Domain-Prior-Regularized Graph Modeling for Anomaly Detection in Cyber-Physical Systems

Anonymous Authors[1]

Abstract

Anomaly detection on multivariate sensor time series is critical for industrial monitoring of cyber-physical systems (CPS), where even subtle deviations from normal behavior can indicate process disruption. Recent graph-based approaches have made significant progress, but they often struggle in small-scale physical systems with scarce labeled anomalies and limited normal data. In such settings, graph-based models tend to capture spurious correlations and produce unstable sensor topologies. We propose DPR-GM (**D**omain-**P**rior-**R**egularized **G**raph **M**odeling), a forecasting-based framework that incorporates system design knowledge into graph construction. DPR-GM leverages a large language model (LLM) to extract directed physical couplings between sensor pairs from system documentation, which are encoded as a binary domain adjacency matrix serving as a structural gate over sensor relations. This gate is then modulated by Pearson correlations estimated from normal training data. The anomaly score is further weighted by sensor-level reliability derived from the coefficient of variation. All graph and weighting components are fixed prior to training and add no learnable parameters. On the SKAB benchmark, DPR-GM outperforms graph-based, statistical, and deep learning baselines across F1, AUROC, and AUPRC, showing that domain-structured graph priors are a practical alternative to fully learned topologies in data-scarce CPS.

1. Introduction

Anomaly detection in multivariate sensor time series is a critical task in industrial cyber-physical systems (CPS), such as water treatment plants, water distribution networks, and chemical process control systems (Mathur & Tippenhauer, 2016; Ahmed et al., 2017; Downs & Vogel, 1993; Katser & Kozitsin, 2020). In such systems, early fault or attack identification can prevent costly downtime, safety incidents, and operational disruptions. Because labeled fault examples are scarce in practice, the problem is commonly approached in an unsupervised setting, where models learn normal dynamics from clean operational data and flag deviations at inference time (Ruff et al., 2018; Su et al., 2019a; Zhang et al., 2019).

Recent work has shown that modeling *inter-sensor relationships* via Graph Neural Networks (GNNs) substantially improves detection quality by capturing propagation patterns across physically coupled sensors (Deng & Hooi, 2021; Zhao et al., 2020a; Dai & Chen, 2022a; Liu et al., 2025). However, existing approaches construct the sensor graph either by end-to-end learning (Deng & Hooi, 2021; Zhang et al., 2022) or by thresholding empirical correlations, both of which are unreliable when normal-operation data is limited. Learned graphs absorb spurious correlations, and correlation-only graphs are destabilized by transient co-movements unrelated to physical causality.

Figure 1 illustrates how system design documentation can constrain noisy data-driven graphs and support more precise anomaly localization. In many CPS, the system design already encodes physical sensor coupling, subsystem membership, and output variable roles. This *domain prior* should constrain graph topology before any data-driven modulation is applied, yet current methods rely solely on inter-sensor correlation, producing noisy graphs and ambiguous localization.

We propose **DPR-GM** (**D**omain-**P**rior-**R**egularized **G**raph **M**odeling), a framework that uses a binary domain adjacency matrix as a structural gate, suppressing any edge not sanctioned by domain knowledge, while modulating permitted edge weights via Pearson correlation estimated from normal training data. Anomaly scoring further incorporates per-sensor reliability weights derived from the coefficient of variation (CV), giving higher influence to sensors that exhibit stable behavior under normal operation. Neither the graph structure nor the node weights involve any learnable parameters, making the prior entirely grounded in domain

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Submitted to the AI for Science workshop (ICML 2026).

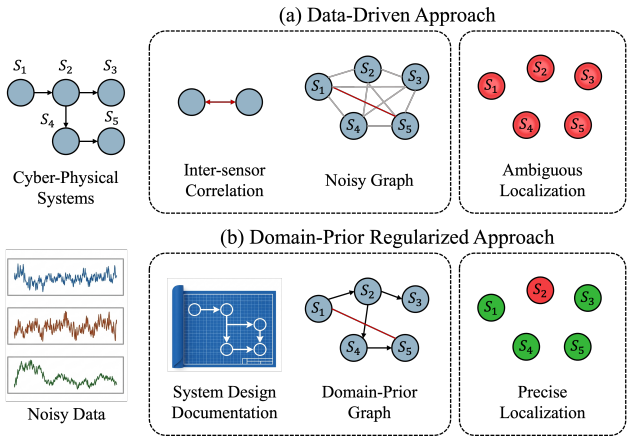

Figure 1. Motivation for domain-prior regularized graph modeling.

knowledge. In this work, we use a large language model (LLM) to automatically extract directed physical couplings from system documentation (Anthropic, 2026), broadening applicability to systems without manual expert annotation.

We evaluate DPR-GM on the SKAB benchmark (Katser & Kozitsin, 2020) under the strict point-wise evaluation protocol, the most demanding regime for cyber-physical system anomaly detection. Through systematic ablation over graph construction variants, we demonstrate that combining domain structure with data-driven edge modulation consistently outperforms each information source in isolation, and improves over graph-based, statistical, and deep learning baselines on threshold-free ranking metrics.

We summarize our contributions as follows:

- We propose **DPR-GM**, which gates sensor graph topology using domain knowledge and modulates edge weights via Pearson correlation, with CV-based reliability weighting for anomaly scoring.
- We demonstrate on SKAB that DPR-GM outperforms graph-based, statistical, and deep learning baselines on threshold-free ranking metrics under the strict point-wise evaluation protocol.
- We show that domain priors in DPR-GM can be automatically extracted from system documentation via LLM, broadening applicability to CPS without manual expert annotation.

2. Related Work

2.1. Multivariate Time-Series Anomaly Detection

Classical approaches to multivariate time-series anomaly detection include statistical process monitoring (Box et al., 2015; Hotelling, 1947), PCA-based reconstruction (Shyu et al., 2003), one-class classifiers (Schölkopf et al., 2001; Tax & Duin, 2004), and density-based methods such as Local Outlier Factor (Breunig et al., 2000) and Isolation Forest (Liu et al., 2008). These methods are computationally efficient and interpretable, but rely on stationarity or linearity assumptions and struggle to capture nonlinear temporal dynamics and high-order inter-variable dependencies.

Deep learning methods overcome these limitations by modeling normal temporal dynamics using forecasting, reconstruction, or hybrid objectives. Forecasting-based models detect anomalies when future observations deviate from predicted values. LSTM-NDT (Hundman et al., 2018) applies stacked LSTMs with non-parametric dynamic thresholding, while THOC (Shen et al., 2020) introduces a dilated RNN with a temporal hierarchical one-class objective. TimesNet (Wu et al., 2023) reshapes 1D time series into 2D tensors via FFT-based period decomposition to jointly capture intra- and inter-period variations. Reconstruction-based models detect anomalies as poorly reconstructed inputs. LSTM-VAE (Park et al., 2018) models probabilistic temporal dynamics but ignores explicit inter-variable dependencies; OmniAnomaly (Su et al., 2019b) addresses this via stochastic recurrent networks with Gaussian state-space models; USAD (Audibert et al., 2020) further improves stability through an adversarial autoencoder framework. Hybrid methods combine both objectives: TranAD (Tuli et al., 2022) uses attention-based encoders with adversarial training; Anomaly Transformer (Xu et al., 2022) introduces an Association Discrepancy criterion that exploits the weaker series associations of anomalous points.

Despite their effectiveness, these non-graph methods treat inter-sensor dependencies only implicitly through shared latent representations. In CPS, faults propagate across electrically, mechanically, hydraulically, and thermally coupled subsystems. Without explicit relational structure, models cannot distinguish a physically meaningful dependency change from an independent channel-level fluctuation, and with limited normal-operation data they have little incentive to learn dependency structure that matches physical causal pathways.

2.2. Graph-Based Multivariate Time-Series Anomaly Detection

Graph-based approaches address this by representing sensors as nodes and inter-sensor dependencies as edges, enabling GNNs to combine spatial and temporal modeling (Wu et al., 2020; Jin et al., 2024). GDN (Deng & Hooi, 2021) learns a sparse sensor graph through embedding similarity and scores anomalies as normalized forecast deviations. MTAD-GAT (Zhao et al., 2020b) applies graph attention over both feature and temporal dimensions with a joint forecasting and reconstruction objective. GReLeN (Zhang

et al., 2022) infers latent graph structure via variational relational inference, using changes in the inferred structure itself as an anomaly signal. Another direction treats the sensor graph as a directed or causal structure. GANF (Dai & Chen, 2022b) places a Bayesian network over time series and decomposes the joint density into conditional normalizing flows for density-based anomaly scoring. GCAD (Liu et al., 2025) estimates dynamic Granger causality and defines anomalies as deviations from learned causal dependency patterns. DVGCRN (Chen et al., 2022) further combines graph learning with variational recurrent modeling to capture fine-grained spatio-temporal dependencies.

However, most graph-based methods rely on data-driven graph construction, which is unreliable when normal-operation data are limited. Learned correlations may reflect coincidental statistical patterns rather than physical causal pathways, and the resulting graph can be unstable across training runs. DPR-GM addresses this by using a directed domain adjacency matrix as an explicit structural prior, anchoring the graph in known physical coupling rather than treating it as a free parameter to be learned. DPR-GM also differs from prior methods by incorporating sensor reliability into anomaly scoring via the coefficient of variation rather than learning channel importance end-to-end, consistent with the finding in CATCH (Wu et al., 2024) that channel importance is critical for multivariate anomaly detection.

3. Methodology

We present **DPR-GM**, a forecasting-based anomaly detection framework built on the principle that domain knowledge determines which edges may exist, while data statistics determine how strong those edges are. As illustrated in Figure 2, the framework proceeds in two stages: (1) *Domain-prior graph construction* uses an LLM to extract directed physical dependencies from system documentation, forming a binary domain adjacency matrix whose edge weights are modulated by signed Pearson correlation estimated from normal training data. (2) *GNN-based forecasting* encodes sensor time series with a shared temporal encoder and propagates them over the resulting domain-prior graph to generate multi-step predictions. Anomaly scores are further weighted by per-sensor reliability derived from the coefficient of variation.

3.1. Domain-Prior-Regularized Graph Construction

Domain prior extraction. In a physical sensor system, sensor relationships are governed by the underlying physical processes rather than being arbitrary. An upstream driving variable predictably propagates its effect to a downstream response variable through a known physical pathway, giving rise to a *directed* dependency that is asymmetric in general. Such structural knowledge is encoded in system design documentation before any data is collected, yet current graph-based methods discard it entirely.

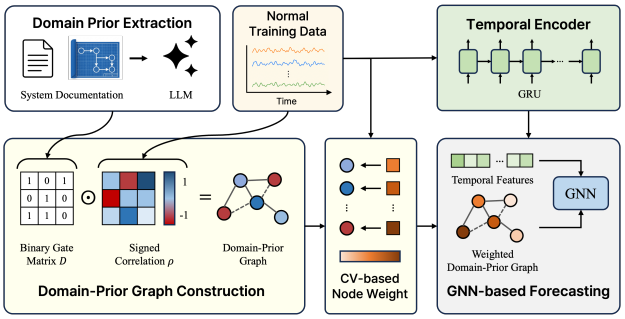

Figure 2. Overview of DPR-GM for domain-prior graph construction and GNN-based forecasting with CV-based node reliability weighting.

We extract this prior by prompting an LLM with system design documents, sensor metadata, and subsystem descriptions. Specifically, we use Claude Sonnet (Anthropic, 2026) in this work. The full prompt is provided in Appendix A. The prompt instructs the model to identify directed physical couplings between sensor pairs, specifying the source sensor, the target sensor, and the physical mechanism of influence. The output is a structured list of directed sensor pairs $(i \to j)$ that are physically sanctioned, which we use to construct the binary domain adjacency matrix $\mathbf{D}$. Directed structure that cannot be reliably recovered from correlation alone is thus made available to the model before training begins.

Domain adjacency matrix. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ denote the sensor graph, where each node $v_i \in \mathcal{V}$ corresponds to a physical sensor and each directed edge $(i, j) \in \mathcal{E}$ represents a physically sanctioned dependency from sensor i to sensor j. We encode this structural knowledge as a binary domain adjacency matrix $\mathbf{D} \in \{0, 1\}^{N \times N}$, where $D_{ij} = 1$ if and only if a directed physical coupling from sensor i to sensor j is specified by system design documentation. All entries outside this specification are set to zero unconditionally, regardless of any empirical evidence. The matrix $\mathbf{D}$ is therefore *asymmetric* in general: the presence of an edge (i, j) does not imply the existence of the reverse edge (j, i).

Correlation modulation. Domain knowledge determines the *topology* of the graph, but not the relative importance of individual edges. To modulate edge strength by empirical evidence, we estimate the Pearson correlation matrix $\boldsymbol{\rho} \in [-1, 1]^{N \times N}$ from normal training data $\mathcal{X}_{\text{train}}$ only:

$$\rho_{ij} = \frac{\text{Cov}(x_i, x_j)}{\sigma_i \sigma_j}, \tag{1}$$

where x_i denotes the time series of sensor i over the training set. The final DPR-GM adjacency matrix $\mathbf{A}$ is defined as:

$$A_{ij} = D_{ij} \cdot \big(0.5 + 0.5\,\rho_{ij}\big). \tag{2}$$

The formulation in Equation (2) encodes two properties simultaneously. First, $\mathbf{D}$ acts as a *binary gate*: any edge not sanctioned by domain knowledge receives zero weight regardless of empirical correlation, preventing spurious sensor-pair associations from entering the graph. Second, for domain edges, the signed correlation term maps positive, zero, and negative correlations to edge weights above, equal to, and below the neutral baseline of 0.5, respectively. Table 1 summarizes representative boundary cases of the resulting signed edge-weighting rule.

Table 1. Edge weight A_{ij} under representative signed-correlation conditions.

Condition	D_{ij}	ρ_{ij}	A_{ij}
Domain edge, positive correlation	1	1.0	1.0
Domain edge, no correlation	1	0.0	0.5
Domain edge, negative correlation	1	-1.0	0.0
No domain edge	0	any	0.0

The use of signed correlation ρ_{ij} encodes an additional semantic in the edge modulation rule. For domain edges, positively correlated sensor pairs receive larger message-passing weights, uncorrelated pairs receive a neutral weight of 0.5, and negatively correlated pairs are downweighted, reaching zero when $\rho_{ij} = -1$. This signed modulation makes the edge strength sensitive to the direction of normal-operation co-movement while still respecting the domain gate $\mathbf{D}$. The 0.5 baseline ensures that domain edges with no observed correlation remain structurally active, preventing physically motivated edges from being silenced solely by insufficient training statistics. Table 1 illustrates representative boundary cases of this signed edge-weighting rule.

3.2. CV-Based Node Reliability Weights

Sensors exhibit heterogeneous variability during normal operation. A sensor with low temporal variability in the normal regime constitutes a reliable reference point: deviations in its readings are more likely to reflect genuine fault conditions than measurement noise. We operationalize this intuition via the CV:

$$\mathrm{CV}_i = \frac{\sigma_i}{|\mu_i| + \varepsilon} \times 100, \tag{3}$$

and define the reliability score $\boldsymbol{r}$ and normalized node weight $\mathbf{w}$ as:

$$r_i = \frac{1}{\mathrm{CV}_i + \varepsilon}, \qquad w_i = \frac{r_i}{\sum_{j=1}^{N} r_j}, \tag{4}$$

where $\varepsilon > 0$ is a small constant for numerical stability. Figure 3 visually contrasts low-CV stable signals with high-CV variable signals underlying the proposed reliability weighting. Both $\mathbf{A}$ and $\mathbf{w}$ are computed exclusively from $\mathcal{X}_{\text{train}}$ and CPS documentation, and remain fixed throughout training and inference without introducing any data-driven parameters into the graph structure.

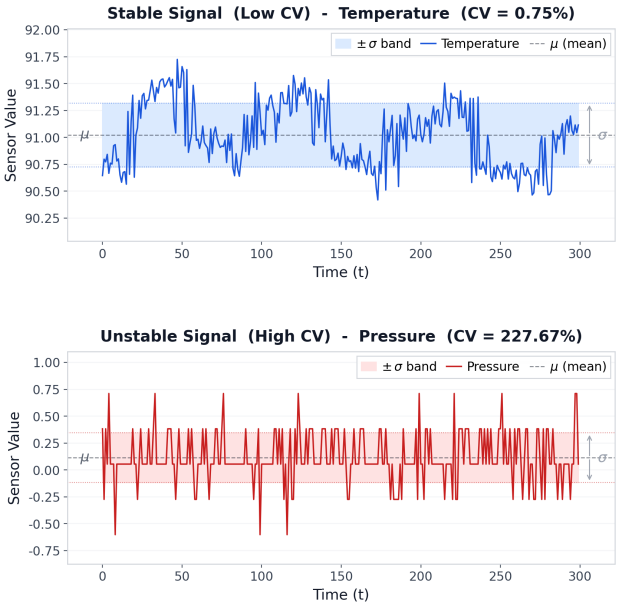

Figure 3. Illustration of CV-based node reliability weighting, where stable low-CV sensors receive higher reliability scores than highly variable high-CV sensors.

3.3. Model Architecture

Input. The model receives a sliding window $\mathbf{X} \in \mathbb{R}^{B\times T\times N}$, where B is the mini-batch size, T is the window length, and N is the number of sensors.

Temporal encoding. A weight-shared GRU is applied independently to each normalized sensor channel. The input is reshaped from $\mathbb{R}^{B\times T\times N}$ to $\mathbb{R}^{BN\times T\times 1}$ so that all N channels share the same recurrent weights. This parameter-efficient design treats the GRU as a common temporal motif encoder for standardized univariate sensor sequences, which helps reduce overfitting in small-scale datasets.

Let $\mathbf{H} \in \mathbb{R}^{BN\times T\times d_h}$ denote the hidden sequence produced by the shared GRU. To capture temporal information beyond the final hidden state, we aggregate the hidden sequence using the final state, temporal mean pooling, and temporal max pooling, followed by a linear projection:

$$\mathbf{h} = \mathrm{Linear}\big(\mathbf{h}_{\text{last}} \,\|\, \mathbf{h}_{\text{mean}} \,\|\, \mathbf{h}_{\text{max}}\big) \in \mathbb{R}^{BN\times d}. \tag{5}$$

Sensor identity embedding. Because the GRU is shared across channels, sensors with similar normalized temporal

patterns may produce similar hidden representations. We therefore concatenate a learnable sensor embedding $\mathbf{e}_i \in \mathbb{R}^{d_e}$ to each temporal representation before graph message passing, allowing the model to distinguish which physical sensor generated the pattern. The sensor embeddings are repeated across the batch to form $\mathbf{E} \in \mathbb{R}^{BN \times d_e}$:

$$\mathbf{h} \leftarrow \big[\mathbf{h} \,\|\, \mathbf{E}\big] \in \mathbb{R}^{BN \times (d+d_e)}. \tag{6}$$

Graph message passing. The DPR-GM adjacency $\mathbf{A}$ is converted to a sparse edge list. For each mini-batch, the same sensor graph is replicated B times and batched by node offset for a single GNN call. A general message-passing layer first aggregates incoming messages:

$$\mathbf{M}_{b,i}^{(\ell)} = \underset{j \in \mathcal{N}_{\text{in}}(i)}{\text{AGGREGATE}} \, \mathbf{m}\Big(\mathbf{h}_{b,i}^{(\ell)}, \mathbf{h}_{b,j}^{(\ell)}, A_{ji}\Big), \tag{7}$$

and then updates the node representation as

$$\mathbf{h}_{b,i}^{(\ell+1)} = \text{UPDATE}\Big(\mathbf{h}_{b,i}^{(\ell)}, \mathbf{M}_{b,i}^{(\ell)}\Big). \tag{8}$$

Here, $\mathcal{N}_{\text{in}}(i) = \{j \mid A_{ji} > 0\}$ denotes the in-neighborhood of node i, and A_{ji} denotes the edge weight from source node j to target node i. The function $\mathbf{m}(\cdot)$ denotes the backbone-specific message sent from source node j to target node i. It may apply a linear transformation, edge-weight modulation, or attention-based reweighting before the resulting messages are combined by the permutation-invariant aggregation operator. The specific instantiation of Equations (7)–(8) is determined by the choice of backbone.

We consider GCN (Kipf & Welling, 2016), GraphSAGE (Hamilton et al., 2017), GAT (Veličković et al., 2017), and Graph Transformer (Shi et al., 2020) backbones. In our implementation, GCN and weighted GraphSAGE explicitly consume the DPR-GM edge weights during message passing. GAT and Graph Transformer use only the nonzero graph support induced by $\mathbf{A}$, while their aggregation weights are learned from node features. Unless otherwise stated, we use weighted GraphSAGE as the default backbone.

Forecasting head. The node representations are projected to k-step-ahead predictions:

$$\hat{\mathbf{Y}} = \text{reshape}\big(\text{Linear}(\mathbf{h})\big) \in \mathbb{R}^{B \times k \times N}. \tag{9}$$

The model is trained to minimize the mean squared error (MSE) between $\hat{\mathbf{Y}}$ and the corresponding future ground-truth window $\mathbf{Y} \in \mathbb{R}^{B \times k \times N}$.

3.4. Anomaly Scoring

At inference, the per-sensor forecast error for each window is computed as:

$$e_{b,i} = \frac{1}{k} \sum_{\tau=1}^{k} \big(\hat{Y}_{b,\tau,i} - Y_{b,\tau,i}\big)^2. \tag{10}$$

To obtain a distribution-free normalized error, IQR normalization is applied using statistics fitted on the training set:

$$\tilde{e}_{b,i} = \max\left(0, \; \frac{e_{b,i} - \text{median}(e_i^{\text{train}})}{\text{IQR}(e_i^{\text{train}})}\right). \tag{11}$$

The window-level anomaly score aggregates the weighted per-sensor errors using a combination of a mean term and a max term:

$$s_b = (1-\alpha)\frac{1}{N}\sum_{i=1}^{N} w_i \, \tilde{e}_{b,i} \; + \; \alpha \max_i \; w_i \, \tilde{e}_{b,i}, \tag{12}$$

where $\alpha = 0.2$ and w_i are the CV-based reliability weights from Equation (4). The mean term captures distributed multi-sensor anomalies, while the max term retains sensitivity to localized high-amplitude deviations in a single sensor.

4. Experiments

4.1. Dataset

We evaluate DPR-GM on the **Skoltech Anomaly Benchmark (SKAB)** (Katser & Kozitsin, 2020), a multivariate time series benchmark collected from a physical water pump testbed under controlled industrial fault conditions. The testbed consists of a water pump, an electric motor, an inverter, two electromagnetic valves, and a set of sensors measuring current, voltage, vibration (two axes), pressure, flow rate, temperature, and thermocouple readings ($N = 8$ sensors in total). The dataset contains 34 fault scenarios across three splits: `valve1` (18,130 time steps), `valve2` (4,312 time steps), and `other` (14,920 time steps), totaling 37,362 time steps in the `overall` split. The anomaly fraction is approximately 35% across all splits. We use the `overall` split as the primary evaluation target.

4.2. Evaluation Protocol and Experimental Setup

We adopt a strict point-wise evaluation protocol, where each time step is scored independently and classified as anomalous only if its anomaly score exceeds a threshold. We do not use point adjustment, event-level matching, or window-based correction, since such post-processing can obscure whether a method detects anomalies at the correct time step. This conservative protocol is motivated by recent critiques showing that common benchmark and evaluation choices in time-series anomaly detection can lead to unreliable comparisons and an inflated impression of progress (Wu & Keogh, 2021). We therefore treat strict point-wise performance as the primary evaluation criterion.

Let TP, FP, TN, and FN denote the point-wise entries of the confusion matrix. We report Precision, Recall, F1 score, Matthews Correlation Coefficient (**MCC**), area under the

Table 2. Graph-based method comparison on SKAB (strict point-wise). Mean (std) over five repeated runs. **Bold** / underline: best / second-best in `overall`. See Section 4.3 for method references.

METHOD	FAULT	PRECISION	RECALL	F1	AUROC	AUPRC	BF-F1
GDN	VALVE1	0.4190(0.3873)	0.0003(0.0004)	0.0007(0.0007)	0.4947(0.0098)	0.3436(0.0063)	0.5166(0.0008)
	VALVE2	0.0000(0.0000)	0.0000(0.0000)	0.0000(0.0000)	0.5270(0.0112)	0.3604(0.0060)	0.5252(0.0057)
	OTHER	0.3555(0.3185)	0.1143(0.1262)	0.1493(0.1515)	0.5788(0.0872)	0.4672(0.0775)	0.5388(0.0221)
	OVERALL	0.4750(0.2713)	0.0460(0.0507)	0.0753(0.0795)	0.5314(0.0314)	0.3820(0.0228)	0.5223(0.0047)
GRELEN	VALVE1	0.3147(0.1668)	0.0996(0.1256)	0.1180(0.1433)	0.4836(0.0278)	0.3492(0.0149)	0.5261(0.0001)
	VALVE2	0.2199(0.1799)	0.1102(0.1346)	0.1283(0.1533)	0.4990(0.0171)	0.3601(0.0099)	0.5318(0.0005)
	OTHER	0.5154(0.1758)	0.0566(0.0523)	0.0947(0.0830)	0.5116(0.0633)	0.3899(0.0698)	0.5308(0.0001)
	OVERALL	**0.4866**(0.1872)	0.0836(0.0877)	0.1161(0.1138)	0.4971(0.0337)	0.3666(0.0348)	0.5286(0.0001)
MTAD-GAT	VALVE1	0.3493(0.0000)	1.0000(0.0000)	0.5178(0.0000)	0.6050(0.0058)	0.5234(0.0128)	0.5246(0.0001)
	VALVE2	0.3518(0.0000)	1.0000(0.0000)	0.5205(0.0000)	0.5898(0.0171)	0.4423(0.0142)	0.5277(0.0061)
	OTHER	0.4402(0.0011)	0.9174(0.0064)	0.5949(0.0008)	0.7506(0.0090)	0.6852(0.0103)	0.6051(0.0065)
	OVERALL	0.3795(0.0002)	**0.9669**(0.0026)	0.5450(0.0004)	0.5794(0.0027)	0.4566(0.0075)	0.5476(0.0018)
DPR-GM (OURS)	VALVE1	0.3480(0.0000)	1.0000(0.0000)	0.5163(0.0000)	0.8315(0.0154)	0.8225(0.0162)	0.7194(0.0165)
	VALVE2	0.3518(0.0000)	1.0000(0.0000)	0.5205(0.0000)	0.8616(0.0187)	0.8590(0.0131)	0.7817(0.0097)
	OTHER	0.4995(0.0428)	0.8890(0.0306)	0.6377(0.0267)	0.8131(0.0008)	0.7638(0.0007)	0.6781(0.0057)
	OVERALL	0.3922(0.0086)	0.9555(0.0123)	**0.5560**(0.0066)	**0.7256**(0.0058)	**0.7165**(0.0084)	**0.6063**(0.0094)

ROC curve (**AUROC**), and area under the precision-recall curve (**AUPRC**). MCC is defined as:

$$\mathrm{MCC} = \frac{\mathrm{TP}\cdot\mathrm{TN} - \mathrm{FP}\cdot\mathrm{FN}}{\sqrt{(\mathrm{TP}+\mathrm{FP})(\mathrm{TP}+\mathrm{FN})(\mathrm{TN}+\mathrm{FP})(\mathrm{TN}+\mathrm{FN})}}. \quad (13)$$

AUROC and AUPRC integrate threshold-free ranking quality over all decision boundaries where $\mathrm{TPR} = \mathrm{TP}/(\mathrm{TP} + \mathrm{FN})$, $\mathrm{FPR} = \mathrm{FP}/(\mathrm{FP} + \mathrm{TN})$, and Prec, Rec are precision and recall parameterized by threshold. For thresholded metrics, the anomaly threshold is selected by Peak-over-Threshold (**PoT**). The oracle best-F1 (**BF-F1**), obtained by sweeping all thresholds, is reported only as a diagnostic upper bound on classification performance.

Experimental setup. Unless otherwise specified, we use an input window length of $T = 30$, a forecasting horizon of $k = 10$, $N = 8$ sensors, and a batch size of $B = 256$. All models are trained on normal training windows and evaluated on held-out fault scenarios using the same preprocessing, windowing, thresholding, and metric computation pipeline. We report mean and standard deviation over five repeated runs. Details of the computational environment are provided in Appendix B.

4.3. Baselines

We compare DPR-GM with two groups of baselines. Graph-based baselines test the advantage of the proposed domain-prior graph over learned or inferred sensor graphs. Non-graph baselines assess the benefit of explicit physical dependency modeling over standard statistical and deep anomaly detection models without graph structure.

Graph-based baselines. We re-implement GDN (Deng & Hooi, 2021), which learns a sparse sensor graph from embedding similarity; MTAD-GAT (Zhao et al., 2020a), which applies feature- and time-oriented graph attention with forecasting and reconstruction losses; and GReLeN (Zhang et al., 2022), which infers latent relations via variational graph learning.

Non-graph baselines. We also compare against statistical methods (T^2+Q, T-squared, MSET), density-based detection (ISF (Liu et al., 2008)), and deep reconstruction-based models (Conv-AE, Vanilla AE, Vanilla LSTM (Hochreiter & Schmidhuber, 1997), LSTM-AE, LSTM-VAE (Park et al., 2018), MSCRED (Zhang et al., 2019)).

4.4. Results

Table 2 compares DPR-GM with graph-based anomaly detection methods across all SKAB fault splits. Table 3 compares DPR-GM with statistical and deep learning baselines on the `overall` split. Table 4 evaluates four GNN architectures within the same DPR-GM framework. Table 5 reports an ablation study over five repeated runs that isolates the contribution of each design component.

Graph-based comparison (Table 2). GDN and GReLeN collapse on SKAB, achieving near-zero F1 scores of 0.075 and 0.116, respectively, with large run-to-run variance. This failure mode is characteristic of end-to-end graph learning under limited normal-operation data. Without sufficient observations to stabilize the learned topology, the model captures spurious correlations, yielding graph structures that vary across runs and degrade detection quality. Because neither method is anchored by domain knowledge, any transient co-movement in the training window can enter the graph as a false edge, leading to inconsistent message-passing behavior at inference.

Table 3. Comparison with statistical and deep learning baselines on SKAB `overall` (strict point-wise, PoT threshold). Mean (std). **Bold**: best overall. Underline: best among non-graph baselines.

METHOD	PRECISION	RECALL	F1	MCC	AUROC	AUPRC	BF-F1
T^2+Q	0.3737	0.9693	0.5394	0.1571	0.6656	0.6133	0.5490
T-SQUARED	0.3745	0.9705	0.5404	0.1614	0.6737	0.6246	0.5580
MSET	0.3695(0.0017)	0.8422(0.0138)	0.5136(0.0042)	0.0823(0.0091)	0.6388(0.0003)	0.5908(0.0006)	0.5336(0.0004)
ISF	0.3452(0.0034)	0.5272(0.1538)	0.4077(0.0591)	−0.0086(0.0056)	0.5105(0.0013)	0.3494(0.0020)	0.5316(0.0007)
CONV-AE	0.3567(0.0025)	0.9942(0.0067)	0.5250(0.0022)	0.0911(0.0155)	0.6296(0.0239)	0.5428(0.0476)	0.5491(0.0026)
VANILLA AE	0.3751(0.0013)	0.9697(0.0003)	0.5409(0.0013)	0.1631(0.0054)	0.6374(0.0015)	0.5634(0.0018)	0.5481(0.0022)
VANILLA LSTM	0.3627(0.0010)	0.9821(0.0023)	0.5298(0.0010)	0.1149(0.0052)	0.6359(0.0013)	0.5623(0.0036)	0.5558(0.0003)
LSTM-AE	0.3557(0.0009)	0.9986(0.0011)	0.5246(0.0009)	0.0919(0.0062)	0.6337(0.0012)	0.5608(0.0019)	0.5525(0.0009)
LSTM-VAE	0.3550(0.0010)	**0.9993**(0.0015)	0.5238(0.0011)	0.0873(0.0093)	0.6694(0.0019)	0.6194(0.0041)	0.5507(0.0013)
MSCRED	0.3690(0.0002)	0.9721(0.0004)	0.5349(0.0002)	0.1381(0.0008)	0.7029(0.0186)	0.6809(0.0312)	0.5809(0.0309)
DPR-GM (OURS)	**0.3922**(0.0086)	0.9555(0.0123)	**0.5560**(0.0066)	**0.2129**(0.0181)	**0.7256**(0.0058)	**0.7165**(0.0084)	**0.6063**(0.0094)

MTAD-GAT avoids this collapse by deriving aggregation weights from node features via attention rather than learning the graph topology end-to-end, recovering competitive F1 (0.545) through near-universal recall. However, its AUROC (0.579) and AUPRC (0.457) remain substantially weaker, revealing that high F1 here is a threshold artifact rather than a reflection of genuine score discrimination. A method that flags nearly every time step as nomalous will inevitably achieve high recall, but its anomaly scores carry little information about the true degree of anomalousness at each point.

DPR-GM addresses both failure modes by grounding the graph in domain knowledge before any data-driven modulation is applied. The binary domain adjacency matrix eliminates spurious edges unconditionally, providing the stable topology that learned methods cannot recover from limited data. Signed Pearson correlation then modulates permitted edge weights by empirical co-movement, so the graph encodes both structural and statistical information without conflating the two. The gains are consistent across all three fault splits. On `valve1` and `valve2`, DPR-GM achieves AUROC above 0.83 and 0.86 respectively, substantially outperforming MTAD-GAT, indicating that domain-prior structure is particularly effective for valve-type faults where physical coupling between sensors is well-defined. On `other`, DPR-GM still achieves AUROC 0.813 and AUPRC 0.764, suggesting that the domain prior generalizes beyond the fault types present during graph construction. The consistency of these improvements further supports the view that DPR-GM's gains stem from a structurally sounder graph rather than from overfitting to a particular anomaly pattern. Overall, DPR-GM attains the best F1, AUROC, AUPRC, and BF-F1 simultaneously, evidence that domain-structured priors yield discriminative anomaly scores rather than recall-biased ones.

Non-graph comparison (Table 3). Among non-graph baselines, Vanilla AE attains the strongest threshold-dependent performance, while MSCRED leads on all threshold-free metrics. DPR-GM improves over every non-graph baseline on every reported metric, with AUROC and AUPRC gains of $+0.023$ and $+0.036$ over MSCRED. The MCC improvement over Vanilla AE ($+0.050$) is particularly informative: because MCC penalizes class-imbalance-driven strategies more strictly than F1, this gain suggests that DPR-GM's decision boundary is genuinely better calibrated rather than simply exploiting the skewed base rate of anomalies. In contrast, classical statistical methods sustain near-saturated recall yet trail substantially on threshold-free metrics, the signature of over-detection.

Backbone comparison (Table 4). We evaluate four GNN backbones within the same DPR-GM framework to identify the most effective aggregation strategy for the domain-prior graph. In our implementation, GCN and SAGE explicitly consume DPR-GM edge weights during message passing, whereas GAT and GT use only the nonzero graph support and learn attention weights from node features. GT follows as a close second on point estimates but exhibits notably higher run-to-run variance. GT's attention mechanism, which is computed from node features rather than domain-prior edge weights, appears to be more sensitive to variations in input representations across runs, trading stability for representational flexibility. SAGE achieves the best performance across every reported metric. The gap between GCN and SAGE suggests that SAGE's separation of self- and neighbor-transformations interacts more favorably with sparse domain-prior graphs than GCN's symmetric normalization. GAT performs substantially worse than all three alternatives since GAT differs from the edge-weight-aware backbones primarily in discarding the prior weights and re-deriving attention from node features. Overall, these results indicate that explicitly consuming domain-prior edge weights during message passing is more effective than re-learning interaction strength from node features, and that among edge-weight-aware backbones, the choice of aggregation scheme further modulates how well the prior is exploited.

Table 4. Backbone comparison on SKAB `overall` over five repeated runs under the strict point-wise protocol. Mean (std). **Bold** / underline: best / second-best. SAGE: GraphSAGE. GT: Graph Transformer.

Backbone	Edge-W	Prac-F1	AUROC	AUPRC
GCN	Yes	0.5471(0.0021)	0.7145(0.0268)	0.6981(0.0371)
GAT	No	0.5236(0.0029)	0.6091(0.0042)	0.5023(0.0054)
SAGE	Yes	**0.5565**(0.0024)	**0.7261**(0.0115)	**0.7188**(0.0168)
GT	No	0.5508(0.0006)	0.7140(0.0201)	0.6959(0.0347)

Ablation study (Table 5). We evaluate the contribution of Edge-W, the signed correlation-based edge weighting, and Node-W, the CV-based node reliability weighting, while keeping the backbone fixed to weighted GraphSAGE. Across five repeated runs, the full DPR-GM achieves the best Prac-F1, AUROC, and AUPRC, and each ablated variant trails this configuration on every metric we report. Removing Edge-W produces a consistent drop across all three metrics, indicating that signed correlation modulation contributes meaningfully to ranking quality even when the domain topology is preserved. Removing Node-W yields a similarly consistent degradation across all three metrics, confirming that CV-based reliability weighting provides information complementary to the graph structure alone. Removing both components produces the weakest ranking performance, with the lowest AUROC and AUPRC in the ablation, showing that the domain-prior topology by itself is not sufficient for the strongest ranking quality. Overall, both Edge-W and Node-W consistently contribute to DPR-GM, and their combination produces the most reliable balance of thresholded detection quality and threshold-free ranking performance.

Table 5. Ablation study on SKAB `overall` over five repeated runs under the strict point-wise protocol. Mean (std). **Bold** / underline: best / second-best. Edge-W denotes signed correlation-based edge weighting, and Node-W denotes CV-based node reliability weighting.

Variant	Prac-F1	AUROC	AUPRC
w/o Edge-W & Node-W	0.5547(0.0040)	0.7091(0.0169)	0.6899(0.0292)
w/o Edge-W	0.5518(0.0031)	0.7209(0.0110)	0.7113(0.0159)
w/o Node-W	0.5539(0.0023)	0.7175(0.0152)	0.7028(0.0270)
DPR-GM	**0.5565**(0.0024)	**0.7261**(0.0115)	**0.7188**(0.0168)

5. Conclusion

We presented DPR-GM, a forecasting-based anomaly detection framework that constructs sensor graphs from domain knowledge rather than fully data-driven graph learning. DPR-GM uses a binary domain adjacency matrix extracted via an LLM from system design documentation to gate edge existence, Pearson correlation to modulate permitted edge strengths, and CV-based node reliability weights to reflect sensor stability in anomaly scoring. All graph and reliability priors are fixed before model training and introduce no additional learnable parameters, making the framework well-suited for data-scarce CPS where overfitting to limited normal-operation statistics is a practical concern. Experiments on the SKAB benchmark under a strict point-wise protocol show that DPR-GM outperforms graph-based, statistical, and deep learning baselines across F1, AUROC, AUPRC, MCC, and BF-F1, demonstrating that explicit domain priors provide a robust and stable alternative to learned graph topology in data-scarce CPS.

6. Limitations and Future Work

DPR-GM is limited by the granularity of available domain knowledge. In SKAB, the documentation provides subsystem membership and signal-flow direction, but not precise causal strengths between sensor pairs. Thus, DPR-GM captures coarse structural constraints rather than fine-grained physical causality. We also validate DPR-GM on a single CPS benchmark using a single LLM, Claude Sonnet 4.6 (Anthropic, 2026). Future work will improve LLM-based prior extraction through cross-LLM validation, prompt robustness analysis, confidence-aware edge extraction, and evaluation across more diverse CPS datasets.

Beyond CPS anomaly detection, many scientific discovery domains generate sensor-based spatio-temporal time-series data together with system documentation, experimental protocols, or domain descriptions. DPR-GM suggests a broader direction in which LLMs transform such documentation into explicit structural priors for graph-based learning. Such priors may support downstream tasks such as forecasting, anomaly detection, imputation, and sensor-state estimation in domains including climate, neuroscience, transportation, maritime systems, and autonomous laboratories. This points toward documentation-grounded AI systems that incorporate symbolic or causal priors into scientific time-series modeling, rather than relying only on correlations learned from limited observations.

References

Ahmed, C. M., Palleti, V. R., and Mathur, A. P. Wadi: a water distribution testbed for research in the design of secure cyber physical systems. In *Proceedings of the 3rd international workshop on cyber-physical systems for smart water networks*, pp. 25–28, 2017.

Anthropic. Claude sonnet 4.6 system card. Technical report, Anthropic, February 2026. URL https://anthropic.com/claude-sonnet-4-6-system-card. Model release date: February 17, 2026.

Audibert, J., Michiardi, P., Guyard, F., Marti, S., and Zuluaga, M. A. USAD: Unsupervised anomaly detection on multivariate time series. In *Proceedings of the ACM*

SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 3395–3404, 2020.

Box, G. E. P., Jenkins, G. M., Reinsel, G. C., and Ljung, G. M. *Time Series Analysis: Forecasting and Control*. Wiley, 5 edition, 2015.

Breunig, M. M., Kriegel, H.-P., Ng, R. T., and Sander, J. LOF: Identifying density-based local outliers. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pp. 93–104, 2000.

Chen, W., Tian, L., Chen, B., Dai, L., Duan, Z., and Zhou, M. Deep variational graph convolutional recurrent network for multivariate time series anomaly detection. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*, pp. 3621–3633, 2022.

Dai, E. and Chen, J. Graph-augmented normalizing flows for anomaly detection of multiple time series. *arXiv preprint arXiv:2202.07857*, 2022a.

Dai, E. and Chen, J. Graph-augmented normalizing flows for anomaly detection of multiple time series. In *International Conference on Learning Representations (ICLR)*, 2022b.

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

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

Hamilton, W., Ying, Z., and Leskovec, J. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017.

Hochreiter, S. and Schmidhuber, J. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

Hotelling, H. Multivariate quality control, illustrated by the air testing of sample bombsights. In Eisenhart, C., Hastay, M. W., and Wallis, W. A. (eds.), *Techniques of Statistical Analysis*, pp. 111–184. McGraw-Hill, New York, 1947.

Hundman, K., Constantinou, V., Laporte, C., Colwell, I., and Soderstrom, T. Detecting spacecraft anomalies using LSTMs and nonparametric dynamic thresholding. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 387–395, 2018.

Jin, M., Koh, H. Y., Wen, Q., Zambon, D., Alippi, C., Webb, G. I., King, I., and Pan, S. A survey on graph neural networks for time series: Forecasting, classification, imputation, and anomaly detection. *arXiv preprint arXiv:2307.03759*, 2024.

Katser, I. D. and Kozitsin, V. O. Skoltech anomaly benchmark (skab). https://www.kaggle.com/dsv/1693952, 2020.

Kipf, T. N. and Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

Liu, F. T., Ting, K. M., and Zhou, Z.-H. Isolation forest. In *Proceedings of the IEEE International Conference on Data Mining*, pp. 413–422, 2008.

Liu, Z., Gao, M., and Jiao, P. Gcad: Anomaly detection in multivariate time series from the perspective of granger causality. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 19041–19049, 2025.

Mathur, A. P. and Tippenhauer, N. O. Swat: A water treatment testbed for research and training on ics security. In *2016 international workshop on cyber-physical systems for smart water networks (CySWater)*, pp. 31–36. IEEE, 2016.

Park, D., Hoshi, Y., and Kemp, C. C. A multimodal anomaly detector for robot-assisted feeding using an LSTM-based variational autoencoder. *IEEE Robotics and Automation Letters*, 3(3):1544–1551, 2018.

Ruff, L., Vandermeulen, R., Goernitz, N., Deecke, L., Siddiqui, S. A., Binder, A., Müller, E., and Kloft, M. Deep one-class classification. In *International conference on machine learning*, pp. 4393–4402. PMLR, 2018.

Schölkopf, B., Platt, J. C., Shawe-Taylor, J., Smola, A. J., and Williamson, R. C. Estimating the support of a high-dimensional distribution. *Neural Computation*, 13(7): 1443–1471, 2001.

Shen, L., Li, Z., and Kwok, J. Timeseries anomaly detection using temporal hierarchical one-class network. In *Advances in Neural Information Processing Systems*, volume 33, pp. 13016–13026, 2020.

Shi, Y., Huang, Z., Feng, S., Zhong, H., Wang, W., and Sun, Y. Masked label prediction: Unified message passing model for semi-supervised classification. *arXiv preprint arXiv:2009.03509*, 2020.

Shyu, M.-L., Chen, S.-C., Sarinnapakorn, K., and Chang, L. A novel anomaly detection scheme based on principal component classifier. In *Proceedings of the IEEE Foundations and New Directions of Data Mining Workshop*, pp. 172–179, 2003.

Su, Y., Zhao, Y., Niu, C., Liu, R., Sun, W., and Pei, D. Robust anomaly detection for multivariate time series through stochastic recurrent neural network. In *Proceedings of the 25th ACM SIGKDD international conference*

on knowledge discovery & data mining, pp. 2828–2837, 2019a.

Su, Y., Zhao, Y., Niu, C., Liu, R., Sun, W., and Pei, D. Robust anomaly detection for multivariate time series through stochastic recurrent neural network. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 2828–2837, 2019b.

Tax, D. M. J. and Duin, R. P. W. Support vector data description. *Machine Learning*, 54(1):45–66, 2004.

Tuli, S., Casale, G., and Jennings, N. R. TranAD: Deep transformer networks for anomaly detection in multivariate time series data. *Proceedings of the VLDB Endowment*, 15(6):1201–1214, 2022.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., and Bengio, Y. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

Wu, H., Hu, T., Liu, Y., Zhou, H., Wang, J., and Long, M. TimesNet: Temporal 2D-variation modeling for general time series analysis. In *International Conference on Learning Representations*, 2023.

Wu, R. and Keogh, E. J. Current time series anomaly detection benchmarks are flawed and are creating the illusion of progress. *IEEE transactions on knowledge and data engineering*, 35(3):2421–2429, 2021.

Wu, X., Qiu, X., Li, Z., Wang, Y., Hu, J., Guo, C., and Yang, B. CATCH: Channel-aware multivariate time series anomaly detection via frequency patching. *arXiv preprint arXiv:2410.12261*, 2024.

Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., and Yu, P. S. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 32(1):4–24, 2020.

Xu, J., Wu, H., Wang, J., and Long, M. Anomaly transformer: Time series anomaly detection with association discrepancy. In *International Conference on Learning Representations*, 2022.

Zhang, C., Song, D., Chen, Y., Feng, X., Lumezanu, C., Cheng, W., Ni, J., Zong, B., Chen, H., and Chawla, N. V. A deep neural network for unsupervised anomaly detection and diagnosis in multivariate time series data. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pp. 1409–1416, 2019.

Zhang, W., Zhang, C., and Tsung, F. Grelen: Multivariate time series anomaly detection from the perspective of graph relational learning. In *IJCAI*, pp. 2390–2397, 2022.

Zhao, H., Wang, Y., Duan, J., Huang, C., Cao, D., Tong, Y., Xu, B., Bai, J., Tong, J., and Zhang, Q. Multivariate time-series anomaly detection via graph attention network. In *2020 IEEE international conference on data mining (ICDM)*, pp. 841–850. IEEE, 2020a.

Zhao, H., Wang, Y., Duan, J., Huang, C., Cao, D., Tong, Y., Xu, B., Bai, J., Tong, J., and Zhang, Q. Multivariate time-series anomaly detection via graph attention network. In *Proceedings of the IEEE International Conference on Data Mining*, pp. 841–850, 2020b.

A. LLM Prompt for Domain Prior Extraction

The following prompt is used to extract directed sensor dependencies from system design documentation using Claude Sonnet 4.6 (Anthropic, 2026). The output is parsed into a structured list of directed sensor pairs $(i \to j)$, which is used to construct the binary domain adjacency matrix $\mathbf{D}$.

System Prompt for Domain Prior Extraction ($N = 8$)

You are a physical systems expert specializing in sensor-based monitoring of industrial equipment. Given a natural language description of a physical sensor system, your task is to identify all directed physical dependencies between sensor pairs based on the underlying physical processes and system design.

Definition. A directed dependency from sensor A to sensor B, written as A $\to$ B, means that a change in the physical quantity measured by A causally and predictably influences the physical quantity measured by B through a known physical mechanism, such as electrical $\to$ mechanical, mechanical $\to$ hydraulic, hydraulic $\to$ thermal, or mechanical $\to$ mechanical propagation. The dependency is asymmetric: A $\to$ B does not imply B $\to$ A.

System description:

{SYSTEM_DESCRIPTION}

At each extraction step, you must:

- Identify all sensors in the description.
- For each sensor, record its name, the physical quantity it measures, and the subsystem it belongs to: electrical, mechanical, hydraulic, or thermal.
- Determine whether each ordered sensor pair has a directed physical dependency based on the system design.
- Include only dependencies that are physically motivated by the system design.
- Do not include sensor pairs that merely co-move statistically without a physical causal pathway.

Physical pathways to consider:

- *Electrical* $\to$ *Mechanical*: voltage or current driving motor torque, rotation, vibration, or speed.
- *Mechanical* $\to$ *Hydraulic*: shaft rotation, pump impeller motion, or mechanical actuation driving flow rate or pressure.
- *Hydraulic* $\to$ *Thermal*: fluid flow, pressure, or circulation affecting temperature or heat dissipation.
- *Mechanical* $\to$ *Mechanical*: vibration, imbalance, or shaft displacement propagating through connected components.

For each directed dependency, provide:

- The source sensor.
- The physical quantity measured by the source sensor.
- The target sensor.
- The physical quantity measured by the target sensor.
- A one-sentence explanation of the physical mechanism.

Output format. Return the complete result as a JSON list. Do not include any preamble, explanation, or markdown formatting outside the JSON.

```
[
  {
    "source": "<sensor name>",
    "source_quantity": "<physical quantity>",
    "target": "<sensor name>",
    "target_quantity": "<physical quantity>",
    "mechanism": "<one-sentence physical explanation>"
  },
  ...
]
```

B. Computational Environment

All experiments were conducted on an Ubuntu 24.04.4 LTS server equipped with two AMD EPYC 7502 32-core CPUs, 2.0 TiB RAM, and four NVIDIA GeForce RTX 3090 GPUs with 24 GB memory each. The server used NVIDIA driver 580.126.09 and CUDA 13.0.