

Semi-bipartite Graph based Representation Learning Method for Remaining Useful Life Prognosis with Missing Values

Yudi Zhang, Hongpeng Yin, Zesong Hu and Tengfei Zhang, *Senior Member, IEEE*

Abstract—Remaining useful life (RUL) prediction plays a pivotal role in prognostics and health management (PHM) systems, which enhances the reliability of operating equipment and reduces maintenance costs. With the advent of Industrial Internet of Things (IIoT) technology, it becomes feasible to obtain performance-degradation data that precisely mirrors the health status of equipment. This facilitates real-time process monitoring of device status and promotes intelligent predictive maintenance methods, thereby achieving more accurate RUL estimations. Nonetheless, IIoT systems often suffer from sensing or communication failures in practical industrial scenarios, leading to fragmented multivariate time-series data with missing observations from partial edge devices, which severely restricts the performance of RUL prediction. To address this issue, a graph-based representation learning method is skillfully proposed for RUL prediction with missing values. Specifically, the observations and features in multivariate time-series are regarded as two distinct nodes types in a semi-bipartite graph. Furthermore, the observed values are viewed as real edges whereas the dependency between timestamps and the correlation between features as virtual edges. In this scheme, the representation learning for multivariate time-series is expressed as the graph-level tasks. The acquired representations in the observation missing mode possess the capability to extract meaningful information and unveil underlying patterns from fragmented data, which bolsters the performance of RUL estimations through direct imputation or end-to-end architecture. The effectiveness and robustness of the proposed method are validated through the results of comparative experiments under different missing patterns using the C-MAPSS dataset, demonstrating its capability to enhance predictive maintenance in IIoT-enabled systems while reducing unexpected failures and maintenance costs.

Index Terms—Remaining useful life (RUL), missing values, representation learning, semi-bipartite graph

I. INTRODUCTION

THE prognostics and health management (PHM) technology has transformed the perception of system safety and product reliability from traditional fault diagnosis to degradation state recognition [1], [2], resulting in a wide

This work was supported in part by the National Natural Science Foundation of China under Grant 62273062, in part by the Fundamental Research Funds for the Central Universities under Grant 2023CDJKYJH029, in part by the Natural Science Foundation of Chongqing under Grant 2022NSCQ-LZX0324 and in part by the National Key Research and Development Program of China under Grant 2023YFB3308800. (*Corresponding author: Hongpeng Yin*)

Yudi Zhang, Hongpeng Yin and Zesong Hu are with the School of Automation, Chongqing University, Chongqing 400044, China (e-mail: zyd971001@126.com; yinhongpeng@cqu.edu.cn; 20241301015@stu.cqu.edu.cn)

Tengfei Zhang is with the College of Automation and College of Artificial Intelligence, Nanjing University of Posts and Telecommunications, Nanjing 210023, China (e-mail: tfzhang@126.com)

variety of applications, such as manufacturing engineering, robot-based systems and aerospace industries. As a crucial task of PHM technology, remaining useful life prediction (RUL) mainly evaluates the performance status of equipment and guides people to repair or replace equipment promptly, thereby effectively preventing safety issues and economic losses caused by machinery malfunctions [3], [4].

Generally, RUL prediction methods could be roughly divided into model-based methods and data-driven methods [5]. Model-based methods attempt to explain the correlation of undergone trends to degradation processes by establishing specific physical models based on certain failure mechanisms [6]. Although they have high accuracy and interpretability for the RUL estimations, it is impractical to acquire substantial comprehensive knowledge and specific degradation mechanisms from large-scale complex systems. With the development of Industrial Internet of Things (IIoT) technology, it becomes feasible to collect a substantial volume of performance-degradation data that accurately reflects the health condition of equipment so the data-driven methods have recently gained considerable attention. Currently, data-driven methods include statistical models, machine learning (ML) and deep learning (DL) [7]. Statistical-based approaches aim to derive the probability density function of monitoring data and further obtain the RUL estimations of degraded equipment by constructing mathematical models such as the Wiener process [8], Gaussian process model [9], etc. Despite this, it is challenging to fulfill specific distribution conditions in non-stationary industrial processes. In contrast, machine learning models [10], such as support vector regression (SVR), random forests (RF), etc., directly establish a mapping between monitoring data and the RUL labels without physical mechanisms or statistical assumptions. However, they also have own limitations as model size or number of operations, exhibiting poor performance in handling coupled or nonlinear issues from complex multivariate industrial time-series data.

Distinct from ML methods, newly emerging deep learning models, such as convolutional neural network (CNN) [11], long short-term memory network (LSTM) [12], [13], gated recurrent unit network (GRU) [14], etc., possess powerful capability of extracting relevant features from numerous condition monitoring data, enabling it to capture complex nonlinear relationships between the preformance-degradation data and the health status of systems without manual feature selection, thus gradually becoming mainstream for RUL prediction. For instance, Zhang et al. [12] proposed a bi-directional LSTM-

based prediction model for RUL estimations of aircraft engines, where the underlying long-term dependencies in time-series data can be efficiently identified. Ren et al. [15] further proposed a deep learning-based RUL prediction method, namely Auto-CNN-LSTM. It combines the merits of CNN and LSTM modules, thereby achieving deeper information extraction in finite data of lithium-ion batteries. Aimed at different working conditions, Ye et al. [16] proposed a selective adversarial adaptation network for predicting RUL of rotating machines, where the feature alignment and knowledge transfer are conducted to reduce the distribution discrepancy between different domains. Zhou et al. [17] proposed a dual-thread GRU network for gear RUL prediction. This dual-thread learning strategy possess the ability to extract nonstationary information from the recurrent path, leading to more robust estimations. In recent years, digital twin technology has revolutionized industrial health management by creating real-time virtual replicas of physical systems [18]. Through sensor-model integration, this approach significantly enhances operational reliability while reducing downtime. For example, Cui et al. [19] developed a full lifecycle dynamic twin model of bearings to generate rich twin data, where the evolution of surface morphology and roller relative slip at various stages are successfully simulated. Feng et al. [20] presented an intelligent health management method driven by digital twin technology to monitor and evaluate the progression of gear surface degradation. The proposed methodology also effectively uncovers the potential characteristics of gear wear propagation and achieves accurate RUL estimations.

However, in real-world IIoT scenarios, time-series data collected from industrial devices frequently suffer from missing observations caused by sensor failures, communication disruptions, or limitations of edge devices. These data imperfections pose serious challenges to existing RUL prediction models. A widely used workaround is to discard incomplete samples or apply statistical imputation techniques (e.g., mean or median filling), but these strategies often result in information loss or fail to adequately capture the essential temporal dynamics and inter-feature dependencies inherent in highly intricate multivariate time-series data. For sake of this, several studies have explored statistical-based imputation technologies in industrial settings. For example, Wu et al. [21] adopted multiple imputation by chained equations for dealing with missing values. Inspired by cumulative damage process, Peng et al. [22] designed a novel data imputation method for the missing shock time by integrating rejection sampling and order statistics. Recently, neural network-based methods leverage the capacity of neural networks to learn complex patterns and impute missing values, offering promising solutions to handle missing observations in large-scale industrial environments. For example, Liu et al. [23] presented a deep sparse auto-encoder for data reconstruction of wind turbines. Lv et al. [24] proposed a multi-feature generation adversarial network (GAN) for solving industrial time-series data with high missing ratio. Jeong et al. [25] proposed a novel graph-based convolutional network for industrial lubricant data imputation considering feature dependency between different data points. Moreover, Gong et al. [26] presented a transformer-based

data imputation model that integrates domain knowledge via operational and measurement constraints, ensuring that the imputed time-series are both realistic and consistent with existing measurements.

It is widely acknowledged that the primary goal of data imputation is to improve the performance of downstream RUL prediction tasks in IIoT scenarios with incomplete datasets. However, existing data imputation methods often fail to fully exploit the valuable information inherently embedded within missing data patterns, which significantly hampers the accuracy of subsequent RUL estimations. This gap underscores the pressing need for an effective learning strategy that can extract and utilize meaningful information from fragmented data, thereby improving its applicability and effectiveness for downstream RUL prediction tasks.

In recent years, representation learning (RL) [27], [28] has emerged as an extensively studied paradigm in the machine learning community, enabling the extraction of meaningful features by uncovering the intrinsic structures within raw data. This capability is particularly crucial for tasks involving missing observations, where the challenge extends beyond simply filling data gaps to effectively leveraging learned representations for tackling specific downstream tasks. Such an approach presents a promising prospect for RUL prediction with incomplete datasets. Motivated by this, a semi-bipartite graph based representation learning method is proposed for RUL prediction with missing values. Specifically, the observations and features in multivariate time-series are regarded as two distinct nodes types in a semi-bipartite graph. Furthermore, the observed values are viewed as real edges whereas the dependency between timestamps and the correlation between features as virtual edges. In this scheme, the representation learning for multivariate time-series is expressed as the graph-level tasks. The acquired representations in the observation missing mode possess the capability to extract meaningful information and unveil underlying patterns from the fragmented data, which bolsters the performance of RUL prediction efficiently.

The contributions of this paper are highlighted as follows:

(1) A graph-based representation learning method is innovatively proposed for RUL prediction with missing values. The learned representation can extract meaningful information and unveil underlying patterns from the fragmented data, thereby achieving excellent RUL estimations.

(2) A semi-bipartite graph is skillfully introduced into the graph-based representation learning, where the mappings of different graph nodes are depicted through real and virtual edges. This structure can capture the dependency between timestamps and the correlation between features more efficiently in multivariate time-series data.

(3) The proposed representation learning model allows us to utilize acquired representations for imputing missing values instantly or serve as a backbone network for directly RUL prediction in an end-to-end fashion, which comprehensively considers the efficiency and accuracy of proposed method.

The rest of this paper is organized as follows. Section II briefly describes the problem formulation of RUL prediction with missing values. Section III presents the proposed graph-based RL method. In Section IV, the superiority and effective-

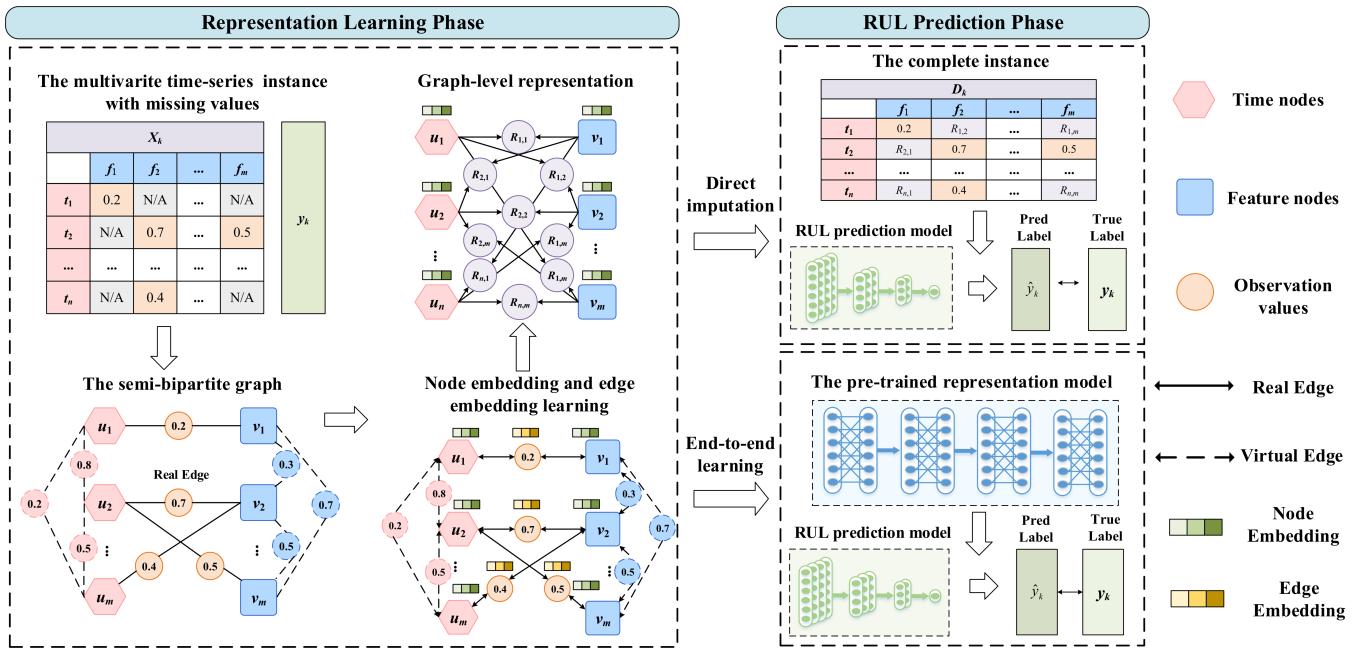


Fig. 1. The illustration of the proposed representation method for RUL prediction with missing values

ness of proposed framework are investigated under different missing patterns using the C-MAPSS dataset. Section V covers some conclusions.

II. PROBLEM FORMULATION

Let $\mathbf{D} = [\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_{N-1}, \mathbf{X}_N] \in \mathbb{R}^N$ represents the set of multivariate time-series instances collected from different machines or time periods and $\mathbf{Y} \in \mathbb{R}^N$ the corresponding labels for RUL values, where the k -th instance is denoted as \mathbf{X}_k and y_k respectively. For $\forall \mathbf{X}_k \in \mathbb{R}^{n \times m}$, it consists of n continuous time points and m features while the observation value of the i -th time point and the j -th feature is denoted as $X_k^{i,j}$. When confronted with fragmented data, the missing items can be recorded as a mask matrix $M_k \in \{0, 1\}^{n \times m}$ where the value of $X_k^{i,j}$ cannot be observed if $M_k^{i,j} = 0$. In this settings, \mathbf{X}_k^o denotes the observed part in k -th instance that satisfies $M_k^{i,j} = 1$ while \mathbf{X}_k^r the residual missing parts. To fill the gap, a continuous data imputation model $f_i(\cdot)$ is expected to establish for restoring missing values as much as possible, which is defined as follows.

$$\min_{\theta} \frac{1}{|N|} \sum_{k=1}^N \mathcal{L}(f_i(\mathbf{X}_k^o; \Psi_k), \mathbf{X}_k^r) \quad (1)$$

where Ψ_k represents the missing ratio of the k -th instance, $\mathcal{L}(\cdot)$ denotes the loss function and θ the optimized parameters of the imputation model $f_i(\cdot)$.

Despite this, the above scheme only targets the missing values themselves, and fails to consider whether the embedded information is effective for RUL prediction. In view of this, we prefer to establish an effective imputation model $f_i(\cdot)$ which is suitable for the downstream prediction model $f_p(\cdot)$. The formulaic expression is as follows.

$$\begin{aligned} & \frac{1}{|N|} \sum_{i=1}^N \mathcal{L}(f_p(f_i(\mathbf{X}_k^o; \Psi_k), \mathbf{X}_k^o), y_k) \\ &= \frac{1}{|N|} \sum_{i=1}^N \mathcal{L}(f_p(\mathbf{X}_k), y_k) \end{aligned} \quad (2)$$

As can be seen from Eq.(2), the prediction loss obtained with complete data is the same as that obtained with observed and imputed data. Moreover, a representation model $f_r(\cdot)$ can be designed to uncover the underlying structures or patterns of fragmented data rather than fill the gap, thereby adapting to downstream prediction tasks in an end-to-end fashion. The formulaic expression is as follows.

$$\begin{aligned} & \frac{1}{|N|} \sum_{i=1}^N \mathcal{L}(f_p(f_r(\mathbf{X}_k^o; \Psi_k)), y_k) \\ &= \frac{1}{|N|} \sum_{i=1}^N \mathcal{L}(f_p(\mathbf{X}_k), y_k) \end{aligned} \quad (3)$$

Effectively fitting this objective provides a promising solution to the RUL prediction with missing values. The key challenge is how to design a feasible imputation or representation model adapting to downstream RUL prediction.

III. METHODOLOGY

In this section, a graph-based representation learning method for RUL prediction with missing values is skillfully proposed, which is shown in Fig. 1. The key idea of this method is to represent multivariate time series with missing values as a semi-bipartite graph, where the mappings of different graph nodes are depicted through real and virtual edges. Furthermore, a novel graph representation learning strategy is

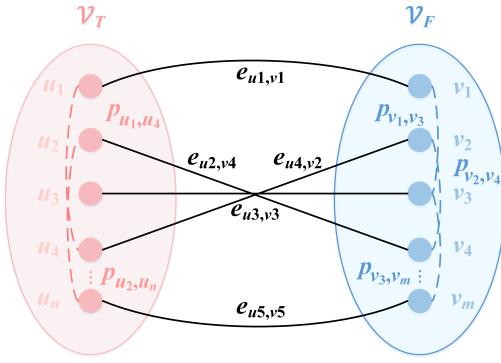


Fig. 2. The structure of the semi-bipartite graph

specially designed for the semi-bipartite graph considering the dependency between timestamps and the correlation between features. Therefore, the representation for multivariate time-series is regarded as the graph-level task which is composed of the concatenations combining different types of node embeddings. In this scheme, the proposed representation model can utilize acquired representations for imputing missing values instantly or serve as a backbone network for directly RUL prediction through the end-to-end architecture.

A. Graph Representation for Incomplete Data

An undirected semi-bipartite graph is defined as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where \mathcal{V} is the node set which includes two distinct types of nodes $\mathcal{V} = \mathcal{V}_T \cup \mathcal{V}_F$, where the time nodes $\mathcal{V}_T = \{u_1, u_2, \dots, u_n\}$, the feature nodes $\mathcal{V}_F = \{v_1, v_2, \dots, v_m\}$. In addition, \mathcal{E} is the edge set that consists of two types of edges, where the real edges $\mathcal{E}_R = \{(e_{u_i, v_j}) | u_i \in \mathcal{V}_T, v_j \in \mathcal{V}_F\}$ only exist between two types of nodes, while the virtual edges $\mathcal{E}_V = \{(p_{u_i, u_j}) | u_i, u_j \in \mathcal{V}_T\} \cup \{(p_{v_i, v_j}) | v_i, v_j \in \mathcal{V}_F\}$ only exist in the same type of nodes. The structure of the semi-bipartite graph is shown in Fig. 2.

Given the incomplete instance X_k and its corresponding mask M_k , the time node u_i and the feature node v_j adopt n -dimensional and m -dimensional one-hot encoding respectively, and the real edge e_{u_i, v_j} is embedded with the corresponding value $X_k^{i,j}$. Distinct from the bipartite graph, the embeddings of the visual edges $p_{u_i, u_j}, p_{v_i, v_j}$ are based on inherent knowl-

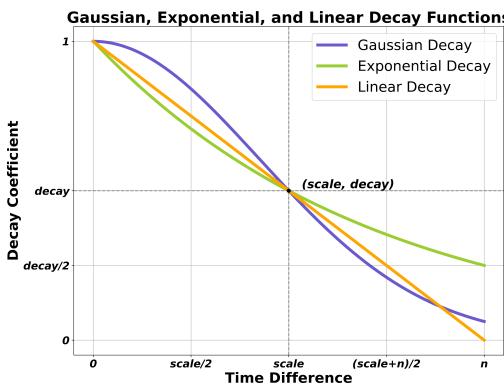


Fig. 3. The illustration of used time decay functions

edge in multivariate time-series which involves the dependency between timestamps and the correlation between features.

For the virtual edge p_{u_i, u_j} connecting two identical nodes in \mathcal{V}_T , it is defined through the time dependency between u_i and u_j , which is as follows.

$$p_{u_i, u_j} = d(|t_i - t_j|) \quad (4)$$

where t_i is the time point of the node u_i , and $d(\cdot)$ denotes the time decay function. The common used function include the linear decay function, exponential decay function and Gaussian decay function, which are formulated as follows.

$$d_1(x) = \exp\left(-\left(\frac{x^2}{2\sigma^2}\right)\right) \quad (5)$$

$$d_2(x) = \exp(-\lambda x) \quad (6)$$

$$d_3(x) = \max(\omega x + b, 0), \omega < 0, b = 1 \quad (7)$$

The above time decay functions are illustrated in Fig. 3 vividly, where $(\text{scale}, \text{decay})$ represents that when the x -axis of the function ranges from 0 to scale , and the y -axis reaches decay . This facilitates adjusting the attenuation rate for different functions.

Afterwards, the time dependency matrix P_t can be defined.

$$P_t = \begin{bmatrix} p_{u_1 u_1} & p_{u_1 u_2} & \dots & p_{u_1 u_n} \\ p_{u_2 u_1} & p_{u_2 u_2} & \dots & p_{u_2 u_n} \\ \dots & \dots & \dots & \dots \\ p_{u_n u_1} & p_{u_n u_2} & \dots & p_{u_n u_n} \end{bmatrix} \in \mathbb{R}^{n \times n} \quad (8)$$

For the virtual edge p_{v_i, v_j} connecting two identical nodes in \mathcal{V}_F , it is also defined through the correlation between v_i and v_j , which is as follows.

$$p_{v_i, v_j} = \sum_{k=1}^N \frac{1}{N} |\rho_k(c_i, c_j)| \quad (9)$$

where c_j is the feature of the node v_j , $\rho_k(\cdot)$ denotes the correlation function in k -th instance and N the total number of samples. For sake of this, the Spearman coefficient is introduced to describe the correlation between different features.

$$\begin{aligned} \rho_k(c_i, c_j) = & \frac{1}{n} \sum_{s=1}^n \left(R(M_{s,i}^k) - \overline{R(M_{s,i}^k)} \right) \left(R(M_{s,j}^k) - \overline{R(M_{s,j}^k)} \right) \cdot \\ & \left(\frac{1}{n} \sum_{s=1}^n \left(R(M_{s,i}^k) - \overline{R(M_{s,i}^k)} \right)^2 \right)^{-1/2} \cdot \\ & \left(\frac{1}{n} \sum_{s=1}^n \left(R(M_{s,j}^k) - \overline{R(M_{s,j}^k)} \right)^2 \right)^{-1/2}. \end{aligned} \quad (10)$$

where $R(\cdot)$ denotes the rank, and $\overline{R(\cdot)}$ the average rank.

In this scheme, the corresponding feature correction matrix P_c can also be defined.

$$P_c = \begin{bmatrix} p_{v_1 v_1} & p_{v_1 v_2} & \dots & p_{v_1 v_m} \\ p_{v_2 v_1} & p_{v_2 v_2} & \dots & p_{v_2 v_m} \\ \dots & \dots & \dots & \dots \\ p_{v_m v_1} & p_{v_m v_2} & \dots & p_{v_m v_m} \end{bmatrix} \in \mathbb{R}^{m \times m} \quad (11)$$

It is mentioned that the embeddings of nodes and real edges need to be updated through subsequent representation learning strategy. But for virtual edges, they are only used to assist in representing nodes without updating.

B. Graph Representation Learning Strategy

The representation learning strategy adopts a graph neural network(GNN) framework inspired by GRAPE [29], which is a variant of GrapeSAGE by introducing edge embeddings [30]. Specially, we extend GRAPE to a semi-bipartite graph by adding relevant knowledge from virtual edges so as to capture the spatiotemporal correlations and further represent the graph-level tasks in multivariate time-series.

For the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, the message passing at each GNN layer l can be achieved by taking the concatenation of the node embeddings $\mathbf{h}_{u_i}^{(l-1)}$, $\mathbf{h}_{v_j}^{(l-1)}$, the edge embeddings $\mathbf{e}_{u_i, v_j}^{(l-1)}$, and the spatiotemporal correlations \mathbf{p}_{u_i, u_j} , \mathbf{p}_{v_i, v_j} .

Taking the learning process of the time node embedding $\mathbf{h}_{u_i}^{(l)}$ at each GNN layer l as an example, the neighbor embedding $\mathbf{n}_{u_i, v}^{(l)}$ and $\mathbf{n}_{u_i, u}^{(l)}$ are updated as follows.

$$\begin{aligned} \mathbf{n}_{u_i, v}^{(l)} &= \sigma(\mathbf{W}_r^{(l)} \cdot \text{CONCAT} \\ &(\mathbf{h}_{u_i}^{(l-1)}, \text{mean}(\{\mathbf{e}_{u_i, v}^{(l-1)} | \forall v \in \mathcal{V}_F\})) \end{aligned} \quad (12)$$

$$\begin{aligned} \mathbf{n}_{u_i, u}^{(l)} &= \sigma(\mathbf{W}_v^{(l)} \cdot \text{CONCAT} \\ &(\mathbf{h}_{u_i}^{(l-1)}, \text{mean}(\{\mathbf{p}_{u_i, u} \cdot \mathbf{n}_{u, v}^{(l)} | \forall u \in \mathcal{V}_T, p_{u_i, u} \in \mathbf{P}_t\})) \end{aligned} \quad (13)$$

where σ is the non-linearity, $\mathbf{W}_r^{(l)}$ and $\mathbf{W}_v^{(l)}$ are the trainable weights of the layer l for $\mathbf{n}_{u_i, v}^{(l)}$ and $\mathbf{n}_{u_i, u}^{(l)}$ respectively.

On this foundation, the time node embedding $\mathbf{h}_{u_i}^{(l)}$ is then updated as follows.

$$\mathbf{h}_{u_i}^{(l)} = \sigma(\mathbf{P}^{(l)} \cdot \text{CONCAT}(\mathbf{h}_{u_i}^{(l-1)}, \mathbf{n}_{u_i, v}^{(l)}, \mathbf{n}_{u_i, u}^{(l)})) \quad (14)$$

where $\mathbf{P}^{(l)}$ is the trainable weight of the layer l for $\mathbf{h}_{u_i}^{(l)}$.

The update strategy of the feature node embedding $\mathbf{h}_{v_j}^{(l)}$ adopts a similar approach to that of $\mathbf{h}_{u_i}^{(l)}$, which is as follows.

$$\begin{aligned} \mathbf{n}_{v_i, u}^{(l)} &= \sigma(\mathbf{W}_r^{(l)} \cdot \text{CONCAT} \\ &(\mathbf{h}_{v_i}^{(l-1)}, \text{mean}(\{\mathbf{e}_{v_i, u}^{(l-1)} | \forall u \in \mathcal{V}_T\})) \end{aligned} \quad (15)$$

$$\begin{aligned} \mathbf{n}_{v_i, v}^{(l)} &= \sigma(\mathbf{W}_v^{(l)} \cdot \text{CONCAT} \\ &(\mathbf{h}_{v_i}^{(l-1)}, \text{mean}(\{\mathbf{p}_{v_i, v} \cdot \mathbf{n}_{v, u}^{(l)} | \forall u \in \mathcal{V}_F, p_{v_i, v} \in \mathbf{P}_c\})) \end{aligned} \quad (16)$$

$$\mathbf{h}_{v_i}^{(l)} = \sigma(\mathbf{P}^{(l)} \cdot \text{CONCAT}(\mathbf{h}_{v_i}^{(l-1)}, \mathbf{n}_{v_i, u}^{(l)}, \mathbf{n}_{v_i, v}^{(l)})) \quad (17)$$

Moreover, the edge embedding $\mathbf{e}_{u_i, v_j}^{(l)}$ is updated as follows.

$$\mathbf{e}_{u_i, v_j}^{(l)} = \sigma(\mathbf{Q}^{(l)} \cdot \text{CONCAT}(\mathbf{e}_{u_i, v_j}^{(l-1)}, \mathbf{h}_{u_i}^{(l)}, \mathbf{h}_{v_j}^{(l)})) \quad (18)$$

where $\mathbf{Q}^{(l)}$ is the trainable weight of the layer l for $\mathbf{e}_{u_i, v_j}^{(l)}$.

To sum up, the above representation learning process of the k -th instance can be summarized as follows.

$$[\mathbf{H}_u^k, \mathbf{H}_v^k] = \mathbf{f}_{rn}(\mathbf{X}_k, \mathbf{M}_k, \mathbf{P}_t, \mathbf{P}_c; \mathcal{G}) \quad (19)$$

where $\mathbf{f}_{rn}(\cdot)$ denotes the representation model of nodes. \mathbf{H}_u^k and \mathbf{H}_v^k are the embedding matrix of the time node and feature node respectively, which are as follows.

$$\mathbf{H}_u^k = [\mathbf{h}_{u_1}^k, \mathbf{h}_{u_2}^k, \dots, \mathbf{h}_{u_n}^k] \in \mathbb{R}^{n \times 1} \quad (20)$$

$$\mathbf{H}_v^k = [\mathbf{h}_{v_1}^k, \mathbf{h}_{v_2}^k, \dots, \mathbf{h}_{v_m}^k] \in \mathbb{R}^{m \times 1} \quad (21)$$

where $\mathbf{h}_{u_i}^k$ and $\mathbf{h}_{v_j}^k$ are the learned embeddings of the time and feature node u_i , v_j respectively in the k -th instance.

In this scheme, the acquired embedding matrix \mathbf{H}_u^k and \mathbf{H}_v^k can be concatenated to represent the entire semi-bipartite graph, which is as follows.

$$\mathbf{R}_k = \mathbf{f}_{rg}(\text{CONCAT}(\mathbf{H}_u^k, \mathbf{H}_v^k)) \quad (22)$$

where $\mathbf{f}_{rg}(\cdot)$ denotes the representation model of the entire graph. \mathbf{R}_k is the representation matrix of the semi-bipartite graph in the k -th instance, which is as follows.

$$\mathbf{R}_k = \begin{bmatrix} R_k^{1,1} & R_k^{1,2} & \dots & R_k^{1,m} \\ R_k^{2,1} & R_k^{2,2} & \dots & R_k^{2,m} \\ \dots & \dots & \dots & \dots \\ R_k^{n,1} & R_k^{n,2} & \dots & R_k^{n,m} \end{bmatrix} \in \mathbb{R}^{n \times m} \quad (23)$$

where $R_k^{i,j}$ represents the predicted observation value of the i -th time point and the j -th feature in the k -th instance.

C. RUL Prediction with Learned Representation

After obtaining the graph-level representation for incomplete data, a concerning issue is how to apply the existing representation to downstream RUL predictions. In this section, we provide two different methods for RUL prediction with missing values, namely direct imputation method and end-to-end learning method.

The direct imputation method aims to design a loss function for the representation learning model, so that the reconstructed representations are as close as possible to the real instances, thereby directly imputing the missing values. The formulaic expression is as follows.

$$\begin{aligned} \mathbf{g}_r &= \frac{1}{|N \times m \times n|} \\ \nabla_{\omega_r} \sum_{k=1}^N \sum_{i=1}^n \sum_{j=1}^m \mathcal{L}_{mse}(R_k^{i,j}, X_k^{i,j}; \mathbf{F}_r) \end{aligned} \quad (24)$$

$$\omega_r^t = \omega_r^{t-1} + \text{Adam}(\mathbf{g}_r, \gamma_r, [\beta_1, \beta_2]) \quad (25)$$

where \mathbf{F}_r denotes the entire representation model covering \mathbf{f}_{rn} and \mathbf{f}_{rg} . Adam denotes the optimizer that computes the update of the gradient \mathbf{g}_r . Besides, the learning rate γ_r and betas $[\beta_1, \beta_2]$ needs to be set additionally.

In this scheme, the filled instance D_k that replaces X_k can be obtained and applied to subsequent RUL predictions, where $D_k^{i,j}$ can be denoted as follows.

$$D_k^{i,j} = \begin{cases} X_k^{i,j} & \text{if } M_k^{i,j} = 1 \\ R_k^{i,j} & \text{if } M_k^{i,j} = 0 \end{cases} \quad (26)$$

The purpose of the end-to-end learning method is to build an end-to-end learning model, where the pre-trained representation learning model serves as the backbone module of the

complete model while the prediction model acts as the head module. During the training process, the backbone module is only fine-tuned to make the representations for incomplete data more suitable for downstream RUL prediction tasks. The formulaic expression is as follows.

$$\mathbf{R}_k = \mathbf{F}_r(\mathbf{X}_k, \mathbf{M}_k, \mathbf{P}_t, \mathbf{P}_c; \mathcal{G}) \quad (27)$$

$$\hat{y}_k = \mathbf{F}_p(\mathbf{R}_k; \mathbf{F}_r) \quad (28)$$

$$[\mathbf{g}_r, \mathbf{g}_p] = \frac{1}{|N|} \nabla_{\omega_r, \omega_p} \sum_{k=1}^N \mathcal{L}_{mse}(\hat{y}_k, y_k; \mathbf{F}_r, \mathbf{F}_p) \quad (29)$$

$$\boldsymbol{\omega}_r^t = \boldsymbol{\omega}_r^{t-1} + Adam(\mathbf{g}_r, \gamma_r, [\beta_1, \beta_2]) \quad (30)$$

$$\boldsymbol{\omega}_p^t = \boldsymbol{\omega}_p^{t-1} + Adam(\mathbf{g}_p, \gamma_p, [\beta_1, \beta_2]) \quad (31)$$

where the learning rate $\gamma_r \ll \gamma_p$, so that the representation learning model fine-tunes the parameters to suit downstream prediction tasks while retaining original knowledge. Specifically, the pseudo-code of the proposed representation method for RUL prediction is demonstrated as above.

Overall, the computational complexity of the proposed representation learning framework consists of three main components: graph construction, GNN-based representation learning, and downstream prediction. The graph construction stage involves the generation of real edges and virtual edges encoding temporal dependencies and feature correlations, resulting in a total complexity of $O(n^2 + Nm^2)$. The representation learning process dominates the runtime with a complexity of $O(NL(n+m)^2d)$, where L is the number of layers and d is the embedding dimension. In the end-to-end paradigm, an additional prediction head contributes a computational cost of $O(N(nmh+nh^2))$, where h is the hidden size of the predictor. Taken together, both direct imputation and end-to-end modes are computationally tractable and scalable, enabling effective RUL prediction for incomplete multivariate time series data.

IV. EXPERIMENTAL STUDY

To evaluate the performance of the proposed representation learning method for RUL prediction, the implementation is developed using Python 3.11.5 within the Spyder 5.4.3 IDE. All experiments are conducted on a high-performance server equipped with an Intel Xeon Platinum 8336C CPU and dual NVIDIA RTX 4090 GPUs.

A. Dataset Description

The Commercial Modular Aero-Propulsion System Simulation (C-MAPSS) dataset developed by NASA has been applied in PHM systems. It is obtained by simulating the full life cycle of turbofan engines running to failure using the simulation software. The engine comprises 21 sensors and eight components, including fan, bypass duct, low-pressure compressor, high-pressure compressor and other devices. In addition, the dataset is categorized into four subsets based on different operating settings and fault modes. Here, the subset FD001 and FD003 are investigated and discussed which pertains to a single fault mode and operation condition. Each subset has a training set and a test set. The training set contains

Algorithm 1 The representation learning method for RUL prediction with missing values

Input: The graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$; The training set $\{\mathbf{X}_k, y_k\}_{k=1}^N$ and mask matrix $\{\mathbf{M}_k\}_{k=1}^N$; The representation model \mathbf{F}_r including f_{rn} and f_{rg} ; The RUL prediction model \mathbf{F}_p
Output: The well-trained parameters $\boldsymbol{\omega}_r$ and $\boldsymbol{\omega}_p$ of the representation model \mathbf{F}_r and the prediction model \mathbf{F}_p .

```

1: Acquire the time dependency matrix  $\mathbf{P}_t$  using Eq.(4) and (8)
2: for  $k = 1, 2, \dots, N$  do
3:   Calculate the correction  $\rho_k(c_i, c_j)$  using Eq.(10)
4: end for
5: Acquire the feature correction matrix  $\mathbf{P}_t$  using Eq.(9) and (11)
6: for  $iteration = 1, 2, \dots, I$  do
7:   Train the representation model  $\mathbf{F}_r$  using Eq.(23) and (24)
8: end for
9: Acquire the well-trained parameters  $\boldsymbol{\omega}_r$  of the model  $\mathbf{F}_r$ 
10: if select the direct imputation method then
11:   Impute the missing values using Eq.(25)
12:   for  $iteration = 1, 2, \dots, I$  do
13:     Train the prediction model  $\mathbf{F}_p$  using the function  $\mathcal{L}_{mse}$ 
14:   end for
15:   Acquire the well-trained parameters  $\boldsymbol{\omega}_p$  of the model  $\mathbf{F}_p$ 
16: else if select the end-to-end learning method then
17:   for  $iteration = 1, 2, \dots, I$  do
18:     Train the representation model  $\mathbf{F}_r$  and the prediction model  $\mathbf{F}_p$  using Eq.(26)-(30)
19:   end for
20:   Acquire the well-trained parameters  $\boldsymbol{\omega}_r$  and  $\boldsymbol{\omega}_p$  of the representation model  $\mathbf{F}_r$  and the prediction model  $\mathbf{F}_p$ 
21: end if
```

complete life cycle data while the test set includes partial life cycle data along with RUL labels.

To evaluate the performance of proposed methods, the root-mean-square error (RMSE), mean absolute error (MAE), mean absolute percentage error (MAPE) and coefficient of determination (R^2) are adopted as follows.

$$RMSE = \sqrt{\frac{1}{N} \sum_{n=1}^N (R_{pred}^{(n)} - R^{(n)})^2} \quad (32)$$

$$MAE = \frac{1}{N} \sum_{n=1}^N |R_{pred}^{(n)} - R^{(n)}| \quad (33)$$

$$MAPE = \frac{1}{N} \sum_{n=1}^N \left| \frac{R^{(n)} - R_{pred}^{(n)}}{R^{(n)}} \right| \times 100\% \quad (34)$$

$$R^2 = 1 - \frac{\sum_{n=1}^N (R^{(n)} - R_{pred}^{(n)})^2}{\sum_{n=1}^N (R^{(n)} - \bar{R})^2} \quad (35)$$

where $R_{pred}^{(n)}$ and $R^{(n)}$ are predicted and real RUL labels of the i -th unit, \bar{R} denotes the average of real RUL labels.

B. Data Preprocessing and Model Design

To enable the representation learning model to fully extract the features of parameter variables, the input samples for the network are structured into a two-dimensional structure, with the first dimension representing time and the second dimension

TABLE I
MODEL STRUCTURE AND PARAMETER SETTINGS

Module	Parameter size	
Hyperparameters	learning rate = 0.001, batch size = 128, epoch = 200, dropout = 0.2	
Backbone	Ours	n_layers = 3, node_in = 25, node_out= 32, edge_in = 1, edge_out = 32
	BDLSTM	n_layers = 3, in = 25, hidden = 25, out = 1
	ALSTM	n_layers = 3, in = 25, hidden = 25, out = 1
	ABGRU	n_layers = 3, in = 25, hidden = 25, out = 1
Head	DCNN	n_layers = 5, in_channel = 1, out_channel = 20, kernal = 3, stride=1, padding=2
	ASTGCN	n_layers = 4, node_in = 4, in = 1, out = 64

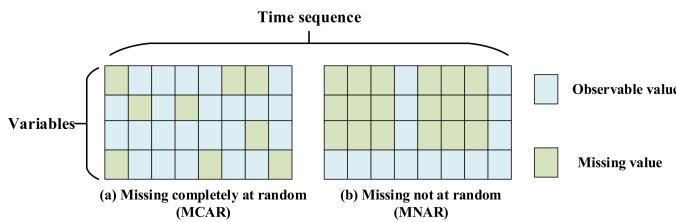


Fig. 4. The illustration of different missing patterns

being variables. Since the operational duration of engines in the training set varies in length, a sliding time window method is adopted to ensure consistency in the shape of input samples. In this settings, the length of time series is set to 30 for segmenting the condition monitoring data of each engine.

Besides, all raw sensor data are normalized using Min-MaxScaler, which scales each feature to the [0, 1] range based on the minimum and maximum values of the training set. This preprocessing step ensures numerical stability during model training and suppresses the influence of extreme values, thereby enhancing the robustness of the learned representations against noisy fluctuations.

Moreover, the networks involved in the experiments consist of proposed representation model for the data imputation or data reconstruction of fragmented sequences, as well as the flexible head model for downstream RUL predictions if the end-to-end learning is selected. The related parameters and structures are shown in Tab. I.

C. Prediction Results and Comparison

For verifying the validity of the proposed method, two different missing data patterns are considered in real industrial scenarios, which is shown in Fig. 4.

In the MCAR scenario, the missing data is entirely random, meaning that the occurrence of missing values is independent of both observed and unobserved data. The missing points occur due to sensor malfunctions or communication issues and are unrelated to the actual measurements. But for the MNAR scenario, the likelihood of absence is associated with some characteristics of the data itself. This introduces a potential bias because the missing data could represent certain patterns,

with missingness tending to cluster within specific groups, variables, or time periods.

1) *MCAR*: In this section, the performance of proposed method are compared with other baselines. Specially, several commonly used imputation methods based on statistics or machine learning are introduced, including multivariate imputation by chained equations (MICE), iterative SVD (SVD), spectral regularization (Spectral) and matrix completion (Matrix). Moreover, the state-of-the-art deep learning-based imputation models, namely GP-VAE [31], GI [32], Zoom2Net [26], GBIM [25], IGRM [33], CSDI [34] and MTSCI [35] are also used for comparison. The average RMSE of feature imputation in FD001 and FD003 with different missing ratios are illustrated in Fig. 5.

As shown in Fig. 5, the proposed method has the lowest RMSE on two subsets compared with other baselines. Although the results cannot fully reflect the downstream prediction ability, it provides some valuable insights into the ability of proposed method to capture dependencies and corrections between nodes and edges in semi-bipartite graph, thereby representing fragmented data reasonably.

To further display the ability of above methods for downstream RUL prognosis, the experimental results using ALSTM model [13] after data imputation or representation with different missing ratios are shown in Tab. II and III, respectively.

According to these results, the performance of 0-Impute is the worst of all baselines indubitably due to discarding valuable information from missing values. The statistics-based and machine learning methods (e.g., SVD, MICE, Spectral, Matrix) improve the downstream predictive accuracy to some extent when confronted with low-missing-rate samples, but the benefits are restricted with the missing ratio increasing. For generative methods, the advantages of GP-VAE and GI cannot be fully exploited when it is only used for imputing the missing values of industrial time-series data rather than image samples. Moreover, the hyperparameters of related network models are difficult to set. Zoom2Net, a Transformer-based imputation method, demonstrates competitive performance by capturing long-range dependencies, particularly under high missing ratios. However, it still falls short compared to the top-performing approaches. Graph-based methods such as GBIM and CDRM achieve moderate results, but their graph designs primarily capture variable-wise correlations while neglecting

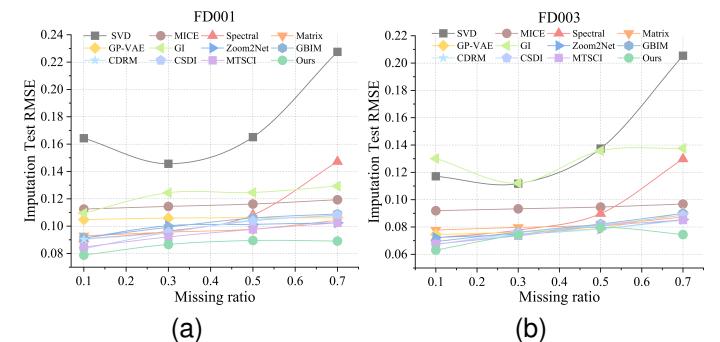


Fig. 5. The average RMSE of feature imputation with different missing ratios.(MCAR) a) FD001; b) FD003

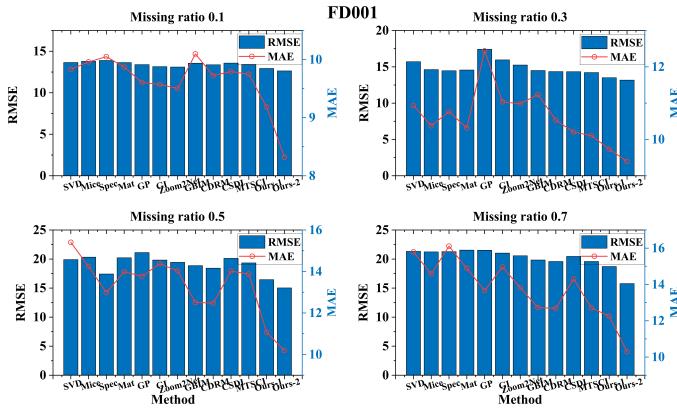


Fig. 6. The RMSE and MAE of prediction results using ALSTM model with different missing ratios in FD001 (MNAR)

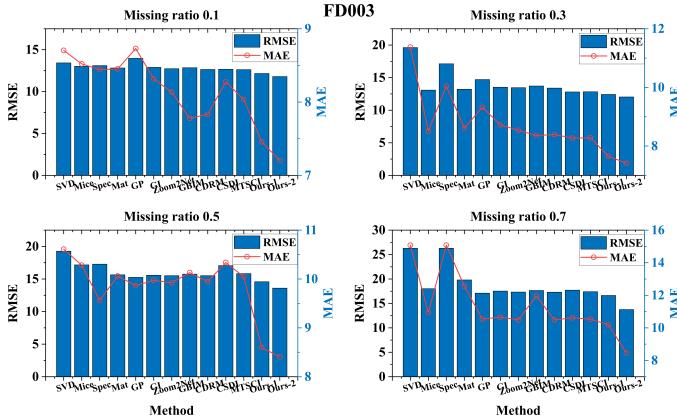


Fig. 7. The RMSE and MAE of prediction results using ALSTM model with different missing ratios in FD003 (MNAR)

long-term temporal structures, which limits their overall effectiveness. Among diffusion-based methods, CSDI outperforms most deep learning baselines, and MTSCI further improves on it by better modeling temporal dynamics. Nevertheless, there remains a significant gap compared to predictions made with fully observed data.

For proposed representation method, the satisfactory results are obtained on two subsets regardless of working conditions or missing ratio. Specially, the underlying information between neighboring nodes are propagated and aggregated with the aid of real and virtual edges. On this basis, the imputed values not only align with the patterns learned by the models, but also take into account the subtle differences within the multivariate time series. Moreover, the proposed model also can be served as the backbone module for involving in downstream RUL prediction directly, where some extra useful knowledge for RUL prediction can be learned other than the point estimations in missing scenarios. Thus, the end-to-end learning method is generally superior to the direct imputation method, especially in cases with high missing ratio.

2) **MNAR:** To simulate MNAR in real industrial settings, a corresponding number of bursts are selected to invalidate data, which is illustrated in Fig. 4. In the experimental setting, when the $x\%$ missing rate is desired, it results corresponding data points invalidated across multiple random bursts for each

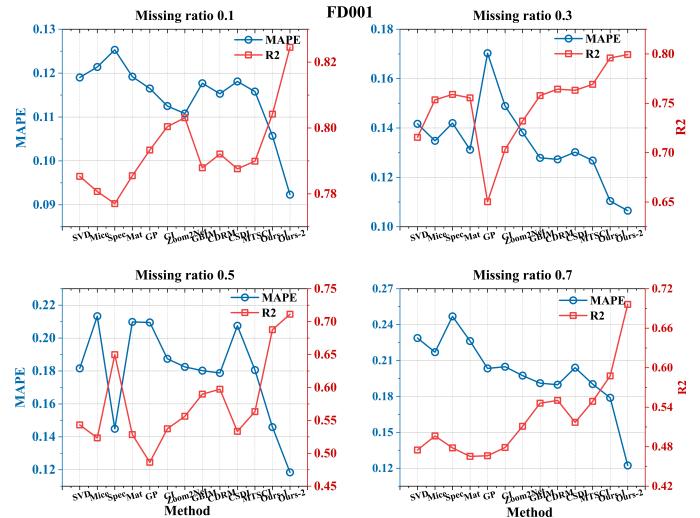


Fig. 8. The MAPE and R^2 of prediction results using ALSTM model with different missing ratios in FD001 (MNAR)

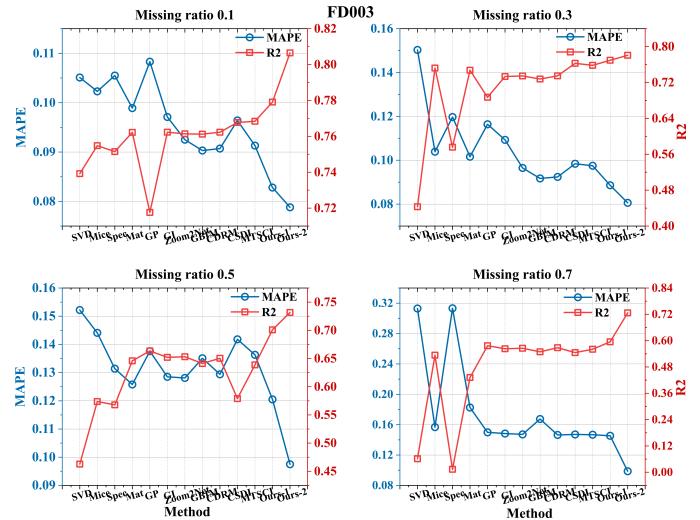


Fig. 9. The MAPE and R^2 of prediction results using ALSTM model with different missing ratios in FD003 (MNAR)

engine, which simultaneously considers specific continuous timestamps and fixed variables.

The experimental results using ALSTM with different missing ratios, including RMSE, MAE, MAPE and R^2 are shown in Fig. 6 - Fig. 9, respectively. These results reflect that traditional imputation methods still struggle with high missing ratios, while deep learning models also show limited effectiveness in handling MNAR scenarios for missing time-series data. This is largely due to their difficulty in accurately capturing the temporal dependencies and intricate variable correlations essential for reconstructing missing patterns. For proposed method, it still achieves satisfactory results on both subsets. This indicates that even when facing more complex MNAR scenarios, the proposed graph representation learning framework remains effective for structured missing data.

3) **Convergence analysis:** Taking the MNAR scenario as an example, the training and validation loss using an end-to-end fashion with different missing ratios is illustrated in Fig. 10. Lower missing rates result in smaller MSE losses, indicating

TABLE II
THE PREDICTION RESULTS USING ALSTM MODEL AFTER IMPUTATION OR REPRESENTATION IN FD001 (MCAR)

FD001	RMSE			Missing ratio			MAE			Missing ratio		
Method	Avg	0.1	0.3	0.5	0.7	Avg	0.1	0.3	0.5	0.7		
No missing												
0-Impute	24.065	19.084	24.069	25.766	27.343	16.605	12.849	16.652	18.063	18.858		
SVD	18.908	14.012	16.374	20.307	24.941	12.843	9.617	10.941	13.303	17.514		
MICE	17.489	13.614	16.067	18.181	22.096	11.942	9.707	11.213	12.335	14.513		
Spectral	19.108	13.882	16.611	21.007	24.933	12.714	9.523	11.261	13.771	16.301		
Matrix	18.161	13.811	16.081	19.578	23.172	12.182	9.506	11.035	13.038	15.149		
GP-VAE	18.511	13.488	15.264	20.813	24.478	12.396	9.422	10.351	13.623	16.191		
GI	18.834	13.346	15.346	21.383	25.261	12.694	9.287	10.414	14.231	16.847		
Zoom2Net	17.451	13.224	15.025	18.883	22.673	11.541	9.135	10.103	12.324	14.60		
GBIM	18.348	13.564	15.551	20.497	23.783	12.268	9.661	10.667	13.372	15.373		
CDRM	17.566	13.373	15.224	18.936	22.734	11.729	9.372	10.497	12.463	14.586		
CSDI	17.623	13.773	15.811	18.674	22.237	11.723	9.463	10.663	12.296	14.472		
MTSCI	17.443	13.831	15.724	18.247	21.973	11.576	9.516	10.492	12.135	14.161		
Ours-Impute then predict	14.958	12.995	13.879	15.766	17.195	10.293	9.024	9.684	10.695	11.769		
Ours-End to end	13.811	12.884	13.379	14.282	14.696	9.792	8.789	9.932	9.935	10.513		

TABLE III
THE PREDICTION RESULTS USING ALSTM MODEL AFTER IMPUTATION OR REPRESENTATION IN FD003 (MCAR)

FD003	RMSE			Missing ratio			MAE			Missing ratio		
Method	Avg	0.1	0.3	0.5	0.7	Avg	0.1	0.3	0.5	0.7		
No missing												
0-Impute	22.929	17.345	21.796	26.285	26.292	13.142	9.951	13.771	13.771	15.075		
SVD	15.491	12.091	13.005	16.008	20.859	9.287	7.413	8.004	9.767	11.966		
MICE	14.929	13.184	14.169	15.033	17.331	9.026	7.965	8.739	9.094	10.307		
Spectral	14.813	12.067	12.938	14.951	19.298	8.877	7.422	8.008	9.092	10.989		
Matrix	13.541	11.974	12.552	13.734	15.905	8.456	7.394	7.858	8.592	9.983		
GP-VAE	16.519	13.325	14.941	16.983	20.829	9.776	8.302	8.855	10.025	11.923		
GI	16.766	13.719	14.667	17.161	21.517	10.002	8.505	8.608	10.153	12.743		
Zoom2Net	15.060	13.127	14.632	15.135	17.346	9.005	8.124	8.575	9.104	10.219		
GBIM	14.681	13.006	13.641	14.971	17.105	8.856	7.911	8.362	8.976	10.175		
CDRM	14.474	12.784	13.363	14.324	17.427	8.821	7.763	8.236	8.735	10.552		
CSDI	14.089	11.996	12.724	14.111	17.528	8.587	7.364	7.811	8.613	10.563		
MTSCI	13.897	12.021	12.647	14.022	16.898	8.414	7.385	7.793	8.542	9.936		
Ours-Impute then predict	12.885	11.792	12.081	12.724	14.943	8.055	7.325	7.713	7.974	9.208		
Ours-End to end	12.352	11.625	11.874	12.569	13.341	7.723	7.222	7.497	7.901	8.273		

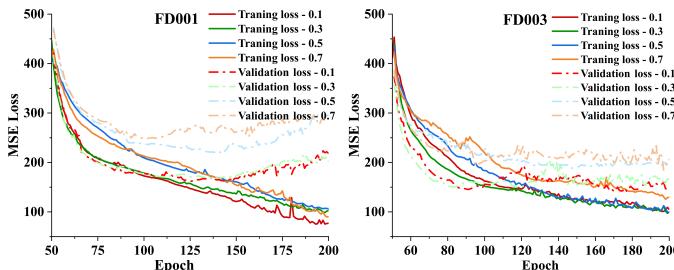


Fig. 10. The training and validation loss using ALSTM model with different missing ratios in an end-to-end fashion (MNAR)

better learning and generalization ability, while higher missing rates lead to increased losses and reduced predictive accuracy due to incomplete information. Despite this, from a global perspective, the end-to-end framework that cascades the proposed semi-bipartite graph representation learning model and the head model still achieves rapid convergence under different missing rates and demonstrates excellent generalization ability.

4) Computational efficiency analysis: To evaluate the computational efficiency of proposed method, the average running times of different deep learning-based methods are compared under different missing rates and missing patterns using the ALSTM head model. Except for Ours-End to end (Ours-2), whose running time corresponds to the model training of end-to-end RUL prediction over 200 epochs and final inference, the running times of other methods are measured based on a sequential two-stage process of data imputation followed by

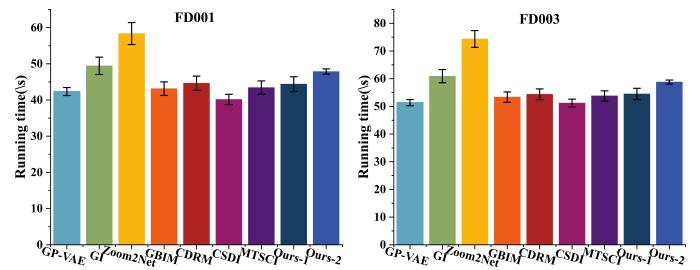


Fig. 11. The average running times of deep learning-based methods under varying missing rates and patterns using the ALSTM head model

downstream RUL prediction, with each stage also trained for 200 epochs and followed by final inference, as illustrated in Fig. 11.

It is found that Ours-Impute then predict (Ours-1) exhibits no significant disadvantage in computational time compared to mainstream deep learning-based data imputation methods, confirming its competitive efficiency during the imputation stage. Moreover, although Ours-End to end (Ours-2) employs an innovative end-to-end framework, its running time remains acceptable relative to its imputation version due to integrating a lightweight head model. This combination enables the end-to-end approach to bypass the traditional two-stage pipeline, thereby improving RUL prediction accuracy under missing data conditions directly while maintaining computational efficiency, which underpins its feasibility for deployment in real-world IIoT environments.

TABLE IV

THE PREDICTIVE RESULTS OF DIFFERENT GRAPH STRUCTURES USING ALSTM MODEL IN AN END-TO-END FASHION (MCAR)

Dataset	Graph structure	Indictor	Missing ratio			
			0.1	0.3	0.5	0.7
FD001	Bipartite	RMSE	13.152	13.811	16.075	16.932
		MAE	9.162	9.431	11.273	12.146
		MAPE	0.1079	0.1182	0.1489	0.1671
		R^2	0.8001	0.7807	0.7046	0.6615
FD003	Semi-Bipartite	RMSE	12.884	13.379	14.282	14.696
		MAE	8.789	9.232	9.935	10.513
		MAPE	0.1026	0.1117	0.1227	0.1294
		R^2	0.8145	0.7935	0.7651	0.7512

D. Ablation experiment

To further validate the effectiveness of proposed framework, some ablation experiments are conducted by systematically removing or modifying key components, such as different graph structures, time decay functions and head models. The impact of these changes on prediction accuracy under different missing rates will be analyzed in MCAR scenarios, providing deeper insights into the contribution of each component to the overall model performance.

1) *Different graph structures*: Compared with the bipartite graph, the semi-bipartite graph has virtual edges within one of the node sets. In this scheme, it can capture the dependency between timestamps and the correlation between features more efficiently. Tab. IV presents the predictive results of two different graph structures using ALSTM model in an end-to-end fashion. The semi-bipartite graph structure generally performs better than the bipartite graph, achieving lower errors and higher predictive accuracy. Besides, the semi-bipartite graph is also less sensitive to fragmented data as the missing ratio increases. These results highlight the importance of proposed semi-bipartite graph structure for robust RUL prediction.

2) *Different time decay functions*: As mentioned above, the time decay functions are used for capturing the dependency between timestamps. Fig. 11 presents the predictive results of proposed method under different time decay functions vividly. With the incorporation of virtual edges into the bipartite graph, some prior knowledge is introduced to facilitate model training. Three commonly used time decay functions effectively characterize the temporal correlations of timestamps, thereby ensuring that the performance of proposed method remains robust and stable overall.

3) *Different head models*: Moreover, the prediction curves of proposed representation learning framework for a given engine under prevalent head modules, including DCNN [11], BDLSTM [12], ALSTM [13], ABGRU [14], ASTGCN [36] are illustrated in Fig. 12 and Fig. 13.

As can be seen, the proposed representation learning model serves as the backbone module for conducting end-to-end learning. These figures illustrate that our proposed method

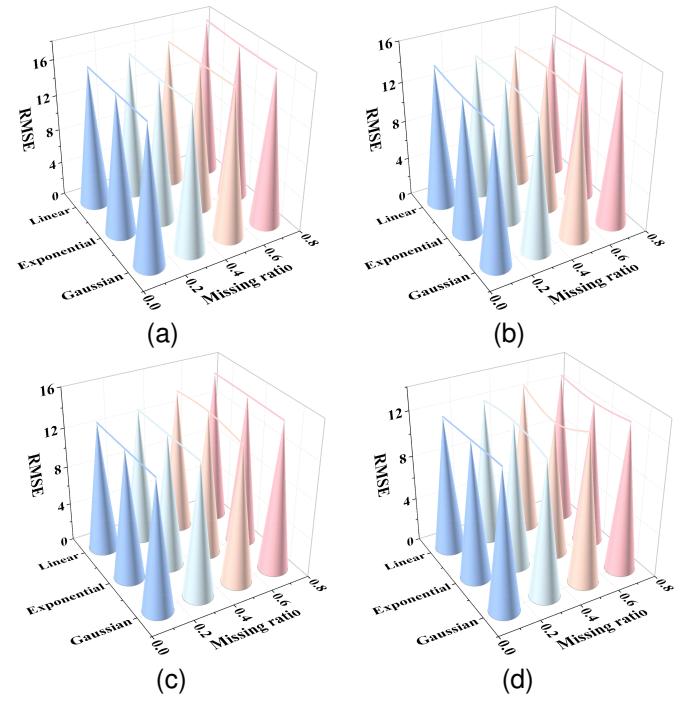


Fig. 12. The prediction results under different time decay functions. a) FD001-Impute then predict; b) FD001-End to end learning c) FD003-Impute then predict; d) FD003-End to end learning

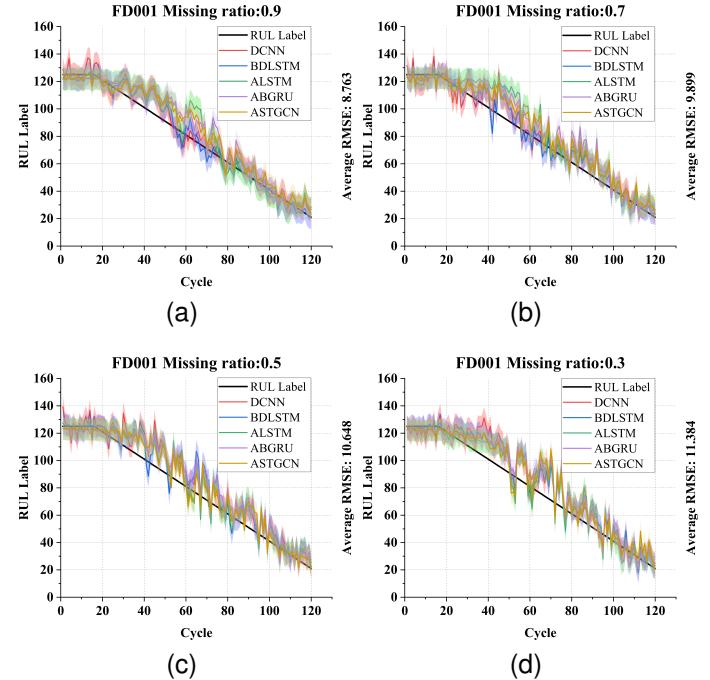


Fig. 13. The prediction curves of the proposed model under different head modules for the 92th engine in subset FD001

performs steadily under some prevalent predictive models in different working conditions or missing rates, which also proves the scalability of the proposed method.

In summary, the experiments conducted on the C-MAPSS dataset indicate that the proposed method outperforms existing imputation techniques in terms of generalization and

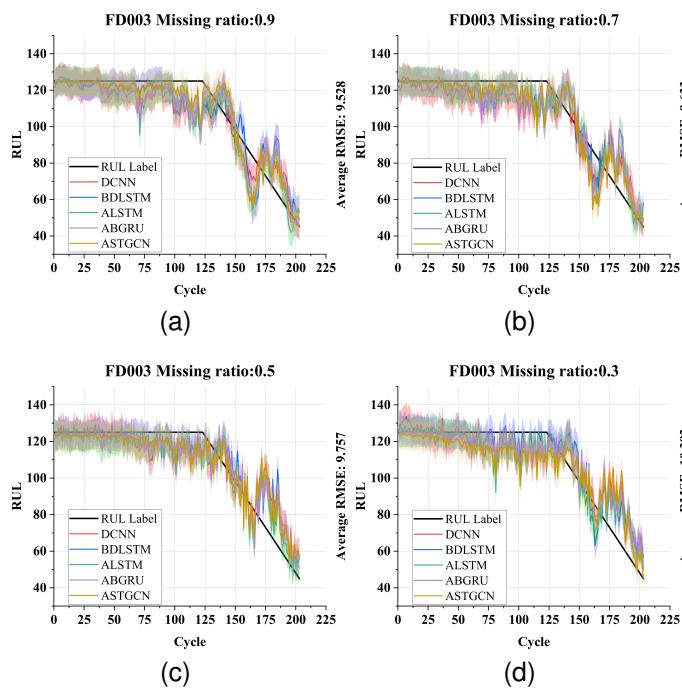


Fig. 14. The prediction curves of the proposed model under different head modules for the 26th engine in subset FD003

robustness, highlighting its effectiveness in improving predictive maintenance in IIoT-enabled systems while reducing unexpected failures and maintenance costs.

V. CONCLUSION

In this study, a novel representation learning method based on the semi-bipartite graph is proposed for RUL prognosis with missing values. Firstly, the fragmented data with observations and features are regarded as the semi-bipartite graph, where the spatiotemporal corrections in multivariate time-series are captured efficiently. Following that, the node and edge embeddings in the semi-bipartite graph are learned through the GNN architecture, thereby achieving graph-level representations. Finally, two different learning methods for handling missing values, namely direct imputation and end-to-end learning, are provided for downstream RUL prediction with the aid of learned representations. The proposed method outperforms the existing imputation techniques and exhibits robustness across different missing patterns when evaluated on the C-MAPSS dataset, which offers a bright prospect for RUL prognosis with fragmented data.

The following points are challenging for further research. 1) This paper only considers numerical data. However, the multi-modal data, including signals, images and videos are becoming common in real industrial. How to integrate the multi-model information in graph-based representation learning is the next problem worth investigating. 2) New data distribution could gradually shift away from that of historical data when confronted with performance degradation or non-stationary processes for complex equipment. How to develop more effective representation learning method for incremental learning-based RUL prognosis with missing values could be

the focus of future research. 3) In real-world industrial applications, missing values are frequently accompanied by noise or corrupted measurements. How to design robust imputation or representation learning methods that can accurately handle noisy or low-quality data remains an important challenge for future study.

REFERENCES

- [1] L. Wang, Y. Chen, X. Zhao and J. Xiang, "Predictive Maintenance Scheduling for Aircraft Engines Based on Remaining Useful Life Prediction," *IEEE Internet Things J.*, vol. 11, no. 13, pp. 23020-23031, Jul. 2024.
- [2] H. Li, Z. Zhang, T. Li and X. Si, "A review on physics-informed data-driven remaining useful life prediction: Challenges and opportunities," *Mech. Syst. Sig. Process.*, vol. 209, Mar. 2024, Art. no. 111120.
- [3] B. Chen P. Li, G. Wan, F. Chen and Q. Liu, "A Method for Predicting the Remaining Useful Life of Aircraft Engines Based on NLSTM and Feature Optimization Strategy," *IEEE Internet Things J.*, vol. 11, no. 22, pp. 36410-36419, Nov. 2024.
- [4] L. Cai, H. Yin, J. Lin and D. Zhao, "An Update-Strategy-Based Gaussian Process Regression Method for Aeroengines Fault Prediction," *IEEE Trans. Ind. Informat.*, vol. 20, no. 2, pp. 1941-1951, Feb. 2024.
- [5] J. He, C. Wu, W. Luo, C. Qian and S. Liu, "Remaining Useful Life Prediction and Uncertainty Quantification for Bearings Based on Cascaded Multiscale Convolutional Neural Network," *IEEE Trans. Instrum. Meas.*, vol. 73, Dec. 2023, Art. no. 3506713.
- [6] D. Yuan, T. Luo, C. Gu and K. Zhu, "The Cyber-Physical System of Machine Tool Monitoring: A Model-Driven Approach With Extended Kalman Filter Implementation," *IEEE Trans. Ind. Informat.*, vol. 19, no. 9, pp. 9576-9585, Sep. 2024.
- [7] J. Zhou, J. Qi, D. Chen and Y. Qin, "Continuous Remaining Useful Life Prediction by Self-Guided Attention Convolutional Neural Network and Memory Consciousness Adjustment," *IEEE Internet Things J.*, vol. 11, no. 19, pp. 31947-31958, Oct. 2024.
- [8] J. Ma *et al.*, "A multi-phase Wiener process-based degradation model with imperfect maintenance activities," *Reliab. Eng. Syst. Saf.*, vol. 232, Apr. 2023, Art. no. 109075.
- [9] Z. Xu, Y. Guo and J. H. Saleh, "Accurate Remaining Useful Life Prediction With Uncertainty Quantification: A Deep Learning and Non-stationary Gaussian Process Approach," *IEEE Trans. Reliab.*, vol. 71, no. 1, pp. 443-456, Mar. 2022.
- [10] M. G. Alfarizi, B. Tajiani, J. Vatin and S. Yin, "Optimized Random Forest Model for Remaining Useful Life Prediction of Experimental Bearings," *IEEE Trans. Ind. Informat.*, vol. 19, no. 6, 7771-7779, Jun. 2023.
- [11] X. Li, Q. Ding, and J-Q. Sun, "Remaining useful life estimation in prognostics using deep convolution neural networks," *Reliab. Eng. Syst. Saf.*, vol. 171, pp. 1-11, Apr. 2018.
- [12] J. Zhang *et al.*, "Long short-term memory for machine remaining life prediction," *J. Manuf. Syst.*, vol. 48, pp. 78-86, 2018.
- [13] Z. Chen *et al.*, "Machine Remaining Useful Life Prediction via an Attention-Based Deep Learning Approach," *IEEE Trans. Ind. Electron.*, vol. 68, no. 3, pp. 2521-2531, Mar. 2021.
- [14] R. Lin, H. Wang, M. Xiong, Z. Hou, and C. Che, "Attention-based gate recurrent unit for remaining useful life prediction in prognostics," *Appl. Soft Comput.*, vol. 143, Aug. 2023, Art. no. 110419.
- [15] L. Ren *et al.*, "A Data-Driven Auto-CNN-LSTM Prediction Model for Lithium-Ion Battery Remaining Useful Life," *IEEE Trans. Ind. Informat.*, vol. 17, no. 5, 3478-3487, May. 2021.
- [16] Z. Ye and J. Yu, "A Selective Adversarial Adaptation Network for Remaining Useful Life Prediction of Machines Under Different Working Conditions," *IEEE Sensors J.*, vol. 17, no. 1, pp. 62-71, Mar. 2023.
- [17] J. Zhou, Y. Qin, J. Luo, S. Wang and T. Zhu, "Dual-Thread Gated Recurrent Unit for Gear Remaining Useful Life Prediction," *IEEE Trans. Ind. Informat.*, vol. 19, no. 7, pp. 8307-8318, Jul. 2023.
- [18] K. Feng *et al.*, "Digital twin-driven intelligent assessment of gear surface degradation," *Mech. Syst. Sig. Process.*, vol. 186, Mar. 2023, Art. no. 109896.
- [19] L. Cui, Y. Xiao, D. Liu, and H. Han, "Digital twin-driven graph domain adaptation neural network for remaining useful life prediction of rolling bearing," *Reliab. Eng. Syst. Saf.*, vol. 245, May. 2024, Art. no. 109991.
- [20] K. Feng, H. Xiao, J. Zhang, and Q. Ni, "A digital twin methodology for vibration-based monitoring and prediction of gear wear," *Wear*, Feb. 2025, Art. no. 205806.

- [21] J-Y. Wu *et al.*, "A joint classification-regression method for multi-stage remaining useful life prediction," *J. Manuf. Syst.*, vol. 58, pp. 109-119, 2021.
- [22] Y. Peng, Y. Wang and Y. Shao, "New Data Augmentation-Driven RUL Prognosis Approach for Cumulative Damage Model Using Incomplete Observations," *IEEE Trans. Instrum. Meas.*, vol. 71, Dec. 2021, Art. no. 3502112.
- [23] X. Liu and Z. Zhang, "A Two-Stage Deep Autoencoder-Based Missing Data Imputation Method for Wind Farm SCADA Data," *IEEE Sensors J.*, vol. 21, no. 9, pp. 29991-30006, Dec. 2023.
- [24] Z. Lv *et al.*, "Multi-feature generation network-based imputation method for industrial data with high missing rate," *Exper. Syst. Appl.*, vol. 227, Oct. 2023, Art. no. 120229.
- [25] S. Jeong *et al.*, "A novel graph-based missing values imputation method for industrial lubricant data," *Comput. Ind.*, vol. 150, Sep. 2023, Art. no. 103937.
- [26] F. Gong *et al.*, "Zoom2Net: Constrained Network Telemetry Imputation," in *Proc. ACM SIGCOMM Conf.*, 2024, pp. 764-777.
- [27] L. Peng *et al.*, "GRLC: Graph Representation Learning With Constraints," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 6, pp. 8609-8622, Jun. 2024.
- [28] W. Hamilton *et al.*, "Inductive Representation Learning on Large Graphs," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 1024-1034.
- [29] J. You *et al.*, "Handling Missing Data with Graph Representation Learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 19075-19087.
- [30] W. Hamilton *et al.*, "Inductive Representation Learning on Large Graphs," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 1024-1034.
- [31] V. Fortuin, D. Baranchuk, G. Ratsch, and S. Mandt, "GP-VAE: Deep probabilistic time series imputation," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2020, pp. 1651-1661.
- [32] M. Kachuee *et al.*, "Generative Imputation and Stochastic Prediction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 3, pp. 1278-1288, Mar. 2022.
- [33] M. Chen *et al.*, "CDRM: Causal disentangled representation learning for missing data," *Knowl.-Based Syst.*, vol. 299, Sep. 2024, Art. no. 112079.
- [34] Y. Tashiro *et al.*, "CSDI: Conditional score-based diffusion models for probabilistic time series imputation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, pp. 24804-24816.
- [35] J. Zhou *et al.*, "MTSCI: A Conditional Diffusion Model for Multivariate Time Series Consistent Imputation," in *Proc. ACM Int. Conf. Inf. Knowl. Manag. (CIKM)*, 2024, pp. 3474-3483.
- [36] X. Zhang *et al.*, "Spatial-temporal dual-channel adaptive graph convolutional network for remaining useful life prediction with multi-sensor information fusion," *Adv. Eng. Informat.*, vol. 57, Aug. 2023, Art. no. 102120.