
Appendix of Self-Perturbed Anomaly-Aware Graph Dynamics for Multivariate Time-Series Anomaly Detection

Jinyu Cai

Institute of Data Science
National University of Singapore
jinyucaai@nus.edu.sg

Yuan Xie*

School of Computing
National University of Singapore
xieyuan_sss@outlook.com

Glynnis Lim

Institute of Data Science
National University of Singapore
glynnis@nus.edu.sg

Yifang Yin

Institute for Infocomm Research
A*STAR, Singapore
yin_yifang@i2r.a-star.edu.sg

Roger Zimmermann

School of Computing
National University of Singapore
dcsrz@nus.edu.sg

See-Kiong Ng

Institute of Data Science
National University of Singapore
seekiong@nus.edu.sg

A Detailed Datasets Description

In this section, we provide a detailed introduction to the datasets used in our experiment:

1. **SWaT:** The SWaT dataset is collected from a real-world operational water treatment testbed, capturing sensor readings and actuator status, over a period of normal and attacked operations [Mathur and Tippenhauer, 2016]. It contains 51 sensor readings and actuator statuses, and the data includes various cyber-physical attacks that cause anomalies in the sensor readings. The training set comprises 496,800 samples representing normal operations, while the test set contains 449,919 samples, which include a mix of normal and anomalous time-series data. The anomaly ratio in the test set is 11.98%. Anomalies are point-wise and represent deviations caused by simulated attacks on the physical process.
2. **SMAP:** The SMAP dataset comprises telemetry data from NASA’s SMAP satellite and includes various channels representing different spacecraft subsystems [Hundman *et al.*, 2018]. It consists of 25 telemetry channels. The training data contains 135,183 normal samples, and the test set includes 427,617 mixed samples. The anomaly ratio in the test set is 13.13%. Anomalies in this dataset are typically point anomalies identified by domain experts, often related to unusual spacecraft behavior or events.
3. **MSL:** The MSL dataset also consists of telemetry data, sourced from the MSL rover (Curiosity) developed by NASA [Hundman *et al.*, 2018]. It includes data from 55 telemetry channels. The training set has 58,317 normal samples, and the test set contains 73,729 mixed samples. The anomaly ratio here is 10.72%. Similar to SMAP, anomalies are point-wise and represent unexpected readings or events in the rover’s telemetry streams.

*Corresponding Author

B Detailed Experimental Settings

- **Data Split:** For all datasets used in our experiments, we followed the same dataset split. Specifically, we preprocess the datasets by dividing them into training and testing sets: the training set comprises exclusive time-series data from normal operational conditions, while the test set contains a mixture of normal and anomalous data segments. A validation set used for hyperparameter tuning and early stopping is created by randomly partitioning 10% of the training data. Consequently, the validation set also exclusively contains normal time-series data. To further preprocess the time-series data for model training, we utilize a sliding window segmentation method, where each time series is segmented into windows of a fixed length of 100. These windows are generated with a stride equal to the window size, resulting in non-overlapping segments.
- **Hyper-Parameter Settings:** Except for the parameter settings provided in the main text, we further supplemented more detailed parameter settings here. Specifically, we use $K = 15$ to construct the K -nearest neighbor (K -NN) graph structure for the spatial graph modeling. For training, we set the batch size uniformly to 64 across all datasets, and the chunks $C = 5$ to segment time-series data for temporal modeling. The trade-off parameter β is set to $\beta = 0.01$ for all datasets. In particular, we evaluated the impact of hyperparameters such as window size and β on the performance (refer to Figure 2 in the main text). In the **Appendix D**, we further evaluated the number of GAT layers and latent dimensions for the anomaly detection performance.
- **Training Details:** For graph construction, we symmetrize the dynamic similarity before sparsification to ensure numerical stability. We adopted Adam [Kingma and Ba, 2014] optimizer with fixed learning rate of 0.001 to train the model. We particularly use an early stopping mechanism to train the model, which means that the training will halt once the validation loss does not improve.
- **Baseline Settings:** We use the publicly available code of all baselines and follow the default hyperparameter settings in their papers to guarantee the reproducibility and fairness of the experiment. We provide the relevant links below to access the baseline methods for each code:
 1. k -NN: <https://github.com/yzhao062/pyod>
 2. OCSVM: <https://github.com/yzhao062/pyod>
 3. LOF: <https://github.com/yzhao062/pyod>
 4. IForest: <https://github.com/yzhao062/pyod>
 5. Deep-SVDD: <https://github.com/xuhongzuo/DeepOD>
 6. COPOD: <https://github.com/yzhao062/pyod>
 7. USAD: <https://github.com/xuhongzuo/DeepOD>
 8. GDN: <https://github.com/d-ailin/GDN>
 9. TcnED: <https://github.com/xuhongzuo/DeepOD>
 10. TranAD: <https://github.com/xuhongzuo/DeepOD>
 11. AnomalyTrans: <https://github.com/xuhongzuo/DeepOD>
 12. NCAD: <https://github.com/xuhongzuo/DeepOD>
 13. Deep IF: <https://github.com/xuhongzuo/DeepOD>
 14. TimesNet: <https://github.com/thuml/TimesNet>
 15. DCdetector: <https://github.com/xuhongzuo/DeepOD>
 16. COUTA: <https://github.com/xuhongzuo/DeepOD>
- **Computational Resources:** All experiments were conducted on a computing server equipped with an Intel(R) Xeon(R) Silver 4310 CPU and an NVIDIA H100 GPU (80GB memory).

C Algorithm Description and Complexity Analysis

We provide a detailed description of SPAGD’s training procedure in Algorithm 1. Additionally, we also analyze the time complexity of SPAGD here. Specifically, SPAGD contains three main components:

- **Self-Perturbation Module:** This module employs a Transformer-based model, where the computational bottleneck is the self-attention mechanism, which has a complexity of $\mathcal{O}(T^2 d_h)$ for an input sequence of length T and embedding dimension d_h of the Transformer. As the number of layers is a fixed hyperparameter, the complexity for this module is $\mathcal{O}(T^2 d_h)$.

Algorithm 1 Detailed training procedure of SPAGD

Input: Training set \mathcal{D} , window size, learning rate, number of neighbors K , number of chunks C , trade-off parameter β .

- 1: Initialize network parameters for each module of SPAGD.
 - 2: **while** not convergence **do**
 - 3: Sample a mini-batch $\{\mathbf{X}_i\}_{i=1}^{N_{\text{mini}}} \subset \mathcal{D}$.
 - 4: Obtain self-perturbed time series $\tilde{\mathbf{X}}_i = \text{Tran}_d(\text{Tran}_e(\mathbf{X}_i; \Theta_e); \Theta_d)$ via Eq. (1).
 - 5: Compute self-perturbation loss \mathcal{L}_{sp} via Eq. (2).
 - 6: Construct static graph A for normal time series via Eqs. (3) and (4).
 - 7: Derive reconstruction residual r via Eq. (5), and construct anomaly-aware graph \tilde{A} for the self-perturbed time series via Eq. (6).
 - 8: Obtain spatial features $\mathbf{H}^{(L)}$ and $\tilde{\mathbf{H}}^{(L)}$ via Eqs. (7) and (8).
 - 9: Partition the learned spatial features into C chunks, and obtain aggregated temporal features $\mathcal{Z}_T, \tilde{\mathcal{Z}}_T$ via Eqs. (9) and (10).
 - 10: Form $\mathcal{Z}_{\text{stack}} = [\mathcal{Z}_T; \tilde{\mathcal{Z}}_T]$ and predict $\hat{p} = \mathcal{P}(\mathcal{Z}_{\text{stack}}; \Theta_{\mathcal{P}})$ via Eq. (11).
 - 11: Compute anomaly detection loss \mathcal{L}_{ad} with pseudo-labels y via Eq. (12).
 - 12: Compute total loss \mathcal{L} via Eq. (13).
 - 13: Back-propagation and update all network parameters by gradient descent on \mathcal{L} .
 - 14: **end while**
-

- **AAGC Module:** The complexity of this module is dominated by the initial graph construction, which involves computing pairwise cosine similarity between all d variables over T timesteps. This results in a time complexity of $\mathcal{O}(d^2T)$. The subsequent dynamic adjustment steps involve calculating node-wise reconstruction residuals ($\mathcal{O}(dT)$) and adjusting the structure of the sparse graph ($\mathcal{O}(d^2)$), which are less complex.
- **Predictor Module:** This module consists of spatial convolution and temporal convolution. The spatial convolution operates on a sparse graph with $|E| = dK$ edges (due to top- K neighbor selection). In our model, the spatial convolution is applied across the temporal dimension T and L layers of GNN with a latent dimension of d_{lat} , resulting in complexity of $\mathcal{O}(LTdKd_{\text{lat}})$. Since L , K , and d_{lat} are fixed, this simplifies to $\mathcal{O}(Td)$. The temporal convolution complexity is also linear in sequence length and the number of channels, resulting in $\mathcal{O}(Td)$.

Thus, the overall time complexity of SPAGD for processing a single time-series window is approximately $\mathcal{O}(T^2d_h + d^2T)$, since T and d are dominating factors in the model. For practical deployment, the complexity of SPAGD is comparable to other transformer-based and GNN-based TSAD methods, such as TranAD [Tuli *et al.*, 2022], AnomalyTrans [Xu *et al.*, 2022], and TimesNet [Wu *et al.*, 2023].

D More Parameter Analysis

In this section, we further analyzed the impact of the additional two key hyperparameters on the anomaly detection performance, including the latent dimension and the number of GNN layers L .

Impact of Latent Dimension Figure 1 shows the anomaly detection performance under different latent dimensions, which range from $[8, 16, \dots, 512]$. The following observations can be made. Performance initially improved as the latent dimension increased (*e.g.*, from 8 or 32). This suggests that an appropriate dimensionality is crucial for the model to capture the intricate patterns and dependencies within the time-series data. However, a trend of performance degradation is observed when the latent dimension exceeds a certain value (*e.g.*, 64). This indicates that excessively large latent dimensions may incorporate noise information in the learned representations, thereby diminishing their discriminability and generalizability. Additionally, the optimal value of the latent dimension varies for each dataset due to its inherent characteristics, specifically 64 for SWaT and SMAP, and 32 for MSL. In general, SPAGD demonstrated stable performance under a wide range of latent dimensions.

Impact of Number of GNN Layers Figure 2 shows the experimental results under different numbers of GNN layers. Our experiments reveal that increasing the depth of the GNN (*i.e.*, L) does

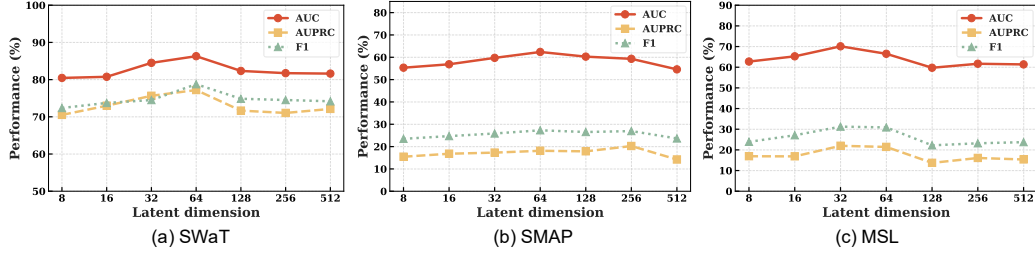


Figure 1: Anomaly detection performance (AUC, AUPRC, F1 in %) under different latent dimension values that range from $[8, 16, \dots, 512]$.

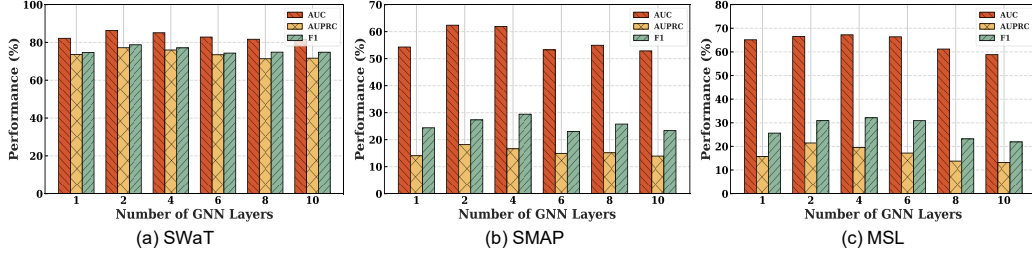


Figure 2: Anomaly detection performance (AUC, AUPRC, F1 in %) under different numbers of GNN layers that range from $[1, 2, 4, 6, 8, 10]$.

not monotonically enhance anomaly detection performance. We observed that promising results are typically achieved with $L = 2$ across all datasets. While a single layer might not be sufficient to capture complex relational information among variables, employing two layers appears to provide an adequate capacity for learning expressive node representations through message passing. In particular, increasing the value of L (e.g., $L = 3$ or $L = 4$) did not yield significant improvements and, in some cases, led to a slight degradation in performance (e.g., $L \geq 6$). This phenomenon is consistent with challenges often encountered in deeper GNN architectures. Firstly, deeper GNNs can suffer from “over-smoothing”, where repeated aggregation of neighbor information causes node representations to become indistinguishable, thereby losing information crucial for identifying anomalies. Secondly, a larger number of layers significantly increases model complexity and the number of parameters, which instead increases the risk of overfitting.

Impact of the Percentage of Anomalous Candidates We performed a sensitivity analysis of m on the SWaT datasets, where m controls the percentage of anomalous candidates in our AAGC module. Table 1 shows the anomaly detection performance under different values of m in the range of $[10\%, 50\%]$. The experimental results indicate that the performance of SPAGD is robust within a reasonable range of m (e.g., 20% to 50%). We can observe that the performance degrades when m is too small (e.g., 10%), as the AAGC module lacks a strong enough signal to dynamically adjust the graph structure. Conversely, if m is too large (e.g., 60%), the performance also slightly decreases as an excessive number of normal variables are incorrectly flagged as anomalous candidates, which introduces noise into the graph structure.

Table 1: Anomaly detection performance under different m .

m	10%	20%	30%	40%	50%	60%
AUC	80.74	83.22	86.30	83.30	83.32	81.58
AUPRC	71.00	72.45	77.20	71.82	73.13	71.39
F1	74.07	74.37	78.77	75.76	74.76	75.48

E More Visualization Results

This section presents extended visualization results to facilitate intuitive comparison of the detection capabilities between SPAGD and other baseline methods. Figures 3 and 4 show anomaly scores of SPAGD and various baseline methods on MSL, where we further supplement several baselines such as DeepSVDD, DeepIF, TimesNet, and COUTA (not shown in the main text). The results consistently demonstrated the superior ability of SPAGD to accurately identify anomalies. For example, the anomaly scores of SPAGD generally exhibit a stable and distinct response when encountering true anomalies, which closely reflect the ground truth segments. In contrast, the visual results of baseline methods, such as DeepSVDD, TimesNet, and COUTA, typically exhibit elevated false alarm rates, missed detections from insufficient or delayed responses to the anomalous events. These visualizations also reinforce the “anomaly reconstruction” problem inherent in many reconstruction-based models (*e.g.*, TranAD), where they generally fail to flag anomalous events. Compared to these models, SPAGD can identify and maintain high-confidence detections in these scenarios. With reference to the challenging case presented in Figure 4, all six baseline methods fail to identify the clear anomalous pattern, yet the anomaly scores of SPAGD successfully reflected its temporal evolution trend caused by anomalous events, which strongly demonstrates the effectiveness of our method. This visual evidence provides compelling qualitative support for the enhanced performance of SPAGD, which demonstrates its advanced practical detection capabilities and potential in complex real-world scenarios.

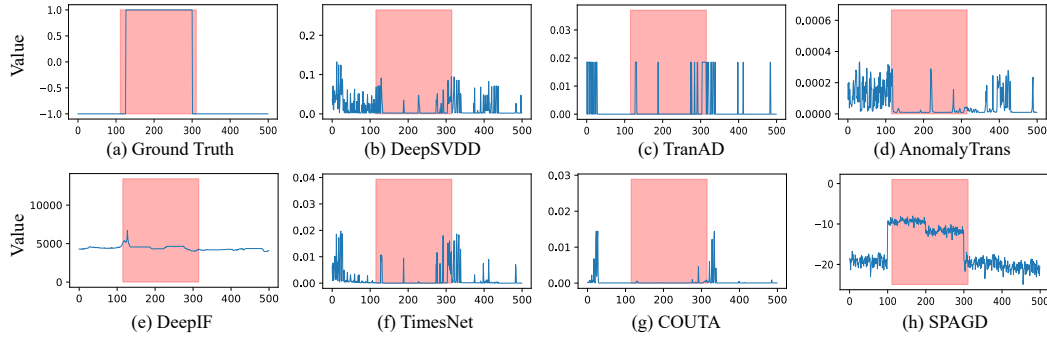


Figure 3: Anomaly score visualization on MSL, with the ground truth time series provided for reference. We randomly sampled data of length 500 for the comparison between SPAGD and several baselines. The red area indicates the ground truth anomaly.

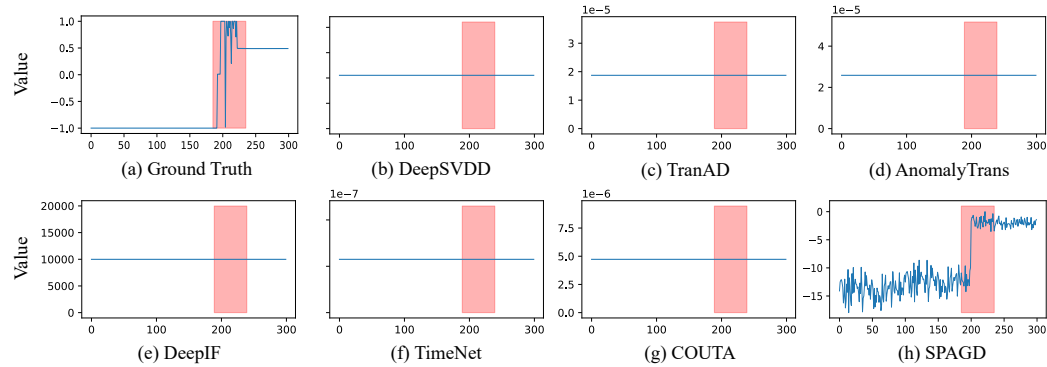


Figure 4: Anomaly score visualization on SMAP, with the ground truth time series provided for reference. We randomly sampled data of length 300 for the comparison between SPAGD and several baselines. The red area indicates the ground truth anomaly.

F Ablation Study of the GNN Backbone

The proposed SPAGD framework employs GAT [Veličković *et al.*, 2018] as the backbone network for the spatio-temporal anomaly detection module. To validate this choice, we conducted comparative experiments by substituting GAT with GCN [Kipf and Welling, 2017] and GraphSAGE [Hamilton *et al.*, 2017]. All other components of SPAGD remained unchanged, and the hyperparameters for GCN and GraphSAGE were kept consistent. Table 2 shows the experiment results, where we observe that GAT consistently outperforms GCN and GraphSAGE. The performance improvement in GAT can be attributed to its attention mechanism, which allows dynamic weighting of variable influence, which is advantageous for modeling changing correlations in anomalies, a characteristic our AAGC module captures and GAT leverages. Additionally, we also observed that GraphSAGE generally performs better than GCN due to better flexibility in aggregator functions, compared to the isotropic aggregation in GCN. Nonetheless, the performance of SPAGD remained relatively stable under different backbone networks, which also demonstrates its robustness.

Table 2: Ablation study results on the three datasets. The best results are marked in **bold**.

Dataset	SWaT			SMAP			MSL			Avg.
Metric	AUC	AUPRC	F1	AUC	AUPRC	F1	AUC	AUPRC	F1	
GCN	84.17	73.65	75.69	58.89	15.66	25.20	66.19	20.33	26.99	49.64
GraphSAGE	84.88	74.97	75.52	61.63	17.07	26.81	65.59	20.28	29.65	50.71
GAT	86.30	77.20	78.77	62.38	18.15	27.32	66.50	21.45	30.89	52.11

G Comparison with Random-Perturbation Strategy

To validate the rationale and effectiveness of the self-perturbed time series in facilitating anomaly detection, we compared SPAGD with a variant using a random-perturbation strategy, where synthetic anomalies were generated by adding random Gaussian noise to normal samples, and followed the same graph construction strategy as normal time-series data to build the relation graph. The generated anomalous time-series data are then used to train a classifier with the same architecture as SPAGD. Figure 5 illustrates the comparative results, which clearly show that SPAGD yielded significantly better anomaly detection performance across all datasets compared with the random-perturbation strategy. This performance gap highlights the limitations of using simple, structure-agnostic noise. Random Gaussian noise is context-unaware and fails to simulate realistic anomaly scenarios, which often involve complex, structured disruptions. Additionally, the random-perturbation strategy lacks adaptivity, whereas self-perturbation in SPAGD evolves as the reconstruction model improves, exposing the detector to a spectrum of deviations. More importantly, SPAGD benefits from joint optimization of self-perturbation and anomaly detection, allowing iterative refinement of self-perturbed anomalies based on feedback from the anomaly detection loss, thus producing diverse potential anomalous samples (from easy to hard) for training the anomaly detection model. This comparison demonstrates the rationale and effectiveness of the self-perturbation mechanism in our SPAGD.

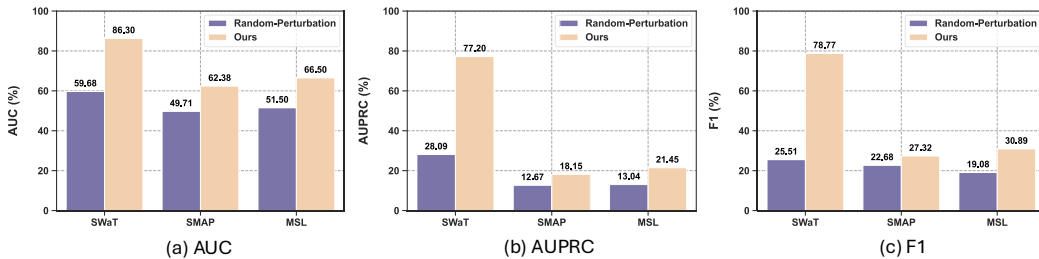


Figure 5: Performance comparison (AUC, AUPRC, and F1 in %) of the self-perturbation strategy and the random-perturbation strategy on SWaT, SMAP, and MSL.

H Comparison with Different Graph Construction Methods

The strategy for constructing the inter-variable graph is a critical architectural choice in GNN-based TSAD frameworks, as it defines the relational structure from which the model learns inter-variable dependencies. In this paper, we adopt a k -NN approach for the graph construction, which is motivated by its widespread use and proven efficacy in prior works [Deng and Hooi, 2021], as well as its ability to achieve a compelling balance between capturing salient local data structures and enforcing graph sparsity, which is crucial for computational efficiency and mitigating the effects of noisy correlations.

To empirically validate this design choice, we conducted a comparative analysis against two common alternatives: (1) a **fully-connected** strategy, where all pairwise similarities serve as edge weights, and (2) an ϵ -NN strategy, which connects nodes whose similarity score surpasses a predefined threshold ϵ . Table 3 summarizes the experimental results on the SWaT, SMAP, and MSL datasets. As shown in the table, the k -NN strategy consistently and significantly outperforms both alternatives across all datasets. The fully-connected approach, while theoretically comprehensive, consistently underperforms. This is because it forces the model to aggregate information over a dense graph, where signals from genuinely correlated variables are diluted by the noise from numerous weak or irrelevant connections. Similarly, the ϵ -NN strategy proves to be a suboptimal choice. Its primary vulnerability is its high sensitivity to the threshold ϵ , a hyperparameter that is difficult to generalize across datasets with varying sensor dynamics and data distributions. An improperly calibrated threshold can lead to a graph that is either too fragmented (missing critical dependencies) or overly dense (reintroducing noise). The unstable performance of ϵ -NN across datasets highlights this inherent lack of robustness. In contrast, by ensuring a consistent number of neighbors for each node, the k -NN strategy provides a more stable and effective graph structure, leading to superior performance across all datasets.

Table 3: Anomaly detection performance under different graph construction methods. The best results are marked in **bold**.

Dataset	SWaT			SMAP			MSL		
Metric	AUC	AUPRC	F1	AUC	AUPRC	F1	AUC	AUPRC	F1
Fully-connected	82.13	71.79	74.45	50.45	13.10	22.68	56.56	12.34	20.79
ϵ -NN	81.41	72.53	74.35	47.07	11.64	22.68	60.42	15.71	22.71
k -NN	86.30	77.20	78.77	62.38	18.15	27.32	66.50	21.45	30.89

I Comparison with Point-Adjusted Results

In this section, we further provide the point-adjusted (PA) results for the proposed SPAGD method and other baseline methods, as summarized in Table 4. It is important to note that point-adjusted metrics may lead to **overestimation** of anomaly detection performance [Wu and Keogh, 2021; Kim *et al.*, 2022; Liu and Paparrizos, 2024], as they can artificially inflate detection accuracy by allowing a time-tolerance window around true anomalies. Despite this, PA results are still commonly reported in recent TSAD literature [Xu *et al.*, 2022; Tuli *et al.*, 2022; Yang *et al.*, 2023; Xu *et al.*, 2024]. Hence, we include PA results here solely for comparative reference. We can observe that SPAGD consistently outperforms all baselines by notable margins across AUC, AUPRC, and F1-score under the PA condition, which highlights its capability to precisely localize anomalies within permissible tolerance windows. However, we would like to **emphasize the necessity of caution when interpreting these PA metrics and advocate for using stricter, non-adjusted metrics** for a fairer and more realistic evaluation (*i.e.*, Table 2 in our main text) of anomaly detection methods.

J Performance Evaluation via VUS-ROC and VUS-PR

To provide a more robust assessment of anomaly detection performance, we extend our evaluation beyond conventional metrics by incorporating the Volume Under the Surface for ROC (VUS-ROC) and PR (VUS-PR) [Paparrizos *et al.*, 2022; Boniol *et al.*, 2025]. Unlike traditional point-based metrics, which are relatively more sensitive to threshold selection and may not adequately capture the continuous nature of time-series anomalies, the VUS metrics provide a more holistic, threshold-independent

Table 4: Point-adjusted anomaly detection performance in terms of AUC, AUPRC, and F1 (in %). Note that the best results are marked in **bold**.

Model	SWaT			SMAP			MSL			Avg.
	AUC	AUPRC	F1	AUC	AUPRC	F1	AUC	AUPRC	F1	
<i>k</i> -NN [Ramaswamy <i>et al.</i> , 2000]	76.11	69.38	71.86	78.09	52.80	57.83	74.38	28.96	44.33	59.30
OCSVM [Schölkopf <i>et al.</i> , 2001]	74.51	67.85	70.77	76.63	51.98	56.38	76.40	27.86	46.19	58.77
LOF [He <i>et al.</i> , 2003]	78.21	67.96	68.05	87.44	78.10	83.07	87.44	66.96	70.88	75.35
IForest [Liu <i>et al.</i> , 2008]	88.42	77.21	70.20	77.20	54.46	59.24	81.67	34.80	49.89	61.45
Deep-SVDD [Ruff <i>et al.</i> , 2018]	96.47	67.59	80.70	81.62	65.71	69.78	97.89	85.00	82.41	80.81
COPOD [Li <i>et al.</i> , 2020]	81.57	72.91	71.06	76.71	54.49	56.03	76.28	28.99	47.05	59.90
USAD [Audibert <i>et al.</i> , 2020]	84.47	78.05	79.77	65.54	50.27	59.37	75.68	24.60	39.78	59.78
GDN [Deng and Hooi, 2021]	96.63	92.94	88.78	89.10	71.24	70.66	96.42	71.53	70.19	83.05
TcnED [Garg <i>et al.</i> , 2021]	87.89	82.52	85.10	77.92	63.89	70.54	81.15	25.69	43.94	66.74
TranAD [Tuli <i>et al.</i> , 2022]	86.75	79.82	82.18	69.82	64.08	71.34	90.38	38.21	55.08	69.76
AnomalyTrans [Xu <i>et al.</i> , 2022]	87.17	78.10	80.47	67.61	62.13	70.74	98.49	89.38	88.52	80.29
NCAD [Carmona <i>et al.</i> , 2022]	96.98	87.30	86.68	68.17	62.98	69.64	95.56	81.40	81.77	81.16
Deep IF [Xu <i>et al.</i> , 2023]	85.66	79.03	79.58	83.96	56.63	61.51	81.36	26.34	45.97	62.23
TimesNet [Wu <i>et al.</i> , 2023]	98.08	94.00	88.06	69.83	64.06	71.59	97.38	88.79	87.85	83.52
DCdetector [Yang <i>et al.</i> , 2023]	96.47	67.59	80.70	77.63	52.83	67.51	86.19	73.38	83.07	76.14
COUTA [Xu <i>et al.</i> , 2024]	89.94	85.85	88.55	76.55	63.09	69.89	97.21	90.61	89.01	82.31
SPAGD	99.64	97.46	92.83	94.56	82.49	77.34	99.63	96.58	91.78	92.48

Table 5: Anomaly detection performance in terms of VUS-ROC and VUS-PR on SWaT and MSL. The best results are marked in **bold**.

Dataset	SWaT		MSL	
Metric	VUS-ROC	VUS-PR	VUS-ROC	VUS-PR
USAD [Audibert <i>et al.</i> , 2020]	53.78	32.86	58.09	16.29
TcnED [Garg <i>et al.</i> , 2021]	62.99	44.30	58.22	13.62
TranAD [Tuli <i>et al.</i> , 2022]	60.10	39.29	49.62	12.06
AnomalyTrans [Xu <i>et al.</i> , 2022]	57.54	36.02	59.30	14.23
NCAD [Carmona <i>et al.</i> , 2022]	44.42	13.84	57.70	13.36
Deep IF [Xu <i>et al.</i> , 2023]	56.48	37.56	58.39	13.67
TimesNet [Wu <i>et al.</i> , 2023]	44.49	21.74	61.49	14.88
DCdetector [Yang <i>et al.</i> , 2023]	52.31	15.01	50.68	12.64
COUTA [Xu <i>et al.</i> , 2024]	77.10	51.86	55.76	14.07
SPAGD	84.93	62.08	60.78	16.48

measure by evaluating the entire area under the surface of curve. This makes it particularly well-suited for a rigorous comparison of model capabilities in real-world scenarios.

Table 5 presents the comparative performance of SPAGD against competitive baselines on the SWaT and MSL datasets. The results provide compelling evidence for the effectiveness of our SPAGD method. For example, on the SWaT dataset, SPAGD achieved a VUS-ROC of 84.93% and a VUS-PR of 62.08%, which demonstrates a substantial margin of improvement over other competitive baselines such as COUTA (77.10% VUS-ROC and 51.86% VUS-PR, respectively). In summary, the evaluation under the VUS metrics further substantiates the effectiveness of SPAGD. This strengthens our claim that the synergy between the self-perturbation mechanism and the anomaly-aware dynamic graph construction allows SPAGD to learn more discriminative representations for TSAD.

K Experimental Statistical Significance Analysis

To validate that the performance gains of SPAGD are statistically meaningful, we conduct a statistical significance analysis. Specifically, we perform the Student’s t-test to compare SPAGD against several strong baselines on the SWaT dataset, where the performance improvements are generally considered to be statistically significant if the p -value of the Student’s t-test $p \leq 0.05$. The analysis was based on the results from five independent runs for each method, where AUC and AUPRC are used as the evaluation metrics.

Over the five runs, our proposed SPAGD achieved a mean AUC of $85.48\% \pm 1.15\%$ and a mean AUPRC of $75.67\% \pm 1.64\%$. Table 6 shows the performance of the baseline methods and the corresponding p -values from the Student’s t-test against SPAGD. We can observe from the table that in each comparison for both AUC and AUPRC, the calculated p -value is substantially lower than the 0.05 threshold. This allows us to reject the null hypothesis—that the performance differences between SPAGD and the baseline methods stem from chance—with high confidence. Overall, the consistently low p -values provide strong statistical evidence to validate its robustness and effectiveness.

Table 6: Student’s t-test results on the SWaT dataset based on five independent runs. The performance of SPAGD is compared against each baseline, with the resulting p -values reported.

Comparison	AUC	p -value	AUPRC	p -value
SPAGD vs. Deep-SVDD	82.48 ± 0.36	0.0005	72.76 ± 0.94	0.0089
SPAGD vs. COUTA	81.43 ± 0.97	0.0003	72.39 ± 1.41	0.0095
SPAGD vs. TcnED	82.18 ± 0.90	0.0010	72.33 ± 1.14	0.0057
SPAGD vs. TranAD	81.77 ± 0.46	0.0002	71.86 ± 0.58	0.0012
SPAGD vs. Deep IF	80.63 ± 0.12	< 0.0001	70.24 ± 0.78	0.0002

L Theoretical Analysis

This appendix presents a theoretical analysis of the key components of the proposed SPAGD framework. Specifically, we aim to theoretically validate

1. **The rationale of self-perturbed samples as proxies for real anomalies.**
2. **The stability of graph representation learning under the dynamic graph adjustments introduced by our anomaly-aware graph construction module.**

These results will provide deeper insights into the reliability of SPAGD.

Notation. We denote the spectral norm by $\|\cdot\|_2$ and the Frobenius norm by $\|\cdot\|_F$. A normal time-series window is $X \sim \mathbb{P}_{\text{norm}}$. The self-perturbed sample is $\tilde{X} = X + \epsilon_\theta(X)$, where $\epsilon_\theta(X)$ is the perturbation. Let A be the (unnormalized) adjacency matrix and let its symmetrically normalized version be $\hat{A} := D^{-1/2}AD^{-1/2}$, where D is the degree matrix. Throughout the analysis, we use the standard fact $\|\hat{A}\|_2 \leq 1$. For theoretical simplicity, we analyze a GCN-like message-passing layer.

L.1 Distributional Proximity of Self-Perturbed Time Series

To theoretically demonstrate the ability of our self-perturbation mechanism to generate meaningful anomalous samples, we first formalize the characteristics of real and self-perturbed anomalies. The following theorem aims to show that the distribution of self-perturbed samples remains in a bounded vicinity of the real anomaly distribution.

Assumption 1. We assume that real anomalies \tilde{X} are formed by adding a noise component η to normal data X . This noise η is drawn from a distribution $\mathbb{P}_{\text{noise}}$ and is independent of X . Furthermore, we assume that there exists a known maximum magnitude for this anomalous noise, such that $\|\eta\|_F \leq \rho_{\max}$ almost surely. This boundedness is a common assumption [Aggarwal, 2017; Goodfellow *et al.*, 2014] for tractability in theoretical analyses of anomaly detection. We define the expected magnitude of real anomalous noise as $\rho_{\text{anom}} := \mathbb{E}[\|\eta\|_F] < \infty$.

Assumption 2. The self-perturbation module generates a reconstruction error $\epsilon_\theta(X) := \mathcal{R}_\theta(X) - X$, where $\mathcal{R}_\theta(X)$ is the output of the reconstruction network. We assume that the magnitude of this generated perturbation is also bounded by ρ_{\max} , i.e., $\|\epsilon_\theta(X)\|_F \leq \rho_{\max}$ almost surely. This ensures that self-perturbed samples do not exhibit magnitudes exceeding those of plausible real anomalies, which is a desired characteristic for generating meaningful auxiliary data. The expected magnitude of these self-perturbations is denoted by $\rho_\theta := \mathbb{E}[\|\epsilon_\theta(X)\|_F] < \infty$. Let $\mathbb{P}_{\tilde{X}}$ and $\mathbb{P}_{\bar{X}}$ denote the distributions of $\tilde{X} := X + \epsilon_\theta(X)$ and $\bar{X} := X + \eta$, respectively.

These assumptions allow us to quantify the closeness between the distribution of self-perturbed samples and that of real anomalies using the 1-Wasserstein distance [Villani and others, 2008; Peyré *et al.*, 2019].

Theorem 1. *Under Assumptions 1 and 2, the 1-Wasserstein distance between the distribution of self-perturbed samples \tilde{X} and real anomalies \bar{X} , i.e., $\mathbb{P}_{\tilde{X}}$ and $\mathbb{P}_{\bar{X}}$, is bounded as:*

$$W_1(\mathbb{P}_{\tilde{X}}, \mathbb{P}_{\bar{X}}) \leq \rho_\theta + \rho_{anom}. \quad (1)$$

Moreover, if the training objective leads to a monotonic decrease in the expected perturbation magnitude $\rho_\theta^{(t)}$ over training iterations t , then the derived upper bound $\rho_\theta^{(t)} + \rho_{anom}$ also monotonically decreases, suggesting a progressively tighter bound.

Proof. To establish the bound on the Wasserstein distance, we employ a specific coupling of the distributions $\mathbb{P}_{\tilde{X}}$ and $\mathbb{P}_{\bar{X}}$. Let X be drawn from \mathbb{P}_{norm} and η from $\mathbb{P}_{\text{noise}}$ independently, we define the joint distribution $\pi(x, \eta) = \mathbb{P}_{\text{norm}}(x)\mathbb{P}_{\text{noise}}(\eta)$. Using a single draw of X and η , we construct a self-perturbed sample $\tilde{x} = x + \epsilon_\theta(x)$ and a real anomaly $\bar{x} = x + \eta$. The pair (\tilde{x}, \bar{x}) constructed this way has marginals $\mathbb{P}_{\tilde{X}}$ and $\mathbb{P}_{\bar{X}}$ respectively.

The 1-Wasserstein distance $W_1(\mathbb{P}_{\tilde{X}}, \mathbb{P}_{\bar{X}})$ is defined as the infimum of $\mathbb{E}[||\tilde{X} - \bar{X}||_F]$ over all couplings of (\tilde{X}, \bar{X}) with marginals $\mathbb{P}_{\tilde{X}}$ and $\mathbb{P}_{\bar{X}}$. Our chosen coupling provides one such joint distribution, so:

$$\begin{aligned} W_1(\mathbb{P}_{\tilde{X}}, \mathbb{P}_{\bar{X}}) &\leq \mathbb{E}_\pi[||\tilde{x} - \bar{x}||_F] \\ &= \mathbb{E}_{X, \eta}[||(X + \epsilon_\theta(X)) - (X + \eta)||_F] \\ &= \mathbb{E}_{X, \eta}[||\epsilon_\theta(X) - \eta||_F] \\ &\leq \mathbb{E}_{X, \eta}[||\epsilon_\theta(X)||_F + ||\eta||_F] \\ &= \mathbb{E}_X[||\epsilon_\theta(X)||_F] + \mathbb{E}_\eta[||\eta||_F] \\ &= \rho_\theta + \rho_{anom}. \end{aligned} \quad (2)$$

The final inequality follows from the triangle inequality for the Frobenius norm (equivalently, the Euclidean norm after vectorization). This completes the proof. \square

This theorem suggests that the “quality” of self-perturbed samples, in terms of their distributional proximity to real anomalies, is influenced by the sum of average perturbation magnitudes. As the reconstruction model improves (reducing ρ_θ), this upper bound on the distance tightens, implying that the self-perturbed samples can become increasingly representative of true anomalies, thus providing more effective auxiliary signals for training.

Remark 1. It is important to note that Theorem 1 provides a *tightening upper bound* provided ρ_θ decreases, but it does not claim $W_1(\mathbb{P}_{\tilde{X}}, \mathbb{P}_{\bar{X}})$ itself decreases monotonically. When the reconstruction objective is ℓ_2 -type (or a valid upper bound thereof), $\rho_\theta = \mathbb{E}[||\epsilon_\theta(X)||_F]$ typically follows the training loss trend, hence the bound tightens in practice.

L.2 Robustness of Dynamic Graph Adjustment

This subsection provides a theoretical analysis of the stability of the GNN representations under small perturbations to the graph structure, which simulates the effect of the anomaly-aware adjacency matrix. The study of GNN stability and robustness, particularly under structural or feature perturbations, is an important research issue [Zügner *et al.*, 2018]. Our analysis in this context aims to show that the GNN outputs used in SPAGD remain stable as the graph structures dynamically evolve under our proposed mechanism.

Assumption 3. Let $\tilde{A} = \hat{A} + \Delta$ denote the effective propagation operator obtained from the anomaly-aware graph, where Δ represents the perturbation. We assume this perturbation is bounded in spectral norm [Horn and Johnson, 2012]: $||\Delta||_2 \leq \tau$. For the GNN layers, we assume that the weight matrix $W^{(l)}$ of any layer l is bounded in spectral norm, i.e., $||W^{(l)}||_2 \leq c_w$. The activation function σ is assumed to be L_σ -Lipschitz continuous [Fazlyab *et al.*, 2019; Gouk *et al.*, 2021]. We define a composite factor $\gamma := L_\sigma c_w$. A critical assumption for the stability of representations is:

$$\alpha := \gamma(1 + \tau) < 1. \quad (3)$$

This condition implies that the combined effect of layer transformations and graph perturbation does not excessively amplify differences across layers.

Definition 1. For any adjacency matrix \hat{A} and input hidden representation H , a GNN layer's transformation [Kipf and Welling, 2017; Veličković *et al.*, 2018] is defined as $\Phi_{\hat{A}}(H) := \sigma(\hat{A}HW^{(l)})$, where $W^{(l)}$ is the weight matrix for the current layer l . Let $H^{(0)} = \tilde{H}^{(0)} = X$ be the initial representations (input features). The subsequent layer representations are computed iteratively for $l = 0, \dots, L - 1$:

$$H^{(l+1)} = \Phi_{\hat{A}}(H^{(l)}), \quad (4)$$

$$\tilde{H}^{(l+1)} = \Phi_{\tilde{A}}(\tilde{H}^{(l)}). \quad (5)$$

The following theorem bounds the accumulation of differences in layer representations due to the graph perturbation Δ .

Theorem 2. Under Assumption 3, let $\Delta^{(l)} := \tilde{H}^{(l)} - H^{(l)}$ be the difference between the hidden representations at layer l for the anomaly-aware graph and the original graph. Then, for any number of layers $L \geq 1$, the Frobenius norm of this difference at the final layer is bounded by:

$$\|\tilde{H}^{(L)} - H^{(L)}\|_F \leq \tau \|X\|_F \frac{1 - \alpha^L}{1 - \alpha}. \quad (6)$$

Furthermore, if the residual weighting mechanism (which forms Δ) modifies at most a fraction p of nodes, and each added weight (entry in Δ) is at most δ_{\max} in magnitude, then it can be shown (e.g., using Gershgorin disc arguments [Horn and Johnson, 2012] for specific perturbation structures) that $\tau \leq \kappa \delta_{\max}$. In such cases, the bound in (6) is linear in both this sparsity factor κ and the maximum perturbation weight δ_{\max} .

Proof. Let $\Delta^{(l)} = \tilde{H}^{(l)} - H^{(l)}$ be the difference in representations at layer l . The difference at the $(l + 1)$ -th layer is $\Delta^{(l+1)} = \sigma(\tilde{A}\tilde{H}^{(l)}W^{(l)}) - \sigma(\hat{A}H^{(l)}W^{(l)})$. Using the L_σ -Lipschitz continuity of σ , the sub-multiplicativity of the Frobenius norm, the property $\|ABC\|_F \leq \|A\|_2\|B\|_F\|C\|_2$, and the bounds from Assumption 3 (specifically $\|\hat{A}\|_2 \leq 1$, which is standard for normalized adjacency matrices, and $\|W^{(l)}\|_2 \leq c_w$):

$$\begin{aligned} \|\Delta^{(l+1)}\|_F &= \|\sigma(\tilde{A}\tilde{H}^{(l)}W^{(l)}) - \sigma(\hat{A}H^{(l)}W^{(l)})\|_F \\ &\leq L_\sigma \|\tilde{A}\tilde{H}^{(l)}W^{(l)} - \hat{A}H^{(l)}W^{(l)}\|_F \\ &= L_\sigma \|(\hat{A} + \Delta)\tilde{H}^{(l)}W^{(l)} - \hat{A}H^{(l)}W^{(l)}\|_F \\ &= L_\sigma \|\hat{A}(\tilde{H}^{(l)} - H^{(l)})W^{(l)} + \Delta\tilde{H}^{(l)}W^{(l)}\|_F \\ &\leq L_\sigma (\|\hat{A}(\tilde{H}^{(l)} - H^{(l)})W^{(l)}\|_F + \|\Delta\tilde{H}^{(l)}W^{(l)}\|_F) \\ &\leq L_\sigma (\|\hat{A}\|_2 \|\Delta^{(l)}\|_F \|W^{(l)}\|_2 + \|\Delta\|_2 \|\tilde{H}^{(l)}\|_F \|W^{(l)}\|_2) \\ &\leq L_\sigma c_w (\|\Delta^{(l)}\|_F + \tau \|\tilde{H}^{(l)}\|_F) \\ &= \gamma (\|\Delta^{(l)}\|_F + \tau \|\tilde{H}^{(l)}\|_F). \end{aligned} \quad (7)$$

Next, we bound $\|\tilde{H}^{(l)}\|_F$. By induction, one can show that $\|H^{(l)}\|_F \leq \gamma^l \|X\|_F$ because $\|\hat{A}\|_2 \leq 1$, σ is L_σ -Lipschitz, and $\|W^{(k)}\|_2 \leq c_w$ for each layer $k < l$. Therefore, using the triangle inequality:

$$\|\tilde{H}^{(l)}\|_F = \|H^{(l)} + \Delta^{(l)}\|_F \leq \|H^{(l)}\|_F + \|\Delta^{(l)}\|_F \leq \gamma^l \|X\|_F + \|\Delta^{(l)}\|_F.$$

Substituting this into inequality (7):

$$\begin{aligned} \|\Delta^{(l+1)}\|_F &\leq \gamma (\|\Delta^{(l)}\|_F + \tau (\gamma^l \|X\|_F + \|\Delta^{(l)}\|_F)) \\ &= \gamma (1 + \tau) \|\Delta^{(l)}\|_F + \gamma^{(l+1)} \tau \|X\|_F \\ &= \alpha \|\Delta^{(l)}\|_F + \gamma^{(l+1)} \tau \|X\|_F, \end{aligned} \quad (8)$$

where $\alpha = \gamma(1 + \tau)$ as defined in Assumption 3.

To unroll this recurrence, starting from $\Delta^{(0)} = \tilde{H}^{(0)} - H^{(0)} = X - X = 0$. Assuming $\gamma \leq 1$ (which is common, e.g., if $L_\sigma = 1$ for ReLU and $c_w \leq 1$, as noted in Remark 2), then for $l \geq 0$, $\gamma^{(l+1)} \leq 1$. Thus, inequality (8) can be further bounded by:

$$\|\Delta^{(l+1)}\|_F \leq \alpha \|\Delta^{(l)}\|_F + \tau \|X\|_F.$$

Unrolling this simplified recurrence:

$$\begin{aligned} \|\Delta^{(L)}\|_F &\leq \alpha \|\Delta^{(L-1)}\|_F + \tau \|X\|_F \\ &\leq \alpha(\alpha \|\Delta^{(L-2)}\|_F + \tau \|X\|_F) + \tau \|X\|_F \\ &= \alpha^2 \|\Delta^{(L-2)}\|_F + (\alpha + 1)\tau \|X\|_F \\ &\dots \\ &\leq \alpha^L \|\Delta^{(0)}\|_F + \tau \|X\|_F \sum_{k=0}^{L-1} \alpha^k, \end{aligned} \tag{9}$$

Since $\Delta^{(0)} = 0$:

$$\begin{aligned} \|\Delta^{(L)}\|_F &= \tau \|X\|_F \sum_{k=0}^{L-1} \alpha^k \\ &= \tau \|X\|_F \frac{1 - \alpha^L}{1 - \alpha}. \end{aligned} \tag{10}$$

This yields the bound in Eq. (6). Finally, the assertion that $\tau \leq \kappa \delta_{\max}$ under conditions of sparse modification (at most κN rows/columns affected, with individual changes $\leq \delta_{\max}$) relies on results such as those derived from Gershgorin circle theorem [Horn and Johnson, 2012] for structured sparse matrices. This establishes the linear dependence of the error bound on κ and δ_{\max} via τ . \square

This theorem provides assurance that the dynamic adjustments to the graph structure, if sufficiently small (controlled τ) and if the GNN layers are well-behaved (leading to $\alpha < 1$), will not lead to an uncontrolled divergence of representations. The bound indicates that the representational difference scales gracefully with the magnitude of graph perturbation τ and the number of layers L .

Corollary 1. *Under the conditions of Theorem 2, unrolling (8) yields, for $\alpha \neq \gamma$,*

$$\|\tilde{H}^{(L)} - H^{(L)}\|_F \leq \tau \|X\|_F \cdot \frac{\gamma(\alpha^L - \gamma^L)}{\alpha - \gamma}. \tag{11}$$

If $\alpha = \gamma$, then $\|\tilde{H}^{(L)} - H^{(L)}\|_F \leq L \gamma^L \tau \|X\|_F$. When additionally $\gamma \leq 1$ and $\alpha < 1$, this reduces to the bound in (6).

Remark 2. The condition $\alpha = \gamma(1 + \tau) < 1$ from Assumption 3 is crucial to ensure that error accumulation remains bounded with depth, where $\gamma := L_\sigma c_w$ is the per-layer Lipschitz factor and $\tau := \|\Delta\|_2$ measures the structural deviation. In common GNN settings (e.g., ReLU with $L_\sigma = 1$ and spectral control $\|W^{(l)}\|_2 \leq c_w \leq 1$), one typically has $\gamma \leq 1$. For $\gamma > 0$, the *necessary and sufficient* condition for $\alpha < 1$ is $\tau < (1 - \gamma)/\gamma$; a more conservative *sufficient* condition is $\tau < 1 - \gamma$, which avoids dividing by γ and still guarantees $\alpha < 1$ whenever $\gamma < 1$. Since the final stability bound scales as $(1 - \alpha^L)/(1 - \alpha)$ (cf. Eq. (6)) and thus increases with both α and the depth L , keeping α comfortably below 1 is practically important. For instance, if $\gamma \leq \frac{1}{2}$ then any $\tau < \frac{1}{2}$ ensures $\alpha < 1$ and stabilizes deeper stacks.

References

- Charu C Aggarwal. *An introduction to outlier analysis*. Springer, 2017.
- Julien Audibert, Pietro Michiardi, Frédéric Guyard, Sébastien Marti, and Maria A Zuluaga. Usad: Unsupervised anomaly detection on multivariate time series. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 3395–3404, 2020.

- Paul Boniol, Ashwin K Krishna, Marine Bruel, Qinghua Liu, Mingyi Huang, Themis Palpanas, Ruey S Tsay, Aaron Elmore, Michael J Franklin, and John Paparrizos. Vus: effective and efficient accuracy measures for time-series anomaly detection. *The VLDB Journal*, 34(3):32, 2025.
- Chris U Carmona, François-Xavier Aubet, Valentin Flunkert, and Jan Gasthaus. Neural contextual anomaly detection for time series. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 2022.
- 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, pages 4027–4035, 2021.
- Mahyar Fazlyab, Alexander Robey, Hamed Hassani, Manfred Morari, and George Pappas. Efficient and accurate estimation of lipschitz constants for deep neural networks. *Advances in Neural Information Processing Systems*, 32, 2019.
- Astha Garg, Wenyu Zhang, Jules Samaran, Ramasamy Savitha, and Chuan-Sheng Foo. An evaluation of anomaly detection and diagnosis in multivariate time series. *IEEE Transactions on Neural Networks and Learning Systems*, 33(6):2508–2517, 2021.
- Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.
- Henry Gouk, Eibe Frank, Bernhard Pfahringer, and Michael J Cree. Regularisation of neural networks by enforcing lipschitz continuity. *Machine Learning*, 110:393–416, 2021.
- Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in Neural Information Processing Systems*, 30, 2017.
- Zengyou He, Xiaofei Xu, and Shengchun Deng. Discovering cluster-based local outliers. *Pattern Recognition Letters*, 24(9-10):1641–1650, 2003.
- Roger A Horn and Charles R Johnson. *Matrix analysis*. Cambridge university press, 2012.
- Kyle Hundman, Valentino Constantinou, Christopher Laporte, Ian Colwell, and Tom Soderstrom. Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 387–395, 2018.
- 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, pages 7194–7201, 2022.
- Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *Proceedings of the International Conference on Learning Representations*, 2017.
- Zheng Li, Yue Zhao, Nicola Botta, Cezar Ionescu, and Xiyang Hu. Copod: copula-based outlier detection. In *Proceedings of the IEEE International Conference on Data Mining*, pages 1118–1123. IEEE, 2020.
- 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.
- Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. In *Proceedings of the IEEE International Conference on Data Mining*, pages 413–422. IEEE, 2008.
- Aditya P Mathur and Nils Ole Tippenhauer. Swat: A water treatment testbed for research and training on ics security. In *Proceedings of the International Workshop on Cyber-Physical Systems for Smart Water Networks*, pages 31–36. IEEE, 2016.

- John Paparrizos, Paul Boniol, Themis Palpanas, Ruey S Tsay, Aaron Elmore, and Michael J Franklin. Volume under the surface: a new accuracy evaluation measure for time-series anomaly detection. *Proceedings of the VLDB Endowment*, 15(11):2774–2787, 2022.
- Gabriel Peyré, Marco Cuturi, et al. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019.
- Sridhar Ramaswamy, Rajeev Rastogi, and Kyuseok Shim. Efficient algorithms for mining outliers from large data sets. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 427–438, 2000.
- Lukas Ruff, Robert Vandermeulen, Nico Goernitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, Emmanuel Müller, and Marius Kloft. Deep one-class classification. In *Proceedings of the International Conference on Machine Learning*, pages 4393–4402. PMLR, 2018.
- Bernhard Schölkopf, John C Platt, John Shawe-Taylor, Alex J Smola, and Robert C Williamson. Estimating the support of a high-dimensional distribution. *Neural Computation*, 13(7):1443–1471, 2001.
- Shreshth Tuli, Giuliano Casale, and Nicholas R Jennings. Tranad: deep transformer networks for anomaly detection in multivariate time series data. *Proceedings of the VLDB Endowment*, 15(6):1201–1214, 2022.
- Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *Proceedings of the International Conference on Learning Representations*, 2018.
- Cédric Villani et al. *Optimal transport: old and new*, volume 338. Springer, 2008.
- Renjie Wu and Eamonn J Keogh. 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.
- Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. Timesnet: Temporal 2d-variation modeling for general time series analysis. In *Proceedings of the International Conference on Learning Representations*, 2023.
- Jiehui Xu, Haixu Wu, Jianmin Wang, and Mingsheng Long. Anomaly transformer: Time series anomaly detection with association discrepancy. In *Proceedings of the International Conference on Learning Representations*, 2022.
- Hongzuo Xu, Guansong Pang, Yijie Wang, and Yongjun Wang. Deep isolation forest for anomaly detection. *IEEE Transactions on Knowledge and Data Engineering*, 35(12):12591–12604, 2023.
- Hongzuo Xu, Yijie Wang, Songlei Jian, Qing Liao, Yongjun Wang, and Guansong Pang. Calibrated one-class classification for unsupervised time series anomaly detection. *IEEE Transactions on Knowledge and Data Engineering*, 2024.
- 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 ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 3033–3045, 2023.
- Daniel Zügner, Amir Akbarnejad, and Stephan Günnemann. Adversarial attacks on neural networks for graph data. In *Proceedings of the ACM SIGKDD international Conference on Knowledge Discovery & Data Mining*, pages 2847–2856, 2018.