Spatiotemporal Graph Neural Networks for Fault Detection and Structural Learning in Chemical Processes: Use Case on the Tennessee Eastman Process

Rayane Ammar Khodja[1,2], Alexandre Voisin[2], Victor Costa[1], Fanny Casteran[1], Benoit Celse[1], and Benoît Iung[2]

[1] *IFP Energies Nouvelles, Rond-point de l'échangeur de Solaize, BP 3, 69360 Solaize, France*
rayane.ammar-khodja@ifpen.fr
victor.costa@ifpen.fr
fanny.casteran@ifpen.fr
benoit.celse@ifpen.fr

[2] *Université de Lorraine, CNRS, CRAN, F-54000 Nancy, France*
alexandre.voisin@univ-lorraine.fr
benoit.iung@univ-lorraine.fr

ABSTRACT

Classical Fault Detection and Diagnosis (FDD) methods, including many data-driven approaches, assume a static normal operating space and interpret deviations from a fixed reference as fault indicators. When Operating Conditions (OCs) vary over time, this assumption breaks down: legitimate transitions trigger false alarms while moderate faults go undetected. We propose a spatiotemporal Graph Neural Network (GNN) framework that decomposes the normal operating space into OC-specific subspaces linked by transition functions, with a dual learning objective combining reconstruction loss and a Deep Support Vector Data Description (DeepSVDD) one-class term. The framework learns adjacency matrices through Graph Attention Networks (GATs) and integrates spatial modelling with temporal encoding to represent process dynamics under evolving OCs.

This paper evaluates the foundational components of the framework, fault detection, spatial graph learning via GATv2 and temporal encoding, on the Tennessee Eastman Process (TEP) benchmark, with training performed exclusively on fault-free data. The spatiotemporal architecture achieves competitive detection performance from reconstruction error alone, with similarity-based feature selection improving both accuracy and graph structure diversity. We then evaluate the physical interpretability of the learned attention matrices against 22 ground-truth sensor pairs derived from the TEP control structure and process topology. The GATv2 attention does not recover all the necessary known physical pairs across multiple hyperparameter configurations, suggesting a structural limitation of reconstruction-driven attention rather than a tuning issue. This result challenges a common assumption in GNN-based FDD: that learned attention weights provide a basis for fault diagnosis and root-cause analysis. The architecture detects faults effectively, but the learned graph does not encode the physical topology needed for interpretable diagnosis, motivating physics-informed graph construction.

1. INTRODUCTION

Fault Detection and Diagnosis (FDD) in chemical processes is critical for ensuring safety, product quality, and economic efficiency (Venkatasubramanian, Rengaswamy, Yin, & Kavuri, 2003). Modern industrial systems generate high-dimensional multivariate time-series data from tens to hundreds of sensors. Data-driven methods, including Principal Component Analysis (PCA) (Chiang, Russell, & Braatz, 2001), autoencoders (Sakurada & Yairi, 2014), and Long Short-Term Memory (LSTM) networks (Filonov, Lavrentyev, & Vorontsov, 2016), have been widely adopted for anomaly detection on such data.

Nevertheless, most existing approaches treat the sensor array as a flat feature vector, ignoring the physical topology of the process (including PCA-based monitoring (Chiang et al., 2001), deep autoencoders (Sakurada & Yairi, 2014), LSTM-based predictive models (Filonov et al., 2016), and the broader data-driven FDD literature reviewed by Alauddin, Khan, Imtiaz, and Ahmed (2018)). In reality, sensors are embedded in a network of material flows, energy balances, and control loops that govern how disturbances propagate through the

plant. Graph Neural Networks (GNNs) offer a natural framework for encoding these structural relationships, with nodes representing sensors and edges capturing inter-sensor dependencies (Wu et al., 2021). By propagating information along graph edges that reflect physical process connections, GNNs can exploit the known topology of the plant to improve both detection sensitivity and diagnostic interpretability.

A critical gap in this literature is the absence of systematic evaluation of whether graph structures learned by attention mechanisms reflect physically meaningful sensor relationships, or merely optimize for the downstream reconstruction objective. Most studies feed raw sensor data to GAT-based models and assume, without verification, that the resulting attention weights capture real process physics. This question is especially relevant for process industries, where interpretability of the learned model is essential for operator trust and root-cause analysis. We review existing GNN-based FDD approaches and their treatment of this assumption in Section 2.

In this paper, we address this gap directly. We build on the MTAD-GAT architecture of Zhao et al. (2020), which is a dual feature–temporal graph attention model originally proposed for infrastructure anomaly detection (details in Section 2), and adapt it to the chemical-process setting, where, to the best of our knowledge, it has not previously been evaluated. Around this backbone, we present a spatiotemporal GNN framework for FDD under dynamic operating conditions and evaluate its foundational components (spatial graph learning and temporal encoding) and fault detection on the TEP benchmark (Downs & Vogel, 1993; Rieth, Amsel, Tran, & Cook, 2017) under fixed operating conditions. The OC-aware extensions, including dual-loss training and multi-mode evaluation, are the subject of ongoing work and outlined in Section 6. Our contributions are:

1. The first application of the MTAD-GAT architecture (Zhao et al., 2020) to a chemical process benchmark (TEP). This transfer required five methodological modifications to the original training and evaluation pipeline, enumerated in Section 4;
2. A systematic evaluation of the physical interpretability of GATv2 (Brody, Alon, & Yahav, 2022) attention weights using a Hit-Rate@k metric against process-knowledge-derived ground truth;
3. An empirical analysis of the gap between reconstruction-driven attention and physical process topology, motivating physics-informed graph construction.

This paper is organised as follows. Section 2 surveys GNN-based FDD methods and their treatment of graph construction and physical interpretability. Section 3 presents the proposed framework, including the OC-aware formulation, the dual-loss objective, and the spatiotemporal architecture used to learn the sensor graph. Section 4 details the experimental protocol, covering the Tennessee Eastman Process benchmark, feature selection, the data pipeline, anomaly scoring, ground-truth physical pairs, and evaluation metrics. Section 5 reports detection performance, graph structure evaluation and the limitations of our work. Section 6 draws final conclusions and outlines directions for future work.

2. RELATED WORK

The earliest GNN-based methods for chemical process FDD relied on graphs derived from process knowledge. Wu et al. (2021) proposed the Process Topology Convolutional Network (PTCN), which feeds a Graph Convolutional Network (Kipf & Welling, 2017) with an adjacency matrix constructed from the plant's piping-and-instrumentation diagram (P&ID). Jia, Yang, Wang, Xu, and Liu (2023) formalized this idea as Topology-Guided Graph Learning (TGGL): a Signed Directed Graph (SDG) prior is injected into the GCN, and the model is biased toward learned structures consistent with known causal relationships. These methods report strong fault classification performance on the Tennessee Eastman Process (TEP), but require either a verified process topology or expert-curated causal graphs.

An additional set of works treat the adjacency matrix itself as a trainable parameter. Deng and Hooi (2021) introduced the Graph Deviation Network (GDN) for multivariate time-series anomaly detection, learning sensor relations from a low-rank embedding rather than from prior knowledge. Kovalenko, Pozdnyakov, and Makarov (2024) extended this idea by allowing multiple parallel adjacency matrices to be optimized end-to-end on TEP, reporting that learned graphs can match or exceed knowledge-derived ones for fault classification. Chen, Liu, Hu, and Ding (2023) proposed Interaction-Aware Graph Neural Networks (IAGNN), in which a heterogeneous graph with multiple edge types is constructed, and edge weights are learned by attention. These trainable-graph methods relax the dependence on expert knowledge, but leave open the question raised by Jia et al. (2023): whether the resulting graphs encode physically meaningful structure or merely statistical co-variation.

Graph Attention Networks (Veličković et al., 2018; Brody et al., 2022) have attracted particular interest because the attention coefficients α_{ij} admit an interpretation as edge weights, making the learned graph nominally inspectable. Zhao et al. (2020) proposed MTAD-GAT, a dual feature–temporal graph attention architecture for multivariate anomaly detection, originally validated on infrastructure benchmarks (SMD, MSL, SMAP) rather than chemical processes. Liu and Jafarpour (2024) combined a CNN feature extractor with a GAT layer guided by Granger-causality maps for TEP fault detection and root-cause analysis. Across these works, the attention matrix is treated as a structural artifact that supports diagnostic interpretation, yet none provides a systematic evaluation of whether the attention weights recover the underlying process

topology.

GNN-based detectors fall into two methodological families. Prediction-based approaches, such as GDN (Deng & Hooi, 2021) and CNN-GAT (Liu & Jafarpour, 2024), train the model to forecast the next timestep and flag deviations between predictions and observations. Reconstruction-based approaches, including MTAD-GAT (Zhao et al., 2020), train an autoencoder to reproduce the input window and flag deviations between input and reconstruction. Both paradigms can in principle drive the learning of a sensor graph through their respective gradients, but they impose different inductive biases: reconstruction emphasises which sensors are predictive of which others within the current window, while prediction emphasises which sensors lead which others in time. Our framework follows the reconstruction paradigm because it aligns with the OC-aware decomposition (Section 3), in which each operating condition defines a static normal manifold rather than a dynamical regime; the structural question we ask is whether reconstruction-driven attention recovers physical topology on TEP.

Most of the works above do not evaluate the physical interpretability of the learned attention or adjacency matrix against an explicit process-knowledge ground truth. Several explicitly assume such interpretability without verifying it (Zhao et al., 2020; Liu & Jafarpour, 2024; Chen et al., 2023); others demonstrate that physics-informed graphs outperform purely data-driven ones for classification (Jia et al., 2023; Wu et al., 2021) but do not isolate the question of whether attention alone can recover the prior. The present work targets this gap.

3. PROPOSED FRAMEWORK

Most reconstruction-based fault detection methods aggregate all training data into a single model of normal behaviour (Alauddin et al., 2018). The resulting normal space must accommodate every operating condition the plant has seen, inflating the decision boundary to encompass the full operational envelope. Abrupt faults of moderate magnitude fall inside this inflated boundary and go undetected; progressive faults such as catalyst deactivation or sensor drift are detected only after the deviation has accumulated well beyond the point where early intervention was possible. The method functions as an extreme-event detector rather than a sensitive fault monitor.

Figure 1 illustrates the problem. Under a single-threshold approach (left), intentional OC transitions widen the normal space $\mathcal{N}$ until only faults of extreme magnitude exceed the boundary. An OC-aware decomposition (right) partitions $\mathcal{N}$ into condition-specific subspaces $\{\mathcal{N}_{OC_1}, \ldots, \mathcal{N}_{OC_K}\}$, each with its own tighter boundary, so that moderate faults become distinguishable from normal variability within any given condition.

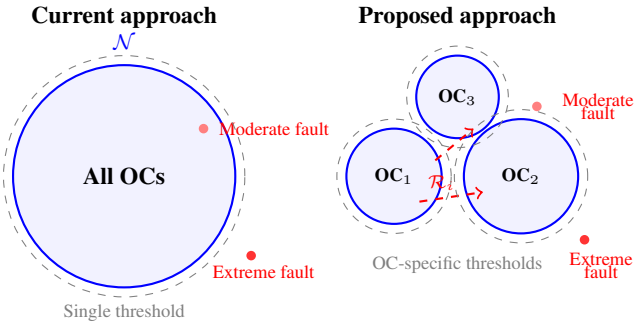

Figure 1. Single-threshold detection (left) versus OC-aware detection (right).

Formally, we define normal behaviour not as a single space but as a collection of OC-specific subspaces linked by transformation functions:

$$\mathcal{N} = \{\mathcal{N}_{\text{prep}},\ \mathcal{N}_{OC_1},\ \ldots,\ \mathcal{N}_{OC_K}\} \tag{1}$$

$$\mathcal{R}_i : \mathcal{N}_{OC_i} \to \mathcal{N}_{OC_{i+1}} \tag{2}$$

This decomposition enables four distinguishable system states: (1) *preparation*: readings inside $\mathcal{N}_{\text{prep}}$ as the plant is brought to its initial operating point; (2) *normal within an OC*: readings inside $\mathcal{N}_{OC_i}$; (3) *normal transition*: evolution following $\mathcal{R}_i$ between adjacent subspaces; and (4) *actual fault*: deviations outside all OC boundaries, preparation envelope, and transition corridors.

The framework combines two complementary objectives (Figure 2). A spatiotemporal autoencoder reconstructs sensor values from their history and neighbourhood context; high reconstruction error signals deviation from learned normal patterns. A Deep SVDD-inspired one-class loss term (Ruff et al., 2018) simultaneously maps normal data into a compact hypersphere in latent space, tightening the decision boundary beyond what reconstruction error alone provides.

The model consumes the multivariate sensor stream as a sequence of overlapping sliding windows. Let $\mathbf{X} \in \mathbb{R}^{L\times N}$ denote a recording of length L over N sensors, where $\mathbf{x}_t \in \mathbb{R}^N$ is the sensor vector at timestep t. A window of length T starting at index w is

$$X_w = \left[\mathbf{x}_w,\ \mathbf{x}_{w+1},\ \ldots,\ \mathbf{x}_{w+T-1}\right]^\top \in \mathbb{R}^{T\times N}, \tag{3}$$

with successive windows separated by a stride s. During training, a batch of B such windows is processed jointly, producing the input tensor of shape (B, T, N). The model reconstructs the last timestep of each window, $\hat{\mathbf{x}}_{w,T} \in \mathbb{R}^N$, from the spatial and temporal context provided by the preceding $T-1$ steps. All loss terms below operate on this windowed representation.

For a single window $X_w \in \mathbb{R}^{T\times N}$ (Eq. 3), the reconstruction loss measures the squared deviation between the actual and

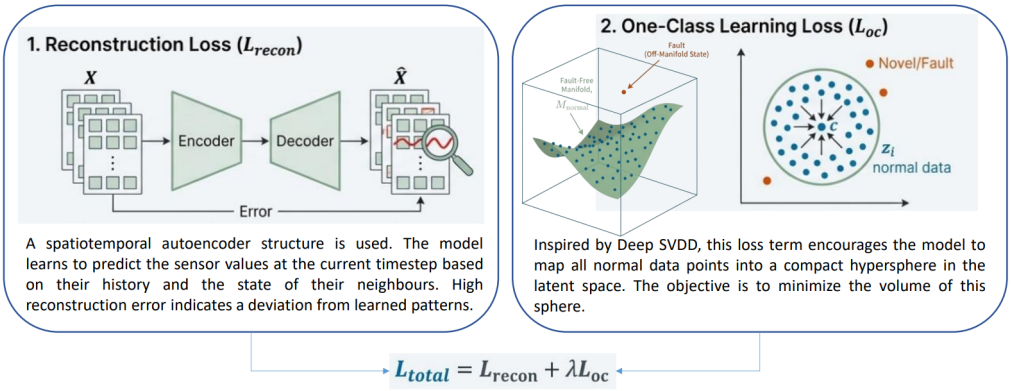

Figure 2. Dual-loss framework. Left: the spatiotemporal Graph Autoencoder (GAE) for reconstruction. Right: a Deep SVDD-inspired one-class loss maps normal data to a compact hypersphere in latent space.

reconstructed sensor values at the last timestep $t = T$:

$$L_{\text{recon}} = \frac{1}{N} \sum_{i=1}^{N} \left(x_{T,i} - \hat{x}_{T,i} \right)^2 \tag{4}$$

where $x_{T,i}$ and $\hat{x}_{T,i}$ denote the observed and reconstructed values of sensor i at the last timestep of the window X_w. The total training loss averages L_{recon} over all windows in a batch.

The one-class loss is calculated as follows:

$$L_{\text{oc}} = \frac{1}{M} \sum_{j=1}^{M} \left\| \phi(\mathbf{x}^{(j)}; \mathcal{W}) - \mathbf{c} \right\|^2 \tag{5}$$

where $\mathbf{x}^{(j)} \in \mathbb{R}^{T \times N}$ is the j-th fault-free training window (Eq. 3), $\phi(\cdot; \mathcal{W})$ is the encoder network with parameters $\mathcal{W}$, M is the number of fault-free training windows, and $\mathbf{c}$ is the hypersphere centre computed as the mean of initial encoder outputs on fault-free data (Ruff et al., 2018).

The total training objective combines both terms:

$$L_{\text{total}} = L_{\text{recon}} + \lambda \, L_{\text{oc}} \tag{6}$$

where λ controls the relative weight of the one-class term. The current paper evaluates the reconstruction component (L_{recon}); integrating the one-class term is part of ongoing work.

The remainder of this paper focuses exclusively on the reconstruction branch of the framework, i.e spatiotemporal graph autoencoder on the left of Figure 2, trained under L_{recon} alone, and evaluates it under fixed operating conditions using the modified MTAD-GAT backbone described above. The central empirical question we examine is whether the GATv2 attention matrix learned through reconstruction alone recovers physically meaningful sensor relationships, or whether reconstruction-driven optimisation alone is insufficient to surface the process topology. The OC-aware decomposition (Eqs. 1 and 2), the one-class loss term L_{oc}, and dynamic-OC evaluation are deferred to future work and revisited in Section 6.

3.1. Spatial Graph Learning

At each timestep, the N sensor channels of a window $X_w \in \mathbb{R}^{T \times N}$ (Eq. 3) are treated as the nodes of a fully connected directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with $|\mathcal{V}| = N$. Each node i is associated with a feature vector $\mathbf{v}_i \in \mathbb{R}^T$ that contains its T recent observations within the window. The graph is fully connected to avoid imposing prior structure on the learning problem: the attention mechanism is asked to discover which pairs of sensors are mutually informative from data alone.

We use GATv2 rather than the original GAT (Veličković et al., 2018) because the latter computes only *static* attention: the ranking of source nodes attended to by a given target is fixed by the learned parameters and does not depend on the target's own features. Brody et al. (2022) show that this restriction prevents the original GAT from representing certain simple node-ranking problems.

For each ordered pair (i, j) of sensors, GATv2 (Brody et al., 2022) computes an attention coefficient α_{ij} that quantifies how much sensor j's features contribute to the updated representation of sensor i. The coefficient is obtained in two steps. First, a shared learnable linear projection W and a learnable attention vector $\mathbf{a}$ produce a scalar score for each pair:

$$e_{ij} = \mathbf{a}^{\top} \, \text{LeakyReLU}\big(W \, [\mathbf{v}_i \, \| \, \mathbf{v}_j] \big), \tag{7}$$

where $\|$ denotes vector concatenation. Second, the scores are normalised across all neighbours of i via a softmax:

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{N} \exp(e_{ik})}. \tag{8}$$

The updated node representation is a weighted sum of neighbour features: $\mathbf{v}'_i = \sigma\big(\sum_j \alpha_{ij} \, W \, \mathbf{v}_j \big)$, where σ is a non-linearity. Stacking the α_{ij} yields a dense adjacency matrix

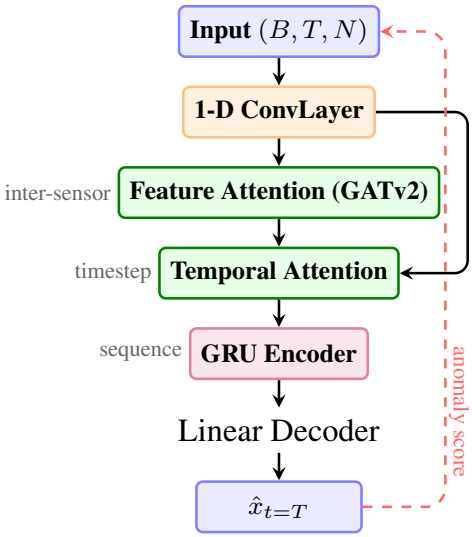

Figure 3. Architecture of the spatiotemporal graph attention autoencoder. The 1-D ConvLayer provides local temporal smoothing. Feature attention (GATv2) learns inter-sensor relationships; temporal attention models timestep dependencies. Both outputs are concatenated and encoded by a GRU. A linear decoder reconstructs $\hat{x}_{t=T}$; reconstruction error serves as the anomaly score.

$A \in \mathbb{R}^{N \times N}$ whose rows sum to one.

3.2. Temporal Encoding

A Gated Recurrent Unit (GRU) layer encodes the temporal dynamics of the concatenated spatial and temporal attention outputs. We chose the GRU over the LSTM for two reasons. First, the GRU's reduced parameter count (no separate cell-state and output gates) is generally associated with faster convergence and reduced overfitting risk when training data is limited, motivating its use in this setting. Second, the spatial and temporal attention layers upstream already provide multiplicative gating over the input sequence, which reduces the marginal value of the additional gating mechanisms an LSTM provides. A linear decoder reconstructs the last timestep of each input window from the GRU's final hidden state. The anomaly score is the per-feature mean squared error between reconstructed and actual values. Figure 3 summarises the complete architecture.

4. EXPERIMENT

4.1. Case Study: Tennessee Eastman Process

The Tennessee Eastman Process (TEP), introduced by Downs and Vogel (1993), is a realistic simulation of a continuous chemical plant comprising five major unit operations: a two-phase reactor, a condenser, a vapor–liquid separator, a recycle compressor, and a reboiled stripper. The process involves eight chemical components, four reactants (A, C, D, E), two products (G, H), one byproduct (F), and one inert (B), undergoing exothermic, irreversible reactions.

The system tracks 52 variables: 22 continuous process measurements ($xmeas_1$–$xmeas_{22}$) covering pressures, temperatures, levels, and flow rates; 19 composition analyser variables ($xmeas_{23}$–$xmeas_{41}$) sampled at 6–15 minute intervals with corresponding dead times; and 11 manipulated variables (xmv_1–xmv_{11}) representing valve positions and flow setpoints.

We use the extended dataset of Rieth et al. (2017), which provides 500 simulation runs of 500 samples each (25 hours at 3-minute intervals) for fault-free training, and 500 runs of 960 samples each (48 hours) for each of the 20 fault types. The fault scenarios span four disturbance mechanisms: step changes in feed conditions and temperatures (Faults 1–7), random variations in feed and cooling water parameters (Faults 8–12), a slow drift in reaction kinetics (Fault 13), valve sticking (Faults 14–15), and five undisclosed disturbances (Faults 16–20). Faults are introduced at sample 161 (8 hours into test simulations) after an initial fault-free period. Complete variable definitions and fault descriptions are provided in the Appendix A.

4.2. Feature Selection

Following Jia et al. (2023), we apply similarity-based feature selection to remove variables with low discriminative information across fault types. For each variable v and fault type z, a similarity coefficient is computed:

$$r_{z,v} = 1 - \frac{\|\mathbf{x}_{z,v} - \mathbf{x}_{0,v}\|_2}{\sum_{z=1}^{Z} \|\mathbf{x}_{z,v} - \mathbf{x}_{0,v}\|_2 + \varepsilon} \tag{9}$$

where $\mathbf{x}_{0,v}$ is the normal-condition signal and ε prevents division by zero. Variables whose coefficient exceeds a threshold across all fault types are removed, reducing the feature set from 52 to 35 variables. The variables removed include composition analyzers with slow sampling rates ($xmeas_{28}$–$xmeas_{41}$), compressor work ($xmeas_{20}$) and two manipulated variables (xmv_5, xmv_9).

4.3. Data Pipeline

The preprocessing pipeline enforces strict separation between training, validation, and test data to prevent information leakage and ensure that evaluation reflects genuine out-of-sample performance.

Normalisation. A `StandardScaler` is fitted exclusively on fault-free training data. The same fitted scaler is applied, without re-fitting, to validation and test splits, so that no distributional information from faulty or held-out data influences the scaling parameters.

Data partitioning. The train/validation split operates at the simulation-run level: each of the 500 fault-free runs (500 samples per run) is assigned entirely to either training (400

runs) or validation (100 runs). This prevents adjacent sliding windows drawn from the same run from appearing in both sets, which would create temporal leakage. Faulty data is reserved exclusively for testing.

Attention weight extraction. The GATv2 feature attention layer stores attention coefficients directly after the softmax operation, ensuring that the extracted weights correspond to the actual pairwise attention distribution used during message passing.

Convolutional preprocessing. A 1-D convolutional layer provides local temporal smoothing before the input reaches both the feature and temporal attention layers. Its output is fed directly into the attention modules.

4.4. Within-Window Detrending

To reduce the sensitivity of GATv2 attention to absolute signal levels, we apply within-window mean subtraction:

$$\tilde{X}_w = X_w - \frac{1}{T}\sum_{t=1}^{T} X_{w,t} \tag{10}$$

where $X_w \in \mathbb{R}^{T\times N}$ is a single window. This forces the attention mechanism to operate on relative signal variations rather than absolute operating-point values.

4.5. Anomaly Scoring

The model is trained as a reconstruction autoencoder on fault-free data. At test time, the anomaly score for each window is the per-feature mean squared error between reconstructed and actual last timestep:

$$s_w = \frac{1}{N}\sum_{i=1}^{N}(x_{w,T,i} - \hat{x}_{w,T,i})^2 \tag{11}$$

Windows from fault-free test data yield low scores; faulty windows produce elevated scores.

4.6. Ground-Truth Physical Pairs

To evaluate whether the learned attention matrix encodes process structure, we compile 22 ground-truth sensor pairs from three sources of TEP process knowledge:

1. **Control loops** (11 pairs): Direct sensor–actuator feedback connections from the decentralised control structure (Downs & Vogel, 1993);
2. **Material flows** (8 pairs): Sensors connected by physical streams (feed, reactor–separator, separator–stripper);
3. **Thermodynamic couplings** (3 pairs): Energy-balance and ideal-gas-law relationships within process units.

These pairs are validated against the SDG analysis of Maurya, Rengaswamy, and Venkatasubramanian (2004) and the bond graph model of Tidriri, Chatti, Verron, and Tiplica (2018). Evaluation uses the Hit-Rate@k metric: the fraction of the top-k learned or correlation-based pairs that match the ground truth. Pearson correlation (absolute value) serves as a competing baseline for graph construction (full list in Appendix B).

4.7. Evaluation Metrics

We evaluate the framework along two axes: fault detection discrimination and graph structure interpretability.

ROC-AUC. The Receiver Operating Characteristic curve plots the true positive rate against the false positive rate as the decision threshold varies. The area under this curve quantifies how well the anomaly score separates faulty from fault-free windows:

$$\text{ROC-AUC} = \int_0^1 \text{TPR}\big(\text{FPR}^{-1}(t)\big)\, dt \tag{12}$$

A score of 1.0 indicates perfect separation; 0.5 corresponds to random ranking.

PR-AUC. The Precision–Recall curve plots precision against recall. Its area is more informative than ROC-AUC when the class distribution is imbalanced, as is the case when fault-free windows substantially outnumber faulty ones:

$$\text{PR-AUC} = \int_0^1 \text{Precision}\big(\text{Recall}^{-1}(t)\big)\, dt \tag{13}$$

A high PR-AUC confirms that elevated anomaly scores correspond predominantly to true faults rather than false alarms.

Hit-Rate@k. To assess whether the learned adjacency matrix encodes physically meaningful structure, we rank all $N(N-1)$ directed sensor pairs by their attention weight (for GAT) or absolute Pearson correlation (for the baseline) and compute the fraction of the top-k pairs that match the 22 ground-truth physical pairs:

$$\text{Hit-Rate@}k = \frac{|\mathcal{P}_k \cap \mathcal{G}|}{k} \tag{14}$$

where $\mathcal{P}_k$ is the set of top-k ranked pairs and $\mathcal{G}$ is the ground-truth set. A high hit-rate indicates that the ranking assigns strong weights to pairs with known physical coupling; a low hit-rate suggests that the ranking reflects reconstruction utility rather than process topology.

4.8. Experimental Setup

Table 1 summarizes the hyperparameter configurations evaluated. All experiments share: GRU hidden dimension 128, 2 GRU layers, temporal attention embedding dimension 64, stride 25, batch size 64, learning rate 10^{-3} with scheduling of `ReduceLROnPlateau` , gradient clipping at norm 1.0, and early stopping with patience 15.

Table 1. Hyperparameter configurations of experiments (`kernel`=7).

Exp.	`feat_embed`	`window`	`dropout`
A	16	50	0.2
B	32	50	0.2
C	64	25	0.2
D	32	25	0.3
E	64	50	0.4

Each experiment trains a model on fault-free data (400 simulation runs for training, 100 for validation) and evaluates on 20 per-fault test sets, each combining fault-free test windows (label 0) with one fault type's windows (label 1). We report ROC-AUC and PR-AUC per fault and averaged across all 20 faults, along with Hit-Rate@k at $k = 10, 20, 30, 50$ for both GAT attention and Pearson correlation against the 22 ground-truth pairs.

5. RESULTS

5.1. Fault Detection Performance

Table 2 presents the mean detection performance across all 20 TEP faults. All window-size-50 configurations achieve comparable performance (mean ROC-AUC $\approx$ 0.77, mean PR-AUC ≈ 0.80), indicating that detection capability is robust to the feature attention embedding dimension and dropout rate. Smaller windows ($T = 25$) reduce performance slightly, consistent with the loss of temporal context for the GRU encoder. Per-fault detection performance for Experiment A is detailed in Table 3; we show the results of this experiment since it gives the best results.

Table 2. Mean detection performance across 20 TEP faults (`f_embed` is feature embedding, `w` is the window size and `drop` is the dropout rate).

Exp.	`f_embed`	`w`	`drop`	ROC-AUC	PR-AUC
A	16	50	0.2	0.772	0.803
B	32	50	0.2	0.771	0.803
C	64	25	0.2	0.752	0.784
D	32	25	0.3	0.754	0.786
E	64	50	0.4	0.771	0.802

Table 4 situates our detection performance against published TEP results for representative reconstruction- and graph-based methods. Direct numerical comparison is constrained by heterogeneous evaluation protocols across the literature: studies differ in the train/test split (fault-free run counts, sample-level versus run-level partitioning), the subset of faults reported (most work excludes Faults 3, 9, and 15, which are widely regarded as undetectable, while we include all 20), the anomaly-score aggregation (per-sample versus windowed), and the metric definition (some report fault detection rate at a fixed false alarm rate rather than ROC-AUC). Our 20-fault mean ROC-AUC of 0.77 and PR-AUC of 0.80 fall within the range reported for comparable reconstruction-driven graph methods on TEP.

Table 3. Per-fault detection performance for Experiment A. Faults marked † are widely regarded as undetectable in the TEP literature and are routinely excluded from comparison studies.

F.	Type	ROC	PR
1	Step	0.700	0.769
2	Step	0.640	0.696
3†	Step	0.500	0.500
4	Step	0.515	0.545
5	Step	0.633	0.702
6	Step	0.912	0.938
7	Step	0.695	0.767
8	Random	0.927	0.954
9†	Random	0.537	0.527
10	Random	0.900	0.931
11	Random	0.887	0.921
12	Random	0.931	0.956
13	Drift	0.918	0.948
14	Sticking	0.918	0.947
15†	Sticking	0.519	0.516
16	Unknown	0.804	0.833
17	Unknown	0.920	0.949
18	Unknown	0.893	0.927
19	Unknown	0.802	0.811
20	Unknown	0.885	0.919
Mean (all 20 faults): ROC-AUC 0.772, PR-AUC 0.803			
Mean (excl. F3, F9, F15): ROC-AUC 0.816, PR-AUC 0.854			

Table 4. Reported TEP detection performance for representative methods. Protocols differ across studies; values are reproduced as published.

Method	Faults	Reported metric
TGGL (Jia et al., 2023)	20/20	FDR ≈ 0.83
CNN-GAT (Liu & Jafarpour, 2024)	20/20	FDR ≈ 0.85
This work (Exp. A)	20/20	ROC-AUC 0.77 PR-AUC 0.80

TEP F1 is taken from a re-implementation reported by Jia et al. (2023).
FDR = Fault Detection Rate at fixed false alarm rate.

5.2. Graph Structure Evaluation

Table 5 reports the Hit-Rate@k for GAT attention and Pearson correlation against the 22 physical ground-truth pairs, for Experiment A (other configurations yield similar results).

Table 5. Hit-Rate@k: fraction of top-k pairs matching physical ground truth (22 pairs total). GAT = learned attention; Pearson = $|\rho_{ij}|$ baseline.

k	GAT hits	GAT rate	Pearson hits	Pearson rate
10	0	0.0%	4	40.0%
20	1	5.0%	4	20.0%
30	2	6.7%	6	20.0%
50	2	4.0%	7	14.0%

Pearson correlation consistently recovers more physical pairs

than GAT attention at all values of k. The six pairs recovered by Pearson in the top-30 are direct control-loop connections with near-unit linear correlation (e.g., $xmeas_{12} \leftrightarrow xmv_7$, $xmeas_{15} \leftrightarrow xmv_8$, $xmeas_{17} \leftrightarrow xmv_{11}$), the strongest and most obvious structural relationships in the TEP.

The GAT recovers only two physical pairs ($xmeas_7 \leftrightarrow xmv_6$ and $xmeas_3 \leftrightarrow xmv_2$), both representing feed–valve connections. Table 6 examines whether this is sensitive to hyperparameter choice. Whether the same effect persists across alternative GNN variants (e.g., the original GAT (Veličković et al., 2018), GCN (Kipf & Welling, 2017)) remains an open question: the failure mode we observe is consistent with the gradient pathology of reconstruction-driven attention identified by Brody et al. (2022) for the static-attention case, but the GATv2 layer used here is strictly more expressive than the original GAT, and the two may differ in their failure modes on this benchmark.

Table 6 confirms that this result is consistent across all hyperparameter configurations: the maximum hit count at $k = 30$ is 1 regardless of embedding dimension, window size, or dropout rate, indicating that the result is consistent across hyperparameter configurations rather than driven by any particular setting.

Table 6. Hit-Rate@30 across hyperparameter configurations.

Exp.	Config.	Hits@30	Rate
A	16 / 50 / 0.2	1	3.3%
B	32 / 50 / 0.2	1	3.3%
C	64 / 25 / 0.2	0	0.0%
D	32 / 25 / 0.3	0	0.0%
E	64 / 50 / 0.4	1	3.3%

5.3. Limitations

This study has several limitations. First, the TEP is a simulated benchmark; real processes present additional challenges including sensor noise, missing data, and more complex operating condition transitions. Second, evaluation is conducted under fixed operating conditions; testing under dynamic conditions is deferred to future work. Third, the ground-truth pairs are derived from process knowledge rather than a formally verified causal model. Fourth, we evaluate a single base architecture; other GNN variants may exhibit different attention-topology relationships.

6. CONCLUSION AND PERSPECTIVES

This paper presented a spatiotemporal GNN framework for fault detection under dynamic operating conditions and evaluated its spatial graph learning via GATv2 attention, temporal encoding on the Tennessee Eastman Process benchmark. Our findings indicate that:

1. The adapted MTAD-GAT architecture achieves competitive fault detection (mean ROC-AUC 0.77, mean PR-AUC 0.80) when trained exclusively on fault-free data, with performance robust to hyperparameter variations;
2. The learned GATv2 attention does *not* encode physically meaningful process topology;
3. This effect is robust across the hyperparameter space of the GATv2 architecture and is consistent with reconstruction-driven attention emphasising predictive utility over physical coupling; whether the same effect persists across alternative GNN variants is the subject of ongoing work.

These results establish that graph learning with spatiotemporal autoencoders is robust for fault detection, but that the learned graph structure is not a reliable proxy for physical process topology. This motivates several directions for future work. First, implementing the hybrid adjacency matrix $A = \alpha A_p + (1 - \alpha)S$ with an SDG-derived physical prior to anchor graph structure in known process physics. Second, incorporating a Deep SVDD dual-loss objective (Ruff et al., 2018) to tighten the boundary of the normal operating space beyond what reconstruction loss alone provides. Third, evaluating performance under dynamic operating conditions using real industrial dataset. Fourth, developing two-level causal interpretability combining GNN Explainer for edge-level subgraph extraction, to advance from detection toward root-cause diagnosis.

NOMENCLATURE

N	Number of sensors (features)
T	Temporal window size (timesteps)
B	Batch size
α_{ij}	GATv2 attention coefficient (sensor j to i)
A	Learned adjacency matrix, $A \in \mathbb{R}^{N \times N}$
A_p	Physics-derived adjacency matrix
S	Data-driven adjacency component
s_w	Window-level anomaly score
$r_{z,v}$	Similarity coefficient (fault z, variable v)
$\hat{x}$	Reconstructed observation

ACKNOWLEDGEMENT

This work is supported by IFP Energies Nouvelles and the Research Centre for Automatic Control of Nancy (CRAN, UMR CNRS 7039), Université de Lorraine.

REFERENCES

Alauddin, M., Khan, F., Imtiaz, S., & Ahmed, S. (2018). A bibliometric review and analysis of data-driven fault detection and diagnosis methods for process systems. *Industrial & Engineering Chemistry Research*, *57*(32), 10801–10823. doi: 10.1021/acs.iecr.8b02091

Brody, S., Alon, U., & Yahav, E. (2022). How attentive are graph attention networks? In *Proceedings of the inter-*

national conference on learning representations (iclr).

Chen, D., Liu, R., Hu, Q., & Ding, S. X. (2023). Interaction-aware graph neural networks for fault diagnosis of complex industrial processes. *IEEE Transactions on Neural Networks and Learning Systems*, *34*(9), 6015–6028. doi: 10.1109/TNNLS.2021.3132376

Chiang, L. H., Russell, E. L., & Braatz, R. D. (2001). *Fault detection and diagnosis in industrial systems*. London: Springer.

Deng, A., & Hooi, B. (2021). Graph neural network-based anomaly detection in multivariate time series. In *Proceedings of the aaai conference on artificial intelligence* (Vol. 35, pp. 4027–4035). AAAI Press.

Downs, J. J., & Vogel, E. F. (1993). A plant-wide industrial process control problem. *Computers & Chemical Engineering*, *17*(3), 245–255.

Filonov, P., Lavrentyev, A., & Vorontsov, A. (2016). Multivariate industrial time series with cyber-attack simulation: Fault detection using an lstm-based predictive data model. In *Nips 2016 time series workshop.* (arXiv:1612.06676)

Jia, M., Yang, T., Wang, Y., Xu, H., & Liu, B. (2023). Topology-guided graph learning for process fault diagnosis. *Industrial & Engineering Chemistry Research*, *62*(7), 3238–3251. doi: 10.1021/acs.iecr.2c03628

Kipf, T. N., & Welling, M. (2017). Semi-supervised classification with graph convolutional networks. In *Proceedings of the international conference on learning representations (iclr).* Toulon, France.

Kovalenko, I., Pozdnyakov, V., & Makarov, I. (2024). GNN with trainable adjacency matrix for fault diagnosis. *IEEE Access*, *12*, 152860–152874. doi: 10.1109/ACCESS.2024.3481331

Liu, Y., & Jafarpour, B. (2024). CNN-GAT with Granger causality for process monitoring. *Computers & Chemical Engineering*, *180*, 108453. doi: 10.1016/j.compchemeng.2023.108453

Maurya, M. R., Rengaswamy, R., & Venkatasubramanian, V. (2004). Application of signed digraphs-based analysis for fault diagnosis of chemical process flowsheets. *Engineering Applications of Artificial Intelligence*, *17*(5), 501–518.

Rieth, C. A., Amsel, B. D., Tran, R., & Cook, M. B. (2017). Additional Tennessee Eastman process simulation data for anomaly detection evaluation. *Harvard Dataverse*, *1*.

Ruff, L., Vandermeulen, R., Görnitz, N., Deecke, L., Siddiqui, S. A., Binder, A., … Kloft, M. (2018). Deep one-class classification. In *Proceedings of the international conference on machine learning (icml)* (pp. 4393–4402).

Sakurada, M., & Yairi, T. (2014). Anomaly detection using autoencoders with nonlinear dimensionality reduction. In *Proceedings of the mlsda workshop* (pp. 4–11).

Tidriri, K., Chatti, N., Verron, S., & Tiplica, T. (2018). Model-based fault detection and diagnosis of complex chemical processes: A case study of the Tennessee Eastman process. *Proceedings of the IMechE, Part I: Journal of Systems and Control Engineering*, *232*(6), 742–760. doi: 10.1177/0959651818764510

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., & Bengio, Y. (2018). *Graph attention networks.* Retrieved from `https://arxiv.org/abs/1710.10903`

Venkatasubramanian, V., Rengaswamy, R., Yin, K., & Kavuri, S. N. (2003). A review of process fault detection and diagnosis: Part I—Quantitative model-based methods. *Computers & Chemical Engineering*, *27*(3), 293–311.

Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., & Yu, P. S. (2021). A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, *32*(1), 4–24.

Zhao, H., Wang, Y., Duan, J., Huang, C., Cao, D., Tong, Y., … Zhang, Q. (2020). Multivariate time-series anomaly detection via graph attention network. In *Proceedings of the ieee international conference on data mining (icdm)* (pp. 841–850).

APPENDIX A: TENNESSEE EASTMAN PROCESS VARIABLES AND FAULT DESCRIPTIONS

Table 7. TEP continuous process measurements (Vertical Layout).

Var.	Name	Unit
xmeas_1	A Feed (stream 1)	kscmh
xmeas_2	D Feed (stream 2)	kg/hr
xmeas_3	E Feed (stream 3)	kscmh
xmeas_4	A and C Feed (stream 4)	kscmh
xmeas_5	Recycle Flow (stream 8)	kscmh
xmeas_6	Reactor Feed Rate (stream 6)	kscmh
xmeas_7	Reactor Pressure	kPa gauge
xmeas_8	Reactor Level	%
xmeas_9	Reactor Temperature	°C
xmeas_10	Purge Rate (stream 9)	kscmh
xmeas_11	Product Sep Temp	°C
xmeas_12	Product Sep Level	%
xmeas_13	Prod Sep Pressure	kPa gauge
xmeas_14	Prod Sep Underflow (stream 10)	m^3/hr
xmeas_15	Stripper Level	%
xmeas_16	Stripper Pressure	kPa gauge
xmeas_17	Stripper Underflow (stream 11)	m^3/hr
xmeas_18	Stripper Temperature	°C
xmeas_19	Stripper Steam Flow	kg/hr
xmeas_20	Compressor Work	kW
xmeas_21	Reactor CW Outlet Temp	°C
xmeas_22	Separator CW Outlet Temp	°C

Note: kscmh = thousand standard cubic metres per hour.

Table 8. TEP composition analyser variables (Unit is in %mol).

Variables	Description	Sampling (h)	Dead Time (h)
xmeas_23 to xmeas_29	Reactor Feed Analysis (Stream 6)	0.1	0.1
xmeas_30 to xmeas_37	Purge Gas Analysis (Stream 9)	0.1	0.1
xmeas_38 to xmeas_41	Product Analysis (Stream 11)	0.25	0.25

Table 9. TEP manipulated variables.

Var.	Name	Unit	Var.	Name	Unit
xmv_1	D Feed Flow (stream 2)	kg/hr	xmv_7	Sep Pot Liquid Flow (stream 10)	m^3/hr
xmv_2	E Feed Flow (stream 3)	kg/hr	xmv_8	Stripper Product Flow (stream 11)	m^3/hr
xmv_3	A Feed Flow (stream 1)	kscmh	xmv_9	Stripper Steam Valve	%
xmv_4	A and C Feed Flow (stream 4)	kscmh	xmv_10	Reactor Cooling Water Flow	m^3/hr
xmv_5	Compressor Recycle Valve	%	xmv_11	Condenser Cooling Water Flow	m^3/hr
xmv_6	Purge Valve (stream 9)	%			

Table 10. TEP fault descriptions and types.

F.	Description	Type	F.	Description	Type
1	A/C feed ratio, B comp. (Str. 4)	Step	10	C feed temp. (Str. 4)	Random
2	B comp., A/C ratio (Str. 4)	Step	11	Reactor CW inlet temp.	Random
3	D feed temp. (Str. 2)	Step	12	Condenser CW inlet temp.	Random
4	Reactor CW inlet temp.	Step	13	Reaction kinetics	Slow Drift
5	Condenser CW inlet temp.	Step	14	Reactor CW valve	Sticking
6	A feed loss (Str. 1)	Step	15	Condenser CW valve	Sticking
7	C header pressure loss (Str. 4)	Step	16	Unknown	—
8	A, B, C comp. (Str. 4)	Random	17	Unknown	—
9	D feed temp. (Str. 2)	Random	18–20	Unknown	—

APPENDIX B: GROUND-TRUTH PHYSICAL PAIRS

Table 11 lists the 22 ground-truth sensor pairs used in the Hit-Rate@k evaluation (Section 4.6).

Table 11. The 22 ground-truth physical sensor pairs used for Hit-Rate@k evaluation, grouped by source of process knowledge.

#	Sensor A	Sensor B	Description
Category 1 — Control loops (11 pairs)			
1	xmeas_7	xmv_4	Reactor pressure / A+C feed flow
2	xmeas_7	xmv_6	Reactor pressure / purge valve
3	xmeas_8	xmv_10	Reactor level / reactor CW flow
4	xmeas_9	xmv_7	Reactor temperature / separator liquid flow
5	xmeas_10	xmv_1	Purge rate / D feed flow
6	xmeas_11	xmv_2	Product sep. temperature / E feed flow
7	xmeas_12	xmv_7	Product sep. level / separator liquid flow
8	xmeas_15	xmv_8	Stripper level / stripper product flow
9	xmeas_17	xmv_11	Stripper underflow / condenser CW flow
10	xmeas_13	xmv_3	Product sep. pressure / A feed flow
11	xmeas_18	xmeas_8	Stripper temperature / reactor level
Category 2 — Material flow connections (8 pairs)			
12	xmeas_1	xmv_3	A feed measurement / A feed flow setpoint
13	xmeas_2	xmv_1	D feed measurement / D feed flow setpoint
14	xmeas_3	xmv_2	E feed measurement / E feed flow setpoint
15	xmeas_4	xmeas_6	A+C feed / reactor feed rate
16	xmeas_6	xmeas_7	Reactor feed rate / reactor pressure
17	xmeas_9	xmeas_11	Reactor temp. / product sep. temperature
18	xmeas_11	xmeas_15	Product sep. temp. / stripper level
19	xmeas_10	xmeas_18	Purge rate / stripper temperature
Category 3 — Thermodynamic couplings (3 pairs)			
20	xmeas_9	xmeas_8	Reactor temperature / reactor level (energy balance)
21	xmeas_7	xmeas_8	Reactor pressure / reactor level (ideal-gas law)
22	xmeas_21	xmeas_8	Reactor CW outlet temp. / reactor level (heat transfer)

APPENDIX C: FEATURE-ATTENTION MATRICES

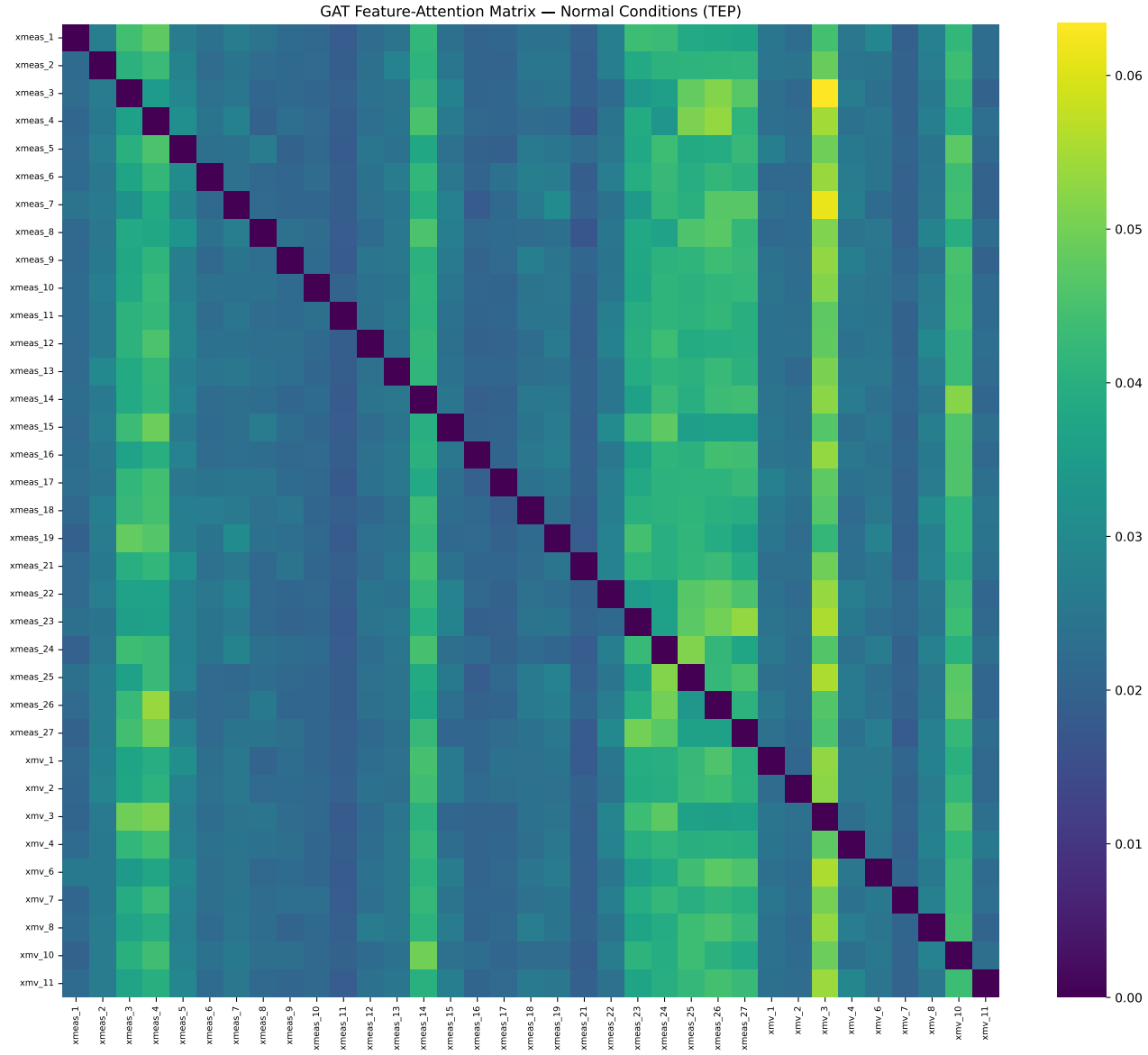

Figure 4. GATv2 feature-attention matrix learned under normal operating conditions (35 features, Experiment A). Bright vertical stripes indicate hub columns — sensors that receive high attention from many source nodes.

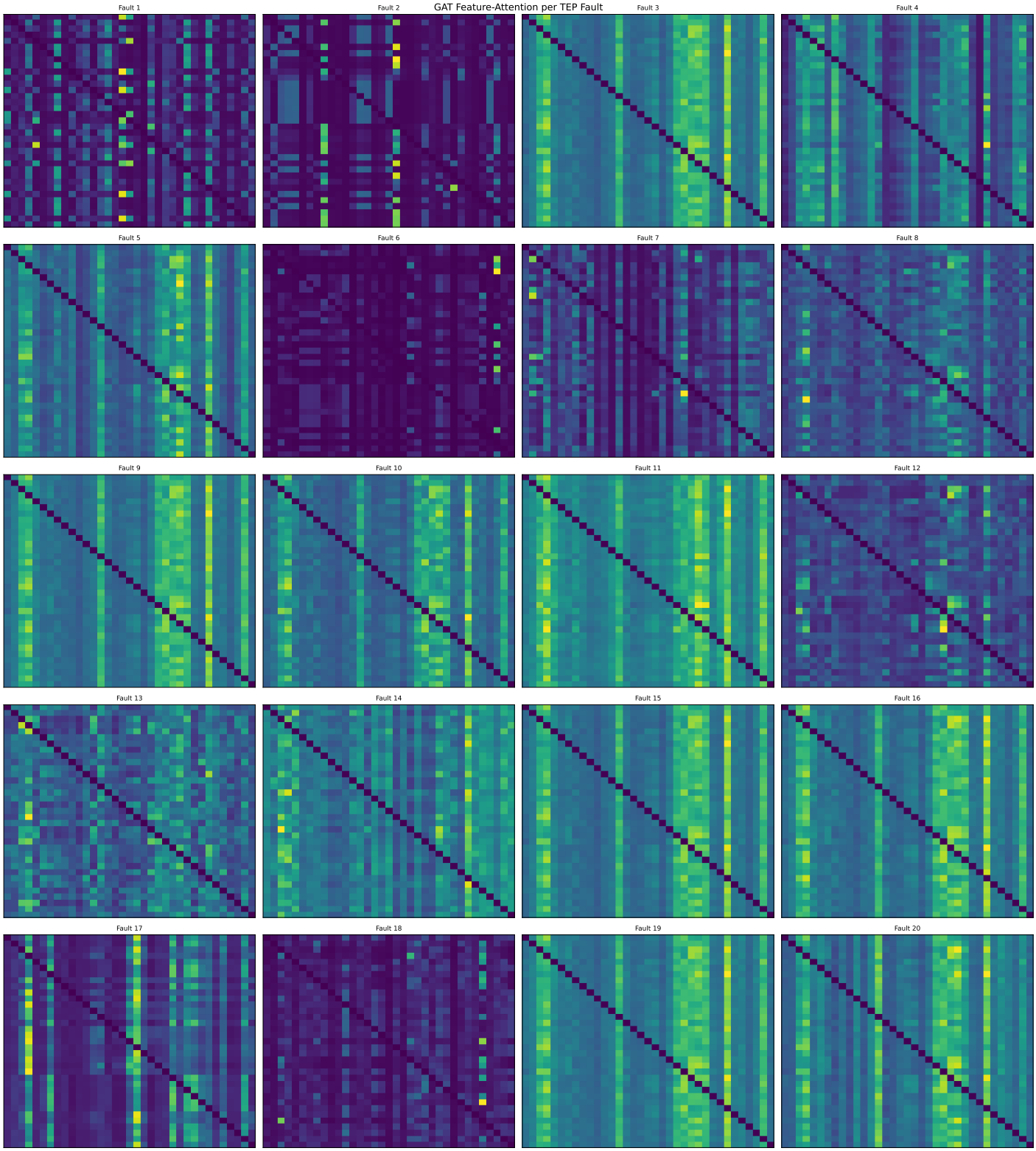

Figure 5. GATv2 feature-attention matrices computed on each of the 20 TEP fault scenarios (Faults 1–20, left-to-right, top-to-bottom). Each panel shares the same axes, ordering, and colour scale as Figure 4: rows and columns index the 35 selected sensors.