Spatio-temporal Graph Neural Network for Fault Diagnosis Modeling of Industrial Robot

1st Wenbo Xiao
Department of Computer Science and Technology
Guangdong University of Technology
Guangzhou, China
0009-0007-5673-9305

2nd Yuwei Wan
Department of Mechanical Engineering
Cardiff University
Cardiff, UK
WanY8@cardiff.ac.uk

3rd Zhuowei Wang
Department of Computer Science and Technology
Guangdong University of Technology
Guangzhou, China
0000-0001-6479-5154

4th Chong Chen
Department of Computer Science and Technology
Guangdong University of Technology
Guangzhou, China
0000-0003-2800-4647

***Abstract*—Industrial robots play a critical role in modern industrial production. They are widely used in tasks requiring high precision and efficiency, such as manufacturing, assembly, and material handling. However, since industrial robots often operate in complex and changing environments, long-term use inevitably leads to issues such as component aging, wear, or other potential faults. These faults not only reduce production efficiency but may also cause equipment damage, production interruptions, or even safety risks. Industrial robot systems are typically composed of multiple highly interconnected components and sensors. Graph Neural Networks (GNNs), which can effectively model multivariable data, have been widely applied to modeling and analyzing such systems. However, different methods of graph construction vary significantly in their ability to capture system structures, dynamic relationships, and multivariable interactions. The impact of these differences on downstream fault diagnosis tasks remains an area that requires further research. To address these challenges, this study proposes a GNN-based fault diagnosis framework to demonstrate the practical effects of different graph construction methods. First, we transformed the state monitoring data of industrial robots into three different graph structures using KNN, radius, and path-based methods. Then, we used a graph attention network to capture the spatial dependencies among various variables. At the same time, a parallel encoder with diagonal masking self-attention (DMSA) was designed to model temporal dependencies. A spatiotemporal attention module was then applied to extract both spatial and temporal features. Finally, the type of fault present in the data was determined. Experimental studies based on real-world industrial robot datasets show that different graph construction methods significantly influence fault diagnosis accuracy. The proposed framework achieved diagnostic accuracies of 87.11%, 88.85%, and 92.68% under the three graph construction methods, respectively.**

***Index Terms*—Industrial robots, Fault diagnosis, Graph Neural Networks (GNNs), Spatiotemporal features**

Yuwei Wan is with the Department of Mechanical Engineering, School of Engineering, Cardiff University, Cardiff, CF24 3AA, UK

Wenbo Xiao, Zhuowei Wang, and Chong Chen are with the School of Computer Science and Technology, Guangdong University of Technology, Guangzhou 510006, China (Corresponding author: Chong Chen; e-mail: chenc2021@gdut.edu.cn).

I. Introduction

Industrial robots are widely used in modern manufacturing. They significantly enhance production automation with their high precision, efficiency, and reliability [1]. However, as robotic systems become more complex, the likelihood of faults during operation increases. This is particularly true in complex industrial environments, where mechanical wear, electrical interference, and environmental changes can affect robot components. Faults not only reduce performance but can also cause production delays, economic losses, or safety risks [2]. Ensuring the stability and reliability of industrial robots has thus become a critical issue [3].

Fault diagnosis techniques have been developed to address this problem. These techniques use data-driven methods to detect anomalies early and locate faults accurately [4]. They help reduce downtime and maintenance costs while improving operational efficiency and safety through real-time monitoring and predictive maintenance. However, with the growing trend toward intelligent and networked robots, fault diagnosis faces new challenges [5]. These include integrating multi-source data, ensuring real-time performance, and improving diagnostic accuracy. Solving these issues is essential for enhancing the stability of robots, extending their lifespan, and advancing smart manufacturing [6].

Existing machine learning algorithms have been widely applied to fault diagnosis in industrial robots. Deep learning methods, in particular, excel at extracting spatial and temporal features [7], capturing hidden patterns and relationships from complex sensor data. Yet, due to the diversity and complexity of robot faults, current methods face limitations in handling multi-source heterogeneous data and modeling spatiotemporal dependencies. For example, convolutional neural networks (CNNs) are effective at capturing local spatial features, but recurrent neural networks (RNNs) struggle with gradient vanishing issues when modeling long-term dependencies in time-series data [8]. Thus, there is a pressing need for a diagnostic framework that can integrate various feature representation capabilities.

Recently, Graph Neural Networks (GNNs) have gained attention for their ability to model complex graph-structured data [9]. Combining GNNs with encoder frameworks can fully utilize the topological relationships and spatiotemporal dynamics within industrial robot systems, leading to more accurate fault diagnosis. Additionally, the diversity of sensor distributions and component relationships in robots places high demands on graph construction methods [10]. Studying different graph construction designs and their impact on diagnostic performance not only helps optimize diagnostic frameworks but also offers new insights into multi-source data integration [11].Therefore, exploring effective graph construction strategies is crucial for advancing fault diagnosis in industrial robots, particularly in capturing the intricate interplay between spatial and temporal dependencies.

To address these key challenges, this study explores a fault diagnosis method for industrial robots based on GNNs. Three different graph construction methods are employed: K-Nearest Neighbors (KNN), radius-based, and path-based. These methods transform raw data into three types of graph structures. A Bidirectional Graph Attention Network is then used to capture spatial dependencies among variables. A parallel encoder, integrated with a Diagonal Masked Self-Attention (DMSA) mechanism [12], is designed to model temporal correlations.

A spatiotemporal attention module is designed as the core component of the proposed framework, aiming to effectively capture the complex dependencies between sensor nodes and time steps. In this module, spatial features extracted by the Bidirectional Graph Attention Network (Bi-GAT) are used as query vectors, while temporal features encoded by the Transformer-based encoder serve as keys and values. This design choice is motivated by the assumption that fault patterns are primarily localized in the spatial domain and are influenced by interactions among different sensor nodes. Using spatial features as queries enables the model to actively attend to relevant temporal dynamics that may vary across locations. In contrast, representing temporal features as keys and values allows the model to provide context-aware information over time. By integrating spatial queries with temporal context through self-attention, the model learns joint spatiotemporal representations that improve the accuracy and robustness of fault diagnosis.

The remaining sections of this paper are organized as follows: Section 2 reviews the application of GNNs in industrial fault diagnosis. Section 3 introduces the proposed GNN-based fault diagnosis method. Section 4 reports the experimental results and analysis. Section 5 discusses the findings, and Section 6 concludes the study.

II. Related Work

Traditional methods for fault diagnosis in industrial systems often rely on deep neural networks (DNNs) and principal component analysis (PCA) to extract temporal features from collected signals [13]. Wen et al. [14] proposes a hybrid fault diagnosis method based on ReliefF, PCA, and DNNs. First, the ReliefF method is used to select fault features and reduce the data dimensions. Then, PCA is applied to further reduce the dimensions, removing data redundancy and improving diagnosis accuracy. Finally, the ReliefF-PCA-DNN model is built and optimized for fault diagnosis using real field data. However, these methods typically overlook the spatial relationships between samples or sensors. To address this, researchers have explored convolutional neural networks (CNNs) to capture spatial features. Despite this, CNNs struggle to encode the latent relationships between samples or sensors into spatial representations due to their inherent limitations [15]. Consequently, there is a pressing need for more advanced methodologies that can effectively model both spatial and temporal dependencies to enhance fault diagnosis performance in complex industrial systems.

Recurrent neural networks (RNNs) are effective for handling time-series data but face challenges like vanishing or exploding gradients when capturing long-term dependencies. Additionally, their sequential computation process makes them inefficient for large-scale industrial data, limiting their application in real-time fault diagnosis [16].

Graph neural networks (GNNs) offer a promising alternative by providing a framework to capture both spatial and temporal information. In GNNs, edges represent connections or relationships between samples or sensors, while graph convolution aggregates information from neighboring nodes. This enables GNNs to integrate relational information effectively, making them highly suitable for advancing fault diagnosis and prediction [17].

Graph convolutional networks (GCNs) are particularly effective in capturing local structures and relationships within systems by aggregating information from adjacent nodes. However, their local aggregation mechanism makes it difficult to detect subtle signal features under varying load conditions. As a result, GCNs may not fully extract global features, limiting their performance in complex fault diagnosis tasks. To address this, Chen et al. [18] proposed a GCN-based fault diagnosis method that combines available measurement data and prior knowledge. First, fault prediction is performed using structural analysis (SA), and the results are converted into an association graph. This graph, along with the measurement data, is fed into a GCN model, which adjusts the influence of measurement data and prior knowledge using weighted coefficients.

Unlike GCNs, graph attention networks (GATs) use attention mechanisms to dynamically adjust the weights of adjacent nodes. This allows them to flexibly identify and focus on important nodes and features. Tang et al. [19] proposed a fault diagnosis method combining GATs with semi-supervised conditional random fields (CRFs). This approach leverages the strengths of CRFs and GATs to model label dependencies and learn object representations, enabling semi-supervised fault diagnosis with limited labeled data.

For irregular time-series data with unequal intervals between observations, Zhang et al. [20] proposed a graph neural network model named RAINDROP. This model constructs a sensor-level graph for each sample and introduces a customized message-passing mechanism to capture temporal dependencies among sensors. The resulting node embeddings are then aggregated into fixed-length representations for downstream tasks such as fault diagnosis.While RAINDROP demonstrates effectiveness in handling irregular sampling and local temporal patterns, it shows limitations in scenarios involving dynamic sensor networks. Specifically, its sample-wise graph construction leads to increased computational overhead and limited scalability when dealing with large-scale or continuously evolving sensor configurations.

While existing GNN-based methods can capture spatial relationships, most rely on a single graph construction approach. This limits their ability to reflect diverse spatial characteristics in data. In complex industrial environments, varying operating conditions and differences in sensor configurations make single graph structures insufficient to adapt to dynamic data patterns. This can lead to reduced performance and lower diagnostic accuracy across different conditions. Furthermore, current models often focus on a single feature dimension, such as time or space, failing to fully capture the complex behaviors and dynamic changes in industrial robots under fault conditions [21].

To improve the accuracy and robustness of fault diagnosis for industrial robots, there is a need for models that integrate multiple graph construction methods and effectively combine spatial and temporal features. Such models can address the shortcomings of existing approaches, providing more precise and reliable fault diagnosis solutions in complex and dynamic industrial environments.

III. METHODOLOGY

This section provides a detailed description of the proposed industrial fault diagnosis method. As shown in Figure 1, the method involves several key steps.

First, data collection and preprocessing are performed. Fault scenarios are simulated by replacing functional motors and gearboxes in industrial robots with faulty ones. Voltage sensors are used to directly collect voltage data from the industrial robot. The collected data is then used to construct three types of graph data: KNN, radius, and path graphs.Next, the graph data and raw data are fed into the GAT layer and the encoder layer. The GAT layer outputs node-level hidden features, while the encoder layer produces time-level hidden features.Finally, a spatiotemporal graph attention mechanism is applied to fuse the node features and time features. This generates the final fault diagnosis results.

A. Graph Construction Method

In this section, we present three distinct methods for constructing graphs from data: K-Nearest Neighbors (KNN), radius-based, and path-based approaches. Each construction method captures different aspects of local or global data similarity, leading to distinct graph topologies. KNN and radius-based graphs typically emphasize local connections, while path-based methods may incorporate more global context. As a result, the choice of construction method not only influences the expressive power of the graph but also affects the computational complexity of the model. Simpler graphs may reduce training time but risk missing important structural patterns, whereas more complex graphs can better capture data relationships at the cost of increased computational burden. By comparing all three methods, we aim to evaluate their respective advantages in revealing informative structures within the data, and to assess how these structural differences influence the overall diagnostic accuracy and efficiency.

We begin with the KNN method, which is widely used due to its flexibility and effectiveness in capturing local data structures.

K-Nearest Neighbors Graph Construction:In the KNN-based graph, each node is connected to its k nearest neighbors based on feature similarity. This method is effective for capturing local patterns in the data and is particularly suitable when the underlying structure is dense and smooth. In this method, nodes represent data samples, and edges indicate similarity between samples. For each sample, the method selects its K nearest samples to form graph edges. KNN graph construction effectively captures local structures between samples and models complex relationships in high-dimensional spaces.

Given $X = \{x_1, x_2, \ldots, x_N\}$, where $x_i \in \mathbb{R}^d$ is a feature vector of dimension d, the distance $d(x_i, x_j)$ between x_i and any other sample x_j $(j \neq i)$ is computed as:

$$d(x_i, x_j) = \|x_i - x_j\| \tag{1}$$

Here, $\|\cdot\|$ represents the Euclidean distance. Next, the K samples closest to x_i are selected, defined as:

$$KNN(x_i) = \{x_j \mid j \in \text{topK}(d(x_i, x_1), \ldots, d(x_i, x_N))\} \tag{2}$$

The adjacency matrix A of the KNN graph is defined as:

$$A_{ij} = \begin{cases} 1, & \text{if } x_j \in KNN(x_i) \text{ or } x_i \in KNN(x_j) \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

In other words, an element $A_{ij} = 1$ indicates an edge between x_i and x_j.

Radius-Based Graph Construction: The radius-based graph connects nodes that fall within a predefined Euclidean distance threshold. This method emphasizes spatial proximity and is sensitive to variations in data density. It performs well when meaningful relationships are expected to occur within localized regions but may fail to capture connections in sparse or unevenly distributed data. Each node is connected only to other nodes within a specified radius. By selecting an appropriate radius r, the method captures local relationships between data samples. This approach is suitable for dense datasets with similar samples.

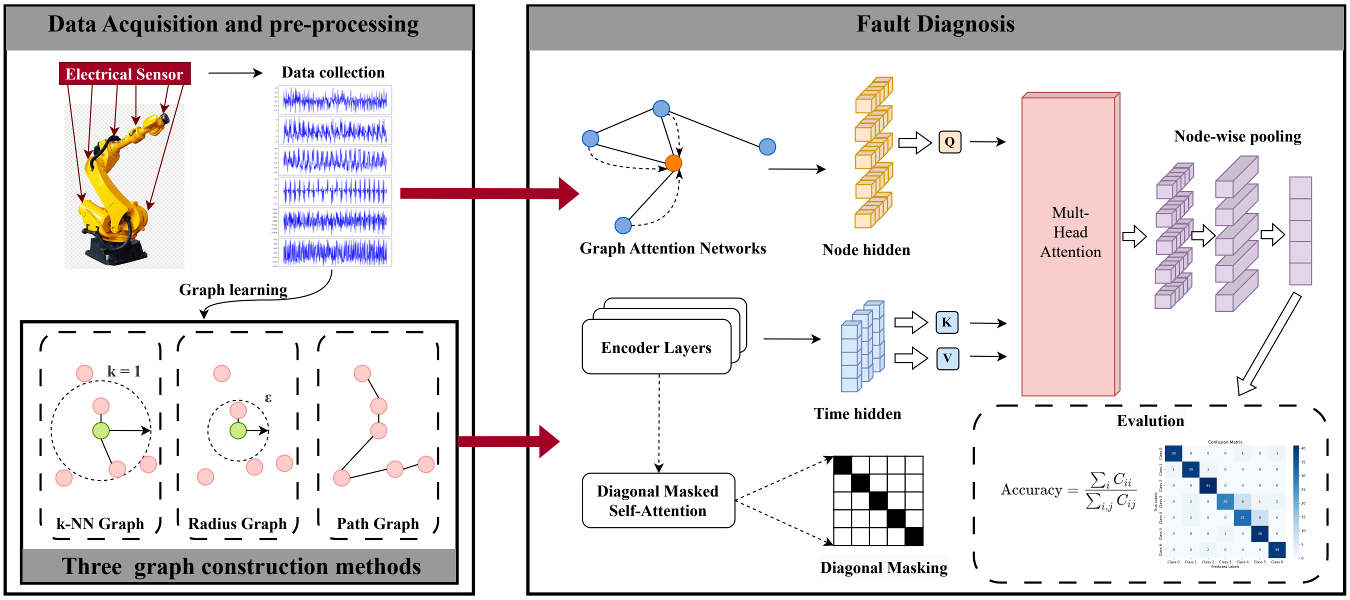

Fig. 1: The overall framework.

Given a dataset $X = \{x_1, x_2, \ldots, x_N\}$, the adjacency matrix A for the radius graph is defined as follows:

$$A_{ij} = \begin{cases} 1, & \text{if } \|x_i - x_j\| \leq r \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

Here, r is the fixed radius that represents the maximum distance for connecting two samples. $\|x_i - x_j\|$ denotes the Euclidean distance between samples x_i and x_j.

Path-Based Graph Construction: The path-based graph is constructed based on the known physical or functional topology of the system, such as the wiring or sensor layout in industrial equipment. Unlike the other two methods, it does not rely on feature similarity or distance but instead encodes domain-specific prior knowledge. This makes it particularly effective in scenarios where the spatial arrangement or signal flow is crucial to understanding system behavior. Unlike KNN or radius-based graphs, the Path Graph connects nodes sequentially to form a chain-like structure. Each node is linked to its neighboring nodes through edges. This method captures the order dependency between nodes and is commonly used for time-series or sensor data processing.

In industrial robot fault diagnosis, the Path Graph is constructed based on sensor data. Each node represents a sensor measurement, while edges represent the relationships between neighboring sensors. This structure captures the spatiotemporal dependencies between adjacent sensors, making it an effective input for graph neural networks. It supports feature learning and fault prediction.

For input data $X = \{x_1, x_2, \ldots, x_N\}$, the connections between nodes are based on their sequence. An edge exists between nodes x_i and x_j if they are adjacent. The adjacency matrix A is constructed as follows:

$$A_{ij} = \begin{cases} 1, & \text{if } j = i + 1 \text{ or } i = j + 1 \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

This structure forms a simple chain with $N-1$ edges, where each edge connects two neighboring nodes.

B. Fault Diagnosis Method

This section presents our fault diagnosis framework that combines three key components: a Graph Attention Network (GAT), an encoder with diagonal masked self-attention, and a spatiotemporal attention mechanism. These components work together to process and analyze sensor data from industrial robots. The GAT serves as our primary feature extractor, capturing both local and global patterns in the data. By combining these three components, our method can effectively identify subtle fault patterns while maintaining robustness against noise in real-world industrial settings. We first describe the GAT component, which forms the foundation of our diagnostic framework.

Graph Attention Network (GAT) Processing: To effectively capture the temporal patterns in time-series data and the complex relationships between nodes, this study uses a Graph Attention Network (GAT) as the core module. GAT employs a learnable attention mechanism to dynamically assign weights to interactions between nodes. This allows the model to preserve local information while extracting global features, enabling high-quality feature representation for downstream tasks.

In the graph attention layer, attention scores are calculated based on concatenated node features. For nodes x_i and x_j in the adjacency matrix A, the attention score is defined as:

$$\alpha_{ij} = \text{softmax}\left(\text{LeakyReLU}\left(\boldsymbol{a}^\top [W_1 \boldsymbol{v}_i \parallel W_1 \boldsymbol{v}_j]\right)\right) \tag{6}$$

Here, $[\boldsymbol{h}_i \parallel \boldsymbol{h}_j]$ denotes the concatenation of features from nodes $\boldsymbol{v}_i$ and $\boldsymbol{v}_j$. The vector $\boldsymbol{a}^\top$ and matrix W_1 are learnable parameters, while the $\mathrm{LeakyReLU}$ function introduces non-linearity. The $\mathrm{softmax}$ function ensures that the attention scores are normalized.

The attention scores are used to weight and aggregate the features of neighboring nodes. The updated representation of node $\boldsymbol{v}_i$, denoted as $\boldsymbol{h}'_i$, is computed as:

$$\boldsymbol{h}'_i = \sigma\left(\sum_{j \in \mathcal{N}(i)} \alpha_{ij} \boldsymbol{h}_j\right) \tag{7}$$

where $\mathcal{N}(i)$ represents the set of neighbors for node $\boldsymbol{v}_i$, and $\sigma(\cdot)$ is the ReLU activation function. After updating all nodes, the resulting feature matrix H_0 is:

$$H_0 = \begin{bmatrix} \boldsymbol{h}'_1 \\ \boldsymbol{h}'_2 \\ \vdots \\ \boldsymbol{h}'_N \end{bmatrix} \in \mathbb{R}^{N \times d} \tag{8}$$

In the final layer, the output of the graph neural network is given by:

$$\hat{H}_1 = \mathrm{LeakyReLU}\left(W_2 H_0 + b\right) \tag{9}$$

where W_2 is a learnable parameter matrix, and b is the bias term.

Encoder with Diagonal Masked Self-Attention (DMSA): In the standard Transformer encoder, the self-attention mechanism allows each position in the sequence to attend to all others, including itself. However, allowing self-attention may lead to redundancy and potentially cause overfitting, especially in time-series tasks where each time step ideally aggregates contextual information from surrounding steps rather than reinforcing its own features. To mitigate this, we introduce a Diagonal Masked Self-Attention mechanism that explicitly prevents each time step from attending to itself.

The masking is applied during the computation of the attention score matrix. First, the raw attention score between query vector $\boldsymbol{q}_i$ and key vector $\boldsymbol{k}_j$ is computed as:

$$e_{i,j} = \frac{\boldsymbol{q}_i \cdot \boldsymbol{k}_j}{\sqrt{d_k}}. \tag{10}$$

Next, diagonal elements of the score matrix (i.e., where $i = j$) are set to a large negative value (e.g., -10^9) to effectively mask self-attention:

$$e_{i,j} = \begin{cases} \frac{\boldsymbol{q}_i \cdot \boldsymbol{k}_j}{\sqrt{d_k}}, & \text{if } i \neq j, \\ -10^9, & \text{if } i = j. \end{cases} \tag{11}$$

Finally, the masked attention scores are normalized using the softmax function to obtain the attention weights:

$$\widetilde{A}_{i,j} = \frac{\exp(e_{i,j})}{\sum_{k=1}^{L} \exp(e_{i,k})}. \tag{12}$$

Through this design, each time step aggregates information from all other steps except itself. This encourages the model to focus on temporal dependencies across different positions and helps enhance generalization by reducing redundancy in the learned representations.

In the encoder layer, the input data $\mathbf{X} \in \mathbb{R}^{L \times d}$ and the mask $M \in \{0,1\}^{L \times d}$ are concatenated. The concatenated input is then passed through a linear transformation and position encoding to obtain intermediate representations:

$$H_1 = \mathrm{Linear}([\mathbf{X}; \mathbf{M}]) + B + PE \tag{13}$$

where $[\mathbf{X}; \mathbf{M}]$ denotes the concatenation of the input data and mask, B is a learnable bias term, and PE represents position encoding.

The intermediate representation H_1 is then fed into the DMSA module. The module performs multi-head attention, followed by ReLU activation, a linear transformation, and bias addition. This produces the updated hidden representation:

$$\hat{H}_2 = \mathrm{DMSA}(H_1) + \mathrm{ReLU}(\mathrm{Linear}(H_1 + B)) \tag{14}$$

Spatiotemporal Attention: In fault diagnosis tasks, an attention mechanism is used to dynamically focus on the key features of each module. This approach enhances the ability to model interactions between features, improving diagnostic accuracy. The outputs of the GAT layer and the encoder layer are denoted as $\hat{H}_1 \in \mathbb{R}^{N \times d_1}$ and $\hat{H}_2 \in \mathbb{R}^{N \times d_2}$, respectively. To ensure consistent dimensions, d_1 is set equal to d_2. The hidden states are projected into the query (Q), key (K), and value (V) vector spaces using learnable linear transformations:

$$Q = W_q \hat{H}_1, \quad K = W_k \hat{H}_2, \quad V = W_v \hat{H}_2. \tag{15}$$

Next, multi-head attention uses scaled dot-product attention to fuse features from the hidden states. This produces the output H_3:

$$H_3 = \mathrm{Concat}(\mathrm{head}_1, \mathrm{head}_2, \ldots, \mathrm{head}_m) W_3. \tag{16}$$

Here, W_3 is the output transformation matrix, and m is the number of attention heads. The i-th attention head is defined as:

$$\mathrm{head}_i = \mathrm{Softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right) V, \tag{17}$$

where d_k represents the feature dimension of each head.

Finally, the hidden states are mapped to the target space using a learnable matrix $W_4 \in \mathbb{R}^{C \times d_{\text{out}}}$, producing the final fault diagnosis result H_{FD}:

$$H_{\text{FD}} = W_4 H_3 + b. \tag{18}$$

Here, W_4 is the transformation matrix, and $H_{\text{FD}} \in \mathbb{R}^C$, where C is the number of fault categories.

C. Loss Function for Fault Diagnosis Task

In the fault diagnosis task for industrial six-axis robots, there are seven fault types, each corresponding to a distinct category. Let C represent the total number of categories, and let y_i denote the true label. The true label is encoded using one-hot encoding, where the value is 1 for the actual category and 0 for all others. Let $\hat{y}_i$ represent the predicted probability for category i. Based on this, the loss function is defined as:

$$\mathcal{L}\left(y, \hat{y}\right) = -\sum_{i=1}^{C} y_i \log \hat{y}_i. \tag{19}$$

This cross-entropy loss function optimizes the model's classification performance by maximizing the predicted probability for the true category. It helps improve both the accuracy and robustness of fault diagnosis.

IV. Experimental Study

A. Experimental Setup

To validate the proposed method, experiments were conducted on a six-axis industrial robot platform. Faulty components were randomly used to replace functional reducers and motors to simulate real-world industrial scenarios. These faulty components exhibited early-stage failure characteristics, such as slight temperature anomalies and unusual noises. During the operation of the robot with faulty components, feedback voltage signals were collected from the motor drives of four axes. Seven fault types were simulated in the experiments.

The dataset includes 655 samples labeled as "Normal" for normal operation. Additionally, six fault categories were included:

1) A composite fault involving a reducer fault in axis one and a motor fault in axis two, with 655 samples labeled as "Composite Fault 1."
2) A composite fault involving a reducer fault in axis one and a motor fault in axis three, with 655 samples labeled as "Composite Fault 2."
3) A composite fault involving a reducer fault in axis three and a motor fault in axis four, with 655 samples labeled as "Composite Fault 3."
4) A single fault involving a reducer fault in axis three, with 655 samples labeled as "Single Fault 1."
5) A single fault involving a motor fault in axis two, with 655 samples labeled as "Single Fault 2."
6) A single fault involving a reducer fault in axis four, with 655 samples labeled as "Single Fault 3."

To investigate the effect of graph construction strategies on fault diagnosis performance, the raw time-series data were transformed into three distinct graph structures: K-nearest neighbor (KNN), radius-based, and path-based graphs. In the KNN graph, each node was connected to its $k = 3$ nearest neighbors based on feature similarity, while the radius graph connected nodes within a Euclidean distance threshold of 0.5. The path-based graph was defined according to the physical topology of the sensor layout, capturing the actual signal transmission pathways among the sensors. These graph structures were embedded into the proposed fault diagnosis framework to assess their impact on diagnostic accuracy. By applying the same model architecture across all graph types, we ensured a consistent evaluation environment. In addition to comparing different graph structures, the proposed method was benchmarked against five commonly used fault diagnosis algorithms [20], [22]–[25], The details are as follows:

1) **GATv2** [22]: GATv2 improves upon the original Graph Attention Network by introducing a more flexible attention mechanism. It allows attention weights to be dynamically adjusted based on the features of the target node, enabling the model to better adapt to varying graph structures.
2) **FourierGNN** [25]: This model incorporates a Fourier Graph Operator (FGO) to perform computations in the frequency domain. By leveraging spectral representations, FourierGNN can more efficiently capture patterns in graphs with complex or rapidly changing structures.
3) **Raindrop** [20]: Raindrop is designed for irregular and multivariate time series. It learns the relationships between sensors by estimating an underlying graph structure from raw observations, and uses this graph to infer missing or misaligned data points.
4) **TodyNet** [24]: TodyNet uses dynamic graphs to model dependencies between variables. It introduces a temporal graph pooling module to enhance the hierarchical representation of time series, and combines this with temporal convolutions in a unified framework for classification tasks.
5) **TimeMIL** [23]: TimeMIL reformulates time series classification as a weakly supervised learning problem using a multiple instance learning approach. It captures temporal patterns through a Transformer-based model, enhanced by learnable wavelet position tokens to reflect sequential dependencies.

To ensure reliable evaluation, the dataset was randomly divided into training and testing sets using a 7:3 ratio. During training, the learning rate was set to 0.001, and the number of attention heads in the model was fixed at 4. All experiments were carried out on an Ubuntu 16.0 server equipped with an Intel i9-10920X 3.50GHz CPU and an NVIDIA GeForce RTX 3090 GPU. The implementation was based on Python 3.8.18, and all models were developed using the PyTorch 2.2.1 framework. These settings were chosen to provide a consistent environment for benchmarking and to ensure reproducibility of the experimental results.

B. Experimental Result

This section presents the performance of the proposed method in the fault diagnosis task. Figure 2 shows the overall accuracy achieved by various algorithms on the industrial robot dataset containing seven fault types. The proposed method outperformed baseline models across all three graph construction methods, demonstrating its effectiveness in fault diagnosis tasks. Among the graph structures, the path-based model achieved the highest accuracy. This result is likely due to the path construction method's ability to capture global information, reduce redundant data, and better align with the dynamic characteristics and fault propagation patterns of industrial systems.

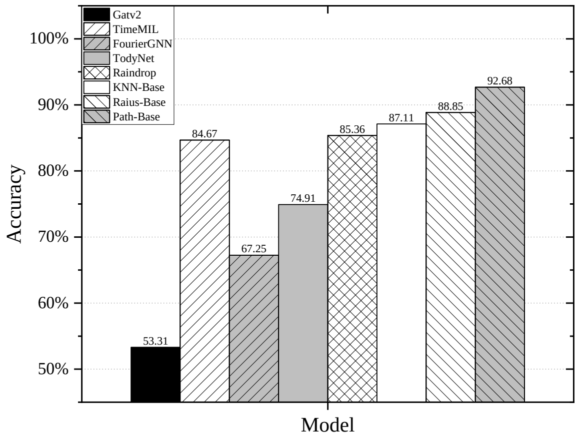

Fig. 2: Comparison of fault diagnosis accuracy across different algorithms.

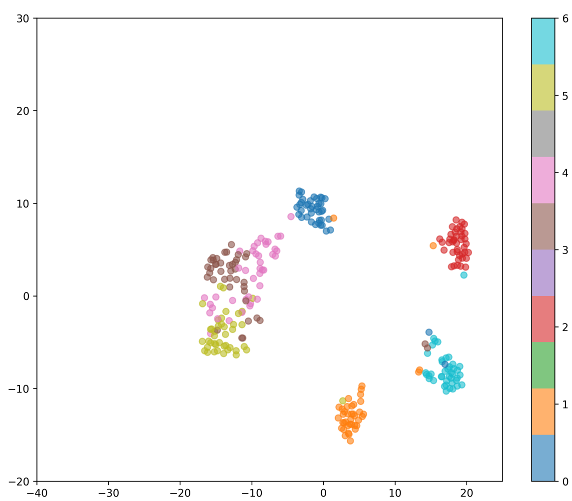

Fig. 3: t-SNE Visualization of KNN-Based Graph Fault Diagnosis.

We used t-SNE visualization to analyze the classification performance of the fault diagnosis model under three graph construction methods on the validation set, as shown in Figure3, Figure4 and Figure5. Labels 0-6 represent single fault 1, single fault 2, single fault 3, composite fault 1, composite fault 2, composite fault 3, and normal operation data. In the t-SNE plot based on KNN, the cluster of single fault 1 is very close to the composite faults, indicating poorer performance. This matches the results in Figure2, where the KNN-based method shows the lowest accuracy compared to the other two methods.This might be the KNN-based graph construction method only looks at local connections and does not take the whole structure into account. This may lead to weak

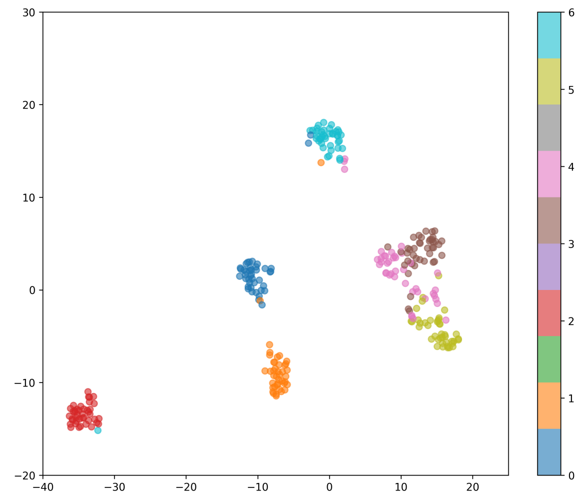

Fig. 4: t-SNE Visualization of Radius-Based Graph Fault Diagnosis.

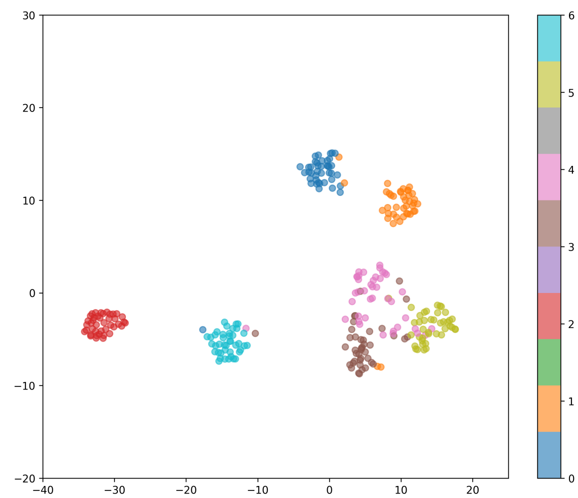

Fig. 5: t-SNE Visualization of Path-Based Graph Fault Diagnosis

information flow. In contrast, the t-SNE plots based on radius and path show clear boundaries between single and composite faults. However, in all three graph construction methods, the clusters of the three composite faults are not well separated. This suggests that the proposed model still has limitations in classifying composite faults.

V. Discussion

Our results show important findings about how different graph methods affect fault diagnosis in industrial robots. The three methods we tested—KNN, radius, and path-based—all worked well, but the path-based approach was the best with 92.68% accuracy. This tells us that the connections between measurements over time are very important for finding faults. This makes sense because robot faults usually appear as small changes in sensor readings that happen over time. However, we found some problems that need more work. The t-SNE results showed that the KNN method had trouble telling single faults apart from composite faults. The radius and path methods did better at this, showing clear separations between these fault types. But even these better methods had trouble telling different composite faults apart from each other. This means our model still needs improvement for handling complex faults. There are also some other limits to consider. The path-based method might not work as well when faults spread in unexpected ways or when sensor readings aren't closely linked in time. Also, while our method successfully combined spatial and temporal features through the attention mechanism, processing multiple graph structures at once could be too slow for systems that need real-time monitoring.

VI. CONCLUSION

This study presents a new way to find faults in industrial robots using graph neural networks. We looked at three ways to build graphs—KNN, radius, and path—to make fault detection better. Our research shows several key findings.

First, we found that the way we build the graph has a big effect on how well the system finds faults. Each method showed different strengths. The KNN method was the basic option but had trouble telling single faults from composite faults. The radius method did better and showed clearer boundaries between different types of faults. The path-based method was the best, with 92.68% accuracy. This method worked well because it could capture information about the whole system and remove extra data that wasn't needed.

Second, our new system design proved effective. We combined graph attention networks with a special self-attention encoder, and this helped us capture both the spatial connections between different parts and how things changed over time. This new design helped the system learn complex patterns in the robot's behavior. The t-SNE analysis showed that our method could separate different types of faults well, but still had some problems telling composite faults apart from each other.

The accuracy numbers were encouraging across all methods: 87.11% for KNN, 88.85% for radius, and 92.68% for path-based graphs. These results show that our approach works well for finding faults in industrial robots. But the t-SNE analysis also showed areas where we need to improve, especially in handling complex faults where multiple things go wrong at once. Our findings have important implications for making industrial robots more reliable. The path-based method's success suggests that looking at how sensor measurements connect over time is crucial for finding faults. This matches what we know about how faults develop in industrial robots, where problems often show up gradually over time.

For future work, we see several important directions. We could test more ways to build graphs that might work even better. We should also try our system in different types of industrial settings to make sure it works well in various situations. More research could focus on making the system better at telling different composite faults apart, since this is still a challenge. We could also look at ways to make the system faster for real-time monitoring, and test how well it works with different types of robots and equipment.

Using newer deep learning techniques could also help make fault diagnosis even better. These improvements would help make industrial robots smarter and more dependable. As industrial robots become more common, having reliable ways to find and fix problems quickly will become even more important.

MATCH & CONTRIBUTION

This contribution aligns closely with the theme of the ICE IEEE 2025 conference on "AI-driven Industrial Transformation: Digital Leadership in Technology, Engineering, Innovation & Entrepreneurship". The paper focuses on enhancing fault diagnosis in industrial robot systems through a graph neural network-based framework that captures both spatial and temporal patterns in multivariable sensor data. By comparing three distinct graph construction methods—KNN, radius, and path-based—the study highlights how the structure of data representation significantly affects diagnostic performance. The integration of graph attention networks with a diagonal masking self-attention mechanism enables the model to effectively learn relationships across both space and time, improving its capacity to identify complex fault patterns. The proposed method is validated using real-world datasets, with results showing a clear correlation between graph structure and diagnostic accuracy. This work contributes to the advancement of data-driven engineering by demonstrating how intelligent graph modeling can improve reliability in automated systems. It addresses the conference's core interests in leveraging AI techniques to drive technological innovation and operational resilience within industrial settings.

ACKNOWLEDGEMENT

Our work is supported by National Natural Science Foundation of China (62302103 and 62472106).

REFERENCES

[1] J. Arents and M. Greitans, "Smart industrial robot control trends, challenges and opportunities within manufacturing," *Applied Sciences*, vol. 12, no. 2, p. 937, 2022.

[2] T. Yin, N. Lu, G. Guo, Y. Lei, S. Wang, and X. Guan, "Knowledge and data dual-driven transfer network for industrial robot fault diagnosis," *Mechanical Systems and Signal Processing*, vol. 182, p. 109597, 2023.

[3] P. Kumar, S. Khalid, and H. S. Kim, "Prognostics and health management of rotating machinery of industrial robot with deep learning applications—a review," *Mathematics*, vol. 11, no. 13, p. 3008, 2023.

[4] P. Jieyang, A. Kimmig, W. Dongkun, Z. Niu, F. Zhi, W. Jiahai, X. Liu, and J. Ovtcharova, "A systematic review of data-driven approaches to fault diagnosis and early warning," *Journal of Intelligent Manufacturing*, vol. 34, no. 8, pp. 3277–3304, 2023.

[5] F. Calabrese, A. Regattieri, M. Bortolini, and F. G. Galizia, "Data-driven fault detection and diagnosis: Challenges and opportunities in real-world scenarios," *Applied Sciences*, vol. 12, no. 18, p. 9212, 2022.

[6] R. Huang, J. Xia, B. Zhang, Z. Chen, and W. Li, "Compound fault diagnosis for rotating machinery: State-of-the-art, challenges, and opportunities," *Journal of dynamics, monitoring and diagnostics*, pp. 13–29, 2023.

[7] J. Jiao, M. Zhao, J. Lin, and K. Liang, "A comprehensive review on convolutional neural network in machine fault diagnosis," *Neurocomputing*, vol. 417, pp. 36–63, 2020.

[8] J. Zhu, Q. Jiang, Y. Shen, C. Qian, F. Xu, and Q. Zhu, "Application of recurrent neural network to mechanical fault diagnosis: A review," *Journal of Mechanical Science and Technology*, vol. 36, no. 2, pp. 527–542, 2022.

[9] Y. Wang, M. Wu, X. Li, L. Xie, and Z. Chen, "A survey on graph neural networks for remaining useful life prediction: Methodologies, evaluation and future trends," *arXiv preprint arXiv:2409.19629*, 2024.

[10] M. A. Alobaidy, J. Abdul-Jabbar, and S. Al-khayyt, "Faults diagnosis in robot systems: a review," *Al-Rafidain Engineering Journal*, vol. 25, no. 2, pp. 164–175, 2020.

[11] T. Li, Z. Zhou, S. Li, C. Sun, R. Yan, and X. Chen, "The emerging graph neural networks for intelligent fault diagnostics and prognostics: A guideline and a benchmark study," *Mechanical Systems and Signal Processing*, vol. 168, p. 108653, 2022.

[12] W. Du, D. Côté, and Y. Liu, "Saits: Self-attention-based imputation for time series," *Expert Systems with Applications*, vol. 219, p. 119619, 2023.

[13] C. Liu, A. Cichon, G. Królczyk, and Z. Li, "Technology development and commercial applications of industrial fault diagnosis system: a review," *The International Journal of Advanced Manufacturing Technology*, pp. 1–33, 2021.

[14] X. Wen and Z. Xu, "Wind turbine fault diagnosis based on relieff-pca and dnn," *Expert Systems with Applications*, vol. 178, p. 115016, 2021.

[15] J. Cen, Z. Yang, X. Liu, J. Xiong, and H. Chen, "A review of data-driven machinery fault diagnosis using machine learning algorithms," *Journal of Vibration Engineering & Technologies*, vol. 10, no. 7, pp. 2481–2507, 2022.

[16] Z. Liu and L. Zhang, "A review of failure modes, condition monitoring and fault diagnosis methods for large-scale wind turbine bearings," *Measurement*, vol. 149, p. 107002, 2020.

[17] Q. Wang, R. Huang, J. Xiong, J. Yang, X. Dong, Y. Wu, Y. Wu, and T. Lu, "A survey on fault diagnosis of rotating machinery based on machine learning," *Measurement Science and Technology*, vol. 35, no. 10, p. 102001, 2024.

[18] Z. Chen, J. Xu, T. Peng, and C. Yang, "Graph convolutional network-based method for fault diagnosis using a hybrid of measurement and prior knowledge," *IEEE transactions on cybernetics*, vol. 52, no. 9, pp. 9157–9169, 2021.

[19] Y. Tang, X. Zhang, Y. Zhai, G. Qin, D. Song, S. Huang, and Z. Long, "Rotating machine systems fault diagnosis using semisupervised conditional random field-based graph attention network," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–10, 2021.

[20] X. Zhang, M. Zeman, T. Tsiligkaridis, and M. Zitnik, "Graph-guided network for irregularly sampled multivariate time series," *arXiv preprint arXiv:2110.05357*, 2021.

[21] J. Yu and Y. Zhang, "Challenges and opportunities of deep learning-based process fault detection and diagnosis: a review," *Neural Computing and Applications*, vol. 35, no. 1, pp. 211–252, 2023.

[22] S. Brody, U. Alon, and E. Yahav, "How attentive are graph attention networks?" *arXiv preprint arXiv:2105.14491*, 2021.

[23] X. Chen, P. Qiu, W. Zhu, H. Li, H. Wang, A. Sotiras, Y. Wang, and A. Razi, "Timemil: Advancing multivariate time series classification via a time-aware multiple instance learning," *arXiv preprint arXiv:2405.03140*, 2024.

[24] H. Liu, D. Yang, X. Liu, X. Chen, Z. Liang, H. Wang, Y. Cui, and J. Gu, "Todynet: temporal dynamic graph neural network for multivariate time series classification," *Information Sciences*, p. 120914, 2024.

[25] K. Yi, Q. Zhang, W. Fan, H. He, L. Hu, P. Wang, N. An, L. Cao, and Z. Niu, "Fouriergnn: Rethinking multivariate time series forecasting from a pure graph perspective," *Advances in Neural Information Processing Systems*, vol. 36, 2024.