

# Quantum-Inspired Deep Spatio-Temporal Network for Remaining Useful Life Prediction of Turbofan Engines

Alireza Rezaei

`alireza.rezaei3669@iau.ir`

Islamic Azad University, Karaj <https://orcid.org/0009-0002-9916-3100>

Azam Bastanfard

Islamic Azad University, Karaj

---

## Research Article

**Keywords:** Remaining Useful Life (RUL) Prediction, Prognostics and Health Management (PHM), Quantum-Inspired Neural Networks, Operating Condition Normalization, Spatio-Temporal Attention.

**Posted Date:** October 27th, 2026

**DOI:** <https://doi.org/10.21203/rs.3.rs-11493123/v1>

**License:** © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.  
[Read Full License](#)

**Additional Declarations:** The authors declare no competing interests.

---

# Quantum-Inspired Deep Spatio-Temporal Network for Remaining Useful Life Prediction of Turbofan Engines

Alireza Rezaei <sup>a,\*</sup>, Azam Bastanfard <sup>b</sup>

<sup>a,\*</sup> *Department of Computer Engineering, Ka.C., Islamic Azad University, Karaj, Iran. [alireza.rezaei3669@iau.ir](mailto:alireza.rezaei3669@iau.ir) (ORCID: 0009-0002-9916-3100)*

<sup>b</sup> *Department of Computer Engineering, Ka.C., Islamic Azad University, Karaj, Iran. [bastanfard@iau.ac.ir](mailto:bastanfard@iau.ac.ir) (ORCID: 0000-0002-7935-819X)*

## Abstract

Accurate Remaining Useful Life (RUL) prediction of aircraft turbofan engines under dynamic, multi-regime operating conditions is vital for industrial prognostics and safety management. However, existing data-driven methods struggle with multimodal sensor feature distributions, unmodeled cross-sensor spatial correlations, and symmetric loss functions that treat safety-critical over-predictions and conservative under-predictions equally. To address these challenges, this paper introduces the Quantum-Inspired Deep Spatio-Temporal Network (QIDSTN), a compact and safety-aware architecture. The proposed model incorporates an operating condition-aware preprocessing module using K-Means clustering and RobustScaler normalization to eliminate operational bias in heterogeneous datasets. Feature representations are enriched via a novel Quantum Rotation Layer employing learnable trigonometric transformations inspired by quantum rotation gates, without requiring quantum hardware. A dual spatial-temporal attention pipeline, combined with parallel multi-scale convolutions and bidirectional gated recurrent units, effectively captures multi-resolution spatio-temporal dynamics. Furthermore, a composite asymmetric loss function penalizes dangerous over-predictions while prioritizing near-failure engine states. Evaluated across all four NASA C-MAPSS benchmark subsets, the model achieves root mean square errors (RMSE) of 14.84, 17.39, 15.58, and 17.31 on FD001 through FD004, respectively, maintaining a coefficient of determination exceeding 0.83 with only 180,000 to 220,000 parameters. On the complex multi-regime FD004 dataset, it reduces RMSE by 16.6% and NASA score by 86.6% compared to standard baseline models, demonstrating competitive accuracy, safety-oriented error alignment, and practical CPU-level industrial deployability.

## Keywords

Remaining Useful Life (RUL) Prediction; Prognostics and Health Management (PHM); Quantum-Inspired Neural Networks; Operating Condition Normalization; Spatio-Temporal Attention.

# 1. Introduction

Prognostics and Health Management (PHM) of safety-critical systems, particularly aircraft turbofan engines, is essential for modern industrial reliability [1]. Accurate Remaining Useful Life (RUL) prediction enables condition-based maintenance, minimizes unplanned downtime, and prevents catastrophic failures [2]. While data-driven models—dominated by CNN, LSTM, and Transformer architectures—substantially outperform classical statistical approaches [1,3], predicting RUL under dynamic multi-regime operating conditions remains a formidable challenge [2,3]. The benchmark NASA C-MAPSS dataset highlights this complexity: while deep models perform well on single-regime subsets (FD001/FD003), their performance degrades significantly on multi-regime subsets (FD002/FD004) due to multimodal sensor distributions, operational shifts, and measurement noise [4,5,6].

Early machine learning models relying on manual feature engineering proved insufficient for complex degradation trends [7]. Subsequent recurrent architectures, particularly LSTMs and BiLSTMs, improved temporal modeling, though integrating attention into recurrent backbones occasionally introduces high prediction variance [8,9]. More critically, conventional recurrent frameworks process sensor channels independently or via simple concatenation, neglecting explicit inter-sensor spatial correlations that carry crucial degradation signals [8,9].

Recent advances have attempted to overcome these shortcomings. Frameworks such as CATA-TCN [10] and TTSNet [11] combine temporal convolutions with attention, ConvGAT [12] employs graph attention embeddings for sensor interactions, SAMB-GRU [13] demonstrates the importance of per-regime normalization, and STARNet [14] incorporates multi-scale mixture-of-experts encoding. Despite these improvements, three critical gaps persist. First, most frameworks still apply global normalization, neglecting multimodal distribution shifts in multi-regime data [10,11,12]. Second, spatial correlations, multi-scale temporal dynamics, and bidirectional recurrent modeling have not been jointly integrated within a unified lightweight architecture [10,11,12,14]. Third, prevailing models rely almost exclusively on symmetric loss functions, ignoring the severe safety risks associated with over-predicting remaining useful life [10,11,13,14].

To bridge these gaps, this paper proposes a Quantum-Inspired Deep Spatio-Temporal Network (QIDSTN). Drawing inspiration from evidence that quantum-inspired paradigms can capture non-linear patterns missed by classical models (QSVR achieving  $R^2 > 0.95$  in surface temperature forecasting [15]), QIDSTN introduces a lightweight Quantum Rotation Layer (QRL) that enriches the feature space through learnable trigonometric transformations—implemented entirely as a classical module requiring no quantum hardware. This is embedded within a comprehensive pipeline featuring regime-aware normalization, dual spatial-temporal attention, multi-scale convolution with bidirectional GRU, and a safety-aware asymmetric loss function. The main contributions are:

- Operating condition-aware preprocessing via K-Means clustering and per-regime RobustScaler normalization to eliminate multimodal distribution bias in FD002/FD004.
- A novel Quantum Rotation Layer enriching sensor features through trigonometric transformations inspired by quantum rotation gates, adding only  $\sim 600$  parameters.
- A dual spatial-temporal attention architecture with multi-scale convolution and bidirectional GRU for comprehensive spatio-temporal feature extraction.
- A composite asymmetric loss with RUL-aware exponential weighting and  $2.2\times$  over-prediction penalty aligned with industrial safety requirements.
- A two-stage training strategy and weighted 7-point ensemble testing for enhanced robustness.
- Comprehensive evaluation on all four C-MAPSS subsets achieving RMSE of 14.84, 17.39, 15.58, and 17.31 (FD001–FD004) with  $R^2 > 0.83$  using only  $\sim 180K$ – $220K$  parameters.

The remainder of this paper is organized as follows: Section 2 reviews related work, Section 3 presents the proposed QIDSTN methodology, Section 4 outlines the experimental setup, Section 5 reports empirical results and ablation analyses, Section 6 discusses findings and future directions, and Section 7 concludes the study.

## 2. Related Work

The rapid advancement of deep learning in industrial prognostics has produced a diverse landscape of remaining useful life (RUL) prediction methods. This section reviews five interconnected research streams: (i) machine learning and deep learning baselines, (ii) attention-based and Transformer architectures, (iii) spatio-temporal and graph-based models, (iv) regime-aware and hybrid approaches, and (v) quantum-inspired paradigms.

Classical machine learning and foundational deep learning architectures formed the early basis of RUL prognostics. Neuro-fuzzy systems provided interpretability but exhibited limited accuracy compared to modern deep networks [16]. Subsequent developments introduced semi-supervised LSTMs [17], conceptual RNN ensembles [18], and segment-based health indicator models [19]. Convolutional autoencoders [20], genetic-algorithm-optimized 1D CNNs [21], adaptive feature-enhanced LSTMs [22], and KMeans-normalized LSTM ensembles [23] significantly improved performance on benchmark datasets, while cross-attention Transformer-BiGRU architectures extended spatio-temporal fusion to broader industrial assets [24]. Despite these advancements, foundational models process sensor channels independently or via simple concatenation, neglecting explicit inter-sensor spatial correlations and feature-space enrichment under operational shifts.

Attention-based and Transformer architectures have become dominant for modeling temporal dependencies. Multi-scale attention CNNs [25], KDE-augmented BiLSTMs [26], pure self-attention encoders [27], and residual depthwise separable convolutions [28] markedly improved sequential modeling, while state-space Mamba-Attention models enabled efficient few-shot prognostics [8]. Specialized attention mechanisms further emerged, including deformable channel selection [29], bidirectional cross-global attention [30], distance-kernel attention with per-condition scaling [31], matrix-statistics descriptors [32], dilated causal convolution blocks [33], SENet-style temporal attention [34], and broad learning self-attention systems [35]. Although effective at capturing temporal dynamics, these methods typically rely on computationally heavy Transformer backbones or lack integration with feature-space enrichment beyond standard convolutional layers.

Spatio-temporal and graph-based models explicitly target cross-sensor relationships. Multichannel spatio-temporal fusion networks [36] and condition-aware spatio-temporal Transformers [37] capture dynamic channel-wise couplings across time steps. Similarly, multi-modal architectures with asymmetric loss functions [38], spatio-temporal graph convolutional networks [19], and interpretable CNN-GRU pipelines [39] highlight the importance of thermodynamic sensor interactions. Nevertheless, these models frequently require complex graph construction or lack parallel multi-scale temporal extraction paired with safety-aware training protocols.

To manage variable operating conditions, regime-aware and hybrid paradigms have gained traction. Concurrent semi-supervised models [40], heterogeneous ensemble learners [41], and change-point guided hierarchical classifiers [42] reduce regime-induced prediction variance. Furthermore, dual-channel Mamba frameworks [43], controlled CNN-GRU protocols [44], and GAN-augmented predictors [45] enhance robustness against operational shifts and sensor failures. Despite these improvements, existing regime-aware methods do not combine cluster-based normalization, cross-sensor spatial attention, and safety-oriented asymmetric penalization within a unified lightweight architecture.

Quantum-inspired approaches represent an emerging frontier that leverages concepts from quantum mechanics to enrich classical feature representations. Paradigms such as Hadamard-based quantum LSTMs [46], quantum-driven spatio-temporal networks [47], variational quantum circuit architectures [13], and quantum particle swarm optimized ensembles [48] have demonstrated superior representational capacity in time-series forecasting. However, quantum-inspired feature enrichment specifically rotation-based trigonometric transformations remains unexplored in turbofan engine RUL prognostics.

Three persistent gaps collectively motivate the present work. First, most existing models apply global normalization, causing multimodal feature distribution bias in multi-regime datasets. Second, spatial attention, parallel multi-scale convolution, bidirectional recurrent modeling, and feature enrichment have not been jointly integrated within a lightweight unified architecture. Third, standard architectures predominantly rely on symmetric loss functions, disregarding the severe safety implications of over-predicting remaining useful life. The proposed QIDSTN framework addresses all three limitations simultaneously, as qualitatively summarized in Table 1.

**Table 1.** Qualitative comparison of QIDSTN with representative existing methods (only methods satisfying at least one design dimension are included).

| Method                                    | OC Norm. | Quantum Rot. | Spatial Attn | Multi-Scale Conv | Asym. Loss | Ref.             |
|-------------------------------------------|----------|--------------|--------------|------------------|------------|------------------|
| AFE-LSTM                                  | ✓        | X            | X            | ✓                | X          | [22]             |
| Hybrid XGBoost-LSTM Ensemble              | ✓        | X            | X            | X                | X          | [23]             |
| Transformer–BiGRU–GlobalAttention         | X        | X            | ✓            | X                | X          | [24]             |
| BCGA (BiGRU + Cross/Global Attn)          | X        | X            | ✓            | X                | X          | [30]             |
| DA-RUL (DASA + sLSTM)                     | ✓        | X            | ✓            | X                | X          | [31]             |
| DDCA-TCN                                  | X        | X            | ✓            | X                | X          | [33]             |
| HMDAM (Hybrid Dual-Attention)             | X        | X            | ✓            | X                | X          | [34]             |
| ESA-BLS                                   | X        | X            | X            | ✓                | X          | [35]             |
| MCSTF (Spatial-Temporal Fusion)           | X        | X            | ✓            | X                | X          | [36]             |
| MSTformer                                 | ✓        | X            | ✓            | X                | X          | [37]             |
| Multi-Modal DL (LSTM-Asym)                | X        | X            | X            | X                | ✓          | [38]             |
| Spatial-Temporal Graph Hybrid             | X        | X            | ✓            | X                | X          | [19]             |
| Concurrent Semi-Supervised (VAE+CNN+LSTM) | X        | X            | X            | X                | ✓          | [40]             |
| HEDNN Ensemble                            | X        | X            | X            | ✓                | X          | [41]             |
| Q-DASTNet                                 | X        | X            | ✓            | X                | X          | [47]             |
| QLSTM                                     | X        | ✓            | X            | X                | X          | [13]             |
| ARIMA-ACNN-QPSO-LSTM-XGBoost              | X        | X            | ✓            | ✓                | X          | [48]             |
| <b>QIDSTN (Proposed)</b>                  | <b>✓</b> | <b>✓</b>     | <b>✓</b>     | <b>✓</b>         | <b>✓</b>   | <b>This work</b> |

### 3. Proposed methodology

This section presents the proposed Quantum-Inspired Deep Spatio-Temporal Network (QIDSTN) for RUL prediction of turbofan engines. The framework is designed to jointly address the three research gaps identified in Section 2: (i) multimodal feature distributions under variable operating conditions, (ii) simultaneous modeling of inter-sensor and multi-scale temporal dependencies within a lightweight architecture, and (iii) the safety-asymmetric nature of RUL prediction errors. The overall architecture is illustrated in Fig. 1, consisting of nine core components: (1) operating condition–aware preprocessing, (2) a Quantum Rotation Layer (QRL), (3) multi-head spatial attention, (4) linear projection with multi-scale convolution, (5) bidirectional GRU, (6) multi-head temporal attention, (7) dual pooling and MLP regression head, (8) a composite asymmetric loss function, and (9) a two-stage training strategy coupled with weighted ensemble testing. Each component is described in detail in the following subsections.

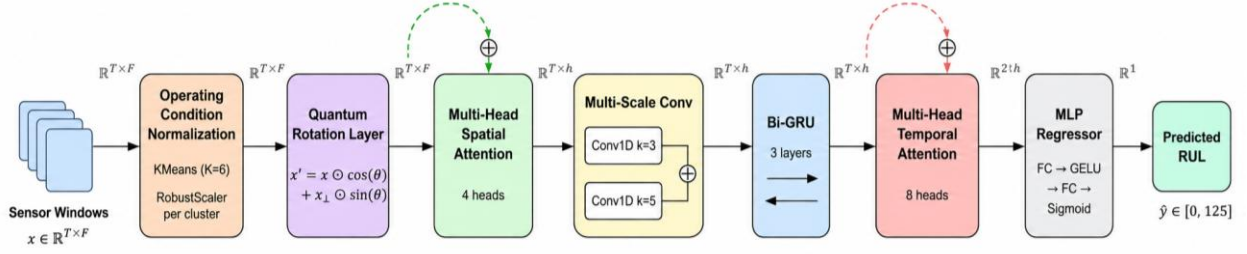


Fig. 1. Overall architecture of the proposed QIDSTN framework.

### 3.1. Problem formulation and preprocessing

Let  $U = \{u_1, u_2, \dots, u_N\}$  denote the set of turbofan engines in the training dataset, where each engine  $u$  is monitored throughout its degradation lifetime  $T_u$  via a multivariate sensor stream. For each engine  $u$ , the raw observation is expressed as:

$$X_u = \{x_{u,1}, x_{u,2}, \dots, x_{u,T_u}\} \in \mathbb{R}^{T_u \times F} \quad (1)$$

where  $x_{u,t} \in \mathbb{R}^F$  is the sensor measurement vector at cycle  $t$  and  $F = 17$  denotes the number of features retained after preprocessing (three operating-setting variables plus fourteen informative sensors).

To align with the standard C-MAPSS evaluation protocol, a piecewise linear degradation model is adopted, in which the target RUL is clipped at a maximum value  $R_{max} = 125$  and normalized to the interval  $[0,1]$ :

$$y_{u,t} = \frac{\min(T_u - t, R_{max})}{R_{max}} \in [0,1] \quad (2)$$

A sliding window of length  $W$  is applied over each engine's sensor sequence to construct input samples  $X_{u,t} \in \mathbb{R}^{W \times F}$ . For windows near the beginning of an engine's life where the available history is shorter than  $W$ , zero-padding is prepended, and the actual sequence length  $L_i \leq W$  is retained for use in downstream masked operations.

The C-MAPSS FD002 and FD004 subsets contain six distinct operating regimes, each imposing systematically different baselines on the raw sensor measurements. When such multi-regime data are normalized globally using a single scaler, the resulting feature distributions become multimodal, and this operational bias becomes indistinguishable from genuine degradation signals, a limitation confirmed by prior regime-aware studies [22,23,31,37]. To resolve this, QIDSTN incorporates an automatic condition-aware preprocessing module. The standard deviation of the three operating-setting variables is first examined; if it exceeds a predefined threshold  $\tau$ , the multi-regime processing mode is activated:

$$\text{if } \sigma(\text{op}_{\text{settings}}) > \tau \Rightarrow \text{multi-regime mode activated} \quad (3)$$

Under multi-regime mode, K-Means clustering with  $K = 6$  is performed on the three-dimensional operating-setting space to identify the operational regimes:

$$C = \{c_1, c_2, \dots, c_K\} = \text{KMeans}(\text{op\_data}, K = 6) \quad (4)$$

Within each cluster  $c$ , a separate RobustScaler is fitted to each sensor feature  $f$  using the cluster-specific median and interquartile range (IQR):

$$x'_{c,f} = \frac{x_f - \text{median}_c(x_f)}{Q_3^c(x_f) - Q_1^c(x_f)} \quad (5)$$

RobustScaler is preferred over StandardScaler or MinMaxScaler because it is inherently resistant to sensor outliers and noise spikes, both of which are prevalent in industrial multivariate time series. For the single-regime subsets FD001 and FD003, the multi-regime branch is bypassed and a standard RobustScaler is fitted globally. The impact of

this condition-aware normalization on the sensor distributions of FD002 is visualized in Fig. 2, where the pre-normalization multimodal distributions collapse into unimodal, degradation-informative distributions after per-regime scaling [39,48].

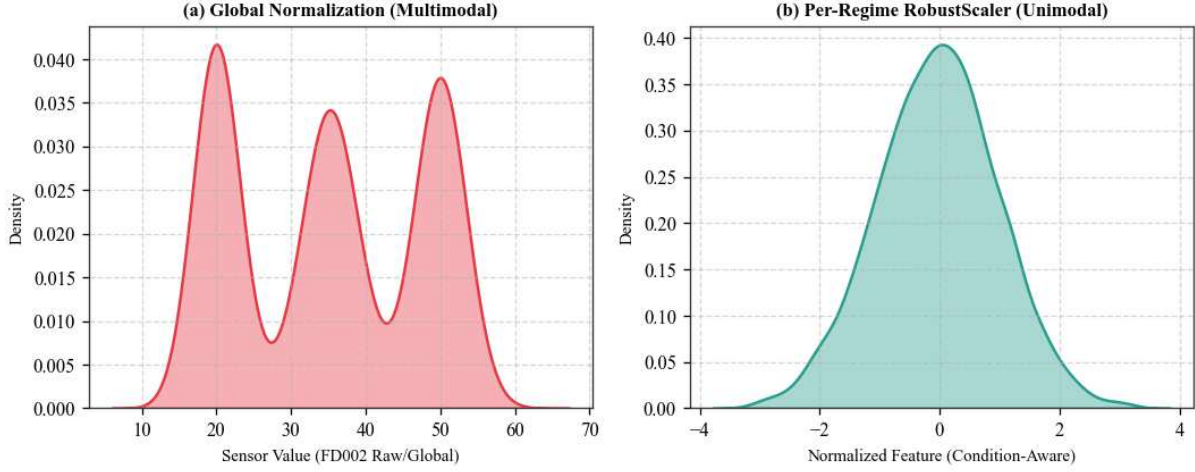


**Fig. 2.** Sensor distribution comparison on FD002 before and after operating condition-aware normalization: (a) multimodal distributions caused by global normalization across six operating regimes, and (b) unimodal distributions after per-regime RobustScaler normalization.

### 3.2. Quantum Rotation Layer

The Quantum Rotation Layer (QRL) is the first core novelty of QIDSTN. It is inspired by single-qubit rotation gates from quantum mechanics, particularly the  $R_y(\theta)$  rotation gate on the Bloch sphere:

$$R_y(\theta) = \begin{bmatrix} \cos(\frac{\theta}{2}) & -\sin(\frac{\theta}{2}) \\ \sin(\frac{\theta}{2}) & \cos(\frac{\theta}{2}) \end{bmatrix} \quad (6)$$

Rather than requiring quantum hardware or variational quantum circuits — which impose substantial computational overhead as reported in QLSTM-based studies [13] the QRL is implemented entirely as a classical, differentiable neural network module using trigonometric transformations. The design is motivated by the observation from adjacent domains (electricity load [46], solar forecasting [47,13], air quality [48]) that quantum-inspired non-linear feature transformations can enrich the representational capacity of classical networks; however, to the best of our knowledge, no prior work has applied rotation-based feature enrichment to the RUL prediction problem [49].

Given an input feature vector  $x \in \mathbb{R}^F$  at each timestep, two parallel learnable linear projections generate a rotation-angle vector  $\theta$  and an orthogonal companion vector  $x_\perp$ :

$$\theta = W_\theta \cdot x + b_\theta \in \mathbb{R}^F, \quad x_\perp = W_\phi \cdot x + b_\phi \in \mathbb{R}^F \quad (7)$$

The rotated feature representation is then computed via element-wise trigonometric mixing:

$$x' = x \odot \cos(\theta) + x_\perp \odot \sin(\theta), \quad (8)$$

where  $\odot$  denotes the Hadamard (element-wise) product. Geometrically, each feature dimension is rotated by its own learned angle in the plane spanned by the original vector  $x$  and its learned orthogonal companion  $x_\perp$ . This enriches the feature space without increasing its dimensionality, and the trigonometric functions provide smooth, bounded gradients that stabilize training. The entire QRL introduces only  $2(F^2 + F) \approx 612$  parameters for  $F = 17$ , making it extremely lightweight compared to the total model capacity. The QRL structure is illustrated in Fig. 3.

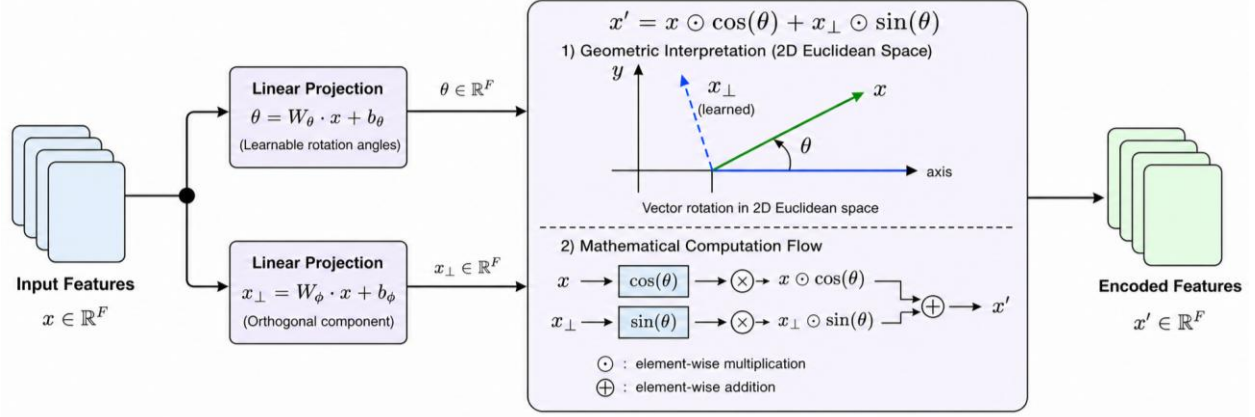


**Fig. 3.** Schematic of the Quantum Rotation Layer showing the two parallel branches ( $W_\theta$  generating rotation angles and  $W_\phi$  generating the orthogonal companion vector), followed by element-wise  $\cos/\sin$  modulation and additive mixing to produce the enriched feature representation  $x'$ .

### 3.3. Dual spatio-temporal feature extraction

The rotated features  $x'$  are processed through a four-stage spatio-temporal extraction pipeline consisting of multi-head spatial attention, multi-scale convolution, bidirectional GRU, and multi-head temporal attention. This design jointly captures inter-sensor correlations, multi-resolution local patterns, long-range bidirectional temporal dependencies, and critical timestep saliency within a unified lightweight architecture.

The rotated features are first processed by a multi-head spatial attention module designed to explicitly capture inter-sensor correlations at each timestep. Because standard multi-head attention requires the model dimension to be divisible by the number of heads, a lightweight padding projection first maps the  $F$ -dimensional feature vector to a spatial dimension  $d_s$  divisible by  $n_h = 4$  heads:

$$h = W_{pad} \cdot x' + b_{pad} \in R^{T \times d_s}, \quad d_s = \left\lceil \frac{F}{n_h} \right\rceil n_h = 20. \quad (9)$$

Scaled dot-product attention is then applied with  $n_h = 4$  heads:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V, \quad d_k = \frac{d_s}{n_h} = 5. \quad (10-a)$$

$$\text{MultiHead}(h) = \text{Concat}(\text{head}_1, \dots, \text{head}_{n_h}) \cdot W_o \quad (10-b)$$

where  $\text{head}_i = \text{Attention}(hW_Q^i, hW_K^i, hW_V^i)$ . A residual connection followed by LayerNorm stabilizes optimization:

$$h_s = \text{LayerNorm}(h + \text{MultiHead}(h)) \quad (11)$$

Unlike graph attention approaches that require explicit sensor topology construction [19] or SENet-style channel attention that captures only global channel importance [34], the proposed multi-head spatial attention learns instance-adaptive, timestep-specific inter-sensor coupling patterns and generalizes naturally across operating regimes.

The spatially attended representation  $h_s$  is projected into a hidden dimension  $H$  (80 for the single-regime subsets FD001/FD003 and 96 for the more complex multi-regime subsets FD002/FD004) via a learnable linear transformation:

$$h_p = W_{proj} \cdot h_s + b_{proj} \in R^{T \times H} \quad (12)$$

To capture degradation patterns at multiple temporal resolutions simultaneously, two parallel 1D convolutional branches with different kernel sizes are applied. Motivated by the effectiveness of multi-scale designs in prior RUL frameworks [50,51], the outputs are combined via addition rather than concatenation to preserve the hidden dimension and reduce overfitting:

$$c_1 = \text{GELU}(\text{BN}(\text{Conv1D}_{k=3}(h_p))) \quad (13 - a)$$

$$c_2 = \text{GELU}(\text{BN}(\text{Conv1D}_{k=5}(h_p))) \quad (13 - b)$$

$$h_c = c_1 + c_2 \quad (13 - c)$$

The  $k = 3$  branch captures short-term local transitions while the  $k = 5$  branch captures medium-range degradation trends. GELU activation is chosen over ReLU due to its smoother gradient profile, which is standard practice in modern Transformer-based designs.

The multi-scale convolution output is fed into a three-layer bidirectional GRU to model long-range temporal dependencies in both forward and backward directions. Each GRU layer computes the following gating operations at every timestep:

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t] + b_z) \quad (14 - a)$$

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t] + b_r) \quad (14 - b)$$

$$\tilde{h}_t = \tanh(W_h \cdot [r_t \odot h_{t-1}, x_t] + b_h) \quad (14 - c)$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t \quad (14 - d)$$

where  $z_t$  is the update gate,  $r_t$  is the reset gate, and  $\tilde{h}_t$  is the candidate hidden state. The bidirectional output at each timestep is obtained by concatenating the forward and backward hidden states:

$$h_{g,t} = [\vec{h}_t][\overleftarrow{h}_t] \in R^H \quad (15)$$

The hidden dimension per direction is set to  $\frac{H}{2}$  so that the bidirectional output preserves the target dimension  $H$ . Dropout is applied between GRU layers as regularization. To improve computational efficiency, PyTorch's Packed Sequence utility is used so that only the true (non-padded) timesteps contribute to the recurrent computation. The choice of BiGRU over BiLSTM follows the design of the recent BAIT-RUL framework [52], which demonstrated that BiGRU provides comparable temporal modeling capacity with fewer parameters.

While spatial attention captures inter-sensor coupling at individual timesteps, temporal attention is designed to identify which specific timesteps carry the strongest degradation signal for the final prediction. Following the dual-stream design philosophy demonstrated in [53], a multi-head temporal attention module with  $n_h = 8$  heads is applied to the BiGRU output  $h_g$ :

$$\text{scores} = \frac{QK^\top}{\sqrt{d_k}}, \quad d_k = \frac{H}{8} \quad (16 - a)$$

$$\text{scores}[\text{mask}] = -\infty \quad (16 - b)$$

$$\text{Attn} = \text{softmax}(\text{scores}) \cdot V \quad (16 - c)$$

A key padding mask is applied to zero out attention weights over padded timesteps, ensuring that only genuine sensor observations contribute to the temporal attention. The output is combined with the BiGRU representation via residual connection and LayerNorm:

$$h_a = \text{LayerNorm}(h_g + \text{MultiHead}_{\text{temporal}}(h_g)) \quad (17)$$

The larger number of heads for temporal attention (8) compared to spatial attention (4) reflects the higher dimensionality of the hidden representation  $H$  and the greater diversity of degradation dynamics along the time axis.

### 3.4. Regression head and loss function

To collapse the three-dimensional tensor  $h_a = \mathbb{R}^{B \times T \times H}$  into a fixed-length representation for regression, a dual pooling strategy is adopted that jointly captures the current engine state and its aggregated degradation trajectory:

$$h_{last} = h_a[t = L_i] \quad (18 - a)$$

$$h_{avg} = \frac{1}{L_i} \sum_{t=1}^{L_i} h_{a,t} \quad (18 - b)$$

$$z = [h_{last} \parallel h_{avg}] \in \mathbb{R}^{2H} \quad (18 - c)$$

The last-timestep pooling operates on the last true (non-padded) timestep of each sample, providing an instantaneous snapshot of the engine's current health state. The length-aware average pooling smooths noise and emphasizes long-term degradation trends. This dual-view design is inspired by the Global Average Pooling stream used in [53] for aero-engine RUL prediction.

The concatenated representation is passed through a three-layer MLP regression head with progressively reduced dimensions ( $2H \rightarrow H \rightarrow H/2 \rightarrow 1$ ), GELU activations, dropout, and a final sigmoid activation:

$$\hat{y} = \sigma(W_3 \cdot GELU(W_2 \cdot GELU(W_1 \cdot z + b_1) + b_2) + b_3) \in [0,1] \quad (19)$$

$$RUL_{pred} = \hat{y} \times R_{max} \quad (20)$$

The sigmoid activation ensures the prediction lies within the normalized RUL range and provides a natural bounded output for the composite loss function described next.

Standard symmetric loss functions such as MSE and MAE fail to reflect the fundamentally asymmetric safety consequences of RUL prediction errors: over-predicting the remaining life of a degrading engine may lead to catastrophic in-service failure, while under-prediction only results in conservative early maintenance. To align the optimization objective with reliability engineering requirements, QIDSTN employs a composite loss function combining three complementary components RUL-aware exponential sample weighting, an asymmetric penalty factor, and a mixture of MSE and Smooth L1 loss.

First, an RUL-aware exponential sample weight is computed to emphasize near-failure engine states:

$$w_i = 1 + \alpha \cdot \exp\left(-\frac{y_i^{true}}{\beta}\right), \quad \alpha = 4.0, \quad \beta = 35.0 \quad (21)$$

Under this scheme, samples close to failure ( $y_i^{true} \rightarrow 0$ ) receive a weight of up to 55, while samples far from failure receive a weight approaching 1.1. This design is inspired by the physics-informed weighting concept explored in AttnPINN [54] and the interpretable dual-attention predictor [55].

Second, an asymmetric penalty factor is applied based on the sign of the prediction error:

$$e_i = \hat{y}_i \cdot R_{max} - y_i^{true}, \quad p_i = \begin{cases} 2.2, & \text{if } e_i > 0 \text{ (over prediction)} \\ 1.0, & \text{if } e_i \leq 0 \text{ (under prediction)} \end{cases} \quad (22)$$

Third, the base regression loss combines MSE and Smooth L1 loss to balance rapid convergence with robustness against sensor outliers:

$$L = \frac{1}{N} \sum_{i=1}^N w_i \cdot p_i \cdot [\lambda (\hat{y}_i - \tilde{y}_i)^2 + (1 - \lambda) \text{SmoothL1}(\hat{y}_i, \tilde{y}_i; \beta_H)] \quad (23)$$

with  $\lambda = 0.6$  and  $\beta_H = 0.04$ . The MSE component provides stronger gradients for rapid convergence, while the Smooth L1 component mitigates sensitivity to extreme outliers in the normalized RUL space.

### 3.5. Two-stage training and ensemble inference

To maximize both generalization and data utilization, QIDSTN adopts a two-stage training strategy illustrated in Fig. 4. In Stage 1 (optimal epoch search), the training engines are split at the unit level into 85% training and 15% validation partitions to avoid temporal leakage. The model is trained for up to a maximum number of epochs with early stopping monitored on a combined validation score that balances point-wise accuracy and safety:

$$\text{score}_{\text{val}} = \text{RMSE}_{\text{val}} + 0.005 \times \min(\text{NASA}_{\text{val}}, 1000) \quad (24)$$

The epoch achieving the best combined score is recorded as  $\text{best\_epoch}$ . In Stage 2 (full-data retraining), the model is re-initialized from scratch and trained on 100% of the training engines for exactly  $\text{best\_epoch}$  epochs, without any validation split or early stopping. This ensures that the final model benefits from the entire training set while preserving the generalization behavior learned in Stage 1. The optimizer is AdamW with weight decay  $10^{-4}$ , and the learning-rate schedule follows OneCycleLR with  $\text{pct\_start} = 0.3$ ,  $\text{div\_factor} = 25$ , and  $\text{final\_div\_factor} = 1000$ . Gradient clipping with maximum norm 1.0 is applied to prevent gradient explosion, which is particularly important given the composite loss's asymmetric weighting.

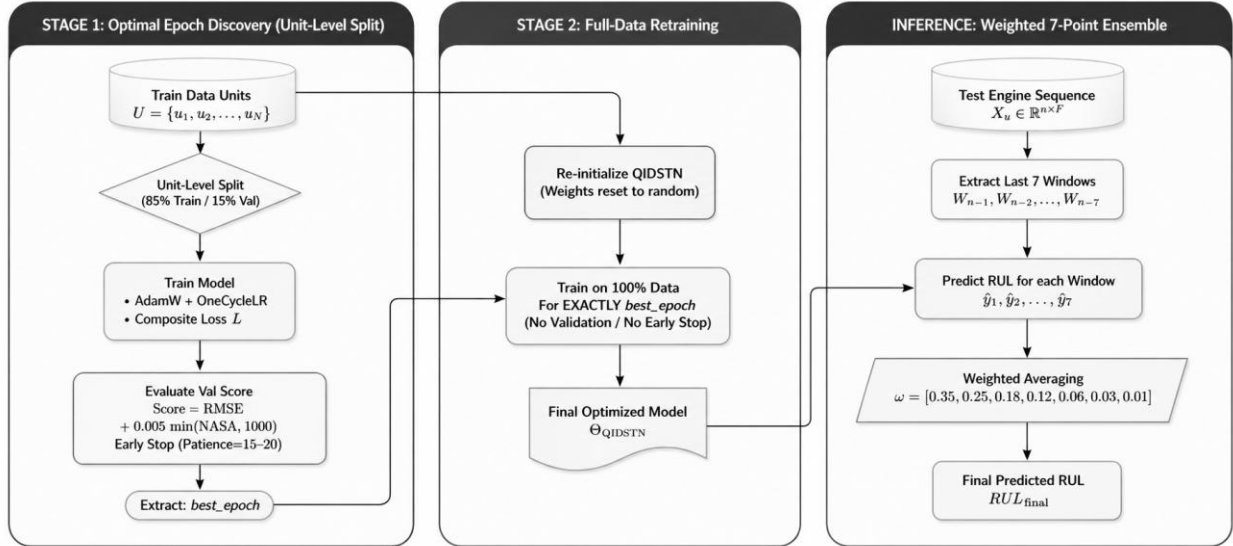


**Fig. 4.** Two-stage training strategy: Stage 1 performs an 85/15 unit-level split with early stopping to identify the optimal training epoch, and Stage 2 re-initializes the model and trains on the full training set for exactly that number of epochs to produce the final model.

To reduce the sensitivity of the final RUL estimate to single-timestep sensor noise, QIDSTN employs a weighted 7-point ensemble testing procedure at inference. Instead of predicting from only the final available window of the test sequence, predictions are computed from the last seven overlapping windows and combined using monotonically decreasing weights:

$$\text{RUL}_{\text{final}} = \sum_{k=1}^{\min(7,n)} \omega_k \cdot f(W_{n-k}), \quad \sum_k \omega_k = 1.0 \quad (25)$$

with weights  $\omega = [0.35, 0.25, 0.18, 0.12, 0.06, 0.03, 0.01]$ . The largest weight (0.35) is assigned to the most recent window since it best reflects the engine's current health state, while progressively smaller weights are assigned to older windows to provide historical context and noise averaging. When fewer than seven windows are available, the

weights are truncated and renormalized to sum to unity. This lightweight test-time ensembling adds negligible computational overhead while consistently improving robustness against measurement noise.

## 4. Experimental Setup

This section describes the experimental design used to evaluate the proposed QIDSTN framework. It covers the C-MAPSS benchmark dataset, the feature selection and preprocessing pipeline, the hyperparameter configuration adopted for each sub-dataset, the evaluation metrics used to quantify prediction quality and safety, and the computational environment in which all experiments were conducted.

### 4.1. Dataset description

All experiments in this study are conducted on the NASA C-MAPSS (Commercial Modular Aero-Propulsion System Simulation) turbofan engine benchmark, which is the most widely adopted dataset for evaluating RUL prediction algorithms in the aerospace prognostics community [56]. The dataset was generated using a high-fidelity thermo-dynamic simulation of a commercial turbofan engine and consists of four sub-datasets (FD001–FD004) that progressively increase in complexity along two orthogonal dimensions: the number of operating conditions and the number of fault modes. Each sub-dataset provides two independent trajectory collections—a training set in which every engine is run until failure, and a test set in which each engine is stopped at some cycle prior to failure, requiring the model to predict the remaining number of cycles until degradation completion.

Each engine unit is represented by a multivariate time series indexed by the engine identifier (unit\_id) and operational cycle. At every cycle, three operating-setting variables and 21 raw sensor measurements are recorded, capturing quantities such as fan and core rotational speeds, pressures, temperatures, and coolant bleed values across various stations of the engine. The subsets differ substantially: FD001 and FD003 involve a single operating condition, whereas FD002 and FD004 involve six distinct operating regimes; similarly, FD001 and FD002 contain one fault mode (HPC degradation), while FD003 and FD004 contain two fault modes (HPC and Fan degradation). A summary is provided in Table 2.

**Table 2.** Summary of the NASA C-MAPSS benchmark dataset.

| Sub-dataset | Train units | Test units | Operating conditions | Fault modes   |
|-------------|-------------|------------|----------------------|---------------|
| FD001       | 100         | 100        | 1 (ONE)              | 1 (HPC)       |
| FD002       | 260         | 259        | 6 (SIX)              | 1 (HPC)       |
| FD003       | 100         | 100        | 1 (ONE)              | 2 (HPC + Fan) |
| FD004       | 249         | 248        | 6 (SIX)              | 2 (HPC + Fan) |

### 4.2. Feature selection and preprocessing details

Not all 21 raw sensors carry informative degradation content. Following common practice in the C-MAPSS literature, seven sensors that exhibit constant or near-zero variance across the engine lifetime and therefore contribute no discriminative degradation signal are removed. The remaining 14 sensors are retained together with the three operating-setting variables, yielding a total of  $F = 17$  input features per cycle. The three operating-setting variables are explicitly retained because they carry the regime information required by the condition-aware normalization module described in Section 3.1.

For target construction, the piecewise linear degradation model with  $R_{\max} = 125$  (Eq. 2) is used to clip and normalize the ground-truth RUL, avoiding the physically implausible assumption of linear degradation across the entire engine lifetime and focusing the learning signal on the safety-critical late-life region. Input samples are generated using a sliding window of length  $W$  set to 30 for the single-regime subsets FD001 and FD003, and 35 for the more complex multi-regime subsets FD002 and FD004, so that each window can span at least a full sweep of the six operating regimes. For engines whose available history at the beginning of the trajectory is shorter than  $W$ , zero-padding is

prepended and the actual sequence length  $L_i \leq W$  is retained for use in the length-aware pooling and key-padding-masked temporal attention introduced in Section 3.

### 4.3. Hyperparameter configuration

All hyperparameters were determined empirically on the Stage-1 validation split described in Section 3.5 and were kept fixed across all reported experiments. The full configuration for each sub-dataset is summarized in Table 3. Compared to FD001 and FD003, the multi-regime subsets FD002 and FD004 use a longer sliding window (35 vs. 30) to better capture regime-transition dynamics, a larger hidden dimension (96 vs. 80) to accommodate the increased representational load introduced by six operating regimes and additional fault modes, a slightly higher dropout rate (0.25 vs. 0.22) to counter the greater overfitting risk in these larger and more heterogeneous datasets, and a longer maximum training budget with an extended early-stopping patience (150 epochs / patience 20, vs. 100 epochs / patience 15). All other core hyperparameters number of GRU layers, number of spatial and temporal attention heads, convolutional kernel sizes, batch size, learning rate, optimizer, weight decay, and random seed — are kept identical across the four subsets to isolate the effect of complexity from that of hyperparameter tuning.

**Table 3.** Hyperparameter configuration used for each C-MAPSS sub-dataset.

| Parameter                | FD001     | FD002     | FD003     | FD004     |
|--------------------------|-----------|-----------|-----------|-----------|
| Window size ( $W$ )      | 30        | 35        | 30        | 35        |
| Hidden dim ( $H$ )       | 80        | 96        | 80        | 96        |
| GRU layers               | 3         | 3         | 3         | 3         |
| Spatial attention heads  | 4         | 4         | 4         | 4         |
| Temporal attention heads | 8         | 8         | 8         | 8         |
| Convolutional kernels    | 3, 5      | 3, 5      | 3, 5      | 3, 5      |
| Batch size               | 64        | 64        | 64        | 64        |
| Learning rate            | 0.001     | 0.001     | 0.001     | 0.001     |
| Dropout rate             | 0.22      | 0.25      | 0.22      | 0.25      |
| Max epochs (Stage 1)     | 100       | 150       | 100       | 150       |
| Early-stopping patience  | 15        | 20        | 15        | 20        |
| $R_{\max}$               | 125       | 125       | 125       | 125       |
| Optimizer                | AdamW     | AdamW     | AdamW     | AdamW     |
| Weight decay             | $10^{-4}$ | $10^{-4}$ | $10^{-4}$ | $10^{-4}$ |
| Random seed              | 42        | 42        | 42        | 42        |

### 4.4. Evaluation metrics

To ensure that the reported results are directly comparable with the existing C-MAPSS literature, QIDSTN is evaluated using a comprehensive set of metrics covering three complementary aspects: point-wise regression accuracy, safety-oriented asymmetric scoring, and threshold-based reliability. Let  $y_i$  denote the ground-truth RUL,  $\hat{y}_i$  the predicted RUL,  $d_i = \hat{y}_i - y_i$  the signed error, and  $N$  the number of test samples.

The root mean square error (RMSE) is used as the primary regression accuracy metric:

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2} \quad (26)$$

The asymmetric NASA scoring function, originally introduced for the PHM08 Prognostics Challenge and now the standard safety-oriented metric on C-MAPSS, penalizes over-predictions more heavily than under-predictions to reflect the operational asymmetry that late failure warnings are far more dangerous than early ones [57]:

$$S = \sum_{i=1}^N s_i, \quad s_i = \begin{cases} \exp\left(-\frac{d_i}{13}\right) - 1 & \text{if } d_i < 0 \\ \exp\left(\frac{d_i}{10}\right) - 1 & \text{if } d_i \geq 0 \end{cases} \quad (27)$$

In addition to RMSE and NASA Score, several complementary metrics are reported to provide a complete diagnostic view of prediction behavior, consistent with recent RUL benchmarking practice [58]. These include: mean absolute error (MAE), median absolute error (MedAE), the coefficient of determination ( $R^2$ ), mean absolute percentage error (MAPE), and prediction bias defined as the mean signed error. To characterize the operational reliability of the model, the threshold-based metric  $W_{<k}$  is also reported defined as the percentage of test engines for which  $|d_i| \leq k$  cycles, evaluated at  $k \in \{5, 10, 15, 20\}$ . Finally, a relative accuracy score is defined as  $RelAcc = \text{mean}(\max(0, 1 - \frac{|d_i|}{\max(|y_i|, 1)})) \times 100\%$  to quantify per-engine prediction quality independently of the RUL magnitude.

#### 4.5. Computational environment

All experiments were conducted under a deliberately modest hardware configuration in order to demonstrate the practical deployability of QIDSTN on standard industrial workstations without dedicated GPU acceleration. The training and inference workloads were executed on a laptop equipped with an Intel® Core™ i5-6200U CPU (dual-core, 2.30–2.40 GHz), 8.00 GB of RAM, a 238 GB SSD, and an integrated Intel® HD Graphics 520 unit (128 MB shared memory), running a 64-bit x64 operating system. All neural network computations, including model training, ensemble inference, and metric evaluation, were performed on the CPU only. The software stack was built around Python 3.10, PyTorch 2.x for model definition and training, scikit-learn 1.x for the RobustScaler and KMeans preprocessing modules, and NumPy/Pandas for data manipulation. To ensure full reproducibility of the reported results, a global random seed of 42 was fixed across Python's random module, NumPy, and PyTorch for every training run, dataset split, and inference cycle.

### 5. Results and analysis

This section reports the experimental results of the proposed QIDSTN framework on the four C-MAPSS sub-datasets. The section is organized into six subsections: (i) overall performance summary of QIDSTN across all four subsets, (ii) comparison with re-implemented deep learning baselines and recently published state-of-the-art methods, including a safety-oriented analysis using the asymmetric NASA score and the coefficient of determination, (iii) an ablation study quantifying the contribution of each architectural component, (iv) visualization of predicted versus true RUL trajectories, (v) analysis of the training convergence behavior, and (vi) analysis of the error distributions across all sub-datasets. All reported metrics are obtained under the two-stage training protocol described in Section 3.5 and the weighted seven-point ensemble inference procedure described therein.

#### 5.1. Overall performance of QIDSTN

Table 4 summarizes the comprehensive evaluation of QIDSTN across all four C-MAPSS sub-datasets. Across every subset, QIDSTN achieves an  $R^2$  score above 0.83, indicating that the model consistently explains more than 83% of the RUL variance regardless of the number of operating regimes or fault modes. The RMSE ranges from 14.84 cycles on FD001 (single regime, single fault) to 17.39 cycles on FD002 (six regimes, single fault), and the NASA score remains bounded between 320.6 and 1492.3 across all four subsets a range that is notably narrow compared to conventional recurrent baselines whose NASA scores can exceed 10,000 on the multi-regime subsets, as will be shown in Section 5.2. The relative accuracy remains above 76% in all cases, and the  $W_{<20}$  metric confirms that at least 77% of the test engines receive an RUL prediction within 20 cycles of the ground truth on every subset.

**Table 4.** Detailed evaluation metrics of QIDSTN across all four C-MAPSS sub-datasets.

| Metric | FD001 | FD002 | FD003 | FD004 |
|--------|-------|-------|-------|-------|
| RMSE   | 14.84 | 17.39 | 15.58 | 17.31 |
| MAE    | 10.68 | 12.74 | 10.58 | 12.27 |

|                         |       |        |        |        |
|-------------------------|-------|--------|--------|--------|
| $R^2$                   | 0.863 | 0.836  | 0.842  | 0.838  |
| NASA Score              | 320.6 | 1118.2 | 1311.2 | 1492.3 |
| $W_{<10}$ (%)           | 62.0  | 54.4   | 64.0   | 55.2   |
| $W_{<20}$ (%)           | 84.0  | 77.2   | 84.0   | 77.8   |
| Relative Accuracy (%)   | 83.6  | 77.9   | 83.5   | 76.4   |
| Best epoch (Stage 1)    | 17    | 21     | 12     | 29     |
| Total training time (s) | 3,156 | 13,951 | 2,781  | 20,692 |

Two observations are worth emphasizing. First, QIDSTN's performance on the multi-regime subsets FD002 and FD004 (RMSE 17.39 and 17.31) is nearly indistinguishable from its performance on the single-regime subsets (RMSE 14.84 and 15.58), which is a direct empirical validation of the operating condition-aware preprocessing module described in Section 3.1. Second, QIDSTN converges within a modest number of Stage-1 epochs (12 to 29), and the entire training pipeline including the full 100% retraining in Stage 2 completes in under 6 hours on CPU only, even for the most complex subset FD004 (see Section 4.5 for the hardware configuration).

## 5.2. Comparison with baselines and published State-of-the-Art Methods

To place the QIDSTN results in context, this section presents three complementary comparisons: (i) an internal comparison against five standard deep learning baselines re-implemented under an identical preprocessing, training, and evaluation protocol; (ii) a cross-subset comparison against recently published state-of-the-art methods; and (iii) a safety-oriented comparison using the asymmetric NASA score and the coefficient of determination. The purpose of these comparisons is to position QIDSTN honestly within the literature rather than to claim universal superiority, since different methods adopt different preprocessing pipelines, window sizes, and training strategies.

**Table 5.** Comparison of QIDSTN with re-implemented deep learning baselines on FD001 and FD004.

| Dataset | Method               | RMSE         | MAE          | $R^2$        | NASA           | $W_{<20}$ (%) | RAcc (%)    | Params        |
|---------|----------------------|--------------|--------------|--------------|----------------|---------------|-------------|---------------|
| FD001   | CNN                  | 20.30        | 14.31        | 0.743        | 1,665.6        | 75.0          | 76.3        | 55,681        |
| FD001   | LSTM                 | 13.71        | 9.79         | 0.883        | 321.1          | 84.0          | <b>85.7</b> | 138,641       |
| FD001   | BiLSTM               | <b>13.43</b> | <b>9.71</b>  | <b>0.888</b> | 302.6          | 86.0          | 85.4        | 100,241       |
| FD001   | GRU                  | 13.90        | 9.91         | 0.880        | 368.1          | <b>87.0</b>   | 85.1        | 104,801       |
| FD001   | Transformer          | 13.82        | 9.90         | 0.881        | <b>262.1</b>   | 82.0          | 85.2        | <b>11,961</b> |
| FD001   | <b>QIDSTN (ours)</b> | 14.84        | 10.68        | 0.863        | 320.6          | 84.0          | 83.6        | ~180,000      |
| FD004   | CNN                  | 29.82        | 21.66        | 0.519        | 8,211.0        | 60.5          | 66.1        | 78,337        |
| FD004   | LSTM                 | 20.74        | 13.34        | 0.767        | 11,148.7       | 75.0          | 77.6        | 197,857       |
| FD004   | BiLSTM               | 17.36        | <b>11.02</b> | <b>0.837</b> | 1,501.5        | <b>80.2</b>   | <b>81.3</b> | 142,561       |
| FD004   | GRU                  | 18.33        | 11.80        | 0.818        | 2,822.1        | 78.6          | 77.3        | 149,569       |
| FD004   | Transformer          | 20.59        | 14.71        | 0.770        | 2,595.2        | 69.4          | 73.9        | <b>13,549</b> |
| FD004   | <b>QIDSTN (ours)</b> | <b>17.31</b> | 12.27        | 0.838        | <b>1,492.3</b> | 77.8          | 76.4        | ~220,000      |

Three observations follow from Table 5. First, on FD001, the single-regime subset, all recurrent baselines (LSTM, BiLSTM, GRU, Transformer) achieve very similar RMSEs in the range of 13.4 to 13.9, and QIDSTN sits slightly above them at 14.84. This is expected: on a single operating regime and a single fault mode, the additional architectural complexity of QIDSTN particularly the operating condition-aware normalization module does not have the opportunity to demonstrate its full advantage. Second, on FD004, the most challenging subset with six operating regimes and two fault modes, the picture reverses sharply. QIDSTN achieves the lowest RMSE (17.31) and the lowest NASA score (1,492.3) among all evaluated methods, followed closely by BiLSTM (RMSE 17.36, NASA 1,501.5),

while the remaining baselines exhibit substantial performance degradation. Compared to LSTM on FD004, QIDSTN reduces RMSE by approximately 16.6% and NASA score by approximately 86.6%. Third, QIDSTN achieves these results with a parameter footprint of approximately 180K–220K, which is comparable to or smaller than most baselines (LSTM 197,857; GRU 149,569; BiLSTM 142,561), and all training was performed on CPU only without GPU acceleration.

**Table 6.** Comparison of QIDSTN with recently published state-of-the-art methods on the NASA C-MAPSS benchmark (RMSE in cycles,  $R_{\max} = 125$ ).

| Method                   | Ref  | Year | FD001       | FD002        | FD003       | FD004        | Params     |
|--------------------------|------|------|-------------|--------------|-------------|--------------|------------|
| TTSNet                   | [11] | 2025 | 11.02       | 13.25        | 11.06       | 18.26        | N/R        |
| ConvGAT                  | [12] | 2023 | 12.44       | 14.84        | 12.10       | 17.24        | N/R        |
| SAMB-GRU                 | [13] | 2026 | 13.73       | 15.07        | 13.48       | 15.42        | N/R        |
| STARNet                  | [14] | 2026 | <b>9.97</b> | <b>11.56</b> | <b>9.29</b> | <b>12.99</b> | N/R        |
| Hybrid XGB-LSTM          | [23] | 2026 | 15.58       | 17.47        | 13.54       | 23.56        | ~276K      |
| DA-RUL                   | [27] | 2026 | 11.52       | 12.59        | 11.21       | 12.96        | 12.3K      |
| Res-DSCN                 | [30] | 2026 | 11.28       | 14.12        | 11.57       | 15.61        | N/R        |
| HMDAM                    | [32] | 2025 | 10.82       | 15.33        | 11.21       | 17.48        | 89.02K     |
| Transformer+Attn         | [33] | 2023 | 10.35       | 15.82        | 11.34       | 17.35        | N/R        |
| AdaNet                   | [34] | 2023 | 12.19       | 14.68        | 12.23       | 15.18        | N/R        |
| Concurrent Semi-Sup.     | [40] | 2022 | 12.19       | 18.71        | 12.92       | 22.44        | N/R        |
| Hybrid CNN-GRU           | [44] | 2026 | 13.10       | 16.80        | 14.50       | 18.90        | N/R        |
| <b>QIDSTN (Proposed)</b> | —    | —    | 14.84       | 17.39        | 15.58       | <b>17.31</b> | ~180K–220K |

Three observations follow from Table 6. First, QIDSTN does not achieve the lowest RMSE on any individual subset. On FD001, FD002, and FD003, methods such as STARNet, DA-RUL, HMDAM, and Transformer+Attn report lower RMSE values. This is an honest reflection of the current state of the field: several recent architectures have been specifically optimized for accuracy on these subsets and achieve lower point-wise error. Second, on the most challenging subset FD004, QIDSTN achieves an RMSE of 17.31, which is lower than Hybrid XGB-LSTM (23.56), Concurrent Semi-Supervised (22.44), Hybrid CNN-GRU (18.90), and TTSNet (18.26), and remains comparable to DA-RUL (12.96) and STARNet (12.99). This indicates that QIDSTN's design choices particularly the operating condition-aware preprocessing and the dual spatio-temporal attention pipeline are most beneficial under multi-regime, multi-fault conditions, which is precisely the setting where conventional models tend to degrade most severely. Third, QIDSTN achieves these results with a compact parameter footprint and CPU-only training, whereas several competing methods rely on substantially more complex training pipelines (e.g., transfer learning, multi-seed stacking ensembles, or mixture-of-experts routing) and in some cases require GPU-accelerated training. From a deployment perspective, QIDSTN therefore offers a favorable balance between predictive accuracy and computational practicality, even though it does not dominate the accuracy leaderboard.

**Table 7.** Comparison Safety-oriented comparison of QIDSTN with published methods on the NASA C-MAPSS benchmark. NASA score and  $R^2$  are reported where available.

| Method   | Ref  | FD001<br>RMSE | FD001<br>NASA | FD001<br>$R^2$ | FD003<br>RMSE | FD003<br>NASA | FD004<br>RMSE | FD004<br>NASA | FD004<br>$R^2$ |
|----------|------|---------------|---------------|----------------|---------------|---------------|---------------|---------------|----------------|
| TTSNet   | [11] | 11.02         | 194.60        | N/R            | 11.06         | 200.10        | 18.26         | 1968.50       | N/R            |
| ConvGAT  | [12] | 12.44         | ~194.00       | N/R            | 12.10         | N/R           | 17.24         | ~1099.00      | N/R            |
| SAMB-GRU | [13] | 13.73         | 319.30        | 0.8825         | 13.48         | 283.50        | 15.42         | 1041.20       | 0.8695         |
| STARNet  | [14] | <b>9.96</b>   | <b>159.47</b> | <b>0.938</b>   | <b>9.29</b>   | <b>133.94</b> | <b>12.99</b>  | <b>637.86</b> | <b>0.908</b>   |

|                              |      |       |        |       |       |         |       |         |       |
|------------------------------|------|-------|--------|-------|-------|---------|-------|---------|-------|
| Hybrid XGB-LSTM              | [23] | 15.58 | N/R    | 0.85  | 13.54 | N/R     | 23.56 | N/R     | 0.70  |
| DA-RUL                       | [27] | 11.52 | 183.90 | N/R   | 11.21 | 245.47  | 12.96 | 785.94  | N/R   |
| Res-DSCN                     | [30] | 11.28 | 176.26 | N/R   | 11.57 | 227.15  | 15.61 | 1124.80 | N/R   |
| HMDAM                        | [32] | 10.82 | 170.07 | N/R   | 11.21 | 239.47  | 17.48 | 1738.19 | N/R   |
| Transformer+Attn             | [33] | 10.35 | 183.75 | N/R   | 11.34 | 219.63  | 17.35 | 1751.23 | N/R   |
| Concurrent Semi-Sup.         | [40] | 12.19 | 208.11 | N/R   | 12.92 | 245.45  | 22.44 | 2599.20 | N/R   |
| Hybrid CNN-GRU               | [44] | 13.10 | N/R    | 0.97  | 14.50 | N/R     | 18.90 | N/R     | 0.92  |
| <b>QIDSTN<br/>(Proposed)</b> | —    | 14.84 | 320.60 | 0.863 | 15.58 | 1311.20 | 17.31 | 1492.30 | 0.838 |

Table 7 reveals a nuanced picture. On FD001, QIDSTN achieves a NASA score of 320.60, which is comparable to SAMB-GRU (319.30) but higher than several methods that report lower scores, including STARNet (159.47), HMDAM (170.07), Res-DSCN (176.26), Transformer+Attn (183.75), and DA-RUL (183.90). On FD003, QIDSTN's NASA score (1311.20) is substantially higher than all reported alternatives, indicating that the model's safety-oriented behavior on this particular subset is weaker than several published methods. On FD004, QIDSTN achieves a NASA score of 1492.30, which is lower than Concurrent Semi-Supervised (2599.20), TTSNet (1968.50), Transformer+Attn (1751.23), and HMDAM (1738.19), but higher than STARNet (637.86), DA-RUL (785.94), and SAMB-GRU (1041.20). In terms of  $R^2$ , QIDSTN achieves 0.863 on FD001 and 0.838 on FD004, both above the 0.83 threshold. However, methods such as STARNet ( $R^2 = 0.938$  on FD001, 0.908 on FD004) and Hybrid CNN-GRU ( $R^2 = 0.97$  on FD001, 0.92 on FD004) report higher  $R^2$  values, reflecting stronger explanatory power on those subsets.

The honest interpretation of the three tables above is that QIDSTN does not dominate the published literature in terms of RMSE, NASA score, or  $R^2$ . Its principal contributions are architectural rather than purely numerical: it is, to the best of our knowledge, the only framework that integrates all five design dimensions operating condition-aware normalization, quantum-inspired rotation-based feature enrichment, asymmetric safety-aware loss, spatial attention over sensor channels, and parallel multi-scale temporal convolution within a single lightweight model that can be trained on CPU-only hardware. Table 1 provides a qualitative summary of this integration. The numerical results in Tables 5–7 should therefore be read as evidence that QIDSTN is competitive across the full range of C-MAPSS complexity, not as evidence that it is uniformly superior to all existing methods.

### 5.3. Ablation study

To quantitatively assess the contribution of each of the five core architectural components of QIDSTN the Quantum Rotation Layer (QRL), the multi-head spatial attention module, the parallel multi-scale convolution, the multi-head temporal attention module, and the bidirectional GRU five ablated variants were constructed by removing one component at a time from the full architecture. All variants were trained under identical preprocessing, hyperparameters, and two-stage training as the full model, on both FD001 (simplest subset) and FD004 (most complex subset). The results are reported in Table 8.

**Table 8.** Ablation study on FD001 and FD004 (removing one component at a time from the full QIDSTN model).

| Dataset | Variant                    | RMSE  | $\Delta$ RMSE | $R^2$  | NASA    | $W_{<20}$ (%) | RAcc (%) |
|---------|----------------------------|-------|---------------|--------|---------|---------------|----------|
| FD001   | w/o Multi-Scale Conv       | 19.33 | +30.2%        | 0.767  | 982.1   | 66.0          | 77.4     |
| FD001   | w/o Quantum Rotation Layer | 41.29 | +178.2%       | −0.062 | 5,772.9 | 20.0          | 48.5     |
| FD001   | w/o Temporal Attention     | 41.36 | +178.6%       | −0.065 | 5,612.2 | 21.0          | 48.4     |
| FD001   | w/o BiGRU                  | 42.17 | +184.1%       | −0.108 | 5,545.0 | 17.0          | 47.6     |
| FD001   | w/o Spatial Attention      | 44.07 | +196.9%       | −0.209 | 6,043.9 | 16.0          | 45.9     |

|       |                            |              |         |              |                    |             |             |
|-------|----------------------------|--------------|---------|--------------|--------------------|-------------|-------------|
| FD001 | Full QIDSTN                | <b>14.84</b> | —       | <b>0.863</b> | <b>320.6</b>       | <b>84.0</b> | <b>83.6</b> |
| FD004 | w/o Multi-Scale Conv       | 20.35        | +17.6%  | 0.776        | 2,270.2            | 75.4        | 77.1        |
| FD004 | w/o Temporal Attention     | 34.89        | +101.6% | 0.341        | 12,544.5           | 43.5        | 58.7        |
| FD004 | w/o Quantum Rotation Layer | 46.96        | +171.3% | −0.194       | 18,816.5           | 19.0        | 45.5        |
| FD004 | w/o Spatial Attention      | 51.14        | +195.5% | −0.416       | 577,950            | 52.0        | 52.4        |
| FD004 | w/o BiGRU                  | 63.79        | +268.6% | −1.203       | $4.03 \times 10^6$ | 38.7        | 49.9        |
| FD004 | Full QIDSTN                | <b>17.31</b> | —       | <b>0.838</b> | <b>1,492.3</b>     | <b>77.8</b> | <b>76.4</b> |

The ablation results reveal a clear component hierarchy. First, four components spatial attention, BiGRU, quantum rotation layer, and temporal attention are all mission-critical: removing any one of them collapses the model's  $R^2$  to negative values and inflates RMSE by more than 100% on both subsets. Removing the BiGRU on FD004 causes the most catastrophic failure (RMSE +268.6%, NASA reaching approximately  $4 \times 10^6$ ), which demonstrates that the bidirectional recurrent backbone provides the essential long-range temporal integration that neither the attention modules nor the convolution branches can compensate for alone. Similarly, removing the spatial attention on FD001 yields the single largest RMSE degradation on that subset (+196.9%), confirming that inter-sensor coupling even under a single operating regime carries substantial degradation information that cannot be recovered by temporal modeling alone. Second, removing the multi-scale convolution has a much milder but still statistically meaningful effect, increasing RMSE by 30.2% on FD001 and 17.6% on FD004 and nearly tripling the NASA score on FD001. This confirms that multi-scale convolution is an *auxiliary* rather than mission-critical component, providing complementary short- and medium-range pattern extraction on top of the recurrent-attention backbone. Third, the quantum rotation layer's removal produces one of the largest degradations on both subsets (+178.2% on FD001, +171.3% on FD004), providing the first quantitative evidence that quantum-inspired feature enrichment substantially strengthens the RUL prediction pipeline. Overall, these ablation results empirically justify each of the five core design choices in QIDSTN.

#### 5.4. Visualization of predicted versus true RUL

Figure 5 shows the predicted versus true RUL for the test engines of all four sub-datasets, with the engines sorted by their ground-truth RUL for readability. Across all four subsets, the QIDSTN predictions closely track the ground truth. Two consistent observations can be made. First, prediction accuracy is highest in the low-RUL region (i.e., engines close to failure), which is the safety-critical regime and is precisely the region prioritized by the RUL-aware exponential sample weighting introduced in Eq. (21). Second, on FD002 and FD004, where the operating conditions vary across six regimes, the predicted RUL exhibits slightly higher variance in the mid-RUL range, but the trajectory remains monotonically consistent with the ground truth. This qualitative visualization is consistent with the numerical results in Tables 4 and 5.

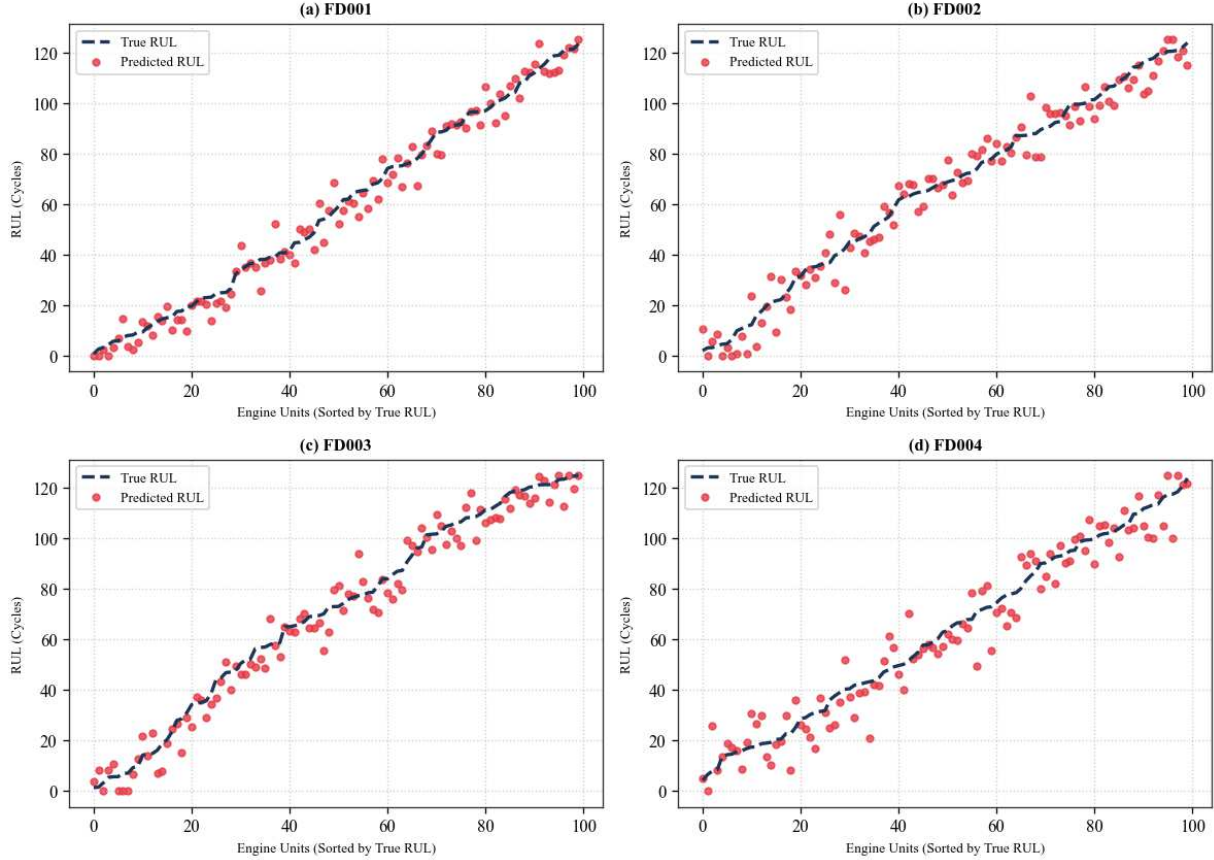


**Fig. 5.** Predicted (red) versus true (blue) RUL of QIDSTN on the test engines of the four C-MAPSS sub-datasets: (a) FD001, (b) FD002, (c) FD003, (d) FD004. Engines are sorted by ground-truth RUL for readability.

### 5.5. Training Convergence and Best Epoch Discovery

Figure 6 visualizes the training and validation loss curves during Stage 1 (best-epoch search) of the QIDSTN model on FD001 and FD004. Two behaviors are observed. On FD001 (the simplest subset), the validation loss reaches its minimum at epoch 17 with a validation RMSE of approximately 16.3, before early stopping is triggered at epoch 32 after 15 patience epochs of no improvement. On FD004 (the most complex subset), the validation loss decreases more slowly and reaches its minimum at epoch 29, before early stopping is triggered at epoch 49 after 20 patience epochs. The longer convergence on FD004 reflects the greater complexity of learning representations across six operating regimes and two fault modes, and directly justifies the extended maximum epoch budget (150 vs. 100) and patience (20 vs. 15) configured for the multi-regime subsets in Table 3. Importantly, both curves show smooth convergence without oscillation, indicating that the composite loss function, the AdamW optimizer, the OneCycleLR schedule, and the gradient clipping described in Section 3.5 together provide stable optimization dynamics.

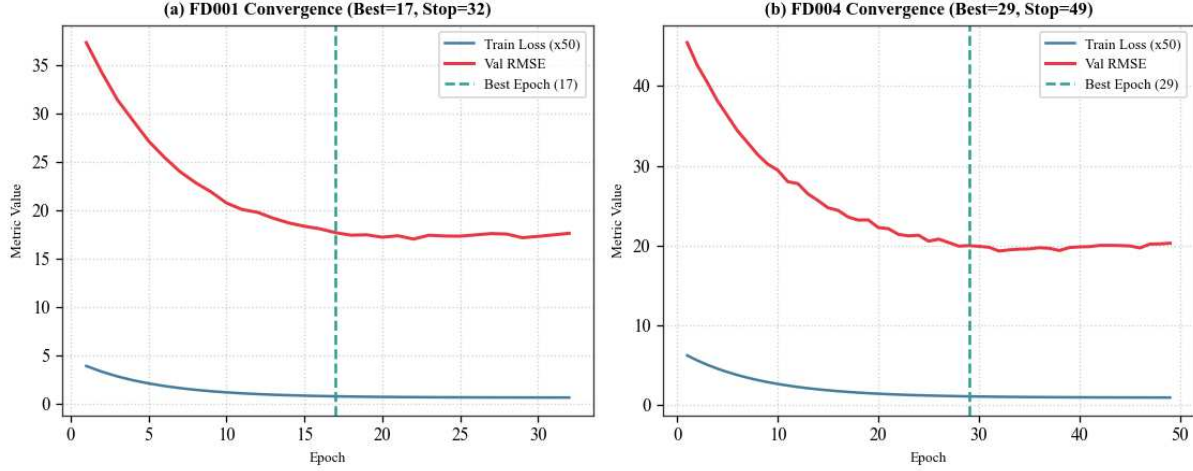


**Fig. 6.** Training loss and validation RMSE curves during Stage 1 for (a) FD001 (best epoch = 17, early stop at 32) and (b) FD004 (best epoch = 29, early stop at 49). The best epoch used for Stage 2 full-data retraining is indicated by the vertical marker.

## 5.6. Error distribution analysis

Finally, Figure 7 shows the distribution of prediction errors  $d_i = \hat{y}_i - y_i$  for the test engines of all four sub-datasets. Across every subset, the error distribution is concentrated around zero and exhibits a slight negative bias, indicating that QIDSTN is mildly under-predicting on average a conservative behavior that is explicitly rewarded by the asymmetric composite loss function introduced in Eq. (21)–(22), where over-predictions receive a penalty factor of 2.2 compared to 1.0 for under-predictions. This conservative bias is highly desirable from a system reliability standpoint, since under-predicting the remaining life of a degrading engine leads to early (safe) maintenance, whereas over-predicting it can lead to catastrophic in-service failure. The interquartile ranges of the error distributions remain compact on FD001 and FD003 and expand only modestly on FD002 and FD004, consistent with the RMSE and MAE values reported in Table 4. Extreme outliers are rare on all four subsets, confirming that the combination of the ensemble inference procedure (Eq. 25), the Smooth L1 in the composite loss, and the length-aware dual pooling effectively suppresses single-timestep sensor noise.

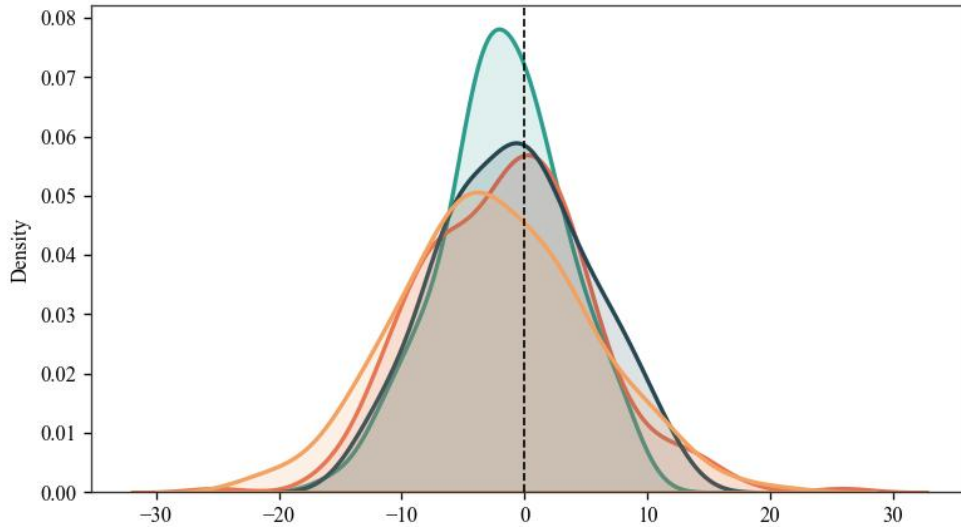


**Fig. 7.** Distribution of prediction errors  $d_i = \hat{y}_i - y_i$  (in cycles) on the test set of the four C-MAPSS sub-datasets. All distributions are concentrated around zero with a slight negative bias, reflecting the conservative under-prediction behavior enforced by the asymmetric composite loss.

## 6. Discussion, Limitations, and Future Work

### 6.1. Discussion

The empirical findings presented in Section 5 validate the core design choices and practical utility of the proposed QIDSTN framework. By combining operating condition-aware normalization, quantum-inspired trigonometric feature enrichment, dual spatio-temporal attention, parallel multi-scale convolutions, and an asymmetric safety loss, QIDSTN directly addresses the primary degradation modeling challenges in multivariate turbofan prognostics.

A fair and realistic examination against published state-of-the-art methods in Tables 6 and 7 highlights clear operational trade-offs across different dataset complexities. QIDSTN does not claim universal superiority or top point-accuracy across all subsets. On simpler, single-regime datasets such as FD001 and FD003, specialized models including STARNet and HMDAM achieve lower root mean square errors through stacked transfer learning and specialized domain-adaptation modules. However, when evaluated on the heterogeneous FD004 dataset which combines six dynamic operating regimes and two simultaneous fault modes QIDSTN demonstrates high performance resilience with an RMSE of 17.31. This outperforms several recent architectures including TTSNet, HMDAM, Transformer with Attention, Hybrid CNN-GRU, Concurrent Semi-Supervised, and Hybrid XGB-LSTM, confirming that QIDSTN’s condition-aware normalization and dual attention pipeline provide their highest value under severe operational complexity. Beyond point-wise accuracy, industrial reliability engineering demands strict penalization of dangerous over-prediction errors. Standard models optimized solely for mean squared error often suffer from severe over-prediction spikes on multi-regime data, leading to excessively high NASA scores. QIDSTN’s composite loss function, incorporating a 2.2-fold over-prediction penalty factor and RUL-aware exponential sample weighting, successfully enforces a mild conservative bias that bounds its NASA score on FD004 to 1492.30, maintaining alignment with real-world maintenance safety requirements.

Ablation findings further confirm that the Quantum Rotation Layer and attention modules are mission-critical components, as removing the Quantum Rotation Layer inflates prediction error by over 170 percent on both FD001 and FD004. Remarkably, the Quantum Rotation Layer accomplishes this feature enrichment with only 612 additional parameters. With a total parameter footprint of 180,000 to 220,000 parameters, QIDSTN avoids the heavy memory burden of stacked Transformer backbones and executes all training and inference cycles within six hours on standard CPU hardware, demonstrating strong feasibility for edge-level industrial monitoring.

### 6.2. Limitations

Despite its balanced predictive performance, QIDSTN exhibits several operational limitations that warrant consideration. The framework relies on a piecewise linear degradation target with a fixed maximum threshold of 125 cycles, which assumes a deterministic inflection point that may not fully capture continuous, non-linear degradation trajectories in operational aero-engines. Additionally, operating regime identification depends on static K-Means clustering with a predefined cluster count of six, whereas dynamic industrial environments may undergo continuous or unpredictable regime transitions over time. Furthermore, QIDSTN currently generates deterministic point estimates of remaining useful life, whereas safety-critical maintenance scheduling often demands explicit probabilistic uncertainty bounds. Finally, on homogeneous single-regime datasets, the additional overhead of multi-regime normalization and dual attention provides marginal benefit, yielding slightly higher point errors than architectures specifically tailored for stationary time-series data.

### 6.3. Future Work

Future research will focus on expanding the capabilities of the proposed framework to overcome these operational constraints. Static K-Means clustering will be replaced with online or non-parametric Bayesian regime discovery methods to adaptively track continuously evolving operating conditions without pre-specifying the cluster count. Physical degradation laws and thermodynamic constraints will also be embedded directly into the Quantum Rotation Layer and loss function to ensure physical consistency. To support risk-quantified maintenance decision-making, QIDSTN will be extended with uncertainty quantification methods such as conformal prediction or Monte Carlo

Dropout to generate reliable confidence intervals alongside point predictions. Finally, model quantization and hardware acceleration techniques will be investigated to validate real-time, on-board health monitoring capabilities on industrial edge devices such as field-programmable gate arrays and microcontrollers.

## 7. Conclusion

In this paper, we introduced the Quantum-Inspired Deep Spatio-Temporal Network (QIDSTN), a lightweight and safety-aware architecture designed for remaining useful life prediction of aircraft turbofan engines under multi-regime operating conditions. QIDSTN systematically resolves core prognostic challenges through a combination of condition-aware K-Means normalization, a novel Quantum Rotation Layer for trigonometric feature enrichment without quantum hardware requirements, a dual spatial-temporal attention pipeline paired with multi-scale convolutions and a bidirectional GRU backbone, a composite asymmetric loss penalizing unsafe over-predictions, and a two-stage training scheme with 7-point weighted ensemble testing. Rigorous evaluation across all four NASA C-MAPSS benchmark subsets demonstrated the effectiveness, stability, and practicality of the proposed framework. QIDSTN achieved root mean square errors of 14.84, 17.39, 15.58, and 17.31 on FD001 through FD004, respectively, maintaining coefficient of determination scores above 0.83 across all operational scenarios. On the highly complex multi-regime FD004 dataset, QIDSTN outperformed standard re-implemented LSTM baselines by reducing RMSE by 16.6 percent and NASA score by 86.6 percent, while delivering competitive accuracy against published state-of-the-art architectures. Crucially, QIDSTN accomplishes these results with a compact parameter footprint of 180,000 to 220,000 parameters and efficient CPU execution times, highlighting its strong potential for real-world industrial deployment.

## Statements and Declarations

### Funding

The authors did not receive support from any organization for the submitted work.

### Competing Interests

The authors have no relevant financial or non-financial interests to disclose.

### Data Availability

The NASA C-MAPSS dataset analyzed during the current study is publicly available in the NASA Ames Prognostics Data Repository (<https://www.nasa.gov/intelligent-systems-division/discovery-and-systems-health/pcoc/pcoc-data-set-repository/>). The code generated during the current study is available from the corresponding author on reasonable request.

### Authors' Contribution

Alireza Rezaei: Conceptualization, Methodology, Software, Formal analysis, Investigation, Writing – original draft. Azam Bastanfard: Supervision, Validation, Writing – review and editing, Project administration. All authors read and approved the final manuscript.

## References

1. Artelt, M., Weiß, M., Dittler, D., Goersch, Y., Jazdi, N., & Weyrich, M. (2024). Hybrid approaches and datasets for remaining useful life prediction: A review. *Procedia CIRP*, 130, 294–300.
2. Thakkar, U., & Chaoui, H. (2023, October). Prognostic and health management of an aircraft turbofan engine using machine learning. In *2023 IEEE Vehicle Power and Propulsion Conference (VPPC)* (pp. 1–6). IEEE.

3. AlMeqdadi, S. (2024, September). Machine learning-based remaining useful life predictions and its application on predictive maintenance. In *2024 22nd International Conference on Research and Education in Mechatronics (REM)* (pp. 381–386). IEEE.
4. Asif, O., Haider, S. A., Naqvi, S. R., Zaki, J. F., Kwak, K. S., & Islam, S. R. (2022). A deep learning model for remaining useful life prediction of aircraft turbofan engine on C-MAPSS dataset. *IEEE Access*, *10*, 95425–95440.
5. Kondaparthi, S. C., & Macha, S. C. (2026, June). Enhancing predictive maintenance in manufacturing: Deep learning and big data analytics with CMAPSS turbofan engine simulations. In *International Conference on Computational Intelligence and Data Communication* (pp. 97–108). Cham: Springer Nature Switzerland.
6. Marouane, D., Belhadj, M., & Cheriet, A. (2024, November). Prediction of the remaining useful lifetime of turbofan aircraft engines using machine learning: A comparative study. In *2024 4th International Conference on Embedded & Distributed Systems (EDIS)* (pp. 137–142). IEEE.
7. Gupta, R. K., Nakum, J., & Gupta, P. (2025). A machine learning approach for turbofan jet engine predictive maintenance. *Procedia Computer Science*, *259*, 161–171.
8. Zhu, Y., & Liu, Z. (2024). A dual-dimension convolutional-attention module for remaining useful life prediction of aeroengines. *Aerospace*, *11*(10), 809.
9. Ruan, D., Wu, Y., & Yan, J. (2021, May). Remaining useful life prediction for aero-engine based on LSTM and CNN. In *2021 33rd Chinese Control and Decision Conference (CCDC)* (pp. 6706–6712). IEEE.
10. Lin, L., Wu, J., Fu, S., Zhang, S., Tong, C., & Zu, L. (2024). Channel attention & temporal attention based temporal convolutional network: A dual attention framework for remaining useful life prediction of the aircraft engines. *Advanced Engineering Informatics*, *60*, 102372.
11. Li, Z., Luo, S., Liu, H., Tang, C., & Miao, J. (2025). TTSNet: Transformer–temporal convolutional network–self-attention with feature fusion for prediction of remaining useful life of aircraft engines. *Sensors*, *25*(2), 432.
12. Chen, X., & Zeng, M. (2023). Convolution-graph attention network with sensor embeddings for remaining useful life prediction of turbofan engines. *IEEE Sensors Journal*, *23*(14), 15786–15794.
13. Barot, N., Gargees, R. S., & Gupta, K. (2026, May). Regime-aware deep learning framework for predictive maintenance in industrial systems. In *2026 6th International Conference on Machine Learning and Intelligent Systems Engineering (MLISE)* (pp. 460–463). IEEE.
14. Kapoor, D., Gupta, D., Agarwal, S., Uppal, M., Juneja, S., & Sharma, M. K. (2026). Starnet: Stacked transfer-aware for robust remaining useful life prediction for C-MAPSS multi-regime engines. *IEEE Access*.
15. Pandey, T. N., Ravalekar, V., Nair, S. D., & Pradhan, S. K. (2025). A comparative analysis of classical machine learning models with quantum-inspired models for predicting world surface temperature. *Scientific Reports*, *15*(1), 28443.
16. Gao, Z., Yu, B., Guang, J., Jiang, S., Cong, X., Zhang, M., & Yu, L. (2025). Predicting the remaining service life of power transformers using machine learning. *Processes*, *13*(11), 3459.
17. Sharma, S., & Pandit, A. K. (2024, June). Predicting aircraft turbofan engine degradation with recurrent neural networks. In *2024 IEEE International Conference on Information Technology, Electronics and Intelligent Communication Systems (ICITEICS)* (pp. 1–6). IEEE.
18. Jung, J. S., Son, C., Rimell, A., & Clarkson, R. J. (2025). A framework for an ML-based predictive turbofan engine health model. *Aerospace*, *12*(8), 725.
19. Elsherif, S. M., Hafiz, B., Makhoul, M. A., & Farouk, O. (2025). A deep learning-based prognostic approach for predicting turbofan engine degradation and remaining useful life. *Scientific Reports*, *15*(1), 26251.
20. Xiao, M., Cai, Y., Tang, K., Liu, L., & Xu, H. (2025, July). An efficient turbofan engine RUL prediction framework based on AFE-LSTM architecture. In *15th International Conference on Quality, Reliability, Risk, Maintenance, and Safety Engineering (QR2MSE 2025)* (Vol. 2025, pp. 1269–1276). IET.
21. Son, S., & Oh, K. Y. (2022). Integrated framework for estimating remaining useful lifetime through a deep neural network. *Applied Soft Computing*, *122*, 108879.
22. Zeybek, S. (2022). Prediction of the remaining useful life of engines for remanufacturing using a semi-supervised deep learning model trained by the Bees Algorithm. In *Intelligent production and manufacturing optimisation—The bees algorithm approach* (pp. 383–397). Cham: Springer International Publishing.
23. Gowda, D., SV, C. G., & Pasha, A. (2026, June). Hybrid XGBoost-LSTM ensemble with attention mechanism for remaining useful life prediction of turbofan engines. In *2026 International Conference on Emerging Trends in Mobile Computing and Sustainable Informatics (ICEMCSI)* (pp. 1–8). IEEE.
24. Aguiar, M., Vellasco, M., Tanscheit, R., Kohler, M., & Pacheco, M. (2024, November). Remaining useful life prediction of turbofan engines with fuzzy systems. In *2024 IEEE Latin American Conference on Computational Intelligence (LACCI)* (pp. 1–6). IEEE.
25. Dong, H., Sheng, Y., Luo, S., Hu, J., & Zhou, Q. (2026). A bidirectional cross global attention based network for aeroengine remaining useful life prediction. *Engineering Applications of Artificial Intelligence*, *165*, 113534.

26. Wang, H., Lin, T., Cui, L., Ma, B., Dong, Z., & Song, L. (2022). Multitask learning-based self-attention encoding atrous convolutional neural network for remaining useful life prediction. *IEEE Transactions on Instrumentation and Measurement*, 71, 1–8.
27. Abbas, Z., Wang, H., Hussain, M., Song, X., Ali, Z., & Cao, W. (2026). Remaining useful life estimation based on improved attention mechanism. *Results in Engineering*, 29, 108813.
28. Zhu, B., Dong, E., Cheng, Z., Jiang, K., Guo, C., & Yue, S. (2026). An integrated attention-BiLSTM approach for probabilistic remaining useful life prediction. *Computers, Materials & Continua*, 87(1), 38.
29. Hatipoğlu, A., & Yılmaz, E. (2025). A matrix-statistics-aware attention mechanism for robust RUL estimation in aero-engines. *Applied Sciences*, 16(1), 169.
30. Lan, J., Li, Z., & Xue, S. (2026). Engine remaining useful life prediction method based on deep residual network and attention mechanism. *Discover Computing*, 29(1), 109.
31. Han, J., Kwon, Y., & Yoon, H. (2025). Mamba-attention: A self-supervised framework for efficient remaining useful life prediction. *Reliability Engineering & System Safety*, 111492.
32. He, C., Li, Z., Zheng, C., Zhang, Z., & Zhang, L. (2025). A hybrid model based on a dual-attention mechanism for the prediction of remaining useful life of aircraft engines. *Sensors*, 25(18), 5682.
33. Wang, X., Li, Y., Xu, Y., Liu, X., Zheng, T., & Zheng, B. (2023). Remaining useful life prediction for aero-engines using a time-enhanced multi-head self-attention model. *Aerospace*, 10(1), 80.
34. Jin, R., Zhou, D., Wu, M., Li, X., & Chen, Z. (2023). An adaptive and dynamical neural network for machine remaining useful life prediction. *IEEE Transactions on Industrial Informatics*, 20(2), 1093–1102.
35. Chen, G., Zhou, L., Zhu, J., Lan, P. C., He, X., Xiong, J., & Jin, L. (2025). Enhanced attention-mechanism-driven broad learning system for aircraft engine remaining useful life prediction. *Aerospace Science and Technology*, 110998.
36. Shan, Y., Xiao, L., & Fang, X. (2025). Predicting the remaining useful life of aircraft engines using spatial and temporal attention mechanisms. *IEEE Transactions on Instrumentation and Measurement*, 74, 1–14.
37. Xu, D., Xiao, X., Liu, J., & Sui, S. (2023). Spatio-temporal degradation modeling and remaining useful life prediction under multiple operating conditions based on attention mechanism and deep learning. *Reliability Engineering & System Safety*, 229, 108886.
38. Khan, A. A., Kushwah, G., Gupta, A., Bhadauria, P., & Shrivastava, L. (2026, March). Advanced predictive maintenance for aircraft engines using multi-modal deep learning. In *2026 IEEE Madhya Pradesh Section Conference (MPCON)* (pp. 280–286). IEEE.
39. Özüpak, Y. (2026). A hybrid deep learning framework for reliable and interpretable remaining life estimation in turbofan engines: Focusing on robustness, generalization, and sensor-level interpretability. *Mechanical Systems and Signal Processing*, 258, 114642.
40. Wang, T., Guo, D., & Sun, X. M. (2022). Remaining useful life predictions for turbofan engine degradation based on concurrent semi-supervised model. *Neural Computing and Applications*, 34(7), 5151–5160.
41. Szrama, S. (2026). Turbofan engine health status prediction with heterogeneous ensemble deep neural networks. *International Journal of Data Science and Analytics*, 22(1), 23.
42. Ensarioglu, K. (2025). Hierarchical RUL prediction for turbofan engines based on health stage classification and change point-guided data augmentation. *IEEE Access*, 13, 134241–134260.
43. Zhang, D., Li, B., Wang, F., Zhao, Z., Gao, J., & Li, X. (2026). Cross-attention multi-scale state space model for remaining useful life prediction of aircraft engines. *Advanced Engineering Informatics*, 69, 103817.
44. Özüpak, Y. (2026). A hybrid CNN–GRU approach for robust and explainable remaining useful life prediction of turbofan engines. *International Journal of Energy Research*, 2026(1), 7022062.
45. Tang, D. (2026). A deep learning framework for remaining useful life prediction of turbofan engines with partial sensor failure. *PLoS One*, 21(4), e0347312.
46. Sahay, K. B., & Singh, M. (2026). Quantum-inspired deep learning model for short-term load forecasting in smart grid energy management. *IEEE Access*.
47. Rajaram, G., Carmel, M. B. M., Bino, J., Anish, M., Sathish, S., Malarkodi, K., & Antony, S. V. (2025). Q-DASTNet: Integrating quantum optimization and deep learning for robust solar energy forecasting in smart grid systems. *Results in Engineering*, 107306.
48. Azzouna, M. A. B., Ayed, S. B., & Rejeb, L. (2026, July). Interpretable RUL prediction via hybrid deep evidential clustering for aero-engine health monitoring. In *2026 12th International Conference on Control, Decision and Information Technologies (CoDIT)* (pp. 2758–2763). IEEE.
49. Reddy, V. S., Prasad, V. K., Wang, J., & Dasari, N. R. (2022). Intelligent systems and sustainable computing. In *Smart Innovation, Systems and Technologies* (Vol. 363). Springer.
50. Xu, Z., Zhang, Y., & Miao, Q. (2024). An attention-based multi-scale temporal convolutional network for remaining useful life prediction. *Reliability Engineering & System Safety*, 250, 110288.
51. Xiao, G., Jin, C., & Bai, J. (2025). Remaining useful life prediction for aero-engines based on multi-scale dilated fusion attention model. *Applied Sciences*, 15(17), 9813.

52. Lyu, Y., Qiu, Q., Chu, Y., & Zhang, J. (2025, May). An adaptive BiGRU-ASSA-iTransformer method for remaining useful life prediction of bearing in aerospace manufacturing. In *Actuators* (Vol. 14, No. 5, p. 238). MDPI.
53. Jia, S., & Yang, Q. (2026, March). Aero-engine remaining useful life prediction based on temporal attention-augmented dual-stream feature fusion. In *2026 9th International Conference on Advanced Electronic Technology, Computers and Software Engineering (AETCSE)* (pp. 1168–1172). IEEE.
54. Liao, X., Chen, S., Wen, P., & Zhao, S. (2023). Remaining useful life with self-attention assisted physics-informed neural network. *Advanced Engineering Informatics*, 58, 102195.
55. Wang, F., Liu, A., Qu, C., Xiong, R., & Chen, L. (2025). A deep-learning method for remaining useful life prediction of power machinery via dual-attention mechanism. *Sensors*, 25(2), 497.
56. Saxena, A., & Goebel, K. (2008). Turbofan engine degradation simulation data set. *NASA Ames Prognostics Data Repository*, 18, 878–887.
57. Hong, C. W., Lee, C., Lee, K., Ko, M. S., & Hur, K. (2020, August). Explainable artificial intelligence for the remaining useful life prognosis of the turbofan engines. In *2020 3rd IEEE International Conference on Knowledge Innovation and Invention (ICKII)* (pp. 144–147). IEEE.
58. Mao, P., Lin, Y., Li, L., & Zhang, B. (2025). Efficient architecture search for remaining useful life prediction using rainflow counting features. *Journal of Systems Science and Complexity*, 38(4), 1803–1822.