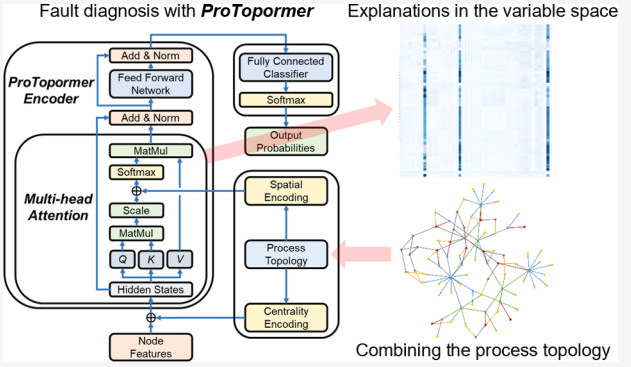
I&EC research
Industrial & Engineering Chemistry Research
pubs.acs.org/IECR
Article

# ProTopormer: Toward Understandable Fault Diagnosis Combining Process Topology for Chemical Processes

Deyang Wu, Xiaotian Bi, and Jinsong Zhao*

Cite This: *Ind. Eng. Chem. Res.* 2023, 62, 8350−8361
Read Online
ACCESS | Metrics & More | Article Recommendations

**ABSTRACT:** In modern chemical processes, fault detection and diagnosis (FDD) is a key part of abnormal situation management (ASM). As researchers continue to improve the fault diagnosis performance of different models, the emphasis on model interpretability and explainability studies has increased in recent years. In this paper, a novel model, ProTopormer, was proposed for fault diagnosis of chemical processes. Self-attention mechanism and process topology knowledge were fully combined to achieve high model performance and good interpretability. Experiments on Tennessee Eastman process showed that the model achieved a high diagnosis rate and a low false alarm rate. Attention weights were visualized and quantitatively analyzed to identify key variables for fault diagnosis, which showed strong explanations for the root causes of different faults.






## 1. INTRODUCTION

For chemical processes, fault detection and diagnosis (FDD) is an important part of abnormal situation management (ASM). The role of FDD is to locate the cause of a failure and initiate subsequent sessions in a timely manner when the process exceeds the control capability of the distributed control systems (DCS). This can serve to reduce the likelihood of accidents and protect the safety, health, and environment of the public. However, real-time FDD is getting more and more difficult for increasingly complex chemical processes.[1] For decades, more and more researchers have been focusing on this area to build more intelligent FDD systems. This also coincides with the trend of developing smart manufacturing today.

From a methodological perspective,[1] FDD models are classified into three categories: quantitative model-based, qualitative model-based, and process history-based. Process history-based models are further classified into qualitative and quantitative models. The latter is also commonly termed as data-driven models. The FDD methods, based on first principles and expert knowledge, require a deep understanding of the process, while the time and labor costs of modeling are relatively high. However, a data-driven approach can rely on data alone to build models in a fast and cost-effective manner. Due to the advancement of the big data era, the accessibility of process data continues to increase, while the field of machine learning (ML) is rapidly developing. Data-driven models are getting more and more attention.

In addition to the early models based on statistical learning methods, such as principal component analysis (PCA), independent component analysis (ICA), partial least squares (PLS), canonical variate analysis (CVA), support vector machine (SVM), and so on, FDD methods based on deep learning models have become more and more popular in the last decade. These used deep learning models include hierarchical deep neural network (HDNN),[2] deep belief network (DBN),[3] convolutional neural network (CNN),[4] and recurrent neural network (RNN).[5] Besides supervised learning, there are also semisupervised and unsupervised learning models used for fault diagnosis.[6] Among them, autoencoders are widely used for fault detection.[7,8]

To compare different models' performance, the Tennessee Eastman (TE) process[9] is usually utilized by researchers as a benchmark. This is a simulation program modeled from a real chemical process. The above machine learning-based approaches continue to reach new highs in fault diagnosis accuracy and are much higher than earlier data-driven approaches. At the cost of high accuracy, the huge number of parameters and the complexity of the networks make it

Received: January 27, 2023
Revised: May 5, 2023
Accepted: May 8, 2023
Published: May 17, 2023





https://doi.org/10.1021/acs.iecr.3c00206
*Ind. Eng. Chem. Res.* 2023, 62, 8350−8361

increasingly difficult to understand these models. The lack of transparency and interpretability makes it difficult to find a basis for selecting the models. This makes it difficult to predict the behavior of the model, whether it is a success or a failure. The current status of ML is more like alchemy, a collection of ad hoc methods.[10] Innovation is needed to integrate data analytics tools with fundamental knowledge to create a robust and scalable solution for industrial processes.[11]

Some researchers have already started to study the interpretability and explainability of deep learning models applied for FDD. The self-attention mechanism, widely used in the field of deep learning, can help us understand the importance of some of the parameters in the networks.[8,12] Some researchers integrate first principles or expert knowledge into the deep learning model itself to improve interpretability and performance.[13,14] Depending on the modification, these models can also be considered as hybrid models. Some researchers have tried to analyze the reasons why deep learning models make specific inferences using explainability and visualization techniques.[15] However, these models do not yet have the advantage of incorporating process knowledge and being interpretable at the same time. The former is the basis for rational modeling and high performance. The latter is important for people to understand and trust the models, thus important for the promotion of applications, especially in high-risk scenarios.

In this paper, a new hybrid model, ProTopormer, was proposed for fault diagnosis based on Graphormer.[16] Based on the deep learning models, ProTopormer incorporates the knowledge of process topology and takes advantage of the self-attention mechanism. The model has both high performance and good interpretability. The major contributions of this work are summarized as follows:

(1) A novel model ProTopormer and the corresponding method were proposed for fault detection and diagnosis.
(2) Self-attention mechanism and process topology knowledge were fully combined to achieve high model performance and good interpretability.
(3) The decision basis for diagnosing faults in the model was explained through visualization of the self-attention mechanism.

The rest of the paper is organized as follows: Section 2 introduces the basic theories about Transformer and Graphormer. Section 3 introduces the ProTopormer model and the corresponding method for fault detection and diagnosis. Section 4 is a case study on the TE process that showed high performance and good interpretability of ProTopormer. Finally, conclusions are drawn in Section 5.

## 2. TRANSFORMER AND GRAPHORMER

The attention mechanism is an important invention in deep learning. It was first proposed to solve the bottleneck in improving the performance of the basic encoder−decoder architecture for neural machine translation.[17] It simply used weighted sum of vectors to aggregate information. Different attention weights assign different importance to each vector as if humans focus their attention on certain places. Since the attention mechanism was proposed, it has gradually become the most widely used module in the field of natural language processing (NLP). Its successful applications have even expanded to fields such as computer vision (CV) and speech signal processing.

There are many different implementations of the attention mechanism, such as self-attention,[18] multidimensional attention,[19] and key value attention.[20] Based on these works, a landmark network architecture, the Transformer, was proposed based solely on attention mechanisms, dispensing with recurrence and convolutions entirely.[21] The traditional CNN and RNN architectures are discarded in Transformer, and the whole network is only composed of self-attention modules and feed-forward neural networks. In Transformer, both the encoder and the decoder are stacked with several identical layers. Each layer in the encoder has two sublayers. One is a multihead self-attention module, and the other is a simple position-wise fully connected network. The residual connection[22] is applied to each sublayer, followed by layer normalization.[23] Both techniques make deep neural networks easier to train until convergence.

The multihead self-attention mechanism is the essence of Transformer. An attention function in transformer is called scaled dot-product attention. Let $H = [h_1^T, h_2^T, \cdots, h_n^T]^T \in \mathbb{R}^{n\times d}$ denote the input matrix to the attention function. Each row vector $h_i \in \mathbb{R}^{1\times d}$ of the input matrix represents the hidden embedding at the position $i$. With three weight matrices $W_Q \in \mathbb{R}^{d\times d_K}$, $W_K \in \mathbb{R}^{d\times d_K}$, and $W_V \in \mathbb{R}^{d\times d_V}$, $H$ is projected to $Q = [q_1^T, q_2^T, \cdots, q_n^T]^T \in \mathbb{R}^{n\times d_K}$, $K = [k_1^T, k_2^T, \cdots, k_n^T]^T \in \mathbb{R}^{n\times d_K}$, and $V = [v_1^T, v_2^T, \cdots, v_n^T]^T \in \mathbb{R}^{n\times d_V}$. Then, the output of the attention function is calculated as

$$\text{attention}(Q, K, V) = \text{softmax}(A)V \tag{1}$$

$$A = \frac{QK^T}{\sqrt{d_K}}, \quad Q = HW_Q, \quad K = HW_K, \quad V = HW_V \tag{2}$$

The attention function can be described as mapping a query and a set of key value pairs to an output. $Q$, $K$, and $V$ denote the stacked query vectors, key vectors, and value vectors, respectively. Then, $A$ can be regarded as a matrix capturing the similarity between query vectors and key vectors. For convenience and convention, generally, we let $d = d_K = d_V$.

In the multihead attention mechanism, information can be extracted from different subspaces. This can help with a wide range of downstream tasks. The heads can also be called parallel attention layers. This concept has some similarity with the channels of a CNN layer. The multihead attention can be calculated as

$$\text{MultiHeadAttention}(Q, K, V) = \text{Concat}(\text{head}_1, \text{head}_2, \cdots, \text{head}_h)W^O \tag{3}$$

$$\text{head}_i = \text{attention}(QW_i^Q, KW_i^K, VW_i^V) \tag{4}$$

$h$ denotes the number of the attention head. $W_i^Q \in \mathbb{R}^{d_{\text{model}}\times d_K}$, $W_i^K \in \mathbb{R}^{d_{\text{model}}\times d_K}$, $W_i^V \in \mathbb{R}^{d_{\text{model}}\times d_V}$, and $W^O \in \mathbb{R}^{hd_V\times d_{\text{model}}}$. For convenience and convention, generally, we let $d_{\text{model}}/h = d_K = d_V$.

Although it has been successful on many tasks done by CNNs and RNNs, Transformer has not been as well adapted to data in graph domains as graph neural networks (GNNs). Semistructured data like graphs are much more common in the


https://doi.org/10.1021/acs.iecr.3c00206
Ind. Eng. Chem. Res. 2023, 62, 8350−8361

real world, such as citation networks, molecular structures, etc. Graphormer was then proposed based on the standard Transformer architecture to make it perform well for graph representation learning.[16] The key design of Graphormer was three structural encodings as shown in Figure 1. These encodings can incorporate structured information from the graph into the model.

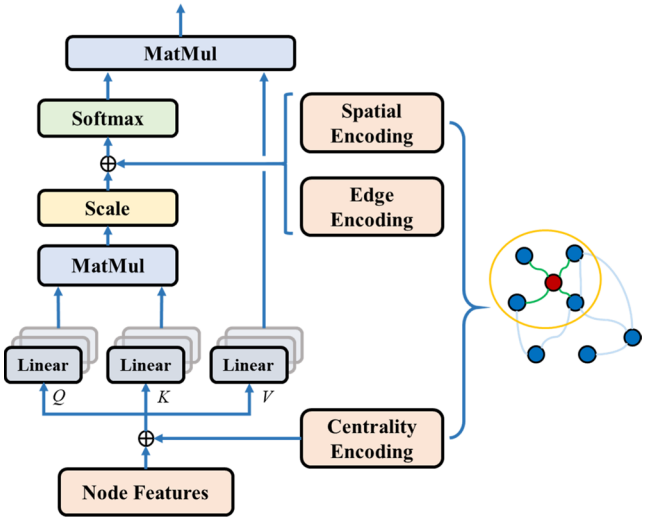



**Figure 1.** Attention mechanism in Graphormer.

The centrality encoding is the information used to describe the importance of nodes in a graph, which is an important signal for graph representation. In Graphormer, the centrality encodings are added to the original input vectors (or node features in graph-structured data) before the vectors are input to the model. The centrality encoding is calculated with nodes' degrees

$$h_i^{(0)} = x_i + z^-_{\deg^-(v_i)} + z^+_{\deg^+(v_i)} \tag{5}$$

$h_i^{(0)}$ denotes the node feature of the node $v_i$ in the graph-structured data. It is also the original vector input to the model. $z^-, z^+ \in \mathbb{R}^d$ are learnable vector embeddings that can be calculated with the in-degree $\deg^-$ and out-degree $\deg^+$ of the node $v_i$. The centrality encoding infuses the node feature with information about the node's importance and complements the model's ability to capture the relationships between nodes.

The spatial encoding is used to describe the global position of a node and the relationship of this node with other non-neighboring nodes. Graphormer uses the function $\phi(v_i, v_j)$: $V \times V \to \mathbb{R}$ to describe the distance between any two nodes $v_i, v_j$ in a graph.

$$(v_i, v_j) = \begin{cases} \mathrm{SPD}(v_i, v_j), & \text{if } v_i, v_j \text{ connected} \\ -1, & \text{else} \end{cases} \tag{6}$$

SPD means the distance of the shorted path between $v_i$ and $v_j$. For every element $A_{i,j}$, a learnable scalar is added to it as a bias term in the self-attention function.

$$A_{i,j} = \frac{(h_i W_Q)(h_j W_K)^{\mathrm{T}}}{\sqrt{d}} + b_{\phi(v_i,v_j)} \tag{7}$$

$b_{\phi(v_i,v_j)}$ is a learnable scalar specified by $\phi(v_i,v_j)$, and it is shared across all of the layers in Graphormer. The spatial encoding provides a very concise way to consider the relationship of a node with all other nodes when utilizing the self-attention mechanism. This gives Graphormer a global view and embeds the graph structural information into the model.

The edge encoding tries to better encode edge features into attention layers. It is specified as another bias term $c_{i,j}$ in the self-attention function to calculate element $A_{i,j}$. Since edge features are not involved in graph-structured data in this paper, they are not discussed further here.

## 3. PROCESS FAULT DIAGNOSIS WITH PROTOPORMER

### 3.1. Input Data Structure with Process Topology.

When we use machine learning methods for process fault diagnosis, the task is often defined as a classification problem. The input sample is considered as a matrix possessing two dimensions: time and variables.

$$X_t = \begin{bmatrix} x_{1,t} & x_{1,(t-1)} & \cdots & x_{1,(t-w+1)} \\ x_{2,t} & x_{2,(t-1)} & \cdots & x_{2,(t-w+1)} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n,t} & x_{n,(t-1)} & \cdots & x_{n,(t-w+1)} \end{bmatrix} \in \mathbb{R}^{n\times w} \tag{8}$$

where $n$ is the number of observed variables and $w$ is the number of observations for each variable. $X_t$ is essentially a stack of $n$ time series with the length of time window $w$.

Such data structure ignores that the order of variables is uncertain in the variable dimension. Further, it does not consider the spatial topology of the observed variables at all. These variables are collected from different locations in the actual process. Therefore, the process topology should be taken into account as a part of the input data. Mathematically, the process topology can be abstracted as a graph $G = (V, E)$, where $V$ is the set of nodes and $E$ is the set of edges connecting different nodes. Nodes correspond to different observed variables, while edges correspond to the physical relationships of different variables in the process.

The key is how to construct a mathematical graph from a chemical process topology. In this paper, the steps proposed in ref 14 is used to transform a process topology into a graph, which considers both stream flows and control loops in the process. The steps are described as follows:

- Step 1: Figure out all of the unit operations and streams that we concern about. Create nodes for every unit operation and stream. These nodes are called unit operation nodes and stream nodes respectively.
- Step 2: Figure out all of the corresponding measurements of these unit operations and streams in step 1. Create nodes for every measurement. These nodes are called measurement nodes.
- Step 3: Create directed edges pointing from every unit operation node or stream node to the corresponding measurement node.
- Step 4: For every stream node, create a directed edge pointing from the upstream unit operation node to the stream node, and a directed edge pointing from the stream node to the downstream unit operation node.
- Step 5: Figure out all of the process variables and manipulated variables of concerned control loops.


https://doi.org/10.1021/acs.iecr.3c00206
Ind. Eng. Chem. Res. 2023, 62, 8350−8361

Process variables are included in the measurement nodes in step 2. Create nodes for every manipulated variable. These nodes are called manipulated variable node. For every control loop, create an abstract node called controller node.

- Step 6: In a control loop, create a directed edge pointing from the corresponding measurement node to the controller node, and a directed edge pointing from the controller node to the corresponding manipulated variable node.
- Step 7: For simplicity of the graph, delete the stream nodes that have no corresponding measurements and then connect the upstream and downstream unit operation nodes with a directed edge directly.

A detailed application example of the case study on the Tennessee Eastman process can be found in Section 4.1. Following the steps above, a process graph $G$ can be obtained from a chemical process, which can be further formulated as an adjacent matrix $A^G$.

To enhance the pure data matrix $X_t$ with a process graph, we define a graph-structured data sample $X_t^G$ as

$$X_t^G = (X_t, A^G) \tag{9}$$

$X_t$ is a stack of time series of different variables (also a stack of node features in the process graph) at timestep $t$ mentioned above. And adjacent matrix of the process graph, $A^G$, describes the relationships of different nodes in the graph. With the data sample $X_t^G$, more information about the process topology can be integrated with pure data matrix

**3.2. Structure of ProTopormer.** General deep learning networks such as CNN and RNN have difficulty in handling the above graph-structured data, but ProTopormer can make good use of their characteristics. The structure of ProTopormer is shown in Figure 2. The calculation of multihead attention in eq 3 is also described in detail. MatMul means matrix multiplication. Scale means division with a scaling factor. Add&Norm means addition and layer normalization.

A centrality encoding is first added to the input vectors to distinguish the importance of different nodes. For an observed variable and the corresponding node in $G$, the input time series or node feature $x_i$, a bias is added to the input vector

$$h_i^{(0)} = x_i + z_{\deg(v_i)} \tag{10}$$

where $z$ is a learnable embedding specified by the degree of node $v_i$. Since $G$ is an undirected graph, there is no difference between out-degree and in-degree.

ProTopormer mainly includes two parts: the encoder and the classifier. The encoder consists of multiple layers with the same structure. The number of layers is set to 3 in this paper and can be adapted to the size of the problem. Each layer of the encoder has two sublayers. One sublayer includes a multihead self-attention function and a layer normalization function. A spatial encoding as eq 7 is used to integrate process topology knowledge into the self-attention mechanism, thus into the network structure. The other sublayer includes a feed-forward network and a layer normalization function. For a ProTopormer encoder with $L$ layers, the $l$th layer's hidden states $H^{(l)} = [h_1^{(l-1),\mathrm{T}}, h_2^{(l-1),\mathrm{T}}, \cdots, h_n^{(l-1),\mathrm{T}}]^{\mathrm{T}}$

$$H'^{(l)} = \mathrm{MultiHeadAttention}(\mathrm{LayerNorm}(H^{(l-1)})) + H^{(l-1)} \tag{11}$$

LayerNorm means a layer normalization function and MultiHeadAttention means a multihead attention function as shown in eq 3. $H^{(l-1)}$ means the output of the $(l-1)$th layer or the input of the $l$th layer of ProTopormer encoder. $H'^{(l)}$ means the intermediate hidden states.

$$H^{(l)} = \mathrm{FeedForwardNet}(\mathrm{LayerNorm}(H'^{(l)})) + H'^{(l)} \tag{12}$$

FeedForwardNet is a simple feed-forward network. $H^{(l)}$ means the output of the $l$th layer of ProTopormer encoder. The hidden states of the $L$th layer or the output of the encoder is $H^{(L)}$.

$H^{(L)}$ is then flattened to a vector as the input of the classifier of ProTopormer. The classifier is a two-layer feed-forward network.

$$z^{(0)} = \mathrm{flatten}(H^{(L)}) \tag{13}$$

Flatten function is used to unstack the hidden state matrix $H^{(L)}$ of the encoder as a vector, which is prepared as the input of the classifier.

$$z^{(1)} = \mathrm{dropout}(p, \mathrm{ReLu}(\mathrm{FeedForwardNet}(\mathrm{LayerNorm}(z^{(0)})))) \tag{14}$$

After a layer normalization function, $z^{(0)}$ is input to the feed-forward classifier and then activated with a ReLU function. To mitigate the overfitting problem, the dropout technique is used with a dropout rate of $p$.

$$z^{(2)} = \mathrm{Dropout}(p, \mathrm{FeedForwardNet}(z^{(1)})) + z^{(0)} \tag{15}$$

$z^{(1)}$ is then input to the second layer of feed-forward classifier. The dropout technique and residual connection are also used for an easier training process.

The last layer of the classifier has $c + 1$ neurons corresponding to the number of classes of the diagnosis or classification task. It includes one normal state and $c$ faulty states of the process defined in advance. $z^{(2)} = [z_0, z_1, \cdots, z_c]^{\mathrm{T}}$ is then input to a Softmax layer to get a probability distribution

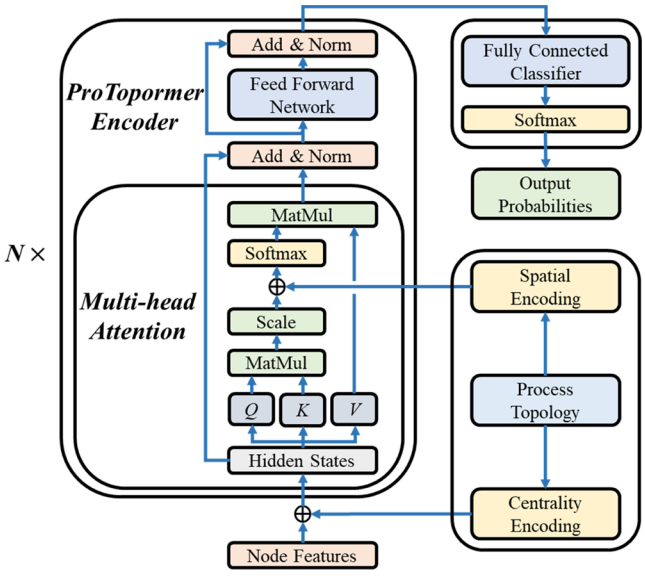



**Figure 2.** ProTopormer structure.



https://doi.org/10.1021/acs.iecr.3c00206
Ind. Eng. Chem. Res. 2023, 62, 8350−8361

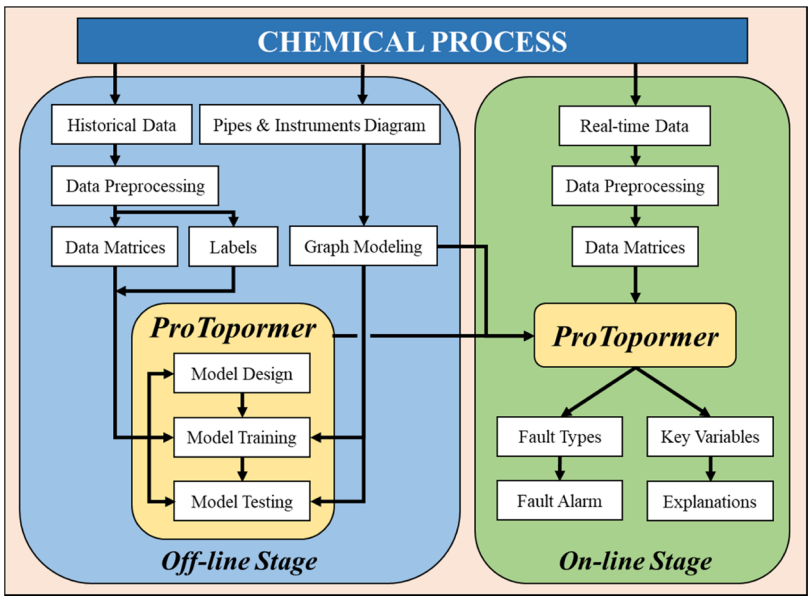



**Figure 3.** Framework of fault diagnosis method based on ProTopormer.

$$\sigma(z_i) = \frac{e^{z_i}}{\sum_j e^{z_j}} \ (i = 0,\ 1, \cdots,\ c) \tag{16}$$

The cross-entropy loss function is finally utilized to calculate the loss for backward propagation.

The multi-layer structure gives ProTopormer nonlinear fitting capabilities and the multihead self-attention mechanism provides the entry to visualize and analyze the weights of the network. Residual connections, layer normalization, and dropout techniques are applied to each sublayer to mitigate the problem of gradient vanishing in deep networks. The centrality encoding and spatial encoding embed the information of process based on variables' importance and relationships. In other words, they work as bridges to connect the data-driven deep network with process topology information, which achieves a knowledge fusion.

### 3.3. Fault Diagnosis Method Based on ProTopormer.

The fault diagnosis methods based on ProTopormer are divided into off-line stage and on-line stage as shown in Figure 3. The procedures are described in detail as below.

*3.3.1. Off-Line Stage.* Step 1: Historical process data in normal state and different faulty states are collected for ProTopormer model building. Then, the data are normalized with the mean and standard deviation of every variable in normal state.

Step 2: The historical data are recognized and labeled with normal or different faulty states. A graph is constructed according to the chemical process topology.

Step 3: Preprocessed historical data are cut into data matrices $X_t \in \mathbb{R}^{n\times w}$, and every data matrix is labeled with a one-hot encoded vector $\boldsymbol{y}$ according to their faulty type. The graph-structured data sample is $X_t^G$. Dataset $\{X_t^G, \boldsymbol{y}\}^N$ is divided into training dataset and testing dataset then.

Step 4: The ProTopormer model is built as the structure shown in Figure 2. The number of encoder layers and the dimension of node embedding should be adapted to the scale of the problem.

Step 5: The model is trained with the training dataset and tested with the testing dataset.

Step 6: The number of encoder layers and the dimension of node embedding should be modified until the testing results are satisfactory. Otherwise, the model should be saved for further application at the on-line stage.

*3.3.2. On-Line Stage.* Step 1: The ProTopormer model saved at the off-line stage is loaded and prepared for real-time fault diagnosis.

Step 2: The real-time process data are collected continuously from the process. Then, the data are normalized with the same mean and standard deviation at the off-line stage.

Step 3: The preprocessed real-time data are cut into data matrices at the off-line stage.

Step 4: Data matrices and the constructed graph form data samples $X_t^G$ as eq 9, which are then input into the loaded ProTopormer model.

Step 5: The ProTopormer model outputs the diagnosis results and key variables related to the diagnosis. If a fault is identified, the model will give an alarm about the faulty state. The key variables can be further used to give explanations about the diagnosis result.

## 4. CASE STUDY ON TENNESSEE EASTMAN PROCESS

### 4.1. Dataset Preparations and Model Training.

Tennessee Eastman process is a simulation program adapted from a real chemical process.[24] It has been utilized as a test benchmark for many studies such as process control and fault diagnosis. The process mainly consists of five unit operations: a reactor, a condenser, a gas−liquid separator, a stripper, and a compressor. This simulation program can generate time series of 52 nonzero variables. They include 41 measurements (real-time or chemical analysis data) and 11 manipulated variables. Even though a revision of the program added more observable variables,[25] this paper used the earliest set of variables.

During the simulation, 20 types of predefined disturbances (IDV) can be introduced to the process and lead to abnormal states. This can provide enough data to study the fault diagnosis performance of different models. Training and testing data are generated with reference to the method in ref 4. In normal state, the process was simulated for 3750 h steadily. For disturbances except IDV 6, each type of


https://doi.org/10.1021/acs.iecr.3c00206
Ind. Eng. Chem. Res. 2023, 62, 8350−8361

disturbance was introduced to the process and simulated for 20 h after an 8 h simulation in a normal state. As for IDV 6, the simulation lasted only 7 h after the disturbance due to early stop. Data generated from the TE process in different states were all sampled, and the frequency is 3 min. At any moment, we used time series of these 52 variables with a 1 h time window for fault diagnosis. Thus, every time series has 20 data points and the input matrix $X_t \in \mathbb{R}^{52\times 20}$. The label or faulty type was set to the number of IDV, and 0 denoted normal state.

The simulation in different states was run for 10 times with different random seeds. Eight runs were selected for model training and the other two runs for testing. Data generated in the normal state were divided by the same ratio of 4 to 1. All of the data should be normalized, sliced into matrices, and labeled by the fault type for model training and testing (Table 1).

**Table 1. Simulation Dataset for Training and Testing**

| fault types | simulation length (training)/h | number of data matrices (training) | simulation length (testing)/h | number of data matrices (testing) |
|---|---|---|---|---|
| IDV 1−5 and 7−20 | 20 × 19 × 8 | 60 800 | 20 × 19 × 2 | 15 200 |
| IDV 6 | 7 × 1 × 8 | 1120 | 7 × 1 × 2 | 280 |
| normal | 3000 | 59 981 | 750 | 14 981 |
| total | 6096 | 121 901 | 1524 | 30 461 |

The graph construction of the process topology requires special attention. It follows the steps proposed in ref 14, which is described in Section 3.1.

There are five types of nodes in the graph: measurement nodes, manipulated variable nodes, unit operation nodes, stream nodes, and controller nodes. The first two types of nodes correspond to the measurements and manipulated variables observed in the chemical process with real numbers. The other three types are intermediate nodes that help interconnect other variable nodes with as few edges as possible. They are also called abstract nodes because there are no observed variables corresponding to these nodes. In the calculation, they need to be initialized first. Following the steps above, TE process can be modeled as an undirected graph with 82 nodes shown in Figure 4, in which different kinds of nodes have different colors

With the TE graph, the datasets are completed as $\{X_t^G, y\}^{N_{\text{train}}}$ and $\{X_t^G, y\}^{N_{\text{test}}}$.

The ProTopormer model utilizes a three-layer encoder and a two-layer feed-forward network as the classifier. The embedding dimension in the encoder is set to 32 from end to end. And the multihead attention mechanism has eight heads. Two layers in the classifier have 256 and 21 neurons, respectively. The model was trained for 200 epochs with a batch size of 64 and a learning rate of 0.0001.

**4.2. Fault Diagnosis Performance of ProTopormer.** ProTopormer was trained well with the training dataset and used the testing dataset to evaluate its fault diagnosis performance. Two indicators, fault diagnosis rate (FDR) and accurate classification rate (ACR) are used. FDR, also called true positive rate, calculates how many samples are correctly classified as the right faulty type. It is an indicator to show the diagnosis performance for different types of faults. ACR is the total classification accuracy of this fault diagnosis model considering all of the testing samples. It is also a class-weighted average of FDR to evaluate the general model performance.

$$\text{FDR}_c = \frac{N_{\text{TP}}^c}{N_{\text{TP}}^c + N_{\text{FN}}^c} = \frac{N_{\text{TP}}^c}{N_{\text{test}}^c} \tag{17}$$

$$\text{ACR} = \frac{\sum_c N_{\text{TP}}^c}{\sum_c N_{\text{TP}}^c + \sum_c N_{\text{FN}}^c} = \frac{\sum_c N_{\text{TP}}^c}{\sum_c N_{\text{test}}^c} = \frac{\sum_c N_{\text{TP}}^c}{N_{\text{test}}} \tag{18}$$

TP and FN stand for true positive samples and false negative samples, respectively.

The testing ACR was changed with the training epochs as shown in Figure 5. As training progressed, the model performance improved continuously until convergence.

The confusion matrix for the testing dataset is shown in Figure 6. Each row of the confusion matrix shows the classification results for a specific type of fault. Thus, the diagonal values represent FDR for each type of fault. Except for a few types of faults, the matrix presents good fault diagnosis results. Faults 9 and 15 are often not well classified because they are particularly easy to confuse with the normal state.

In terms of diagnosis results for each type of fault, ProTopormer's performance is compared with relevant studies in Table 2. The comparison includes deep learning models based on CNN, RNN, and GCN, as well as fault diagnosis methods that also use graph structures. First, for samples in the normal state, our model achieves the highest classification accuracy of 0.9927. This is a very important metric, which shows that the model has only a 0.73% chance of false positives in the normal state. A low false alarm rate is the basis for trust in this fault diagnosis method. Because frequent false alarms will weaken the credibility of the model. This is where our model has a sufficient advantage.

Among the 20 different fault classifications, 9 types achieve an accuracy of 0.98 or higher, 12 types achieve 0.95 or higher, and 15 types achieve 0.9 or higher. Compared with other models, our model achieves the highest classification accuracy in the classification of normal state and 6 types of states (fault 1, 4, 6, 7, 11, 16). Particularly, it achieves far better results than other models in the diagnosis of fault 16. Another 6 types of faults (fault 2, 5, 8, 12, 14, 19) are classified with no more than 1% difference from the highest accuracy, achieving close diagnosis performance.

Table 2 shows that the FDR of fault 15 is very low for ProTopormer. This is determined by the nature of fault 15 itself. This fault is defined as sticking of the condenser cooling water valve. Due to the relative stability of the condenser's state, its cooling water outlet temperature has not changed significantly during the simulation. This makes fault 15 has very little effect on the operation state. It is particularly difficult to distinguish fault 15 from the normal state samples in terms of the fluctuations of the process variables. Since fault 15 is very similar to the normal state, there is doubt about whether it is a real faulty state of the TE process. So, there are many researchers who have disregarded this so-called fault 15.

In general, there is a trade-off between the magnitude of the FDR of fault 15 and that of the normal state samples. An increase in one of the FDRs may lead to a decrease in the other. This is also evident from the test results of the models based on CNN, RNN, and GCN. The outstanding advantage of ProTopormer is the extremely high FDR of the normal state, which implies a very low false alarm rate. Of course, the cost is that samples of fault 15 are more easily ignored and



https://doi.org/10.1021/acs.iecr.3c00206
Ind. Eng. Chem. Res. 2023, 62, 8350−8361

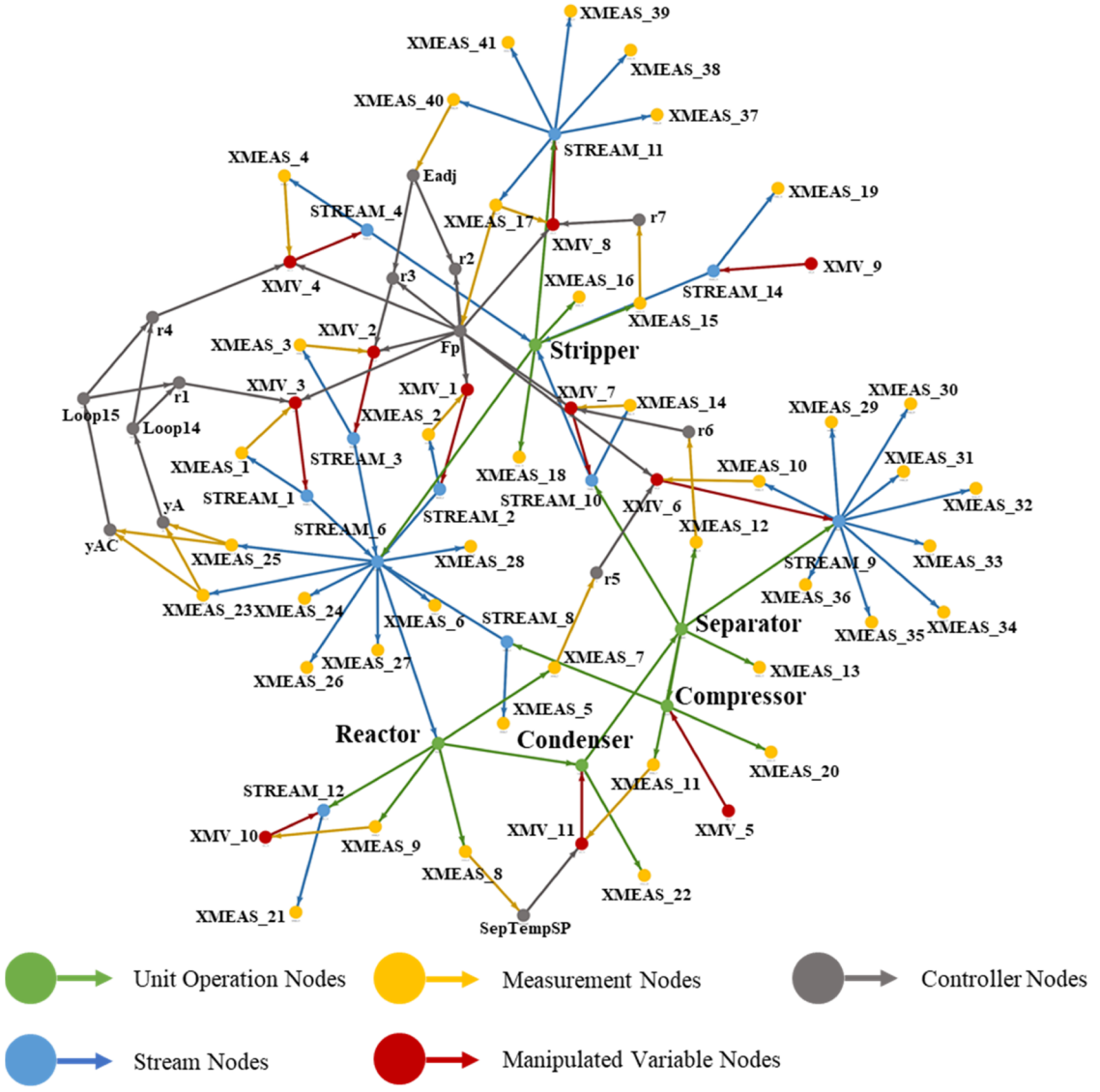



**Figure 4.** TE graph.

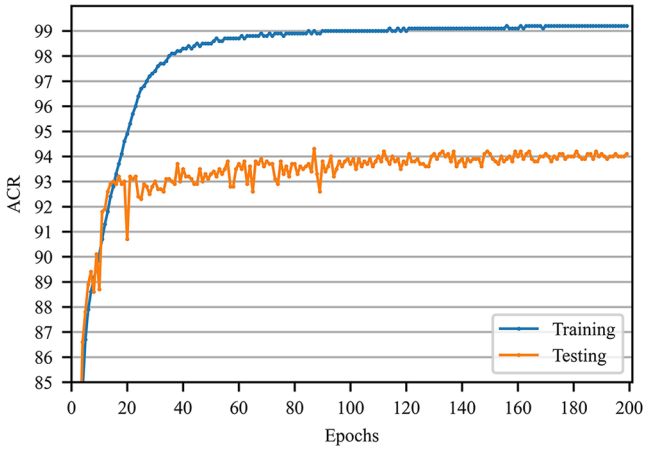



**Figure 5.** Change of testing ACR with training epochs.

detected as the normal state. But there will also be a higher possibility of the model being trusted by the operators since it can avoid alarm flooding with an extremely low false alarm rate.

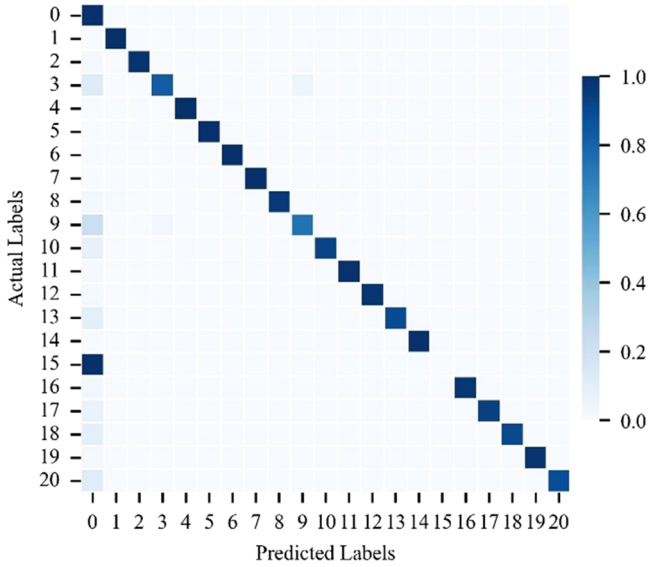



**Figure 6.** Confusion matrix for testing dataset.



https://doi.org/10.1021/acs.iecr.3c00206
Ind. Eng. Chem. Res. 2023, 62, 8350−8361

**Table 2. FDR and ACR Comparison of Fault Diagnosis Results**

| fault type | DCNN[4] | BiGRU[5] | CGN[26] | PTCN[14] | target transformer[27] | ProTopormer (random attention weights) | ProTopormer (ours) |
|---|---|---|---|---|---|---|---|
| normal | 0.978 | 0.969 | 0.985 | 0.9924 | | 0.6349 | 0.9927 |
| IDV 1 | 0.986 | 0.986 | 0.975 | 0.9931 | 0.9975 | 0 | 0.9975 |
| IDV 2 | 0.985 | 0.972 | 0.980 | 0.9819 | 0.9844 | 0 | 0.9825 |
| IDV 3 | 0.917 | 0.935 | | 0.8804 | 0.9938 | 0 | 0.8263 |
| IDV 4 | 0.976 | 0.974 | 0.824 | 0.9956 | 0.9962 | 0 | 1 |
| IDV 5 | 0.915 | 0.998 | 0.980 | 0.9786 | 0.9188 | 0.0125 | 0.9925 |
| IDV 6 | 0.975 | 1 | 1 | 1 | 0.9821 | 0 | 1 |
| IDV 7 | 0.999 | 1 | 1 | 1 | 0.9994 | 0 | 1 |
| IDV 8 | 0.922 | 0.753 | 0.966 | 0.9160 | 0.9556 | 0 | 0.9638 |
| IDV 9 | 0.584 | 0.807 | | 0.6601 | 0.6869 | 0.1088 | 0.7450 |
| IDV 10 | 0.964 | 1 | 0.881 | 0.9276 | 0.9769 | 0 | 0.9237 |
| IDV 11 | 0.984 | 0.965 | 0.778 | 0.9798 | 0.9806 | 0 | 0.9850 |
| IDV 12 | 0.956 | 0.961 | 0.981 | 0.9704 | 0.9706 | 0.8250 | 0.9737 |
| IDV 13 | 0.957 | 0.953 | 0.758 | 0.8969 | 0.9612 | 0 | 0.8950 |
| IDV 14 | 0.987 | 0.996 | 0.986 | 0.9964 | 0.9875 | 0 | 0.9937 |
| IDV 15 | 0.28 | 0.541 | | 0.0035 | 0.3406 | 0 | 0.0025 |
| IDV 16 | 0.442 | 0.788 | 0.814 | 0.9685 | 0.5269 | 0 | 0.9700 |
| IDV 17 | 0.945 | 0.97 | 0.848 | 0.9254 | 0.9475 | 0 | 0.9300 |
| IDV 18 | 0.939 | 0.923 | 0.685 | 0.9049 | 0.9425 | 0 | 0.9050 |
| IDV 19 | 0.986 | 0.926 | 0.964 | 0.9650 | 0.9869 | 0 | 0.9812 |
| IDV 20 | 0.933 | 0.981 | 0.871 | 0.8825 | 0.9425 | 0 | 0.8863 |
| ACR | 0.882 | 0.927 | | 0.9392 | 0.9039 | 0.3371 | 0.9426 |
| ACR w/o IDV 9 and 15 | 0.934 | 0.952 | 0.904 | 0.9729 | 0.9473 | 0.3528 | 0.9742 |

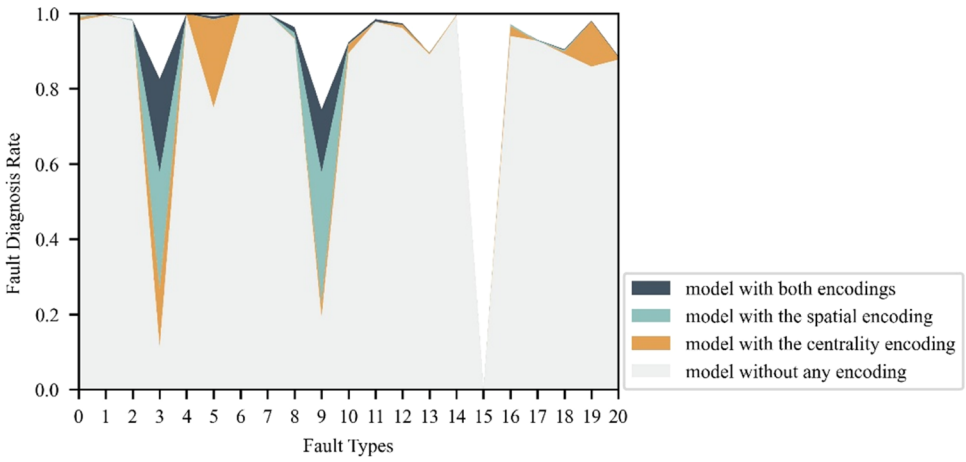



**Figure 7.** Ablation experiments.

In terms of overall classification performance, ProTopormer achieves the highest ACR of 0.9426. If faults 9 and 15, which are easily confused with normal states, are excluded, the ACR reaches an even higher 0.9742, again the highest among all models. This shows that our model has a very good overall fault diagnosis performance. This is due to the strength of the ProTopormer model itself. On the one hand, the self-attention mechanism allows the model to be more sensitive to the local details of the data. The attention mechanism also greatly improves the interpretability of the model, which will be discussed further below. On the other hand, by means of spatial and centrality encoding, we integrate process topological knowledge into the model architecture in a special form. This allows the model to reference the properties of the physical process itself during training and inference, thus enabling the fusion of knowledge. The experiment results show that the mixture of data-driven model and process knowledge provides a significant advantage in fault diagnosis performance.

To illustrate that attention weights play important roles in model performance, we completed a set of controlled experiments. We randomly sampled over a normal distribution and normalized the values with the Softmax function to obtain random attention weights as an alternative to the strictly calculated attention weights. The adjusted model was tested on the same dataset, and the corresponding results are shown in Table 2. It shows that ProTopormer with random attention weights only classifies the test samples into four categories: normal state, fault 9, and fault 12. The FDR of fault 5 is basically 0, and the FDRs of the remaining faults are all 0. The FDRs of normal state, fault 9, and fault 12 are higher, which are 0.6349, 0.1088, and 0.8250, respectively. This indicates that the model mainly classifies the test samples into these three categories. It is worth noting that the FDR of fault 12 is high enough to 0.8250, which may be because this fault sample has more prominent features and can be captured by the model without a complicated feature extraction process. The main


https://doi.org/10.1021/acs.iecr.3c00206
Ind. Eng. Chem. Res. 2023, 62, 8350−8361

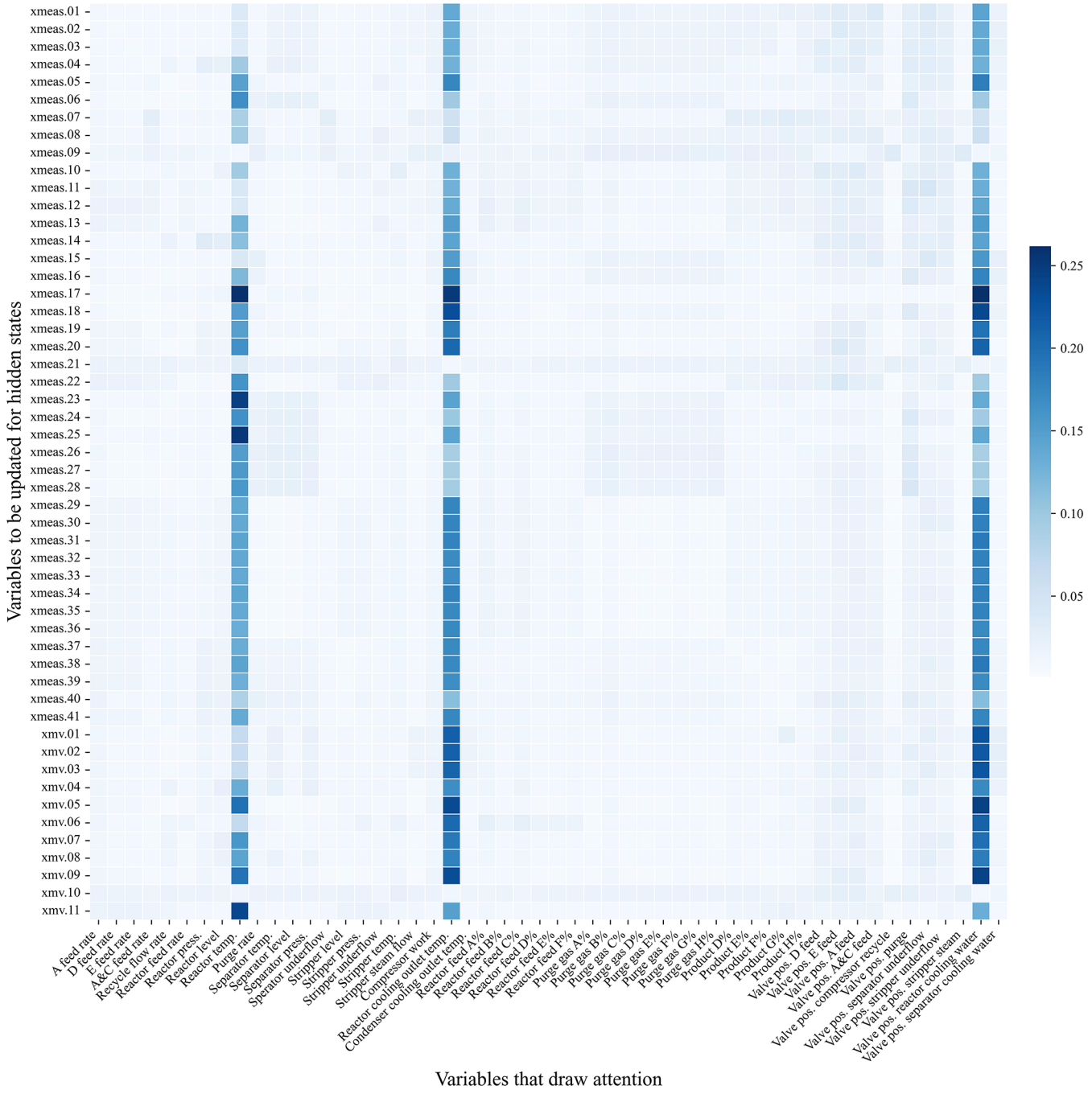



**Figure 8.** Visualization of average attention weight matrix of fault 14.

reason for the high FDR of the normal state is that test samples except for fault 12 are mainly classified into the normal state. Even so, the model is essentially unusable for fault diagnosis in the actual process. This controlled experiment illustrates that the rigorously calculated attention weights do play a very important role in the process of fault diagnosis by the model.

**4.3. Ablation Experiments.** A key mechanism in ProTopormer is the centrality and spatial encoding. These two encodings work as bridges to connect the data-driven model and process knowledge. The centrality encoding passes the information about node importance. And the spatial encoding concludes the process topology by describing connections between different nodes. This approach to integrate process knowledge into the network is very necessary. Ablation experiments were conducted to indicate this, and the results are shown in Figure 7. ACRs and FDRs for every type of fault are compared between different models. Differences between these models are whether the centrality encoding or the spatial encoding is utilized. In general, the model with both encodings achieves the highest ACR of 0.9426. Then follows an ACR of 0.9303 from the model with only the spatial encoding. The model with only the centrality encoding gets an ACR of 0.9103. The lowest ACR of 0.8905 comes from the model without utilizing any encoding.

In terms of FDRs for specific faults, the trend is almost the same as ACRs of different models. The model with both encoding always has the highest FDR for each type of fault. And the model without any encoding always has the lowest



https://doi.org/10.1021/acs.iecr.3c00206
Ind. Eng. Chem. Res. 2023, 62, 8350−8361

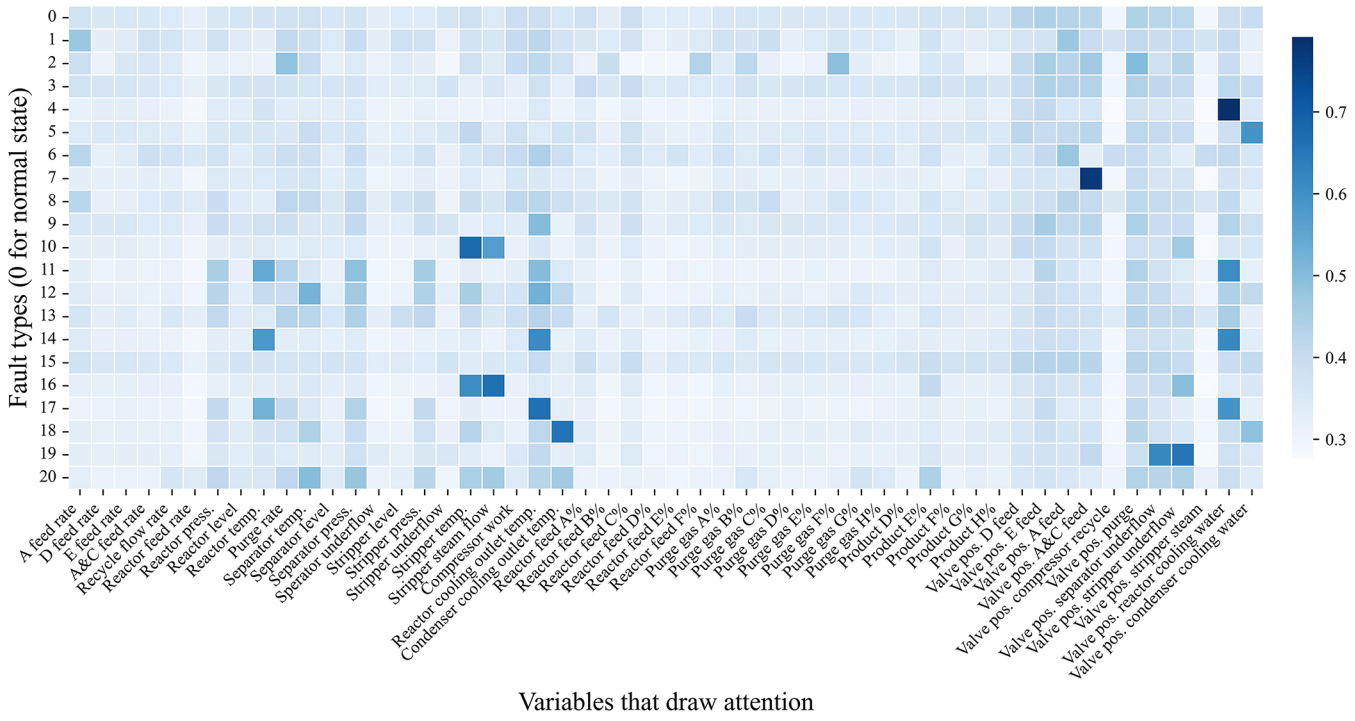



**Figure 9.** Variables with different attention weights for different faults.

FDR. This trend is especially obvious for faults 3, 5, 9, and 15, where there show troughs in the curves.

These results show that any encoding of these two used in the network can improve the general performance of the pure data-driven model. The spatial encoding can increase ACR by about 4%, and the centrality encoding can increase ACR by about 2%. Thus, spatial encoding plays a more important role in improving the general performance. Because the connectivity between nodes can provide richer information than just the importance of nodes, the combination of the two has a positive synergistic effect since the best results can be achieved by the combination of both encodings.

**4.4. Interpretability and Explainability of ProTopormer.** In many deep learning-based models for fault diagnosis, the results are always given without any explanations. Due to the complexity of the model, the meanings of network weights are unclear. This leads to a lack of interpretability of the models, which makes people distrust them.

In ProTopormer, the self-attention mechanism with special encodings guarantees the performance of the model. At the same time, it provides an entry to analyze its decision-making basis. The key lies in the attention weights in the self-attention mechanism. In every encoder layer, the output of the attention function is the product of the attention weight matrix and stacked hidden states. In mathematics, every node or position updates its hidden state using the weighted sum of hidden states in all positions.

$$\begin{aligned}&\text{attention}(Q, K, V)_i \\ &= \text{softmax}(A_i)HW_V \\ &= \left(\sum_{j=1}^{n} \text{softmax}(A_i)_j h_j\right) W_V \end{aligned} \tag{19}$$

This process is very similar to the graph convolutional operations in GCN. Essentially, they are similar ways of message passing among all of the nodes. This means that information in each node is valued differently by other nodes when updating their hidden states. Thus, with the attention weight matrix, it is easy to analyze nodes' values to others. If some nodes always have a high attention weight, it means they are naturally important to the fault diagnosis process because information of these nodes has a greater impact on the diagnosis results.

A sample should be input to ProTopormer to obtain the attention weight matrix. For every type of fault, attention weight matrices obtained with all samples are averaged to get the average attention weight matrix. It can be visualized as a heatmap to help understand which variables' features are important to fault diagnosis.

Figure 8 gives an example of the heatmap of an average attention weight matrix from fault 14. The heatmap clearly shows that many variables pay more attention to three variables: reactor temperature, reactor cooling water outlet temperature, and reactor cooling water valve position signal. This heatmap allows us to understand, to some extent, the reasons why the model classified the samples as fault 14.

In fact, the disturbance that led to fault 14 was a stuck reactor cooling water valve. This disturbance may lead to unstable reactor temperature and cooling water outlet temperature, which in turn affects the stability of the valve position signal. For fault 14, the basis of classification of the model is consistent with the physics.

Attention weights corresponding to different faults can be averaged and stacked together for comparison, as shown in Figure 9. This figure shows that ProTopormer focuses on very different variables when diagnosing different faults. It is important to note that this figure gives only the variables that the model is more concerned with when making diagnosis.



https://doi.org/10.1021/acs.iecr.3c00206
Ind. Eng. Chem. Res. 2023, 62, 8350−8361

Table 3. Actual Root Causes for Different Faults and Variables with Largest Attention Weights[a]

| fault | actual root cause | top 5 variables with the largest attention weights |
|---|---|---|
| 1 | A/C feed ratio decreases (A&C feed) | $Q_{(\text{A feed})}$, $u^{Q}_{(\text{A feed})}$, $T_{(\text{reactor cooling})}$, $Q_{(\text{purge})}$, $u^{Q}_{(\text{purge})}$ |
| 2 | B composition increases (A&C feed) | $u^{Q}_{(\text{purge})}$, $x^{F}_{(\text{purge})}$, $Q_{(\text{purge})}$, $u^{Q}_{(\text{A\&C feed})}$, $u^{Q}_{(\text{E feed})}$ |
| 3 | D feed temperature increases | $u^{Q}_{(\text{E feed})}$, $u^{Q}_{(\text{purge})}$, $u^{Q}_{(\text{A\&C feed})}$, $u^{Q}_{(\text{A feed})}$, $u^{Q}_{(\text{reactor cooling})}$ |
| 4 | reactor cooling inlet temperature increases | $u^{Q}_{(\text{reactor cooling})}$, $u^{Q}_{(\text{E feed})}$, $u^{Q}_{(\text{D feed})}$, $u^{Q}_{(\text{purge})}$, $T_{(\text{reactor})}$ |
| 5 | condenser cooling inlet temperature increases | $u^{Q}_{(\text{condenser cooling})}$, $u^{Q}_{(\text{A\&C feed})}$, $u^{Q}_{(\text{D feed})}$, $u^{Q}_{(\text{purge})}$, $T_{(\text{stripper})}$ |
| 6 | A feed loss | $u^{Q}_{(\text{A feed})}$, $T_{(\text{reactor cooling})}$, $Q_{(\text{A feed})}$, $u^{Q}_{(\text{reactor cooling})}$, $u^{Q}_{(\text{E feed})}$ |
| 7 | C feed header pressure loss-reduced availability | $u^{Q}_{(\text{A\&C feed})}$, $u^{Q}_{(\text{purge})}$, $u^{Q}_{(\text{reactor cooling})}$, $T_{(\text{separator})}$, $u^{Q}_{(\text{E feed})}$ |
| 8 | A, B, C feed composition changes randomly | $u^{Q}_{(\text{A feed})}$, $Q_{(\text{A feed})}$, $T_{(\text{reactor cooling})}$, $Q_{(\text{purge})}$, $W_{(\text{compressor})}$ |
| 9 | D feed temperature changes randomly | $T_{(\text{reactor cooling})}$, $u^{Q}_{(\text{E feed})}$, $u^{Q}_{(\text{purge})}$, $u^{Q}_{(\text{reactor cooling})}$, $u^{Q}_{(\text{A\&C feed})}$ |
| 10 | C feed temperature changes randomly | $T_{(\text{stripper})}$, $Q_{(\text{stripper steam})}$, $u^{Q}_{(\text{stripper underflow})}$, $u^{Q}_{(\text{D feed})}$, $u^{Q}_{(\text{E feed})}$ |
| 11 | reactor cooling water inlet temperature changes randomly | $u^{Q}_{(\text{reactor cooling})}$, $T_{(\text{reactor})}$, $T_{(\text{reactor cooling})}$, $P_{(\text{separator})}$, $P_{(\text{stripper})}$ |
| 12 | condenser cooling inlet temperature changes randomly | $T_{(\text{reactor cooling})}$, $T_{(\text{separator})}$, $P_{(\text{separator})}$, $T_{(\text{stripper})}$, $u^{Q}_{(\text{reactor cooling})}$ |
| 13 | reaction kinetics drift slowly | $u^{Q}_{(\text{reactor cooling})}$, $P_{(\text{separator})}$, $Q_{(\text{purge})}$, $T_{(\text{reactor cooling})}$, $u^{Q}_{(\text{purge})}$ |
| 14 | reactor cooling water valve sticking | $u^{Q}_{(\text{reactor cooling})}$, $T_{(\text{reactor cooling})}$, $T_{(\text{reactor})}$, $u^{Q}_{(\text{E feed})}$, $u^{Q}_{(\text{purge})}$ |
| 15 | condenser cooling water valve sticking | $u^{Q}_{(\text{E feed})}$, $u^{Q}_{(\text{A feed})}$, $u^{Q}_{(\text{purge})}$, $u^{Q}_{(\text{A\&C feed})}$, $u^{Q}_{(\text{D feed})}$ |

[a] $T$, $P$, $L$, $Q$, $x$, and $u$ are temperature, pressure, level, flow rate, composition, and controller signal, respectively. Subscript notes unit operations or streams. Superscript notes physical quantities or components.

However, it can be found that these variables are very closely related to the root causes of these faults.

In several faults, attention weights are focused on a few major variables, such as faults 4, 5, 7, 10, and 14. To diagnose fault 4, the heatmap shows that the reactor cooling water valve position signal is the most important variable. The corresponding disturbance is an increase of reactor cooling water inlet temperature, which may lead to changes of reactor temperature and cooling water valve position signal. For fault 5, the condenser cooling water valve position signal is the most important, which is closely related to the disturbance of an increase of condenser cooling water inlet temperature. For fault 7, A and C feed valve position signal has significantly the largest attention weight. This was caused by the pressure loss of A and C feed. For fault 10, C feed temperature changed randomly, which might influence the stripper first. That is why the stripper temperature and its steam flow temperature have the largest two attention weights in the heatmap. And fault 14 has been analyzed previously.

Top five variables with the largest attention weights in Figure 9 are listed in Table 3. The first column shows the diagnosis results of the model. The second column is the actual root cause of the fault. For each fault, we bold the name of variables directly related to the root cause of the fault. Since the TE process gives root causes of only 15 faults, the table contains only these 15 faults too.

Faults 4, 5, 7, 10, and 14 have been analyzed previously. For fault 1, the disturbance might directly cause the A composition in reactor feed, thus influencing A feed's rate. For fault 2, B composition increases in A and C feed will influence the valve signal. For fault 3, D feed temperature will easily influence the reactor's temperature and cooling water flow. For fault 6, A feed loss will directly increase the A feed valve signal. For fault 8, the composition change will be quickly reflected in the A, B, and C feed rates. For faults 9, 11, 12, and 13, the disturbances can easily relate to the reactor's temperature and cooling water flow. As for fault 15, there are no obvious relationships between variables and the root cause. When the reactor temperature keeps stable, the condenser cooling water flow does not need to be adjusted either. So, the valve sticking may not have any impact. This may be the reason why fault 15 is difficult to distinguish from the normal state.

Based on ProTopormer's attention weights, the key variables for fault diagnosis were found for each type of faults. This allowed us to get a glimpse of the model's decision-making basis. After analyzing the faults one by one, we found that the key variables given by the model are the root causes themselves or strongly correlated to the root causes (some root causes do not have corresponding process variables).

In Table 3, the root causes of the 14 types of faults are included in the top 5 variables with the largest attention weights. This has a high practical application value. In the actual fault diagnosis process, real root causes for the abnormal situation are unknown. The model can give not only the fault type but also key variables to the operators as reference. This can improve the efficiency of failure checking and trouble-shooting.

## 5. CONCLUSIONS

A novel model ProTopormer and the corresponding method were proposed for fault detection and diagnosis. Self-attention mechanism and process topology knowledge were fully combined to achieve high model performance and good interpretability. For the case study on the TE process, ProTopormer achieved a very high accurate classification rate of 94.26% and a very low false alarm rate of 0.73%. Disregarding faults 9 and 15, the accuracy can reach a much higher 97.42%.

The process topology knowledge was integrated into the data-driven model by the centrality encoding and the spatial encoding. Ablation experiments showed the significance of both encodings for model performance improvement.



https://doi.org/10.1021/acs.iecr.3c00206
Ind. Eng. Chem. Res. 2023, 62, 8350−8361

Attention weights were visualized and quantitatively studied to explore the key variables that ProTopormer focused on when performing fault diagnosis. For each type of fault, the model gave several key variables, which had strong physical correlations with the root cause. This fully demonstrated the interpretability and explainability of ProTopormer.

Both the performance and interpretability of the models are important foundations for their practical application. Both should not be neglected in the research. And the integration of knowledge and data-driven models is beneficial to both aspects. Exploring ways to combine knowledge and models will be an important and promising direction in future research.

## AUTHOR INFORMATION


### Corresponding Author

**Jinsong Zhao** − *State Key Laboratory of Chemical Engineering, Department of Chemical Engineering, Tsinghua University, Beijing 100084, China; Beijing Key Laboratory of Industrial Big Data System and Application, Tsinghua University, Beijing 100084, China*; Email: jinsongzhao@tsinghua.edu.cn

### Authors

**Deyang Wu** − *State Key Laboratory of Chemical Engineering, Department of Chemical Engineering, Tsinghua University, Beijing 100084, China*; orcid.org/0000-0001-9569-7768

**Xiaotian Bi** − *State Key Laboratory of Chemical Engineering, Department of Chemical Engineering, Tsinghua University, Beijing 100084, China*

Complete contact information is available at: https://pubs.acs.org/10.1021/acs.iecr.3c00206


### Notes

The authors declare no competing financial interest.

## ACKNOWLEDGMENTS

The authors gratefully acknowledge support from the National Natural Science Foundation of China (nos. 21878171 and 62003004) and the National Science and Technology Innovation 2030 Major Project (2018AAA0101605) of the Ministry of Science and Technology of China.

## REFERENCES


(1) Venkatasubramanian, V.; Rengaswamy, R.; Yin, K.; Kavuri, S. N. A Review of Process Fault Detection and Diagnosis Part I: Quantitative Model-Based Methods. *Comput. Chem. Eng.* **2003**, 293−311.

(2) Xie, D.; Bai, L. In *A Hierarchical Deep Neural Network for Fault Diagnosis on Tennessee-Eastman Process*, 2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA), 2015; pp 745−748.

(3) Zhang, Z.; Zhao, J. A Deep Belief Network Based Fault Diagnosis Model for Complex Chemical Processes. *Comput. Chem. Eng.* **2017**, *107*, 395−407.

(4) Wu, H.; Zhao, J. Deep Convolutional Neural Network Model Based Chemical Process Fault Diagnosis. *Comput. Chem. Eng.* **2018**, *115*, 185−197.

(5) Zhang, S.; Bi, K.; Qiu, T. Bidirectional Recurrent Neural Network-Based Chemical Process Fault Diagnosis. *Ind. Eng. Chem. Res.* **2020**, *59*, 824−834.

(6) Zheng, S.; Zhao, J. A Self-Adaptive Temporal-Spatial Self-Training Algorithm for Semisupervised Fault Diagnosis of Industrial Processes. *IEEE Trans. Ind. Inf.* **2022**, *18*, 6700−6711.

(7) Cheng, F.; He, Q. P.; Zhao, J. A Novel Process Monitoring Approach Based on Variational Recurrent Autoencoder. *Comput. Chem. Eng.* **2019**, *129*, No. 106515.

(8) Bi, X.; Zhao, J. A Novel Orthogonal Self-Attentive Variational Autoencoder Method for Interpretable Chemical Process Fault Detection and Identification. *Process Saf. Environ. Prot.* **2021**, *156*, 581.

(9) Downs, J. J.; Vogel, E. F. A Plant-Wide Industrial Process Control Problem. *Comput. Chem. Eng.* **1993**, *17*, 245−255.

(10) Venkatasubramanian, V. The Promise of Artificial Intelligence in Chemical Engineering: Is It Here, Finally? *AIChE J.* **2019**, *65*, 466−478.

(11) Qin, S. J.; Chiang, L. H. Advances and Opportunities in Machine Learning for Process Data Analytics. *Comput. Chem. Eng.* **2019**, *126*, 465−473.

(12) Bai, Y.; Zhao, J. A Novel Transformer-Based Multi-Variable Multi-Step Prediction Method for Chemical Process Fault Prognosis. *Process Saf. Environ. Prot.* **2023**, *169*, 937−947.

(13) Sivaram, A.; Venkatasubramanian, V. XAI-MEG: Combining Symbolic AI and Machine Learning to Generate First-Principles Models and Causal Explanations. *AIChE J.* **2022**, *68*, No. e17687.

(14) Wu, D.; Zhao, J. Process Topology Convolutional Network Model for Chemical Process Fault Diagnosis. *Process Saf. Environ. Prot.* **2021**, *150*, 93−109.

(15) Wu, D.; Zhao, J. Understand How CNN Diagnoses Faults with Grad-CAM. In *Computer Aided Chemical Engineering*; Yamashita, Y.; Kano, M., Eds.; Elsevier, 2022; Vol. *49*, pp 1537−1542.

(16) Ying, C.; Cai, T.; Luo, S.; Zheng, S.; Ke, G.; He, D.; Shen, Y.; Liu, T.-Y. Do Transformers Really Perform Badly for Graph Representation?. 2021, arXiv:2106.05234. arXiv.org e-Print archive. https://arxiv.org/abs/2106.05234.

(17) Bahdanau, D.; Cho, K.; Bengio, Y. Neural Machine Translation by Jointly Learning to Align and Translate. 2016, arXiv:1409.0473. arXiv.org e-Print archive. https://arxiv.org/abs/1409.0473.

(18) Yang, Z.; Yang, D.; Dyer, C.; He, X.; Smola, A.; Hovy, E. In *Hierarchical Attention Networks for Document Classification*, Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies; Association for Computational Linguistics: San Diego, California, 2016; pp 1480−1489.

(19) Lin, Z.; Feng, M.; dos Santos, C. N.; Yu, M.; Xiang, B.; Zhou, B.; Bengio, Y. A Structured Self-Attentive Sentence Embedding. 2017, arXiv:1703.03130. arXiv.org e-Print archive. https://arxiv.org/abs/1703.03130.

(20) Daniluk, M.; Rocktäschel, T.; Welbl, J.; Riedel, S. Frustratingly Short Attention Spans in Neural Language Modeling. 2017, arXiv:1702.04521. arXiv.org e-Print archive. https://arxiv.org/abs/1702.04521.

(21) Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; Polosukhin, I. Attention Is All You Need. In *Advances in Neural Information Processing Systems*; Curran Associates, Inc., 2017; Vol. *30*.

(22) He, K.; Zhang, X.; Ren, S.; Sun, J. In *Deep Residual Learning for Image Recognition, 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, 2016; pp 770−778.

(23) Ba, J. L.; Kiros, J. R.; Hinton, G. E. Layer Normalization. 2016, arXiv:1607.06450. arXiv.org e-Print archive. https://arxiv.org/abs/1607.06450.

(24) Ricker, N. L. Optimal Steady-State Operation of the Tennessee Eastman Challenge Process. *Comput. Chem. Eng.* **1995**, *19*, 949−959.

(25) Bathelt, A.; Ricker, N. L.; Jelali, M. Revision of the Tennessee Eastman Process Model. *IFAC-PapersOnLine* **2015**, *48*, 309−314.

(26) Lou, C.; Li, X.; Atoui, M. A. Bayesian Network Based on an Adaptive Threshold Scheme for Fault Detection and Classification. *Ind. Eng. Chem. Res.* **2020**, *59*, 15155−15164.

(27) Wei, Z.; Ji, X.; Zhou, L.; Dang, Y.; Dai, Y. A Novel Deep Learning Model Based on Target Transformer for Fault Diagnosis of Chemical Process. *Process Saf. Environ. Prot.* **2022**, *167*, 480−492.