RETHINKING ENCODER LEARNING IN TIME-SERIES ANOMALY DETECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Time-series anomaly detection (TSAD) commonly learns representations from historical observations to distinguish normal from anomalous behavior. However, in the absence of anomaly supervision, learning alone faces a fundamental limitation: its objective does not directly specify which aspects of the data are actually relevant for anomaly detection. Historical observations, meanwhile, provide a concrete basis for evaluating new observations. This raises a fundamental question: if historical observations can serve directly as empirical references, is encoder learning really necessary for strong anomaly detection? As a direct test of this question, we introduce **F**ixed **A**nomaly **R**epresentation with **O**bserved References (**FARO**), a deliberately simple architecture with no encoder training. FARO fixes a randomly initialized encoder, stores historical features in channel-specific reference banks, and scores new observations by their deviation from nearest historical references. The resulting channel-wise deviations support both anomaly detection and localization. Despite its simple design, FARO achieves state-of-the-art anomaly detection performance on TSB-AD-U/M and competitive channel localization on SMD, HAI, and SWaT. More importantly, controlled experiments show that, with historical references available, encoder training provides no consistent additional gain and can even degrade detection. This pattern also holds across multiple existing TSAD backbones. Together, these findings challenge the default reliance on encoder learning in TSAD and highlight empirical references as a strong alternative for exploiting historical observations. The source code is available at AnonymizedURL.

1 INTRODUCTION

Modern real-world systems, including industrial processes, network infrastructures, and physiological monitoring platforms, continuously record their evolving states through multiple sensors and variables (Zamanzadeh Darban et al., 2024). In such systems, unexpected events or failures can directly affect operational stability and reliability, making it important to quickly identify deviations from normal behavior (Dai et al., 2024). Time-series anomaly detection (TSAD) aims to identify time points or intervals that depart from such normal regularities. In multivariate time series, it is further important not only to detect when an anomaly occurs, but also to identify which channels are associated with it, making anomaly localization an important task (Shimillas et al., 2026).

In real-world settings, anomalies are rare, and it is often impractical to label all possible anomalous behaviors in advance. Consequently, the data available for training typically consist of unlabeled historical observations (Goswami et al., 2023). Most modern TSAD methods use such data to train an encoder, with the goal of constructing a representation space that effectively captures the structure of normal time-series patterns and their differences from anomalous states. Reconstruction- and forecasting-based methods learn representations by reconstructing the input or predicting future observations and use the resulting errors as anomaly scores (Audibert et al., 2020; Deng & Hooi, 2021). Association- and contrastive-based methods encourage relationships and differences among temporal patterns to be reflected in the learned representation (Xu et al., 2022; Yang et al., 2023), while patch-memory-based methods detect anomalies by comparing learned representations against stored normal references (Park & Kang, 2026).

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.

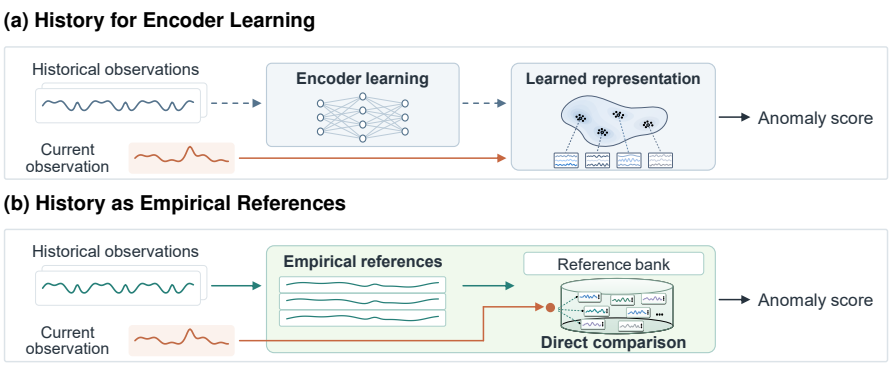

Figure 1: Historical observations in time-series anomaly detection. (a) Historical observations serve as training data for encoder learning. (b) Historical observations serve as empirical references against which current observations are evaluated.

Real-world time-series anomalies, however, are often not characterized by a fixed anomalous pattern, but instead manifest as deviations from the behavior that each time series typically exhibits. Many existing methods attempt to learn from data which signals distinguish anomalous from normal behavior and capture them in the representation. Yet in the absence of supervised anomaly information, the training objective does not directly specify which differences are actually important for anomaly detection, and there is no guarantee that training with this objective will align with the signals needed for detection (Goswami et al., 2023; Kim et al., 2026). This creates an inherent limitation in explicitly capturing detection-relevant signals.

At the same time, time series often contain recurring behaviors and local patterns (Xu et al., 2018). If a current observation is normal, a similar pattern is therefore likely to have appeared in the past, whereas anomalous observations tend to depart from such recurring behavior and differ more substantially from historical observations. In such settings, retaining behavior that has actually been observed in the past as a direct basis for comparison provides a natural alternative. Historical comparison has also been used in prior TSAD methods (Boniol et al., 2021; Park & Kang, 2026), but when empirical references already provide a direct basis for comparison, the additional role of encoder learning remains unclear.

This raises a more fundamental question: if historical observations can be directly referenced and compared, is encoder learning really necessary for strong anomaly detection?

To answer this question, we introduce Fixed Anomaly Representation with Observed References (FARO), a deliberately simple anomaly detection module. FARO uses a single randomly initialized encoder, fixed and shared across all series and channels, and retains the extracted historical features in channel-specific reference banks. Specifically, FARO measures the distance between each test patch and its nearest historical references as a channel-wise deviation and aggregates these deviations for anomaly detection and channel localization.

This simple design leads to several notable findings. FARO achieves strong performance on TSB-AD-U and TSB-AD-M (Liu & Paparrizos, 2024) without encoder training and also delivers strong anomaly localization performance on SMD, HAI, and SWaT. More importantly, extensive controlled ablations show that, once a reference bank is available, encoder training provides no consistent additional performance improvement and can even degrade detection performance in some cases. These results show that strong anomaly detection can be achieved using empirical references from each series' own historical data, without encoder learning. Additional experiments across multiple backbones further show that, when a reference bank is introduced, encoder learning does not provide a consistent additional contribution.

Our contributions are summarized as follows.

- We systematically examine whether encoder learning is necessary for strong TSAD when historical observations are retained as empirical references, and isolate the additional contribution of encoder learning under this setting.

- We propose FARO, a simple detector that requires no separate training. FARO retains historical observations in a reference bank and measures how much new observations differ from them to perform both anomaly detection and channel localization.
- We further show that, across diverse existing TSAD models on a standard benchmark, randomly initialized models using historical references outperform their trained counterparts as originally proposed, while encoder learning provides no consistent additional gain under the same reference setting.

2 Related Work

Learning from historical observations for TSAD. A prominent line of recent TSAD research trains models on historical observations and uses their outputs or learned representations to distinguish normal from anomalous behavior. Anomaly Transformer and DCdetector learn temporal associations and contrastive representations, respectively (Xu et al., 2022; Yang et al., 2023), whereas CATCH and KAN-AD model normal behavior through frequency-aware reconstruction and forecasting (Wu et al., 2025; Zhou et al., 2025). More recent work also uses pretrained time-series representations for TSAD, as in TSPulse (Ekambaram et al., 2026). Across these approaches, learning the representation or model for anomaly detection remains a central design choice.

Historical References for TSAD. A complementary line of work uses comparisons between queries and retained observations or reference patterns directly for anomaly scoring. RandomAD uses random convolutional features for anomaly detection, but still relies on target training data for feature and temporal-scale selection (Xi & Lin, 2026). PaAno combines learned patch representations with nearest-neighbor scoring against historical references (Park & Kang, 2026). We systematically evaluate whether encoder learning provides additional benefits when anomaly detection can be performed through direct comparison with historical references.

Channel localization and diagnosis. Recent work has also explored finer-grained diagnosis in multivariate TSAD, including identifying channels that exhibit anomalous behavior. SARAD learns inter-feature spatial associations and performs diagnosis using variable-wise reconstruction errors (Dai et al., 2024), while ALoRa-Loc derives contribution weights from a trained Transformer to account for how anomalous inputs propagate across reconstructed variables (Shimillas et al., 2026). AERCA further focuses explicitly on root-cause analysis (Han et al., 2025). FARO instead localizes affected channels using channel-specific empirical references, measuring each channel against its own history without learning inter-variable structure.

3 FARO: Fixed Encoder with Empirical References

Given the unlabeled training history of each series, FARO (Figure 2) uses a randomly initialized, fixed encoder to extract window features and stores them in channel-specific reference banks. It compares query features with these references to detect anomalies and localize anomalous channels.

3.1 Fixed Time-Frequency Encoder

To compare current and historical windows without target-specific encoder learning, FARO maps them into a shared time–frequency representation. The temporal branch emphasizes window shape, while the frequency branch represents spectral amplitude patterns.

For a time series with C channels, let $\mathbf{p}_{n,c} \in \mathbb{R}^L$ denote the length-L window from channel c starting at index n. Within each run, a single encoder f_{θ_0} with randomly initialized parameters θ_0 processes reference and query windows across all series and channels, with no weight or running-statistics updates. It combines temporal and spectral features:

$$\begin{aligned} \mathbf{h}^{\mathrm{T}}_{n,c} &= g^{\mathrm{T}}_{\theta_0}(\mathrm{Standardize}(\mathbf{p}_{n,c}))\,, \\ \mathbf{h}^{\mathrm{F}}_{n,c} &= g^{\mathrm{F}}_{\theta_0}(\log(1+|\mathrm{rFFT}(\mathbf{p}_{n,c})|))\,, \\ \mathbf{z}_{n,c} &= f_{\theta_0}(\mathbf{p}_{n,c}) = \mathrm{Norm}\big([\mathbf{h}^{\mathrm{T}}_{n,c} \| \mathbf{h}^{\mathrm{F}}_{n,c}]\big)\,. \end{aligned} \tag{1}$$

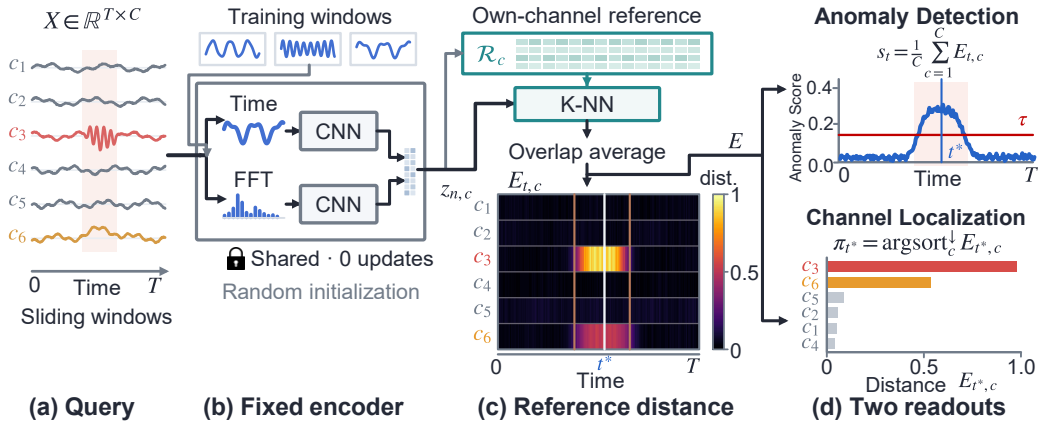

Figure 2: FARO overview. (a–b) A shared, randomly initialized fixed encoder extracts time–frequency features from query and training windows. (c) Distances to nearest references from the same series and channel are averaged over overlapping windows. (d) Averaging scores across channels gives detection scores; ranking channels provides localization.

Here, $\mathbf{h}^{\mathrm{T}}_{n,c}, \mathbf{h}^{\mathrm{F}}_{n,c} \in \mathbb{R}^D$ are the normalized temporal and frequency features of dimension D; $\mathbf{z}_{n,c} \in \mathbb{R}^{2D}$ is their combined representation. $\|$ denotes concatenation, Norm denotes ℓ_2 normalization, and rFFT denotes the real Fourier transform.

$\mathrm{Standardize}$ subtracts each window's mean and divides by its standard deviation to emphasize temporal shape. The frequency branch uses the raw window's log-amplitude spectrum. Each branch mapping g applies fixed convolutions, global average pooling, a fixed projection, and ℓ_2 normalization, giving the two views equal weight in the subsequent reference comparison. Architecture and dimension settings are given in Appendix A.1. The resulting fixed representation is then used to compare each query against its historical references.

3.2 Empirical Reference Scoring

FARO evaluates each query window against its most similar historical references from the same series and channel. We therefore retain a separate reference bank for each channel c:

$$\mathcal{R}_c = (\mathbf{z}_{m,c})_{m \in \mathcal{I}}, \qquad |\mathcal{I}| \le B, \tag{2}$$

where $\mathcal{I}$ contains the start indices of retained training windows and B is the maximum number of references per channel. Here, n and m denote query and reference window starts, respectively. Let $\mathcal{N}_k(\mathbf{z}_{n,c}; \mathcal{R}_c)$ denote the start indices of the k nearest eligible references to $\mathbf{z}_{n,c}$ under cosine distance. The mean distance to these references defines the window-level deviation, with larger values indicating greater departures from historical behavior:

$$d_{n,c} = \frac{1}{k} \sum_{m \in \mathcal{N}_k(\mathbf{z}_{n,c}; \mathcal{R}_c)} \left(1 - \cos(\mathbf{z}_{n,c}, \mathbf{z}_{m,c})\right). \tag{3}$$

To obtain a channel-level anomaly score at each timestamp, we average the deviations of windows covering that timestamp. Let $\mathcal{W}(t)$ denote their start indices. The resulting score is:

$$E_{t,c} = \frac{1}{|\mathcal{W}(t)|} \sum_{n \in \mathcal{W}(t)} d_{n,c}. \tag{4}$$

The same channel-wise deviations yield the detection score s_t by averaging and the localization ranking π_t by sorting in descending order:

$$s_t = \frac{1}{C} \sum_{c=1}^{C} E_{t,c} \quad \text{(detection)}, \qquad \pi_t = \text{argsort}_c^{\downarrow} E_{t,c} \quad \text{(localization)}. \tag{5}$$

Reference selection and timestamp-scoring details are given in Appendix A.2.

4 EXPERIMENTS

We first evaluate whether FARO can achieve strong anomaly detection and channel localization using historical references without encoder learning. We then examine whether historical-reference comparison also benefits existing model backbones, followed by ablation studies assessing the additional contribution of encoder learning and individual design choices.

4.1 EXPERIMENTAL SETUP

Benchmarks and metrics. We evaluate detection on all 350 univariate series in TSB-AD-U and 180 multivariate series in TSB-AD-M (Liu & Paparrizos, 2024). We use VUS-PR as the primary metric, complemented by range-based metrics (VUS-ROC and Range-F1) and point-wise metrics (AUC-PR, AUC-ROC, and Point-F1) (Paparrizos et al., 2022). We evaluate channel localization on SMD, HAI, and SWaT using Hit Rate (HR), Normalized Discounted Cumulative Gain (NDCG), and Interpretation Score (IPS) (Shimillas et al., 2026). HR and NDCG assess affected-channel recovery and ranking at anomalous timestamps, while IPS evaluates recovery over anomaly intervals. We select the top-ranked channels at 100% and 150% of the annotated affected-channel count. Dataset and metric details are in Appendix B.

Baselines. We compare FARO with classical statistical and machine-learning methods such as PCA (Aggarwal, 2017), KMeansAD (Liu & Paparrizos, 2024), and KShapeAD (Paparrizos & Gravano, 2015; Liu & Paparrizos, 2024); recent neural models such as CATCH (Wu et al., 2025), KAN-AD (Zhou et al., 2025), and PaAno (Park & Kang, 2026); pretrained time-series models such as MOMENT-FT (Goswami et al., 2024), TSPulse-FT (Ekambaram et al., 2026), and Time-RCD (Lan et al., 2026); and the random-feature-based detector RandomAD (Xi & Lin, 2026). Full baseline lists, result sources, evaluation protocols, and implementation details are provided in Appendix B.

FARO configuration. FARO uses unlabeled historical data, windows of length 96, three nearest references, and at most 3,000 reference features per channel. We report FARO's main results averaged over 10 independently initialized encoders. FARO's main detection results follow the released TSB-AD evaluation procedure (Liu & Paparrizos, 2024), consistent with PaAno (Park & Kang, 2026). Evaluation intervals and baseline-specific settings are detailed in Appendix B.2. Ablations in Section 4.4 use matched query segments.

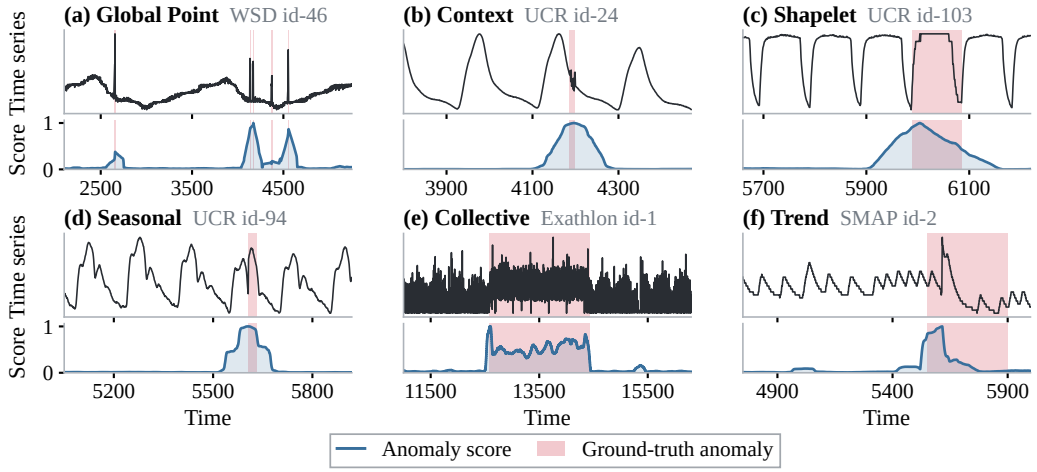

Figure 3: Qualitative examples of FARO. Additional multivariate examples are provided in Appendix F.

Table 1: Detection performance on the complete TSB-AD-U/M tracks. FARO results are averaged over 10 seeds. **Bold** and underlining indicate the best and second-best performance for each metric within each track, respectively. The full comparison and evaluation protocols are in Appendices C.1 and B.

	Method	VUS-PR	VUS-ROC	Range-F1	AUC-PR	AUC-ROC	Point-F1
TSB-AD-U: 350 univariate series							
Classical	Sub-PCA (2017)	0.42	0.76	0.41	0.37	0.71	0.42
	KShapeAD (2015)	0.40	0.76	0.40	0.35	0.74	0.39
Learned	DLinear (2023)	0.39	0.84	0.36	0.38	0.78	0.43
	PatchTST (2023)	0.43	0.84	0.40	0.43	0.81	0.47
	iTransformer (2024)	0.42	0.82	0.39	0.41	0.78	0.45
	KAN-AD (2025)	0.33	0.77	0.35	0.30	0.70	0.35
	PaAno (2026)	0.53	0.89	0.49	0.47	0.87	0.52
Pretrained	MOMENT-FT (2024)	0.39	0.76	0.35	0.30	0.69	0.35
	TSPulse-FT (2026)	0.55	0.87	0.52	0.48	0.85	0.55
	Time-RCD (2026)	0.52	0.87	0.47	0.50	0.84	0.54
Fixed	RandomAD (2026)	0.43	0.86	0.35	0.33	0.81	0.40
	FARO (ours)	**0.57**	**0.90**	**0.54**	**0.51**	**0.88**	**0.56**
TSB-AD-M: 180 multivariate series							
Classical	KMeansAD (2024)	0.29	0.73	0.33	0.25	0.69	0.31
	PCA (2017)	0.31	0.74	0.29	0.31	0.70	0.37
Learned	DLinear (2023)	0.35	0.74	0.26	0.35	0.72	0.40
	PatchTST (2023)	0.39	0.72	0.38	0.33	0.67	0.41
	iTransformer (2024)	0.35	0.72	0.26	0.34	0.70	0.40
	ModernTCN (2024)	0.35	0.73	0.26	0.34	0.69	0.38
	CATCH (2025)	0.33	0.74	0.26	0.32	0.71	0.37
	KAN-AD (2025)	0.33	0.76	0.29	0.34	0.74	0.39
	ScatterAD (2025)	0.24	0.63	0.22	0.21	0.58	0.27
	PaAno (2026)	0.43	0.79	0.41	0.38	0.76	0.43
Pretrained	TSPulse-FT (2026)	0.39	0.72	0.46	0.41	0.72	0.46
	Time-RCD (2026)	0.32	0.71	0.27	0.34	0.72	0.39
Fixed	**FARO (ours)**	**0.48**	**0.81**	**0.47**	**0.44**	**0.79**	**0.49**

4.2 Main Results

Anomaly detection. FARO achieves the best performance among the compared methods across all six detection metrics on both complete TSB-AD tracks (Table 1). These results show that historical references can support strong anomaly detection without encoder learning. The full comparison and per-seed results are provided in Appendix C. Figure 3 illustrates FARO's anomaly scores across different anomaly types.

Table 2: Channel localization on SMD, HAI, and SWaT. We report HR@$P\%$, NDCG@$P\%$, and IPS@$P\%$ at $P \in \{100, 150\}$. **Bold** and underlining mark the best and second-best performance in each column, respectively. All baselines use released implementations; evaluation details are in Appendix B.2.

	SMD						**HAI**						**SWaT**					
Method	HR@$P\%$		NDCG@$P\%$		IPS@$P\%$		HR@$P\%$		NDCG@$P\%$		IPS@$P\%$		HR@$P\%$		NDCG@$P\%$		IPS@$P\%$	
P	100	150	100	150	100	150	100	150	100	150	100	150	100	150	100	150	100	150
TranAD	0.40	0.51	0.42	0.48	**0.73**	**0.86**	0.05	0.07	0.05	0.06	0.12	0.18	0.03	0.04	0.03	0.04	0.13	0.14
ATF-UAD	0.34	0.45	0.36	0.42	0.64	0.76	0.03	0.05	0.04	0.05	0.05	0.06	0.03	0.05	0.03	0.04	0.07	0.09
USAD	0.36	0.47	0.38	0.44	0.58	0.72	0.05	0.08	0.04	0.06	0.11	0.17	0.05	0.08	0.05	0.07	0.19	0.28
GDN	0.43	0.56	0.46	0.54	0.68	0.82	0.05	0.08	0.05	0.07	0.06	0.11	0.11	0.15	0.12	0.14	0.18	0.26
SARAD	0.56	0.71	0.60	0.69	0.61	0.76	0.03	0.08	0.03	0.06	0.06	0.10	0.09	0.11	0.09	0.10	0.26	0.41
AERCA	0.32	0.45	0.32	0.40	0.58	0.71	0.09	0.12	0.09	0.11	0.14	0.19	0.05	0.06	0.05	0.06	0.16	0.22
DAEMON	0.18	0.32	0.19	0.27	0.17	0.25	0.04	0.05	0.04	0.05	0.05	0.06	0.02	0.03	0.02	0.03	0.01	0.02
ALoRa-Loc	0.38	0.52	0.39	0.48	0.39	0.56	0.04	0.07	0.04	0.05	0.04	0.06	0.01	0.01	0.01	0.01	0.04	0.06
FARO (ours)	**0.66**	**0.78**	**0.68**	**0.76**	0.58	0.73	**0.18**	**0.26**	**0.19**	**0.24**	**0.20**	**0.28**	**0.15**	**0.16**	**0.16**	**0.16**	**0.51**	**0.56**

Channel localization. FARO achieves competitive channel localization performance across SMD, HAI, and SWaT (Table 2). Unlike approaches that learn cross-channel relationships or derive model-

based attributions, FARO directly uses its channel-wise deviations for localization. Averaging these deviations produces the overall anomaly score, while ranking them identifies the channels with the strongest anomalous behavior. Detection and localization therefore use the same evidence, without additional supervision or a separate localization model. Qualitative examples are provided in Appendix F.

4.3 Historical References for Existing Models

To test whether encoder learning adds value when historical observations can be directly compared, we apply reference-based detection to existing models with different architectures and training objectives. Within each model, trained and randomly initialized versions use the same historical observations as references. We also compare reference-based detection with each trained model's original method. Model-specific settings are in Appendix G.

Table 3: Mean VUS-PR over three seeds on all 180 TSB-AD-M and 350 TSB-AD-U series. Untrained models retain their random initialization. CATCH requires multivariate input (—).

Model	TSB-AD-M			TSB-AD-U		
	Original	Reference-based		Original	Reference-based	
	Trained	Trained	Untrained	Trained	Trained	Untrained
KAN-AD	0.334	**0.431**	0.430	0.319	0.393	**0.398**
ModernTCN	0.346	0.433	**0.435**	0.320	0.476	**0.476**
ScatterAD	0.240	0.329	**0.331**	0.259	**0.384**	0.379
CATCH	0.330	**0.412**	0.405	—	—	—

Historical-reference detection improves performance over each trained model's original method in every reported model–track comparison (Table 3). Randomly initialized models with references also outperform trained models using their original method. When historical observations are compared in the same way, encoder learning provides no consistent additional gain.

PaAno (Park & Kang, 2026) already uses historical references but still trains its encoder. To isolate encoder learning, we keep its reference method unchanged and compare full training with Untrained (BN-only), where encoder weights remain at random initialization and only BatchNorm statistics are updated. This condition performs comparably to full training and better on one track, indicating no consistent additional benefit from encoder learning (Table 4).

Table 4: Training and BN calibration in PaAno. Mean VUS-PR over three seeds on TSB-AD-M/U.

Condition	M	U
Trained	**0.427**	0.530
Untrained (BN-only)	0.407	**0.546**

Together, these results show that historical observations are not merely data used to train an encoder, but can themselves provide strong anomaly signals across diverse architectures without target-specific encoder learning. Across these comparisons, encoder learning provides no consistent additional benefit in exploiting the same historical observations.

4.4 Ablation Studies

Adding encoder learning to FARO. We assess the additional contribution of encoder learning within FARO by training its encoder under five representative objectives. Fixed and trained encoders start from the same initialization and are evaluated using identical reference windows, test windows, and scoring procedures, differing only in whether the encoder is trained. After training, reference and test features are recomputed using the trained encoder.

Table 5: Encoder learning in FARO. VUS-PR changes after training with five objectives (trained minus matched fixed encoder) on TSB-AD-M/U. Details are in Appendix D.

Track	Masked recon.	InfoNCE	Triplet	Forecasting	One-class
M	-0.0037	-0.0052	-0.0061	-0.0027	-0.0001
U	-0.0024	-0.0030	-0.0034	$+0.0035$	-0.0053

None of the five objectives improves detection performance on both tracks relative to the matched fixed encoder (Table 5).

On a predefined TSB-AD-U subset, all five objectives reduce validation loss during training, but this does not consistently improve detection on those series. The VUS-PR gain varies across checkpoints and can decline with further training (Figure 4). Thus, lower objective loss does not consistently lead to better anomaly detection with reference windows fixed. Details are in Appendix D.

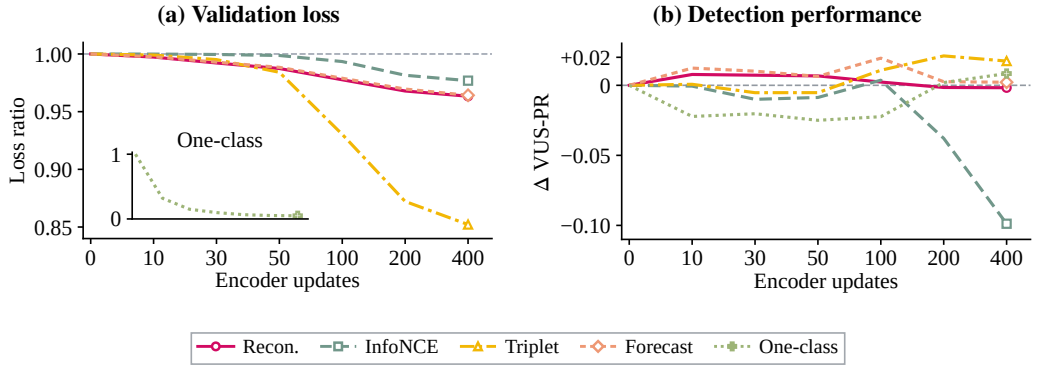

Figure 4: Training trajectories on a predefined TSB-AD-U subset. (a) Validation loss and (b) detection performance across checkpoints. The one-class loss appears in an inset. Details are in Appendix D.

Table 6: Feature and reference choices in FARO. We report mean VUS-PR on M/U and HR@100% on SMD.

(a) Feature views

Features	M	U	SMD
Time + freq.	**0.485**	**0.583**	**0.663**
Time only	0.409	0.426	0.511
Freq. only	0.481	0.551	0.659

(b) Channel specificity

Bank	M	SMD
Own channel	**0.485**	**0.663**
Pooled channels	0.464	0.632
Other channels	0.279	0.278

(c) Reference scoring

Method	M	U	SMD
k-NN	**0.485**	**0.583**	0.663
Centroid	0.411	0.483	0.584
Diag. Gaussian	0.430	0.535	**0.675**

Having examined the role of encoder learning, we next evaluate the design choices underlying FARO (Table 6).

Feature views. The frequency view performs better than the temporal view alone on both tracks. Combining the two gives the best detection performance on both tracks and the best SMD localization, suggesting that temporal features complement the stronger frequency view.

Channel specificity. Own-channel references perform best on TSB-AD-M and SMD. Pooling all channels causes a smaller decline than excluding the evaluated channel's history, showing the value of comparing each channel with its own past.

Reference scoring. Empirical k-NN scoring outperforms centroid and diagonal-Gaussian alternatives on both detection tracks, supporting comparison with nearby historical instances over the tested global summaries. The diagonal-Gaussian alternative gives the highest HR@100% on SMD.

Additional encoder-construction and reference-source comparisons and sensitivity analyses are reported in Appendix E.

4.5 Computational Efficiency

To examine the computational cost of exploiting historical observations as empirical references without target-specific encoder learning, we profile three TSB-AD-M workloads spanning different sequence lengths and channel counts (Appendix E.3). Preparation includes all target-specific processing before scoring, while inference denotes the first complete scoring pass over the full series; peak memory is measured during inference.

Table 7: Mean computational cost across three profiled TSB-AD-M workloads. Lower is better.

Method	Preparation ↓ (s)	Inference ↓ (s)	Total runtime ↓ (s)	Peak inference memory ↓ (MiB)
FARO	**0.557**	**0.334**	**0.891**	139.3
PaAno	5.054	0.407	5.461	172.8
KAN-AD	4.830	0.447	5.277	**82.3**
CATCH	44.072	0.976	45.048	837.3
TSPulse-FT	1,576.761	294.448	1,871.209	6,801.1

FARO has the lowest mean preparation, inference, and total runtime among the compared detectors, whereas KAN-AD uses the least peak inference memory (Table 7). These results complement our main finding by showing that historical observations can be exploited as empirical references without target-specific encoder learning while remaining computationally efficient.

5 Scope of Empirical References

Our results show that historical observations can be used directly as empirical references without target-specific encoder learning. The scope of this approach, however, depends on whether the retained references capture the behavior relevant to anomaly detection and remain representative as the underlying system evolves.

Relational structure. FARO compares each channel with its own historical behavior and does not explicitly model cross-channel dependencies. It may therefore be less sensitive to anomalies expressed primarily through changes in relationships across channels. Extending empirical references to capture cross-channel context is a natural direction.

Adaptive reference management. FARO constructs a reference bank from the available historical observations and uses it as the basis for comparison. In long-term monitoring, changes in operating regimes or normal behavior may make the initial references less representative of current behavior. Developing adaptive mechanisms that update and maintain the reference bank as new behavior is observed is therefore an important extension.

Together, these considerations highlight empirical reference design as a central research direction in TSAD: what historical behavior to retain, which relationships to represent, and how to maintain useful references as systems evolve.

6 Conclusion

FARO achieves strong anomaly detection and channel localization without target-specific encoder learning, by directly leveraging historical observations as empirical references. Across existing models, randomly initialized models that directly compare against historical observations outperform their trained counterparts under the original detection schemes. More importantly, once a reference bank is available, encoder learning provides no consistent additional performance gain. These findings indicate that previously observed behavior itself can serve as a strong basis for evaluating new observations. Therefore, rather than treating encoder learning as a default assumption in TSAD, future research should more systematically explore how historical observations themselves can be retained and exploited through reference banks as a primary source of anomaly evidence.

AI USE STATEMENT

Generative AI tools were used for information retrieval and literature discovery, limited feedback on methodological and experimental choices, including feedback on experimental design, and assistance with manuscript organization, language editing, and refinement. All AI-generated feedback related to methodology and experiments was independently evaluated by the authors, and all final methodological and experimental decisions were made by the authors. The analyses, interpretation of experimental results, and conclusions were conducted and determined by the authors. All AI-assisted material was reviewed by the authors, who take full responsibility for the final work.

ETHICS STATEMENT

This work studies time-series anomaly detection using existing benchmark datasets. We do not collect new data from human participants.

REPRODUCIBILITY STATEMENT

The anonymous code repository linked in the abstract provides the FARO implementation, experiment settings, and numerical results. Appendix A details FARO's implementation and configuration, while Appendix B describes the datasets, evaluation protocols, baseline settings, result sources, and hyperparameter search spaces. Additional experimental settings and per-seed results appear in the remaining appendices.

REFERENCES

Charu C. Aggarwal. *Outlier Analysis*. Springer, second edition, 2017. doi: 10.1007/978-3-319-47578-3.

Julien Audibert, Pietro Michiardi, Frédéric Guyard, Sébastien Marti, and Maria A. Zuluaga. USAD: Unsupervised anomaly detection on multivariate time series. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 3395–3404, 2020.

Paul Boniol, John Paparrizos, Themis Palpanas, and Michael J. Franklin. SAND: Streaming subsequence anomaly detection. *Proceedings of the VLDB Endowment*, 14(10):1717–1729, June 2021. doi: 10.14778/3467861.3467863.

Xuanhao Chen, Liwei Deng, Feiteng Huang, Chengwei Zhang, Zongquan Zhang, Yan Zhao, and Kai Zheng. DAEMON: Unsupervised anomaly detection and interpretation for multivariate time series. In *IEEE 37th International Conference on Data Engineering*, pp. 2225–2230, 2021.

Zhihao Dai, Ligang He, Shuang-Hua Yang, and Matthew Leeke. SARAD: Spatial association-aware anomaly detection and diagnosis for multivariate time series. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Angus Dempster, Daniel F. Schmidt, and Geoffrey I. Webb. MiniRocket: A very fast (almost) deterministic transform for time series classification. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 248–257, 2021. doi: 10.1145/3447548.3467231.

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

Vijay Ekambaram, Subodh Kumar, Arindam Jati, Sumanta Mukherjee, Tomoya Sakai, Pankaj Dayama, Wesley M. Gifford, and Jayant Kalagnanam. TSPulse: Tiny pre-trained models with disentangled representations for rapid time-series analysis. In *The Fourteenth International Conference on Learning Representations*, 2026.

Jin Fan, Zehao Wang, Huifeng Wu, Danfeng Sun, Jia Wu, and Xin Lu. An adversarial time–frequency reconstruction network for unsupervised anomaly detection. *Neural Networks*, 168: 44–56, 2023. doi: 10.1016/j.neunet.2023.09.018.

Mononito Goswami, Cristian Ignacio Challu, Laurent Callot, Lenon Minorics, and Andrey Kan. Unsupervised model selection for time series anomaly detection. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=gOZ_pKANaPW.

Mononito Goswami, Konrad Szafer, Arjun Choudhry, Yifu Cai, Shuo Li, and Artur Dubrawski. MOMENT: A family of open time-series foundation models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 16115–16152, 2024.

Xiao Han, Saima Absar, Lu Zhang, and Shuhan Yuan. Root cause analysis of anomalies in multivariate time series through Granger causal discovery. In *The Thirteenth International Conference on Learning Representations*, 2025.

Jongwon Kim, Samuel Yoon, Young Myoung Ko, Yerin Kim, Sung Il Kim, and Jaeung Tae. Complexity- and statistics-guided anomaly detection in time series foundation models. In *International Conference on Learning Representations*, 2026.

Taesung Kim, Jinhee Kim, Yunwon Tae, Cheonbok Park, Jang-Ho Choi, and Jaegul Choo. Reversible instance normalization for accurate time-series forecasting against distribution shift. In *International Conference on Learning Representations*, 2022.

Tian Lan, Hao Duong Le, Jinbo Li, Wenjun He, Meng Wang, Chenghao Liu, and Chen Zhang. Towards foundation models for zero-shot time series anomaly detection: Leveraging synthetic data and relative context discrepancy. In *International Conference on Machine Learning*, 2026.

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

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. iTransformer: Inverted transformers are effective for time series forecasting. In *International Conference on Learning Representations*, 2024.

DongHao Luo and Xue Wang. ModernTCN: A modern pure convolution structure for general time series analysis. In *The Twelfth International Conference on Learning Representations*, 2024.

Mohsin Munir, Shoaib Ahmed Siddiqui, Andreas Dengel, and Sheraz Ahmed. DeepAnT: A deep learning approach for unsupervised anomaly detection in time series. *IEEE Access*, 7:1991–2005, 2019. doi: 10.1109/ACCESS.2018.2886457.

Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In *The Eleventh International Conference on Learning Representations*, 2023.

John Paparrizos and Luis Gravano. k-Shape: Efficient and accurate clustering of time series. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, pp. 1855–1870, 2015.

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.

Jinju Park and Seokho Kang. PaAno: Patch-based representation learning for time-series anomaly detection. In *International Conference on Learning Representations*, 2026.

Qichao Shentu, Beibu Li, Kai Zhao, Yang Shu, Zhongwen Rao, Lujia Pan, Bin Yang, and Chenjuan Guo. Towards a general time series anomaly detector with adaptive bottlenecks and dual adversarial decoders. In *The Thirteenth International Conference on Learning Representations*, 2025.

Charalampos Shimillas, Kleanthis Malialis, Konstantinos Fokianos, and Marios M. Polycarpou. Low rank transformer for multivariate time series anomaly detection and localization. In *International Conference on Learning Representations*, 2026.

Ya Su, Youjian Zhao, Chenhao Niu, Rong Liu, Wei Sun, and Dan Pei. Robust anomaly detection for multivariate time series through stochastic recurrent neural network. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 2828–2837, 2019.

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

Xingjian Wu, Xiangfei Qiu, Zhengyu Li, Yihang Wang, Jilin Hu, Chenjuan Guo, Hui Xiong, and Bin Yang. CATCH: Channel-aware multivariate time series anomaly detection via frequency patching. In *International Conference on Learning Representations*, 2025.

Wenjie Xi and Jessica Lin. RandomAD: A random kernel-based anomaly detector for time series. In *Machine Learning and Knowledge Discovery in Databases. Research Track*, volume 16013 of *Lecture Notes in Computer Science*, pp. 159–175. Springer, 2026.

Haowen Xu, Wenxiao Chen, Nengwen Zhao, Zeyan Li, Jiahao Bu, Zhihan Li, Ying Liu, Youjian Zhao, Dan Pei, Yang Feng, Jie Chen, Zhaogang Wang, and Honglin Qiao. Unsupervised anomaly detection via variational auto-encoder for seasonal KPIs in web applications. In *Proceedings of the 2018 World Wide Web Conference*, pp. 187–196, 2018.

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

Yiyuan Yang, Chaoli Zhang, Tian Zhou, Qingsong Wen, and Liang Sun. DCdetector: Dual attention contrastive representation learning for time series anomaly detection. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 3033–3045, 2023.

Tao Yin, Shaochen Fu, Zhibin Zhang, Li Huang, Xiaohong Zhang, Yiyuan Yang, Kaixiang Yang, and Meng Yan. ScatterAD: Temporal-topological scattering mechanism for time series anomaly detection. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Zahra Zamanzadeh Darban, Geoffrey I. Webb, Shirui Pan, Charu C. Aggarwal, and Mahsa Salehi. Deep learning for time series anomaly detection: A survey. *ACM Computing Surveys*, 57(1):15, 2024. doi: 10.1145/3691338.

Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 11121–11128, 2023.

Quan Zhou, Changhua Pei, Fei Sun, Han Jing, Zhengwei Gao, Haiming Zhang, Gaogang Xie, Dan Pei, and Jianhui Li. KAN-AD: Time series anomaly detection with Kolmogorov–Arnold networks. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 79136–79149. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/zhou25u.html.

A FARO IMPLEMENTATION

A.1 FIXED ENCODER CONSTRUCTION

Each branch uses four one-dimensional convolutional layers with widths $1 \to 6 \to 12 \to 6 \to 8$, kernel sizes $7, 5, 3, 3$, ReLU activations, and global average pooling. The temporal and frequency inputs follow Eq. 1.

Both branches apply the same fixed projection to their pooled vector $\mathbf{u} \in \mathbb{R}^8$:

$$m(\mathbf{u}) = \mathrm{LayerNorm}(W\mathbf{u} + \mathbf{b}). \tag{6}$$

The projection uses a constant zero anchor and is shared across all series and channels. Each branch output is ℓ_2-normalized; concatenation and a final normalization yield a $2D = 16$-dimensional feature with $D = 8$ per branch.

The branches have independently initialized random convolutional weights. Construction uses deterministic prefix extraction from a randomly initialized 128-dimensional parent network, with fan-in correction. Evaluation-mode BatchNorm is folded into the convolutions before scoring. All coefficients and BatchNorm running statistics remain fixed. The two branches store 1,588 coefficients; the projection and LayerNorm store 88, totaling 1,676. Seeds 2027–2036 define the ten benchmark encoders.

A.2 WINDOW SAMPLING AND REFERENCE SCORING

For a concatenated training–test timeline of length T_i, window starts use stride

$$q_i = \min\{L, \max(1, \lceil T_i/40{,}000 \rceil)\}. \tag{7}$$

Only complete windows are formed, and the final start $T_i - L$ is included when absent from the regular grid. Reference windows lie wholly within the training prefix, using the corresponding stride and final complete start. We select up to B references at evenly spaced positions in the ordered candidate list. For candidate starts $a_{i,0}, \ldots, a_{i,P_i-1}$ and $K_i = \min(B, P_i)$, the selected indices are

$$\mathcal{I}_i = \left\{ a_{i,\lfloor j(P_i-1)/(K_i-1) \rfloor} : j = 0, \ldots, K_i - 1 \right\}, \qquad K_i > 1. \tag{8}$$

For $K_i = 1$, the first candidate is retained. All channels use the same selected time indices and maintain separate feature banks.

Cosine 3-NN scoring excludes a reference with the same absolute window start as the query. Repeated patterns at different starts remain eligible. Window scores are averaged at each covered timestamp as in Eq. 4.

A.3 CONFIGURATION SELECTION

The configuration search used the official TSB-AD tuning collections (48 univariate and 20 multivariate series), with VUS-PR as the detection criterion. The selected configuration was then fixed across evaluation series. Table 8 records the candidate sets and selected values; sensitivity analyses are reported separately in Appendix E.2.

Table 8: FARO configuration search on the TSB-AD tuning collections.

Setting	Candidates	Selected
Stored encoder coefficients	896, 1,676, 2,712, 4,004, 5,552	1,676
Window length L	48, 96, 192	96
Neighbors k	1, 3, 5	3
Reference cap B per channel	500, 1,000, 3,000, 6,000	3,000
Mixer input	Target anchor, zero anchor, direct	Zero anchor
Detection aggregation	Mean, top-quartile mean, maximum	Mean

B Datasets and Evaluation

B.1 Datasets, Splits, and Label Access

Detection uses the complete TSB-AD-U/M tracks (Liu & Paparrizos, 2024); channel localization uses SMD, HAI, and SWaT (Table 9). Each TSB-AD evaluation series has a predefined training prefix. FARO and the controlled experiments use training history without anomaly-label filtering; this history may contain anomalies. Configuration selection uses separate tuning collections. Evaluation-track labels are used only to compute metrics.

Table 9: Dataset statistics. Lengths count timestamps and are averaged over series for TSB-AD and machines for SMD. HAI and SWaT each contain one evaluated system.

Dataset	Task	Series	Channels	Training length	Test length
TSB-AD-U	Detection	350	1	6,976	44,910
TSB-AD-M	Detection	180	2–248	14,967	93,859
SMD	Localization	28	38	25,300	25,301
HAI	Localization	1	79	216,001	402,005
SWaT	Localization	1	51	495,000	449,919

B.2 Evaluation Metrics and Intervals

Detection. FARO and baseline reruns use the same six-metric TSB-AD evaluator (Liu & Paparrizos, 2024), with VUS-PR as the primary measure. Metrics are computed per series and averaged across each complete track. VUS-PR and VUS-ROC (Paparrizos et al., 2022) use 250 thresholds and a series-specific period. AUC-PR uses average precision, and AUC-ROC uses the standard ROC area. Point-F1 is the maximum point-wise F1 over thresholds; Range-F1 maximizes the benchmark's range-based F1 over 100 uniformly spaced score thresholds. All six metrics use unadjusted scores and labels, without point adjustment.

Evaluation intervals. Following the released TSB-AD and PaAno evaluation implementations (Liu & Paparrizos, 2024; Park & Kang, 2026), FARO constructs reference banks from the designated training prefix and evaluates the complete training–test series. TSPulse-FT also evaluates the full series after prefix fine-tuning. Time-RCD uses a zero-shot checkpoint with scores and labels restricted to the post-prefix suffix. Table 10 reports FARO and Time-RCD on identical post-prefix suffixes, using FARO's saved scores and the same evaluator. The comparisons in Section 4.3 use the full timeline for all conditions. Encoder-learning controls and reference-scoring ablations match evaluation segments within each comparison.

Table 10: Mean VUS-PR on identical post-prefix suffixes.

Track	FARO	Time-RCD
TSB-AD-U	**0.5834**	0.5169
TSB-AD-M	**0.4875**	0.3182

Channel localization. Localization uses annotated test timestamps and intervals. Let G_t be the annotated affected-channel set at timestamp t, and let π_t rank the C channels by decreasing anomaly score. For $P \in \{100, 150\}$, we retrieve $k_t = \min\{C, \lceil P|G_t|/100 \rceil\}$ channels and compute

$$\mathrm{HR}_t@P\% = \frac{|G_t \cap \{\pi_t(j)\}_{j=1}^{k_t}|}{|G_t|}, \qquad \mathrm{NDCG}_t@P\% = \frac{\sum_{j=1}^{k_t} \mathbf{1}[\pi_t(j) \in G_t] / \log_2(j+1)}{\sum_{j=1}^{|G_t|} 1/\log_2(j+1)}. \quad (9)$$

Both metrics are averaged over annotated anomalous timestamps. IPS ranks channels by their maximum score within each anomaly interval and computes HR against the union of affected channels in that interval, using the same retrieval rule. It then averages equally over intervals (Shimillas et al., 2026). SMD pools timestamps for HR/NDCG and intervals for IPS across all 28 machines within each run; HAI and SWaT are evaluated separately.

B.3 Detection Baselines and Tuning

Baseline methods. We compare FARO with PCA and Sub-PCA (Aggarwal, 2017), KMeansAD (Liu & Paparrizos, 2024), KShapeAD (Paparrizos & Gravano, 2015; Liu & Paparrizos, 2024), DeepAnT (Munir et al., 2019), USAD (Audibert et al., 2020), OmniAnomaly (Su et al., 2019), DCdetector (Yang et al., 2023), DLinear and NLinear (Zeng et al., 2023), PatchTST (Nie et al., 2023), iTransformer (Liu et al., 2024), ModernTCN (Luo & Wang, 2024), CATCH (Wu et al., 2025), KAN-AD (Zhou et al., 2025), ScatterAD (Yin et al., 2025), PaAno (Park & Kang, 2026), DADA (Shentu et al., 2025), MOMENT-FT (Goswami et al., 2024), TSPulse-FT (Ekambaram et al., 2026), Time-RCD (Lan et al., 2026), and RandomAD (Xi & Lin, 2026).

Result sources. Benchmark rows use the six-metric results reported by Park & Kang (2026), with VUS-PR checked against TSB-AD aggregates (Liu & Paparrizos, 2024). PaAno uses its published ten-seed means; our three-seed reproduction on the complete tracks agrees on all six metrics to two decimal places. The four multivariate reruns in Section 4.3 share their original-detector results with the main detection table; their settings are specified in Appendix G.

Table 11: Result sources for the main and full detection comparisons.

Result source	Methods
Benchmark results	U: Sub-PCA, KShapeAD, DeepAnT, USAD, MOMENT-FT; M: KMeansAD, PCA, DeepAnT, OmniAnomaly
Published results	PaAno (U/M; ten-seed aggregate)
Released-code reruns	DLinear, NLinear, PatchTST, iTransformer, DCdetector, DADA, TSPulse-FT, Time-RCD (U/M); KAN-AD (U/M); RandomAD (U); CATCH, ModernTCN, ScatterAD (M)

Hyperparameter selection. For tuned reruns, we select one configuration per track by mean VUS-PR on each complete tuning collection, then fix it for the corresponding evaluation track. Training-prefix anomaly labels are excluded from training and selection. Table 12 reports the candidates and selected values. TSPulse-FT, Time-RCD, and RandomAD retain their released configurations. Benchmark search spaces follow Liu & Paparrizos (2024); Park & Kang (2026). The four multivariate reruns in Section 4.3 instead use the fixed training configurations in Appendix G.2.

Table 12: Search spaces and selected configurations for baseline reruns. L is window length and lr is learning rate.

Method	Candidates	Selected
DLinear	$L \in \{60, 80, 100\}$	U: 100; M: 60
NLinear	$L \in \{60, 80, 100\}$	U: 60; M: 80
PatchTST (U)	Epochs $\in \{5, 10, 15\}$	10 epochs
iTransformer	$L \in \{64, 96\}$; lr $\in \{10^{-4}, 5\times10^{-5}\}$	U: $L = 64$; M: $L = 96$; lr 10^{-4}
DCdetector	$L \in \{80, 100\}$; lr $\in \{10^{-4}, 10^{-5}\}$	$L = 100$; lr 10^{-4}
KAN-AD (U)	$L \in \{32, 64, 96\}$; lr $\in \{10^{-2}, 10^{-3}, 10^{-4}\}$	$L = 64$; lr 10^{-2}
DADA	Batch size $\in \{32, 64, 96\}$	U: 96; M: 32

Reconstruction adapter. DLinear and NLinear (Zeng et al., 2023), PatchTST on U (Nie et al., 2023), and iTransformer (Liu et al., 2024) use their released architectures with a common reconstruction adapter. Scaling statistics are fitted on the unlabeled training prefix, which is split chronologically into 80% optimization and 20% validation data. Timestamp scores average squared reconstruction errors over channels and overlapping complete windows, including the final window.

Native anomaly scores. DCdetector uses attention discrepancy. KAN-AD uses absolute prediction error on U; on M, it averages channel-mean squared reconstruction error across overlapping windows. CATCH uses temporal/frequency reconstruction error; DADA (Shentu et al., 2025) uses its masked-reconstruction checkpoint score; TSPulse-FT uses its fine-tuned anomaly score; and

Time-RCD uses zero-shot anomaly-head probabilities. RandomAD retains MiniROCKET (Dempster et al., 2021)/KSS, Euclidean 3-NN, autocorrelation-based window selection, and peak-contrast scoring.

B.4 Channel-Localization Baselines

We run the released implementations of TranAD (Tuli et al., 2022), ATF-UAD (Fan et al., 2023), USAD (Audibert et al., 2020), GDN (Deng & Hooi, 2021), SARAD (Dai et al., 2024), AERCA (Han et al., 2025), DAEMON (Chen et al., 2021), and ALoRa-Loc (Shimillas et al., 2026). Native and cross-dataset runs preserve each implementation's channel-wise score and use the common HR/NDCG/IPS evaluator. Baseline results average three runs, except ALoRa-Loc on SMD, which averages five.

B.5 Statistical Reporting

Trained baseline reruns average three runs with complete track coverage; DADA averages three stochastic masking runs of its released checkpoint. Deterministic checkpoint inference runs once. FARO metrics are averaged across series within each seed, then summarized across ten seeds (Appendix C.2).

C BENCHMARK RESULTS

C.1 FULL DETECTION COMPARISON

Table 13: Full detection comparison on TSB-AD-U/M. FARO is shown in **bold** and reports means over ten independently evaluated fixed encoders. Sources and scoring intervals are specified in Appendix B; DLinear, NLinear, iTransformer, and univariate PatchTST use the reconstruction adapter specified there. FARO's per-seed results are in Appendix C.2.

	Method	VUS-PR	VUS-ROC	Range-F1	AUC-PR	AUC-ROC	Point-F1
TSB-AD-U: 350 univariate series							
Classical	Sub-PCA	0.42	0.76	0.41	0.37	0.71	0.42
	KShapeAD	0.40	0.76	0.40	0.35	0.74	0.39
Learned	DeepAnT	0.34	0.79	0.35	0.33	0.71	0.38
	USAD	0.36	0.71	0.40	0.32	0.66	0.37
	DCdetector	0.09	0.59	0.14	0.05	0.47	0.10
	DLinear	0.39	0.84	0.36	0.38	0.78	0.43
	NLinear	0.35	0.77	0.35	0.33	0.71	0.37
	PatchTST	0.43	0.84	0.40	0.43	0.81	0.47
	iTransformer	0.42	0.82	0.39	0.41	0.78	0.45
	KAN-AD	0.33	0.77	0.35	0.30	0.70	0.35
	PaAno	0.53	0.89	0.49	0.47	0.87	0.52
Pretrained	DADA	0.31	0.77	0.32	0.30	0.71	0.34
	MOMENT-FT	0.39	0.76	0.35	0.30	0.69	0.35
	TSPulse-FT	0.55	0.87	0.52	0.48	0.85	0.55
	Time-RCD	0.52	0.87	0.47	0.50	0.84	0.54
Fixed	RandomAD	0.43	0.86	0.35	0.33	0.81	0.40
	FARO (ours)	**0.57**	**0.90**	**0.54**	**0.51**	**0.88**	**0.56**
TSB-AD-M: 180 multivariate series							
Classical	KMeansAD	0.29	0.73	0.33	0.25	0.69	0.31
	PCA	0.31	0.74	0.29	0.31	0.70	0.37
Learned	DeepAnT	0.31	0.76	0.37	0.32	0.73	0.37
	OmniAnomaly	0.31	0.69	0.37	0.27	0.65	0.32
	DCdetector	0.05	0.47	0.14	0.05	0.48	0.10
	DLinear	0.35	0.74	0.26	0.35	0.72	0.40
	NLinear	0.33	0.70	0.26	0.31	0.67	0.38
	PatchTST	0.39	0.72	0.38	0.33	0.67	0.41
	iTransformer	0.35	0.72	0.26	0.34	0.70	0.40
	ModernTCN	0.35	0.73	0.26	0.34	0.69	0.38
	CATCH	0.33	0.74	0.26	0.32	0.71	0.37
	KAN-AD	0.33	0.76	0.29	0.34	0.74	0.39
	ScatterAD	0.24	0.63	0.22	0.21	0.58	0.27
	PaAno	0.43	0.79	0.41	0.38	0.76	0.43
Pretrained	DADA	0.30	0.71	0.24	0.30	0.68	0.34
	TSPulse-FT	0.39	0.72	0.46	0.41	0.72	0.46
	Time-RCD	0.32	0.71	0.27	0.34	0.72	0.39
Fixed	**FARO (ours)**	**0.48**	**0.81**	**0.47**	**0.44**	**0.79**	**0.49**

C.2 Detection across Encoder Seeds

Tables 14 and 17 report individual FARO results over ten seeds, with their mean and standard deviation. Each run uses a single fixed encoder shared across series and channels for inference.

Table 14: FARO detection across ten encoder seeds on the complete evaluation tracks. Metrics use the full timeline. Values are rounded to three decimals.

TSB-AD-M

Seed	VUS-PR	VUS-ROC	Range-F1	AUC-PR	AUC-ROC	Point-F1
2027	0.478	0.806	0.460	0.445	0.783	0.480
2028	0.491	0.816	0.471	0.447	0.790	0.497
2029	0.478	0.809	0.464	0.430	0.780	0.482
2030	0.493	0.820	0.466	0.449	0.793	0.497
2031	0.485	0.812	0.476	0.449	0.788	0.496
2032	0.456	0.803	0.455	0.419	0.779	0.466
2033	0.498	0.818	0.495	0.463	0.792	0.505
2034	0.496	0.812	0.467	0.451	0.783	0.496
2035	0.470	0.809	0.469	0.430	0.784	0.488
2036	0.494	0.813	0.473	0.450	0.786	0.500
Mean	**0.484**	**0.812**	**0.470**	**0.443**	**0.786**	**0.491**
Std. dev.	0.013	0.005	0.011	0.013	0.005	0.012

TSB-AD-U

Seed	VUS-PR	VUS-ROC	Range-F1	AUC-PR	AUC-ROC	Point-F1
2027	0.558	0.909	0.519	0.484	0.885	0.536
2028	0.579	0.900	0.549	0.522	0.875	0.565
2029	0.577	0.901	0.531	0.516	0.873	0.560
2030	0.556	0.895	0.526	0.495	0.867	0.545
2031	0.591	0.900	0.562	0.535	0.876	0.574
2032	0.531	0.905	0.492	0.467	0.878	0.515
2033	0.596	0.903	0.561	0.539	0.879	0.572
2034	0.585	0.906	0.540	0.523	0.879	0.565
2035	0.580	0.901	0.555	0.521	0.874	0.563
2036	0.575	0.899	0.542	0.517	0.871	0.557
Mean	**0.573**	**0.902**	**0.538**	**0.512**	**0.876**	**0.555**
Std. dev.	0.019	0.004	0.022	0.023	0.005	0.018

C.3 SUBSET-WISE DETECTION RESULTS

Table 15: Subset-wise FARO detection performance on TSB-AD-U. For each subset, rows report the mean and standard deviation (SD) across ten independently evaluated fixed encoders after averaging series within each subset. Avg. dim. is the mean number of channels; the average rows reproduce Table 1.

Subset	# series	Avg. dim.	Statistic	VUS-PR	VUS-ROC	Range-F1	AUC-PR	AUC-ROC	Point-F1
UCR	70	1	Mean	0.419	0.912	0.466	0.399	0.900	0.442
			SD	0.048	0.013	0.037	0.044	0.014	0.037
YAHOO	30	1	Mean	0.655	0.913	0.475	0.556	0.902	0.594
			SD	0.053	0.008	0.071	0.072	0.010	0.063
WSD	20	1	Mean	0.418	0.943	0.531	0.372	0.939	0.424
			SD	0.020	0.014	0.023	0.024	0.012	0.017
CATSv2	1	1	Mean	0.353	0.762	0.177	0.520	0.754	0.587
			SD	0.012	0.007	0.034	0.006	0.007	0.007
Daphnet	1	1	Mean	0.441	0.936	0.420	0.428	0.936	0.509
			SD	0.048	0.013	0.152	0.047	0.011	0.039
Exathlon	30	1	Mean	0.900	0.984	0.752	0.894	0.983	0.903
			SD	0.019	0.005	0.040	0.020	0.005	0.012
IOPS	15	1	Mean	0.353	0.903	0.390	0.268	0.871	0.369
			SD	0.024	0.007	0.019	0.036	0.018	0.034
LTDB	8	1	Mean	0.596	0.779	0.588	0.510	0.745	0.597
			SD	0.055	0.038	0.049	0.061	0.039	0.033
MGAB	8	1	Mean	0.133	0.876	0.370	0.199	0.868	0.323
			SD	0.036	0.039	0.043	0.024	0.041	0.027
MITDB	7	1	Mean	0.376	0.871	0.343	0.343	0.824	0.392
			SD	0.033	0.019	0.034	0.033	0.023	0.034
MSL	7	1	Mean	0.589	0.898	0.602	0.539	0.869	0.591
			SD	0.109	0.047	0.056	0.096	0.052	0.083
NAB	23	1	Mean	0.586	0.843	0.631	0.565	0.823	0.611
			SD	0.042	0.016	0.033	0.045	0.019	0.038
NEK	8	1	Mean	0.642	0.859	0.651	0.633	0.872	0.674
			SD	0.041	0.016	0.043	0.042	0.016	0.047
OPPORTUNITY	27	1	Mean	0.330	0.743	0.413	0.323	0.731	0.394
			SD	0.047	0.044	0.044	0.050	0.045	0.043
Power	1	1	Mean	0.144	0.647	0.228	0.135	0.626	0.249
			SD	0.015	0.014	0.028	0.025	0.015	0.017
SED	2	1	Mean	0.877	0.992	0.707	0.673	0.984	0.697
			SD	0.054	0.005	0.038	0.057	0.006	0.036
SMAP	17	1	Mean	0.911	0.980	0.872	0.895	0.977	0.873
			SD	0.029	0.006	0.023	0.031	0.006	0.023
SMD	33	1	Mean	0.748	0.959	0.734	0.733	0.958	0.733
			SD	0.060	0.006	0.036	0.058	0.007	0.042
Stock	8	1	Mean	0.692	0.834	0.116	0.077	0.469	0.151
			SD	0.015	0.004	0.058	0.001	0.006	0.000
SVDB	18	1	Mean	0.628	0.968	0.573	0.560	0.949	0.612
			SD	0.058	0.008	0.041	0.062	0.012	0.046
SWaT	1	1	Mean	0.510	0.778	0.244	0.736	0.838	0.791
			SD	0.022	0.013	0.003	0.008	0.006	0.000
TAO	2	1	Mean	0.879	0.935	0.270	0.110	0.484	0.209
			SD	0.008	0.002	0.155	0.005	0.014	0.000
TODS	13	1	Mean	0.677	0.864	0.290	0.320	0.742	0.380
			SD	0.021	0.007	0.028	0.032	0.012	0.019
TSB-AD-U average			Mean	**0.573**	**0.902**	**0.538**	**0.512**	**0.876**	**0.555**
			SD	**0.019**	**0.004**	**0.022**	**0.023**	**0.005**	**0.018**

Table 16: Subset-wise FARO detection performance on TSB-AD-M. For each subset, rows report the mean and standard deviation (SD) across ten independently evaluated fixed encoders after averaging series within each subset. Avg. dim. is the mean number of channels; the average rows reproduce Table 1.

Subset	# series	Avg. dim.	Statistic	VUS-PR	VUS-ROC	Range-F1	AUC-PR	AUC-ROC	Point-F1
CATSv2	5	17	Mean	0.364	0.809	0.467	0.395	0.800	0.467
			SD	0.125	0.062	0.057	0.104	0.053	0.107
CreditCard	1	29	Mean	0.066	0.687	0.046	0.015	0.715	0.072
			SD	0.008	0.011	0.011	0.005	0.013	0.022
Daphnet	1	9	Mean	0.385	0.935	0.402	0.370	0.934	0.487
			SD	0.051	0.013	0.138	0.058	0.013	0.039
Exathlon	25	20.16	Mean	0.872	0.946	0.796	0.890	0.962	0.903
			SD	0.018	0.002	0.022	0.019	0.002	0.024
GECCO	1	9	Mean	0.255	0.944	0.323	0.349	0.956	0.410
			SD	0.038	0.011	0.065	0.070	0.010	0.052
Genesis	1	18	Mean	0.674	0.995	0.551	0.480	0.994	0.539
			SD	0.277	0.009	0.142	0.282	0.009	0.201
GHL	23	19	Mean	0.011	0.419	0.017	0.010	0.401	0.033
			SD	0.001	0.023	0.003	0.001	0.025	0.003
LTDB	4	2.25	Mean	0.412	0.767	0.481	0.364	0.747	0.467
			SD	0.041	0.027	0.059	0.046	0.025	0.038
MITDB	11	2	Mean	0.359	0.864	0.485	0.441	0.840	0.492
			SD	0.032	0.010	0.045	0.045	0.012	0.036
MSL	14	55	Mean	0.483	0.870	0.557	0.448	0.862	0.521
			SD	0.049	0.011	0.037	0.052	0.012	0.037
OPPORTUNITY	7	248	Mean	0.110	0.407	0.283	0.097	0.396	0.186
			SD	0.005	0.031	0.019	0.005	0.032	0.011
PSM	1	25	Mean	0.202	0.631	0.308	0.184	0.625	0.249
			SD	0.020	0.016	0.027	0.024	0.015	0.010
SMAP	25	25	Mean	0.672	0.944	0.687	0.669	0.956	0.699
			SD	0.047	0.009	0.023	0.050	0.008	0.035
SMD	20	38	Mean	0.457	0.886	0.517	0.467	0.902	0.512
			SD	0.060	0.026	0.035	0.040	0.014	0.031
SVDB	28	2	Mean	0.498	0.880	0.500	0.456	0.860	0.512
			SD	0.053	0.020	0.032	0.060	0.022	0.041
SWaT	2	58.50	Mean	0.480	0.855	0.366	0.477	0.856	0.606
			SD	0.048	0.014	0.053	0.070	0.013	0.041
TAO	11	3	Mean	0.701	0.844	0.093	0.080	0.468	0.158
			SD	0.005	0.002	0.068	0.001	0.006	0.000
TSB-AD-M average			Mean	**0.484**	**0.812**	**0.470**	**0.443**	**0.786**	**0.491**
			SD	**0.013**	**0.005**	**0.011**	**0.013**	**0.005**	**0.012**

C.4 Channel Localization across Encoder Seeds

Table 17: FARO channel localization across ten encoder seeds. HR and NDCG average annotated anomalous timestamps; IPS averages annotated intervals. Values are rounded to three decimals.

SMD

Seed	HR@100%	HR@150%	NDCG@100%	NDCG@150%	IPS@100%	IPS@150%
2027	0.636	0.761	0.660	0.734	0.557	0.714
2028	0.676	0.799	0.704	0.777	0.612	0.751
2029	0.677	0.801	0.703	0.776	0.601	0.735
2030	0.658	0.781	0.691	0.764	0.600	0.732
2031	0.634	0.766	0.662	0.741	0.565	0.715
2032	0.628	0.765	0.647	0.729	0.534	0.702
2033	0.629	0.755	0.652	0.728	0.560	0.699
2034	0.664	0.789	0.695	0.770	0.615	0.753
2035	0.683	0.811	0.705	0.781	0.590	0.748
2036	0.670	0.788	0.699	0.770	0.604	0.738
Mean	**0.656**	**0.782**	**0.682**	**0.757**	**0.584**	**0.729**
Std. dev.	0.022	0.019	0.023	0.022	0.028	0.020

HAI

Seed	HR@100%	HR@150%	NDCG@100%	NDCG@150%	IPS@100%	IPS@150%
2027	0.292	0.392	0.319	0.381	0.337	0.441
2028	0.162	0.227	0.172	0.212	0.170	0.230
2029	0.198	0.258	0.208	0.244	0.211	0.285
2030	0.131	0.196	0.129	0.169	0.148	0.230
2031	0.150	0.235	0.148	0.201	0.204	0.252
2032	0.227	0.298	0.243	0.287	0.285	0.370
2033	0.184	0.255	0.184	0.227	0.193	0.267
2034	0.147	0.206	0.151	0.188	0.148	0.230
2035	0.190	0.294	0.196	0.261	0.189	0.296
2036	0.149	0.209	0.161	0.198	0.156	0.237
Mean	**0.183**	**0.257**	**0.191**	**0.237**	**0.204**	**0.284**
Std. dev.	0.048	0.059	0.056	0.062	0.062	0.070

SWaT

Seed	HR@100%	HR@150%	NDCG@100%	NDCG@150%	IPS@100%	IPS@150%
2027	0.155	0.166	0.167	0.174	0.535	0.591
2028	0.143	0.148	0.153	0.156	0.515	0.540
2029	0.144	0.154	0.154	0.160	0.455	0.551
2030	0.152	0.166	0.163	0.172	0.515	0.545
2031	0.144	0.150	0.155	0.159	0.495	0.561
2032	0.158	0.169	0.170	0.177	0.495	0.611
2033	0.139	0.147	0.148	0.153	0.495	0.551
2034	0.142	0.147	0.153	0.156	0.515	0.530
2035	0.156	0.167	0.167	0.174	0.515	0.566
2036	0.144	0.151	0.155	0.159	0.515	0.545
Mean	**0.148**	**0.157**	**0.159**	**0.164**	**0.505**	**0.559**
Std. dev.	0.007	0.010	0.008	0.009	0.022	0.025

D Encoder-Learning Controls

D.1 Matched Comparison and Training Settings

Table 5 compares fixed and trained encoders over three initializations, matching the training prefix, reference indices, test windows, scorer, and aggregation. All compiled encoder coefficients are trainable; no live BatchNorm layers are present. Reference and test features are recomputed after training, and VUS-PR is evaluated on the test segment.

Training uses unlabeled history without label filtering or a validation split. AdamW uses weight decay 10^{-4}, batch size up to 256, and cosine learning-rate decay to one tenth of the initial rate. Each objective selects a learning rate from $\{10^{-4}, 3\times10^{-4}, 10^{-3}\}$ and an update budget from $\{50, 100\}$ using mean VUS-PR on the separate tuning collection, with equal weight for M and U. Evaluation labels are excluded from selection. Each run retains the encoder with the smallest recorded training minibatch loss.

Table 18: Objective settings for Table 5. Auxiliary heads are trained jointly and used only for their training objectives.

Objective	Loss and auxiliary head	Initial LR	Updates
InfoNCE	Symmetric in-batch cross-entropy; temperature 0.2; no head	10^{-4}	50
Triplet	Cosine-distance hinge loss; margin 0.1; no head	10^{-4}	50
Masked reconstruction	MSE on a masked contiguous 24/96 span; linear $16\to96$ decoder	10^{-4}	100
Forecasting	Next-window MSE; linear $16 \to 96$ prediction head	10^{-4}	100
One-class	Squared distance to a fixed initial feature center; no head	3×10^{-4}	100

InfoNCE and triplet use same-channel temporal positives and in-batch negatives. Reconstruction and forecasting targets are standardized per window. Auxiliary heads and the one-class center are excluded from anomaly scoring.

D.2 Held-Out Training Trajectories

Figure 4 uses a predefined subset of 23 TSB-AD-U series and three initializations. Each history is split chronologically into 80% for training and 20% for held-out loss, without overlapping observations. Held-out inputs and evaluation conditions remain fixed. The reference bank uses the full training prefix, including the held-out segment, and retains the same indices throughout.

Each objective starts from the matched initial encoder and runs for 400 continuous updates using the learning rate in Table 18, decayed over the full run. We evaluate checkpoints at the steps shown in the figure, without early stopping or checkpoint replacement. Test performance is excluded from subset and checkpoint selection. Both reference and test features are recomputed at each checkpoint.

Loss ratios and VUS-PR changes are computed relative to initialization, then averaged over seeds within each series and across series. Reconstruction and forecasting losses include the jointly trained heads.

E Reference-Scoring Ablations and Computational Efficiency

E.1 Feature Representation and Reference Controls

Table 19: Encoder-construction and reference-source controls in FARO. M/U report VUS-PR and SMD reports HR@100%. Panel (a) reports changes from matched FARO baselines (Δ = control − FARO); panel (b) reports means.

(a) Encoder construction

Control	Δ M	Δ U	Δ SMD
Orthogonal-16	−0.1525	−0.1419	−0.5069
No encoder	−0.0533	−0.0960	−0.4961

(b) Reference source

Source	M	U	SMD
Own series	**0.4748**	**0.5848**	**0.663**
Foreign series	0.3472	0.3631	0.380

Feature representation. The single-view controls retain one eight-dimensional branch; FARO concatenates both branches. We also compare FARO with a fixed 16-dimensional random orthogonal projection (Orthogonal-16) and time–frequency features without an encoder, holding references

and scoring fixed within each comparison (Appendix A.2). Both controls yield lower VUS-PR on both tracks and lower HR@100% on SMD.

Reference source. Foreign banks use up to three other series from the same source family, selected without anomaly labels and matched in bank size. Multivariate comparisons require identical ordered channel schemas. Results average over foreign banks within each target and encoder seed. Eligible cohorts contain 165 M series, 345 U series, and 28 SMD machines. Other U ablations contain 349 finite series; one query segment has no positive labels.

Replacing own-series references with compatible foreign-series references reduces detection on both tracks and localization on SMD, with the encoder, scorer, and bank size held fixed.

Channel specificity. Pooled banks balance windows across channels, including the query channel. Other-channel banks exclude its history and average over three fixed channel derangements, with the encoder and scorer unchanged.

Reference representation. Centroid and diagonal-Gaussian scoring summarize the same training features as the empirical bank; both representation and scoring change.

E.2 Sensitivity Analysis

The bank sweep varies the reference cap with evenly spaced sampling, holding the encoder and query grid fixed. The L and k sweeps retain the selected cap. Figure 5 shows detection sensitivity.

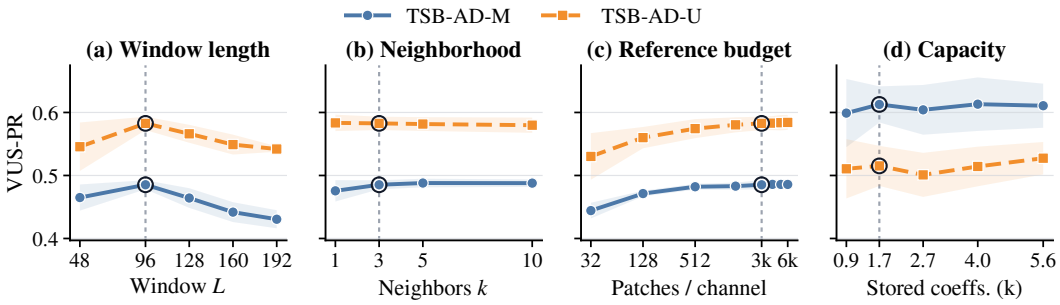

Figure 5: Sensitivity to window length, neighbors, reference budget, and encoder capacity. Panels (a–c): query-only VUS-PR over three encoders on M/U. Panel (d): full-timeline VUS-PR over ten encoders on the M20/U48 tuning collections. Bands show one standard deviation; outlined markers identify the selected settings.

E.3 Computational Efficiency

We profile three TSB-AD-M workloads selected without anomaly labels or profiling results to cover different sequence lengths and numbers of channels.

All measurements use a single NVIDIA RTX A6000 with three fresh-process runs per method–workload pair. Preparation includes target-specific processing before scoring, while inference measures the first complete scoring pass over the full series. Total runtime is their sum, and peak GPU memory is measured during inference. Each method uses the configuration corresponding to its reported detection result; shared pretraining costs are excluded.

Table 20: Workloads used for computational profiling.

Source	Channels	Train length	Test length
MSL id-2	55	883	2,652
Exathlon id-18	18	11,665	34,995
Exathlon id-9	31	10,766	32,300

F Qualitative Examples

F.1 Detection and Channel Localization Examples

All examples use fixed encoder seed 2027. Channel traces follow increasing channel index, matching the evidence-field axis. Shading marks labeled anomaly intervals; orange traces identify annotated affected channels.

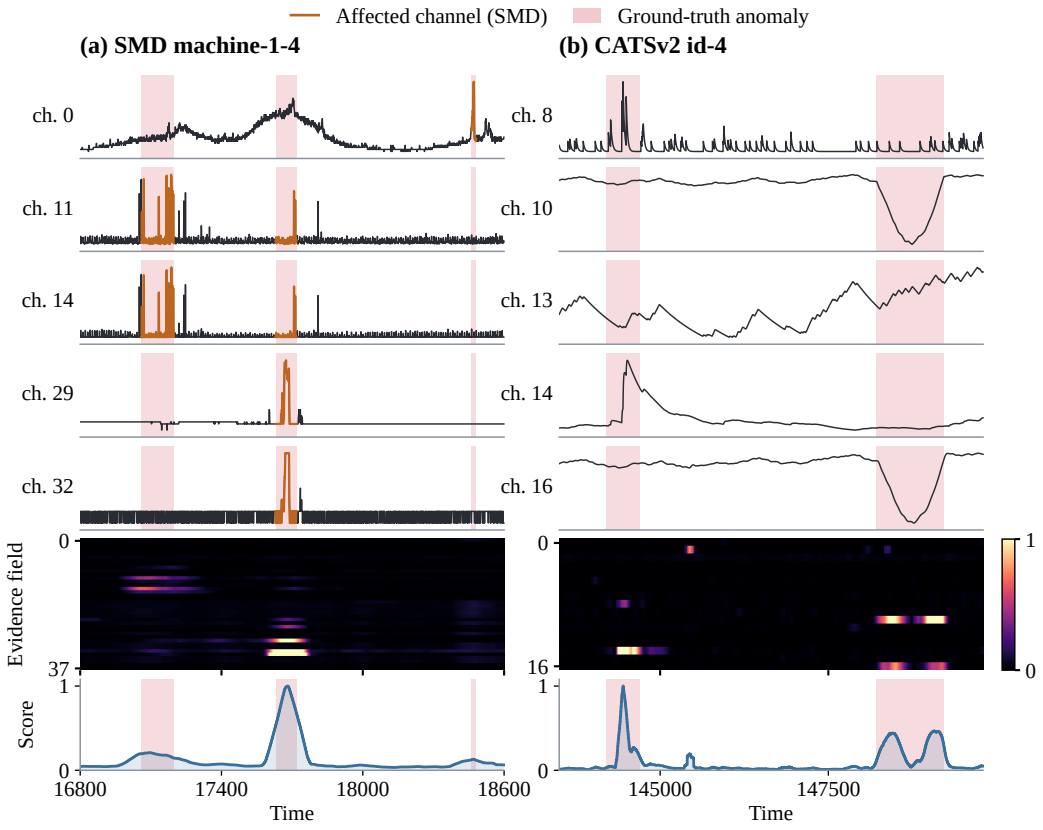

Figure 6: Channel-level evidence and anomaly scores. (a) SMD machine-1-4, with annotated affected channels highlighted. (b) CATSv2 id-4, with two labeled anomaly intervals. Each example shows channel observations, the evidence field, and the detection score. Channel traces use separate vertical scales without clipping; evidence fields and detection scores are normalized for display.

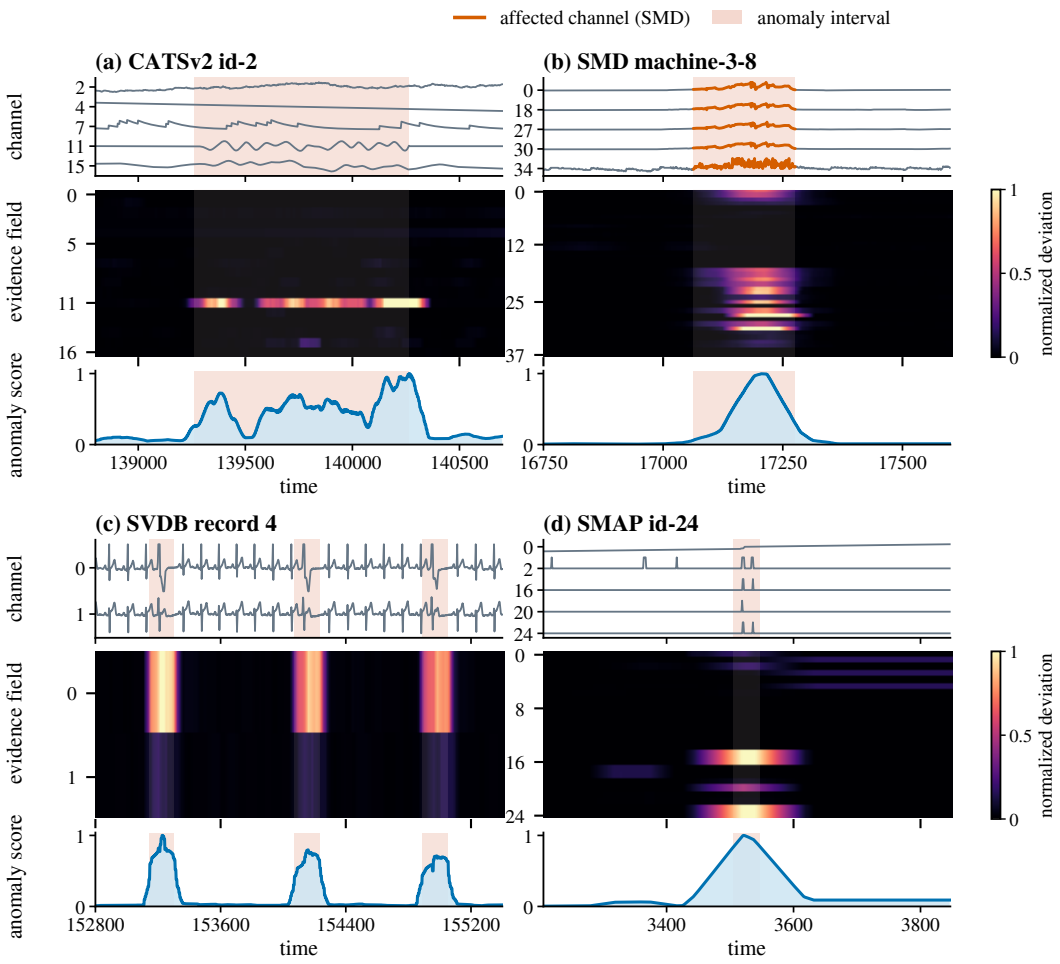

Figure 7: Additional qualitative examples. (a) CATSv2 id-2; (b) SMD machine-3-8; (c) SVDB record 4; (d) SMAP id-24.

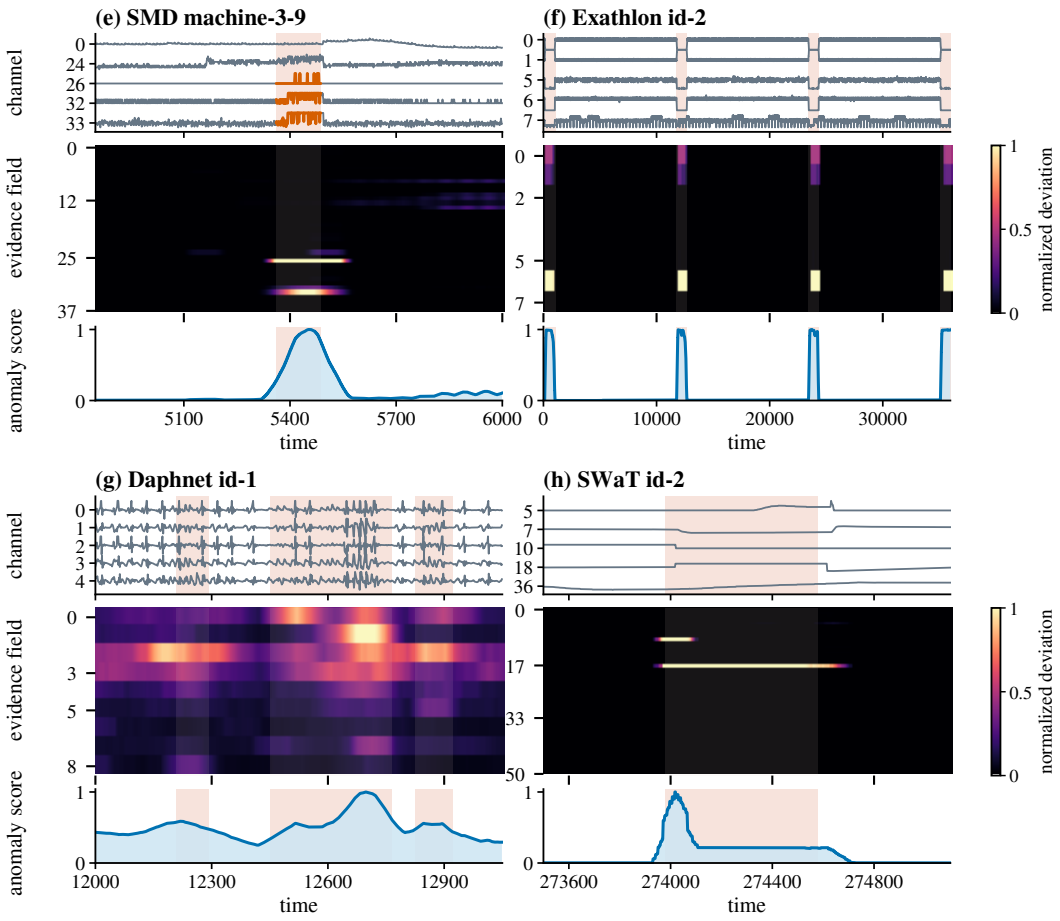

Figure 8: Additional qualitative examples (continued). (e) SMD machine-3-9; (f) Exathlon id-2; (g) Daphnet id-1; (h) SWaT id-2.

G Historical References for Existing Detectors

G.1 Comparison and Evaluation

We evaluate all 180 TSB-AD-M and 350 TSB-AD-U series with seeds 2027–2029. Each model uses the designated unlabeled training prefix, without anomaly-label filtering. The trained original and reference variants share one checkpoint; the untrained variant uses that run's random initialization. The two reference variants use identical raw reference items and recompute both reference and query features with their respective models. Reference selection uses the training prefix, including its validation portion. VUS-PR evaluation follows the procedure in Appendix B.2. We average the three seed metrics within each series, then average equally across series. The multivariate original-detector results are the same runs reported in Table 1.

G.2 Training and Reference Definitions

We use released implementations of KAN-AD (Zhou et al., 2025), ModernTCN (Luo & Wang, 2024), ScatterAD (Yin et al., 2025), and CATCH (Wu et al., 2025), with the training objectives, budgets, and checkpoint rules in Table 21. Reference definitions were developed on a separate eight-series multivariate collection disjoint from the 180 evaluation series, then fixed for evaluation. The tables below specify the multivariate implementations. ModernTCN and ScatterAD retain these settings for univariate evaluation. KAN-AD uses its released univariate forecasting implementation: window length 96, order 2, batch size 1,024, Adam with learning rate 0.01, up to 100 epochs, validation patience 3, and StepLR decay by 0.75 every five epochs. It uses the final model, with

chronological 80/20 splitting, first differences, and training-fitted standardization. Reference entries use the 96-dimensional feature before the final convolution; scores are aligned to the forecast target.

Table 21: Training objectives, budgets, and checkpoint selection.

Model	Objective	Budget / checkpoint
KAN-AD	Reconstruction MSE	Up to 100 epochs, patience 5; best validation MSE
ModernTCN	Reconstruction MSE	10 epochs, patience 10; best validation MSE
ScatterAD	Scattering, temporal, and contrastive losses	At least 2 epochs and 300 updates; final epoch
CATCH	Temporal, channel, and frequency losses	3 epochs; best validation temporal MSE

KAN-AD, ModernTCN, and CATCH use chronological training–validation splits (80/20 by default).

Reference-based detection. Each bank contains at most 3,000 entries, with entry units defined in Table 22. Scores average three nearest-reference distances, with feature coordinates scaled by their reference-bank standard deviations. We exclude the identical reference ID, while other overlapping observations remain eligible. The original detector follows its native detection procedure. For reference-based detection, the model-specific features, normalization, scoring, and timestamp aggregation are specified below, with the same rules applied to the trained and untrained versions of each model.

Table 22: Model-specific reference entries and scoring. All distances are Euclidean after bank-based standardization, with the block weights stated below.

Model	Reference entry	Distance and timestamp aggregation
KAN-AD	Per-channel feature before the final linear layer	Channel mean, then overlap average
ModernTCN	Per-channel patch feature plus RevIN mean and standard deviation (Kim et al., 2022)	Mean squared distance within each block, then equal-weight block averaging and square root; patch-aligned scores
ScatterAD	Joint online-encoder feature per timestamp	Timestamp-aligned scores
CATCH	Temporal and frequency residuals over all channels at each timestamp	Frequency block weight 0.05 in squared distance; original timestamp grid