GRAPHS ON STAGE FOR INTERPRETABLE TIME SERIES ANOMALY DETECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Multivariate time series monitoring is critical for safeguarding complex systems, where anomalies typically stem from interactions among interdependent dimensions. However, current detection paradigms face an inherent trade-off: traditional graph methods offer interpretability, but lag in empirical performance, while modern deep learning models provide high expressiveness at the cost of speed and transparency. We present STAGE, a self-supervised framework combining representation learning with explicit relational structure. STAGE models the time series dimensions as vertices in a dynamic graph, pruning edges via a multi-head attention mechanism to isolate genuine cross-channel dependencies. For explainability, STAGE derives instantaneous, post-hoc attributions by occluding individual dimension nodes and observing the shift in anomaly scores, requiring zero retraining. On the largest benchmark, TSB-AD-M, STAGE achieves a VUS-PR of 0.44, matching the best baseline (out of 30) and outperforming the runner-up by 12%. Crucially, it matches both the detection accuracy and diagnostic interpretability of state-of-the-art methods, which cannot provide real-time explanations.

1 INTRODUCTION

Multivariate time series capture complex sequential dynamics across interconnected systems, where individual channels track distinct sensors, transactions, or system components over time. Detecting anomalies in these environments is essential to preempt system-wide failures, diagnose operational drift, and flag malicious activity Blázquez-García et al. (2022).

Consequently, recent years have witnessed a surge in Multivariate Time Series Anomaly Detection (MTSAD) methods that model the entire system. The most competitive approaches rely on deep neural architectures trained under unsupervised objectives, most commonly using reconstruction or forecasting error Su et al. (2019); Zhao et al. (2020); Özer et al. (2026). Despite their strong performance, these models remain black boxes where diagnosing why a failure occurred or pinpointing the culprit variables and their compromised inter-dependencies remains notoriously difficult. On the other hand, graph-based approaches Boniol & Palpanas (2020); Graf et al. (2025); Deng & Hooi (2021) that explicitly model inter-dependencies via relational edges offer intuitive structural interpretability, yet they often fall short on accuracy compared to unconstrained deep models.

However, existing explainable MTSAD methods introduce a severe and largely overlooked operational bottleneck: **the computational cost of explanation**. In safety-critical and high-throughput systems, detecting an anomaly is only the first step; domain experts must immediately examine the alert and isolate the root cause. Yet, existing state-of-the-art interpretable and attribution methods require extensive training runs and incur minutes of inference latency per window to compute attributions (e.g., via perturbation or gradient rollouts) rendering these methods impractical.

In this paper, we present STAGE, a self-supervised end-to-end framework for multivariate time series anomaly detection. STAGE models each dimension of a multivariate time series as a node and explicitly captures inter-dimensional dependencies via dynamic edges, all while preserving permutation invariance across dimensions. Existing graph-based methods Boniol & Palpanas (2020); Wenig & Papenbrock (2024); Zhao et al. (2020); Deng & Hooi (2021); Tang et al. (2023) represent the multivariate time series as a single static graph. However, this assumption fails to capture correlations between time windows that characterize transient anomalies. To overcome this limitation, STAGE constructs an adaptive, window-specific graph for each temporal window. After pruning spurious

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.

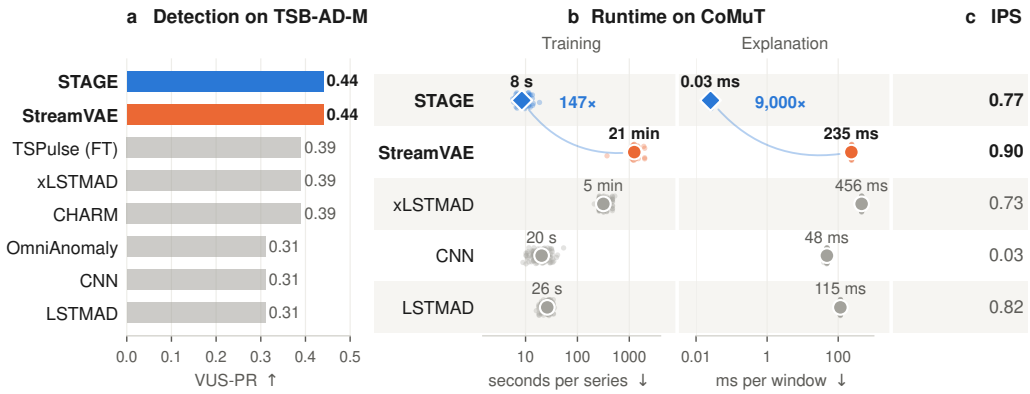


Figure 1: What STAGE achieves: **a.** State-of-the-art accuracy detection on the largest benchmark, TSB-AD-M; **b.** Fast runtime on both training and explanation (147x and 9.000x faster than the other state-of-the-art method), and **c.** high quality of explanations (IPS on CoMuT in the Figure).

connections via a learnable attention mechanism, we encode node representations using a GCN Kipf & Welling (2017), enforce dimension permutation invariance through DeepSet aggregation Zaheer et al. (2017), and score anomalies using a Deep SVDD objective Ruff et al. (2018).

As shown in Figure 1, STAGE has three major strenghts: **(a).** Evaluated on TSB-AD-M, the largest public TSAD benchmark, STAGE matches the state-of-the-art, outperforming the second-best method by 12% in VUS-PR. **(b).** By leveraging graph structures, it does so by taking 147x less training time and 9000x less runtime to produce explanations on the CoMuT benchmark; **(c).** Achieves high explainability scores (in terms of IPS Li et al. (2021)).

Crucially, STAGE is so fast in generating explanations that it enables humans-in-the-loop in the process. Explanations, in the form of attribution vectors, can be generated in milliseconds, and we leverage this by also introducing a visualization tool that closes the end-to-end process and produces live explanations for every window in the time series (Section 5.3).

Our contributions are as follows: **(i).** We introduce STAGE, an extremely fast, graph-based, permutation-invariant model for multivariate TSAD; **(ii).** We show that STAGE achieves state-of-the-art performance on the largest TSAD benchmark; **(iii).** We prove that STAGE is able to explain the anomalies with post-hoc occlusion thanks to its graph representation; **(iv).** We release a replication package, and the full implementation and a visualization tool, that helps explaining anomaly detections to the end-user. [1].

2 RELATED WORK

We categorize time series anomaly detection approaches into *distance-*, *density-*, *prediction-*, and *graph-based* methods based on their underlying assumptions regarding temporal structure, recurrence, noise, and anomaly types, following Boniol et al. (2024).

Distance-based approaches Keogh et al. (2007); Liu et al. (2009); Yeh et al. (2016); Linardi et al. (2020) identify anomalies by quantifying the dissimilarity relative to previously observed patterns. This family includes proximity-, clustering-, and discord-based approaches that identify subsequences whose closest matches exhibit unusually large distances . These methods are particularly effective when anomalies are to rare subsequences that deviate significantly from the series' dominant patterns.

Density-based methods Boniol & Palpanas (2020); Wenig & Papenbrock (2024); Liu et al. (2008); Gao et al. (2020) assess the degree of isolation of an observation or subsequence within an induced representation space. Rather than relying solely on raw temporal distances, these methods estimate local or global data density and identify sparse or weakly supported regions as anomalous. This category includes several subfamilies, such as distribution-, graph-, tree-, and encoding- methods. Such methods are attractive when the notion of anomaly is better captured through structural relations

[1] https://anonymous.4open.science/r/stage-iclr-4BE1

than through direct pointwise comparison. We describe graph-based representations in detail below, as they are most closely related to our approach.

Prediction-based methods model expected temporal behaviour and utilize prediction error as the anomaly signal. They are commonly divided into *forecasting-based* and *reconstruction-based* approaches Benecki et al. (2021); Sakurada & Yairi (2014). Forecasting-based detectors predict future values from past observations and interpret large residuals as indicators of anomalous behavior. In contrast, reconstruction-based methods learn compact representations of normal temporal patterns and define anomalies as observations that deviate substantially from the normality and thus feature large reconstruction error. These approaches are particularly effective when normal dynamics exhibit sufficient regularity to allow for reliable prediction or reconstruction. Moreover, such methods rely on a self-supervising training phase that uses only observed time series (i.e., self-supervision). For this reason, prediction-based approaches include also recent time series foundation models Ansari et al. (2025); Graf et al. (2025); Feofanov et al. (2025); Woo et al. (2024); Goswami et al. (2024); Das et al. (2024); Auer et al. (2026); Cohen et al. (2025).

Prediction-based methods naturally provide explanations by computing the reconstruction error of the individual dimensions. We show that a graph-based approach can achieve comparable explainability and accuracy with a substantial speedup in both training and explanation time.

Graph-based largely follow one of two paradigms. The first maps subsequences to nodes of a graph whose edges capture transitions between recurring patterns Boniol & Palpanas (2020). Series2Graph++ Wenig & Papenbrock (2024) extends this graph of subsequences to the multivariate and distributed settings. The second paradigm uses graph neural networks to learn relationships between the variables of a multivariate series. MTAD-GAT Zhao et al. (2020) treats each univariate series as a node of a complete graph and applies graph attention to learn inter-feature dependencies. GDN Deng & Hooi (2021) and its recurrent extension GRN Tang et al. (2023) score anomalies by measuring prediction errors over a single, static dependency graph learned from dimension embeddings. Consequently, they cannot account for temporal variations in inter-dimension relationships.

STAGE differs in three fundamental respects: **(i) Dynamic topology:** Instead of fixing a global adjacency matrix across the dataset like GDN and GRN, STAGE generates an adaptive, window-specific graph to capture transient relationships. **(ii) Permutation invariance:** STAGE is strictly invariant to dimension ordering, eliminating bias toward arbitrary channel indexing (Appendix C.1). **(iii) Structural anomaly attribution:** Rather than relying on forecasting residuals that conflate prediction errors with root causes, we leverage our graph representations for occlusion-based attribution (Section 5.3), directly isolating each dimension's causal impact on the anomaly score.

3 BACKGROUND

A *multivariate time series* $\mathbf{T} \in \mathbb{R}^{D \times n}$ is a sequence $\mathbf{T} = [T^{(0)}, T^{(1)}, \ldots, T^{(D-1)}]$ where each dimension $j \in [0, D-1]$ is a univariate time series

$$T^{(j)} = [T_0^{(j)}, T_1^{(j)}, \ldots, T_{n-1}^{(j)}], \quad T_k^{(j)} \in \mathbb{R}. \tag{1}$$

Time series are often segmented and processed in smaller parts known as subsequences or windows. Specifically, a univariate subsequence $T_{i,\ell}^{(j)} \in \mathbb{R}^\ell$ is a contiguous block of values of fixed length ℓ, usually much smaller than the total length of $\mathbf{T}$, and easily generalizable to the multivariate case:

$$T_{i,\ell} = [T_{i,\ell}^{(0)}, T_{i,\ell}^{(1)}, \ldots, T_{i,\ell}^{(D-1)}], \quad i \in [0, n-\ell]. \tag{2}$$

As a consequence, we can view a multivariate time series as a collection $\mathbf{T}_\ell = \{T_{0,\ell}, T_{1,\ell}, \ldots, T_{n-\ell,\ell}\}$ of all subsequences of length ℓ. Given a multivariate time series, we formulate anomaly detection as follows.

Problem 1 (Multivariate Time Series anomaly Detection (MTSAD)) *Let $\mathbf{T} \in \mathbb{R}^{D \times n}$ be a multivariate time series. The goal of anomaly detection is to find a function*

$$f : \mathbf{T} \to S, \quad S = [s_0, s_1, \ldots, s_{n-1}], \quad s_i \in \{\mathcal{N}, \mathcal{A}\}. \tag{3}$$

that assigns to each timestamp i of a time series $\mathbf{T}$ a binary label indicating whether the local behaviour is normal ($\mathcal{N}$) or anomalous ($\mathcal{A}$).

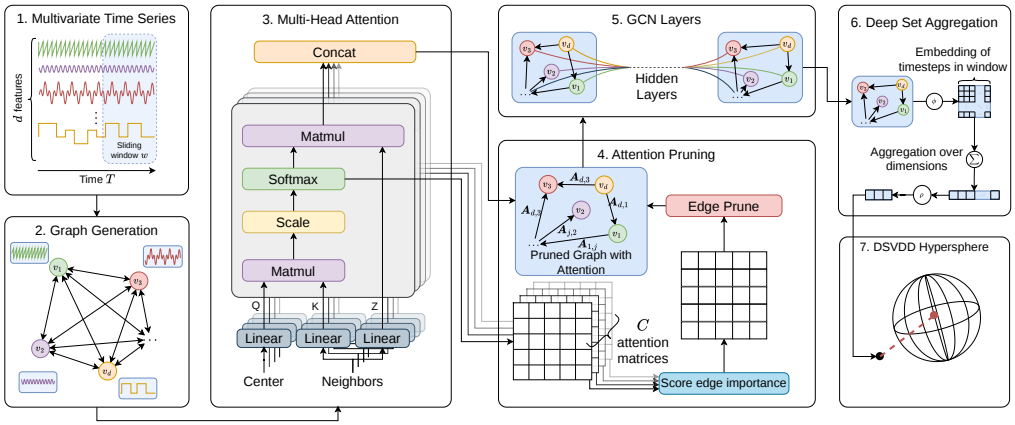


Figure 2: The architecture of STAGE.

This naturally extends to f operating over subsequences $T_{i,\ell}$ rather than points T_i and outputting a label per subsequence. Often, as the binary classification is difficult to infer directly, a continuous anomaly score where $S_i \in [0, 1]$ is used instead. To convert this into a binary decisions, a fixed threshold $\tau \in [0, 1]$ is typically used.

Explainability in MTSAD. Explainability refers to the problem of finding, for each time step, the subset of the input responsible for the anomaly.

Problem 2 (Scoring-based Spatial Anomaly Explanation) *Let* $\mathbf{T} \in \mathbb{R}^{D \times n}$ *be a multivariate time series* anomaly explanation *aims to find a function*

$$f_e : \mathbf{T} \to E_T, \quad E_T = [E_{T,0}, E_{T,1}, \ldots, E_{T,n-1}] \qquad E_{T,i} = [d_0, \ldots, d_{D-1}], \quad d_k \in [0, 1]$$

where $E_{T,i} = [d_0, \ldots, d_{D-1}]$ *quantifies the degree of abnormality of each dimension at time step* i.

The scores defined in Problem 2 rank the dimensions from most to least anomalous. Similar to Problem 1, a threshold τ can be applied to identify anomalous dimensions. In practice, because selecting an optimal threshold is notoriously difficult, the top-k scoring dimensions for a fixed k are typically classified as anomalous instead Su et al. (2019).

4 STAGE

Achieving explainability while retaining the effectiveness of neural methods requires careful architectural design. STAGE is built to combine the strengths of both approaches: graphs are used not only to encode the time series, but also as an intuitive representation to explain anomaly detections to the end user. Here, we describe how STAGE[2] achieves this dual objective. Structured around the architecture depicted in Figure 2, Section 4.1 details STAGE's components, while Section 4.2 describes the post-hoc explainability mechanism that makes STAGE interpretable.

4.1 THE ARCHITECTURE OF STAGE

Time Series as Graphs. To model time series input as a graph, STAGE performs the following transformation. Each subsequence $T_{i,\ell}$ is encoded as a fully connected graph $G_{\mathbf{T}_{i,\ell}}$, where $T_{i,\ell}^{(j)}$ is node v_j's feature vector. Figure 2 (Box 1-2) displays this transformation.

Contrary to single time steps, we mark each subsequence $G_{\mathbf{T}_{i,\ell}}$ as anomalous iff any of the observations in $T_{i,\ell}$ is anomalous. We employ a continuous scoring function and assign to each window a single score. Since each time step belongs to multiple subsequences, the final score for a time

[2]**S**hallow **T**ransformer model for **A**nomaly detection in multivariate time series with **G**raph **E**mbeddings

step is an aggregation of the scores of all windows containing this time step. As such, we need two functions; one that produces a matrix of per window scores $\mathbf{S}_\ell \in \mathbb{R}^{(n-\ell)\times\ell}$, and one aggregating individual scores into a final vector $\mathbf{S_T} \in \mathbb{R}^n$. For a visual representation, refer to Figure 5. By aggregating over an entire column of $\boldsymbol{S}_\ell$, the resulting aggregated score of the observation at time t represents every window that contains t. Intuitively, an anomalous observation should result in a large score after aggregation. Two examples are depicted in Figure 5.

Edge modelling and pruning. To capture the intrinsic correlations between dimensions occurring in multivariate time series, we employ a Graph Transformer Yunsheng et al. (2021). Graph transformers learn a correlation matrix, conveniently represent as a weighted graph, by exploiting the attention mechanism Vaswani et al. (2017). Since the number of dimensions is usually small (i.e., less than 100) the initially dense correlation matrix conveniently fits any consumer hardware's memory and allows for modelling all possible correlations among dimensions.

The transformer layer of STAGE (Figure 2, Box 3) features three attention heads[3]. After the transformer layer, the updated vertex embeddings and the C learned attention matrices are passed to the pruning step.

Viewing these attention matrices as analogous to weighted adjacency matrices, we prune the input graph before passing it along a GCN Kipf & Welling (2016) to learn a per-dimension representation. To this end, we aggregate information from the C attention matrices and compute, for each edge (i, j), the mean $\mu_{i,j}$ and variance $\sigma^2_{i,j}$ across all attention heads.

To allow for differences in the values across attention heads, while still keeping values that more heads agree on, we scale the weight of each edge by $\left(1 - \sigma^2_{i,j}\right)$, and assigning the resulting score $s_{i,j}$ to each edge: $s_{i,j} = \mu_{i,j}(1 - \sigma^2_{i,j})$.

This approach implies that an edge with a low mean attention weight can achieve higher score than an edge with a higher mean attention weight provided the variance is low. Intuitively, the model favours attention weights that consistent among heads. Having calculated the edge weights, we compute v_j's node importance ι_j as the mean of the incoming edges $\iota_j = \sum_{v_i \in \mathcal{N}(v_j)} \frac{s_{i,j}}{|\mathcal{N}(v_j)|}$.

Using these scores, passed through a sigmoid function to soften the impact to less important edges so that they are not pruned too early in the training process, we compute the pruned weighted adjacency matrix $\boldsymbol{A}$ (Figure 2, Box 4), by removing edges less important then their destination vertex.

Definition 1 (Pruned Adjacency Matrix) *Let ς be a sigmoid function. Given a score $s_{i,j}$ for each edge (i, j) and importance ι_j for each vertex v_j, the pruned adjacency matrix $\boldsymbol{A}$ is given by:*

$$\boldsymbol{A} \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}, \quad \boldsymbol{A}_{i,j} = p(m_{i,j}) \quad m_{i,j} = 2\varsigma(s_{i,j} - \iota_j) - 1 \quad p(x) = \begin{cases} x & \text{if } x > 0, \\ 0 & \text{if } x \leq 0. \end{cases} \tag{4}$$

The pruned adjacency matrix produced by the pruning step, alongside the updated vertex representations from the attention layer, is the input to the GCN layer (Figure 2, Box 5).

Deep Set Aggregator. Directly flattening the output of the GCN layer as an input to a loss function would reinstate the ordering bias that was avoided by the graph representation, voiding the desiderata of permutation invariance. To prevent this, we employ a Deep Set aggregation Zaheer et al. (2017) that first combines each timestamp in a window into a single vector, compresses it through an MLP, aggregates each dimension, and finally projects it to the center vector used by the loss function.

Loss function. We treat anomaly detection as a semi-supervised one class classification task that aims to learn the normal state of the data. As such, anomalous observations are those that deviate enough from the learned distribution. To this end, we employ Deep Support Vector Data Description (Deep SVDD) Ruff et al. (2018), an objective function that combines one class classification with representation learning. Deep SVDD maps points corresponding to normal states as close as possible to the center of a hypersphere in this space and pushes anomalous points away from the center. The analytical form of DSVDD is

$$\min_{\mathcal{W}} \ \frac{1}{n} \sum_{i=1}^{n} ||\phi\left(\boldsymbol{x}_i; \mathcal{W}\right) - c||^2, \tag{5}$$

[3]The number of attention heads was chosen as a hyperparameter from early testing.

where $\phi(\cdot)$ is the output of an arbitrary neural network, and c is the centre of the hypersphere to be learned. This hypersphere can have arbitrary dimensionality, but one must be weary of the curse of dimensionality. STAGE is scored using a hypersphere in three dimensions. In the wrong implementation, Deep SVDD could learn trivial embeddings, mapping every output to zero regardless of input (hypersphere collapse Ruff et al. (2018)). STAGE avoids this by excluding bias terms and unbounded activations. The Deep SVDD objective function is visualised in Figure 2, Box 7.

4.2 POST-HOC EXPLAINABILITY

Besides detecting anomalies, STAGE should be able to explain the reason for the detected anomaly. Formally, STAGE should produce per-dimension scores $E_{T,i}$ as in Problem 2 alongside the anomaly score. Prediction-based methods yield these scores directly by computing per-dimension residuals between predicted and observed values. STAGE has no such residual, so we derive the per-dimension scores with a different mechanism.

An obvious candidate is to extract scores before aggregating them with Deep Sets, which applies ϕ to every vertex (i.e., dimension) before summing. As such, one could score each $\phi(v_j)$ against a per-dimension center. This solution, however, implicitly assumes independence among dimensions, failing to capture multivariate anomalies. Instead, for a window $T_{i,l}$, we aim to quantify the contribution of each dimension to the overall score of $T_{i,l}$.

The mechanism of occlusion measures this contribution directly. For each window, we measure difference between the initial score and the one obtained ignoring one dimension. Since every dimension of a subsequence is a vertex of $G_{T_{i,\ell}}$, ignoring one dimension corresponds to deleting a vertex together with its incident edges and score the resulting graph with respect to the centre c. This process involves no retraining and requires only a forward pass (batched with D graphs) on the trained model. Thus, occlusion leverages the native graph-representation and its order-invariance.

Definition 2 (Occlusion Score) *Let f be the map from a window graph to the embedding produced by STAGE, let c be the centre of the hypersphere, and let $s(G)$ be the anomaly score of Equation 5. For a window graph G with vertices $v_0, \ldots, v_{D-1}$, write $G \setminus v_j$ for the graph obtained by removing vertex v_j and its incident edges. The occlusion score of dimension j is $d_j = s(G) - s(G \setminus v_j)$.*

A large positive d_j indicates that the removal of dimension j moved the point corresponding to the subsequence closer to the centre of the hypersphere thus revealing that such dimension has a large impact on the score. Conversely, a value close to 0 means no impact, and a negative value indicates a negative impact. Definition 2 scores a single window; as with anomaly scores, each time step belongs to multiple overlapping windows. Every window containing time step t contributes its attribution vector $[d_0, \ldots, d_{D-1}]$ to that step, and these contributions are summed dimension by dimension. Importantly, computing explanations leaves detection results unchanged and requires no modifications to the training process.

This occlusion-based method, removing one node at a time, works naturally with graph representations, and extending it to other domains is not trivial. Moreover, graphs can be used to present the explanation to the end-user in an intuitive way: we show this by also building a dedicated visualization tool (section 5.3).

5 EXPERIMENTAL EVALUATION

In this section, we first detail the experimental setup, including datasets and models, that we adhere to (section 5.1). Then, in section 5.2, we show results of STAGE against the baselines in terms of accuracy. Finally, in section 5.3, we show numerical explainability results for STAGE and how graphs can be used to aid interpretability of results, thanks to the visualization tool that we developed and release together with STAGE.

5.1 EXPERIMENTAL SETUP

All experiments run on a server with the following configuration. **CPU:** Intex Xeon Silver 4410T 40-Core, **GPU:** 2x NVIDIA 4090, **OS:** Ubuntu 22.04.5 LTS (64 bit). Every computed metric is an average over 5 runs in an effort to account for the variability inherent in deep methods.

	Mean	CATSv2	CreditCard	Daphnet	Exathlon	GECCO	GHL	Genesis	LTDB	MITDB	MSL	OPPORTUNITY	PSM	SMAP	SMD	SVDB	SWaT	TAO
MAD_STAGE	0.44	0.06	0.04	0.41	0.89	0.06	0.03	0.11	0.44	0.29	0.42	0.13	0.17	0.51	0.28	0.56	0.36	0.79
StreamVAE	0.44	0.12	0.06	0.38	0.92	0.04	0.02	0.42	0.44	0.16	0.44	0.15	0.20	0.48	0.40	0.48	0.39	0.77
xLSTMAD	0.39	0.23	0.07	0.28	0.94	0.19	0.02	0.67	0.42	0.14	0.33	0.31	0.40	0.28	0.40	0.34	0.52	0.74
CHARM	0.39	0.06	0.05	0.48	0.90	0.07	0.05	0.10	0.46	0.49	0.22	0.09	0.24	0.18	0.30	0.55	0.49	0.69
TSPulse (FT)	0.39	0.07	0.00	0.36	0.90	0.18	0.01	0.02	0.57	0.13	0.20	0.06	0.14	0.31	0.35	0.48	0.13	0.93
TSPulse (ZS)	0.36	0.05	0.00	0.35	0.89	0.18	0.01	0.01	0.36	0.07	0.20	0.06	0.14	0.30	0.35	0.39	0.13	0.93
MMPAD	0.35	0.05	0.01	0.04	0.84	0.03	0.06	0.01	0.60	0.37	0.10	0.36	0.11	0.07	0.10	0.63	0.27	0.77
TimeRCD	0.32	0.23	0.04	0.15	0.93	0.08	0.02	0.00	0.21	0.24	0.23	0.28	0.22	0.28	0.46	0.15	0.17	0.19
CNN	0.31	0.08	0.02	0.21	0.68	0.03	0.02	0.10	0.33	0.14	0.35	0.16	0.22	0.19	0.35	0.19	0.41	1.00
OmniAnomaly	0.31	0.04	0.02	0.34	0.84	0.02	0.07	0.00	0.44	0.11	0.22	0.18	0.16	0.12	0.17	0.35	0.15	0.81
+21 more	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

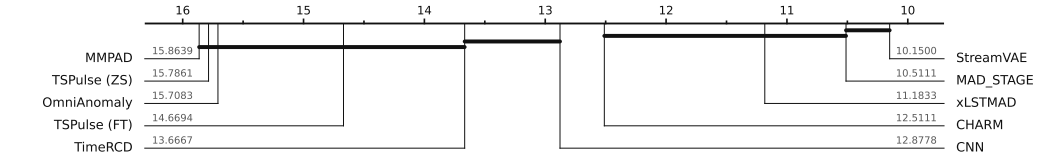


Figure 3: Top: VUS-PR per suite in TSB-AD-M-Eval for the top-10 best performing methods; leftmost column is the mean across all datasets; colours normalized across each row, darker colours indicate higher VUS-PR. Bottom: critical difference diagram of the same methods, ranked by average placement on all TSB-AD-M datasets; connected lines indicate lack of statistical difference.

Datasets. For efficiency, we test our method on TSB-AD-M (Liu & Paparrizos (2024)), the biggest benchmark in the literature. However, not all the suites in TSB-AD-M are labeled for explainability, and runtimes are usually not reported. For this reason, to test these two aspects, we use the SMD Su et al. (2019) and CoMuT Wenig & Papenbrock (2024) benchmarks. Dataset details are reported in Appendix B. Please note that SMD is a subset of TSB-AD-M.

Metrics. Anomalies in time series data are seldom contained to a single time step, therefore it is more accurate to model anomalies as ranges instead of singular points. VUS-PR (Paparrizos et al. (2022)) utilizes a continuous score functions combined with the idea that many models view the time series in windows, and is the preferred reported metrics in most TSAD benchmarks, such as TSB-AD-M (Liu & Paparrizos (2024)). We adapt to this setting by using VUS-PR as the main metric. For explainability, we measure IPS, as described by Li et al. (2021). IPS is a well-established metric for multivariate TSAD (refer to Appendix B.2 for details).

Baselines. We compare STAGE against 30 baselines on the TSB-AD-M benchmark. The reported numbers are the same as in the benchmark page[4] for each dataset. However, few of the methods have systematically tested on explainability before, so comparable results are not available. To this end, we re-ran TSB-AD-M's top-ranked baselines on CoMuT and SMD, using attribution method SHAP (Lundberg & Lee (2017)) for explainability. SHAP is the closest general-purpose explainability method to our occlusion mechanism available in the literature.

Ablations. We run a comprehensive ablation study on the model configuration of STAGE, ablating the number GCN layers and Deep Set layers. We also omit the pruning layer: in this case, pooling is applied to the attention weights, allowing for their use as edge weights in the following layers. Additionally, we ablate the ϕ and ρ layers of the deep set aggregation: in this case, they are replaced by a single linear layer. The results of these ablations are visualized in Figure 8 in Appendix C, where we show that the full STAGE configuration is the best performing and most consistent. Note that the ablations are performed on the datasets TSB-AD-M-Tuning, a configuration of TSB-AD-M specifically designed for tuning models, and CoMuT.

5.2 ACCURACY RESULTS

[4] https://thedatumorg.github.io/TSB-AD/

> *STAGE is the most accurate method available, matching the state-of-the-art StreamVAE on TSB-AD-M. It surpasses the second best baseline by 12%, and ranks first on multiple datasets. We show that this result is statistically significant via Friedman-Wilcoxon tests.*

Comparing against the TSB-AD-M leaderboard, Figure 6, we outperform every model on the TSB-AD-M-Eval dataset, matching StreamVAE. With a VUS-PR of 0.44 both of these surpass the other current best performers that achieve VUS-PR 0.39. Thus, STAGE outperforms all others current state-of-the-art methods by at least 12%. Looking at the scores for the individual data-suites within TSB-AD-M (Figure 6), STAGE performs especially well on datasets from two domains: Spacecraft (SMAP and MSL) and healthcare (DAPHNET, LTDB, MITDB, and SVDB).

Spacecraft (SMAP & MSL). The *SMAP* and *MSL* datasets consist of telemetry data from the Soil Moisture Active Passive satellite and the Curiosity rover respectively Hundman et al. (2018). STAGE, with a VUS-PR of 0.51, outperforms every model on *SMAP*. On the *MSL* suite, STAGE achieves VUS-PR of 0.42, surpassed only by KMEANSAD and STREAMVAE with VUS-PR of 0.44. Notably, as opposed to STAGE, KMEANSAD is designed specifically for anomaly detection in spacecraft systems Yairi et al. (2001), whereas STAGE is a general approach. Despite this, compared to the models benchmarked by Liu & Paparrizos (2024), STAGE is the strongest candidate for anomaly detection in spacecraft telemetry data.

Healthcare (Daphnet, LTDB, MITDB & SVDB). These datasets come from many different sources. Daphnet Bachlin et al. (2010) features sensors measuring movement in patients with Parkinsons, where anomalies are caused by incidents of "Freezing of Gait" symptoms. Both MITDB and LTDB are subsets of PhysioNet Goldberger et al. (2000), a large collection of different medical data. Much of it is cardiovascular data (ECG and measurements of blood pressure and oxygen levels), but it also includes many other sources from sleep tracking to motion tracking. SVDB Greenwald et al. (1990) consists of ECG diagrams, with anomalies being cases of arrhythmia (irregular heartbeat). On two of the healthcare datas suites, Daphnet and SVDB, STAGE is the second best performer. Notably, STAGE outperforms STREAMVAE significantly on this benchmark.

Statistical testing. To assess statistical significance, we present a critical difference diagram Demšar (2006) computed on the TSB-AD-M-Eval dataset. The diagram is computed following the method described in Ismail Fawaz et al. (2019). First, a Friedman test (Friedman (1940)) is performed, computing the average rank of all models across all time series. The average rank is presented as the top axis. Then, Wilcoxon signed-rank test and Holm correction determines which methods are statistically different Wilcoxon (1945).

Observing the results in Figure 3, STAGE significantly outperforms all models except xLSTMAD and STREAMVAE. In the plot, methods with differences that are not statistically significant are connected by a horizontal line. Interestingly, TSPULSE (FT) achieves the same mean VUS-PR of 0.39 as CHARM and xLSTMAD, but has significant lower average placement. Likely this is caused by TSPULSE (FT) performing very well on a few datasets, but worse overall; this effect is not captured by the mean score, but rather the Friedman test. The Critical Difference Diagram clearly shows that STAGE exhibits state-of-the-art performance with statistical significance over almost all baselines ($p \leq 0.05$ for all pairwise Wilcoxon tests) except StreamVAE, for which it matches VUS-PR: the difference between the two models is not statistically significant. The individual p-values of each Wilcoxon tests are visualized in Figure 9, Appendix A.

5.3 EFFICIENCY AND EXPLAINABILITY

> *STAGE is orders of magnitude faster w.r.t. the state-of-the-art in producing explanations. We also introduce a visualization tool to explain STAGE's detections to end users.*

Producing explanations fast. Figure 4 (left) juxtaposes explanation runtime against attribution accuracy (measured via IPS) on the CoMuT and SMD datasets for the five top-performing methods in the TSB-AD-M benchmark. STAGE achieves an IPS comparable to the best-performing method (StreamVAE) while being orders of magnitude faster. This efficiency is achieved because our occlusion method exploits the native graph representation of STAGE, requiring only a single forward pass over a batch of D graphs. The competing methods do not natively support explainability. As

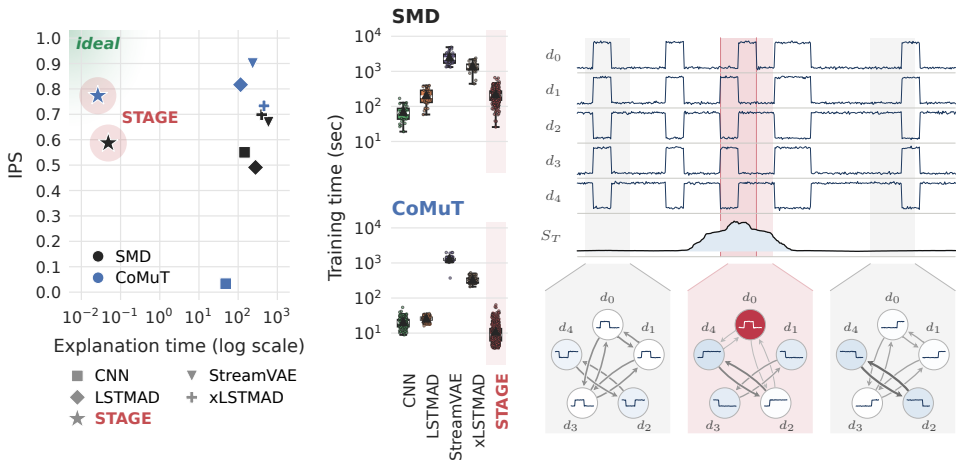


Figure 4: **Left:** IPS against Explanation time (milliseconds) on SMD and CoMuT. **Center:** Training runtimes (seconds) on CoMuT and SMD. **Right:** graphs as an explainability tool. Each time window is encoded as a graph; node colour is the occlusion score d_j (red = higher; blue= lower).

such, occlusion cannot be applied without substantial architectural modifications, forcing reliance on post-hoc explainers such as SHAP Lundberg & Lee (2017), which are far more computationally expensive. As a result, STAGE explains a window in 0.03 ms on CoMuT and 0.05 ms on SMD, while StreamVAE needs 235 ms and 592 ms respectively: roughly $9{,}000\times$ and $12{,}000\times$ faster.

Training runtime. Figure 4 (center) shows the training time, in seconds, on SMD and CoMuT. STAGE trains faster than all competing methods on CoMuT while remaining competitive on SMD. Against StreamVAE, the baseline with the closest detection performance, the median training time on CoMuT is 9 s for STAGE versus 1,248 s ($\approx$21 min) for StreamVAE; on SMD STAGE takes 209 s versus 2,296 s ($\approx$38 min) of StreamVAE. In other words, STAGE trains respectively $140\times$ and $11\times$ faster than StreamVAE on these datasets.

Graphs as an explainability tool. We designed STAGE as an end-to-end anomaly detection tool, and numerical results alone are not sufficient to explain the detected anomalies to the end user. For this reason, we additionally provide a visualization tool to help the user's comprehension. Figure 4 (right) shows an excerpt of such a tool: since STAGE encodes each window in a graph (unlike previous graph-based methods), each graph can be anomalous or not. Users can interact with the graphs that are deemed anomalous and examine the score by inspecting or removing each individual node and edge. Since each node encodes a dimension, the anomaly reflects directly to the time series. Because an explanation with STAGE takes milliseconds, the tool keeps the trained model loaded and recomputes scores and explanations on demand: the user can move to any window, remove any set of vertices and immediately see the rewired graph and the new score, or open the matrix of pairs for the current window. The tool is publicly available in the repository.

6 CONCLUSION

Reaching explainability, accuracy, and speed is the holy grail of any machine learning model. In this paper, we introduced STAGE, an end-to-end framework that explicitly captures inter-variable relationships in multivariate time series via dynamic graphs. On TSB-AD-M, the largest benchmark for multivariate anomaly detection, STAGE matches state-of-the-art detection accuracy. Crucially, its native graph representation enables a lightweight occlusion mechanism that yields attribution quality on par with the strongest baselines at a fraction of the computational runtime, thus enabling real-time, interactive root-cause analysis. Ultimately, STAGE bridges the gap between the interpretability of relational graphs and the predictive power of deep neural architectures.

AI USE STATEMENT

In this work, we used generative AI tools for aid in implementing methods.

We have **not** used generative AI tools for formalizing mathematical claims, assisting with translation, helping develop theoretical models, providing critical ingredients for proving mathematical claims, providing feedback on research methodology, supporting qualitative and thematic data analysis, and interpreting results.

Finally, generating synthetic datasets, assisting in writing proofs, proposing hypotheses, and cleaning datasets are not applicable to this work.

We have reviewed all AI-assisted work: LLM-generated code was verified and tested for correctness by 2 authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We made our best effort to make results reproducible. Our repository is publicly available at `https://anonymous.4open.science/r/stage-iclr-4BE1`, and contains a detailed README file on how to reproduce experiments. Specifically, we made available a `reproduce.sh` file that, when run, will reproduce STAGE experiments on TSB-AD-M and Comut.

Throughout working with the TSAD models, datasets and metrics included in this paper, challenges with reproducibility were encountered. All models used showed large variations in their experimental results. To alleviate these challenges, a testing framework which allowed for careful version control and repetition of experiments was built.

REFERENCES

Abdul Fatir Ansari, Oleksandr Shchur, Jaris Küken, Andreas Auer, Boran Han, Pedro Mercado, Syama Sundar Rangapuram, Huibin Shen, Lorenzo Stella, Xiyuan Zhang, Mononito Goswami, Shubham Kapoor, Danielle C. Maddix, Pablo Guerron, Tony Hu, Junming Yin, Nick Erickson, Prateek Mutalik Desai, Hao Wang, Huzefa Rangwala, George Karypis, Yuyang Wang, and Michael Bohlke-Schneider. Chronos-2: From univariate to universal forecasting, 2025.

Andreas Auer, Patrick Podest, Daniel Klotz, Sebastian Böck, Günter Klambauer, and Sepp Hochreiter. Tirex: Zero-shot forecasting across long and short horizons with enhanced in-context learning. *NeurIPS*, 38:57529–57580, 2026.

Marc Bachlin, Meir Plotnik, Daniel Roggen, Inbal Maidan, Jeffrey M. Hausdorff, Nir Giladi, and Gerhard Troster. Wearable assistant for parkinson's disease patients with the freezing of gait symptom. *IEEE Trans. Inf. Technol. Biomed.*, 14(2):436–446, 2010.

Pawel Benecki, Szymon Piechaczek, Daniel Kostrzewa, and Jakub Nalepa. Detecting anomalies in spacecraft telemetry using evolutionary thresholding and lstms. In *GECCO*, pp. 143–144, 2021.

Ane Blázquez-García, Angel Conde, Usue Mori, and José A. Lozano. A review on outlier/anomaly detection in time series data. *CSUR*, 54(3):56:1–56:33, 2022.

Paul Boniol and Themis Palpanas. Series2graph: Graph-based subsequence anomaly detection for time series. *PVLDB*, 13(12):1921–1934, 2020.

Paul Boniol, Qinghua Liu, Mingyi Huang, Themis Palpanas, and John Paparrizos. Dive into time-series anomaly detection: A decade review. *arXiv preprint arXiv:2412.20512*, 2024.

Ben Cohen, Emaad Khwaja, Youssef Doubli, Salahidine Lemaachi, Chris Lettieri, Charles Masson, Hugo Miccinilli, Elise Ramé, Qiqi Ren, Afshin Rostamizadeh, et al. This time is different: An observability perspective on time series foundation models. *arXiv preprint arXiv:2505.14766*, 2025.

Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. A decoder-only foundation model for time-series forecasting. In *ICML*, volume 235 of *PMLR*, pp. 10148–10167, 21–27 Jul 2024.

Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *JMLR*, 7:1–30, 2006.

Ailin Deng and Bryan Hooi. Graph neural network-based anomaly detection in multivariate time series. In *AAAI*, volume 35, pp. 4027–4035, 2021.

Vasilii Feofanov, Songkang Wen, Marius Alonso, Romain Ilbert, Hongbo Guo, Malik Tiomoko, Lujia Pan, Jianfeng Zhang, and Ievgen Redko. Mantis: Lightweight calibrated foundation model for user-friendly time series classification. *arXiv preprint arXiv:2502.15637*, 2025.

Milton Friedman. A comparison of alternative tests of significance for the problem of m rankings. *The Annals of Mathematical Statistics*, 11(1):86–92, 1940.

Yifeng Gao, Jessica Lin, and Constantin Brif. Ensemble grammar induction for detecting anomalies in time series. In *EDBT*, pp. 85–96, 2020.

Ary Goldberger, Luís Amaral, Leon Glass, Jeffrey Hausdorff, Plamen Ivanov, Roger Mark, Joseph Mietus, George Moody, Chung-Kang Peng, and H. Stanley. Physiobank, physiotoolkit, and physionet : Components of a new research resource for complex physiologic signals. *Circulation*, 101:E215–20, 07 2000.

Mononito Goswami, Konrad Szafer, Arjun Choudhry, Yifu Cai, Shuo Li, and Artur Dubrawski. MOMENT: A family of open time-series foundation models. In *ICML*, volume 235 of *PMLR*, pp. 16115–16152, 21–27 Jul 2024.

Lars Graf, Thomas Ortner, StanisĹ WoĹşniak, Angeliki Pantazi, et al. Flowstate: Sampling rate invariant time series forecasting. *arXiv preprint arXiv:2508.05287*, 2025.

Scott David Greenwald, Ramesh S Patil, and Roger G Mark. *Improved detection and classification of arrhythmias in noise-corrupted electrocardiograms using contextual information*. IEEE, 1990.

Kyle Hundman, Valentino Constantinou, Christopher Laporte, Ian Colwell, and Tom Soderstrom. Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding. In *KDD*, KDD '18, pp. 387–395. ACM, 2018.

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

Eamonn Keogh, Stefano Lonardi, Chotirat Ann Ratanamahatana, Li Wei, Sang-Hee Lee, and John Handley. Compression-based data mining of sequential data. *DMKD*, 14(1):99–129, 2007.

Thomas N Kipf and Max Welling. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308*, 2016.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017.

Zhihan Li, Youjian Zhao, Jiaqi Han, Ya Su, Rui Jiao, Xidao Wen, and Dan Pei. Multivariate time series anomaly detection and interpretation using hierarchical inter-metric and temporal embedding. In *KDD*, pp. 3220–3230, 2021.

Michele Linardi, Yan Zhu, Themis Palpanas, and Eamonn Keogh. Matrix profile goes MAD: variable-length motif and discord discovery in data series: M. linardi et al. *DMKD*, 34(4): 1022–1071, 2020.

Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. In *ICDM*, pp. 413–422, 2008.

Qinghua Liu and John Paparrizos. The elephant in the room: Towards a reliable time-series anomaly detection benchmark. In *NeurIPS*, 2024.

Yubao Liu, Xiuwei Chen, Fei Wang, and Jian Yin. Efficient detection of discords for time series stream. In *APWeb*, pp. 629–634. Springer, 2009.

Scott M. Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *NeurIPS*, pp. 4768–4777, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

Kadir-Kaan Özer, René Ebeling, and Markus Enzweiler. STREAM-VAE: Dual-path routing for slow and fast dynamics in vehicle telemetry anomaly detection. In *IEEE IV*, pp. 444–451, 2026.

John Paparrizos, Paul Boniol, Themis Palpanas, Ruey S. Tsay, Aaron J. Elmore, and Michael J. Franklin. Volume under the surface: A new accuracy evaluation measure for time-series anomaly detection. *PVLDB*, 15(11):2774–2787, 2022.

Lukas Ruff, Robert Vandermeulen, Nico Goernitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, Emmanuel Müller, and Marius Kloft. Deep one-class classification. In *ICML*, volume 80 of *PMLR*, pp. 4393–4402, 10–15 Jul 2018.

Mayu Sakurada and Takehisa Yairi. Anomaly detection using autoencoders with nonlinear dimensionality reduction. In *MLSDA*, pp. 4–11, 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 *KDD*, pp. 2828–2837, 2019.

Chaofan Tang, Lijuan Xu, Bo Yang, Yongwei Tang, and Dawei Zhao. GRU-based interpretable multivariate time series anomaly detection in industrial control system. *Computers & Security*, 127:103094, 2023.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, L ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, volume 30, 2017.

Dennis Wagner, Tobias Michels, Florian C.F. Schulz, Arjun Nair, Maja Rudolph, and Marius Kloft. TimeseAD: Benchmarking deep multivariate time-series anomaly detection. *TMLR*, 2023.

Phillip Wenig and Thorsten Papenbrock. Series2graph++: Distributed detection of correlation anomalies in multivariate time series. In *DAWAK*, pp. 225–230, 2024.

Phillip Wenig, Sebastian Schmidl, and Thorsten Papenbrock. Anomaly detectors for multivariate time series: The proof of the pudding is in the eating. In *ICDEW*, pp. 96–101, 2024.

Frank Wilcoxon. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80–83, 1945.

Gerald Woo, Chenghao Liu, Akshat Kumar, Caiming Xiong, Silvio Savarese, and Doyen Sahoo. Unified training of universal time series forecasting transformers. In *ICML*, volume 235 of *PMLR*, pp. 53140–53164, 21–27 Jul 2024.

Takehisa Yairi, Yoshikiyo Kato, and Koichi Hori. Fault detection by mining association rules from house-keeping data. In *i-SAIRAS*, volume 3, 2001.

Chin-Chia Michael Yeh, Yan Zhu, Liudmila Ulanova, Nurjahan Begum, Yifei Ding, Hoang Anh Dau, Diego Furtado Silva, Abdullah Mueen, and Eamonn Keogh. Matrix profile I: all pairs similarity joins for time series: a unifying view that includes motifs, discords and shapelets. In *ICDM*, pp. 1317–1322, 2016.

Shi Yunsheng, Huang Zhengjie, Feng Shikun, Zhong Hui, Wang Wenjing, and Sun Yu. Masked label prediction: Unified message passing model for semi-supervised classification. *IJCAI*, pp. 1548–1554, 08 2021.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander Smola. Deep sets. *NeurIPS*, 30, 2017.

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 *ICDM*, pp. 841–850, 2020.

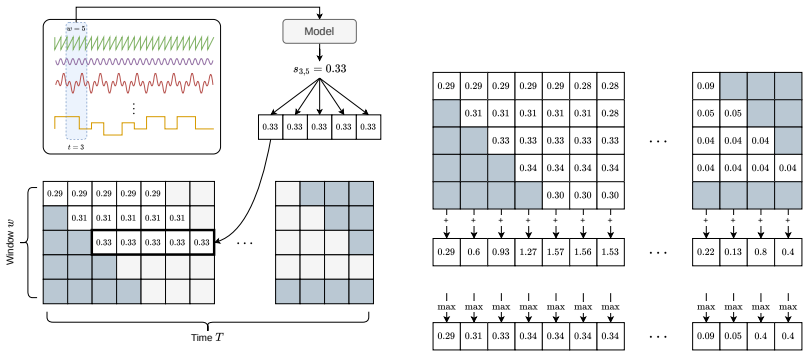


Figure 5: Scoring Aggregation and Matrix.

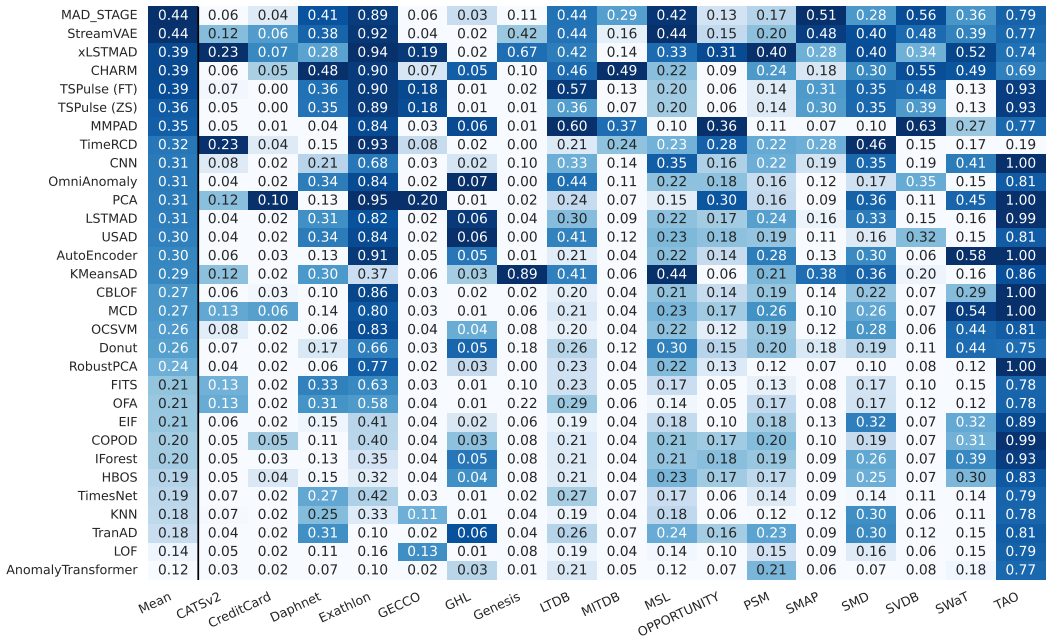


Figure 6: VUS-PR scores per suite in TSB-AD-M-Eval. Leftmost column displays mean across all datasets. Colours normalized across each row, darker colours indicate higher VUS-PR.

A Full Results

All results of Stage are obtained with the Scoring aggregation described in section 4.1 and visualized in Figure 5. In this section, we present the full results against all the baselines scored by the TSB-AD-M benchmark. Figures 6 and 7 show, respectively, the full scores across all datasets in the TSB-AD-M benchmark and the critical difference diagram of all methods. These are extended figures from the ones we presented in the main text, including methods lower than the top 10 in terms of median VUS-PR.

However, the conclusions stay the same: Stage matches StreamVAE in terms of VUS-PR, and is significantly different from all methods except StreamVAE and xLSTMAD. The large amount of baselines available for TSB-AD-M shows that it is not an easy benchmark.

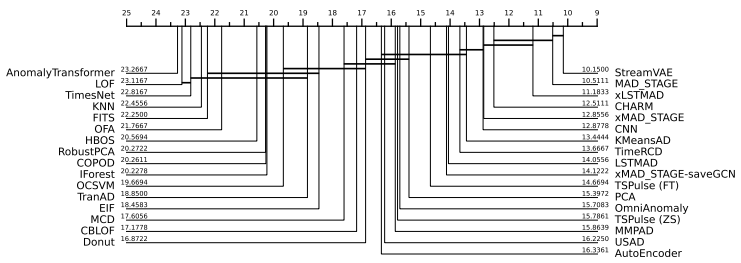


Figure 7: Critical Difference Diagram computed using VUS-PR of top models of TSB-AD-M leaderboard and STAGE. Methods ranked by average placement on all TSB-AD-M datasets. Connected methods are not statistically different.

B FULL EXPERIMENTAL SETTINGS

For the empirical results, we include the following three datasets, each aiming to provide solid foundation for different experiments and metrics.

B.1 DATASETS

TSB-AD-M. TSB-AD-M Liu & Paparrizos (2024) is a diverse set of curated multivariate time series datasets, combining 40 different datasets from many different domains. TSB-AD-M is the premier benchmark for TSAD methods and as such the focus of our accuracy analysis. We evaluate accuracy on the *Eval* subset of TSB-AD-M, consisting of 180 multivariate time series. We conduct our ablation study on the *Tuning* subset, as it is highly varied and the individual time-series posses different characteristics.

SMD. xThe ServerMachineDataset consists of data gathered from 28 different machines over a 5-week period Su et al. (2019). Notably, SMD includes labels for every anomaly detailing which dimension contribute to the anomaly. SMD is included within the TSB-AD-M benchmark, but due to the per dimension labels, we include it as a stand-alone dataset for our experiments on explainability. Per the analysis of Wagner, Michels, Schulz, *et al* Wagner et al. (2023), only a subset of SMD is used.

CoMuT. CoMuT (Correlated Multivariate Time Series)Wenig et al. (2024) is a synthetic dataset, generated specifically to test the ability of models to predict correlation anomalies. Each time series in CoMuT consists of some base dimensions and some correlated dimensions. The base dimensions have values randomly jumping between 0 and 1 and the correlated dimensions relate to this value in some way. For instance, a correlated dimension might contain the opposite value or 2 times the base value. An anomaly is then introduced by the base dimension jumping before the correlated dimensions. As the base dimension is just random jumps, this anomaly cannot be detected without considering the correlated dimensions Wenig et al. (2024). As it is always the first dimension causing the anomaly, CoMuT is suitable for explainability experiments.

B.2 METRICS

As detailed in Section 5.1, the state-of-the-art metric for detection accuracy is VUS-PR Paparrizos et al. (2022), and we base our evaluation on this metric. For explainability, we use IPS (InterPretation Score) Li et al. (2021), an established metric for multivariate anomaly detection. In short, IPS measures whether the model can identify the anomalous dimensions of a multivariate time series during an anomaly.

Let $\Phi_1, \ldots, \Phi_A$ be the ground-truth anomalies detected by the model, where each Φ_a is a contiguous interval of time points. At each time point $t \in \Phi_a$, the model assigns every dimension i an anomaly score AS_t^i, where higher values indicate stronger attribution to that dimension. The score of dimension i for the whole anomaly is its maximum over the interval, $AS_a^i = \max_{t \in \Phi_a} AS_t^i$. Let G_a be the set of ground-truth anomalous dimensions for Φ_a, let I_a be the $|G_a|$ dimensions with the

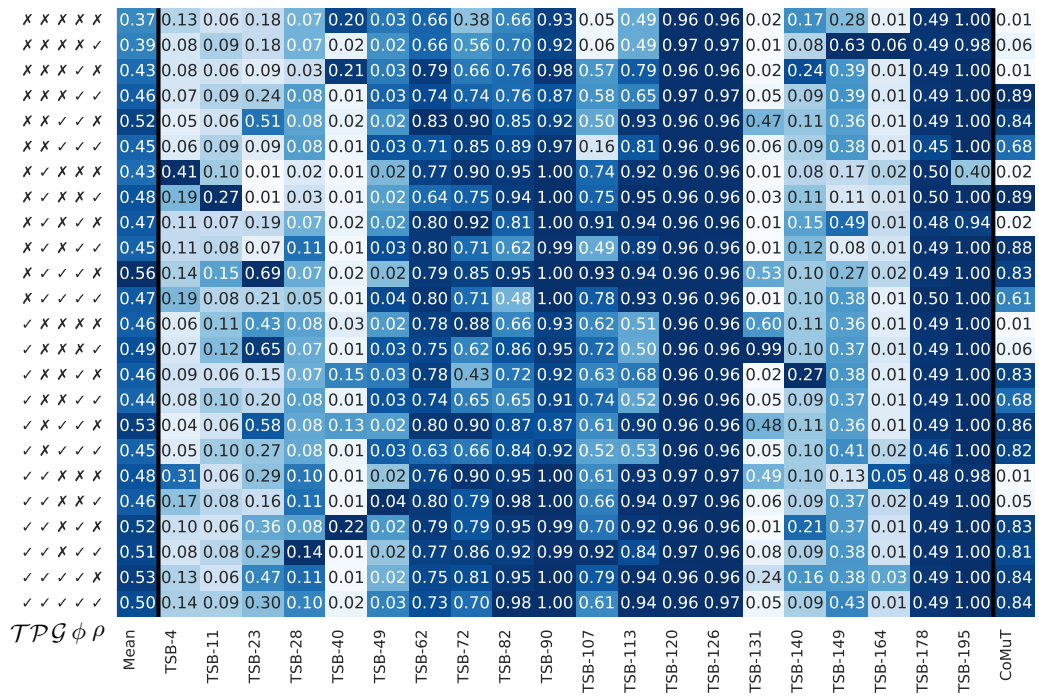


Figure 8: Result of ablation on TSB-AD-M-Tuning and CoMuT. Colours normalized across columns, darker colours indicate higher VUS-PR. $\mathcal{T}$ = Transformer layer, $\mathcal{P}$ = Pruning, $\mathcal{G}$ = GCN Layer, ϕ = DeepSet layer ϕ , ρ = DeepSet layer ρ

highest AS_a^i, and let N_a be the number of time points in Φ_a flagged as anomalous by the model. IPS is then defined as

$$\text{IPS} = \sum_{a=1}^{A} \frac{N_a}{\sum_{b=1}^{A} N_b} \cdot \frac{|G_a \cap I_a|}{|G_a|}, \tag{6}$$

i.e., the fraction of ground-truth anomalous dimensions recovered among the model's top $|G_a|$, averaged over anomalies and weighted by N_a. IPS ranges from 0 to 1. Since only detected anomalies enter the sum, missed anomalies do not affect it.

B.3 Baselines

In the accuracy evaluation, we compare against the entire TSB-AD-M leaderboard, using the scores published on the TSB-AD-M leaderboard. In the explainability evaluation, we compare against the models CNN, OmniAnomaly, AutoEncoder and LSTMAD chosen as they perform well on TSB-AD-M (CNN especially being the top performer from the original paper), and are simple to modify to output explanations, due to being prediction-based. We also run these four models on TSB-AD-M and report the scores alongside the scores from the leaderboard. The baselines are setup with the optimal hyper parameters reported by Liu & Paparrizos (2024). All experiments are run with 1000 epochs, window size of 100 and early stopping patience of 10, except the AutoEncoder, which is run for 100 epochs (as done in Liu & Paparrizos (2024)), as running for 1000 epochs is infeasible without early stopping.

C Ablations

Figure 8 shows the results of the ablation studies described in Section 5.1. We perform the ablation study on the model configuration with three attention heads, three GCN layers, and with deep set aggregator defined by a three layer MLP ϕ, a two layer MLP ρ and $d_h = 32$. We choose these parameter, as they have been the best performing across initial evaluation. In some cases the model

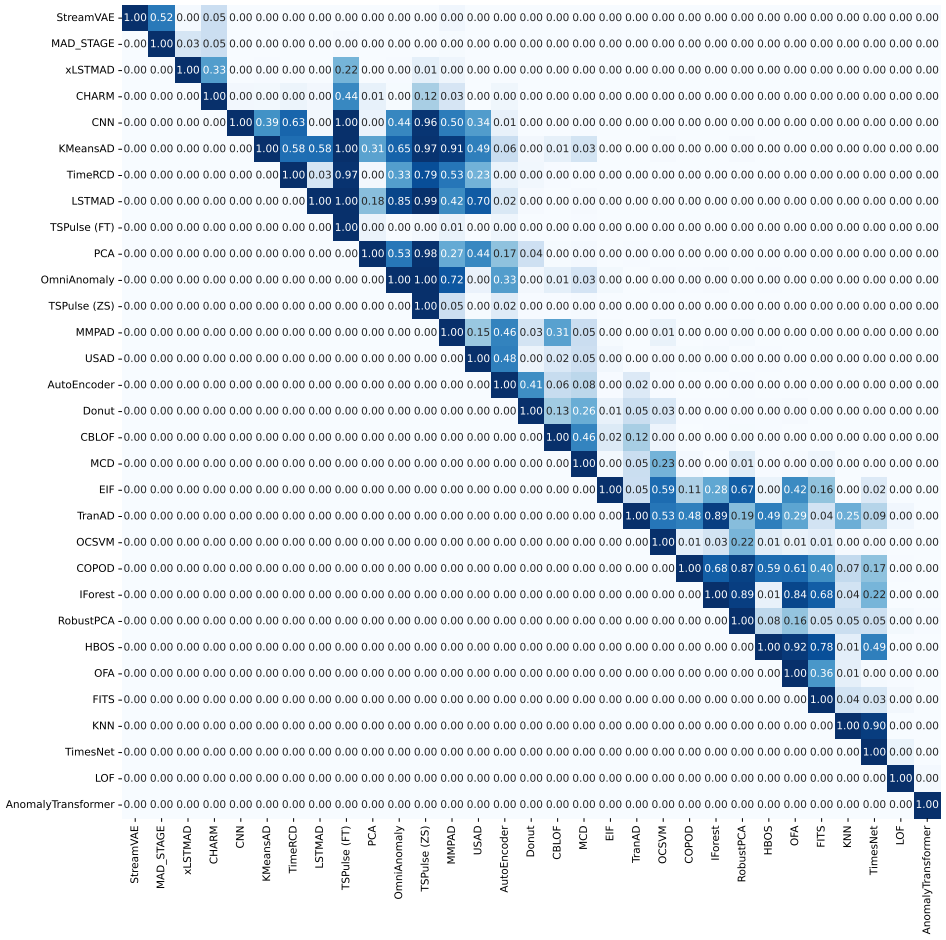


Figure 9: p-values of pair-wise Wilcoxon tests

architecture has been slightly altered, to allow for performing the ablations. Specifically, when the pruning layer is omitted, pooling is applied to the attention weights, allowing for their use as edge weights in the following layers. Additionally, when the ϕ and ρ layer of the deep set aggregation is not present, it is replaced by a single linear layer, transforming the data to the correct shapes.

We performed the ablation study on two different datasets. The CoMuT dataset and the Tuning subset of the TSB-AD-M Dataset. The combined results can be seen in Figure 8.

Observing the results, the full STAGE configuration performs consistently great. Interestingly, observing Figure 8, we do not see the full configuration performing best, nor do we see many variations that fall off significantly in accuracy. Nevertheless, there is no dataset where the full configuration performs significantly worse than the majority of configurations. At the same time there are no models that perform significantly better overall, except for the model omitting the transformer and the ρ function of the DeepSet layer (mean VUS-PR 0.56). To investigate the feasibility of this configuration, we run this variant of the model on the full Eval subset of TSB-AD-M and again on CoMuT. The variant showed comparable performance on TSB-AD-M-Eval, but achieved a VUS-PR score of only 0.02 when recomputing CoMuT, showing that this variant is less stable than the complete STAGE configuration. For comparison, the worst score achieved on CoMuT by the complete STAGE configuration, across our many hours of experiments, is 0.81. Given these results, we keep the complete STAGE configuration.

Table 1: Order invariance on CoMuT.*Top dimension* is the dimension with the largest occlusion score d_j at the highest-scoring window of the labelled event (i.e., what STAGE finds anomalous); the true culprit is 0 in the original order and 2 in the permuted one.

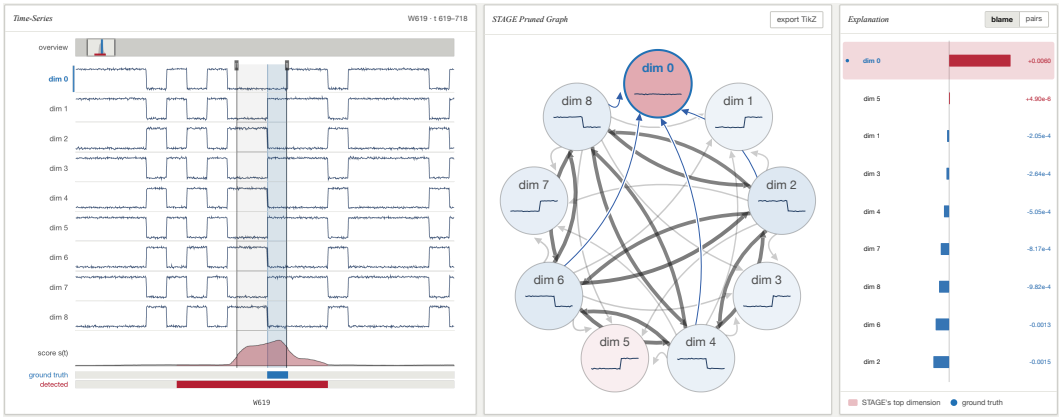
		Top dimension			
Series	Window	original	permuted	d_{top}	VUS-PR
comut_40x9_0	619	0	2	5.96×10^{-3}	0.965
comut_40x9_1	2937	0	2	6.38×10^{-3}	0.991
comut_40x9_2	2472	0	2	3.31×10^{-3}	0.984
comut_40x9_3	987	0	2	9.18×10^{-3}	0.953
comut_40x9_4	3225	0	2	6.39×10^{-3}	0.965

C.1 Ablation on Order Invariance

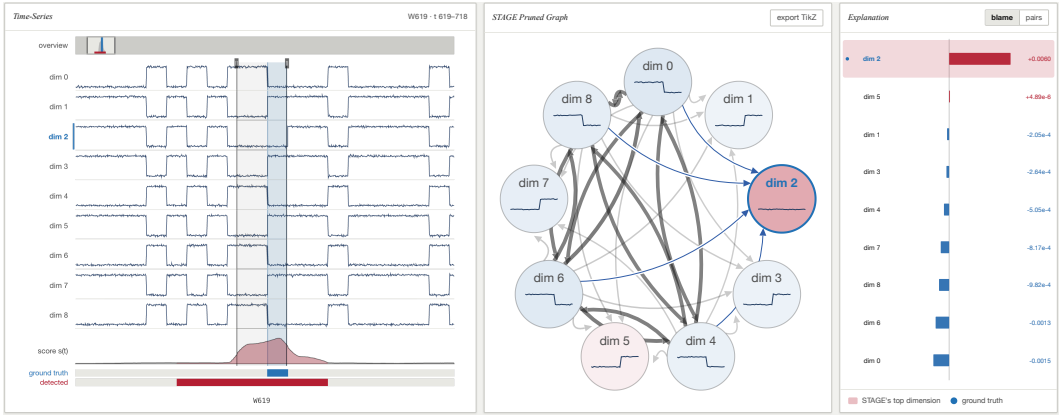
STAGE treats the dimensions of a window as a set: each one becomes a vertex of the graph, and the Deep Sets head averages over them. As a consequence, reordering the input has no effect on detection. We test this directly: on the five comut_40x9 series, the culprit of the single labelled anomaly is dimension 0. We swap dimensions 0 and 2 in the test data and run the *same* trained checkpoint on it, so the culprit is now dimension 2. Crucially, the model is not retrained; only the order in which it sees the dimensions changes.

Table 1 reports the result at the window with the highest anomaly score inside each labelled event. In every series, STAGE finds dimension 0 as anomalous in the original order and dimension 2 as anomalous in the permuted one, with the same occlusion score d_j. The full ranking of the nine dimensions is identical (minus the swap of dimensions 0 and 2). Detection is also unaffected: VUS-PR and hits@1 are identical across the two orders on all five series.

This is also visible in the visualization tool. Figure 10 shows the same setting in the live explanation tool on comut_40x9_0. In the series panel, the channel that fails to flip at $t = 680$ is dimension 0 in (a) and dimension 2 in (b). STAGE's top dimension follows it: the red vertex and the top blame bar move from dimension 0 to dimension 2, with the same score of $+0.0060$. Every other bar keeps its value, and only dimensions 0 and 2 trade places in the ranking.



(a) original order: dimension 0 is the culprit



(b) dimensions 0 and 2 swapped: dimension 2 is the culprit

Figure 10: Screenshots of the live explanation tool on `comut_40x9_0`. The labelled anomaly is $t = 680$–719 (blue band). Left: the series, anomaly score and detections; centre: the pruned graph; right: the occlusion score d_j of every dimension.