DIFFTSAD: DIFFUSION-GUIDED DYNAMIC GRAPH LEARNING FOR UNSUPERVISED MULTIVARIATE TIME-SERIES ANOMALY DETECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Multivariate time-series anomaly detection (MTSAD) is critical for monitoring complex systems, while it suffers from three key limitations: the anomaly reconstruction paradox under severe class imbalance, epoch-dependent optimization conflicts where evolving reconstruction fidelity inverts supervision signals during co-training, and rigid static graph topologies that fail to capture anomaly-induced structural shifts. We propose **DiffTSAD**, a framework that fundamentally decouples generative anomaly simulation from discriminative learning. By leveraging a frozen pre-trained diffusion model as a stationary perturbation prior, DiffTSAD injects severity-calibrated perturbations to establish a *distribution-invariant* binary classification paradigm, eliminating the non-stationary supervision of *epoch-dependent pseudo-labeling* methods. Diffusion-induced residuals further guide dynamic graph construction to adaptively reweight inter-variable dependencies, while a topology-aware Graph Attention Network with chunk-wise dilated temporal convolutions enables fine-grained spatio-temporal representation learning. Experiments on three real-world benchmarks demonstrate that DiffTSAD achieves state-of-the-art or competitive performance.

1 INTRODUCTION

Multivariate Time-Series Anomaly Detection (MTSAD) has emerged as a critical task for monitoring complex systems, including industrial infrastructure Ismail Fawaz et al. (2019), financial risk management Wen et al. (2022), and cybersecurity Luo et al. (2021). Recent deep learning advancements have notably enhanced spatio-temporal dependency modeling, driving progress across diverse benchmarks Pang et al. (2021); Wang et al. (2024). The field has transitioned from statistical paradigms to deep learning–based approaches, giving rise to three dominant paradigms: prediction-based Deng & Hooi (2021); Liu et al. (2023), representation-based Xu et al. (2022), and reconstruction-based Zeng et al. (2026). Among these, reconstruction-based methods have been widely adopted for unsupervised MTSAD, initially employing autoencoders Tuli et al. (2022); Su et al. (2019) and more recently integrating GNNs and Transformers to jointly model spatial correlations and temporal dynamics Zhao et al. (2020); Deng & Hooi (2021); Cai et al. (2026).

Despite recent progress, reconstruction-based MTSAD faces three limitations. **First**, the inherent scarcity of anomalous samples triggers the *anomaly reconstruction paradox*: models inadvertently reconstruct abnormal patterns with high fidelity, blurring the discriminative boundary Zeng et al. (2026); Yang et al. (2023b). **Second**, prevailing architectures rely on static or heuristic topologies Deng & Hooi (2021); Zhao et al. (2020) that fail to adapt to anomaly-induced structural shifts. Recent dynamic graph attempts derive adjacency matrices from scratch-trained residuals, which exhibit high variance across epochs and introduce structural noise Zeng et al. (2026). **Third**, conventional

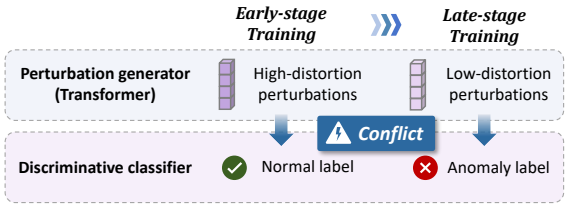


Figure 1: The epoch-dependent optimization conflict under co-training: early-stage (left) vs. late-stage (right) training.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

co-training paradigms, which we term *epoch-dependent pseudo-labeling*, suffer from a structural optimization conflict: the evolving reconstruction fidelity induces non-stationary supervision, where early-stage high-distortion perturbations are labeled normal while late-stage subtle deviations induce severe supervision ambiguity due to overlap with normal samples, destabilizing classifier convergence Cai et al. (2026). This non-stationary semantic drift is illustrated in Figure 1.

To this end, we propose **DiffTSAD**[1], a novel framework that establishes a *distribution-invariant discriminative learning* paradigm by fundamentally decoupling generative anomaly simulation from discriminative representation learning. Unlike conventional pipelines, DiffTSAD employs a *frozen, pre-trained diffusion backbone* as a frozen generative reference to inject severity-calibrated perturbations. This decoupled design ensures the perturbation distribution remains strictly *epoch-invariant*, converting the detection task into a well-posed stationary binary classification problem and entirely circumventing the optimization conflicts inherent in the *epoch-dependent pseudo-labeling* paradigm. These perturbations dynamically guide graph construction to capture anomaly-distorted spatial correlations, while a unified spatio-temporal network learns robust discriminative boundaries. Crucially, freezing the generator is not a mere module substitution but a structural necessity that alters the optimization landscape. The main contributions are summarized as follows:

- We formalize a **distribution-invariant discriminative learning** paradigm for unsupervised MTSAD, in which a frozen pre-trained diffusion model serves as a stationary perturbation operator. By decoupling generative simulation from discriminative learning, this design guarantees epoch-invariant supervision signals, fundamentally resolving the label inversion and optimization conflicts inherent in co-trained reconstruction methods.
- We propose a residual-driven dynamic graph strategy that adaptively modulates inter-variable dependencies based on perturbation magnitudes, capturing anomaly-induced structural shifts and overcoming the rigidity of static topologies under non-stationary distribution changes.
- We design a topology-aware spatio-temporal encoder that integrates learnable structural node embeddings into a graph attention mechanism, coupled with chunk-wise dilated temporal convolutions, enabling fine-grained representation learning that jointly captures local fluctuations and long-range temporal dependencies.

2 Related Work

MTSAD Paradigms: Traditional methods (ARIMA Box et al. (2015), isolation forests Liu et al. (2008), OCSVM Cortes & Vapnik (1995)) rely on stationarity assumptions and handcrafted features, limiting applicability to high-dimensional dynamics. Deep learning has shifted toward automatic representation learning: forecasting-based methods flag anomalies via prediction residuals Deng & Hooi (2021); Tuli et al. (2022), contrastive methods learn discriminative latent spaces Yang et al. (2023b); Darban et al. (2025), and reconstruction-based methods identify anomalies through reconstruction discrepancies Audibert et al. (2020). Despite their success, these approaches often rely on rigid topologies and suffer from the anomaly reconstruction paradox.

Spatio-Temporal Modeling: Sequence-centric models (Anomaly Transformer Xu et al. (2022), TranAD Tuli et al. (2022)) capture temporal dynamics but overlook inter-variable interactions. Early graph-based efforts (OmniAnomaly Su et al. (2019), MTAD-GAT Zhao et al. (2020)) and recent frameworks (GDN Deng & Hooi (2021), GRELEN Zhang et al. (2022)) explicitly model sensor correlations but rely on rigid or heuristic static topologies. SPAGD Cai et al. (2026) introduces dynamic graphs yet remains constrained by *epoch-dependent pseudo-labeling* conflicts.

Diffusion for Time Series: VAEs Kingma & Welling (2013) and GANs Goodfellow et al. (2020) suffer from posterior/mode collapse. DDPMs Ho et al. (2020) offer stable training and high-fidelity generation, excelling in imputation Gao et al. (2024) and forecasting Ye et al. (2025). Existing diffusion-based detectors Chen et al. (2023); Xiao et al. (2023); Yang et al. (2023a) primarily rely on reconstruction or imputation residuals for anomaly scoring, leaving them susceptible to the anomaly reconstruction paradox. Crucially, the diffusion forward process provides a principled mechanism for controlled noise injection with severity governed by a predefined schedule—a property underutilized in existing pipelines. DiffTSAD leverages this as an epoch-invariant perturbation operator, fundamentally distinguishing it from residual-based approaches.

[1]Code can be found at: https://anonymous.4open.science/r/DiffTSAD-6E75.

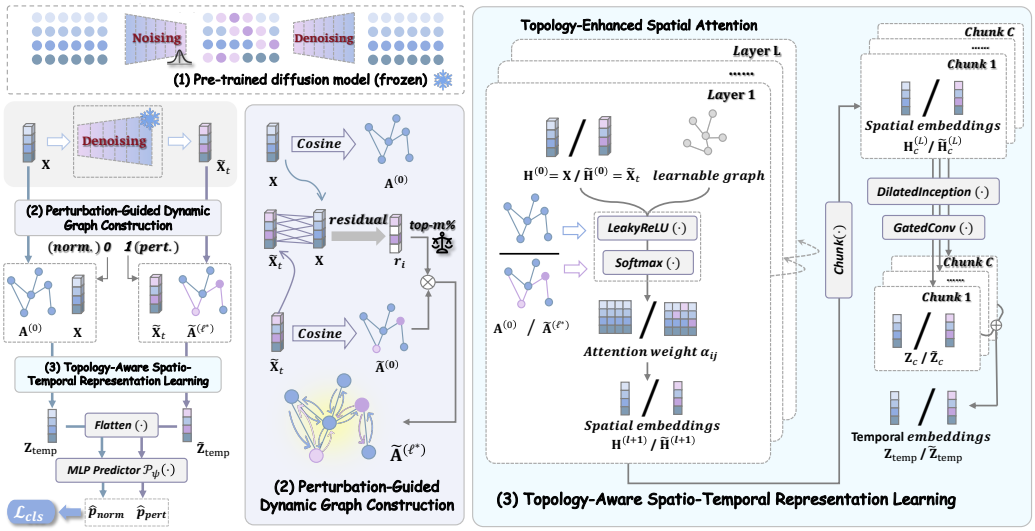


Figure 2: Overall architecture of the proposed DiffTSAD framework.

3 METHODOLOGY

We consider a multivariate time series segment $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T]^\top \in \mathbb{R}^{T \times d}$, where each $\mathbf{x}_t \in \mathbb{R}^d$ denotes observations of d variables at timestamp t. Since the training set $\mathcal{D}_{\text{train}}$ contains only normal sequences, the task is to learn a scoring function $S_\Theta : \mathbb{R}^{T \times d} \to [0, 1]$ that quantifies deviation from the normal manifold. To resolve the *anomaly reconstruction paradox* under this unsupervised setting, DiffTSAD transforms the task into a stable proxy classification problem: a frozen diffusion model injects severity-calibrated perturbations $\tilde{\mathbf{X}}$ into clean sequences $\mathbf{X}$, and the network is optimized to distinguish originals ($y = 0$) from perturbed counterparts ($y = 1$). This design provides epoch-invariant supervision signals and enables robust generalization to unseen anomalies.

Figure 2 illustrates the architecture of DiffTSAD, which orchestrates unsupervised MTSAD through a decoupled three-stage pipeline that fundamentally separates generative anomaly simulation from discriminative representation learning, thereby eliminating the epoch-dependent optimization conflicts prevalent in scratch-trained reconstruction methods. (1) In the first stage, a *frozen pre-trained diffusion model* injects severity-calibrated noise into normal sequences, generating controlled perturbations without altering the underlying normal manifold. (2) The second stage introduces a *perturbation-guided dynamic graph construction* mechanism, which computes node-wise perturbation deviation scores between original and perturbed inputs and adaptively modulates inter-variable connectivity to generate instance-specific adjacency matrices: $\mathbf{A}^{(0)}$ for the normal stream and $\tilde{\mathbf{A}}^{(\ell^*)}$ for the perturbed stream. (3) The third stage employs a *topology-aware spatio-temporal representation learning* that processes both streams in parallel via shared parameters. It integrates a topology-enhanced spatial attention mechanism, which fuses learnable node structural embeddings with feature-based attention, followed by chunk-wise temporal modeling via dilated inception and gated convolutions. The resulting spatio-temporal embeddings are flattened and fed into a shared MLP predictor to output discriminative probabilities $\hat{p}_{\text{norm}}$ and $\hat{p}_{\text{pert}}$. The entire framework is optimized via a binary classification loss $\mathcal{L}_{\text{cls}}$, ensuring that the generative backbone remains epoch-invariant while the classifier learns a decision boundary between normal dynamics and anomalies.

3.1 CONTROLLED ANOMALY SIMULATION VIA PRE-TRAINED DIFFUSION

Conventional reconstruction-based paradigms (Cai et al., 2026) employ a scratch-trained autoencoder to generate pseudo-anomalous samples via reconstruction residuals. However, this co-training paradigm suffers from a critical optimization conflict: during early training stages, the reconstructor exhibits high randomness and poor fidelity, generating perturbations with elevated anomaly degrees that are paradoxically labeled as normal; conversely, in later stages, refined reconstructions yield subtle deviations that overlap heavily with normal samples, inducing severe supervision ambiguity and destabilizing classifier convergence. To circumvent this *epoch-dependent pseudo-labeling*

conflict, we adopt a **pre-trained diffusion model** (Ho et al., 2020) as a *frozen, epoch-invariant perturbation operator*, whose continuous noise schedule provides a principled severity spectrum unavailable to deterministic encoders.

Concretely, the diffusion backbone is first pre-trained on $\mathcal{D}_{\text{train}}$ to minimize the noise prediction objective $\mathcal{L}_{\text{diff}} = \mathbb{E}_{t,\mathbf{X},\boldsymbol{\epsilon}}[\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\tilde{\mathbf{X}}_t, t)\|^2]$, then *frozen* ($\nabla_\theta \mathcal{L} = \mathbf{0}$) throughout the detection phase. Given a normal window $\mathbf{X}$, we obtain its perturbed counterpart $\tilde{\mathbf{X}}_\ell$ by mapping a diffusion level $\ell \in [0, 1]$ to a discrete timestep $t = \lfloor \ell \cdot T_{\text{diff}} \rfloor$ and applying the standard forward transition:

$$\tilde{\mathbf{X}}_\ell = \sqrt{\bar{\alpha}_t}\,\mathbf{X} + \sqrt{1 - \bar{\alpha}_t}\,\boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{1}$$

where $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$ is the cumulative noise schedule. When $\ell \to 0$, the sample remains virtually unchanged; when $\ell \to 1$, it approaches pure noise. We discretize the trajectory into $\mathcal{L} = \{0.0, 0.1, \ldots, 1.0\}$ and select the optimal level ℓ^* per dataset in Section 4.4. Since the backbone is frozen, the perturbation distribution $\mathcal{P}(\tilde{\mathbf{X}}_\ell \mid \mathbf{X})$ remains strictly epoch-invariant, transforming the task into a stationary proxy classification problem. This pre-training constitutes a one-time, dataset-level investment reusable across all downstream configurations. We formally show in Appendix A.1 that freezing the perturbation generator renders the proxy distribution epoch-invariant, so the standard i.i.d. VC generalization bound applies (Proposition 1); in contrast, co-trained perturbations induce a non-stationary labeled distribution for which this argument breaks down (Remark 1).

3.2 Perturbation-Guided Dynamic Graph Construction

Unlike static graph frameworks (Deng & Hooi, 2021; Zhang et al., 2022) that maintain fixed topologies, our mechanism adaptively constructs distinct graph structures for normal and perturbed samples based on diffusion-induced residuals, reflecting the varying dependency patterns under different anomaly severities.

Residual-Based Node Scoring: Given $\mathbf{X} \in \mathbb{R}^{T \times d}$ and its perturbed counterpart $\tilde{\mathbf{X}}_{\ell^*}$, we compute the node-specific residual score for the i-th variable as:

$$r_i = \frac{1}{T} \sum_{t=1}^{T} \left| X_{t,i} - \tilde{X}_{\ell^*,t,i} \right|, \tag{2}$$

which aggregates the absolute deviation across the temporal dimension. Higher r_i indicates stronger perturbation effects. We normalize via $\hat{r}_i = \sigma(r_i) \in [0, 1]$.

Adaptive Adjacency Matrix Construction: For normal samples ($\ell = 0$), the base adjacency matrix $\mathbf{A}^{(0)} \in \mathbb{R}^{d \times d}$ is constructed via Z-scored dot-product similarity:

$$\mathbf{Z} = \mathbf{X}^\top \mathbf{X}, \quad \mathbf{A}^{(0)} = \sigma\left(\frac{\mathbf{Z} - \mu(\mathbf{Z})}{\text{std}(\mathbf{Z}) + \epsilon_0} \right), \tag{3}$$

which preserves coupling magnitude while smoothly suppressing spurious weak correlations through sigmoid saturation.

For perturbed samples ($\ell^* > 0$), we first compute $\tilde{\mathbf{A}}^{(0)}$ from $\tilde{\mathbf{X}}_{\ell^*}$ via Eq. 3, then identify the top-$m\%$ highest-residual variables as the anomalous candidate set $\mathcal{M}$, and inject their residual scores into the baseline similarity:

$$\tilde{\mathbf{A}}_{ij}^{(\ell^*)} = \tilde{\mathbf{A}}_{ij}^{(0)} + \mathbb{I}(i \in \mathcal{M})\,\hat{r}_i + \mathbb{I}(j \in \mathcal{M})\,\hat{r}_j - \delta_{ij}\,\mathbb{I}(i \in \mathcal{M})\,\hat{r}_i, \tag{4}$$

where $\mathbb{I}(\cdot)$ is the indicator function and δ_{ij} the Kronecker delta preventing diagonal double-counting. By Weyl's inequality, this injection induces a spectrally bounded perturbation: $\max_i |\lambda_i(\tilde{\mathbf{A}}_{\text{pre}}^{(\ell^*)}) - \lambda_i(\tilde{\mathbf{A}}^{(0)})| \leq 2\sqrt{dm} + 1$, where $m = |\mathcal{M}|$ (see Proposition 2, Appendix A.2). This row-column injection explicitly strengthens connections involving perturbation-sensitive variables, enabling the graph topology to dynamically reflect anomaly-induced dependency shifts. A Top-k masking operator is further applied to retain only the k strongest edges per node, inducing sparsity for computational efficiency.

Binary Label Assignment: The original sequence $\mathbf{X}$ is labeled $y = 0$ (normal), while its perturbed counterpart $\tilde{\mathbf{X}}_{\ell^*}$ receives $y = 1$ (perturbed). This establishes a dual-stream proxy classification paradigm: $(\mathbf{X}, \mathbf{A}^{(0)}) \to y = 0$ and $(\tilde{\mathbf{X}}_{\ell^*}, \tilde{\mathbf{A}}^{(\ell^*)}) \to y = 1$, providing stable, epoch-invariant structural supervision for the downstream spatio-temporal encoder.

3.3 TOPOLOGY-AWARE SPATIO-TEMPORAL REPRESENTATION LEARNING

Building upon the instance-adaptive topologies $\mathbf{A}^{(0)}$ and $\tilde{\mathbf{A}}^{(\ell^*)}$ from Section 3.2, we design a unified dual-stream architecture with shared parameters to extract discriminative spatio-temporal features from normal and perturbed sequences in parallel.

Topology-Enhanced Spatial Attention: Standard GATs compute attention solely from feature similarity, overlooking structural priors in sensor networks. We therefore employ a **TopoGraphLayer** that integrates learnable node structural embeddings into the attention mechanism. For the l-th layer at a generic timestamp t (omitted for brevity), let $\mathbf{h}_i^{(l)} \in \mathbb{R}^{d_h}$ denote the feature of node i and $\mathbf{e}_i \in \mathbb{R}^{d_e}$ its learnable graph embedding. The computation proceeds in three steps:

$$\mathbf{u}_i = \mathbf{W}^{(l)}\mathbf{h}_i^{(l)}, \tag{5}$$

$$\gamma_i = \mathbf{w}_f^\top \mathbf{u}_i + \mathbf{v}_s^\top \mathbf{e}_i, \tag{6}$$

$$s_{ij} = \text{LeakyReLU}(\gamma_i + \gamma_j), \tag{7}$$

where $\mathbf{W}^{(l)} \in \mathbb{R}^{d_h \times d_h}$ is a shared projection matrix, and $\mathbf{w}_f, \mathbf{v}_s$ are learnable attention vectors that dynamically weight feature semantics and topological position, respectively. The normalized attention weight is obtained via softmax over the neighborhood $\mathcal{N}(i)$ induced by the stream-specific adjacency matrix:

$$\alpha_{ij} = \frac{\exp(s_{ij})}{\sum_{k\in\mathcal{N}(i)} \exp(s_{ik})}, \tag{8}$$

and node representations are updated through weighted message passing:

$$\mathbf{h}_i^{(l+1)} = \phi\left(\sum_{j\in\mathcal{N}(i)} \alpha_{ij}\, \mathbf{W}^{(l)}\mathbf{h}_j^{(l)}\right), \tag{9}$$

where $\phi(\cdot)$ is the activation function. All attention parameters ($\mathbf{W}^{(l)}, \mathbf{w}_f, \mathbf{v}_s$) are shared across timestamps and streams; distinct graph topologies alone guide stream-specific dependency modeling. After L layers, stacking all timestamps yields spatial embeddings $\mathbf{H}^{(L)}, \tilde{\mathbf{H}}^{(L)} \in \mathbb{R}^{T\times d\times D_s}$.

Chunk-wise Temporal Modeling: We partition the temporal axis into C non-overlapping chunks, each processed by gated dilated inception modules with exponentially increasing dilation factors:

$$\mathbf{Z}_c = \text{GatedConv}\left(\text{DilatedInception}(\mathbf{H}_c^{(L)}; \Theta_{\text{temp}})\right), \tag{10}$$

where DilatedInception$(\cdot)$ applies parallel 1D convolutions at multiple dilation rates to extract multi-scale patterns, and GatedConv$(\cdot)$ implements a multiplicative gating $\mathbf{Z}_c = \tanh(\mathbf{F}_c) \odot \sigma(\mathbf{G}_c)$ that adaptively modulates information flow. Chunk-wise outputs are concatenated along the feature dimension: $\mathbf{Z}_{\text{temp}} = \text{Concat}(\mathbf{Z}_1, \ldots, \mathbf{Z}_C) \in \mathbb{R}^{d\times D_T}$. The temporal parameters Θ_{temp} are shared across both streams, yielding final representations $\mathbf{Z}_{\text{temp}}, \tilde{\mathbf{Z}}_{\text{temp}} \in \mathbb{R}^{d\times D_T}$.

Discriminative Anomaly Prediction: The spatio-temporal embeddings are flattened and projected through a shared MLP $\mathcal{P}_\psi(\cdot)$:

$$\hat{p}_{\text{norm}} = \sigma\left(\mathcal{P}_\psi(\text{Flatten}(\mathbf{Z}_{\text{temp}}))\right), \quad \hat{p}_{\text{pert}} = \sigma\left(\mathcal{P}_\psi(\text{Flatten}(\tilde{\mathbf{Z}}_{\text{temp}}))\right), \tag{11}$$

where $\sigma(\cdot)$ is the sigmoid activation. The dual-stream binary supervision ($y = 0$ for normal, $y = 1$ for perturbed) forces the shared encoder to learn a sharp decision boundary separating normal dynamics from severity-calibrated deviations.

3.4 OPTIMIZATION OBJECTIVE

Training: Since the diffusion backbone is frozen ($\nabla_{\Theta_{\text{diff}}}\mathcal{L} = \mathbf{0}$) during the detection stage, the optimization objective solely focuses on the perturbation-guided graph constructor and the spatio-temporal classifier. Given a batch of B normal time series $\{\mathbf{X}^{(n)}\}_{n=1}^B$ and their perturbed counterparts $\{\tilde{\mathbf{X}}_{\ell^*}^{(n)}\}_{n=1}^B$, we assign binary labels $y = 0$ (normal) and $y = 1$ (perturbed). The spatio-temporal module processes both streams in parallel, yielding prediction probabilities $\hat{p}_{\text{norm}}^{(n)}$ and $\hat{p}_{\text{pert}}^{(n)}$.

The optimization objective is a standard binary cross-entropy loss:

$$\mathcal{L}_{\text{cls}} = -\frac{1}{2B}\sum_{n=1}^{B}\left[\log\left(1-\hat{p}_{\text{norm}}^{(n)}\right)+\log\left(\hat{p}_{\text{pert}}^{(n)}\right)\right], \tag{12}$$

where $\hat{p}_{\text{norm}}^{(n)} = \sigma\left(\mathcal{P}_{\boldsymbol{\psi}}(\mathbf{v}^{(n)})\right)$ and $\hat{p}_{\text{pert}}^{(n)} = \sigma\left(\mathcal{P}_{\boldsymbol{\psi}}(\tilde{\mathbf{v}}^{(n)})\right)$ denote the predicted probabilities for the normal and perturbed streams of the n-th sample, respectively. By relying on epoch-invariant perturbations from the frozen backbone, the classifier converges efficiently without being misled by transient reconstruction artifacts.

Inference and Anomaly Scoring: During inference, for each test sequence $\mathbf{X}_{\text{test}}$, we construct both streams and their corresponding adjacency matrices $\mathbf{A}_{\text{test}}^{(0)}$ and $\tilde{\mathbf{A}}_{\text{test}}^{(\ell^*)}$ using the same procedure as in training. The classifier outputs a probability $\hat{p}_{\text{cls}} = \sigma(\mathcal{P}_{\psi}(\mathbf{v}_{\text{test}}))$ as the discriminative anomaly indicator. To complement this, we quantify generative reconstruction fidelity: the frozen diffusion backbone produces a reconstruction $\hat{\mathbf{X}}_{\text{recon}}$ (averaged over 3 single-step trials to mitigate variance), preserving discriminative residuals by strictly avoiding test-time adaptation. The aggregated reconstruction error is $\Delta = \frac{1}{T\cdot d}\sum_{t=1}^{T}\sum_{i=1}^{d}|\mathbf{X}_{\text{test},t,i} - \hat{\mathbf{X}}_{\text{recon},t,i}|$. Finally, we integrate both signals through a convex formulation:

$$S(\mathbf{X}_{\text{test}}) = \alpha\cdot\hat{p}_{\text{cls}} + (1-\alpha)\cdot\sigma\left(\frac{\Delta-\text{median}(\Delta)}{\text{IQR}(\Delta)+\epsilon}\right), \tag{13}$$

where the interquartile range $\text{IQR}(\Delta)$ replaces standard deviation for robust, distribution-aware normalization (Liu et al., 2024; Yin et al., 2026). Since reconstruction errors in real-world telemetry often exhibit heavy-tailed distributions, anchoring to the median and IQR prevents outlier-driven score compression and ensures well-calibrated probabilities under severe class imbalance. For completeness, the complete training and inference procedures of DiffTSAD are summarized in Algorithm 1 (Appendix B).

4 Evaluation

4.1 Datasets and Experimental Settings

Datasets: We evaluate DiffTSAD on three widely adopted real-world benchmarks: MSL and SMAP (Hundman et al., 2018), collected from spacecraft telemetry and environmental monitoring, and SWaT (Mathur & Tippenhauer, 2016), an industrial water treatment testbed. All raw series are Min-Max normalized to $[0, 1]$. Following the standard unsupervised MTSAD protocol, all models are trained exclusively on normal samples and evaluated on test sets containing a mixture of normal and anomalous instances. Detailed dataset statistics are summarized in Appendix C.1.

Baselines: We compare DiffTSAD against two categories of baselines: (1) classical machine learning methods: k-NN (Fix, 1985), OCSVM (Cortes & Vapnik, 1995), LOF (Breunig et al., 2000), IForest (Liu et al., 2008), and COPOD (Li et al., 2020); (2) deep learning methods: USAD (Audibert et al., 2020), NCAD (Carmona et al., 2021), Anomaly Transformer (AT) (Xu et al., 2022), IMDiffusion (Chen et al., 2023), DiffAD (Xiao et al., 2023), DDMT (Yang et al., 2023a), TimesNet (Wu et al., 2023), DCdetector (Yang et al., 2023b), and SPAGD (Cai et al., 2026), which serves as the most directly related co-training baseline. Brief descriptions of each method are provided in Appendix C.2.

Metrics: We report AUC, AUPRC, and F1-score. To ensure rigorous and unbiased evaluation, we deliberately exclude performance-inflating post-processing such as Point Adjustment (PA) (Kim et al., 2022; Liu & Paparrizos, 2024), computing all metrics directly on raw anomaly scores. Further details are provided in Appendix C.4.

4.2 Main Results and Analysis

Table 1 presents a comprehensive comparison between DiffTSAD and representative baselines across three benchmarks, where DiffTSAD consistently achieves state-of-the-art or highly competitive results under diverse operational conditions and severe class imbalance.

Table 1: Performance comparison of DiffTSAD and baseline models on three datasets. Metrics include AUC, AUPRC, and F1 (in %). The best and second-best results are highlighted in **bold** and underlined, respectively.

Dataset	Metric	DiffTSAD (our)	SPAGD	DCdetector	TimesNet	DDMT	DiffAD	IMDiffusion	AT	NCAD	USAD	COPOD	IForest	LOF	OCSVM	k-NN
MSL	AUC	**64.80**	53.63	50.10	55.06	54.03	51.84	61.02	52.19	59.40	61.67	60.56	56.91	56.37	59.33	59.32
	AUPRC	**20.60**	14.51	10.55	11.79	17.94	18.74	20.25	11.33	14.13	12.86	18.17	16.85	20.06	17.58	17.55
	F1	**33.27**	19.40	19.09	21.26	26.02	25.02	31.38	19.25	21.98	27.12	32.77	29.59	27.61	32.61	32.40
	Avg.	**39.56**	29.18	26.58	29.37	32.66	31.87	37.55	27.59	31.84	33.88	37.17	34.45	34.68	36.51	36.42
SMAP	AUC	**51.44**	44.48	49.79	39.90	48.63	40.36	40.46	40.03	39.67	39.51	40.04	39.78	42.94	39.87	39.73
	AUPRC	**15.14**	10.60	12.76	11.36	13.79	11.45	11.96	11.39	11.69	11.41	12.02	11.21	12.35	11.46	11.48
	F1	25.23	22.70	22.70	22.69	24.27	24.65	25.17	22.78	22.68	22.69	25.08	25.08	**25.40**	25.08	25.08
	Avg.	**30.60**	25.93	28.42	24.65	28.90	25.49	25.86	24.73	24.68	24.54	25.71	25.36	26.90	25.47	25.43
SWaT	AUC	**82.15**	64.28	49.68	32.45	31.89	21.71	23.90	79.77	19.00	79.67	81.66	78.74	73.33	76.06	77.30
	AUPRC	70.42	23.86	12.20	14.21	12.70	12.43	10.85	62.28	8.77	70.24	**71.57**	66.87	45.23	67.27	68.28
	F1	72.37	28.15	21.97	21.80	22.73	22.60	22.68	**73.03**	21.76	72.60	70.85	66.24	53.18	71.79	71.95
	Avg.	**74.98**	38.76	27.95	22.82	22.44	18.91	19.14	71.69	16.51	74.17	74.69	70.62	57.25	71.71	72.51

Table 2: Ablation study of DiffTSAD on the MSL and SMAP datasets. Metrics include AUC, AUPRC, and F1 (in %). We evaluate the impact of different modules by removing or replacing them. The best results are highlighted in **bold**.

Module	Variant	MSL			SMAP		
		AUC	**AUPRC**	**F1**	**AUC**	**AUPRC**	**F1**
Reconstruction Module	w Transformer	37.66	11.45	24.95	51.01	14.44	24.59
	w/o Reconstruction	51.39	15.85	24.81	50.35	13.61	22.16
Perturbation-Guided Dynamic Graph	w/o Gap Enhancement	64.46	20.40	32.85	51.23	14.79	24.96
	w/o Dynamic Graph	64.55	20.42	32.48	51.12	14.43	25.07
Spatio-Temporal Representation	w/o Temporal Graph	60.15	18.64	28.82	50.85	14.81	24.25
	w/o Spatial Graph	61.91	17.59	30.77	49.06	14.35	24.16
Topological Attention	w/o TopoGraphLayer	59.90	18.28	30.02	49.90	14.47	23.50
	w/o Embedding	61.27	19.67	29.71	49.26	14.50	24.67
	w GCN	63.41	19.74	30.87	51.29	15.09	25.16
Hybrid Anomaly Scoring	w/o Error Reconstruction	56.33	17.87	24.81	49.48	14.20	24.25
	w/o Classifier	53.98	14.50	19.49	49.43	14.18	24.16
	w Linear Combination	64.10	20.47	30.83	51.39	15.04	25.17
	w/o Smoothing	54.91	18.72	27.14	51.16	14.57	25.14
DiffTSAD (our)		**64.80**	**20.60**	**33.27**	**51.44**	**15.14**	**25.23**

Dataset-Specific Performance. On **MSL**, DiffTSAD secures the highest scores across all metrics, outperforming the second-best models by up to 5.08% in AUC and 1.73% in AUPRC, confirming its capacity to capture high-dimensional telemetry dependencies. For **SMAP**, it attains the top AUC and AUPRC; while LOF reports marginally higher F1 due to threshold sensitivity, our superior ranking metrics indicate more reliable probability calibration. On the industrial benchmark **SWaT**, DiffTSAD achieves the highest AUC and average score, with AUPRC and F1 trailing the best baselines by less than 1.6% and 0.9%, reflecting stability in distinguishing subtle attacks without overfitting.

Comparison with Baselines. DiffTSAD outperforms its best baselines by average margins of 5.35% (MSL), 5.88% (SMAP), and 0.39% (SWaT). This validates our core architectural shift: replacing a scratch-trained reconstructor with a frozen diffusion simulator eliminates epoch-dependent label inversion, while the topology-aware GAT captures finer structural priors than standard message-passing or pure self-attention paradigms (e.g., DCdetector, TimesNet). While baselines remain competitive under point-wise scoring, deep reconstruction models degrade from the reconstruction paradox; DiffTSAD bridges it via discriminative learning and generative residuals.

Robustness under Severe Class Imbalance. A critical advantage lies in strong AUPRC performance under extreme anomaly scarcity (10.72%–13.13%). Unlike methods relying on post-hoc score manipulation (e.g., Point Adjustment), our hybrid scoring fuses classifier confidence with diffusion-guided reconstruction discrepancies. This enables precise threshold-agnostic ranking, confirming that perturbation-induced residual graphs effectively highlight anomaly-sensitive interactions to maintain high precision without sacrificing recall.

4.3 Ablation Study

To validate the necessity of each core component, we conduct ablation studies on MSL and SMAP across five dimensions in Table 2. The full DiffTSAD consistently achieves optimal performance across all metrics, confirming the synergistic value of our design choices.

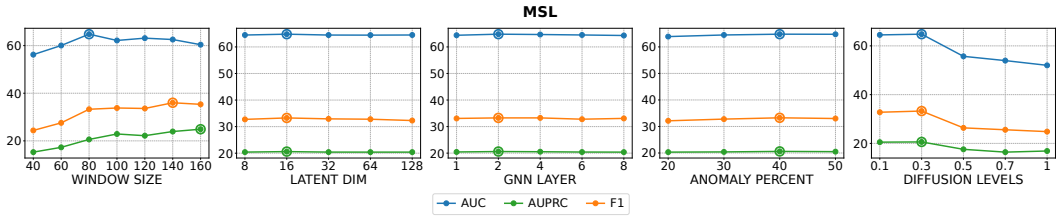


Figure 3: Sensitivity analysis of key hyperparameters on MSL across AUC, AUPRC, and F1 (in %).

Reconstruction Mechanism. `w Transformer` replaces the diffusion backbone with a scratch-trained Transformer autoencoder mimicking conventional reconstruction pipelines, while `w/o Reconstruction` removes the generative module entirely, feeding identical normal samples to both streams. The `w/o Reconstruction` variant significantly outperforms `w Transformer` , underscoring that scratch-trained reconstructors inevitably suffer from epoch-dependent optimization conflicts: as reconstruction fidelity evolves, the perturbation distribution becomes non-stationary, destabilizing the classifier's decision boundary. By contrast, our frozen diffusion simulator provides severity-calibrated, distribution-invariant perturbations, decoupling generative uncertainty from discriminative learning.

Perturbation-Guided Dynamic Graph. `w/o Dynamic Graph` replaces instance-adaptive topologies with a static KNN similarity matrix; its consistent performance drop confirms that static topologies fail to capture anomaly-induced dependency shifts. `w/o Gap Enhancement` disables residual-driven adjacency reweighting; its moderate degradation indicates that while baseline similarity provides a reasonable structural prior, residual-driven reweighting is essential for amplifying perturbation-sensitive correlations and focusing the classifier on structurally disrupted variables.

Spatio-Temporal Representation. Isolating the temporal pathway (`w/o Temporal Graph`) or the spatial message-passing pathway (`w/o Spatial Graph`) yields comparable declines, verifying that dual-stream fusion is non-redundant: temporal convolutions capture sustained anomalous trends and multi-scale fluctuations, while spatial GATs model cross-variable contagion effects, and their joint optimization distinguishes transient sensor noise from systemic fault propagation.

Topological Attention. `w/o TopoGraphLayer` reverts to standard GATConv without structural priors, and `w/o Embedding` retains GAT but removes node embedding injection; both exhibit substantial drops in AUPRC, demonstrating that standard attention overlooks critical structural priors in sensor networks. Injecting learnable node embeddings allows the model to weigh connections by feature similarity and topological importance. Additionally, `w GCN`, which substitutes attention with uniform Laplacian aggregation, underperforms our design, confirming that adaptive neighborhood weighting is superior to uniform smoothing for heterogeneous inter-variable dynamics.

Hybrid Anomaly Scoring. `w/o Error Reconstruction` (classifier confidence only) and `w/o Classifier` (reconstruction discrepancies only) both lag behind the full model, validating that fusing discriminative confidence with generative deviation metrics mitigates individual modality blind spots. `w Linear Combination` uses fixed fusion weights without dynamic calibration; the full model's marginal gain over it highlights the necessity of distribution-aware calibration. `w/o Smoothing` removes the IQR-normalized mapping and temporal smoothing; its degradation shows their importance in suppressing stochastic denoising variance and stabilizing threshold-agnostic ranking. Collectively, every component is indispensable for state-of-the-art performance.

4.4 Parameter Sensitivity Analysis

To evaluate the stability and practical deployability of DiffTSAD, we analyze five critical hyperparameters on MSL: input window size, latent dimension, GNN layer depth, anomaly candidate percentage, and diffusion level (Figure 3); the full study is deferred to Appendix D.1.

The model demonstrates remarkable robustness to architectural hyperparameters: varying `Latent Dim` (8–128), `GNN Layer` (1–8), and `Anomaly Percent` (20%–50%) yields minimal fluctuation, with optima at 16, 2, and 40%, respectively. This confirms that the topology-aware attention and residual-driven graph construction capture essential spatio-temporal patterns without deep stacking or exhaustive threshold tuning. In contrast, `Window Size` and `Diffusion Level` require domain-aware calibration. MSL peaks at a window of 80, since overly short windows truncate crit-

ical temporal context while overly long windows introduce redundant fluctuations. The diffusion level exhibits a clear *interior optimum* at $\ell^* = 0.3$ with sharp degradation beyond 0.5, consistent with the proxy–target trade-off (Appendix A.3): insufficient perturbation renders proxy anomalies indistinguishable from normal samples, whereas excessive perturbation pushes the proxy distribution away from the severity profile of real anomalies.

4.5 Efficiency Analysis

Table 3: Training and inference efficiency comparison between DiffTSAD and SPAGD on the MSL dataset. ↓: lower is better. Best results in **bold**.

Metric	SPAGD	DiffTSAD
Pre-train (min)	—	0.26
Main Train (min)	7.34	5.40
Total (min) ↓	7.34	**5.66**
Latency (ms) ↓	37.49	32.12
GPU Mem (MB) ↓	11562	4986
Trainable (M) ↓	2.66	2.49

A natural concern regarding the decoupled design is whether the additional diffusion pre-training introduces prohibitive overhead relative to end-to-end baselines. Table 3 compares DiffTSAD with SPAGD on MSL; full results on all three benchmarks are deferred to Appendix D.2. Despite the extra stage, DiffTSAD reduces total training time by 22.9%: the epoch-invariant supervision from the frozen backbone allows the classifier to converge within 20 epochs, whereas SPAGD's jointly trained reconstructor requires substantially more epochs to reach a comparable yet inferior steady state. The pre-training itself is lightweight and constitutes a one-time, dataset-level investment amortized over all downstream runs. At inference, DiffTSAD achieves a 14.3% latency reduction and a 56.9% memory reduction compared to SPAGD. This efficiency gain is primarily attributable to two factors: (1) our topology-aware GAT and chunk-wise dilated convolutions incur lower computational overhead than the global self-attention mechanism of SPAGD's Transformer reconstructor for the given sequence lengths; (2) our reconstruction-error computation employs a highly efficient single-step denoising estimate, and the frozen backbone stores no optimizer states or gradient buffers during the main inference phase. The full comparison confirms consistent gains across datasets, averaging 44.8% training-time, 15.6% latency, and 39.3% memory reductions.

Time Complexity. The computational cost of DiffTSAD decomposes into a one-time pre-training stage and a per-experiment training/inference stage. During pre-training, the Transformer noise predictor processes each window at $\mathcal{O}(T^2 \cdot d_{\text{model}})$ per sample due to self-attention; once frozen, it incurs zero gradient cost during main training. Per training batch, perturbation generation via the closed-form forward schedule (Eq. 1) requires $\mathcal{O}(B \cdot T \cdot d)$ without network evaluation, while dynamic graph construction involves $\mathcal{O}(B \cdot T \cdot d^2)$ for $\mathbf{Z} = \mathbf{X}^\top \mathbf{X}$ and $\mathcal{O}(B \cdot d^2)$ for Top-k sparsification. The L-layer TopoGraphLayer operates on the resulting sparse graph at $\mathcal{O}(B \cdot L \cdot T \cdot d \cdot k \cdot D_h)$ since $|\mathcal{E}| = d \cdot k \ll d^2$, and chunk-wise dilated convolutions (kernel size K) process temporal features at $\mathcal{O}(B \cdot d \cdot D_h \cdot T \cdot K)$. At inference, the reconstruction error is estimated through M independent reconstruction trials, each requiring a closed-form noising step plus a single evaluation of the frozen predictor, reducing the per-trial cost from $\mathcal{O}(T_{\text{diff}} \cdot T^2 \cdot d_{\text{model}})$ for full iterative reverse sampling to $\mathcal{O}(T^2 \cdot d_{\text{model}})$.

5 Conclusion

In this paper, we presented DiffTSAD, a diffusion-guided anomaly detection framework that resolves critical optimization conflicts inherent in conventional unsupervised MTSAD pipelines. By decoupling a frozen pre-trained diffusion model from the discriminative classifier, we introduce severity-calibrated perturbation simulation that eliminates epoch-dependent label inversion and stabilizes training dynamics. The proposed residual-driven dynamic graph construction adaptively modulates inter-variable correlations based on perturbation magnitudes, effectively capturing anomaly-induced structural disruptions. Coupled with a topology-aware enhanced Graph Attention Network and chunk-wise dilated temporal convolutions, the dual-stream architecture achieves robust discrimination between normal patterns and diffusion-simulated anomalies. Comprehensive evaluations on MSL, SMAP, and SWaT datasets confirm that DiffTSAD consistently outperforms state-of-the-art baselines or highly competitive performance in F1-score and AUPRC, validating its effectiveness. Future work will explore extending the diffusion perturbation mechanism to cross-domain adaptation, incorporating online learning for streaming deployment, and optimizing computational efficiency for edge-device integration.

AI USE STATEMENT

In this work, we used generative AI tools for minor copyediting and linguistic polishing of the manuscript. We have not used generative AI tools for ideation, theoretical derivations, model architecture design, code implementation, experimental execution, data analysis, or figure generation, and no other tasks with required disclosure are applicable to this work.

REPRODUCIBILITY STATEMENT

To ensure the reproducibility of our work, we provide comprehensive details across the main text, appendix, and supplementary materials. We release an accessible source code repository, which includes the complete implementation of the DiffTSAD framework, all experimental configurations, hyperparameter settings, and data preprocessing pipelines. For theoretical results, clear explanations of assumptions and complete proofs for Proposition 1 (Stationary proxy supervision) and Proposition 2 (Bounded spectral perturbation) are provided in Appendix A. Detailed descriptions of the datasets, including their statistics and preprocessing steps (e.g., Min-Max normalization), are documented in Appendix C.1. Furthermore, the complete training and inference algorithms are summarized in Algorithm 1 (Appendix B), and extensive ablation studies and sensitivity analyses are provided in Sections 4.3, 4.4, and Appendix D to facilitate the replication of our empirical results.

REFERENCES

Julien Audibert, Pietro Michiardi, Frédéric Guyard, Sébastien Marti, and Maria A Zuluaga. Usad: Unsupervised anomaly detection on multivariate time series. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 3395–3404, 2020.

George EP Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung. *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. Lof: identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pp. 93–104, 2000.

Jinyu Cai, Yuan Xie, Glynnis Lim, Yifang Yin, Roger Zimmermann, and See-Kiong Ng. Self-perturbed anomaly-aware graph dynamics for multivariate time-series anomaly detection. *Advances in Neural Information Processing Systems*, 38:84785–84805, 2026.

Chris U Carmona, François-Xavier Aubet, Valentin Flunkert, and Jan Gasthaus. Neural contextual anomaly detection for time series. *arXiv preprint arXiv:2107.07702*, 2021.

Yuhang Chen, Chaoyun Zhang, Minghua Ma, Yudong Liu, Ruomeng Ding, Bowen Li, Shilin He, Saravan Rajmohan, Qingwei Lin, and Dongmei Zhang. Imdiffusion: Imputed diffusion models for multivariate time series anomaly detection. *arXiv preprint arXiv:2307.00754*, 2023.

Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

Zahra Zamanzadeh Darban, Geoffrey I Webb, Shirui Pan, Charu C Aggarwal, and Mahsa Salehi. Carla: Self-supervised contrastive representation learning for time series anomaly detection. *Pattern Recognition*, 157:110874, 2025.

Ailin Deng and Bryan Hooi. 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.

Evelyn Fix. *Discriminatory analysis: nonparametric discrimination, consistency properties*, volume 1. USAF school of Aviation Medicine, 1985.

Hongfan Gao, Wangmeng Shen, Xiangfei Qiu, Ronghui Xu, Jilin Hu, and Bin Yang. Diffimp: Efficient diffusion model for probabilistic time series imputation with bidirectional mamba backbone. 2024.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Kyle Hundman, Valentino Constantinou, Christopher Laporte, Ian Colwell, and Tom Soderstrom. Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 387–395, 2018.

Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar, and Pierre-Alain Muller. Deep learning for time series classification: a review. *Data mining and knowledge discovery*, 33(4):917–963, 2019.

Siwon Kim, Kukjin Choi, Hyun-Soo Choi, Byunghan Lee, and Sungroh Yoon. Towards a rigorous evaluation of time-series anomaly detection. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pp. 7194–7201, 2022.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Zheng Li, Yue Zhao, Nicola Botta, Cezar Ionescu, and Xiyang Hu. Copod: copula-based outlier detection. In *2020 IEEE international conference on data mining (ICDM)*, pp. 1118–1123. IEEE, 2020.

Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. In *2008 eighth ieee international conference on data mining*, pp. 413–422. IEEE, 2008.

Qinghua Liu and John Paparrizos. The elephant in the room: Towards a reliable time-series anomaly detection benchmark. *Advances in Neural Information Processing Systems*, 37:108231–108261, 2024.

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. itransformer: Inverted transformers are effective for time series forecasting. *arXiv preprint arXiv:2310.06625*, 2023.

Zhe Liu, Xiang Huang, Jingyun Zhang, Zhifeng Hao, Li Sun, and Hao Peng. Multivariate time-series anomaly detection based on enhancing graph attention networks with topological analysis. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pp. 1555–1564, 2024.

Yuan Luo, Ya Xiao, Long Cheng, Guojun Peng, and Danfeng Yao. Deep learning-based anomaly detection in cyber-physical systems: Progress and opportunities. *ACM Computing Surveys (CSUR)*, 54(5):1–36, 2021.

Aditya P Mathur and Nils Ole Tippenhauer. 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.

Guansong Pang, Chunhua Shen, Longbing Cao, and Anton Van Den Hengel. Deep learning for anomaly detection: A review. *ACM computing surveys (CSUR)*, 54(2):1–38, 2021.

Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

Ya Su, Youjian Zhao, Chenhao Niu, Rong Liu, Wei Sun, and Dan Pei. 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, 2019.

Shreshth Tuli, Giuliano Casale, and Nicholas R Jennings. Tranad: Deep transformer networks for anomaly detection in multivariate time series data. *arXiv preprint arXiv:2201.07284*, 2022.

Vladimir N Vapnik. An overview of statistical learning theory. *IEEE transactions on neural networks*, 10(5):988–999, 1999.

Yihang Wang, Yuying Qiu, Peng Chen, Kai Zhao, Yang Shu, Zhongwen Rao, Lujia Pan, Bin Yang, and Chenjuan Guo. Rose: Register-assisted general time series forecasting with decomposed frequency learning. 2024.

Qingsong Wen, Linxiao Yang, Tian Zhou, and Liang Sun. Robust time series analysis and applications: An industrial perspective. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, pp. 4836–4837, 2022.

Hermann Weyl. Das asymptotische verteilungsgesetz der eigenwerte linearer partieller differentialgleichungen (mit einer anwendung auf die theorie der hohlraumstrahlung). *Mathematische Annalen*, 71(4):441–479, 1912.

Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. Timesnet: Temporal 2d-variation modeling for general time series analysis. In *International Conference on Learning Representations*, 2023.

Chunjing Xiao, Zehua Gou, Wenxin Tai, Kunpeng Zhang, and Fan Zhou. Imputation-based time-series anomaly detection with conditional weight-incremental diffusion models. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 2742–2751, 2023.

Jiehui Xu, Haixu Wu, Jianmin Wang, and Mingsheng Long. Anomaly transformer: Time series anomaly detection with association discrepancy. In *International Conference on Learning Representations*, 2022.

Chaocheng Yang, Tingyin Wang, and Xuanhui Yan. Ddmt: Denoising diffusion mask transformer models for multivariate time series anomaly detection. *arXiv preprint arXiv:2310.08800*, 2023a.

Yiyuan Yang, Chaoli Zhang, Tian Zhou, Qingsong Wen, and Liang Sun. Dcdetector: Dual attention contrastive representation learning for time series anomaly detection. In *Proceedings of the 29th ACM SIGKDD conference on knowledge discovery and data mining*, pp. 3033–3045, 2023b.

Weiwei Ye, Zhuopeng Xu, and Ning Gui. Non-stationary diffusion for probabilistic time series forecasting. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=afpc1MFMYU.

Tao Yin, Shaochen Fu, Zhibin Zhang, Li Huang, Xiaohong Zhang, Yiyuan Yang, Kaixiang Yang, and Meng Yan. Scatterad: Temporal-topological scattering mechanism for time series anomaly detection. *Advances in Neural Information Processing Systems*, 38:52266–52297, 2026.

Shicong Zeng, Guoqing Chao, Junquan Wei, Yanwei Yu, Zhijin Wang, and Dianhui Chu. Lagraph: Laplacian-guided graph learning for time series anomaly detection. *IEEE Transactions on Knowledge and Data Engineering*, 2026.

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

Hang Zhao, Yujing Wang, Juanyong Duan, Congrui Huang, Defu Cao, Yunhai Tong, Bixiong Xu, Jing Bai, Jie Tong, and Qi Zhang. Multivariate time-series anomaly detection via graph attention network. In *2020 IEEE international conference on data mining (ICDM)*, pp. 841–850. IEEE, 2020.

A THEORETICAL ANALYSIS AND PROOFS

The following analysis is intended as a motivation for the design of DIFFTSAD, rather than a full finite-sample guarantee for deep networks. We formalize three points: (i) freezing the diffusion backbone yields a stationary proxy supervision signal (§A.1); (ii) residual-driven adjacency injection induces a norm-controlled spectral perturbation before sparsification (§A.2); and (iii) the diffusion level controls a proxy–target trade-off (§A.3).

A.1 STABILITY OF DECOUPLED TRAINING

Since the diffusion backbone is frozen after pre-training, the perturbation mechanism defines a fixed conditional distribution $P(\tilde{X} \mid X, \ell^*)$. We can therefore view the proxy task as learning from a fixed labeled distribution.

Proposition 1 (Stationary proxy supervision)**.** *Let*

$$\mathcal{D} = \frac{1}{2} P_{\mathrm{norm}}(X)\, \delta_{y=0} + \frac{1}{2} P_{\mathrm{norm}}(X)\, P(\tilde{X} \mid X, \ell^*)\, \delta_{y=1}$$

be the joint proxy distribution induced by a frozen diffusion model at level ℓ^*. Let $\mathcal{H}$ be a hypothesis class with finite VC dimension $d_{\mathrm{VC}} < \infty$, and let $\hat{f}_n$ be an empirical risk minimizer over n i.i.d. samples from $\mathcal{D}$. Then, for any $\delta > 0$, with probability at least $1 - \delta$,

$$\mathrm{err}_{\mathcal{D}}(\hat{f}_n) \le \inf_{f \in \mathcal{H}} \mathrm{err}_{\mathcal{D}}(f) + O\left(\sqrt{\frac{d_{\mathrm{VC}} \ln(n/d_{\mathrm{VC}}) + \ln(1/\delta)}{n}} \right). \tag{14}$$

The bound is standard and may be loose for deep models. Its role here is structural: because the diffusion backbone is frozen, the proxy distribution $\mathcal{D}$ is epoch-invariant, so the proxy classification task is well posed from the perspective of statistical learning theory. This statement does not by itself guarantee that diffusion-induced perturbations match real anomalies; it only removes distribution drift caused by a jointly trained generator.

Remark 1 (Non-stationarity of co-trained perturbations)**.** *In a co-training scheme, let $Q_t(X, \tilde{X}) = P_{\mathrm{norm}}(X)\, P_t(\tilde{X} \mid X)$ be the joint distribution induced by the reconstructor/generator at epoch t. If the perturbation severity $s(t) = \mathbb{E}_{(X,\tilde{X}) \sim Q_t}\left[\|X - \tilde{X}\|_1 \right]$ changes across epochs, then Q_t changes. Consequently, the effective labeled proxy distribution $\mathcal{D}_t$ is non-stationary, and the standard i.i.d. argument underlying Proposition 1 does not directly apply. This does not imply that co-training must diverge; rather, it requires a separate non-stationary learning analysis. In practice, the semantics of generated "positives" may drift from coarse large deviations to subtle near-normal deviations, making the proxy supervision less stable. This motivates freezing the perturbation generator.*

A.1.1 PROOF OF PROPOSITION 1

Proof. Because the diffusion backbone is frozen after pre-training, the conditional perturbation distribution $P(\tilde{X} \mid X, \ell^*)$ is fully determined by the fixed noise schedule. Since the training inputs $\{X^{(n)}\}_{n=1}^{N}$ are drawn i.i.d. from P_{norm}, the joint proxy distribution $\mathcal{D}$ is a fixed distribution independent of the training epoch. Drawing n i.i.d. samples from $\mathcal{D}$, the classical VC generalization bound Shalev-Shwartz & Ben-David (2014) gives: with probability at least $1 - \delta$,

$$\mathrm{err}_{\mathcal{D}}(\hat{f}_n) \le \inf_{f \in \mathcal{H}} \mathrm{err}_{\mathcal{D}}(f) + \sqrt{\frac{8\big(d_{\mathrm{VC}} \ln(2en/d_{\mathrm{VC}}) + \ln(4/\delta)\big)}{n}}.$$

Absorbing constants into the $O(\cdot)$ notation yields Eq. 14. □

Relation to paired dual-stream training. The training objective in the main text uses paired normal/perturbed sequences generated from the same X. This does not change the stationarity argument: the pairs are i.i.d. from $Q(X, \tilde{X}) = P_{\mathrm{norm}}(X)\, P(\tilde{X} \mid X, \ell^*)$, and the expected paired loss is fixed. A formal paired generalization bound can be obtained by applying uniform convergence to the paired-loss class; the precise complexity term may differ from the standard VC term above. Since our goal is to motivate the decoupled design, we state the simpler standard form.

Remark on tightness. For deep neural networks, d_{VC} can be very large, rendering the bound in Eq. 14 potentially vacuous. The value of Proposition 1 is therefore not the tightness of the rate, but the structural observation that freezing the perturbation generator removes epoch-dependent distribution drift. We do not claim convergence of non-convex optimization to a global minimizer.

A.1.2 Derivation of Remark 1

Let

$$Q_t(X, \tilde{X}) = P_{\mathrm{norm}}(X)\, P_t(\tilde{X} \mid X)$$

denote the joint distribution induced by the reconstructor/generator G_t at epoch t. Define the perturbation severity

$$s(t) = \mathbb{E}_{(X,\tilde{X})\sim Q_t}\big[\|X - \tilde{X}\|_1\big].$$

If $s(t+1) \neq s(t)$, then the expectation of the measurable function $(X, \tilde{X}) \mapsto \|X - \tilde{X}\|_1$ under Q_{t+1} differs from its expectation under Q_t. Hence $Q_{t+1} \neq Q_t$. Equivalently, the induced labeled proxy distribution $\mathcal{D}_t$ varies with t. The classical ERM consistency theorem (Vapnik, 1999) assumes that all training samples are drawn i.i.d. from a single fixed distribution. Under co-training, the effective training distribution changes across epochs, so this standard i.i.d. argument does not directly apply. This does not prove that co-training fails or diverges; it only indicates that a non-stationary learning analysis would be required.

A.2 Spectral Effect of Dynamic Graph Injection

We analyze the residual injection step before Top-k masking. To avoid overloading the symbol Δ, which is used for the reconstruction error in the inference score, we denote the adjacency perturbation matrix by Γ.

Proposition 2 (Bounded spectral perturbation of the pre-mask graph)**.** *Let $\tilde{A}^{(0)} \in \mathbb{R}^{d\times d}$ be the baseline symmetric adjacency matrix for the perturbed stream before residual injection, and let $\tilde{A}^{(\ell^*)}_{\mathrm{pre}} = \tilde{A}^{(0)} + \Gamma$ be the corresponding matrix after residual injection but before Top-k masking, where Γ is the injection term defined by Eq. 4. Let $m = |\mathcal{M}|$ be the number of high-residual candidate nodes. Then*

$$\max_{1\le i\le d} \big|\lambda_i\big(\tilde{A}^{(\ell^*)}_{\mathrm{pre}}\big) - \lambda_i\big(\tilde{A}^{(0)}\big)\big| \le \|\Gamma\|_2 \le 2\sqrt{d\,m} + 1, \tag{15}$$

where $\lambda_i(\cdot)$ denotes the i-th eigenvalue in non-increasing order.

Proposition 2 should be read as a motivation rather than a full guarantee for the final sparse graph: it shows that the residual-driven injection itself is norm-controlled before sparsification. The subsequent Top-k masking is nonlinear and implementation-dependent; the proposition does not by itself prove preservation of global connectivity or local amplification after masking. The empirical effect of the full dynamic graph module is validated in the ablation study.

A.2.1 Proof of Proposition 2

Proof. **Step 1: Symmetry.** The baseline matrix $\tilde{A}^{(0)}$ is symmetric because it is constructed from a symmetric similarity transformation and element-wise activation. The injection term in Eq. 4 can be written as

$$\Gamma_{ij} = \mathbb{I}(i \in \mathcal{M})\hat{r}_i + \mathbb{I}(j \in \mathcal{M})\hat{r}_j - \delta_{ij}\,\mathbb{I}(i \in \mathcal{M})\hat{r}_i,$$

which is symmetric in (i, j) by direct inspection. Hence $\tilde{A}^{(\ell^*)}_{\mathrm{pre}} = \tilde{A}^{(0)} + \Gamma$ is symmetric, and Weyl's inequality (Weyl, 1912) applies.

Step 2: Weyl's inequality. For symmetric matrices $A, B \in \mathbb{R}^{d\times d}$, Weyl's inequality states

$$\max_i \big|\lambda_i(A + B) - \lambda_i(A)\big| \le \|B\|_2.$$

Setting $A = \tilde{A}^{(0)}$ and $B = \Gamma$ gives

$$\max_i \big|\lambda_i\big(\tilde{A}^{(\ell^*)}_{\mathrm{pre}}\big) - \lambda_i\big(\tilde{A}^{(0)}\big)\big| \le \|\Gamma\|_2.$$

Step 3: Bounding $\|\Gamma\|_2$. Let $\hat{r}_{\mathcal{M}} \in \mathbb{R}^d$ denote the residual vector with $(\hat{r}_{\mathcal{M}})_i = \hat{r}_i$ if $i \in \mathcal{M}$ and 0 otherwise, and let $\mathbf{1} \in \mathbb{R}^d$ be the all-ones vector. Then

$$\Gamma = \hat{r}_{\mathcal{M}}\mathbf{1}^\top + \mathbf{1}\hat{r}_{\mathcal{M}}^\top - \mathrm{diag}(\hat{r}_{\mathcal{M}}).$$

By the triangle inequality and sub-multiplicativity of the spectral norm,

$$\begin{aligned}\|\Gamma\|_2 &\le \|\hat{r}_{\mathcal{M}}\mathbf{1}^\top\|_2 + \|\mathbf{1}\hat{r}_{\mathcal{M}}^\top\|_2 + \|\mathrm{diag}(\hat{r}_{\mathcal{M}})\|_2 \\ &= \|\hat{r}_{\mathcal{M}}\|_2\|\mathbf{1}\|_2 + \|\mathbf{1}\|_2\|\hat{r}_{\mathcal{M}}\|_2 + \|\hat{r}_{\mathcal{M}}\|_\infty \\ &= 2\sqrt{d}\,\|\hat{r}_{\mathcal{M}}\|_2 + \|\hat{r}_{\mathcal{M}}\|_\infty.\end{aligned}$$

Since $\hat{r}_i \in [0,1]$ and at most m entries are non-zero, $\|\hat{r}_{\mathcal{M}}\|_2 \le \sqrt{m}$ and $\|\hat{r}_{\mathcal{M}}\|_\infty \le 1$. Therefore, $\|\Gamma\|_2 \le 2\sqrt{dm}$. Combining Steps 2 and 3 yields Eq. 15. □

Scope of the statement. Proposition 2 concerns the symmetric matrix after residual injection but before Top-k masking. The final Top-k masked graph is a nonlinear, sparse, and implementation-dependent transformation of this pre-mask matrix. If the masking operation is row-wise and does not enforce symmetry, the final adjacency may be directed, and Weyl's inequality does not directly apply to its eigenvalues. Thus, the proposition should be interpreted as showing that the residual injection itself is spectrally controlled, rather than as a complete spectral guarantee for the final sparse graph.

A.3 Perturbation Severity as a Proxy–Target Trade-off

The diffusion level ℓ controls the strength of simulated perturbations. To discuss its effect, let $\mathcal{D}_{\mathrm{test}}$ denote the labeled test distribution containing both normal and anomalous windows. For a classifier f_ℓ trained on the proxy distribution

$$\mathcal{D}_\ell = \frac{1}{2}P_{\mathrm{norm}}(X)\,\delta_{y=0} + \frac{1}{2}P_{\mathrm{norm}}(X)\,P(\tilde{X} \mid X, \ell)\,\delta_{y=1},$$

a standard total-variation argument gives

$$\mathrm{err}_{\mathcal{D}_{\mathrm{test}}}(f_\ell) \le \mathrm{err}_{\mathcal{D}_\ell}(f_\ell) + D_{\mathrm{TV}}\big(\mathcal{D}_\ell, \mathcal{D}_{\mathrm{test}}\big) \le \underbrace{\inf_{f\in\mathcal{H}} \mathrm{err}_{\mathcal{D}_\ell}(f)}_{\text{source risk}} + \underbrace{\epsilon_{\mathrm{approx}}(\ell)}_{\text{excess risk}} + \underbrace{D_{\mathrm{TV}}\big(\mathcal{D}_\ell, \mathcal{D}_{\mathrm{test}}\big)}_{\text{proxy–target mismatch}}, \tag{16}$$

where

$$\epsilon_{\mathrm{approx}}(\ell) = \mathrm{err}_{\mathcal{D}_\ell}(f_\ell) - \inf_{f\in\mathcal{H}} \mathrm{err}_{\mathcal{D}_\ell}(f) \ge 0.$$

Eq. 16 is intended as a trade-off perspective rather than a rigorous guarantee of the optimal level. A smaller ℓ produces perturbations closer to normal data, which may make the proxy classes harder to separate; a larger ℓ produces stronger perturbations, which may increase separability but also increase mismatch to subtle real anomalies. In our implementation, ℓ is selected from a finite candidate set, so an empirical minimizer over validation performance exists trivially. A continuous extreme-value argument would require additional assumptions, such as continuity of $\ell \mapsto \mathcal{D}_\ell$ and of the relevant error terms, which we do not rely on.

Remark 2 (Dataset-dependent severity preference)**.** *On SMAP, many anomalies are high-severity sensor failures, so stronger perturbations may be closer to real faults; this is consistent with the observed improvement at larger ℓ. On MSL, anomalies are often subtle contextual drifts, so overly strong perturbations may increase proxy–target mismatch, leading to an interior optimum. This explanation is empirical and qualitative; Eq. 16 does not by itself prove the monotonicity of D_{TV} or of the total bound.*

A.3.1 Derivation of the Severity Trade-off

Let $\mathcal{D}_{\mathrm{test}}$ denote the labeled test distribution containing both normal and anomalous windows. For example, if the test anomaly rate is ρ, one may write

$$\mathcal{D}_{\mathrm{test}} = (1-\rho)P_{\mathrm{norm,test}}(X)\,\delta_{y=0} + \rho P_{\mathrm{anom,test}}(X)\,\delta_{y=1}.$$

This avoids the ambiguity of evaluating binary classification on an anomaly-only distribution.

For any classifier f_ℓ and any two labeled distributions $\mathcal{D}$ and $\mathcal{D}'$, the difference in expected 0-1 loss is bounded by their total variation distance:

$$\left|\mathrm{err}_{\mathcal{D}}(f_\ell) - \mathrm{err}_{\mathcal{D}'}(f_\ell)\right| \le D_{\mathrm{TV}}(\mathcal{D}, \mathcal{D}').$$

This is a standard property of total variation distance. Applying it with $\mathcal{D} = \mathcal{D}_\ell$ and $\mathcal{D}' = \mathcal{D}_{\mathrm{test}}$ gives

$$\mathrm{err}_{\mathcal{D}_{\mathrm{test}}}(f_\ell) \le \mathrm{err}_{\mathcal{D}_\ell}(f_\ell) + D_{\mathrm{TV}}\big(\mathcal{D}_\ell, \mathcal{D}_{\mathrm{test}}\big).$$

By definition,

$$\epsilon_{\mathrm{approx}}(\ell) = \mathrm{err}_{\mathcal{D}_\ell}(f_\ell) - \inf_{f \in \mathcal{H}} \mathrm{err}_{\mathcal{D}_\ell}(f) \ge 0,$$

so

$$\mathrm{err}_{\mathcal{D}_\ell}(f_\ell) \le \inf_{f \in \mathcal{H}} \mathrm{err}_{\mathcal{D}_\ell}(f) + \epsilon_{\mathrm{approx}}(\ell).$$

Substituting into the previous display yields Eq. 16.

Interpretation. The source risk $\inf_{f \in \mathcal{H}} \mathrm{err}_{\mathcal{D}_\ell}(f)$ is the best-in-class risk under the proxy distribution, not the target risk under $\mathcal{D}_{\mathrm{test}}$. The total variation term absorbs proxy–target mismatch, including differences in anomaly semantics and class prior. We do not assert that $\epsilon_{\mathrm{approx}}(\ell)$ is monotone in general. Larger ℓ may increase proxy separability, but it may also move the proxy perturbations away from subtle real anomalies. Similarly, $D_{\mathrm{TV}}(\mathcal{D}_\ell, \mathcal{D}_{\mathrm{test}})$ need not be monotone without additional assumptions on the data and the diffusion process.

Existence of a selected level. In the implementation, ℓ is chosen from a finite candidate set, e.g., $\ell \in \{0.0, 0.1, \ldots, 1.0\}$. Therefore, an empirical minimizer over a validation criterion exists trivially. If one instead treats ℓ as continuous, an extreme-value argument would require additional continuity or lower-semicontinuity assumptions on the map $\ell \mapsto \mathcal{D}_\ell$ and on the relevant error terms. Since our algorithm only uses a finite grid, we do not rely on such continuous-domain assumptions.

B Summary of DiffTSAD Training and Inference

For reproducibility, we summarize the complete decoupled training and hybrid inference pipeline of DiffTSAD in Algorithm 1.

C Datasets and Experimental Settings

C.1 Datasets

We evaluate DiffTSAD on three widely adopted real-world benchmarks: MSL (Mars Science Laboratory) Hundman et al. (2018), SMAP (Soil Moisture Active Passive) Hundman et al. (2018), and SWaT (Secure Water Treatment) Mathur & Tippenhauer (2016). These datasets comprise multivariate time series collected from sensor networks monitoring spacecraft telemetry, environmental conditions, and industrial water treatment processes, respectively. Following standard preprocessing protocols, all raw time series are normalized to the range $[0, 1]$ using Min-Max scaling. Adhering to the standard unsupervised MTSAD protocol, all models are trained exclusively on normal samples, while evaluation is conducted on test sets containing a mixture of normal and anomalous instances. Detailed dataset statistics, including dimensionality, sample sizes, and anomaly ratios, are summarized in Table 4. MSL and SMAP feature point and contextual faults across 55 and 25 channels, respectively, whereas SWaT provides a rigorous industrial control system benchmark with 51 variables and 11.98% anomaly prevalence, encompassing complex physical and cyber-physical attacks. Collectively, these benchmarks offer a comprehensive evaluation ground for assessing detection robustness across varying dimensionalities, anomaly severities, and domain-specific distribution shifts.

Table 4: Overview of datasets used in the experiments.

Dataset	Dimension	# Train samples (Normal)	# Test samples (Mixed)	Anomaly Rate (Testing)
MSL	55	58,317	73,729	10.72%
SMAP	25	135,183	427,617	13.13%
SWaT	51	496,800	449,919	11.98%

Algorithm 1 DiffTSAD: Decoupled Training and Hybrid Inference

Require: Normal training set $\mathcal{D}_{\text{train}}$; diffusion levels $\mathcal{L}$ with optimal ℓ^*; top-$m\%$ ratio; top-k; fusion weight α; threshold τ.
Ensure: Anomaly scores $\{S(\mathbf{X}_{\text{test}})\}$ and predictions $\{\hat{y}\}$.

// Phase 0: One-time diffusion pre-training
Train $\boldsymbol{\epsilon}_\theta$ on $\mathcal{D}_{\text{train}}$: $\min_\theta \mathcal{L}_{\text{diff}} = \mathbb{E}_{t,\mathbf{X},\boldsymbol{\epsilon}}[\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\tilde{\mathbf{X}}_t, t)\|^2]$
Freeze θ ($\nabla_\theta \mathcal{L} = \mathbf{0}$) for all subsequent phases
// Phase 1: Stationary proxy classification training
for each epoch; each batch $\{\mathbf{X}^{(n)}\}_{n=1}^{B} \subset \mathcal{D}_{\text{train}}$ **do**
  $t \leftarrow \lfloor \ell^* \cdot T_{\text{diff}} \rfloor$; sample $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  $\tilde{\mathbf{X}}_{\ell^*}^{(n)} \leftarrow \sqrt{\bar{\alpha}_t}\,\mathbf{X}^{(n)} + \sqrt{1-\bar{\alpha}_t}\,\boldsymbol{\epsilon}$ {frozen perturbation}
  $r_i \leftarrow \frac{1}{T}\sum_t |X_{t,i}^{(n)} - \tilde{X}_{\ell^*,t,i}^{(n)}|$; $\hat{r}_i \leftarrow \sigma(r_i), \forall i \in \{1,\dots,d\}$
  $\mathbf{A}^{(0)} \leftarrow \sigma\big((\mathbf{Z} - \mu(\mathbf{Z}))/(\text{std}(\mathbf{Z}) + \epsilon)\big)$ with $\mathbf{Z} = \mathbf{X}^{(n)\top}\mathbf{X}^{(n)}$
  $\tilde{\mathbf{A}}^{(0)} \leftarrow$ same construction applied to $\tilde{\mathbf{X}}_{\ell^*}^{(n)}$
  $\mathcal{M} \leftarrow$ top-$m\%$ indices of $\{r_i\}$
  $\tilde{\mathbf{A}}_{ij}^{(\ell^*)} \leftarrow \tilde{\mathbf{A}}_{ij}^{(0)} + \mathbb{I}(i \in \mathcal{M})\hat{r}_i + \mathbb{I}(j \in \mathcal{M})\hat{r}_j - \delta_{ij}\mathbb{I}(i \in \mathcal{M})\hat{r}_i$; apply Top-$k$ masking
  $(\mathbf{Z}_{\text{temp}}, \tilde{\mathbf{Z}}_{\text{temp}}) \leftarrow$ shared TopoGraphLayers + chunk-wise temporal convs on $(\mathbf{X}^{(n)}, \mathbf{A}^{(0)})$ and $(\tilde{\mathbf{X}}_{\ell^*}^{(n)}, \tilde{\mathbf{A}}^{(\ell^*)})$
  $\hat{p}_{\text{norm}}^{(n)} \leftarrow \sigma(\mathcal{P}_\psi(\text{Flatten}(\mathbf{Z}_{\text{temp}})))$; $\hat{p}_{\text{pert}}^{(n)} \leftarrow \sigma(\mathcal{P}_\psi(\text{Flatten}(\tilde{\mathbf{Z}}_{\text{temp}})))$
  $\mathcal{L}_{\text{cls}} \leftarrow -\frac{1}{2B}\sum_{n=1}^{B}\Big[\log(1 - \hat{p}_{\text{norm}}^{(n)}) + \log \hat{p}_{\text{pert}}^{(n)}\Big]$
  Update graph constructor & classifier parameters via $\nabla\mathcal{L}_{\text{cls}}$ {θ remains frozen}
end for
// Phase 2: Inference and hybrid anomaly scoring
for each test sequence $\mathbf{X}_{\text{test}}$ **do**
  Construct $\mathbf{A}_{\text{test}}^{(0)}$, $\tilde{\mathbf{A}}_{\text{test}}^{(\ell^*)}$ as in Phase 1
  $\hat{p}_{\text{cls}} \leftarrow \sigma(\mathcal{P}_\psi(\mathbf{v}_{\text{test}}))$
  $\hat{\mathbf{X}}_{\text{recon}} \leftarrow$ mean of M=3 single-step reconstruction trials under frozen θ
  $\Delta \leftarrow \frac{1}{Td}\sum_{t,i} |X_{\text{test},t,i} - \hat{X}_{\text{recon},t,i}|$
  $S \leftarrow \alpha\,\hat{p}_{\text{cls}} + (1-\alpha)\,\sigma\big((\Delta - \text{median}(\Delta))/(\text{IQR}(\Delta) + \epsilon)\big)$
  $\hat{y} \leftarrow \mathbb{I}(S \geq \tau)$
end for
return $\{S(\mathbf{X}_{\text{test}}), \hat{y}\}$

C.2 Baselines

To assess the performance of DiffTSAD, we compare it to fourteen methods that belong to two categories: classical machine learning approaches and deep learning–based methods.

- **Classical Machine Learning Methods**
  - **k-NN** Fix (1985): identifies anomalies by computing the average distance to the k-nearest neighbors in the feature space, assuming that anomalous points are significantly farther from their neighbors than normal points.
  - **OCSVM** Cortes & Vapnik (1995): constructs a hyperplane that separates normal data from the origin in a high-dimensional feature space, treating samples falling outside the decision boundary as anomalies without requiring explicit anomalous labels during training.
  - **LOF** Breunig et al. (2000): detects anomalies by measuring local density deviation of each point with respect to its neighbors, identifying instances with substantially lower density than their neighbors as outliers in multivariate time series.
  - **IForest** Liu et al. (2008): isolates anomalies by randomly selecting features and split values, exploiting the fact that anomalies are few and different, thus requiring shorter average path lengths to be isolated in the tree structure.

- **COPOD** Li et al. (2020): employs empirical copulas to model the dependence structure among variables without parametric assumptions, computing anomaly scores based on tail probabilities of the joint distribution.

- **Deep Learning Methods**
  - **USAD** Audibert et al. (2020): employs encoder-decoder paradigm with two adversarial autoencoders, amplifying reconstruction errors through adversarial training to enhance sensitivity to subtle anomalies in multivariate time series.
  - **NCAD** Carmona et al. (2021): employs graph neural networks to model temporal dependencies within a contextual framework, learning representations that distinguish normal patterns from anomalous deviations through neighborhood aggregation.
  - **AnomalyTrans (AT)** Xu et al. (2022): adopts a Transformer-based architecture with association discrepancy learning, distinguishing anomalies by comparing prior and series associations to capture temporal dependencies and deviation patterns.
  - **IMDiffusion** Chen et al. (2023): combines time series imputation with diffusion models, leveraging step-wise denoising errors and voting ensemble for anomaly scoring, published in VLDB 2024.
  - **DiffAD** Xiao et al. (2023): proposes a conditional weight-incremental diffusion imputation framework with a density-ratio-based selection strategy and multi-scale state space modeling to handle anomaly concentration scenarios, published in KDD 2023.
  - **DDMT** Yang et al. (2023a): integrates Transformer architectures with denoising diffusion models via an adaptive dynamic neighbor mask mechanism to mitigate information leakage during reconstruction, published in arXiv 2023.
  - **TimesNet** Wu et al. (2023): transforms 1D time series into 2D tensors to capture intra-period and inter-period variations simultaneously, employing convolutional neural networks for comprehensive temporal modeling and anomaly identification.
  - **DCdetector** Yang et al. (2023b): utilizes dual-attention asymmetric design with contrastive learning, creating permutation-invariant embeddings through pure contrastive loss to achieve superior discrimination between normal and anomalous patterns.
  - **SPAGD** Cai et al. (2026): generates auxiliary samples via reconstruction residuals and dynamically adjusts graph structures for spatio-temporal classification, serving as the most directly related baseline to our DiffTSAD.

C.3 SETTINGS

All experiments are conducted on a single RTX 4090 GPU. The experimental configuration consists of two stages: *Pre-trained Diffusion* and *Main Training*. For the diffusion backbone, we adopt a sliding window length of 100, with model hyperparameters set to $d_{\text{model}} = 64$, 8 attention heads, and 4 encoder layers. The diffusion process follows a linear noise schedule with $T = 1000$ steps, $\beta_{\text{start}} = 10^{-4}$, and $\beta_{\text{end}} = 0.02$. It is trained for 200 epochs using batch size 64 and a learning rate of 10^{-3}. During main training phase, the diffusion weights are frozen. The spatio-temporal classifier is optimized for 20 epochs with a batch size of 32, an initial learning rate of 10^{-3}, and an early stopping patience of 5 epochs. The top-k in dynamic graph construction is set to 15. In the Graph Attention Network, D_{in}, D_{out}, and D_s are all set to 16. The Adam optimizer is employed throughout both stages. During inference, the fusion weight α is set to 0.5 to balance discriminative and generative signals.

C.4 METRICS

We evaluate detection performance using three standard metrics: Area Under the ROC Curve (AUC), Area Under the Precision-Recall Curve (AUPRC), and the F1-score. AUC and AUPRC provide threshold-independent assessments of the model's ranking capability, with AUPRC being particularly informative under severe class imbalance. The F1-score is computed at the optimal threshold that maximizes the harmonic mean of precision and recall, offering a balanced measure of point-wise detection accuracy. To ensure rigorous and unbiased evaluation, we deliberately **exclude** performance-inflating post-processing techniques such as Point Adjustment (PA) from our assessment protocol Kim et al. (2022); Liu & Paparrizos (2024). All reported metrics are computed

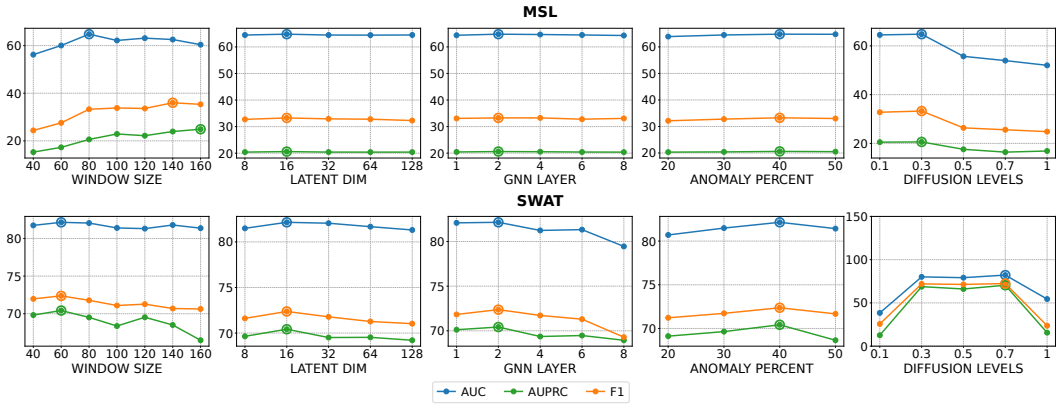


Figure 4: Sensitivity analysis of key hyperparameters on model performance across AUC, AUPRC, and F1 (in %).

directly on raw anomaly scores against ground-truth labels, strictly adhering to point-wise evaluation standards without label relaxation or score manipulation.

D Experiment Results

D.1 Parameter Sensitivity Analysis

To evaluate the stability and practical deployability of DiffTSAD, we conduct a comprehensive sensitivity analysis on five critical hyperparameters: input window size, latent dimension, number of GNN layers, anomaly candidate percentage, and diffusion noise level. Experiments are performed on MSL and SWaT with all other configurations fixed in Figure 4.

The model demonstrates remarkable robustness to architectural and structural hyperparameters. Varying the `Latent Dim` (8–128) and `GNN Layer` depth (1–8) yields minimal performance fluctuation, with optimal values at 16/2 for both MSL and SWaT. This indicates that the proposed topology-aware attention and chunk-wise temporal modeling efficiently capture essential spatio-temporal patterns without requiring deep stacking or high-dimensional embeddings, thereby avoiding gradient vanishing and overfitting. Similarly, the `Anomaly Percent` (top-$m\%$ variables selected for dynamic graph adjustment) remains highly stable across 20%–50%, peaking at 40% on both datasets. This confirms that the residual-driven graph construction reliably identifies perturbation-sensitive nodes without being overly sensitive to the threshold, significantly reducing the computational burden of exhaustive grid search in real-world deployment.

In contrast, `Window Size` and `Diffusion Levels` exhibit dataset-dependent sensitivity. For window size, MSL achieves peak performance at 80, while SWaT favors 60. Windows that are too short truncate critical temporal context and impair long-range dependency modeling, whereas excessively long windows introduce redundant fluctuations and increase computational overhead. Regarding diffusion levels, both datasets exhibit a clear *interior optimum*: MSL peaks at $\ell^* = 0.3$ with sharp degradation beyond 0.5, while SWaT peaks at $\ell^* = 0.7$ with substantial drops at both extremes ($\ell = 0.1$ and $\ell = 1.0$). This pattern is consistent with the proxy–target trade-off analysis in Section A: insufficient perturbation (ℓ too small) renders proxy anomalies indistinguishable from normal samples, while excessive perturbation (ℓ too large) pushes the proxy distribution away from the severity profile of real anomalies, increasing distribution mismatch. The differing optimal levels across MSL ($\ell^* = 0.3$) and SWaT ($\ell^* = 0.7$) further confirm that perturbation severity must be calibrated to domain-specific anomaly characteristics—MSL anomalies are predominantly subtle contextual drifts favoring mild perturbations, whereas SWaT encompasses more severe physical/cyber-physical attacks that require stronger simulated deviations.

Overall, DiffTSAD exhibits strong hyperparameter stability, with only temporal context length and diffusion intensity requiring domain-aware calibration. The consistent optimal ranges across metrics and datasets confirm that the framework is lightweight to tune and highly adaptable to diverse industrial monitoring scenarios.

Table 5: Training and inference efficiency comparison between DiffTSAD and SPAGD. ↓: lower is better. Best results in **bold**.

Dataset	Method	Training Time (min) ↓			Inference / Resources		
		Pre-train	Main Train	Total	Latency (ms) ↓	GPU Mem (MB) ↓	Trainable (M) ↓
MSL	SPAGD	—	7.34	7.34	37.49	11562	2.66
	DiffTSAD	0.26	5.40	**5.66**	32.12	4986	2.49
SMAP	SPAGD	—	26.62	26.62	34.24	2599	1.51
	DiffTSAD	0.49	10.78	**11.27**	28.85	2221	1.44
SWaT	SPAGD	—	100.29	100.29	35.73	10052	2.51
	DiffTSAD	3.84	42.35	**46.19**	29.68	5378	2.40

D.2 Efficiency Analysis

A natural concern regarding the decoupled design is whether the additional diffusion pre-training stage introduces prohibitive computational overhead relative to end-to-end baselines. To address this, we conduct a comprehensive wall-clock timing comparison between DiffTSAD and SPAGD under the same settings.

As reported in Table 5, DiffTSAD achieves lower total training time than SPAGD across all three benchmarks despite incorporating an additional pre-training stage, with an average reduction of 44.8%. This result may appear counterintuitive but follows directly from the optimization dynamics of the decoupled paradigm. Because the frozen diffusion backbone provides epoch-invariant supervision, the downstream classifier converges within 20 epochs under early stopping, whereas SPAGD's jointly trained Transformer reconstructor requires substantially more epochs to reach a comparable yet still inferior steady state. The diffusion pre-training itself is remarkably lightweight, constituting less than 9% of the total training budget even on the largest SWaT dataset, owing to early stopping that terminates optimization well before the 200-epoch budget is exhausted.

Furthermore, the pre-training cost is a one-time, dataset-level investment that is amortized across all downstream experiments. In our experimental campaign, the frozen backbone trained once per dataset is reused across the main comparison, all ablation variants, and the full hyperparameter sensitivity analysis, totaling approximately 90 independent training runs. The effective per-experiment overhead attributable to pre-training is therefore negligible. This amortization paradigm is analogous to the pre-train-then-adapt workflow widely adopted in foundation models, where the upfront cost is distributed over a large number of downstream configurations.

At inference time, DiffTSAD achieves an average latency reduction of 15.6% over SPAGD. This gain over SPAGD is primarily attributable to the architectural differences: our topology-aware GAT and chunk-wise temporal convolutions incur lower latency than the global self-attention mechanism of SPAGD's Transformer reconstructor. Furthermore, our reconstruction-error computation employs a highly efficient single-step denoising estimate, avoiding any multi-step iterative processes inherent to standard diffusion models. In terms of resource footprint, the frozen backbone stores no optimizer states or gradient buffers during main training, yielding an average reduction of 39.3% in peak GPU memory. The trainable parameter count is also marginally smaller, as the diffusion parameters are excluded from gradient computation. Collectively, these results confirm that the decoupled architecture does not introduce net computational cost but instead yields consistent efficiency improvements over end-to-end co-training while delivering superior detection accuracy.

Time Complexity. The computational cost of DiffTSAD decomposes into a one-time diffusion pre-training stage and a per-experiment training and inference stage. During pre-training, the Transformer-based noise predictor processes each training window at $\mathcal{O}(T^2 \cdot d_{\text{model}})$ per sample due to self-attention, with early stopping typically terminating well before the 200-epoch budget. Once frozen, the diffusion backbone incurs zero gradient cost during main training. Per training batch, perturbation generation via the closed-form forward schedule $\tilde{\mathbf{X}}_t = \sqrt{\bar{\alpha}_t}\,\mathbf{X} + \sqrt{1-\bar{\alpha}_t}\,\boldsymbol{\epsilon}$ requires $\mathcal{O}(B \cdot T \cdot d)$ without any network evaluation, while dynamic graph construction involves $\mathcal{O}(B \cdot T \cdot d^2)$ for the similarity matrix $\mathbf{Z} = \mathbf{X}^\top \mathbf{X}$ and $\mathcal{O}(B \cdot d \cdot k)$ for Top-k sparsification. The L-layer TopoGraphLayer operates on the resulting sparse graph at $\mathcal{O}(B \cdot L \cdot T \cdot d \cdot k \cdot D_h)$ since $|\mathcal{E}| = d \cdot k \ll d^2$, and chunk-wise dilated convolutions process temporal features at $\mathcal{O}(B \cdot d \cdot D_h \cdot D_t)$. At inference, the reconstruction error is estimated through M independent **single-step reconstruc-**

tion trials, each requiring a closed-form noising step followed by a single evaluation of the frozen noise predictor, reducing the per-trial cost from $O(T_{diff} \cdot T^2 \cdot d_{model})$ for full iterative reverse sampling (**as used in standard diffusion detectors**) to $O(T^2 \cdot d_{model})$ for our single-step denoising estimate.