoDeR produces smoother and more coherent degradation trajectories in the feature space. These results demonstrate that the proposed temporal-aware weighting mechanism effectively preserves degradation continuity during representation learning.

Table 6. Ablation study of DCTE components

Method	FD001		FD002		FD003		FD004	
	RMSE	Score	RMSE	Score	RMSE	Score	RMSE	Score
DCTE	**12.59**	**236**	**14.03**	**857**	**12.17**	**226**	**16.71**	**1291**
w/o Cross Attention	12.64	296	14.93	985	12.71	238	18.59	1732
w/o Spatial Part	12.76	315	15.52	1093	12.87	261	18.37	1750
w/o Temporal Part	13.67	386	18.56	1472	13.12	325	19.28	2283

Table 7. Ablation study results on TACL

Method	FD001		FD002		FD003		FD004	
	RMSE	Score	RMSE	Score	RMSE	Score	RMSE	Score
CoDeR	**11.87**	**204**	**13.52**	**703**	**11.53**	**205**	**15.68**	**1023**
w/o α_i, β_j	12.35	294	15.87	1183	12.25	236	18.28	1979

3.3 Generalization Study on N-CMAPSS

3.3.1 Data Description and Preprocessing

To further evaluate the robustness and cross-unit generalization capability of the proposed framework under realistic operating conditions, additional experiments are conducted on the N-CMAPSS dataset. Compared with the relatively simplified operating settings in C-MAPSS, N-CMAPSS contains high-frequency flight data collected under more realistic and dynamically varying operating conditions, including multiple flight phases and more complex degradation patterns. These characteristics introduce larger distribution variations and make the dataset more suitable for evaluating the robustness and cross-condition generalization ability of RUL prediction models.

In this study, the DS02 subset is selected for evaluation. Following the commonly adopted experimental setting, engine units 2, 5, 10, 16, 18, and 20 are used for training, while units 11, 14, and 15 are reserved for testing. The detailed experimental configuration is summarized in Table 8.

Considering the high sampling frequency and large data volume of N-CMAPSS, the raw sensor signals are downsampled before model training to reduce computational overhead. Fourteen sensor measurements together with four operating-condition

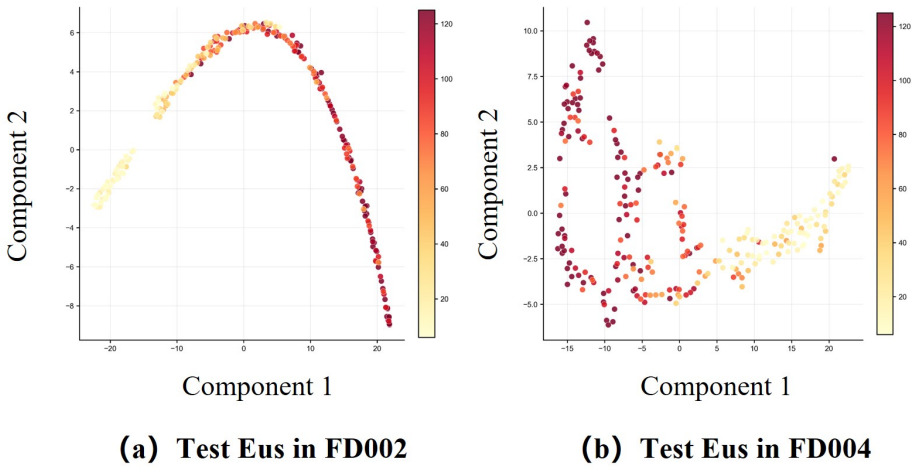


Figure 13. T-SNE visualization of embedding space trained with standard InfoNCE on FD002 and FD004.

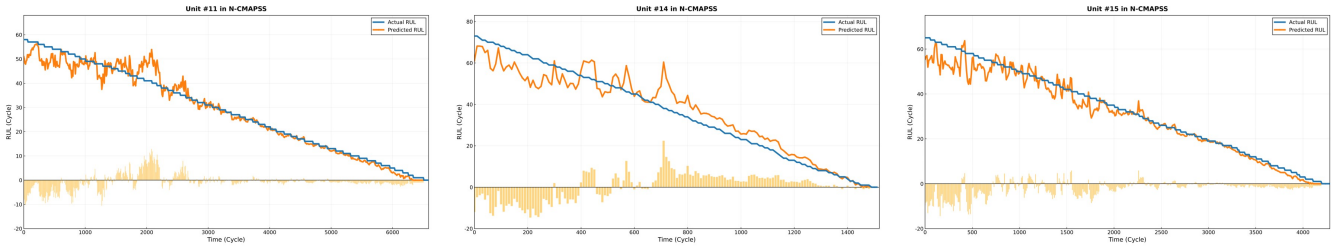


Figure 14. RUL prediction results of the proposed CoDeR model on representative engine units (#11, #14, and #15) from the N-CMAPSS dataset.

variables are selected as model inputs. All input features are normalized using Min-Max scaling, and the same sliding-window strategy adopted in the C-MAPSS experiments is applied for sample generation.

Table 8. Description of the N-CMAPSS DS02 sub-dataset

Description	Units	Flight Classes	Fault Modes	Training Units	Testing Units	Avg. Cycles
DS02	9	3	2	2, 5, 10, 16, 18, 20	11, 14, 15	72

3.3.2 Generalization Results

The quantitative comparison results on the N-CMAPSS dataset are summarized in Table 9. Compared with existing methods, the proposed CoDeR framework achieves the best overall RMSE and further improves upon the DCTE baseline. These results indicate that the proposed representation learning strategy remains effective under realistic and time-varying operating conditions.

Representative prediction trajectories are illustrated in Fig. 14. Despite the increased complexity of flight conditions and degradation patterns, CoDeR maintains relatively stable prediction behavior across different engine units. In particular, the predicted trajectories remain close to the ground truth during the degradation stage, suggesting that the learned representations transfer reasonably well to unseen operational environments.

To further investigate the generalization behavior of the learned representations, t-SNE visualization is conducted on the features extracted from the test set. As shown in Fig. 15, the representations learned by DCTE exhibit partial overlap and local fragmentation, whereas CoDeR forms a more compact and continuous feature distribution.Although the degradation trajectories in N-CMAPSS are more irregular than those in C-MAPSS, the proposed TACL module still improves the structural organization of the representation space under complex flight conditions.

Table 9. Performance comparison on the N-CMAPSS dataset

Method	RMSE
MOO-ELM[39]	7.29 ± 0.37
LRP[40]	9.84 ± 0.53
CPI-GAN[41]	3.82 ± 0.11
iRULER[42]	5.11 ± 0.53
GT-MRNet[43]	5.59 ± 0.11
DCTE (Ours)	3.92 ± 0.39
CoDeR (Ours)	$\mathbf{3.41 \pm 0.16}$

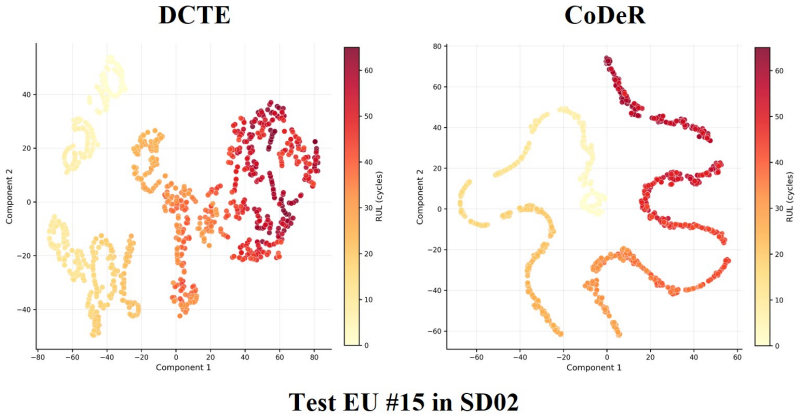


Figure 15. T-SNE visualization of feature representations learned by DCTE and CoDeR on the N-CMAPSS dataset.

4 Conclusion

This paper presents CoDeR, a novel degradation representation learning framework for RUL prediction. By integrating the DCTE backbone with the TACL module, CoDeR learns degradation-aware features from multivariate sensor data and improves the continuity of degradation trajectories in the representation space. Experiments on the C-MAPSS and N-CMAPSS benchmarks demonstrate competitive prediction accuracy and strong generalization under diverse operating conditions.

t-SNE visualizations further show that CoDeR produces smoother degradation trajectories than existing methods, enabling more stable degradation modeling and highlighting the effectiveness of degradation-oriented representation learning for prognostics.

Nevertheless, several limitations remain. The computational cost and real-time deployment of the Transformer-based architecture require further investigation, while the TACL module introduces additional optimization overhead during training. Future work will focus on improving computational efficiency and facilitating practical industrial deployment.

Funding

This work was supported by the Shanxi Key Research and Development Program of China (Grant No. 202402100101008).

Author contributions statement

W.L. conceived the framework and performed the experiments. R.N. analyzed the data and interpreted the results. S.H. conducted the data processing and result analysis. B.L. provided technical support and methodology design. Z.Z. supervised the research and revised the manuscript.

References

1. Yang, Z., Kanniainen, J., Krogerus, T. & Emmert-Streib, F. Prognostic modeling of predictive maintenance with survival analysis for mobile work equipment. *Sci. Reports* **12**, 8529, DOI: https://doi.org/10.1038/s41598-022-12572-z (2022).

2. Kothamasu, R., Huang, S. H. & VerDuin, W. H. System health monitoring and prognostics—a review of current paradigms and practices. *The Int. J. Adv. Manuf. Technol.* **28**, 1012–1024, DOI: https://doi.org/10.1007/978-1-84882-472-0_14 (2006).

3. Li, W. & Li, T. Comparison of deep learning models for predictive maintenance in industrial manufacturing systems using sensor data. *Sci. Reports* **15**, 23545, DOI: https://doi.org/10.1038/s41598-025-08515-z (2025).

4. Ferreira, C. & Gonçalves, G. Remaining useful life prediction and challenges: A literature review on the use of machine learning methods. *J. Manuf. Syst.* **63**, 550–562, DOI: https://doi.org/10.1016/j.jmsy.2022.05.010 (2022).

5. Hu, G. *et al.* Remaining useful life prediction of mechanical equipment based on time-series auto-correlation decomposition and cnn. *Meas. Sci. Technol.* **35**, 105104, DOI: https://doi.org/10.1088/1361-6501/ad5c8c (2024).

6. Li, X., Ding, Q. & Sun, J.-Q. Remaining useful life estimation in prognostics using deep convolution neural networks. *Reliab. Eng. & Syst. Saf.* **172**, 1–11, DOI: https://doi.org/10.1016/j.ress.2017.11.021 (2018).

7. Heimes, F. O. Recurrent neural networks for remaining useful life estimation. In *2008 international conference on prognostics and health management*, 1–6, DOI: https://doi.org/10.1109/phm.2008.4711422 (IEEE, 2008).

8. Zheng, S., Ristovski, K., Farahat, A. & Gupta, C. Long short-term memory network for remaining useful life estimation. In *2017 IEEE international conference on prognostics and health management (ICPHM)*, 88–95, DOI: https://doi.org/10.1109/icphm.2017.7998311 (IEEE, 2017).

9. Elsheikh, A., Yacout, S. & Ouali, M.-S. Bidirectional handshaking lstm for remaining useful life prediction. *Neurocomputing* **323**, 148–156, DOI: https://doi.org/10.1016/j.neucom.2018.09.076 (2019).

10. Wang, Y., Suo, C. & Zhao, Y. Multi-head attention-based variational autoencoders ensemble for remaining useful life prediction of aero-engines. *Meas. Sci. Technol.* **36**, 016131, DOI: https://doi.org/10.1109/tencon.2019.8929267 (2025).

11. Ragab, M. *et al.* Attention-based sequence to sequence model for machine remaining useful life prediction. *Neurocomputing* **466**, 58–68, DOI: https://doi.org/10.1016/j.neucom.2021.09.022 (2021).

12. Chen, Z. *et al.* Machine remaining useful life prediction via an attention-based deep learning approach. *IEEE Transactions on Ind. Electron.* **68**, 2521–2531, DOI: https://doi.org/10.1109/tie.2020.2972443 (2020).

13. Zhang, Y. & Yan, J. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *The eleventh international conference on learning representations* (2023).

14. Fu, S. *et al.* Pstformer: A novel parallel spatial-temporal transformer for remaining useful life prediction of aeroengine. *Expert. Syst. with Appl.* **265**, 125995, DOI: https://doi.org/10.1016/j.eswa.2024.125995 (2025).

15. Qin, M., Xu, G., Chen, X. & Li, W. Multi-sensor spatial and multiscale temporal transformer guided by frequency-domain attention for aeroengine remaining useful life prediction. *Reliab. Eng. & Syst. Saf.* 112535, DOI: https://doi.org/10.1016/j.ress.2026.112535 (2026).

16. Zhang, Z., Song, W. & Li, Q. Dual-aspect self-attention based on transformer for remaining useful life prediction. *IEEE Transactions on Instrumentation Meas.* **71**, 1–11, DOI: https://doi.org/10.1109/tim.2022.3160561 (2022).

17. Chen, T., Kornblith, S., Norouzi, M. & Hinton, G. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, 1597–1607, DOI: 10.48550/arXiv.2002.05709 (PmLR, 2020).

18. He, K., Fan, H., Wu, Y., Xie, S. & Girshick, R. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 9729–9738, DOI: https://doi.org/10.1109/cvpr42600.2020.00975 (2020).

19. Oord, A. v. d., Li, Y. & Vinyals, O. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748* DOI: 10.48550/arXiv.1807.03748 (2018).

20. Wu, Y., Dong, Z., Chen, C., Zhou, W. & Zhou, J. H. Mixup your own pairs. *arXiv preprint arXiv:2309.16633* DOI: 10.48550/arXiv.2309.16633 (2023).

21. Khosla, P. *et al.* Supervised contrastive learning. *Adv. neural information processing systems* **33**, 18661–18673, DOI: 10.48550/arXiv.2004.11362 (2020).

22. Liu, J. & Chen, S. Timesurl: Self-supervised contrastive learning for universal time series representation learning. In *Proceedings of the AAAI conference on artificial intelligence*, vol. 38, 13918–13926, DOI: https://doi.org/10.1609/aaai.v38i12.29299 (2024).

23. Abdelfattah, M., Hassan, M. & Alahi, A. Maskclr: Attention-guided contrastive learning for robust action representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 18678–18687, DOI: https://doi.org/10.1109/cvpr52733.2024.01767 (2024).

24. Zha, K., Cao, P., Son, J., Yang, Y. & Katabi, D. Rank-n-contrast: learning continuous representations for regression. *Adv. Neural Inf. Process. Syst.* **36**, 17882–17903, DOI: https://doi.org/10.52202/075280-0786 (2023).

25. Zhu, Q., Zhou, Z., Li, Y. & Yan, R. Contrastive bilstm-enabled health representation learning for remaining useful life prediction. *Reliab. Eng. & Syst. Saf.* **249**, 110210, DOI: https://doi.org/10.1016/j.ress.2024.110210 (2024).

26. Li, Y. *et al.* Domain invariant and consistent ordinal representation learning for remaining useful life prediction of bearings. *IEEE Transactions on Ind. Informatics* DOI: https://doi.org/10.1109/tii.2024.3454772 (2024).

27. Saxena, A., Goebel, K., Simon, D. & Eklund, N. Damage propagation modeling for aircraft engine run-to-failure simulation. In *2008 international conference on prognostics and health management*, 1–9, DOI: 10.1109/PHM.2008.4711414 (IEEE, 2008).

28. Arias Chao, M., Kulkarni, C., Goebel, K. & Fink, O. Aircraft engine run-to-failure dataset under real flight conditions for prognostics and diagnostics. *Data* **6**, 5, DOI: 10.3390/data6010005 (2021).

29. Vaswani, A. *et al.* Attention is all you need. *Adv. neural information processing systems* **30**, DOI: https://doi.org/10.65215/nxvz2v36 (2017).

30. Zheng, S., Ristovski, K., Farahat, A. & Gupta, C. Long short-term memory network for remaining useful life estimation. In *2017 IEEE international conference on prognostics and health management (ICPHM)*, 88–95, DOI: https://doi.org/10.1109/icphm.2017.7998311 (IEEE, 2017).

31. Qian, W., Chu, Y., Lu, Z., Tian, L. & Yu, B. Macptd: Self-supervised rul prediction for aero-engines via frequency-constrained disentanglement and dual-domain contrastive learning. *IEEE Sensors J.* **26**, 3237–3249, DOI: 10.1109/JSEN.2025.3633792 (2025).

32. Cao, L., Li, Y., Chen, J., Wu, W. & Zhang, H. Dual-channel degradation monitoring based on graph neural network for aero-engine remaining useful life prediction. *IEEE Transactions on Instrumentation Meas.* **74**, 1–19, DOI: 10.1109/TIM.2025.3560753 (2025).

33. Liu, Z.-Z., Sun, T. & Sun, X.-M. A spatial-temporal graph structure automatic feedback learning system with tensor fusion and its application on engine rul prediction. *Energy* 137253, DOI: 10.1016/j.energy.2025.137253 (2025).

34. He, C., Li, Z., Zheng, C., Zhang, Z. & Zhang, L. A hybrid model based on a dual-attention mechanism for the prediction of remaining useful life of aircraft engines. *Sensors* **25**, 5682, DOI: 10.3390/s25185682 (2025).

35. Zhu, J., Ma, J., Wu, J. & Fan, L. Efficient temporal adjacent transformer for remaining useful life prognostics. *Eng. Appl. Artif. Intell.* **166**, 113572, DOI: https://doi.org/10.1016/j.engappai.2025.113572 (2026).

36. Zhang, Q., Liu, Q. & Ye, Q. An attention-based temporal convolutional network method for predicting remaining useful life of aero-engine. *Eng. Appl. Artif. Intell.* **127**, 107241, DOI: https://doi.org/10.1016/j.engappai.2023.107241 (2024).

37. 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. *Reliab. Eng. & Syst. Saf.* **241**, 109662, DOI: https://doi.org/10.1016/j.ress.2023.109662 (2024).

38. Zhao, K., Jia, Z., Jia, F. & Shao, H. Multi-scale integrated deep self-attention network for predicting remaining useful life of aero-engine. *Eng. Appl. Artif. Intell.* **120**, 105860, DOI: https://doi.org/10.1016/j.engappai.2023.105860 (2023).

39. Mo, H. & Iacca, G. Multi-objective optimization of extreme learning machine for remaining useful life prediction. In *International conference on the applications of evolutionary computation (Part of EvoStar)*, 191–206, DOI: 10.1007/978-3-031-02462-7_13 (Springer, 2022).

40. Koutroulis, G., Mutlu, B. & Kern, R. Constructing robust health indicators from complex engineered systems via anticausal learning. *Eng. Appl. Artif. Intell.* **113**, 104926, DOI: 10.1016/j.engappai.2022.104926 (2022).

41. Xiong, J., Fink, O., Zhou, J. & Ma, Y. Controlled physics-informed data generation for deep learning-based remaining useful life prediction under unseen operation conditions. *Mech. Syst. Signal Process.* **197**, 110359, DOI: https://doi.org/10.1016/j.ymssp.2023.110359 (2023).

42. Qin, L., Zhang, S., Sun, T. & Zhao, X. An interpretable neuro-dynamic scheme with feature-temporal attention for remaining useful life estimation. *IEEE Transactions on Ind. Informatics* **20**, 5505–5516, DOI: https://doi.org/10.1109/tii.2023.3333933 (2023).

43. Ren, L., Wang, H., Mo, T. & Yang, L. T. A lightweight group transformer-based time series reduction network for edge intelligence and its application in industrial rul prediction. *IEEE Transactions on Neural Networks Learn. Syst.* DOI: https://doi.org/10.1109/tnnls.2023.3347227 (2024).