


# PD-STGNN: A PHYSICS-INFORMED DYNAMIC SPATIO-TEMPORAL GRAPH NEURAL NETWORK FOR AERO-ENGINE THRUST PREDICTION

Chenchen Song[1], Hao Shi[1], Zilang Huang[1] & Hong Xiao[1]

[1]School of Power and Energy, Northwestern Polytechnical University, 710072, Xi'an, Shaanxi, PR China

## Abstract

Accurate thrust prediction is essential for aero-engine control optimization, flight safety, and health management. Although digital twin technology offers an effective route for virtual thrust sensing, conventional data-driven models often lack physical interpretability, limiting their reliability in safety-critical aerospace applications. To address this issue, this study proposes a Physics-informed Dynamic Spatio-Temporal Graph Neural Network (PD-STGNN) for aero-engine thrust prediction. The proposed framework maps multi-sensor time-series data onto a component-level graph that follows the actual gas-path topology, thereby constructing a physically meaningful spatial representation of engine components. A Multi-Head Dynamic Graph Attention (MHDGA) mechanism is further introduced to infer time-varying aerodynamic and thermodynamic couplings among components, while static physical priors constrain the learned graph toward fluid-mechanical consistency. Experiments on 20 hours of real ground-test data from 10 twin-spool turbofan engines show that PD-STGNN delivers accurate and robust thrust prediction across steady-state and transient operating regimes. In denormalized physical units, the maximum MAPE and RMSPE are bounded by 2.04% and 5.41%, respectively, outperforming mainstream deep learning baselines.

**Keywords:** Aero-engine; Thrust prediction; Graph neural network; Physical knowledge; Digital twin modeling

## 1. Introduction

Thrust prediction is paramount for aero-engine performance assessment and is intrinsically linked to Engine Health Management (EHM). Accurate thrust estimation provides critical support for monitoring performance degradation, evaluating propulsion efficiency, and ensuring long-term operational reliability[1, 2]. However, achieving real-time thrust monitoring under practical flight conditions presents significant challenges. Due to constraints on onboard sensor deployment and harsh operational environments, direct thrust measurement is often impractical, forcing reliance on limited indirect thermodynamic parameters that typically fail to fully capture the engine's transient dynamic characteristics[3, 4]. To address this challenge, Digital Twin (DT) technology has been introduced into the aero-engine performance assessment process[5, 6]. By constructing a high-fidelity digital model that continuously assimilates real-time flight data, DT technology functions as a 'virtual sensor.' Leveraging DT technology, researchers can infer high-accuracy thrust outputs from limited onboard parameters, thereby providing a comprehensive, real-time method for performance validation and state awareness in data-constrained flight environments[7, 8]. Therefore, DT technology plays a significant role in real-time engine performance assessment.

The core element of constructing an aero-engine digital twin system lies in establishing a high-fidelity mathematical model that accurately represents the aerothermodynamic behavior of the physical entity. As a tool endowed with robust non-linear function approximation capabilities, neural networks provide an effective technical approach for the high-fidelity modeling of aero-engines[9, 10]. In the

nascent stage of digital twin model development, researchers predominantly adopted data-driven methods to construct the input-output mapping models of the engine[11, 12]. Kim et al. [13]proposed a data-driven modeling approach to predict the performance of gas turbines for aircraft engines during transient operations, with output parameters such as engine net thrust and low-pressure shaft rotating speed. Andrianantara et al. [14] developed two interconnected neural networks using the MLP algorithm to establish a mapping relationship between input parameters and key engine performance metrics. Training results on simulated datasets demonstrated the model's capability to accurately predict engine fan speed, corrected thrust, and fuel flow rate.

As the demand for model accuracy and reliability continues to grow, research on digital twins for aircraft engines has gradually shifted from purely data-driven methods toward hybrid approaches that integrate both data and physical knowledge [15, 16]. Xiao et al. [17]proposed Engineformer, an aero-engine digital twin modeling architecture that integrates prior physical knowledge with the Transformer framework. The model achieved state-of-the-art performance in both exhaust gas temperature (EGT) and remaining useful life (RUL) prediction tasks, robustly validating its reliability in characterizing aero-engine performance and predicting degradation. Song et al. [18]proposed a physics-informed two-stage modeling approach for aero-engine performance prediction. This method encodes the topological structure of engine components into a static adjacency matrix, which alongside a dynamic adjacency matrix describing the inter-component coupling strengths, simultaneously guides the network training. Evaluated on real-world datasets, the model achieved minimum MAPE of 1.23%, 1.46%, and 2.25% in predicting thrust ($F$), compressor discharge pressure ($P_3$), and EGT, respectively. Li et al. [19] proposed a sensor health diagnosis framework that integrates physics-based models with deep learning, combining physics-guided neural network (PGNN) and CNN to achieve highly accurate fault detection under dynamic operating conditions. Experimental results demonstrate that the method significantly outperforms purely data-driven approaches, with diagnostic accuracy improved to 95.9%.

Based on the above analysis, compared with traditional data-driven methods, digital twin models that incorporate physical priors can more effectively capture the spatio-temporal correlations during engine operation, thereby achieving higher accuracy and reliability in performance prediction and fault diagnosis tasks[20, 21, 22, 23]. This study proposes a Physics-informed Dynamic Spatio-Temporal Graph Neural Network (PD-STGNN) framework for aero-engine thrust prediction. By explicitly incorporating the engine's internal gas-path flow and thermodynamic coupling laws into the network architecture, the proposed framework captures both complex spatio-temporal coupling characteristics and nonlinear transient dynamics. This approach provides a high-fidelity, interpretable solution for engine performance prediction. The main contributions of this work are summarized as follows:

- A physics-informed spatio-temporal modeling framework is proposed, which maps multi-sensor monitoring data into a component-level graph structure conforming to the actual engine gas-path topology, thereby establishing a unified spatio-temporal representation.
- A dynamic graph learning mechanism is proposed, which synergistically integrates static physical structural priors with adaptive dynamic graph updates. This mechanism effectively captures the time-varying coupling relationships among engine components under fluctuating operating conditions, thereby enabling a precise characterization of the engine's nonlinear internal dynamics.
- The proposed method is validated on real aero-engine operational datasets, and the experimental results demonstrated its effectiveness and robustness in modeling engine dynamics and predicting thrust, highlighting its potential for practical engineering applications.

## 2. Methodology

### 2.1 Overview Framework

To accurately predict the thrust performance of aero-engines under complex operating conditions and fully exploit their internal aerodynamic and thermodynamic coupling laws, this paper proposes a Physics-informed Dynamic Spatio-Temporal Graph Neural Network (PD-STGNN) framework. The

overall workflow of the PD-STGNN is illustrated in Fig. 1. Multi-sensor monitoring data are first fed into a Component-wise Linear Transformation (CLT) module. Strictly following the actual gas path flow, this module maps the multi-dimensional heterogeneous sensor time-series signals onto seven key physical component nodes, thereby constructing a foundational spatial representation with explicit topological significance.

Subsequently, the data flow into the core spatiotemporal modeling stage, which consists of multi-layer configurable spatiotemporal feature extractors. In this stage, the model not only deeply mines the state evolution trends of each independent component in the temporal dimension but also innovatively introduces a dynamic graph learning mechanism with physics-informed constraints in the spatial dimension. This mechanism utilizes the prior matrix—which represents the actual gas paths and coaxial mechanical connections—as a learnable bias. While guiding the network to strictly adhere to the objective laws of fluid mechanics, it adaptively infers the implicit dynamic directed coupling paths among components, and subsequently accomplishes the highly efficient aggregation of deep spatial information via multi-hop graph convolutions.

Finally, these multi-scale, hierarchical spatiotemporal features extracted by the stacked network are globally concatenated and flattened in the Readout Layer. Empowered by a Thrust Regression Layer (TRL) equipped with regularization strategies, the framework successfully achieves an end-to-end mapping from multivariate condition monitoring data to precise scalar thrust values.

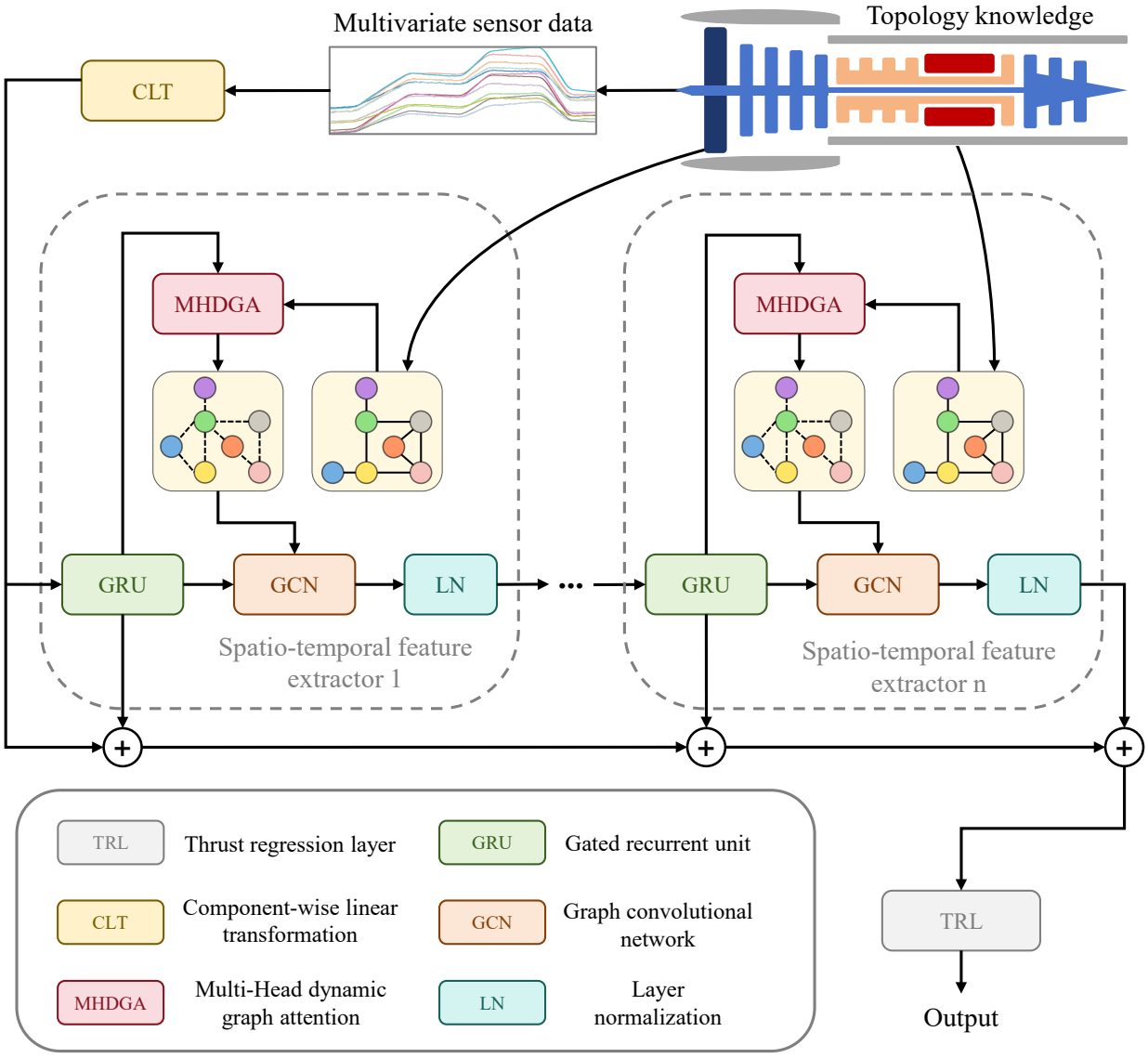



Figure 1 – The overview framework of the PD-STGNN.

## 2.2 Physics-Guided Graph Structure Construction for Aero-Engines

In the condition monitoring of aero-engines, raw multi-sensor time-series data typically exist in a flattened matrix format, which inherently struggles to intuitively represent the complex aerodynamic and thermodynamic topological relationships within the engine. To overcome the limitations of such purely data-driven representations, this section proposes a physics-guided graph structure construction method aimed at transforming heterogeneous sensor signals into graph nodes with explicit physical significance. For a typical turbofan engine, its macroscopic physical structure can be primarily divided into core components, including the fan, low-pressure compressor (LPC), high-pressure compressor (HPC), combustor, high-pressure turbine (HPT), low-pressure turbine (LPT), and nozzle. Among these, the fan and LPC constitute the low-pressure rotor system, while the HPC and HPT form the high-pressure rotor system. These physical components are not only sequentially arranged in space but also deeply coupled through the internal gas path flow and coaxial mechanical systems, working synergistically to complete the thermodynamic cycle of air compression, combustion, and thrust generation.

Based on the aforementioned physical structural priors of the engine, this paper strictly partitions the sensor measurement variables into feature subsets corresponding to the seven specific components. Considering the disparity in the number of sensors allocated to different components, and to satisfy the alignment requirements of graph neural networks regarding node feature dimensions, this study utilizes a Component-wise linear transformation (CLT) module to perform independent linear mapping on the subsets of each component. The calculation process is as follows:

$$X^{'}_{i(1:T)} = \mathrm{FC}\left(X_{i(1:T)}\right) \tag{1}$$

where $X^{'}_{i(1:T)} \in R^{T\times C}$ and $X_{i(1:T)} \in R^{T\times n_i}$, $C$ denotes the unified projection dimension for all components. By concatenating the projected features of all engine components, the graph-structured data that integrates the engine's structural information ($X_{(1:T)} \in R^{C\times T\times N}$) can be represented as follows:

$$X_{(1:T)} = \mathrm{concat}\left(X^{'}_{1(1:T)}, X^{'}_{2(1:T)}, ..., X^{'}_{N(1:T)}\right) \tag{2}$$

Following the structural transformation of node representations, establishing the spatial correlations among these nodes serves as a critical prerequisite for deep spatiotemporal graph learning. Distinct from conventional graph neural networks that rely entirely on data-driven approaches or randomly initialized adjacency matrices, this study defines the seven core physical components of the engine as graph nodes. The aerodynamic gas path conduction and coaxial mechanical connections between these components are explicitly encoded as topological edges, thereby generating a deterministic physical prior adjacency matrix. As shown in Figure 2, where a value of 1 indicates that two components are directly connected, and 0 indicates that they are not directly connected. In the subsequent deep feature interactions, this static structural graph acts as a rigorous physical constraint and structural bias. It fundamentally ensures that the implicit feature propagation and evolution within the network strictly comply with the objective physical laws of fluid mechanics and structural dynamics.

## 2.3 Temporal Dependency Module

Following the graph data construction based on component topology, synergistically modeling the dynamic evolution trends of individual physical nodes in the temporal dimension becomes a critical prerequisite for accurate engine thrust prediction. Since the aerodynamic and thermodynamic states of an aero-engine exhibit strong temporal correlations and time-lag effects during actual operation, transient features from a single time step struggle to comprehensively reflect the underlying rules of degradation and performance evolution. Therefore, this section introduces the temporal dependency module, which aims to deeply mine the dynamic variation trends of each core component node within a continuous time window. This module employs a standard Gated Recurrent Unit (GRU) as the core temporal modeling network to establish nonlinear dependencies along the time axis. By accurately capturing the potential evolution trajectories of each component state over time, the module provides node hidden state representations rich in historical contextual information for subsequent spatial

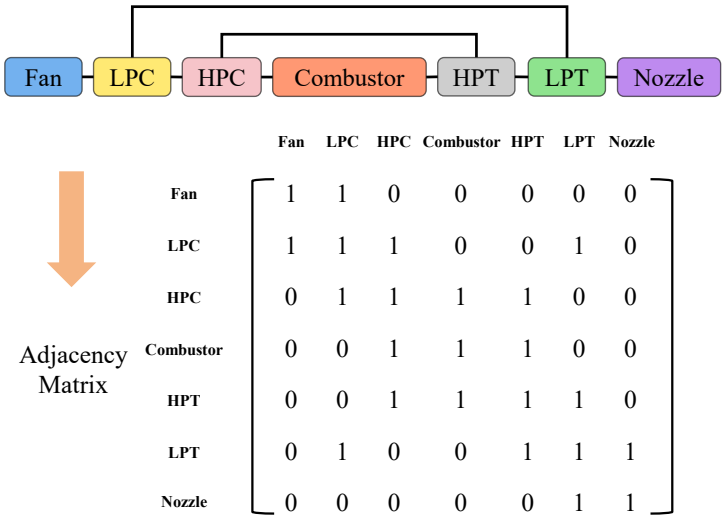



Figure 2 – Static representation of engine structural information.

feature aggregation. As illustrated in the Figure 3, the computation process of the GRU can be expressed as follows:

$$z_t = \sigma\left(W_z[h_{t-1}, x_t]\right) \tag{3}$$

$$r_t = \sigma\left(W_r[h_{t-1}, x_t]\right) \tag{4}$$

$$\tilde{h}_t = \tanh\left(W[r_t \odot h_{t-1}, x_t]\right) \tag{5}$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t \tag{6}$$

where $z_t$ is the update gate, $r_t$ is the reset gate, $\tilde{h}_t$ is the candidate hidden state, and $h_t$ denotes the final hidden state at time step $t$, $\odot$ represents element-wise multiplication, and $\sigma(\cdot)$ denotes the sigmoid activation function.

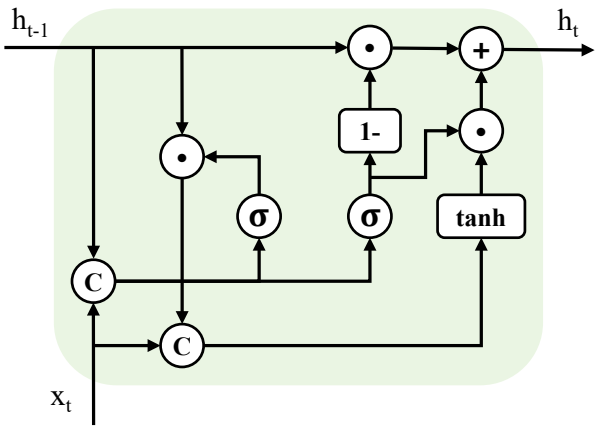



Figure 3 – Architecture of the GRU.

## 2.4 Construction of the Dynamic Engine Graph Structure

Notably, although the macroscopic physical structure of the aero-engine remains fixed during the test process, the internal interaction mechanisms among components (e.g., energy transfer and thermal coupling) continuously evolve with changing operating conditions, exhibiting pronounced time-varying and non-stationary characteristics. Consequently, merely employing a static prior graph is often insufficient to comprehensively characterize the full picture of the complex dynamic evolution during

engine tests. To address this limitation, this study further introduces a dynamic graph learning mechanism to capture the implicit dynamic coupling relationships among components in real time during engine operation. As illustrated in Figure 4, driven by varying operating conditions, the adjacency matrix describing the spatiotemporal interactions among components adapts and evolves over time, thereby more accurately capturing the nonlinear trajectories of inter-component interactions. This mechanism significantly enhances the model's spatial representation capability regarding the time-varying coupling characteristics of the engine, laying a solid structural foundation for high-precision engine performance prediction.

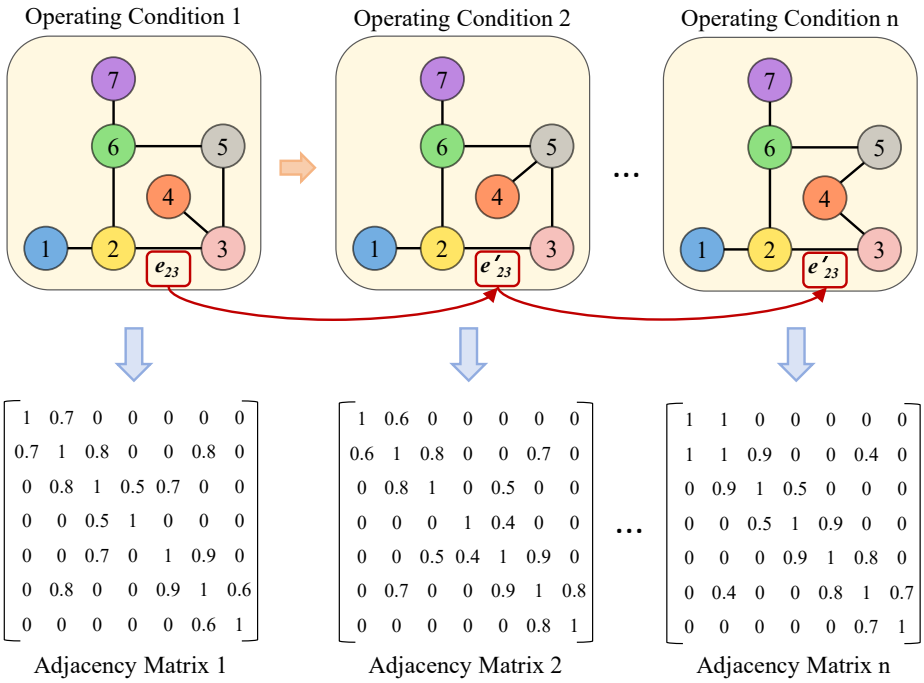



Figure 4 – Dynamic Representation of Component Coupling Relationships.

To accurately characterize the non-stationary dynamic coupling features of the aero-engine during the test process in the spatiotemporal dimension, as shown in Figure 5, this section proposes a Multi-Head Dynamic Graph Attention (MHDGA) module. This module aims to overcome the inherent limitation of traditional static adjacency matrices in adaptively adjusting spatial correlation strengths. By deeply integrating multi-head subspace decoupling, bidirectional temporal context encoding, and a physics-informed prior injection mechanism, this module leverages an adaptively evolving dynamic adjacency matrix to precisely characterize the time-varying coupling strengths among the core engine components under variable operating conditions.

Suppose the input to the temporal dependency module is $X \in R^{B\times T\times N\times D}$. After merging the batch dimension and the node dimension via dimensional reshaping, the tensor is fed into a Bidirectional Gated Recurrent Unit (Bi-GRU) to capture temporal dependencies. By extracting the network's output at the final time step, the node state representation matrix $H$, which fuses the deep temporal evolution history, is obtained. The corresponding calculation process can be expressed as:

$$H = \text{Bi-GRU}(X_{\text{reshaped}}) \in R^{B\times N\times D_h} \tag{7}$$

where $B$ denotes the batch size, $N$ is the number of core physical component nodes, and $D_h$ is the hidden layer feature dimension.

After obtaining the temporal context representation $H$, this module introduces a multi-head attention mechanism to decouple and mine the potential directed aerodynamic and thermodynamic coupling paths among components within different feature subspaces. For each attention head $m$ (out of $M$ heads in total), the network maps the node representation $H$ into a Query matrix and a Key matrix

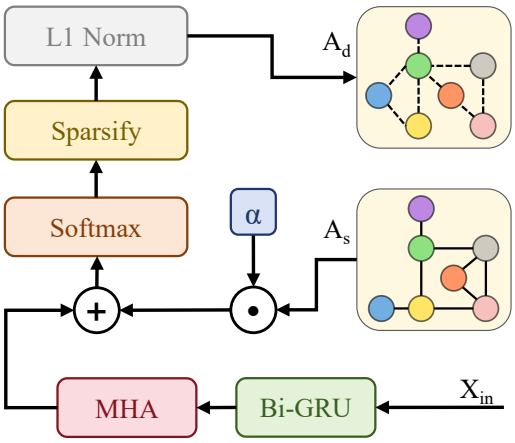



Figure 5 – Architecture of the MHDGA.

through independent linear transformation matrices $W_q^{(m)} \in R^{D_h \times d_k}$ and $W_k^{(m)} \in R^{D_h \times d_k}$, respectively, where $d_k = D_h/M$ represents the feature dimension of each subspace. By calculating the matrix inner product between the query and key vectors, the intrinsic asymmetric spatial correlation scores among components are derived. The spatial correlation scores calculated from all attention heads are then fused via arithmetic averaging to construct the generalized data-driven dynamic correlation matrix $\bar{S}$:

$$S_m = \frac{(HW_q^{(m)})(HW_k^{(m)})^T}{\sqrt{d_k}} \in R^{B \times N \times N} \tag{8}$$

$$\bar{S} = \frac{1}{M} \sum_{m=1}^{M} S_m \in R^{B \times N \times N} \tag{9}$$

The scaling factor $\sqrt{d_k}$ in the denominator is utilized to alleviate the issue of excessively large inner product values caused by high dimensionality, ensuring the smooth flow of gradients during back-propagation.

However, a purely data-driven correlation matrix is highly susceptible to introducing spurious correlation noise. To address this, the module innovatively proposes a physics-informed soft injection mechanism. This mechanism softly injects the static physical prior adjacency matrix $A_{\text{static}}$ of the engine into the data-driven correlation matrix $\bar{S}$ in the form of a bias term. Meanwhile, a learnable confidence coefficient $\alpha \in R$ is introduced, endowing the network with the flexibility to autonomously adjust its degree of trust in the physical priors. The combined correlation logits matrix $L$ after integration can be expressed as:

$$L = \bar{S} + \alpha \cdot A_{\text{static}} \tag{10}$$

Based on this, to ensure the global normalization of the graph attention weights and to filter out weak interference noise, the network performs a Softmax normalization operation on the matrix $L$ along the column dimension. Subsequently, a dynamic sparsification filtering operator is introduced to forcibly zero out the weights of pseudo-correlated edges whose contributions are below a predefined sparsity threshold $\tau$. To prevent the sparsification operation from disrupting the standard probability distribution of the weights, an L1 norm normalization is ultimately applied to the processed matrix. This yields the adaptive dynamic adjacency matrix $A_{\text{dynamic}}$, which possesses both physical interpretability and dynamic adaptability:

$$\tilde{A} = \text{Softmax}(L) \tag{11}$$

$$\hat{A}_{ij} = \begin{cases} \tilde{A}_{ij}, & \text{if } \tilde{A}_{ij} \geq \tau \\ 0, & \text{if } \tilde{A}_{ij} < \tau \end{cases} \tag{12}$$

$$A_{\text{dynamic}} = \text{F.normalize}(\hat{A}, p = 1, \text{dim} = -1) \tag{13}$$

### 2.5 Graph Convolution Module

Graph neural networks extract spatial dependencies through message passing between nodes. In this study, to accurately capture the physical data-flow relationships and spatial interactions between different engine components, a standard Graph Convolutional Network (GCN) module is employed. The GCN operates directly on the graph structure, updating the state of each component node by aggregating feature information from its connected neighbors.

In each graph convolution layer, the feature representations of engine component nodes are updated by linearly combining the features of their immediate physical neighbors, followed by a non-linear transformation. Given the input feature matrix $H^{(l)}$ at the $l$-th layer and the adaptively learned dynamic adjacency matrix $A_{\text{dynamic}}$, the layer-wise message-passing process of the graph convolution can be defined as:

$$H^{(l+1)} = \sigma\left(A_{\text{dynamic}} H^{(l)} W^{(l)}\right) \tag{14}$$

where $H^{(l)}$ and $H^{(l+1)}$ denote the node representations at layer $l$ and $l+1$, respectively, with $H^{(0)} = H_{\text{in}}$ representing the initial input features. $W^{(l)}$ is the learnable weight matrix for the $l$-th layer, and $\sigma(\cdot)$ denotes a non-linear activation function.

By stacking multiple graph convolution layers, the model can iteratively propagate information across the graph, enabling each component node to learn broader topological features beyond its first-order neighborhood. Through this hierarchical aggregation, the network yields highly discriminative spatiotemporal node representations $H_{\text{out}}$, which comprehensively integrate both the local operating states and the global topological dynamics of the aero-engine.

## 3. Experimental setup

### 3.1 Datasets description

To validate the effectiveness of the proposed network architecture, a total of 20 hours of ground test data were collected from 10 twin-spool turbofan engines. The dataset covers a variety of typical operating conditions, including engine start-up, idle, acceleration, deceleration, steady-state regimes, and multiple transient processes. For each engine, comprehensive sensor measurements of environmental parameters, control parameters, and state parameters were recorded, and all sensor signals were down-sampled to 1 Hz. All engine tests were conducted on a ground test bench. To mitigate the impact of randomness introduced by a single data split, we performed four independent random splits on the dataset comprising the 10 twin-spool turbofan engines. In each split, 8 engines were allocated for training, 1 for validation, and 1 for testing, thereby constructing four independent sets of training, validation, and test subsets.

### 3.2 Sensor Parameter Selection

To construct a high-fidelity and physically meaningful thrust prediction model, the scientific selection of sensor parameters is of paramount importance. Thrust generation in a twin-spool turbofan engine is a complex thermodynamic process governed synergistically by external environmental conditions, control commands, and internal rotor dynamics. As listed in Table 1, eight measurable physical parameters are carefully selected as network inputs in this study.

Environmental parameters (ambient pressure $P_0$ and ambient temperature $T_0$) establish the inlet boundary conditions of the engine. They directly determine the inlet air density and corrected mass flow rate, serving as the fundamental baseline for the entire thermodynamic cycle calculation.Control and geometry parameters explicitly reflect the engine's operating regime and control intentions. The power lever angle ($PLA$) represents the pilot's thrust demand and is highly correlated with the main fuel flow. The coordinated adjustment of the low-pressure and high-pressure guide vane angles ($\alpha_1$ and $\alpha_2$) aims to modify the airflow incidence angles across the compressor stages, ensuring aerodynamic matching and surge margins. Furthermore, the nozzle throat diameter ($D_8$) directly controls the engine's expansion ratio and the exhaust process, exerting a decisive impact on the exhaust velocity and the generation of final available thrust.State parameters (low-pressure rotor physical speed $n_1$ and high-pressure rotor physical speed $n_2$), acting as the core system state variables, directly characterize the dynamic response levels of mechanical work absorbed and delivered by the core engine and the fan.

To transform the one-dimensional time-series sensor data into spatial node features suitable for the graph neural network, the selected physical parameters are mapped onto seven key gas path components of the engine, as detailed in Table 2.

Table 1 – Physical sensor parameters for experimental validation.

| Index | Description | Label |
|---|---|---|
| 1 | Ambient pressure | $P_0$ |
| 2 | Ambient temperature | $T_0$ |
| 3 | Power lever angle | $PLA$ |
| 4 | Low pressure guide vane angle | $\alpha_1$ |
| 5 | High pressure guide vane angle | $\alpha_2$ |
| 6 | Low pressure rotor physical speed | $n_1$ |
| 7 | High pressure rotor physical speed | $n_2$ |
| 8 | Diameter of nozzle throat | $D_8$ |
| 9 | Thrust | $F$ |

Table 2 – Component-based categorization of multi-sensor data.

| Index | Component | Sensors |
|---|---|---|
| 1 | Fan | $T_0$, $P_0$ |
| 2 | LPC | $n_1$, $\alpha_1$ |
| 3 | HPC | $n_2$, $\alpha_2$ |
| 4 | Combustor | $PLA$ |
| 5 | HPT | $n_2$ |
| 6 | LPT | $n_1$ |
| 7 | Nozzle | $D_8$ |

## 3.3 Data Preprocessing

Before model training, we normalized the raw sensor data. Since different sensors have distinct units and value ranges, directly feeding the original data into the network may lead to unstable gradient updates, thereby affecting the convergence speed and prediction performance. To address this issue, we adopt the Min–Max normalization method. This ensures that all input features share a consistent numerical scale, improving the training efficiency and stability of the model. The normalization process can be expressed as:

$$x_{norm}^{(i)} = \frac{x^{(i)} - x_{min}^{(i)}}{x_{max}^{(i)} - x_{min}^{(i)}} \cdot (a-b) + b \tag{15}$$

here, $x_{norm}$ denotes the normalized value of the $i-th$ sensor, $x$ represents the original value of the $i-th$ sensor, $x_{max}$ and $x_{min}$ denote the maximum and minimum values of the $i-th$ sensor, $a$ and $b$ denote the maximum and minimum value of the target normalization interval,respectively.

In addition, a larger number of training samples enables the network to learn more informative and diverse features. The sliding window is a commonly used data augmentation technique that extracts overlapping subsequences by moving along the time series with a fixed step size, thereby significantly increasing the volume of training data and enhancing the model's generalization ability. As shown in Figure 6, we employ the sliding window approach to process the original multi-sensor time-series data, generating a more abundant set of samples for model training.

## 3.4 Model Configuration

After data preprocessing, the processed dataset was used to train the model. We implemented the PD-STGNN in PyTorch and trained on a server with an NVIDIA GeForce RTX 3080 GPU. During training, the Adam optimizer ($\beta_1 = 0.9$ and $\beta_2 = 0.999$) was employed, and the mean squared error

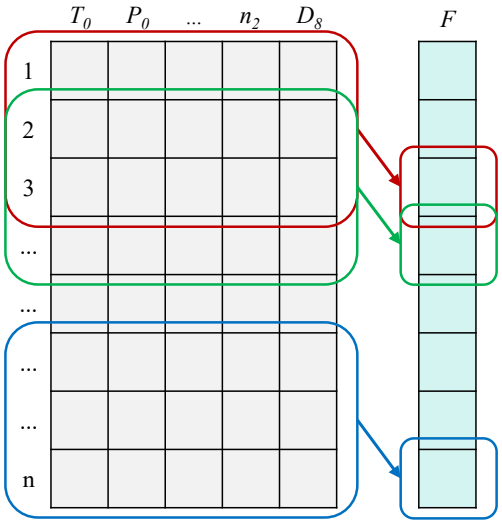



Figure 6 – Sampling of original data within a sliding time window.

(MSE) was adopted as the loss function. The learning rate followed a cosine annealing schedule, decreasing gradually from $1e-3$ to $1e-6$. The batch size was set to 256, and the total number of training epochs was 100. An early stopping strategy was adopted during training, with a patience of 10 epochs. Specifically, the training process was terminated if the validation loss did not decrease for 10 consecutive epochs. The architectural parameters for the network framework are summarized in Table 3.

Table 3 – Architectural parameter settings for PD-STGNN training.

| Hyper-parameters | Value |
|---|---|
| Time step | 32 |
| Hidden units in GRU | 24 |
| Number of layers in GRU | 2 |
| Dropout rate in GRU | 0.3 |
| Component-wise unified feature dimension | 16 |
| Number of stacked Spatio-temporal feature extractors | 3 |
| Number of attention heads $M$ (MHDGA) | 4 |
| Hidden units for Query/Key mapping (MHDGA) | 32 |
| Initial confidence coefficient $\alpha$ (MHDGA) | 3 |
| Output feature dimension $D_{out}$ (GCN) | 32 |
| Number of layers in GCN | 2 |

## 4. Experimental results and discussion

### 4.1 Evaluation metrics

To evaluate the model's prediction performance, the mean absolute percentage error (MAPE) and the root mean square percentage error (RMSPE) are used as quantitative metrics. MAPE measures the overall level of relative error by computing the average percentage difference between the predicted values and the true values, where a smaller value indicates lower average prediction error. RMSPE quantifies the standard deviation of the relative prediction errors in percentage form, reflecting the model's overall prediction accuracy and its sensitivity to larger deviations. The MAPE and RMSPE are calculated as follows:

$$\mathrm{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{\hat{y}_i - y_i}{y_i}\right| \times 100\% \tag{16}$$

$$\mathrm{RMSPE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\frac{\hat{y}_i - y_i}{y_i}\right)^2} \times 100\% \tag{17}$$

here, $\hat{y}_i$ and $y_i$ denote the predicted value and the true value at time step $i$, respectively.

## 4.2 Prediction Results of the PD-STGNN

This section comprehensively evaluates the thrust prediction performance of the proposed PD-STGNN framework across multiple datasets. Table 4 details the prediction errors of the real thrust after denormalization. The results demonstrate that the model achieves exceptionally high prediction accuracy and outstanding generalization performance across four independently split datasets. Specifically, the MAPE for thrust prediction are 1.72%, 1.77%, 2.04%, and 1.96%, respectively, all of which are strictly bounded below a remarkably low level of 2.1%. Furthermore, the RMSPE is below 5.5%. This series of quantitative analyses compellingly verifies that, even when confronted with complex unseen operating conditions, the proposed framework retains a robust capability to deliver highly stable and high-fidelity prediction outputs.

Table 4 – MAPE and RMSPE of predicted thrust.

| Parameter | dataset1 | | dataset2 | | dataset3 | | dataset4 | |
|---|---|---|---|---|---|---|---|---|
| | MAPE | RMSPE | MAPE | RMSPE | MAPE | RMSPE | MAPE | RMSPE |
| $F$ | 1.72% | 3.43% | 1.77% | 3.71% | 2.04% | 4.14% | 1.96% | 5.41% |

To facilitate a clearer analysis of the results, the predicted percentage error (PE) was plotted as a boxplot to evaluate the quality of the predictions. The calculation method of PE can be expressed as follows:

$$\mathrm{PE} = \frac{\hat{y}_i - y_i}{y_i} \times 100\% \tag{18}$$

Figure 7 illustrates the distributions of the thrust prediction percentage error across the four datasets. It can be observed that the error distributions for all datasets are highly concentrated around the zero-error baseline. The vast majority of errors are confined to a remarkably narrow interval and exhibit excellent symmetrical characteristics, with the maximum absolute error strictly bounded within 8%. This further demonstrates that the model possesses the robust capability to maintain high prediction accuracy even when encountering extreme values or outliers.

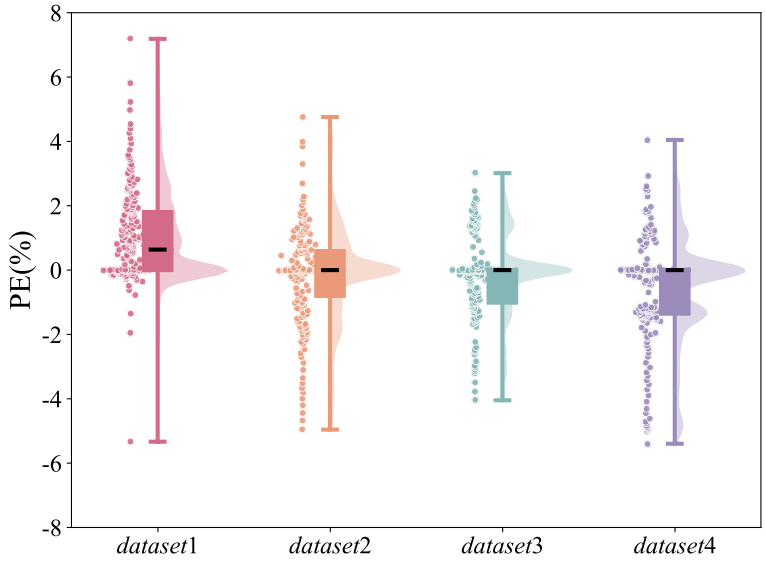



Figure 7 – Thrust prediction error distribution.

Figure 8 selects the complete test sequence from one of the datasets to illustrate in detail the dynamic evolution trajectories of the normalized predicted thrust and the ground truth across consecutive time

steps. As clearly observed from the figure, both in stable steady-state operating regimes and during frequent, drastic transient variations of acceleration and deceleration, the prediction curves of the PD-STGNN perfectly overlap with the ground-truth trajectories. The model not only successfully captures the non-linear transient response of thrust variations but also exhibits no discernible time lag or overshoot. This compellingly demonstrates the effectiveness of the physics-informed priors and the spatiotemporal dynamic graph learning mechanism embedded within the framework for capturing complex aerodynamic and thermodynamic transient evolution laws.

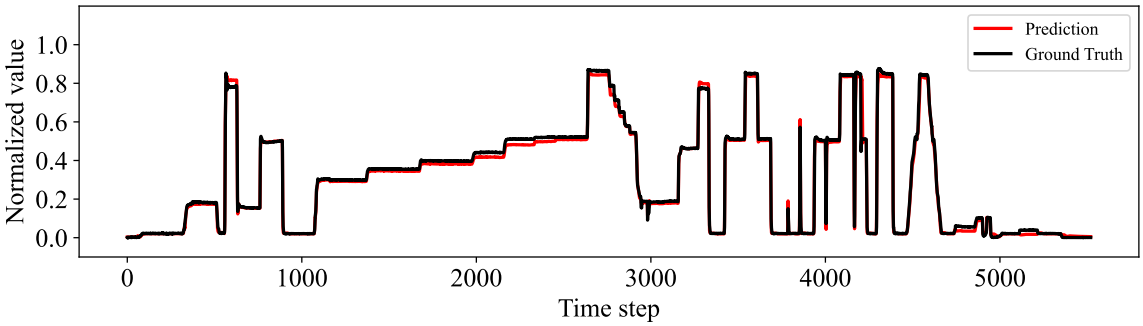



Figure 8 – Thrust prediction results.

To comprehensively evaluate the superiority of the proposed model, the PD-STGNN is compared with several mainstream deep learning baseline models, including LSTM, GRU, Transformer, and GCN. Table 5 explicitly details the quantitative evaluation results of thrust prediction across the four datasets. It is crucial to emphasize that all error metrics (MAPE and RMSPE) in the table are calculated after denormalizing the network outputs back to their actual physical scales, thereby genuinely reflecting the absolute predictive accuracy of the models in practical engineering applications. Experimental results demonstrate that the PD-STGNN framework exhibits significant performance advantages across the majority of evaluation metrics, achieving the optimal values in seven out of the eight assessed criteria.

Table 5 – Prediction results of $F$ using various methods.

| Method | dataset1 | | dataset2 | | dataset3 | | dataset4 | |
|---|---|---|---|---|---|---|---|---|
| | MAPE | RMSPE | MAPE | RMSPE | MAPE | RMSPE | MAPE | RMSPE |
| LSTM | 1.97% | 6.06% | 1.93% | 5.83% | 2.93% | 6.45% | 2.59% | 6.52% |
| GRU | 2.23% | 7.37% | 1.91% | 6.37% | 2.78% | 6.37% | 2.87% | 7.57% |
| Transformer | 2.67% | 4.86% | 1.78% | 4.84% | 3.37% | 5.84% | 2.55% | 6.56% |
| GCN | 1.81% | 3.89% | 2.92% | 4.55% | 2.18% | 5.05% | **1.78%** | 6.34% |
| PD-STGNN | **1.72%** | **3.43%** | **1.77%** | **3.71%** | **2.04%** | **4.14%** | 1.96% | **5.41%** |

## 5. Conclusion

This study proposed a Physics-informed Dynamic Spatio-Temporal Graph Neural Network framework for accurate aero-engine thrust prediction under complex operating conditions. Breaking through the black-box limitations of traditional pure data-driven models, this framework explicitly embeds the engine's internal aerodynamic and thermodynamic coupling laws into the network architecture. Specifically, multi-sensor condition monitoring data are first mapped into a component-wise graph structure based on the actual gas path flow, establishing a foundational spatial representation with clear topological significance. Subsequently, the model deeply mines temporal state evolution trends and innovatively introduces a dynamic graph learning mechanism (MHDGA) equipped with physical prior constraints. This mechanism adaptively infers implicit dynamic directed coupling paths among components while strictly adhering to fluid mechanics laws.

To fully demonstrate the effectiveness and engineering feasibility of the proposed framework, comprehensive experimental studies were conducted on 20 hours of real ground test data collected from

10 twin-spool turbofan engines. The model was systematically evaluated against multiple mainstream deep learning baselines across diverse steady-state and transient operating regimes. The main conclusions and contributions are summarized as follows:

- A physics-informed spatio-temporal graph framework is proposed for aero-engine thrust prediction. By mapping sensor data to a seven-node gas-path topology and unifying features via CLT module, this approach transforms raw multi-sensor data into a graph representation with explicit physical significance.
- A Multi-Head Dynamic Graph Attention mechanism is designed to infer time-varying interactions among components. By injecting a static physical adjacency matrix via a learnable parameter, it ensures the adaptively evolving dynamic graph strictly adheres to fluid mechanics laws while eliminating spurious correlations.
- Comprehensive experiments on 20 hours of real ground-test data from 10 turbofan engines demonstrate the superiority of the PD-STGNN. Outperforming mainstream baselines, it bounds the thrust prediction MAPE below 2.1% and RMSPE below 5.5%, showcasing exceptional accuracy and robustness across complex operating regimes.

Future research could focus on extending the PD-STGNN framework to multi-task learning paradigms, enabling the simultaneous prediction of diverse performance and health parameters (such as exhaust gas temperature and compressor outlet pressure) throughout the full life cycle of the aero-engine.

## 6. Contact Author Email Address

2022201912@mail.nwpu.edu.cn

## 7. Copyright Statement

The authors confirm that they, and/or their company or organization, hold copyright on all of the original material included in this paper. The authors also confirm that they have obtained permission, from the copyright holder of any third party material included in this paper, to publish it as part of their paper. The authors confirm that they give permission, or have obtained permission from the copyright holder of this paper, for the publication and distribution of this paper as part of the ICAS proceedings or as individual off-prints from the proceedings.

## References

[1] Hang Zhou, Maryam Farsi, Andrew Harrison, Ajith Kumar Parlikad, and Alexandra Brintrup. Civil aircraft engine operation life resilient monitoring via usage trajectory mapping on the reliability contour. *Reliab. Eng. Syst. Saf.*, 230:108878, 2023.

[2] Sangjo Kim, Ju Hyun Im, Myungho Kim, Junghoe Kim, and You Il Kim. Diagnostics using a physics-based engine model in aero gas turbine engine verification tests. *Aerosp. Sci. Technol.*, 133:108102, 2023.

[3] Amare D Fentaye, Aklilu T Baheta, Syed I Gilani, and Konstantinos G Kyprianidis. A review on gas turbine gas-path diagnostics: State-of-the-art methods, challenges and opportunities. *Aerospace*, 6(7):83, 2019.

[4] Minglan Xiong, Huawei Wang, Qiang Fu, and Yi Xu. Digital twin–driven aero-engine intelligent predictive maintenance. *The International Journal of Advanced Manufacturing Technology*, 114(11):3751–3761, 2021.

[5] Yufeng Huang, Jun Tao, Gang Sun, Tengyun Wu, Liling Yu, and Xinbin Zhao. A novel digital twin approach based on deep multimodal information fusion for aero-engine fault diagnosis. *Energy*, 270:126894, 2023.

[6] Hong Xiao, Zhe-Zhu Xu, Lae-Sung Kim, Dong-yang Li, and Sung-Ki Lyu. Optimization scheme of genetic algorithm and its application on aeroengine fault diagnosis. *Int. J. Precis. Eng. Manuf.*, 16(4):735–741, 2015.

[7] Chuming Gao, Zilang Huang, Aiyang Yu, Hong Xiao, and Zhishu Zhang. Similarity theory-driven dimensionless parameter migration framework: Digital twin approach for cross-condition gas turbine performance monitoring. *Appl. Therm. Eng.*, page 129712, 2026.

[8] Dasheng Xiao, Shuo Song, Hong Xiao, and Zhanxue Wang. Predicting the performance status of aero-engines using a spatio-temporal decoupled digital twin modeling method. *Adv. Eng. Inform.*, 65:103218, 2025.

[9] Qingfei Min, Yangguang Lu, Zhiyong Liu, Chao Su, and Bo Wang. Machine learning based digital twin framework for production optimization in petrochemical industry. *Int. J. Inf. Manage.*, 49:502–519, 2019.

[10] Stefan Mihai, Mahnoor Yaqoob, Dang V Hung, William Davis, Praveer Towakel, Mohsin Raza, Mehmet Karamanoglu, Balbir Barn, Dattaprasad Shetve, Raja V Prasad, et al. Digital twins: A survey on enabling technologies, challenges, trends and future prospects. *IEEE Commun. Surv. Tutor.*, 24(4):2255–2291, 2022.

[11] Vehbi Emrah Atasoy, Ahmet Esat Suzer, and Selcuk Ekici. A comparative analysis of exhaust gas temperature based on machine learning models for aviation applications. *J. Energy Resour. Technol.-Trans. ASME*, 144(8):082101, 2022.

[12] Zepeng Wang and Yongjun Zhao. Data-driven exhaust gas temperature baseline predictions for aero-engine based on machine learning algorithms. *Aerospace*, 10(1):17, 2022.

[13] Sangjo Kim, Kuisoon Kim, and Changmin Son. Transient system simulation for an aircraft engine using a data-driven model. *Energy*, 196:117046, 2020.

[14] Rojo Princy Andrianantara, Georges Ghazi, and Ruxandra M Botez. Aircraft engine performance model identification using artificial neural networks. In *AIAA propulsion and energy 2021 forum*, page 3247, 2021.

[15] Xuan Wang, Chen Kong, Yunxiao Han, and Juntao Chang. Adaptive physical loss weighting model for temperature field prediction in the aero-engine combustor. *Appl. Therm. Eng.*, page 126707, 2025.

[16] Jiahao Gao, Youren Wang, and Zejin Sun. An interpretable rul prediction method of aircraft engines under complex operating conditions using spatio-temporal features. *Meas. Sci. Technol.*, 35(7):076003, 2024.

[17] Dasheng Xiao, Aiyang Yu, Shuo Song, Hong Xiao, and Zhanxue Wang. Engineformer: A digital twin model for predicting aero-engine performance and degradation. *Adv. Eng. Inform.*, 71:104273, 2026.

[18] Chenchen Song, Hao Shi, Hong Xiao, and Aiyang Yu. Aircraft engine performance prediction via a physics-informed two-stage network framework. *Aerospace Science and Technology*, page 112051, 2026.

[19] Huihui Li, Linfeng Gou, Huacong Li, and Zhidan Liu. Physics-guided neural network model for aeroengine control system sensor fault diagnosis under dynamic conditions. *Aerospace*, 10(7):644, 2023.

[20] Yufeng Huang, Jun Tao, Junyi Zhao, Gang Sun, Kai Yin, and Junyi Zhai. Graph structure embedded with physical constraints-based information fusion network for interpretable fault diagnosis of aero-engine. *Energy*, 283:129120, 2023.

[21] Ming Jin, Huan Yee Koh, Qingsong Wen, Daniele Zambon, Cesare Alippi, Geoffrey I Webb, Irwin King, and Shirui Pan. A survey on graph neural networks for time series: Forecasting, classification, imputation, and anomaly detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, pages 10466–10485, 2024.

[22] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *Int. Conf. Mach. Learn.*, pages 1263–1272. Pmlr, 2017.

[23] Indro Spinelli, Simone Scardapane, and Aurelio Uncini. Adaptive propagation graph convolutional network. *IEEE Trans. Neural Netw. Learn. Syst.*, 32(10):4755–4760, 2020.