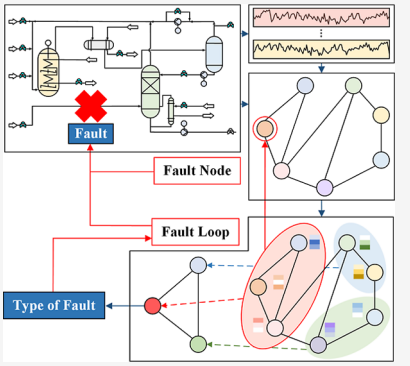
I&EC research
Industrial & Engineering Chemistry Research
pubs.acs.org/IECR
Article

Topology-Guided Graph Learning for Process Fault Diagnosis

Mingwei Jia, Junhao Hu, Yi Liu,* Zengliang Gao, and Yuan Yao*

Cite This: *Ind. Eng. Chem. Res.* 2023, 62, 3238−3248
Read Online
ACCESS | Metrics & More | Article Recommendations

ABSTRACT: Faults in the process industry can be diagnosed using various data-driven methods, but the intrinsic relationships between inputs and outputs, particularly the physical consistency of model prediction logic, have received little attention. To address this issue, we propose a topology-guided graph learning fault diagnosis framework that combines the concept of graphs with process physics. Our framework focuses on knowledge embedding and explanation and includes several key components: a topology graph based on the flowchart, a self-attention mechanism to discover distinctive knowledge from data, graph convolution to capture variable relationships, graph pooling to coarsen graph data, and a gating mechanism to establish long-term dependencies. We also use a graph explainer to assess the physical consistency of the model's prediction logic. We demonstrate the feasibility of our method using the Tennessee Eastman process and show that it is not a black-box model but rather has natural advantages in terms of effectiveness and explanation.



1. INTRODUCTION

The complexity of industrial systems is on the rise, and this presents new challenges for fault diagnosis (FD) in the modern process industry. To effectively address these challenges, FD methods must be able to quickly and accurately identify faults in the face of a large number of monitored variables and dynamic interactions.[1] Despite this, FD remains a difficult task due to the complexity of the processes involved and the propagation of faults.[2] Typically, FD methods can be divided into two main categories: knowledge/model-driven and data-driven.[3] While model-driven methods offer high interpretability, they can be challenging to construct in a short time frame owing to the complex mechanisms of the processes involved.[4] In contrast, data-driven methods that do not rely on prior knowledge have become increasingly popular in recent years.[5]

Conventional FD methods include multivariate analysis methods,[6] support vector machines (SVMs)[7] and other kernel methods,[8] Bayesian methods (BMs),[9] etc. Multivariate analysis methods, such as principal component analysis (PCA),[6] transform higher-dimensional process data into a lower-dimensional feature space, preserving significant information about fault variables. Kernel methods, like SVM and kernel PCA (KPCA), have been widely used in the FD of industrial processes. Zhang developed an FD method for chemical processes that combines KPCA and SVM.[10] BMs have been widely applied because they can handle smearing effects.[11,12] For example, Amin et al. presented an integrated method for FD using PCA and BM.[13] However, both exact and approximate algorithms for calculating posteriors in BM can lead to NP-hard problems, which can be a challenge for large models.[14] Despite their potential, these methods often require a significant amount of domain expertise to determine fault features in both spatial and temporal domains, and the FD accuracy is still not high enough for widespread practical use.[15]

The rapid development of DL has led to its widespread use in both academic and industrial fields. DL excels at analyzing large-scale, high-dimensional data without the need for hand-designed heuristics, making it a popular choice for FD.[16,17] For example, Mirzaei et al. compared and visualized the performance of recurrent neural networks such as long short-term memory (LSTM) in FD.[18] However, DL has limitations when it comes to security-related FD applications as it can be difficult to incorporate knowledge.[19] In particular, the logic of a DL-based FD model may not align with physical processes, leading to the creation of spurious relationships between variables. Additionally, few studies have focused on incorporating physical knowledge into DL or, more importantly, on embedding the underlying physical intuitions of industrial processes into the model training process, which would make DL models more physically meaningful and applicable to different scenarios.

Physics-informed neural networks (PINNs) have emerged as a promising approach to address challenges in the modern process industry due to their high reliability and transparency.[20] In general, PINNs incorporate physical information into the model through the use of partial differential equations

Received: October 7, 2022
Revised: February 1, 2023
Accepted: February 1, 2023
Published: February 13, 2023

https://doi.org/10.1021/acs.iecr.2c03628
Ind. Eng. Chem. Res. 2023, 62, 3238−3248

(PDEs).[21] With recent advancements, PINNs have become increasingly versatile, providing more options for their implementation.[22] Compared to purely data-driven models, PINNs prioritize the integration of prior knowledge, which is crucial for effective modeling.[23] The key elements of PINNs include[24] (1) preprocessing input data to extract hidden physical information and/or supplement it with production parameters and simulation data; (2) ensuring physical consistency between model inputs and outputs by penalizing any output that does not comply with physical principles using a loss function; (3) incorporating physical intuition into activation functions that lack physical meaning; (4) training the model to perform a physically meaningful analysis of the input information during model optimization, which differs from (3) as it fundamentally alters the data representation, while (3) only changes how the data is processed without modifying the data itself; and (5) evaluating the physical consistency of the model's predictive logic, which focuses on explaining the behavior of the model. Using this approach, PINNs enhance model interpretability and reduce errors by relying on physical principles.[25] However, the application of PINNs in the process industry can be limited as first principles of the process may be complicated. Thus, it is important to find ways to incorporate other forms of prior knowledge, such as flowcharts.

As the concept of graphs has gained popularity, attention has been focused on its ability to describe relationships between entities. It is no surprise that graph-based DL techniques are well-suited for the field of FD as they can take into account the relationships between process variables, for example, through the use of graph neural networks (GNNs).[26] Importantly, the properties of graphs can provide useful ideas for seamlessly integrating different types of process information, such as variable relationship which can be described using a symbolic directed graph (SDG)[27,28] and process status indicated by measured data. In this vein, a prominent model, the graph convolutional network (GCN),[29] has been adopted in the field of FD. For instance, Yu et al. used a deep GCN for the FD of wind turbine gearboxes,[30] and Chen et al. developed a GCN-based FD model for pulsed rectifiers.[31] Although these initial applications of GCNs in the field of FD are promising, they do not incorporate prior knowledge, which has led to the "black-box" problem, where it is difficult to explain the behavior of the model. Therefore, these initial applications are insufficient for ensuring physical consistency.

In this study, we propose a graph-based FD framework, called topology-guided graph learning (TGGL), which aims to improve the accuracy and physical consistency of FD by leveraging the concept of graphs as a bridge connecting process mechanisms and the data-driven model. The TGGL framework builds a SDG from the flowchart to describe the process topology and augments it with a self-attention mechanism (SAM)[32] which is constrained by a regularization loss. The information is then propagated according to the topological graph structure using the GCN structure and a convolutional gating mechanism.[33] Additionally, the physical consistency of the model is assessed using the GNN explainer.[34] As a result, TGGL enables predictions that are consistent with physics and ensures model transparency. The performance of the TGGL-based FD framework is evaluated on the Tennessee Eastman (TE) process[35] and is compared with several popular methods to demonstrate the effectiveness and interpretability of the physics-based architecture design of the neural network.

2. PRELIMINARIES

Definition 1: $G = (\mathbf{V}, \mathbf{A}, \mathbf{E})$ denotes a directed graph data, where $\mathbf{V}$ denotes the set of vertices (or nodes), $\mathbf{E}$ the set of edges, and $\mathbf{A}$ the adjacency matrix of graph G with $\mathbf{V}$ nodes or variables. Typically, the elements in adjacency matrix $\mathbf{A}$ are 0 and 1; 0 means that no connection exists between two nodes and 1 indicates that a connection exists. If the elements in $\mathbf{A}$ are not binarized, the element indicates the strength of the connection between two nodes. G denotes the relationships of nodes in the spatial dimension, and the network structure does not change with time. In this study, the graph structure is directed.

Definition 2: $\mathbf{X}_G \in \mathbb{R}^{C\times V\times T}$ denotes the graph signal matrix, where C denotes the number of channels and T the time step of the observation.

2.1. Symbolic Directed Graph. The nodes in an SDG can represent physical factors such as pressure, temperature, liquid level, and flow, and the direction lines indicate the relationships between nodes. If the initial node has a positive (incremental) effect on the end node, a solid arrow connects them; otherwise, a dashed arrow connects them. By using the concept of SDG, a topology graph of process variables can be constructed, which helps the GCN encode data in a physically meaningful way. Figure 1 shows a simple SDG; + indicates a positive relationship and - indicates a negative relationship; for example, an increase in A leads to an increase in B and an increase in B leads to a decrease in C.

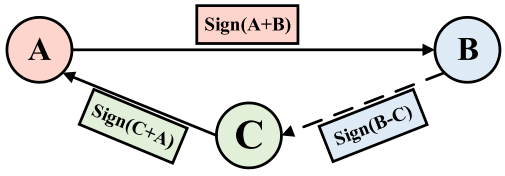


Figure 1. Simple SDG.

2.2. Graph Convolutional Network. In order to propagate fault information according to predefined variable relationships, the TGGL framework introduces a GCN to encode process variables based on the topology graph. The graph convolutional operation can be formulated as follows[28]

$$\mathrm{GCN}(\mathbf{X}_G, \mathbf{A}) = \sigma(\mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}\mathbf{X}_G\mathbf{U}\cdot\mathbf{\Psi}) \quad (1)$$

where "*" denotes the convolution operation; $\mathbf{\Psi}$ represents a convolution kernel of size (1×1); $\mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$ is the convolution kernel obtained by the approximate Fourier transform; $\mathbf{D}$ denotes the degree-diagonal matrix of adjacency matrix $\mathbf{A}$, whose diagonal value is the sum of the elements of each row in $\mathbf{A}$; $\mathbf{X}_G$ denotes the input of the graph convolution layer; $\mathbf{U}$ is a weight matrix; and σ denotes an activation function, such as the rectified linear unit. Figure 2 illustrates the GCN structure. The red nodes aggregate information from their neighboring 1-hop nodes (circled).

2.3. Convolutional Long Short-Term Memory Network. Generally, data with faults in chemical processes are highly time-variant. To establish the long-term dependencies of the data in the time dimension, the convolutional gating mechanism from ConvLSTM is utilized. The formula for ConvLSTM is as follows[33]

https://doi.org/10.1021/acs.iecr.2c03628
Ind. Eng. Chem. Res. 2023, 62, 3238−3248

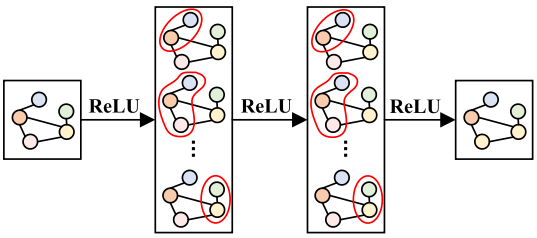


Figure 2. GCN with two graph convolution layers.

$$\begin{aligned}
\mathbf{I}_t &= \sigma(\mathbf{W}_{xi}\mathbf{X}_t + \mathbf{W}_{hi}\mathbf{H}_{t-1} + \mathbf{W}_{ci}\mathbf{C}_{t-1} + \mathbf{B}_i)\\
\mathbf{F}_t &= \sigma(\mathbf{W}_{xf}\mathbf{X}_t + \mathbf{W}_{hf}\mathbf{H}_{t-1} + \mathbf{W}_{cf}\mathbf{C}_{t-1} + \mathbf{B}_f)\\
\mathbf{C}_t &= \mathbf{F}_t\cdot\mathbf{C}_{t-1} + \mathbf{i}_t\cdot \tanh(\mathbf{W}_{xc}\mathbf{X}_t + \mathbf{W}_{hc}\mathbf{H}_{t-1} + \mathbf{B}_c)\\
\mathbf{O}_t &= \sigma(\mathbf{W}_{xo}\mathbf{X}_t + \mathbf{W}_{ho}\mathbf{H}_{t-1} + \mathbf{W}_{co}\mathbf{C}_{t-1} + \mathbf{B}_o)\\
\mathbf{H}_t &= \mathbf{O}_t\cdot \tanh(\mathbf{C}_t)
\end{aligned} \tag{2}$$

where "·" represents the Hadamard product, t denotes the number of time steps, $\mathbf{I}_t$ is the result of the input gate, $\mathbf{F}_t$ is the result of the forget gate, $\mathbf{C}_t$ is the current result of the memory unit, $\mathbf{O}_t$ is the result of the output gate, $\mathbf{H}_t$ is the output result of the module at the current moment, and $\mathbf{W}$ and $\mathbf{B}$ are the trainable convolution kernel and bias, respectively. Figure 3 illustrates the structure of a ConvLSTM.

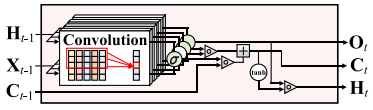


Figure 3. Structure of a ConvLSTM.

2.4. Graph Pooling. Typically, a fault in a process occurs initially in a localized unit containing a subgroup of process variables and then propagate throughout the system. As such, treating the variable subgroup as a "supernode" can be beneficial for FD. To achieve this, the proposed method known as TGGL uses graph pooling[36] to coarsen the nodes in the graph. To coarsen an adjacency matrix from an SDG, the number of supernodes, designated as K, is initially determined. The spectral clustering algorithm is subsequently utilized to derive the cluster assignment matrix $\mathbf{R} \in \mathbb{R}^{V\times K}$. The coarsening matrix $\mathbf{A}_{coar} \in \mathbb{R}^{K\times K}$ is defined according to $\mathbf{R}$. In this study, $\mathbf{R}$ is used to obtain coarsening graph signal $\mathbf{X}_{coar}$. Herein, $\mathbf{X}_{coar}$ and $\mathbf{A}_{coar}$ serve as the input data and adjacency matrix for the next layer, respectively.[36]

$$\begin{aligned}
\mathbf{A}_{coar} &= \mathbf{R}^T\mathbf{A}\mathbf{R}\\
\mathbf{X}_{coar} &= \mathbf{R}^T\mathbf{X}_G
\end{aligned} \tag{3}$$

As illustrated in Figure 4, graph pooling coarsens the graph structure and signal, resulting in a new adjacency matrix and a coarsened graph signal.

2.5. GNN Explainer. After making predictions with the model, it is essential to provide an explanation of which variables contributed to the model's prediction. This is because not all variables are given equal consideration by the model during the inferencing process. To identify the key contributing factors, the GNN explainer[34] is utilized. It aims to identify the compact subgraph that plays a crucial role in the final prediction by learning a mask $\mathbf{M}$, which can mask certain

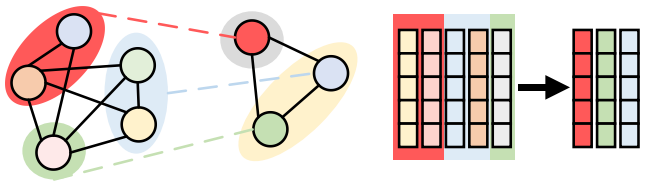


Figure 4. Pooling process of the graph structure and signal.

nodes and edges. For more detailed information on the mathematical derivation process, please refer to ref 34. The mathematical expression of the GNN explainer is as follows[34]

$$\mathbf{A}_S = \mathbf{A}\cdot\mathrm{sigmoid}(\mathbf{M}) \tag{4}$$

where $\mathbf{A}_s$ denotes the masked adjacency matrix, $\mathbf{M}$ represents the learnable mask matrix, and $\mathbf{A}$ is the adjacency matrix of G. The sigmoid(.) function is applied to $\mathbf{M}$, mapping its values to the range of $[0,1]$.

It is important to note that when interpreting the results, controlling the size of the subgraph is necessary to avoid the complexity of the subgraph leading to an overwhelming or unhelpful explanation. This can be achieved by utilizing the following losses.

$$\begin{aligned}
\mathrm{Loss} &= \mathrm{Loss}_{pred} + \mathrm{Loss}_{size} + \mathrm{Loss}_{dis}\\
\mathrm{Loss}_{size} &= \|\mathbf{M}\|_2\\
\mathrm{Loss}_{dis} &= \mathrm{H}(\mathbf{M})
\end{aligned} \tag{5}$$

The GNN explainer requires retraining the mask on the trained model. Loss_{pred} represents the cross-entropy loss between the predicted and true classes during the training process. Loss_{size} and Loss_{dis} are the penalty terms for the mask matrix value and its discreteness, respectively. $\|.\|_2$ represents the L2-norm. The schematic of the GNN explainer is illustrated in Figure 5.

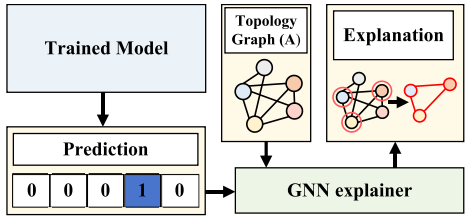


Figure 5. Schematic of the GNN explainer.

3. TGGL-BASED FAULT DIAGNOSIS FRAMEWORK

This section presents the TGGL-based FD framework, as illustrated in Figure 6. The subsequent sections will provide further elaboration on the framework.

3.1. Framework of TGGL-Based Fault Diagnosis. In the TGGL-based FD framework, the adjacency matrix of the SDG serves as a carrier of prior knowledge, enabling the fusion of the process mechanism and the data-driven model. A GCN is utilized to extract information from the adjacency matrix. To improve the model's ability to detect faulty nodes and loops, graph pooling is introduced to coarsen the nodes. Furthermore, building upon GCN, a convolutional gate mechanism is implemented to construct long-term dependencies of the data. Finally, the GNN explainer is introduced to provide evidence for the physical consistency of the model's behavior. The

https://doi.org/10.1021/acs.iecr.2c03628
Ind. Eng. Chem. Res. 2023, 62, 3238−3248

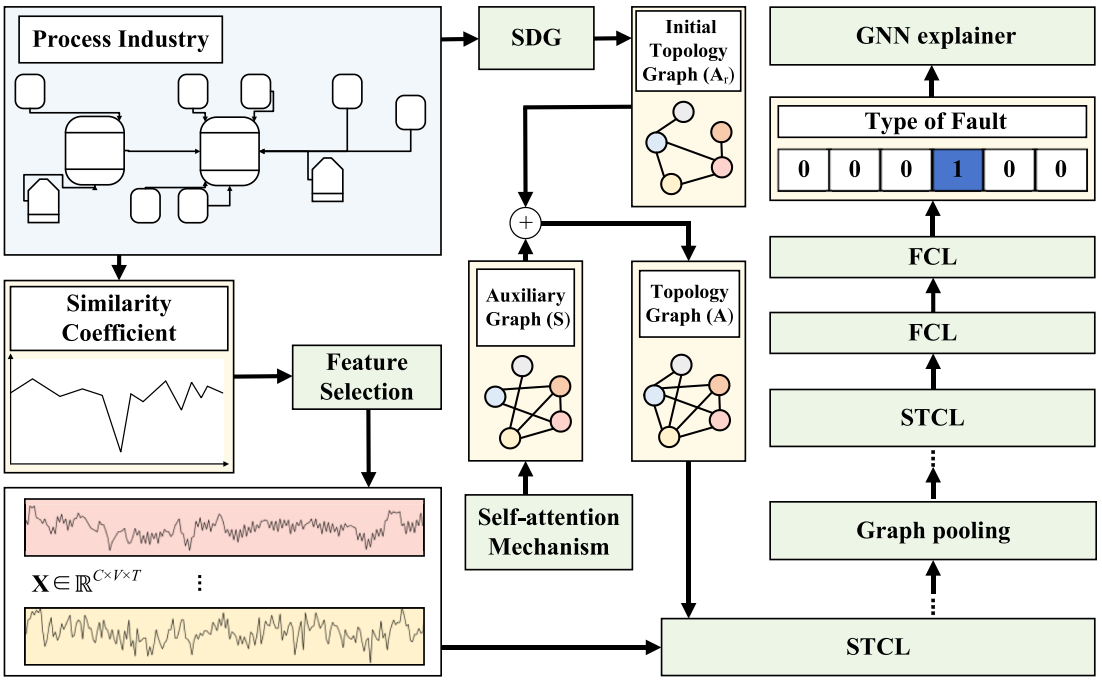


Figure 6. Framework of TGGL-based FD.

proposed model has a unique ability to integrate prior knowledge and process data, as compared to traditional DL-based FD models that lack prior knowledge, due to its physics-based design architecture.

In summary, the TGGL-based FD framework can be described as follows. (1) A similarity coefficient r is defined to measure the contribution of each variable to the fault, and features are selected based on r. (2) An initial topology graph in the form of SDG is constructed from the flowchart and represented by an adjacency matrix $\mathbf{A}_r$, where the elements in $\mathbf{A}_r$ take on the values of $(-1, 0, 1)$, indicating positive, no, and negative influences, respectively. (3) An auxiliary graph is constructed by the SAM and its adjacency matrix is denoted as $\mathbf{S}$ to complement $\mathbf{A}_r$ and obtain the topological graph, where the elements in $\mathbf{S}$ are mapped to $[-1, 1]$, indicating the strength of the connection between the variables. (4) The topological graph is represented by an adjacency matrix $\mathbf{A}$, and a spatial−temporal convolution layer (STCL) is constructed based on it, which includes a spatial convolutional layer (SCL) and a temporal convolutional layer (TCL). (5) Multi-layer STCLs are stacked to form an encoding structure. (6) To improve graph representation, graph pooling is applied to coarsen the graph structure and signal. The $\mathbf{X}_G$ encoded by STCLs and $\mathbf{A}$ are pooled together by the pooling layer to obtain $\mathbf{X}_{coar}$ and $\mathbf{A}_{coar}$, which are used as the input for the next STCL. (7) Two fully connected layers (FCLs) are used to build the mapping between data and fault categories. (8) Finally, the GNN explainer is trained to evaluate the behavior and logic of the model based on the predictions while querying fault loops and variables. Steps (1), (2), (3), (4), (6), and (8) are described in more detail in the following sections. The proposed algorithm is also outlined in Algorithm 1.

3.2. Variable Selection and Topology Graph Construction. Typically, the abundant data collected from the chemical process is characterized as "data-rich but information-poor", which necessitates feature selection. Given process data $\mathbf{X} \in \mathbb{R}^{Z \times V \times T}$ with Z states, V variables, and T time steps, the relevance of variable v to state z can be qualitatively represented by measuring the similarity between the normal sequence $\mathbf{x}_{0,v}$ and the fault sequence $\mathbf{x}_{z,v}$, where z represents the fault type and 0 defines the normal type. The similarity coefficient reflects the similarity in shape and value between the sequences. Therefore, the similarity coefficient $r_{z,v}$ is defined in terms of Euclidean distance

$$r_{z,v} = 1 - \frac{\|\mathbf{x}_{z,v} - \mathbf{x}_{0,v}\|_2}{\sum_{z=1}^{Z} \|\mathbf{x}_{z,v} - \mathbf{x}_{0,v}\|_2 + \varepsilon} \tag{6}$$

where ε is an infinitely small quantity to prevent the denominator from 0 and z and v represent the states and variables of $\mathbf{x}$. The closer the $r_{z,v}$ is to 1, the more similar the sequences, and the less dependent the state change is on the variable. A threshold is set for feature selection, and the variable is removed when r is greater than the threshold.

After feature selection, the data of V_s variables is obtained, and an SDG is constructed according to the process flowchart as an initial topology graph $\mathbf{A}_r$. Variables that are not directly connected in the topology graph may exhibit strong correlations in different samples as the fault propagates. However, with just GCN, capturing this correlation may require a large number of layers. To simplify the model structure, SAM[32] is employed to capture this correlation and construct an auxiliary graph that provides connections between two nodes that are not physically connected, thus completing $\mathbf{A}_r$. In this study, the auxiliary graph is represented by an adjacency matrix $\mathbf{S}$. In SAM, query $(\mathbf{Q})$ and key $(\mathbf{K})$ are used to facilitate building the attention matrix, and the number of heads h can be set to obtain different attention matrices in parallel. The input data $\mathbf{X}_G$ is simultaneously used as the query $(\mathbf{Q})$ and key $(\mathbf{K})$ to map to the multi-head space through different trainable weights. Each projected part is computed in parallel to obtain the correlations between any two variables. Subsequently, all attention matrices are concatenated, and an FCL is used to map the result to matrix $\mathbf{S}$. Since the elements

https://doi.org/10.1021/acs.iecr.2c03628
Ind. Eng. Chem. Res. 2023, 62, 3238−3248

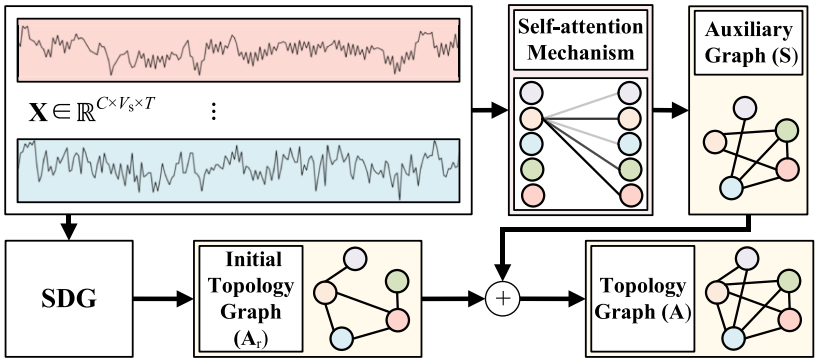


Figure 7. Completion process of the initial topology graph.

in $\mathbf{A}_r$ are (−1, 0, 1), Z-score normalization is used to map $\mathbf{S}$ to [−1,1]

$$\mathbf{S} = \mathrm{Z} - \mathrm{Score}(\mathrm{FCL}(\mathrm{Concat}(\mathbf{head}_1, \ldots, \mathbf{head}_h)))$$

$$\mathbf{head}_h = \mathrm{SelfAttention}(\mathbf{QW}_h^Q, \mathbf{KW}_h^K)$$

$$\mathrm{SelfAttention}(\mathbf{Q}_h, \mathbf{K}_h) = \mathrm{softmax}\left(\frac{\mathbf{Q}_h \mathbf{K}_h^{\mathrm{T}}}{\sqrt{N}}\right)$$

7where h is the number of attention heads and $\mathbf{W}_h^Q$ and $\mathbf{W}_h^K$ are the mapping weights of $\mathbf{Q}$ and $\mathbf{K}$, respectively. As depicted in Figure 7, the selected data is used to construct the SDG and compute the attention coefficients. Afterward, the initial topological graph $\mathbf{A}_r$ and auxiliary graph $\mathbf{S}$ are obtained based on them. Finally, the topology graph adjacency matrix $\mathbf{A}$ is obtained by combining $\mathbf{A}_r$ and $\mathbf{S}$.

3.3. Spatial−Temporal Convolution Layer. Typically, the process variables, particularly the fault variables, exhibit strong coupling and time variation. The STCL is responsible for capturing these variable relationships and building long-term dependencies. The STCL is split into the SCL and TCL. The SCL uses GCN to encode the data based on the adjacency matrix A, whose elements lie in [−1, 1]. The TCL uses ConvLSTM to establish long-term dependencies of data in the time dimension. Figure 8 illustrates the STCL structure, where

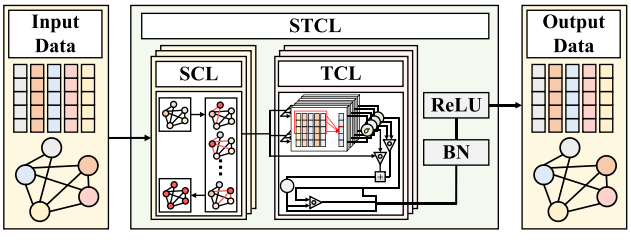


Figure 8. Internal structure of STCL.

batch normalization (BN) and rectified linear unit (ReLU) are used as the batch normalization and activation functions, respectively. By combining the SCL and TCL, the STCL can encode and embed the data in the spatial−temporal dimension.

3.4. Graph Pooling Layer. To advance graph representation, graph pooling layers were inserted in the STCLs. The data $\mathbf{X}_G$ encoded by the previous STCL layer is pooled to obtain pooled data $\mathbf{X}_{coar}$. The adjacency matrix $\mathbf{A}$ of the topology graph is pooled to obtain pooled adjacency matrix $\mathbf{A}_{coar}$. Then, $\mathbf{X}_{coar}$ and $\mathbf{A}_{coar}$ are used as the inputs and adjacency matrix for the next STCL. The graph pooling layer promotes the globalization of the graph representation and reduces the parameter capacity of the model.[36]

3.5. GNN Explainer Layer. The GNN explainer serves as the core of the GNN explainer layer, responsible for evaluating the model's prediction logic. In the GNN explainer, a mask is added to the trained FD model and retrained as a new parameter. This process allows us to identify the subgraphs and nodes that have the greatest impact on the model's prediction, which can be considered as the loops and variables most likely to cause faults.

4. CASE STUDY

Detailed comparisons and analysis are presented, with the TE process serving as a representative example in this study. Researchers in any process industry can apply the findings and techniques of this study to their own FD efforts. The code for this work, including all necessary code and datasets, can be accessed at https://github.com/MingweiJia/TGGL.

4.1. Simulation Setup. The benchmark TE process is a simulated system based on a real-world chemical process, commonly used in research related to process optimization, control, monitoring, and FD. The TE process involves four gaseous reactants (A, C, D, and E) and two liquid products (G and H). Additionally, a small amount of inert, insoluble component B is present in each reactant. In the reactor, four reactions occur simultaneously in the presence of a catalyst, two of which result in the production of liquid products G and H, as outlined below

$$\begin{aligned} &\mathrm{A}_{(g)} + \mathrm{C}_{(g)} + \mathrm{D}_{(g)} \rightarrow \mathrm{G}_{(liq)} \\ &\mathrm{A}_{(g)} + \mathrm{C}_{(g)} + \mathrm{E}_{(g)} \rightarrow \mathrm{H}_{(liq)} \\ &\mathrm{A}_{(g)} + \mathrm{E}_{(g)} \rightarrow \mathrm{F}_{(liq)} \\ &3\mathrm{D}_{(g)} \rightarrow 2\mathrm{F}_{(liq)} \end{aligned} \quad (8)$$

where (g) represents gas and (liq) denotes liquid. All of the reactions are exothermic and irreversible, and the rate of each reaction is determined by the temperature and concentration of the reactant gases.[35]

The dataset used in this study has a data sampling interval of 3 min and includes 52 variables and 21 types of faults, where fault 0 represents normal operation, S1-41 are process variables, and MV1-10 are manipulated variables. The training and test sets each contain 500 and 960 samples for each fault. The faults are introduced into the training and test sets at the

https://doi.org/10.1021/acs.iecr.2c03628
Ind. Eng. Chem. Res. 2023, 62, 3238−3248

first and eighth hour, respectively; therefore, data from hours 0−1 and 0−8 in the training and test sets are excluded. Additionally, fault 6 in the training and test sets causes a machine halt after 7 and 14 h, respectively. As a result, fault 6 in the training and test sets includes only 140 and 280 samples, respectively. The dataset used in this study can be obtained from the Harvard database.[37]

In this section, data is collected using a moving window approach. To maintain the diagnostic accuracy of fault 0, the amount of data from fault 0 in the training set is increased fivefold compared to the data from the other faults. The width of the moving window is set to 40 samples. In the training set, the sampling frequency for fault 0 (normal) and fault 6 is one sample, while the sampling frequency for the remaining faults is five. This results in 441, 101, and 89 samples of fault 0, fault 6, and the remaining faults, respectively. In the test set, the sampling frequency for fault 6 is two and that for the remaining faults is five. This results in 121 and 153 samples of fault 6 and the remaining faults, respectively. The first 89 samples of each fault are used as the new test set, and the 90−120 samples of each fault are used as the validation set.

After normalizing the data, the training set is used to calculate the similarity coefficient of each variable as outlined in eq 5. Figure 9 illustrates the similarity coefficients of two

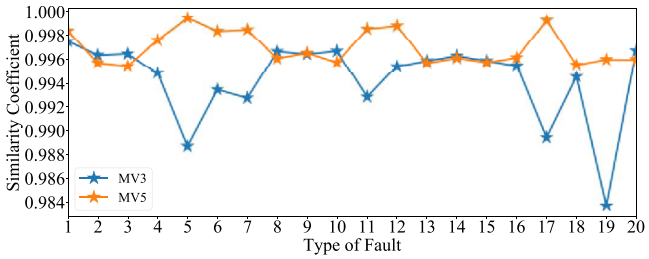


Figure 9. Similarity coefficient of two representative variables.

representative variables under different faults. The similarity coefficient of MV3 varies between faults, while the similarity coefficient of MV5 is consistently high across all faults. Therefore, variables like MV5 should be eliminated. To keep as many variables as possible, the threshold is set to 0.99. As a result, MV5 (compressor circulation valve), MV9 (stripper steam valve), and S20 (compressor power) were removed. Additionally, S28-41 were excluded because it is challenging to measure them in real-time.

According to the flowchart of the TE process,[34] the SDG was created for the remaining 35 variables. Figure 10 shows examples of how the SDG was derived using the control systems of composition of D in stream 6 and the reactor temperature as examples. For instance, an increase in the feed amount MV1 causes an increase in the mass flow rate S2 of component D, which in turn leads to an increase of the composition of D in stream 6, that is, S26. To compensate such a disturbance, the controller reduces the value of S26 by decreasing MV1. This is represented by solid arrows from MV1 to S2 and from S2 to S26 and dashed arrows from S26 to MV1 in Figure 10a. Similarly, an increase in the reactor temperature S9 leads to an increase in the reactor cooling water flow rate MV10 to compensate the disturbance, which causes a decrease in the reactor cooling water outlet temperature S21. This is represented by solid arrows from S9 to MV10, from MV10 to S21, and from S21 to S9 in Figure

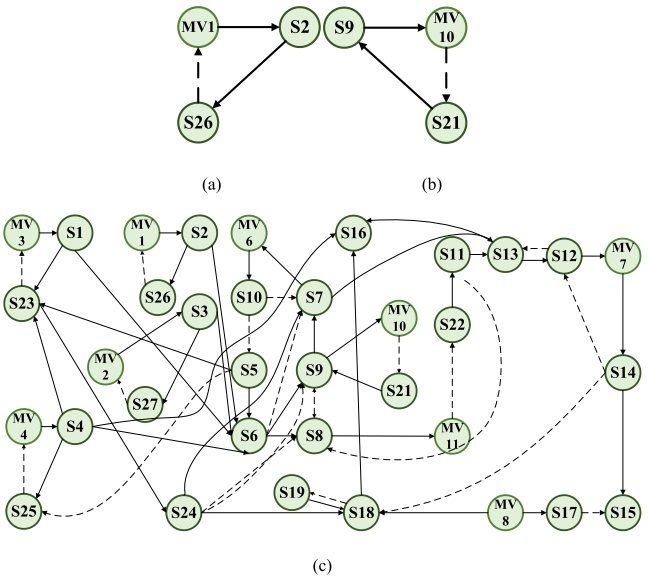


Figure 10. Topology graph of the (a) control system regulating the composition of D, (b) reactor temperature control system, and (c) entire TE process.

10b. Following the similar way, the complete SDG of the TE process can be developed as shown in Figure 10c. In conclusion, the input and output of the model can be represented by the following equations

$$\begin{cases} \mathbf{X} = [\text{S1} - \text{S19}, \text{S21} - \text{S27}, \text{MV1} - \text{MV4}, \text{MV6} \\ \qquad - \text{MV8}, \text{MV10}, \text{MV11}] \\ Y = [y]\ y \in [0,20] \end{cases} \tag{9}$$

The pooled graph has been precomputed as the graph pooling process does not have any trainable parameter. The number of supernodes (K = 10) was determined through trial and error for pooling the topology graph. The cluster assignment matrix (R) was obtained using the spectral clustering algorithm and was used to coarsen the input adjacency matrix ($\mathbf{A}$) and the input graph data ($\mathbf{X}_G$) of the graph pooling layer. As seen in Figure 11, the pooled graph

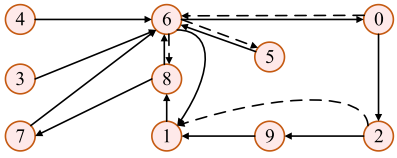


Figure 11. Pooled topology graph of the TE process.

bears resemblance to the topology graph, indicating that the pooling process effectively captures the inherent structure of the graph. Table 1 provides a correspondence between the supernodes and subnodes.

The FD rate (FDR) was used to evaluate the model. Additionally, the mean and variance of the training set were used to normalize the validation and test sets. In model architecture, going deep is an important direction to improve network representation capabilities. However, blindly increasing the number of layers may degrade the performance. Therefore, a series of model architectures are set for selection, and the graph pooling layer is set in the middle. Table 2 shows the FDR of TGGL with different architectures on the

https://doi.org/10.1021/acs.iecr.2c03628
Ind. Eng. Chem. Res. 2023, 62, 3238−3248

Table 1. Dependencies between Nodes

supernodes	subnodes of corresponding supernodes
0	S22, S11, MV11
1	S19, S18
2	S12, S13, S14, MV7
3	S3, S27, MV2
4	S2, S26, MV1
5	S9, S21, MV10
6	S6, S7, S8, S10, S24, MV6
7	S1, S23, MV2
8	S4, S5, S16, S25, MV4
9	S17, S15, MV8

Table 2. FDR of TGGL with Different Architectures

model architecture	FDR (%)
(16, 32, 64, 128)	69.3
(32, 64, 128, 256)	72.4
(64, 128, 256, 512)	76.5
(16, 16, 32, 32, 64, 64, 128, 128)	81.4
(32, 32, 64, 64, 128, 128, 256, 256)	86.7
(64, 64, 128, 128, 256, 256, 512, 512)	84.2
(16, 16, 16, 32, 32, 32, 64, 64, 64, 128, 128, 128)	84.9
(32, 32, 32, 64, 64, 64, 128, 128, 128, 256, 256, 256)	82.7
(64, 64, 64, 128, 128, 128, 256, 256, 256, 512, 512, 512)	80.1

validation set. The number of layers and channels of the model symbolizes the model capacity. Models with significantly small capacity are prone to underfitting whereas those with considerably large capacity are prone to overfitting. Hence, with the increase of model capacity, FDR presents a trend of first increasing and then decreasing. Finally, the number of layers and channels among the TGGL models was determined to be 8 and (32, 32, 64, 64, 128, 128, 256, 256), respectively, where the graph pooling layer was added between the fourth and fifth STCL.

4.2. RESULTS AND DISCUSSION

To evaluate the performance of our proposed model, we conducted a comparative study with SVM,[7] LSTM,[18] ConvLSTM,[33] and GCN.[29] Additionally, we included the multivariate exponentially weighted moving average PCA and Bayesian method (MEWMA-PCA-BM)[13] as a comparison for the integrated framework. As shown in Table 3, our model, TGGL, outperforms the baseline models in FD. This is because TGGL incorporates prior knowledge of the TE process, allowing the learning process to have a physical meaning and leading to improved FD performance for 21 different types of faults.

For indistinguishable fault types such as 0, 9, and 15, TGGL comprehensively outperforms the baseline model. As shown in Figure 12, the confusion matrix of TGGL illustrates this.

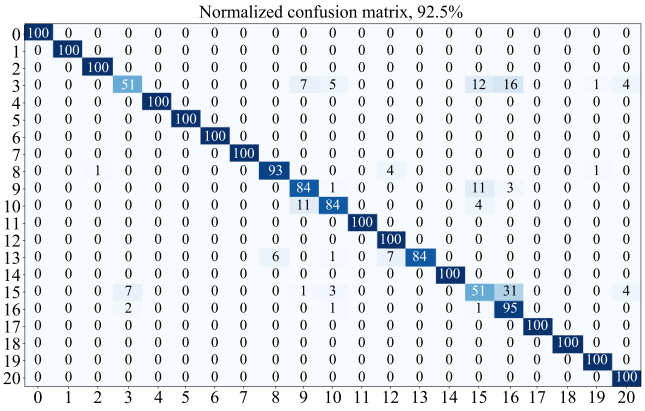


Figure 12. Confusion matrix of TGGL.

Specifically, fault 15, which is the sticking of a condenser cooling water valve, has a negligible effect on the operating conditions and is therefore often not detected as an abnormal condition by general process monitoring methods. However,

Table 3. FDR of Baseline Models and TGGL

type of fault	SVM	MEWMA-PCA-BM	LSTM	ConvLSTM	GCN	TGGL
0	0.78	0.55	0.94	0.93	0.96	**1**
1	0.73	0.90	1	0.98	1	**1**
2	0.71	0.91	1	1	1	**1**
3	0.32	0.10	**0.76**	0.69	0.72	0.52
4	0.80	0.89	1	1	1	**1**
5	0.79	0.93	1	1	1	**1**
6	0.79	0.91	1	1	1	**1**
7	0.72	0.90	1	1	1	**1**
8	0.62	0.88	**0.99**	0.92	0.98	0.93
9	0.45	0.09	0.19	0.32	0.57	**0.84**
10	0.52	0.45	**0.89**	0.80	0.82	0.84
11	0.80	0.80	0.99	0.92	0.96	**1**
12	0.80	0.89	0.99	0.92	1	**1**
13	0.72	0.87	**0.96**	0.82	0.83	0.84
14	0.81	0.51	1	1	1	**1**
15	0.02	0.10	0.06	0.31	0.39	**0.52**
16	0.72	0.63	0.86	0.79	0.92	**0.95**
17	0.76	0.87	0.99	0.96	1	**1**
18	0.72	0.82	0.97	0.96	0.98	**1**
19	0.43	0.13	0.74	0.82	0.89	**1**
20	0.88	0.49	0.99	0.93	0.99	**1**
avg	62.3%	64.8%	87.3%	86.0%	90.5%	92.5%

https://doi.org/10.1021/acs.iecr.2c03628
Ind. Eng. Chem. Res. 2023, 62, 3238−3248

the diagnostic performance of TGGL for this fault is greatly improved when compared to that of baseline models. The confusion matrix shows that the main misclassification for fault 15 in TGGL is as fault 16. Additionally, when TGGL classifies fault 3, some instances of it are classified as faults 15 and 16, which indicate that in order to improve the accuracy of fault 15, the accuracy of fault 3 may be slightly sacrificed. Therefore, for indistinguishable fault types such as 3, the accuracy of TGGL may be slightly lower than that of baseline models.

After verifying the effectiveness of the model, and in keeping with the principles of transparency and reliability, the GNN explainer was used to evaluate the physical consistency of the model's predictions. The GNN explainer was used to identify the key nodes that have the greatest impact on the model's predictions. Additionally, the GNN explainer was trained using the same structure and parameters as TGGL in order to obtain the necessary mask matrix (M) and identify the key nodes of the fault. The GNN explainer can identify key supernodes and subnodes because the topology and pooled graphs exist in TGGL. Furthermore, it can identify the fault loop and corresponding key variables as each supernode corresponds to a loop of multiple subnodes. To prevent the GNN explainer from spreading its attention to too many nodes, only two key nodes were set to be determined.

Taking fault 4 as an example, the GNN explainer was trained with 208 samples, including 89 samples of fault 4, 89 samples of the normal condition, and 30 samples of other faults. The evaluation strategy of the GNN explainer was to first identify the key supernodes and then the corresponding subnodes in order to evaluate the model's logic. Therefore, the key supernodes were first determined based on the pooled graph. Figure 13 shows the mask scores for the mask matrix of the pooled graph.

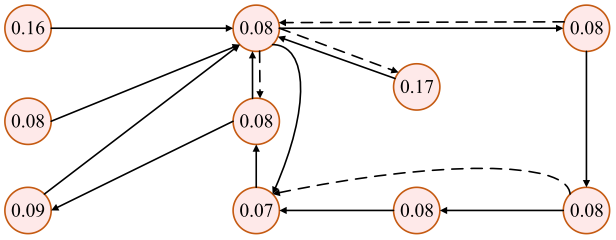


Figure 13. Mask score of the pooled graph in fault 4.

The GNN explainer identified supernodes 4 and 5 as the key nodes according to the mask score. These supernodes correspond to the loops of subnodes (S2, S26, MV1) and (S9, S21, MV10), respectively. By referencing the TE process flowchart, it can be determined that supernodes 4 and 5 control the supply of reactant D and the temperature of the reactor, respectively. In light of the mechanism of the TE process and the nature of fault 4, the two supernodes identified by the GNN explainer can be described as follows.

1. Supernode 4: the inlet temperature of the reactor cooling water affects the output temperature of the reactor cooling water (S21) as well, and fault 4 directly impacts the reactor temperature (S9). In order to balance the changes caused by the fault, the manipulated variable (MV10) uses feedback control when the process variables S9 and S21 change.
2. Supernode 5: the consumption rate of reactant D is at its highest when the reaction rates are equal in eq 8 because all reactions are irreversible exothermic processes. As a result, the consumption rate of reactant D is most sensitive to changes in reactor temperature, and fault 4 directly impacts the reactor temperature. Consequently, the concentration of reactant D and its closely related variables are significantly affected when fault 4 occurs.

In summary, the variables within supernodes 4 and 5, identified by the GNN explainer, are certainly affected by fault 4 and change drastically. To have a more detailed evaluation of the model prediction logic, the GNN explainer was further used to compute the contributions of the six subnodes (S2, S26, MV1, S9, S21, MV10) corresponding to supernodes 4 and 5. Figure 14 illustrates the contributions of the six subnodes.

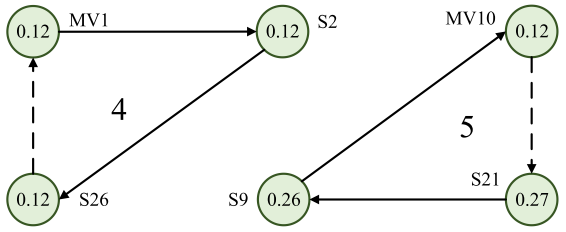


Figure 14. Mask score of the six subnodes in fault 4.

The GNN explainer places more attention on subnodes S9 and S21 and selects them as key variables for model prediction when the number of nodes is limited to 2. This analysis shows that fault 4 directly impacts the reactor temperature (S9) and that the change of the inlet temperature of the reactor cooling water (S21) causes the change of the outlet temperature of the reactor cooling water. Therefore, taking fault 4 as an example, the prediction logic of the model is physically consistent. Figure 15 visualizes the data of S9 and S21, both of which show significant changes when the fault is introduced. This indicates that the underlying physics of TGGL accurately captures the process mechanism.

Given the GNN explainer's ability to evaluate the model's logic, it was used to try and identify the root cause of an unknown fault. For example, taking an unknown fault 17 as an example, Figure 16 illustrates the contribution of supernodes in the pooled graph to the model's predictions.

The GNN explainer places the most attention on supernodes 6 and 9, which correspond to the subnode loops (S6, S7, S8, S10, S24, MV6) and (S15, S17, MV8), respectively. By referencing the flowchart, it can be inferred that supernode 6 controls the flow of the reactor and supernode 9 controls the flow of the stripper. Therefore, it is tentatively determined that fault 17 affects the flow circulation between the reactor and stripper. To further determine the root cause of the failure in more detail, the GNN explainer was used to compute the contributions of the nine subnodes (S6, S7, S8, S10, S24, MV6, S15, S17, MV8) corresponding to supernodes 6 and 9. Figure 17 illustrates the contributions of the nine subnodes.

https://doi.org/10.1021/acs.iecr.2c03628
Ind. Eng. Chem. Res. 2023, 62, 3238−3248

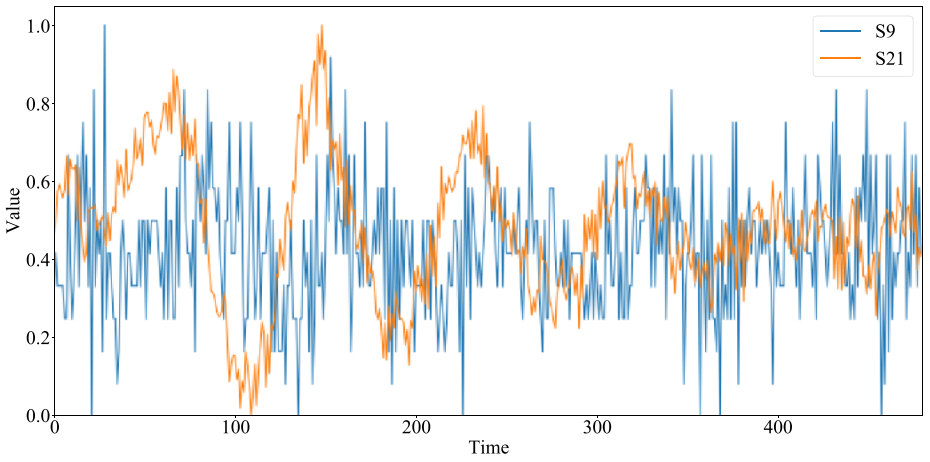


Figure 15. Data visualization for S9 and S21.

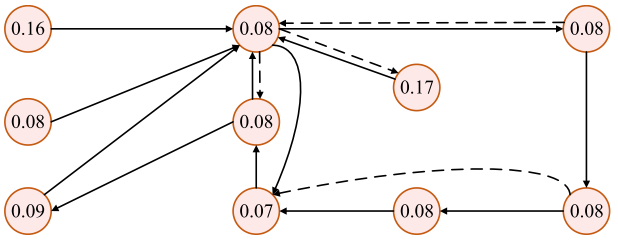


Figure 16. Mask score of the pooled graph in fault 17.

Algorithm 1: The TGGL-based FD framework

Input: input variable $\mathbf{X}_G \in \mathbb{R}^{V \times T}$, adjacency matrix $\mathbf{A}_r$, cluster assignment matrix $\mathbf{R}$, label $\mathbf{Y}$.

Output: prediction $\mathbf{Y}'$, mask $\mathbf{M}$.

Hyperparameters: the number of supernodes K, *epochs_prediction*, *epochs_explainer*, *learning rate*.

Initialize parameters: the SAM parameters ($\boldsymbol{\theta}_{\mathrm{SAM}}$), the STCL parameters ($\boldsymbol{\theta}_{\mathrm{STCL}}$), and the FCL parameters ($\boldsymbol{\theta}_{\mathrm{FCL}}$).
for i =1, 2, …, *epochs_prediction* **do**
  $\mathbf{S} \leftarrow \mathrm{SAM}(\mathbf{X}_G|\boldsymbol{\theta}_{\mathrm{SAM}})$
  $\mathbf{A} \leftarrow \mathbf{S}+\mathbf{A}_r$
  $\mathbf{X}_G \leftarrow \mathrm{STCL}(\mathbf{X}_G, \mathbf{A}|\boldsymbol{\theta}_{\mathrm{STCL}})$
  $\mathbf{A}_{\mathrm{coar}} \leftarrow \mathbf{R}^{\mathrm{T}}\mathbf{A}\mathbf{R}$
  $\mathbf{X}_{\mathrm{coar}} \leftarrow \mathbf{R}^{\mathrm{T}}\mathbf{X}_G$
  $\mathbf{X}_G \leftarrow \mathrm{STCL}(\mathbf{X}_{\mathrm{coar}}, \mathbf{A}_{\mathrm{coar}}|\boldsymbol{\theta}_{\mathrm{STCL}})$
  $\mathbf{Y}' \leftarrow \mathrm{FCL}(\mathbf{X}_G|\boldsymbol{\theta}_{\mathrm{FCL}})$
  $(\boldsymbol{\theta}_{\mathrm{SAM}}, \boldsymbol{\theta}_{\mathrm{STCL}}, \boldsymbol{\theta}_{\mathrm{FCL}}) \leftarrow (\boldsymbol{\theta}_{\mathrm{SAM}}, \boldsymbol{\theta}_{\mathrm{STCL}}, \boldsymbol{\theta}_{\mathrm{FCL}}) + \textit{learning rate} \times (\boldsymbol{\theta}_{\mathrm{SAM}}, \boldsymbol{\theta}_{\mathrm{STCL}}, \boldsymbol{\theta}_{\mathrm{FCL}})$
End
for i =1, 2, …, *epochs_explainer* **do**
  $\mathbf{A}_s \leftarrow \mathbf{A} \circ \mathrm{sigmoid}(\mathbf{M})$
  $\mathbf{X}_G \leftarrow \mathrm{STCL}(\mathbf{X}_G, \mathbf{A})$
  $\mathbf{Y}' \leftarrow \mathrm{FCL}(\mathbf{X}_G)$
  $\mathbf{M} \leftarrow \mathbf{M} + \textit{learning rate} \times \mathbf{M}$
End
Return $\mathbf{Y}'$, $\mathbf{M}$

Subnodes MV8 and S15 receive more attention from the GNN explainer and are selected as key variables for model prediction when the number of nodes is limited to 2. This suggests that fault 17 primarily affects the working state of the stripper and is likely caused by a change in the flow rate of component B (S24). This in turn causes changes in the reactor pressure (S7) and liquid level (S8). Additionally, the reactor feed rate (S16) and reactor discharge valve (MV6) are affected due to the presence of a feedback controller. As the product of the reactor enters the stripper, the liquid level (S15) and the stripper flow (S17) change, and the stripper liquid product flow (MV8) is affected due to the presence of a feedback controller.

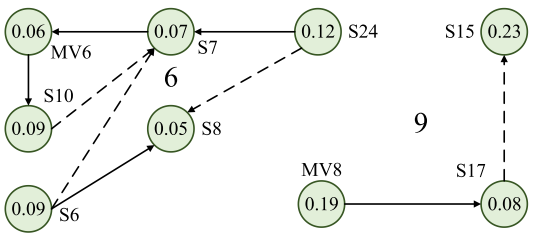


Figure 17. Mask score of the nine subnodes in fault 17.

5. CONCLUSIONS

This study proposes a TGGL-based FD framework that addresses the lack of interpretability and transparency in data-driven FD methods, which often suffer from black-box problems when using deep learning (DL) technology. The TGGL framework embeds process physics knowledge into the model using various components, ensuring the model's physical consistency. The reliability of the knowledge mining process is guaranteed by a loss function, and a topology graph is used to give the model's learning process a physical meaning. The physical consistency of the model's prediction logic is evaluated using the GNN explainer. The results show that the proposed model performs well on indistinguishable faults (faults 3, 9, and 15) due to the incorporation of prior knowledge. The TGGL framework effectively addresses the challenges of knowledge embedding, utilization, and evaluation in DL-based FD tasks, and researchers can replace or add components to create effective and reliable models.

The complexity of the process industry has led to an increase in the likelihood of equipment faults, resulting in poor system performance, an overload of alarms for operators, and, in some cases, catastrophic incidents.[38] An effective FD method can reduce false positives and negatives, and good interpretability is crucial for operators to trust the results. Our proposed method, TGGL, can serve as a useful FD system in the process industry to prevent catastrophic incidents. For instance, after TGGL diagnoses a fault, the operator can trace the root cause and take timely action to prevent loss. Additionally, if faults frequently occur in a particular process flow, operators can consider redesigning this process flow to improve reliability and prevent danger in the industry.

https://doi.org/10.1021/acs.iecr.2c03628
Ind. Eng. Chem. Res. 2023, 62, 3238−3248

AUTHOR INFORMATION

Corresponding Authors

Yi Liu − *Institute of Process Equipment and Control Engineering, Zhejiang University of Technology, Hangzhou 310023, People's Republic of China;* orcid.org/0000-0002-4066-689X; Email: yliuzju@zjut.edu.cn

Yuan Yao − *Department of Chemical Engineering, National Tsing Hua University, Hsinchu 300044, Taiwan;* orcid.org/0000-0002-0025-6175; Email: yyao@mx.nthu.edu.tw

Authors

Mingwei Jia − *Institute of Process Equipment and Control Engineering, Zhejiang University of Technology, Hangzhou 310023, People's Republic of China*

Junhao Hu − *Institute of Process Equipment and Control Engineering, Zhejiang University of Technology, Hangzhou 310023, People's Republic of China*

Zengliang Gao − *Institute of Process Equipment and Control Engineering, Zhejiang University of Technology, Hangzhou 310023, People's Republic of China*

Complete contact information is available at: https://pubs.acs.org/10.1021/acs.iecr.2c03628

Notes

The authors declare no competing financial interest.

ACKNOWLEDGMENTS

This work was supported by the National Natural Science Foundation of China (grant nos. 62022073 and 61873241) and National Science and Technology Council, ROC (grant no. NSTC 111-2221-E-007-002).

NOMENCLATURE

BN	batch normalization
ConvLSTM	convolutional LSTM
DL	deep learning
FCL	fully connected layer
FD	fault diagnosis
FDR	fault diagnosis rate
GCN	graph convolutional network
GNN	graph neural network
LSTM	long short-term memory
PDE	partial differential equation
PINN	physics-informed neural network
ReLU	rectified linear unit
SAM	self-attention mechanism
SCL	spatial convolutional layer
SDG	symbolic directed graph
STCL	spatial−temporal convolution layer
SVMs	support vector machines
TCL	temporal convolutional layer
TE	Tennessee Eastman
TGGL	topology-guided graph learning

SYMBOL

$\mathbf{A}$	adjacency matrix of the graph G
$\mathbf{A}_r$	adjacency matrix of SDG
$\mathbf{A}_{coar}$	coarsening matrix of graph pooling
$\mathbf{B}$	bias of the convolution
C	number of convolutional channels
$\mathbf{C}$	current result of the memory unit
$\mathbf{D}$	degree matrix of the adjacency matrix
$\mathbf{E}$	set of edges
$E_G[.]$	expectation
$\mathbf{F}_t$	result of the forget gate
G	graph
G_S	subgraph of the graph G
$H()$	element-level entropy
$\mathbf{H}_t$	output result of the module at the current moment
h	number of attention heads
$\mathbf{I}_t$	result of the input gate
K	number of supernodes
$\mathbf{K}$	key to facilitate building the attention matrix
$Loss_{dis}$	penalty terms for the mask matrix discreteness
$Loss_{pred}$	cross-entropy loss between the predicted and label classes during training
$Loss_{size}$	penalty terms for the mask matrix value
$MI()$	mutual information
$\mathbf{M}$	matrix that can mask certain nodes and edges
$\mathbf{O}_t$	result of the output gate
$\mathbf{U}$	trainable weight of the GCN
$\mathbf{Q}$	query to facilitate building the attention matrix
$\mathbf{R}$	cluster assignment matrix
$r_{z,v}$	similarity coefficient of the vth variable in the zth state
$\mathbf{S}$	adjacency matrix of the auxiliary graph constructed by SAM
T	time step of observation
$\mathbf{V}$	set of vertices
V_s	variables obtained after feature selection
$\mathbf{W}$	kernel of the convolution
$\mathbf{W}_h^K$	mapping weights of $\mathbf{Q}$
$\mathbf{W}_h^Q$	mapping weights of $\mathbf{K}$
$\mathbf{X}$	process data before feature selection
$\mathbf{X}_G$	graph signal matrix after feature selection
$\mathbf{X}_{coar}$	output of the graph pooling layer
$\mathbf{Xs}$	associated features in the graph signal matrix
$\mathbf{x}_{0,v}$	vth normal sequence
$\mathbf{x}_{z,v}$	vth fault sequence
$\mathbf{Y}$	prediction
Z	numbers of states
z	states of $\mathbf{x}$
$+$	positive effect
$-$	negative effect
$*$	convolution operation
$\cdot$	Hadamard product
ε	infinitely small quantity
$\|\|.\|\|_2$	two-norm
$\mathbf{\Psi}$	size of the convolution kernel

REFERENCES

(1) Venkatasubramanian, V.; Rengaswamy, R.; Yin, K.; Kavuri, S. N. A review of process fault detection and diagnosis. *Comput. Chem. Eng.* **2003**, *27*, 293−311.

(2) Arunthavanathan, R.; Khan, F.; Ahmed, S.; Imtiaz, S. An analysis of process fault diagnosis methods from safety perspectives. *Comput. Chem. Eng.* **2021**, *145*, 107197.

(3) Yin, S.; Ding, S. X.; Xie, H.; Luo, H. A review on basic data-driven approaches for industrial process monitoring. *IEEE Trans. Ind. Electron.* **2014**, *61*, 6418−6428.

(4) Isermann, R. Model-based fault-detection and diagnosis - status and applications. *Annu. Rev. Control* **2005**, *29*, 71−85.

(5) Chen, H.; Jiang, B.; Ding, S.; Huang, B. Data-driven fault diagnosis for traction systems in high-speed trains: A survey,

challenges, and perspectives. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 1700−1716.
(6) Alauddin, M.; Khan, F.; Imtiaz, S.; Ahmed, S. A bibliometric review and analysis of data-driven fault detection and diagnosis methods for process systems. *Ind. Eng. Chem. Res.* **2018**, *57*, 10719−10735.
(7) Yélamos, I.; Escudero, G.; Graells, M.; Puigjaner, L. Performance assessment of a novel fault diagnosis system based on support vector machines. *Comput. Chem. Eng.* **2009**, *33*, 244−255.
(8) Gharahbagheri, H.; Imtiaz, S. A.; Khan, F. Root cause diagnosis of process fault using KPCA and Bayesian network. *Ind. Eng. Chem. Res.* **2017**, *56*, 2054−2070.
(9) Amin, M. T.; Khan, F.; Ahmed, S.; Imtiaz, S. A data-driven Bayesian network learning method for process fault diagnosis. *Proce. Saf. Environ. Prot.* **2021**, *150*, 110−122.
(10) Zhang, Y. W. Enhanced statistical analysis of nonlinear processes using KPCA, KICA and SVM. *Chem. Eng. Sci.* **2009**, *64*, 801−811.
(11) Don, M. G.; Khan, F. Dynamic process fault detection and diagnosis based on a combined approach of hidden Markov and Bayesian network model. *Chem. Eng. Sci.* **2019**, *201*, 82−96.
(12) Amin, M. T.; Khan, F.; Ahmed, S.; Imtiaz, S. A novel data-driven methodology for fault detection and dynamic risk assessment. *Can. J. Chem. Eng.* **2020**, *98*, 2397−2416.
(13) Amin, M. T.; Khan, F.; Imtiaz, S.; Ahmed, S. Robust process monitoring methodology for detection and diagnosis of unobservable faults. *Ind. Eng. Chem. Res.* **2019**, *58*, 19149−19165.
(14) Salmerón, A.; Rumí, R.; Langseth, H.; Nielsen, T. D.; Madsen, A. L. A review of inference algorithms for hybrid Bayesian networks. *J. Artif. Intell. Res.* **2018**, *62*, 799−828.
(15) Wu, H.; Zhao, J. S. Deep convolutional neural network model based chemical process fault diagnosis. *Com. Chem. Eng.* **2018**, *115*, 185−197.
(16) Wu, Q.; Ding, K.; Huang, B. Approach for fault prognosis using recurrent neural network. *J. Intell. Manuf.* **2020**, *31*, 1621−1633.
(17) Zhao, Z.; Wu, S.; Qiao, B.; Wang, S.; Chen, X. Enhanced sparse period-group lasso for bearing fault diagnosis. *IEEE Trans. Ind. Electron.* **2018**, *66*, 2143−2153.
(18) Mirzaei, S.; Kang, J. L.; Chu, K. Y. A comparative study on long short-term memory and gated recurrent unit neural networks in fault diagnosis for chemical processes using visualization. *J. Taiwan Ins. Chem. E.* **2022**, *130*, 104028.
(19) LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436−444.
(20) Huang, B.; Wang, J. Applications of Physics-Informed Neural Networks in Power Systems-A Review. *IEEE Trans. Power Syst.* **2023**, *38*, 572−588.
(21) Raissi, M.; Perdikaris, P.; Karniadakis, G. E. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.* **2019**, *378*, 686−707.
(22) Meng, C.; Seo, S.; Cao, D.; Griesemer, S.; Liu, Y. When physics meets machine learning: A survey of physics-informed machine learning. Mar. **2022**, arXiv:2203.16797.
(23) Shen, S.; Lu, H.; Sadoughi, M.; Hu, C.; Nemani, V.; Thelen, A.; Webster, K.; Darr, M.; Sidon, J.; Kenny, S. A physics-informed deep learning approach for bearing fault detection. *Eng. Appl. Artif. Intel.* **2021**, *103*, 104295.
(24) Karniadakis, G. E.; Kevrekidis, I. G.; Lu, L.; Perdikaris, P.; Wang, S.; Yang, L. Physics-informed machine learning. *Nat. Rev. Phys.* **2021**, *3*, 422−440.
(25) Guo, S.; Agarwal, M.; Cooper, C.; Tian, Q.; Gao, R. X.; Guo, W. G.; Guo, Y. B. Machine learning for metal additive manufacturing: Towards a physics-informed data-driven paradigm. *J. Manuf. Syst.* **2022**, *62*, 145−163.
(26) Wu, Z.; Pan, S.; Chen, F.; Long, G.; Zhang, C.; Philip, S. Y. A comprehensive survey on graph neural networks. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *32*, 4−24.
(27) Kramer, M. A.; Palowitch, B. L., Jr A rule-based approach to fault diagnosis using the signed directed graph. *AIChE J.* **1987**, *33*, 1067−1078.
(28) Wu, D.; Zhao, J. Process topology convolutional network model for chemical process fault diagnosis. *Process Saf. Environ. Prot.* **2021**, *150*, 93−109.
(29) Kip, F. T. N.; Welling, M. Semi-supervised classification with graph convolutional networks. In *Proceedings of the 2016 International Conference on Learning Representations*; OpenReview.net: San Juan: Puerto Rico, 2016;.
(30) Yu, X.; Tang, B.; Zhang, K. Fault diagnosis of wind turbine gearbox using a novel method of fast deep graph convolutional networks. *IEEE Trans. Instrum. Meas.* **2021**, *70*, 1−14.
(31) Chen, Z.; Xu, J.; Peng, T.; Yang, C. Graph convolutional network-based method for fault diagnosis using a hybrid of measurement and prior knowledge. *IEEE Trans. Cybern.* **2022**, *52*, 9157−9169.
(32) Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; Polosukhin, I. Attention is all you need. In *Proceedings of the 2017 Advances in Neural Information Processing Systems*; MIT Press: Long Beach Ca, 2017;.
(33) Shi, X.; Chen, Z.; Wang, H.; Yeung, D. Y.; Wong, W. K.; Woo, W. C. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In *Proceedings of the 2015 Advances in Neural Information Processing Systems*; MIT Press: Montreal, 2015;.
(34) Ying, Z.; Bourgeois, D.; You, J.; Zitnik, M.; Leskovec, J. Gnnexplainer: Generating explanations for graph neural networks. In *Proceedings of the 2019 Advances in Neural Information Processing Systems*; MIT Press: Vancouver, 2019; pp 9240−9251.
(35) Downs, J. J.; Vogel, E. F. A plant-wide industrial process control problem. *Comput. Chem. Eng.* **1993**, *17*, 245−255.
(36) Ma, Y.; Wang, S.; Aggarwal, C. C.; Tang, J. Graph convolutional networks with eigenpooling. *Proceedings of the 2019 Knowledge Discovery & Data Mining*; ACM Press: Vancouver, 2019, pp 723−731.
(37) Rieth, C. A.; Amsel, B. D.; Tran, R.; Cook, M. B. Additional Tennessee Eastman process simulation data for anomaly detection evaluation. *Harvard Dataverse* **2017**, *1*, 2017.
(38) Goel, P.; Datta, A.; Mannan, M. S. Industrial alarm systems: Challenges and opportunities. *J. Loss Prev. Process Ind.* **2017**, *50*, 23−36.

https://doi.org/10.1021/acs.iecr.2c03628
Ind. Eng. Chem. Res. 2023, 62, 3238−3248