SCAR: CONDITION-AWARE RETRIEVAL OF LOCAL NORMAL REFERENCES FOR MULTIVARIATE TIME SERIES ANOMALY DETECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Industrial systems frequently switch among operating regimes, and the same local pattern may be normal under one regime yet anomalous under another. The normal reference used for anomaly scoring should therefore vary with operating conditions. Most existing methods, however, still treat the normal reference as a globally uniform object rather than one determined dynamically by query-specific conditions. We propose SCAR (Slow-fast Condition-Aware Retrieval), which dynamically constructs a condition-aware local normal reference for each query and uses it as the primary basis for anomaly scoring. SCAR decomposes each time window into a slowly varying state component and a rapidly varying residual component. It first retrieves regime-consistent candidate windows through state-level retrieval and then selects locally similar normal patches through patch-level context retrieval. SCAR then compares the query against the retrieved reference to obtain a memory distance, combines it with complementary evidence including state novelty and reconstruction diagnostics, and fuses all signals into a final anomaly score via empirical CDF calibration. On five public multivariate time-series anomaly detection benchmarks, SCAR achieves the best AUROC on every dataset as well as the highest average AUROC and AP among 24 baselines. Ablation studies show that this advantage stems from the complete design chain built around condition-aware local normal references rather than from any single component in isolation. Mechanism validation on the TEP dataset, which provides ground-truth regime annotations, further confirms that the learned state space effectively recovers true operating modes and that retrieved normal references predominantly share the same regime as the query.

1 INTRODUCTION

Industrial multivariate time-series anomaly detection aims to continuously monitor complex systems and promptly identify abnormal behavior. A central challenge is that industrial systems typically operate under multiple regimes, such as different load levels, process stages, or environmental conditions. The observed signals accordingly contain both slowly varying components reflecting regime evolution and rapidly varying components capturing local dynamics. Since normal behavior varies across regimes, the same local waveform may be normal in one regime but anomalous in another. Anomaly scoring should therefore not rely on a regime-agnostic global normal reference alone, but should account for both the operating regime and the local temporal context of each observation.

Existing methods have made progress in reconstruction, association modeling, contrastive learning, memory mechanisms, and nearest-neighbor search, yet many still compare each query against a global model, a unified representation space, or a shared candidate set at the scoring stage. When operating conditions are relatively homogeneous, this uniform-reference approximation may be acceptable. However, when a system transitions across multiple regimes, normal patterns from different conditions become mixed within the same reference space, making it harder to separate regime variation from genuine anomalies. As shown in Figure 1, a concrete example arises in the TEP dataset: an anomalous query from Mode 1 can exhibit a local shape highly similar to that of a normal segment from Mode 3. If regime differences are ignored and references are selected solely by

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.

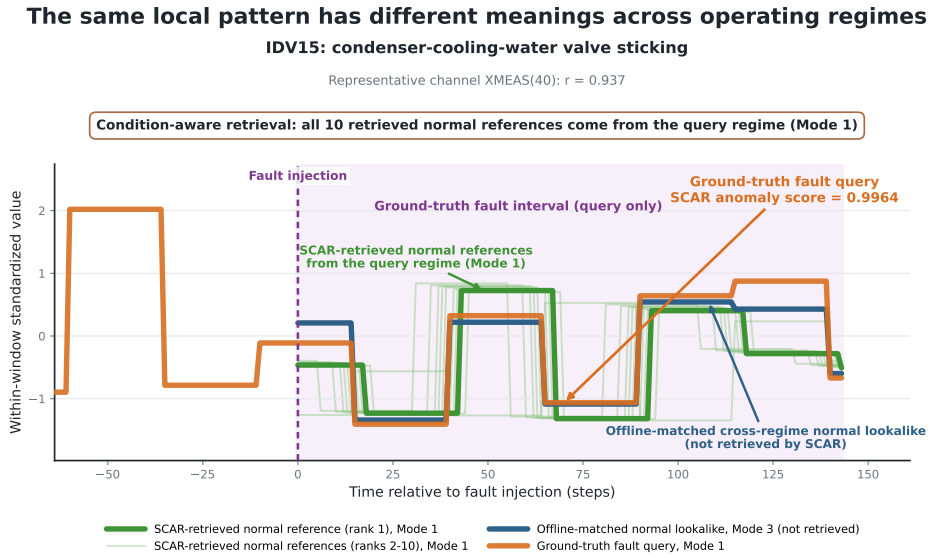


Figure 1: Condition-aware normal references for TEP fault IDV15. The Mode 1 fault query (orange) resembles an offline-matched Mode 3 normal segment (blue; $r = 0.937$), potentially misleading global retrieval. SCAR retrieves ten Mode 1 references (green) and assigns the query an anomaly score of 0.9964.

morphological similarity from all normal samples, such a cross-regime normal segment can make the query appear consistent with normal behavior, suppressing its anomaly score and potentially causing it to be misclassified as normal. SCAR instead retrieves normal segments that are compatible with the query's operating regime and relevant to its local temporal context. The query's deviation from normal behavior under the current regime then becomes visible, providing a more reliable basis for anomaly detection. The normal reference underlying anomaly scoring should therefore no longer be treated as a globally uniform object, but as a condition-aware object jointly determined by the operating regime and local temporal context of each query.

Realizing this shift requires constructing, for each query, a normal reference set matched to its regime and local temporal context. Because slowly and rapidly varying components coexist in industrial time series, this goal gives rise to two design requirements. First, state-level filtering: the reference set should be restricted to samples from the same or a similar regime as the query, reducing systematic bias from cross-regime comparisons. Second, context-level refinement: within the state-compatible candidate pool, the reference should be further narrowed to samples sharing a similar local temporal context with the query, excluding irrelevant matches caused by different transient phases, local trends, or short-term disturbances. Nevertheless, how to construct such a hierarchical, condition-aware reference around regime and local temporal context, and integrate it systematically into the anomaly scoring process, remains largely unexplored in multivariate time-series anomaly detection.

To this end, we propose SCAR (Slow-fast Condition-Aware Retrieval), a framework that shifts anomaly scoring from comparison against a globally uniform reference to comparison against a condition-aware local reference. SCAR first applies slow-fast temporal decomposition to separate each input window into a slowly varying state component and a rapidly varying residual component, which characterize the current operating regime and support fine-grained local comparison, respectively. SCAR then builds a two-level condition-aware retrieval mechanism: the first level uses a state descriptor to retrieve candidate windows from similar regimes, and the second level retrieves patches sharing a similar local temporal context with the query within the state-filtered candidate pool, dynamically constructing a condition-aware local normal reference subset for each query. Finally, anomaly evidence is extracted from multiple complementary diagnostic views and fused into a final score through a calibration mechanism that maps heterogeneous signals onto a common scale.

The main contributions of this paper are as follows:

- We argue that the normal reference used for anomaly scoring in industrial multivariate time-series anomaly detection should shift from a globally uniform object to a condition-aware object jointly determined by the operating regime and local temporal context of each query, reducing the risk that cross-regime normal patterns mislead anomaly decisions.
- We propose SCAR, which combines slow-fast temporal decomposition with two-level condition-aware retrieval to dynamically construct a regime-matched and locally context-relevant normal reference for each query. SCAR uses the query's deviation from this reference as its primary anomaly evidence, complements it with state novelty and reconstruction diagnostics, and obtains the final anomaly score through calibrated fusion on a common scale.
- SCAR achieves state-of-the-art AUROC on five standard multivariate time-series anomaly detection benchmarks. Ablation studies verify the necessity of the complete design chain built around condition-aware local normal references. The complete six-mode TEP evaluation further confirms at the mechanism level that SCAR retrieves condition-compatible normal references and preserves calibrated separation across operating modes.

2 Related work

This section reviews representative multivariate time-series anomaly detection methods from the perspective of the normal reference on which anomaly scoring relies.

Reconstruction-based methods represent a typical paradigm. Early autoencoders and their robust variants use reconstruction error as the primary anomaly signal (Sakurada & Yairi, 2014; Zhou & Paffenroth, 2017), while subsequent methods such as LSTM encoder–decoders, OmniAnomaly, TranAD, and DADA improve reconstruction through temporal modeling, probabilistic modeling, or Transformer architectures (Malhotra et al., 2015; 2016; Su et al., 2019; Tuli et al., 2022; Shentu et al., 2025). Although these methods differ in model structure and reconstruction mechanism, their scoring process typically relies on the normal pattern characterized by a global decoder and its reconstruction target. Reconstruction error thus measures deviation from a global reference that does not adjust with the operating regime or local temporal context of the query.

Another line of work characterizes normality within a unified representation, association, or relational space. Anomaly Transformer detects anomalies through the discrepancy between prior and series associations (Xu et al., 2022), DCdetector strengthens the decision boundary via dual-attention branches and contrastive learning (Yang et al., 2023), and graph-based methods such as GDN, InterFusion, and GSC-AD further model inter-variable dependencies (Deng & Hooi, 2021; Li et al., 2021; Zhang et al., 2024). Unlike reconstruction-based methods, these approaches draw anomaly evidence from temporal associations, contrastive representations, or variable relational structures. However, their scoring process typically still rests on a unified representation space, relational structure, or comparison criterion, without explicitly distinguishing the normal reference range that should apply under different operating conditions.

Decomposition and multi-scale methods enhance complex temporal representation through frequency-domain decomposition, cross-scale modeling, or multi-granularity alignment. TFAD (Zhang et al., 2022), D^{3}R (Wang et al., 2023), CATCH (Wu et al., 2025), DualTF (Nam et al., 2024), and CrossAD (Li et al., 2025) demonstrate that different components and scales play non-equivalent roles in industrial time series: slowly varying trends, frequency structures, and local dynamics may all affect anomaly judgment. In most existing designs, however, these structural cues serve representation enhancement, reconstruction improvement, or detection-performance gains, and have not been employed to construct a condition-aware normal reference organized around regime and local temporal context. Nevertheless, these methods provide an important insight for our work: regime-related information and local dynamics should be modeled separately.

The methods most closely related to ours are memory-, prototype-, and retrieval-based approaches. Unlike single global models, these methods attempt to support anomaly judgment with multiple memory items, prototypes, or instance-level references. MEMTO, DMemAD, PUAD, H-PAD, and PGRF-Net characterize normality with memory items or prototypes (Song et al., 2023; Zhou et al., 2025; Li et al., 2023; Shen, 2025; Jeong & Yoon, 2026), while kNN, LOF, DAMP, GDFlex, and PaAno derive anomaly evidence from instance-level or subsequence-level distances (Ramaswamy et al., 2000; Breunig et al., 2000; Lu et al., 2022; Imamura, 2025; Park & Kang, 2026). These

methods move beyond abstract global summaries toward scoring based on multiple normal patterns, neighboring instances, or reference segments.

In most cases, however, memory items, prototypes, or instance-level references are still organized within a candidate space shared across all queries. The query therefore faces a globally shared reference set rather than one dynamically determined by the current regime and local temporal context. Some methods dynamically allocate attention weights or perform similarity matching at inference time, but this adaptivity operates within a fixed candidate pool rather than first determining which samples should enter the comparison set based on query conditions. In other words, although these methods introduce multiple memory items, prototypes, or instance-level references, they have not truly moved beyond the scope of a globally shared normal reference, but have merely replaced a single global model with a globally shared multi-pattern reference set.

Work in adjacent domains also illustrates the potential of explicit retrieval and hierarchical candidate filtering. PatchCore detects visual anomalies via a memory bank and nearest-neighbor patch retrieval (Roth et al., 2022), RAID introduces hierarchical retrieval for visual anomaly detection (Cai et al., 2026), and HF-Net adopts coarse-to-fine retrieval in visual localization (Sarlin et al., 2019). Although not designed for multivariate time-series anomaly detection, these studies show that explicit case retrieval can provide more direct evidence than a global summary, and that hierarchical candidate filtering can suppress irrelevant references before fine-grained comparison. For industrial time series under multiple regimes, applying retrieval to normal-reference construction requires targeted modeling of regime and local temporal context that accounts for the structural properties of time-series data.

Overall, existing methods have advanced multivariate time-series anomaly detection along several axes, but at the scoring stage most still assume that the normal reference can be provided by a global model, a unified representation space, or a shared candidate set. This paper approaches the problem from the definition and construction of the normal reference, arguing that it should no longer be treated as a global object shared across all queries but dynamically determined by the operating regime and local temporal context of each query. The design of SCAR is built around this goal of condition-aware normal-reference construction.

3 METHOD

3.1 PROBLEM FORMULATION AND OVERVIEW

Let $Y = 0$ denote normal operation and let R be an unobserved operating regime. When a system has multiple regimes, its normal distribution is a mixture of regime-conditional distributions:

$$p(\mathbf{x} \mid Y = 0) = \sum_{r} \pi_r \, p(\mathbf{x} \mid Y = 0, R = r). \tag{1}$$

A global-reference detector searches this mixture directly. Consequently, a normal segment from an incompatible regime can be locally similar to an anomalous query and produce an artificially small anomaly distance. SCAR addresses this reference bias by approximating conditioning on the latent regime and local temporal context before measuring content deviation. The conditioning variables are represented by descriptors learned directly from the training windows.

SCAR detects anomalies by comparing each query patch with a query-specific local normal reference. A useful intuition is a transient whose waveform is normal during one operating mode but indicates a fault during another: waveform-only retrieval may use the former to normalize the latter, whereas SCAR first excludes state-incompatible references. As shown in Fig. 2, the method is organized into three connected modules. (a) Slow-fast decomposition and representation maps a query window $\mathbf{X}$ to a state descriptor $\mathbf{h}$, a gated residual content representation $\mathbf{z}$, and a neighborhood-pooled context representation $\mathbf{c}$. (b) Two-level condition-aware retrieval uses $\mathbf{h}$ to filter the normal reference memory into state-compatible windows, then uses $\mathbf{c}$ to retrieve a query-specific local normal reference. The content representation $\mathbf{z}$ measures the resulting memory deviation. (c) This deviation is fused with state novelty and multi-scale reconstruction diagnostics after each diagnostic has been calibrated by its own empirical CDF.

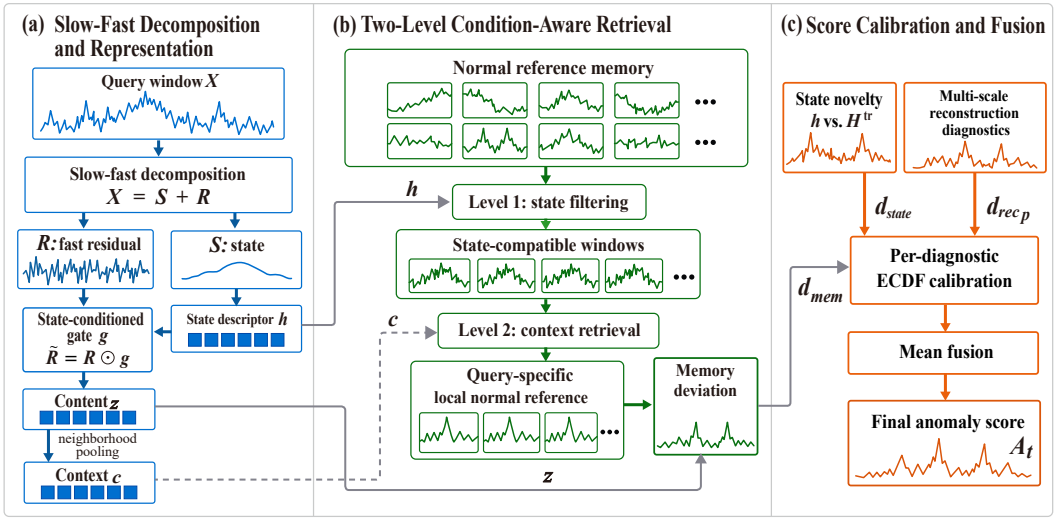


Figure 2: SCAR architecture. (a) Slow-fast decomposition produces state $\mathbf{h}$, content $\mathbf{z}$, and context $\mathbf{c}$ representations. (b) State filtering and context retrieval construct a query-specific normal reference. (c) State novelty, memory deviation, and per-scale reconstruction diagnostics are calibrated by separate empirical CDFs and averaged into the final score A_t.

The representations are learned offline, after which the encoder is frozen to construct the normal reference memory and fit the calibration distributions. At inference, Level 1 and Level 2 refer only to the two retrieval levels: state filtering and context retrieval, respectively.

The three representations have separate roles. The window-level state descriptor $\mathbf{h}$ restricts which historical windows may provide references; the patch-level context representation $\mathbf{c}$ selects temporally compatible patches within those windows; and the content representation $\mathbf{z}$ measures deviation from the resulting local normal reference. Together, they organize the retrieval process into candidate filtering, reference selection, and distance computation.

STSD and the self-supervised objectives organize these representations for the hierarchical retrieval and score-fusion pipeline. Appendix A gives complete pseudocode for Stage A, Stage B, and inference.

3.2 Slow-Fast Decomposition and Representation

Slow-fast decomposition. Industrial time series exhibit both slowly evolving state variation and fast local fluctuations. STSD decomposes the input as $\mathbf{X} = \mathbf{S} + \mathbf{R}$. STSD first obtains a low-frequency-biased initialization in the frequency domain, then refines it with a lightweight temporal convolutional correction network:

$$\mathbf{S} = \mathbf{S}_0 + \Delta_\theta\big(\mathrm{Concat}(\mathbf{X}, \mathbf{S}_0)\big), \qquad \mathbf{R} = \mathbf{X} - \mathbf{S}. \tag{2}$$

The decomposition is guided by a low-frequency prior on $\mathbf{S}$, second-order temporal smoothness regularization, residual-only patch encoding, and self-supervised targets on raw signal patches. The frequency-domain low-pass gate, its initialization, and the smoothness regularization are detailed in Appendix B.2.

State, context, and content representations. Stage A produces three retrieval-oriented representations: a window-level state descriptor $\mathbf{h}$ for coarse candidate filtering, a patch-level context representation $\mathbf{c}$ for selecting a compatible local reference, and a patch-level content representation $\mathbf{z}$ for computing the anomaly distance to that reference.

The state component $\mathbf{S}$ is compressed into a window-level state descriptor $\mathbf{h} = f_{\mathrm{state}}(\mathbf{S}) \in \mathbb{R}^{d_s}$. At each scale $p \in \mathcal{P}$, the residual is partitioned into $N_p = L/p$ non-overlapping patches. A state-conditioned channel-wise gate $\mathbf{g} = \sigma\big(f_{\mathrm{gate}}(\mathbf{h})\big) \in (0,1)^C$ modulates every residual patch:

$\widetilde{\mathbf{R}}_i^{(p)} = \mathbf{R}_i^{(p)} \odot \mathbf{g}$. Each modulated patch is encoded by a shared one-dimensional convolutional trunk followed by a scale-specific projection head:

$$\mathbf{z}_i^{(p)} = f_{\text{scale}}^{(p)}\big(f_{\text{trunk}}(\widetilde{\mathbf{R}}_i^{(p)})\big), \qquad \mathbf{z}_i^{(p)} \in \mathbb{R}^{d_z}. \tag{3}$$

Here, f_{trunk} extracts reusable local residual patterns across patch scales, while $f_{\text{scale}}^{(p)}$ maps the scale-p features into a common d_z-dimensional space. Sharing the trunk transfers local structure across scales; separate projection heads account for their different temporal extents. The context representation is obtained by local mean pooling over neighboring content representations:

$$\mathbf{c}_i^{(p)} = \frac{1}{|\mathcal{C}_i^{(p)}|} \sum_{j \in \mathcal{C}_i^{(p)}} \mathbf{z}_j^{(p)}, \qquad \mathbf{c}_i^{(p)} \in \mathbb{R}^{d_z}, \qquad \mathcal{C}_i^{(p)} = \{j : |j - i| \le r_c,\ 1 \le j \le N_p\}. \tag{4}$$

Masked reconstruction recovers raw signal patches from masked residual-derived representations, encouraging the encoder to retain local residual information while leveraging state information from the gate. Next-patch prediction forecasts the next raw patch from the content representation, promoting temporal coherence in the content-representation space. Both losses are mean ℓ_1 errors averaged across scales. The detailed formulations of the masking, reconstruction, and prediction heads are provided in Appendix B.3.

Combining the two self-supervised objectives with the decomposition regularizer gives the Stage-A objective

$$\mathcal{L} = \mathcal{L}_{\text{mask}} + \lambda_{\text{pred}}\mathcal{L}_{\text{pred}} + \lambda_{\text{smooth}}\mathcal{L}_{\text{smooth}}. \tag{5}$$

3.3 Two-level condition-aware retrieval

The normal reference memory is constructed after the representation encoder has been trained and frozen. We apply it to training windows designated for memory construction, extracting window-level state descriptors and multi-scale patch-representation pairs $(\mathbf{z}, \mathbf{c})$. Deterministic reconstruction scores rank the training patches at each scale; the highest-scoring fraction is excluded by quantile purification, followed by optional coreset compression. The retained normal entries form the memory bank. Details of memory construction, including grouped masking, quantile purification, and coreset compression, appear in Appendix B.4. The retained memory entry at scale p is represented as $(j, i, \mathbf{z}_{j,i}^{(p)}, \mathbf{c}_{j,i}^{(p)})$, where j indexes the source window and i indexes the patch within it. The state bank $\mathcal{H}^{\text{tr}} = \{\mathbf{h}_j\}$ stores the training state descriptors.

Level 1: state filtering. Given a query descriptor $\mathbf{h}_w$, Level 1 filters the memory to the M nearest training windows in state space:

$$\mathcal{J}_M(w) = \text{Top-M NN}(\mathbf{h}_w, \mathcal{H}^{\text{tr}}). \tag{6}$$

The resulting state-compatible windows define the candidate pool for all subsequent patch-level comparisons. This restriction promotes state consistency within the query-specific local normal reference. In parallel, the state-novelty score compares $\mathbf{h}_w$ with the normal state bank $\mathcal{H}^{\text{tr}}$ as the mean squared distance to its k_s nearest neighbors:

$$a_w = \frac{1}{k_s} \sum_{\mathbf{h}' \in \text{NN}_{k_s}(\mathbf{h}_w, \mathcal{H}^{\text{tr}})} \|\mathbf{h}_w - \mathbf{h}'\|_2^2. \tag{7}$$

Level 2: context retrieval. Within the state-compatible candidate patch pool, Level 2 retrieves the K memory patches whose context representations are nearest to the query. They form the query-specific local normal reference:

$$\mathcal{N}_i^{(p)}(w) = \text{Top-K NN}\big(\mathbf{c}_{w,i}^{(p)}, \mathcal{B}^{(p)}(w);\ \text{key} = \mathbf{c}\big), \tag{8}$$

where $\mathcal{B}^{(p)}(w)$ collects all retained memory entries from windows in $\mathcal{J}_M(w)$.

Memory deviation. From the retrieved local reference, the k_{nn} nearest neighbors are further selected by content distance. Their average distance from the query content representation is the patch-level memory deviation:

$$s_{w,i}^{(p,\mathrm{mem})} = \frac{1}{\tilde{k}} \sum_{m=1}^{\tilde{k}} \left\| \mathbf{z}_{w,i}^{(p)} - \mathbf{z}_{j_m,i_m}^{(p)} \right\|_2^2, \tag{9}$$

where (j_m, i_m) denotes the index pair of the m-th content-nearest memory entry within the retrieved neighborhood, and $\tilde{k} = \min\big(k_{\mathrm{nn}}, |\mathcal{N}_i^{(p)}(w)|\big)$.

This sequence makes the roles of the three representations explicit: $\mathbf{h}$ filters the normal reference memory, $\mathbf{c}$ retrieves the query-specific local normal reference, and $\mathbf{z}$ measures the resulting memory deviation. Only this final content-space distance is used as the memory anomaly evidence.

3.4 Score calibration and fusion

SCAR combines three diagnostic views that characterize different forms of deviation from normality. The *state-novelty score* a_w asks whether the slowly varying operating state of the entire query window is covered by the normal state bank. It is a window-level support measure: larger values indicate that the query lies farther from previously observed normal operating states.

The *patch-level memory deviation* $s_{w,i}^{(p,\mathrm{mem})}$ asks whether the local dynamics of a query patch resemble historical normal patches under a compatible state and context. It is a non-parametric, example-based measure: after $\mathbf{h}$ and $\mathbf{c}$ determine the relevant reference set, the score is the content-space distance between $\mathbf{z}$ and its retrieved normal examples.

The *per-scale reconstruction score* $e_{w,i}^{(p)}$ asks how well the learned normal dynamics can recover the observed patch from its surrounding information. It is a model-based measure of local predictability. Scores at shorter patch scales emphasize brief deviations, whereas scores at longer scales emphasize persistent changes. The three diagnostics therefore describe normal-state coverage, deviation from condition-matched examples, and compatibility with the learned temporal model, respectively.

Per-diagnostic calibration and mean fusion. At inference, reconstruction diagnostics are obtained via the same deterministic grouped-masking procedure used during memory construction. The per-scale reconstruction score $e_{w,i}^{(p)}$ (defined in Appendix B.4) and the patch-level memory score $s_{w,i}^{(p,\mathrm{mem})}$ are each expanded to time-step level by assigning the patch score to every time step covered by that patch. Let $s_t^{(p,\mathrm{mem})}$ denote the time-step memory score expanded from scale p. Multi-scale memory evidence is aggregated as

$$s_t^{\mathrm{mem}} = \frac{\sum_{p\in\mathcal{P}} p^{-1} s_t^{(p,\mathrm{mem})}}{\sum_{p\in\mathcal{P}} p^{-1}}. \tag{10}$$

Reconstruction diagnostics are retained separately at each scale, and the state-novelty score is broadcast to every time step in the query window. The diagnostic set is $\Omega = \{\mathrm{mem}, \mathrm{state}\} \cup \{\mathrm{rec}_p \mid p \in \mathcal{P}\}$. The mapping from raw scores to the unified diagnostic notation $d_t^{(\rho)}$ follows the three views in Fig. 2: $d_t^{(\mathrm{mem})} = s_t^{\mathrm{mem}}$ is the memory deviation, $d_t^{(\mathrm{state})} = a_w$ is the state novelty broadcast to every time step in window w, and $d_t^{(\mathrm{rec}_p)}$ is the time-expanded reconstruction diagnostic at scale p.

Each diagnostic $d_t^{(\rho)}$, $\rho \in \Omega$, is calibrated to $[0,1]$ via its empirical CDF $\widehat{F}_\rho$ estimated on the training data after memory construction. The calibrated value is the percentile of the current diagnostic relative to its normal training distribution; for example, a value of 0.99 exceeds 99% of the corresponding training scores. The final anomaly score is the unweighted mean of all calibrated diagnostics:

$$A_t = \frac{1}{|\Omega|} \sum_{\rho\in\Omega} \widehat{F}_\rho\big(d_t^{(\rho)}\big). \tag{11}$$

Details of patch-to-time expansion, multi-scale weighting, and empirical CDF calibration appear in Appendix B.5.

Table 1: Main results (AR: AUROC; AP: average precision). Best: red bold; second: blue underline.

Method	MSL		PSM		SMAP		SMD		SWaT		Avg.	
	AR	AP	AR	AP	AR	AP	AR	AP	AR	AP	AR	AP
Classical shallow detectors												
OCSVM	0.5240	0.1530	0.6190	0.4180	0.3930	0.1020	0.6020	0.1040	0.6570	0.1700	0.5590	0.1894
PCA	0.5520	0.1570	0.6480	0.4680	0.3960	0.1040	0.6790	0.1280	0.8190	0.7260	0.6188	0.3166
IForest	0.5240	0.1140	0.5420	0.3340	0.4870	0.1220	0.6640	0.1220	0.3460	0.0930	0.5126	0.1570
HBOS	0.5740	0.1320	0.6200	0.3940	0.5850	0.1480	0.6260	0.1450	0.8310	**0.7380**	0.6472	0.3114
kNN	0.6230	0.1921	0.7374	**0.5421**	0.6179	0.1622	0.7531	0.1836	0.8177	0.7184	0.7098	0.3597
LOF	0.5624	0.1232	0.7336	0.4482	0.6088	**0.1798**	0.6952	0.1444	0.8013	0.7125	0.6803	0.3216
Reconstruction- and forecasting-based models												
AE	0.5620	0.1480	0.6500	0.4650	0.5220	0.1260	0.7740	0.1880	0.8170	0.7130	0.6650	0.3280
LSTM-VAE	0.5212	0.0452	0.4915	0.4022	0.5083	0.0419	0.5005	0.0415	0.4959	0.0413	0.5035	0.1144
TranAD	0.5755	0.1038	0.6279	0.4300	0.4068	0.1020	0.7212	**0.2575**	0.3230	0.1260	0.5309	0.2039
Transformer and general time-series backbones												
Anomaly Transformer	0.5080	0.1070	0.5140	0.2980	0.5110	0.1300	0.5080	0.0460	0.4050	0.1210	0.4892	0.1404
DCdetector	0.5070	0.1070	0.5010	0.2810	0.5220	0.1320	0.5020	0.0430	0.5010	0.1210	0.5066	0.1368
TimesNet	0.6130	0.1460	0.5920	0.3910	0.4530	0.1140	0.7270	0.1410	0.2880	0.1070	0.5346	0.1798
PatchTST	0.6370	0.1570	0.5860	0.3780	0.4480	0.1150	0.7360	0.1470	0.2420	0.0850	0.5298	0.1764
ModernTCN	0.6330	0.1460	0.5930	0.3850	0.4550	0.1140	0.7220	0.1300	0.2440	0.0930	0.5294	0.1736
iTransformer	0.6110	0.1510	0.5920	0.3830	0.4090	0.1140	0.7450	0.1460	0.2420	0.0840	0.5198	0.1756
Decomposition and time-frequency methods												
TFAD	0.5000	0.1060	0.5000	0.2790	0.5000	0.1280	0.5000	0.0420	0.5000	0.1230	0.5000	0.1356
D3R	0.6526	0.1699	0.5003	0.2631	0.4135	0.1062	0.6420	0.1224	0.5665	0.1330	0.5550	0.1589
DualTF	0.5760	0.1560	0.6000	0.4110	0.4780	0.1220	0.6310	0.0690	0.5670	0.1430	0.5704	0.1802
CATCH	0.6640	0.1670	0.6520	0.4340	0.5040	0.1310	0.8110	0.1720	0.3450	0.1660	0.5952	0.2140
Memory-, prototype-, and retrieval-based methods												
MEMTO	0.4999	0.1048	0.4975	0.2696	0.5959	0.1629	0.7324	0.1035	0.4541	0.1145	0.5560	0.1511
H-PAD	0.5999	0.1506	0.7501	0.5183	0.5913	0.1530	0.7649	0.1405	0.8154	0.5399	0.7043	0.3005
PUAD	0.5656	0.1524	0.7353	0.5300	0.4466	0.1141	0.7632	0.1532	0.8252	0.6580	0.6672	0.3215
PGRF-Net	0.5120	0.1329	0.6991	0.4639	0.4947	0.1265	0.5118	0.0601	0.5203	0.1209	0.5476	0.1809
PaAno	0.7115	0.1980	0.5218	0.3380	0.4735	0.1335	0.7620	0.1737	0.2992	0.0946	0.5536	0.1876
SCAR (Ours)	**0.7910**	**0.2690**	**0.8015**	0.5186	**0.6274**	0.1672	**0.8155**	0.1823	**0.8444**	0.6958	**0.7760**	**0.3666**

4 Experiments

4.1 Experimental setup

Datasets. We evaluate point-wise detection on spacecraft telemetry (MSL and SMAP; Hundman et al., 2018), server monitoring (PSM; Abdulaal et al., 2021; SMD; Su et al., 2019), and industrial control (SWaT; Goh et al., 2016). The complete extended TEP dataset (Liu et al., 2024) covers six modes and 28 faults per mode (168 sequences). Its sequence-level labels support mechanism validation, not point-wise comparison. Appendix H evaluates all 350 univariate and 180 multivariate TSB-AD test series; Table 4 gives dataset statistics.

Metrics. We report threshold-free point-wise AUROC for overall ranking and average precision (AP) for top-ranked anomalies under class imbalance.

Baselines. Table 1 compares 24 baselines spanning shallow detectors, reconstruction/forecasting, general backbones, time-frequency methods, and memory/prototype/retrieval approaches (Tuli et al., 2022; Xu et al., 2022; Wu et al., 2025; Li et al., 2023; Park & Kang, 2026). All methods share splits, preprocessing, and AR/AP evaluation. Scores covered by CATCH (Wu et al., 2025) are taken from that benchmark; other methods are reproduced from official implementations.

Model selection and implementation. Tuning uses only a training-derived validation split; test data are reserved for final evaluation. Table 5 in Appendix C lists the defaults.

4.2 Main benchmark results

Table 1 shows that SCAR achieves the best AUROC on every dataset and the best macro-average AUROC and AP. It also has the best MSL AP; kNN, LOF, TranAD, and HBOS lead AP on PSM, SMAP, SMD, and SWaT, respectively. Against PaAno, the closest reproducible patch-reference method, SCAR improves average AUROC/AP by 22.24/17.90 percentage points. Appendices D and M provide extended metrics and cases.

4.3 Ablation Study

Table 2: Ablations: A1 representation, A2 retrieval, A3 calibration, A4 scale. Colors follow Table 1.

Group	Variant	MSL		PSM		SMAP		SMD		SWaT		Avg.	
		AR	AP	AR	AP	AR	AP	AR	AP	AR	AP	AR	AP
A1	w/o decomposition	0.7343	0.2468	0.7993	**0.5933**	0.5052	0.1241	0.8101	0.1857	0.7964	0.3238	0.7291	0.2947
A1	w/o state-guided modulation	0.6960	0.2035	0.7861	0.5656	0.4549	0.1151	0.8111	0.1940	0.7844	0.4383	0.7065	0.3033
A2	global retrieval	0.7200	0.2344	0.7943	0.5444	0.5307	0.1306	0.8115	0.1772	0.4745	0.1075	0.6662	0.2388
A2	state-only retrieval	0.7092	0.2282	0.8009	0.5167	0.5118	0.1215	0.8131	0.1790	0.7871	0.3262	0.7244	0.2743
A2	context-only retrieval	0.7110	0.2298	0.8005	0.5169	0.5184	0.1266	0.8110	0.1776	0.4750	0.1075	0.6632	0.2317
A3	raw-max	0.7895	0.2656	0.7093	0.4976	0.4611	0.1094	0.7296	0.1731	0.6860	0.2552	0.6751	0.2602
A3	zscore-mean	0.7021	0.2195	0.5795	0.3436	0.5047	0.1218	0.7624	0.1552	0.7053	0.1947	0.6508	0.2070
A4	single-scale short	0.7558	0.2400	0.7874	0.5148	0.5052	0.1197	0.7956	**0.2243**	0.6949	0.1870	0.7078	0.2572
A4	single-scale long	0.7652	0.2516	0.7740	0.5551	0.4591	0.1248	0.8108	0.2039	0.8221	**0.7228**	0.7262	**0.3716**
	SCAR (Ours)	**0.7910**	**0.2690**	**0.8015**	0.5186	**0.6274**	**0.1672**	**0.8155**	0.1823	**0.8444**	0.6958	**0.7760**	0.3666

Table 2 tests the complete chain: A1 organizes operating conditions, A2 retrieves local references, A3 calibrates evidence, and A4 covers temporal scales. Global, state-only, and context-only retrieval all underperform the two-level design, especially on SMAP and SWaT, showing that state compatibility and local relevance are complementary. A1 mainly helps heterogeneous MSL, SMAP, and SWaT; A3 improves cross-dataset consistency; and A4 avoids the dataset-specific tradeoffs of either single scale. The gain thus depends on the chain as a whole.

4.4 Mechanism Validation

The complete TEP evaluation directly tests all six modes and 28 faults per mode. Table 3 reports SMC@K/SFR of 1.00/9.65 and SMR@K of 0.84, showing mode recovery and regime-matched retrieval. The normalized fault-normal gap is 1.38; cross-mode FPR standard deviation and 95%/99% tail errors are 0.073 and 0.057/0.014; dominant-evidence consistency is 0.74. Thus compatible references remain discriminative and calibrated while retaining complementary cues. Definitions and the full six-mode visualization appear in Appendices F and G. On unlabeled benchmarks, Appendix I shows that state filtering materially changes reference sets while full retrieval preserves strong temporal continuity. Appendices H, J, K, and L report TSB-AD generalization, memory robustness, scale controls, and cost.

Table 3: Mechanism validation on the complete six-mode TEP evaluation (168 fault sequences).

Group	Metric	Dir.	Value	Implication
State	SMC@K	↑	1.0000	Nearest neighbors are mode-consistent.
	SFR	↑	9.6520	Operating modes are well separated.
Retrieval	SMR@K	↑	0.8357	Most retrieved memories match the query mode.
Separation	Normalized fault-normal gap	↑	1.3754	Faults are clearly separated from normals.
Calibration	FPR std. across modes	↓	0.0731	False-alarm rates are stable across modes.
	95% tail error	↓	0.0572	Tail calibration remains stable.
	99% tail error	↓	0.0135	Extreme-tail calibration remains stable.
Evidence	Dom. evidence consistency	↑	0.7357	Faults activate distinct evidence channels.

↑: higher is better; ↓: lower is better. SMC@K: State-Mode Consistency@K; SFR: State Separation Ratio; SMR@K: Same-Mode Retrieval Ratio@K.

5 Conclusion

SCAR replaces a global normal reference with one matched to each query's regime and local context. It achieves the best AUROC on all five benchmarks; ablations support the complete design, the full six-mode TEP study confirms condition-compatible retrieval and calibration, and 530 TSB-AD series establish broader generalization. Memory-dependent inference and the frequency-domain decomposition prior remain limitations, but the underlying condition-aware view is independent of either implementation.

AI USE STATEMENT

Generative AI tools were used only for writing, editing, and formatting assistance. They were not used as a component of the proposed method or to generate experimental results. The authors reviewed all AI-assisted material and take responsibility for the final content, claims, and artifacts in this work.

REPRODUCIBILITY STATEMENT

The model architecture, learning objectives, retrieval procedure, and anomaly-score construction are specified in Section 3. Appendix A provides complete pseudocode for training, memory construction, calibration, and inference; Appendix B gives the extended formulations; and Appendix C reports the dataset statistics, evaluation protocol, and implementation settings. Appendices D and E report extended evaluation metrics and diagnostic score decomposition, respectively. Appendices F and G provide TEP metric definitions and mechanism visualization. Appendices H, I, and J report cross-benchmark generalization, retrieval diagnostics, and memory robustness. Appendices K and L report sensitivity and scale controls and computational cost, respectively. Appendix M presents qualitative cases on the five benchmark datasets.

REFERENCES

Ahmed Abdulaal, Zhuanghua Liu, and Tomer Lancewicki. Practical approach to asynchronous multivariate time series anomaly detection and localization. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pp. 2485–2494, 2021.

Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. LOF: Identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, pp. 93–104, 2000.

Mingxiu Cai, Zhe Zhang, Gaochang Wu, Tianyou Chai, and Xiatian Zhu. RAID: Retrieval-augmented anomaly detection. *arXiv preprint arXiv:2602.19611*, 2026.

Ailin Deng and Bryan Hooi. Graph neural network-based anomaly detection in multivariate time series. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(5):4027–4035, 2021.

Jonathan Goh, Sridhar Adepu, Khurum Nazir Junejo, and Aditya Mathur. A dataset to support research in the design of secure water treatment systems. In *International Conference on Critical Information Infrastructures Security*, pp. 88–99. Springer, 2016.

Markus Goldstein and Andreas Dengel. Histogram-based outlier score (HBOS): A fast unsupervised anomaly detection algorithm. *KI-2012: Poster and Demo Track*, 1:59–63, 2012.

Kyle Hundman, Valentino Constantinou, Christopher Laporte, Ian Colwell, and Tom Soderstrom. Detecting spacecraft anomalies using LSTMs and nonparametric dynamic thresholding. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 387–395, 2018.

Makoto Imamura. Generalized discords for time series anomaly detection with flexible subsequence lengths. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 1013–1024, 2025.

Jahoon Jeong and Hyunsoo Yoon. PGRF-Net: A prototype-guided relational fusion network for diagnostic multivariate time-series anomaly detection. In *The Fourteenth International Conference on Learning Representations*, 2026.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547, 2019.

Beibu Li, Qichao Shentu, Yang Shu, Hui Zhang, Ming Li, Ning Jin, Bin Yang, and Chenjuan Guo. CrossAD: Time series anomaly detection with cross-scale associations and cross-window modeling. In *Advances in Neural Information Processing Systems*, 2025.

Yuxin Li, Wenchao Chen, Bo Chen, Dongsheng Wang, Long Tian, and Mingyuan Zhou. Prototype-oriented unsupervised anomaly detection for multivariate time series. In *International Conference on Machine Learning*, pp. 19407–19424. PMLR, 2023.

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 *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pp. 3220–3230, 2021.

Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. In *2008 Eighth IEEE International Conference on Data Mining*, pp. 413–422. IEEE, 2008.

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

Zeyi Liu, Chen Li, and Xiao He. Evidential ensemble preference-guided learning approach for real-time multimode fault diagnosis. *IEEE Transactions on Industrial Informatics*, 20(4):5495–5504, 2024. doi: 10.1109/TII.2023.3332112.

Yue Lu, Renjie Wu, Abdullah Mueen, Maria A Zuluaga, and Eamonn Keogh. Matrix profile XXIV: Scaling time series anomaly detection to trillions of datapoints and ultra-fast arriving data streams. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 1173–1182, 2022.

Pankaj Malhotra, Lovekesh Vig, Gautam Shroff, and Puneet Agarwal. Long short term memory networks for anomaly detection in time series. In *Proceedings of the 23rd European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)*, pp. 89–94, 2015.

Pankaj Malhotra, Anusha Ramakrishnan, Gaurangi Anand, Lovekesh Vig, Puneet Agarwal, and Gautam Shroff. LSTM-based encoder-decoder for multi-sensor anomaly detection. *arXiv preprint arXiv:1607.00148*, 2016.

Youngeun Nam, Susik Yoon, Yooju Shin, Minyoung Bae, Hwanjun Song, Jae-Gil Lee, and Byung Suk Lee. Breaking the time-frequency granularity discrepancy in time-series anomaly detection. In *Proceedings of the ACM Web Conference 2024*, pp. 4204–4215, 2024.

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

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

Sridhar Ramaswamy, Rajeev Rastogi, and Kyuseok Shim. Efficient algorithms for mining outliers from large data sets. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, pp. 427–438, 2000.

Karsten Roth, Latha Pemula, Joaquin Zepeda, Bernhard Schölkopf, Thomas Brox, and Peter Gehler. Towards total recall in industrial anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14318–14328, 2022.

Mayu Sakurada and Takehisa Yairi. Anomaly detection using autoencoders with nonlinear dimensionality reduction. In *Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis*, pp. 4–11, 2014.

Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12716–12725, 2019.

Bernhard Schölkopf, Robert C Williamson, Alex Smola, John Shawe-Taylor, and John Platt. Support vector method for novelty detection. *Advances in Neural Information Processing Systems*, 12, 1999.

Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. In *International Conference on Learning Representations*, 2018.

Ke-Yuan Shen. Learn hybrid prototypes for multivariate time series anomaly detection. In *The Thirteenth International Conference on Learning Representations*, 2025.

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 *International Conference on Learning Representations*, 2025.

Mei-Ling Shyu, Shu-Ching Chen, Kanoksri Sarinnapakorn, and Liwu Chang. A novel anomaly detection scheme based on principal component classifier. In *Proceedings of the Foundations and New Directions in Data Mining Workshop, IEEE International Conference on Data Mining (ICDM'03)*, pp. 1–8, 2003.

Junho Song, Keonwoo Kim, Jeonglyul Oh, and Sungzoon Cho. Memto: Memory-guided transformer for multivariate time series anomaly detection. *Advances in Neural Information Processing Systems*, 36:57947–57963, 2023.

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. doi: 10.14778/3514061.3514067.

Chengsen Wang, Zirui Zhuang, Qi Qi, Jingyu Wang, Xingyu Wang, Haifeng Sun, and Jianxin Liao. Drift doesn't matter: Dynamic decomposition with diffusion reconstruction for unstable multivariate time series anomaly detection. *Advances in Neural Information Processing Systems*, 36:10758–10774, 2023.

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.

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.

Chaoli Zhang, Tian Zhou, Qingsong Wen, and Liang Sun. TFAD: A decomposition time series anomaly detection architecture with time-frequency analysis. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pp. 2497–2507, 2022.

Zhen Zhang, Zhiqiang Geng, and Yongming Han. Graph structure change-based anomaly detection in multivariate time series of industrial processes. *IEEE Transactions on Industrial Informatics*, 20 (4):6457–6466, 2024.

Chong Zhou and Randy C Paffenroth. Anomaly detection with robust deep autoencoders. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 665–674, 2017.

Mingle Zhou, Xingli Wang, Jiachen Li, Delong Han, and Gang Li. Unsupervised dual-domain memory model for time series anomaly detection. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pp. 5538–5546, 2025.

A PSEUDOCODE OF THE SCAR PIPELINE

All notation follows Section 3 and Appendix B. $\mathcal{D}_{\mathrm{A}}$ denotes the Stage-A training windows, $\mathcal{D}_{\mathrm{B}}$ denotes the full training windows used for memory construction and empirical CDF fitting, and $\Omega = \{\mathrm{mem}, \mathrm{state}\} \cup \{\mathrm{rec}_p \mid p \in \mathcal{P}\}$. Throughout the pseudocode, *Stage* denotes an offline processing phase: Stage A learns representations and Stage B constructs memory and fits calibration CDFs. *Level* is reserved for the two retrieval operations used in diagnostic extraction: state filtering (Level 1) and context retrieval (Level 2). These offline stages are distinct from the three modules in Fig. 2: panel (a) constructs the representations, panel (b) retrieves a query-specific local normal reference and measures memory deviation, and panel (c) calibrates and averages the diagnostics. Stage B prepares both the normal reference memory for panel (b) and the calibration CDFs for panel (c).

A.1 STAGE A: SLOW-FAST DECOMPOSITION AND REPRESENTATION

Algorithm 1: Stage A: Slow-fast Decomposition and Representation

Input: Stage-A training windows $\mathcal{D}_{\mathrm{A}}$; patch sizes $\mathcal{P}$; masking ratio ρ; context radius r_c; loss weights λ_{pred} and $\lambda_{\mathrm{smooth}}$
Output: Trained Stage-A model

for $epoch \leftarrow 1$ **to** E **do**
  foreach *mini-batch* $\mathbf{X} \sim \mathcal{D}_{\mathrm{A}}$ **do**
    $(\mathbf{S}, \mathbf{R}) \leftarrow \mathrm{STSD}(\mathbf{X})$
    $\mathbf{h} \leftarrow f_{\mathrm{state}}(\mathbf{S})$
    $\mathbf{g} \leftarrow \sigma(f_{\mathrm{gate}}(\mathbf{h}))$
    foreach $p \in \mathcal{P}$ **do**
      $(\mathbf{R}^{(p)}, \mathbf{X}^{(p)}) \leftarrow \textsc{Patchify}(\mathbf{R}, \mathbf{X}, p)$; $N_p \leftarrow L/p$
      $\widetilde{\mathbf{R}}^{(p)} \leftarrow \mathbf{R}^{(p)} \odot \mathbf{g}$
      $\mathbf{z}^{(p)} \leftarrow f_{\mathrm{scale}}^{(p)}\big(f_{\mathrm{trunk}}(\widetilde{\mathbf{R}}^{(p)})\big)$
      $\mathbf{c}^{(p)} \leftarrow \textsc{LocalMean}(\mathbf{z}^{(p)}, r_c)$
      sample mask set $\mathcal{S}^{(p)}$ with $|\mathcal{S}^{(p)}| = \max(1, \lfloor \rho N_p \rfloor)$
      replace $\mathbf{z}_i^{(p)}$ by $\mathbf{z}_{\mathrm{mask}}$ for $i \in \mathcal{S}^{(p)}$ to form $\bar{\mathbf{Z}}^{(p)}$
      $\widehat{\mathbf{X}}^{(p)} \leftarrow \textsc{CompletionHead}^{(p)}(\bar{\mathbf{Z}}^{(p)})$
      $\widehat{\mathbf{X}}_{2:N_p}^{(p)} \leftarrow \textsc{PredHead}^{(p)}(\mathbf{z}_{1:N_p-1}^{(p)})$
      compute $\mathcal{L}_{\mathrm{mask}}^{(p)}$ only on masked positions in $\mathcal{S}^{(p)}$
      compute $\mathcal{L}_{\mathrm{pred}}^{(p)}$
    $\mathcal{L}_{\mathrm{mask}} \leftarrow \frac{1}{|\mathcal{P}|} \sum_{p \in \mathcal{P}} \mathcal{L}_{\mathrm{mask}}^{(p)}$
    $\mathcal{L}_{\mathrm{pred}} \leftarrow \frac{1}{|\mathcal{P}|} \sum_{p \in \mathcal{P}} \mathcal{L}_{\mathrm{pred}}^{(p)}$
    compute $\mathcal{L}_{\mathrm{smooth}}$ from $\mathbf{S}$
    $\mathcal{L} \leftarrow \mathcal{L}_{\mathrm{mask}} + \lambda_{\mathrm{pred}} \mathcal{L}_{\mathrm{pred}} + \lambda_{\mathrm{smooth}} \mathcal{L}_{\mathrm{smooth}}$
    update all Stage-A parameters by back-propagation
return trained Stage-A model

A.2 DETERMINISTIC GROUPED RECONSTRUCTION SCORING

Algorithm 2: Deterministic Grouped Reconstruction Scoring

Input: Frozen Stage-A model; input batch $\mathbf{X}$; patch sizes $\mathcal{P}$; number of groups n_g
Output: Encoded features and per-scale deterministic reconstruction scores $\{e^{(p)}\}_{p \in \mathcal{P}}$

encode $\mathbf{X}$ to obtain $\mathbf{S}$, $\mathbf{R}$, $\mathbf{h}$, $\{\mathbf{z}^{(p)}\}_{p \in \mathcal{P}}$, $\{\mathbf{c}^{(p)}\}_{p \in \mathcal{P}}$, and $\{\mathbf{X}^{(p)}\}_{p \in \mathcal{P}}$
foreach $p \in \mathcal{P}$ **do**
  let N_p be the number of patches at scale p
  initialize $e^{(p)} \leftarrow \mathbf{0}$ for all patch positions
  for $g \leftarrow 1$ **to** n_g **do**
    define group
$$\mathcal{G}_g^{(p)} = \{i \mid (i-1) \bmod n_g = g - 1\};$$
    mask only positions in $\mathcal{G}_g^{(p)}$ to form the grouped masked token sequence
    run $\textsc{CompletionHead}^{(p)}$ on this grouped masked sequence
    foreach $i \in \mathcal{G}_g^{(p)}$ **do**
      $e_i^{(p)} \leftarrow \frac{1}{pC} \left\| \widehat{\mathbf{X}}_i^{(p)} - \mathbf{X}_i^{(p)} \right\|_1$
return encoded features and $\{e^{(p)}\}_{p \in \mathcal{P}}$

A.3 STAGE B: NORMAL REFERENCE MEMORY CONSTRUCTION AND CDF FITTING

Algorithm 3: Stage B: Normal Reference Memory Construction and CDF Fitting

Input: Frozen Stage-A model; full training windows $\mathcal{D}_{\mathrm{B}}$; patch sizes $\mathcal{P}$; cleaning ratio r
Output: Memory banks $\{\widetilde{\mathcal{M}}_p^{\mathrm{tr}}\}_{p\in\mathcal{P}}$; state bank $\mathcal{H}^{\mathrm{tr}}$; empirical CDFs $\{\widehat{F}_\rho\}_{\rho\in\Omega}$

initialize $\mathcal{H}^{\mathrm{tr}} \leftarrow \emptyset$
foreach $p \in \mathcal{P}$ **do**
  initialize $\mathcal{M}_p^{\mathrm{raw}} \leftarrow \emptyset$; $\mathcal{E}_p \leftarrow \emptyset$
foreach *training batch* $\mathbf{X} \sim \mathcal{D}_{\mathrm{B}}$ *with window indices* j **do**
  $(\text{encoded}, \{e^{(p)}\}_{p\in\mathcal{P}}) \leftarrow \textsc{DeterministicScores}(\mathbf{X})$
  append window-level state descriptors $\mathbf{h}_j$ to $\mathcal{H}^{\mathrm{tr}}$
  foreach $p \in \mathcal{P}$ **do**
    foreach *window* j *and patch position* i *in the batch* **do**
      append $(j, i, \mathbf{z}_{j,i}^{(p)}, \mathbf{c}_{j,i}^{(p)})$ to $\mathcal{M}_p^{\mathrm{raw}}$
      append $e_{j,i}^{(p)}$ to $\mathcal{E}_p$
foreach $p \in \mathcal{P}$ **do**
  $\tau_p \leftarrow (1-r)$-quantile of $\mathcal{E}_p$
  retain only entries in $\mathcal{M}_p^{\mathrm{raw}}$ with $e_{j,i}^{(p)} \le \tau_p$
  optionally compress retained entries into $\widetilde{\mathcal{M}}_p^{\mathrm{tr}}$
build a state-space index on $\mathcal{H}^{\mathrm{tr}}$
run the same pointwise diagnostic extraction as in Algorithm 4 on $\mathcal{D}_{\mathrm{B}}$, but without CDF calibration, to obtain covered training diagnostics
flatten the covered training time-step diagnostics into $\{d_n^{(\rho,\mathrm{tr})}\}_{n=1}^{N_\rho}$ for each $\rho \in \Omega$
`// No self-exclusion rule is applied during calibration.`
foreach $\rho \in \Omega$ **do**
  fit empirical CDF $\widehat{F}_\rho$ from $\{d_n^{(\rho,\mathrm{tr})}\}_{n=1}^{N_\rho}$
return $\{\widetilde{\mathcal{M}}_p^{\mathrm{tr}}\}_{p\in\mathcal{P}}$, $\mathcal{H}^{\mathrm{tr}}$, and $\{\widehat{F}_\rho\}_{\rho\in\Omega}$

A.4 INFERENCE AND POINTWISE ANOMALY SCORING

Algorithm 4 computes the raw diagnostics. The state-space search supports both Level 1 filtering and the separate state-novelty diagnostic. Level 2 selects the local normal reference using context; content distances within this reference then yield memory deviation. Reconstruction diagnostics are computed by Algorithm 2 at each patch scale.

Algorithm 4: Inference: Window-Level Diagnostic Extraction

Input: Frozen Stage-A model; query windows $\mathcal{D}_{\mathrm{te}}$; memory banks $\{\widetilde{\mathcal{M}}_p^{\mathrm{tr}}\}_{p\in\mathcal{P}}$; state bank $\mathcal{H}^{\mathrm{tr}}$; retrieval parameters M, K, k_s, k_{nn}
Output: Window-level diagnostics $\{d_{w,\ell}^{(\rho)}\}_{\rho\in\Omega}$

foreach *query batch* $\mathbf{X} \sim \mathcal{D}_{\mathrm{te}}$ *with window start indices* $\{s_w\}$ **do**
  $(\text{encoded}, \{e^{(p)}\}_{p\in\mathcal{P}}) \leftarrow \textsc{DeterministicScores}(\mathbf{X})$
  foreach *window* w *in the batch* **do**
    retrieve the top $\max(M, k_s)$ nearest training windows in state space using $\mathbf{h}_w$
    set $\mathcal{J}_M(w)$ as the top-M state-compatible windows (Level 1: state filtering)
    $a_w \leftarrow \frac{1}{k_s} \sum_{\mathbf{h}' \in \mathrm{NN}_{k_s}(\mathbf{h}_w, \mathcal{H}^{\mathrm{tr}})} \left\|\mathbf{h}_w - \mathbf{h}'\right\|_2^2$
    foreach $p \in \mathcal{P}$ **do**
      form the state-filtered candidate patch pool $\mathcal{B}^{(p)}(w)$ from retained memory entries in windows $\mathcal{J}_M(w)$
      foreach *query patch* i **do**
        `// Level 2: context retrieval of the local normal reference.`
        $\mathcal{N}_i^{(p)}(w) \leftarrow \text{Top-K NN}\big(\mathbf{c}_{w,i}^{(p)}, \mathcal{B}^{(p)}(w);\ \text{key} = \mathbf{c}\big)$
        select for memory-deviation computation $\tilde{k} = \min(k_{\mathrm{nn}}, |\mathcal{N}_i^{(p)}(w)|)$ content-nearest entries within $\mathcal{N}_i^{(p)}(w)$
        denote these selected entries by $\{(j_m, i_m)\}_{m=1}^{\tilde{k}}$
        $s_{w,i}^{(p,\mathrm{mem})} \leftarrow \frac{1}{\tilde{k}} \sum_{m=1}^{\tilde{k}} \left\|\mathbf{z}_{w,i}^{(p)} - \mathbf{z}_{j_m,i_m}^{(p)}\right\|_2^2$
      expand each patch-level memory deviation $s_{w,i}^{(p,\mathrm{mem})}$ to window-level time-step scores $s_{w,\ell}^{(p,\mathrm{mem})}$
      expand each reconstruction score $e_{w,i}^{(p)}$ in the same way to obtain $d_{w,\ell}^{(\mathrm{rec}_p)}$
    $s_{w,\ell}^{\mathrm{mem}} \leftarrow \frac{\sum_{p\in\mathcal{P}} p^{-1} s_{w,\ell}^{(p,\mathrm{mem})}}{\sum_{p\in\mathcal{P}} p^{-1}}$, for $\ell = 1, \ldots, L$
    set $d_{w,\ell}^{(\mathrm{mem})} \leftarrow s_{w,\ell}^{\mathrm{mem}}$, for $\ell = 1, \ldots, L$
    set $d_{w,\ell}^{(\mathrm{state})} \leftarrow a_w$, for $\ell = 1, \ldots, L$
return window-level diagnostics $\{d_{w,\ell}^{(\rho)}\}_{\rho\in\Omega}$

A.5 SCORE CALIBRATION AND MEAN FUSION

Algorithm 5 first averages raw diagnostic values over overlapping windows. It then applies a separate empirical CDF to each pointwise diagnostic and takes their unweighted mean, as in panel (c) of Fig. 2. The diagnostic set has $|\mathcal{P}| + 2$ members: one memory-deviation diagnostic, one state-novelty diagnostic, and one reconstruction diagnostic per scale.

Algorithm 5: Inference: Per-diagnostic CDF Calibration and Mean Fusion

Input: Window-level diagnostics $\{d_{w,\ell}^{(\rho)}\}_{\rho\in\Omega}$; window start indices $\{s_w\}$; CDFs $\{\widehat{F}_\rho\}_{\rho\in\Omega}$
Output: Pointwise anomaly scores $\{A_t\}$

initialize pointwise accumulators $\mathrm{sum}_t^{(\rho)} \leftarrow 0$ for all $\rho \in \Omega$
initialize counts $\mathrm{cnt}_t \leftarrow 0$
foreach *window* w **do**
  for $\ell \leftarrow 1$ **to** L **do**
    $t \leftarrow s_w + \ell - 1$
    add $d_{w,\ell}^{(\mathrm{mem})}$ to $\mathrm{sum}_t^{(\mathrm{mem})}$
    add $d_{w,\ell}^{(\mathrm{state})}$ to $\mathrm{sum}_t^{(\mathrm{state})}$
    foreach $p \in \mathcal{P}$ **do**
      add $d_{w,\ell}^{(\mathrm{rec}_p)}$ to $\mathrm{sum}_t^{(\mathrm{rec}_p)}$
    increment cnt_t
foreach *time step* t **do**
  average overlapping-window contributions:

$$d_t^{(\rho)} \leftarrow \frac{\mathrm{sum}_t^{(\rho)}}{\mathrm{cnt}_t}, \qquad \forall \rho \in \Omega;$$

  compute the final calibrated anomaly score:

$$A_t \leftarrow \frac{1}{|\Omega|} \sum_{\rho\in\Omega} \widehat{F}_\rho\left(d_t^{(\rho)}\right).$$

return $\{A_t\}$

B EXTENDED METHOD FORMULATIONS

This section provides the mathematical formulations omitted from the main text. All notation follows Section 3. The conditional-reference formulation explains the two retrieval levels in Fig. 2(b); the decomposition and patch-encoding details support panel (a); and time-step aggregation, calibration, and mean fusion specify the diagnostic inputs and scoring operations in panel (c).

B.1 CONDITIONAL-REFERENCE FORMULATION AND REPRESENTATION ROLES

Equation 1 motivates replacing a single global reference set with an empirical approximation to a normal distribution conditioned on the query state and context. For query window w and scale p, SCAR constructs the nested sets

$$\widetilde{\mathcal{M}}_p^{\mathrm{tr}} \supseteq \mathcal{B}^{(p)}(w) \supseteq \mathcal{N}_i^{(p)}(w), \tag{12}$$

where $\widetilde{\mathcal{M}}_p^{\mathrm{tr}}$ is the complete retained memory at scale p, $\mathcal{B}^{(p)}(w)$ contains only entries whose source windows pass state filtering, and $\mathcal{N}_i^{(p)}(w)$ contains the context-nearest entries for query patch i. The memory deviation is then evaluated in content space only within $\mathcal{N}_i^{(p)}(w)$.

The representations are therefore deliberately asymmetric. The window-level state descriptor $\mathbf{h}$ is the Level 1 state-filtering key and determines admissibility of source windows. The context descriptor $\mathbf{c}$ is the Level 2 context-retrieval key and determines relevance within the admissible pool. The content descriptor $\mathbf{z}$ is the value used for anomaly-distance computation after the reference set has been fixed. Their use in scoring follows state filtering, context retrieval, and content-space distance computation, respectively. Representation construction follows the dependencies in panel (a): $\mathbf{h}$ conditions the residual gate, the gated residual yields $\mathbf{z}$, and neighborhood pooling of $\mathbf{z}$ yields $\mathbf{c}$ (Eqs. 3–4).

B.2 SLOW-FAST DECOMPOSITION DETAILS

B.2.1 FREQUENCY-DOMAIN LOW-PASS GATE

Let $\mathbf{X} \in \mathbb{R}^{L\times C}$ be an input window and $\mathbf{x}_c \in \mathbb{R}^L$ its c-th variable. The real-input FFT along the temporal axis yields $\hat{\mathbf{x}}_c \in \mathbb{C}^F$, where

$$F = \lfloor L/2 \rfloor + 1. \tag{13}$$

Let $\boldsymbol{\alpha} \in \mathbb{R}^{C\times F}$ be the learnable gate logits, with gate values $\mathbf{G} = \sigma(\boldsymbol{\alpha}) \in (0,1)^{C\times F}$. Element-wise,

$$G_{c,k} = \sigma(\alpha_{c,k}), \qquad \hat{S}_{0,c,k} = G_{c,k}\,\hat{x}_{c,k}. \tag{14}$$

Stacking all channels yields the channel-by-frequency representation

$$\mathcal{F}(\mathbf{X}) \in \mathbb{C}^{C\times F}. \tag{15}$$

The initial state component is then recovered via the inverse transform:

$$\mathbf{S}_0 = \mathcal{F}^{-1}\big(\mathbf{G} \odot \mathcal{F}(\mathbf{X})\big). \tag{16}$$

B.2.2 LOW-PASS INITIALIZATION

To impose a soft low-pass bias, define the normalized frequency index as

$$\omega_k = \frac{k}{F-1}, \qquad k = 0, \ldots, F-1. \tag{17}$$

The gate logits are then initialized as

$$\alpha_{c,k}^{(0)} = \beta(\omega_0 - \omega_k), \qquad \forall\, c,\ k, \tag{18}$$

where $\omega_0 \in (0,1)$ is a normalized cutoff frequency and $\beta > 0$ controls the transition sharpness. The defaults are $\omega_0 = 0.25$ and $\beta = 4$. Because $\boldsymbol{\alpha}$ is tied to F, this formulation assumes a fixed window length L.

B.2.3 TEMPORAL REFINEMENT AND SMOOTHNESS

The correction network Δ_θ (Eq. 2) consists of three temporal 1-D convolutional layers with kernel sizes 7, 5, and 3, stride 1, symmetric padding, and GELU activations between layers.

To promote smooth temporal evolution of the state component, a second-order ℓ_1 regularizer is applied:

$$\mathcal{L}_{\text{smooth}} = \frac{1}{(L-2)C} \sum_{t=2}^{L-1} \sum_{c=1}^{C} \left| S_{t+1,c} - 2S_{t,c} + S_{t-1,c} \right|. \tag{19}$$

The default weight is $\lambda_{\text{smooth}} = 0.01$.

B.3 PATCH ENCODING AND SELF-SUPERVISED HEADS

B.3.1 PATCH PARTITION

For simplicity, L is assumed divisible by every patch size $p \in \mathcal{P}$, yielding

$$N_p = L/p. \tag{20}$$

B.3.2 MASKED RECONSTRUCTION HEAD

At each scale p, a mask set $\mathcal{S}^{(p)} \subset \{1, \ldots, N_p\}$ is randomly sampled such that

$$|\mathcal{S}^{(p)}| = \max\big(1, \lfloor \rho N_p \rfloor\big), \tag{21}$$

where ρ is the masking ratio. Masked positions are replaced by a learnable mask token $\mathbf{z}_{\text{mask}} \in \mathbb{R}^{d_z}$:

$$\bar{\mathbf{z}}_i^{(p)} = \begin{cases} \mathbf{z}_{\text{mask}}, & i \in \mathcal{S}^{(p)}, \\ \mathbf{z}_i^{(p)}, & i \notin \mathcal{S}^{(p)}. \end{cases} \tag{22}$$

Let $\bar{\mathbf{Z}}^{(p)}$ denote the resulting token sequence. The completion head applies fixed sinusoidal positional encoding followed by a Transformer encoder:

$$\mathbf{H}^{(p)} = f_{\text{trans}}^{(p)}\big(\text{PE}(\bar{\mathbf{Z}}^{(p)})\big). \tag{23}$$

The raw patch is then reconstructed as

$$\widehat{\mathbf{X}}_i^{(p)} = f_{\text{out}}^{(p)}\big(\mathbf{H}_i^{(p)}\big) \in \mathbb{R}^{p\times C}. \tag{24}$$

The per-scale reconstruction loss over masked positions is

$$\mathcal{L}_{\text{mask}}^{(p)} = \frac{1}{|\mathcal{S}^{(p)}|} \sum_{i\in\mathcal{S}^{(p)}} \frac{1}{pC} \left\| \widehat{\mathbf{X}}_i^{(p)} - \mathbf{X}_i^{(p)} \right\|_1. \tag{25}$$

B.3.3 Next-patch prediction head

Next-patch prediction is an auxiliary representation-learning objective in Stage A. Its error is not a diagnostic in the final fusion set Ω; the per-scale diagnostics in Fig. 2(c) are deterministic masked-reconstruction errors. The prediction head forecasts the next raw patch from the current content representation:

$$\widehat{\mathbf{X}}_{i+1}^{(p)} = f_{\text{pred}}^{(p)}\big(\mathbf{z}_i^{(p)}\big), \qquad i = 1, \ldots, N_p - 1. \tag{26}$$

The per-scale prediction loss is

$$\mathcal{L}_{\text{pred}}^{(p)} = \frac{1}{N_p - 1} \sum_{i=1}^{N_p-1} \frac{1}{pC} \left\| \widehat{\mathbf{X}}_{i+1}^{(p)} - \mathbf{X}_{i+1}^{(p)} \right\|_1. \tag{27}$$

Both losses are averaged across scales:

$$\mathcal{L}_{\text{mask}} = \frac{1}{|\mathcal{P}|} \sum_{p\in\mathcal{P}} \mathcal{L}_{\text{mask}}^{(p)}, \qquad \mathcal{L}_{\text{pred}} = \frac{1}{|\mathcal{P}|} \sum_{p\in\mathcal{P}} \mathcal{L}_{\text{pred}}^{(p)}. \tag{28}$$

B.4 Deterministic reconstruction scoring and memory construction

B.4.1 Grouped masking

To obtain deterministic reconstruction scores, the patch indices $\{1, \ldots, N_p\}$ are partitioned into n_g disjoint groups:

$$\mathcal{G}_g^{(p)} = \{i \mid (i-1) \bmod n_g = g - 1\}, \qquad g = 1, \ldots, n_g. \tag{29}$$

In the g-th forward pass, only patches in $\mathcal{G}_g^{(p)}$ are masked, ensuring that each patch receives exactly one reconstruction score

$$e_i^{(p)} = \frac{1}{pC} \left\| \widehat{\mathbf{X}}_i^{(p)} - \mathbf{X}_i^{(p)} \right\|_1. \tag{30}$$

B.4.2 Quantile purification

At each scale, the threshold τ_p is set to the $(1-r)$-quantile of the reconstruction-score distribution. Patches with

$$e_i^{(p)} > \tau_p \tag{31}$$

are discarded. Purification is applied to memory-construction samples; validation and test samples retain all patches. This step corresponds to the self-cleaning component in Table 5.

B.4.3 Coreset compression

After purification, memory entries at each scale may be further compressed in the context-representation space for efficiency. When compression is applied, approximate farthest-point selection (Sener & Savarese, 2018) or stratified random subsampling is used depending on the candidate-set size. This step reduces memory size without altering the subsequent retrieval or scoring definitions. In the main experiments, compression is disabled by default (Table 5 uses a keep ratio of 1.0).

The retained entries at scale p after purification and compression are stored as

$$\widetilde{\mathcal{M}}_p^{\text{tr}} = \left\{ \big(j, i, \mathbf{z}_{j,i}^{(p)}, \mathbf{c}_{j,i}^{(p)}\big) \right\}_{(j,i)\in\mathcal{I}_p}. \tag{32}$$

B.5 TIME-STEP AGGREGATION, SCORE CALIBRATION, AND FUSION

B.5.1 PATCH-TO-TIME EXPANSION

Within a query window, each patch-level diagnostic $u_{w,i}^{(p)}$ is expanded to the time steps covered by the corresponding patch:

$$\tilde{u}_{w,t}^{(p)} = u_{w,i}^{(p)}, \qquad t \in \{(i-1)p+1, \ldots, ip\}. \tag{33}$$

The state-novelty score a_w is broadcast to all time steps in window w. After per-scale memory aggregation, raw contributions from overlapping windows are averaged separately for each diagnostic before applying its empirical CDF, as specified in Algorithm 5.

B.5.2 MULTI-SCALE MEMORY WEIGHTING

The expanded per-scale memory diagnostics are aggregated via inverse patch-size weighting:

$$w_p = \frac{p^{-1}}{\sum_{p' \in \mathcal{P}} p'^{-1}}. \tag{34}$$

This weighting prevents coarser patches from dominating solely because they cover longer temporal spans. Reconstruction diagnostics are retained separately at each scale to preserve anomaly evidence across different temporal granularities.

B.5.3 PER-DIAGNOSTIC ECDF CALIBRATION AND MEAN FUSION

For each diagnostic type $\rho \in \Omega$, the empirical CDF is estimated in Stage B from the training diagnostics obtained with the frozen encoder and constructed normal reference memory:

$$\widehat{F}_\rho(v) = \frac{1}{N_\rho} \sum_{n=1}^{N_\rho} \mathbf{1}\left[d_n^{(\rho,\mathrm{tr})} \le v\right]. \tag{35}$$

For a new value v, $\widehat{F}_\rho(v)$ is its percentile rank among the normal training diagnostics of type ρ. Applying this monotone transformation separately to each diagnostic preserves its ranking while placing heterogeneous score distributions on a common scale. The final arithmetic mean in Eq. 11 gives weight $1/(|\mathcal{P}|+2)$ to state novelty, aggregated memory deviation, and each scale-specific reconstruction diagnostic after calibration. Thus the reconstruction scales remain separate inputs to the mean; they are not first collapsed into a single reconstruction score. No self-exclusion rule is applied during calibration.

C EXPERIMENTAL SETUP

C.1 DATASET STATISTICS

Table 4: Dataset statistics.

(a) Point-wise benchmark datasets

Dataset	Domain	Channels	Train	Test	Anom. Ratio	Val. Split	Source / Description
MSL	Spacecraft telemetry	55	58,317	73,729	10.53%	tail	NASA spacecraft telemetry
SMAP	Spacecraft telemetry	25	135,183	427,617	12.79%	tail	NASA satellite telemetry
PSM	Server monitoring	25	132,481	87,841	27.76%	interleaved	eBay application/server metrics
SMD	Server monitoring	38	708,405	708,420	4.16%	interleaved	Server machine monitoring
SWaT	Industrial control	51	495,000	449,919	12.14%	tail	Secure Water Treatment testbed

(b) Complete TEP mechanism-validation benchmark

Field	Value
Dataset	TEP
Domain	Industrial process
#Channels	53
Mode IDs	M1–M6 (normal identifier: IDV0)
Fault IDs	IDV1–IDV28 under each operating mode
Operating Modes	6
Faults per Mode	28

Table 4: Dataset statistics (continued).

Fault Test Sequences	168
Label Granularity	Sequence-level
Source / Description	Tennessee Eastman selected multimode fault data

For point-wise benchmark datasets, the anomaly ratio is computed on the test split. For TEP, memory is constructed from the IDV0 normal files of all six modes, and mechanism validation covers every combination of the six modes and 28 fault identifiers. The validation split is constructed from normal training files using a tail split.

C.2 Implementation settings

Additional baseline sources. OCSVM, PCA, IForest, HBOS, and LSTM-VAE follow Schölkopf et al. (1999), Shyu et al. (2003), Liu et al. (2008), Goldstein & Dengel (2012), and Park et al. (2018), respectively. Sources for the remaining baselines are cited in Sections 2 and 4.1.

Table 5: Common implementation settings of SCAR. Model, retrieval, memory, and calibration settings are shared across datasets; optimization settings are selected on the validation split.

Component	Setting	Value
Windowing and validation		
Windowing	Sequence length	128
Windowing	Train / memory / test stride	1 / 1 / 1
Validation	Validation ratio	0.15
Validation	Validation gap	Enabled
Representation learning		
Representation	Patch sizes	8, 32
Representation	Latent dimension	128
Representation	Slow-fast decomposition	Enabled
Representation	State-guided modulation	Enabled
Masking	Mask ratio	0.25
Masking	Number of mask groups	4
Retrieval and memory		
Retrieval	State-level candidate windows	50
Retrieval	Context-level retrieved patches	20
Retrieval	Content-nearest neighbors	5
Retrieval	State-novelty neighbors	10
Retrieval	Two-level retrieval	Enabled
Retrieval	Context-key retrieval	Enabled
Memory	Purification ratio	0.02
Memory	Coreset keep ratio	1.0
Memory	Maximum patches per scale	200,000
Nearest-neighbor search	FAISS (Johnson et al., 2019) exact-search threshold	100,000
Nearest-neighbor search	IVF nprobe	16
Scoring and calibration		
Reconstruction diagnostics	Completion head	Enabled
Reconstruction diagnostics	Self-cleaning	Enabled
Reconstruction diagnostics	Score fusion	Enabled
Final scoring	Score aggregation	CDF-mean
Calibration	Empirical CDF estimation	Training scores after memory construction
Optimization and hardware		

Continued on the next page.

Table 5: Common implementation settings (continued).

Component	Setting	Value
Training	Optimizer	AdamW
Training	Stage-A epochs	100
Training	Learning rate	Selected from {5e-4, 1e-3, 2e-3}
Training	Train batch size	Selected from {128, 256}
Training	Eval / memory / test batch size	Selected from {128, 512}
Training	Early-stopping patience	Selected from {6, 10, 15}
Training	Weight decay	1e-4
Training	Gradient clipping	1.0
Training	Scheduler	CosineAnnealingLR
Training	Minimum learning-rate ratio	0.01
Hardware	GPU	NVIDIA GeForce RTX 4090

Learning rate, batch size, and early-stopping patience are selected on the validation split. The test split is used only for final evaluation.

D Point-wise, event-aware, and range-aware evaluation

This experiment examines how SCAR's performance changes with the evaluation protocol. Table 6 groups threshold-free ranking metrics (AUROC and AP), thresholded point-wise metrics (Point-F1 and PA-F1), event-aware affiliation metrics, and range-aware VUS metrics. AUROC and AP repeat the main results for comparison; Point-F1 and PA-F1 use test-score threshold sweeps and therefore describe best-achievable performance rather than a deployed threshold-selection rule.

Table 6: Additional evaluation metrics of SCAR on five point-wise benchmark datasets. AUROC and AP correspond to the main metrics reported in the main text. Point-F1 and PA-F1 are reported as best-achievable values obtained by sweeping thresholds on the test scores, while Aff-P/Aff-R/Aff-F1 and VUS-ROC/VUS-PR provide event-aware or range-aware evaluations.

Dataset	AUROC	AP	Point-F1	PA-F1	Aff-P	Aff-R	Aff-F1	VUS-ROC	VUS-PR
MSL	0.7910	0.2690	0.3665	0.5999	0.9498	0.0797	0.1471	0.7831	0.3047
PSM	0.8015	0.5186	0.6531	0.7903	0.5314	1.0000	0.6940	0.6575	0.4779
SMAP	0.6274	0.1672	0.2674	0.6642	0.7955	0.0938	0.1678	0.6073	0.1855
SMD	0.8155	0.1823	0.2488	0.4316	0.5152	0.9989	0.6798	0.7550	0.1322
SWaT	0.8444	0.6958	0.7271	0.8255	0.5327	0.9994	0.6949	0.6568	0.4820
Avg.	0.7760	0.3666	0.4526	0.6623	0.6649	0.6344	0.4767	0.6919	0.3165

Avg. denotes the macro average across the five datasets. PA-F1 denotes point-adjusted F1. Aff-P, Aff-R, and Aff-F1 denote affiliation-based precision, recall, and F1, respectively. VUS-ROC and VUS-PR denote volume-under-surface metrics computed from ROC- and PR-based range-aware evaluations.

The protocols expose different detection behaviors. On MSL and SMAP, affiliation precision is high (0.9498 and 0.7955), but recall is low (0.0797 and 0.0938), indicating limited event coverage despite precise detected regions. PSM, SMD, and SWaT exhibit the reverse balance: affiliation recall is near one, while precision is about 0.52–0.53. Point adjustment also raises F1 substantially, especially on SMAP (0.2674 to 0.6642). These differences motivate reporting ranking, event-aware, and range-aware measures together rather than interpreting a single thresholded metric as a complete account of detection quality.

E Diagnostic evidence and score fusion

This experiment tests which evidence sources contribute to detection and whether their fusion benefits every dataset equally. Table 7 decomposes the final CDF-mean score into its constituent diagnostic signals, reporting the detection performance of each signal individually and after fusion. No single diagnostic dominates uniformly; different signals are preferred on different datasets. State novelty performs best on MSL and SMAP, completion-based diagnostics are most effective on PSM, SMD, and SWaT, and local KNN evidence is competitive on MSL. CDF-mean has the highest

AUROC on PSM, SMD, and SWaT, but state novelty has higher AUROC and AP on MSL and SMAP. Completion scores also exceed the fused AP on PSM, SMD, and SWaT. Thus the evidence supports complementary, dataset-dependent signals rather than uniform gains from fusion. CDF-mean combines them on a shared calibrated scale without choosing a different scoring branch for each test dataset.

Table 7: Performance of individual diagnostic scores and CDF-mean fusion. AR denotes AUROC and AP denotes average precision. The best result in each row is shown in red bold and the second-best in blue underline.

Dataset	Metric	CDF-Mean	Completion-8	Completion-32	KNN	State Novelty
MSL	AR	0.7910	0.5755	0.6384	0.8053	**0.8279**
	AP	0.2690	0.1606	0.1751	0.2929	**0.3087**
PSM	AR	**0.8015**	0.7929	0.7921	0.7031	0.6206
	AP	0.5186	**0.5720**	0.5280	0.4431	0.3744
SMAP	AR	0.6274	0.5412	0.4593	0.4573	**0.6438**
	AP	0.1672	0.1319	0.1217	0.1095	**0.1727**
SMD	AR	**0.8155**	0.7971	0.8142	0.7286	0.7358
	AP	0.1823	0.1806	**0.1905**	0.1465	0.1518
SWaT	AR	**0.8444**	0.8397	0.8437	0.7956	0.8100
	AP	0.6958	0.7463	**0.7468**	0.6957	0.5305

Completion-8 and Completion-32 denote the reconstruction diagnostics $d_t^{(\mathrm{rec}_8)}$ and $d_t^{(\mathrm{rec}_{32})}$. KNN denotes memory deviation $d_t^{(\mathrm{mem})}$ from the query-specific local normal reference, and State Novelty denotes $d_t^{(\mathrm{state})}$. CDF-Mean calibrates each diagnostic separately and takes their unweighted mean.

F TEP MECHANISM METRIC DEFINITIONS

This section defines the mechanism metrics used in the multi-mode TEP analysis. Let $\mathcal{A}$ denote the set of audit-normal windows and $\mathcal{F}$ the set of fault windows. Let $\mathcal{M}$ be the set of operating modes, and let

$$\mathcal{A}_m = \{i \in \mathcal{A} : y_i = m\}, \qquad \mathcal{F}_m = \{i \in \mathcal{F} : y_i = m\},$$

where y_i is the true operating-mode label of window i. We write $\mathbf{h}_i \in \mathbb{R}^{d_s}$ for the learned state representation of window i, δ_i for its memory distance, and s_i for its final anomaly score. For a global threshold quantile q, we define

$$\tau_q = \mathrm{Quantile}_q\big(\{s_i \mid i \in \mathcal{A}\}\big),$$

where the threshold is computed from all audit-normal windows.

State-mode consistency. SMC@K measures whether local neighborhoods in the learned state space align with the true operating modes. For each audit-normal window i, let $\mathrm{kNN}_K(i)$ denote its K nearest audit-normal neighbors in the learned state space, excluding itself. Then

$$\mathrm{SMC@K} = \frac{1}{|\mathcal{A}|} \sum_{i \in \mathcal{A}} \frac{1}{K} \sum_{j \in \mathrm{kNN}_K(i)} \mathbf{1}(y_j = y_i).$$

In this experiment, K is fixed because each audit-normal sample has at least K valid neighbors after self-exclusion.

State separation ratio. SFR measures between-mode separation relative to within-mode dispersion in the learned state space. For each mode m, define the mode centroid and the global centroid as

$$\boldsymbol{\mu}_m = \frac{1}{|\mathcal{A}_m|} \sum_{i \in \mathcal{A}_m} \mathbf{h}_i, \qquad \bar{\boldsymbol{\mu}} = \frac{1}{|\mathcal{A}|} \sum_{i \in \mathcal{A}} \mathbf{h}_i.$$

The state separation ratio is defined as

$$\mathrm{SFR} = \frac{\frac{1}{|\mathcal{M}|}\sum_{m\in\mathcal{M}} \|\boldsymbol{\mu}_m - \bar{\boldsymbol{\mu}}\|_2^2}{\frac{1}{|\mathcal{M}|}\sum_{m\in\mathcal{M}} \frac{1}{|\mathcal{A}_m|}\sum_{i\in\mathcal{A}_m} \|\mathbf{h}_i - \boldsymbol{\mu}_m\|_2^2}.$$

Both terms are averaged over modes so that each mode contributes equally regardless of sample count.

Same-mode retrieval ratio. SMR@K measures whether the retrieved normal references for fault windows come from the same true operating mode. For a fault window i, let $\tilde{y}_{i,r}$ denote the mode label of its r-th retrieved normal reference, and let K_i be the number of valid retrieved references used for this window. Then

$$\mathrm{SMR@K} = \frac{1}{|\mathcal{F}|}\sum_{i\in\mathcal{F}} \frac{1}{K_i}\sum_{r=1}^{K_i} \mathbf{1}(\tilde{y}_{i,r} = y_i).$$

Fault-normal separation. The fault-normal gap measures whether fault windows lie farther from their retrieved normal references than audit-normal windows do. For each operating mode m, define the mode-wise raw memory-distance gap as

$$\Delta_m = \frac{1}{|\mathcal{F}_m|}\sum_{i\in\mathcal{F}_m} \delta_i - \frac{1}{|\mathcal{A}_m|}\sum_{i\in\mathcal{A}_m} \delta_i.$$

The raw fault-normal gap is the average over modes that contain both audit-normal and fault windows:

$$\mathrm{Gap} = \frac{1}{|\mathcal{M}_\Delta|}\sum_{m\in\mathcal{M}_\Delta} \Delta_m, \qquad \mathcal{M}_\Delta = \{m \in \mathcal{M} : |\mathcal{A}_m| > 0,\ |\mathcal{F}_m| > 0\}.$$

The complete six-mode evaluation in Table 3 applies the same difference-of-means functional to the normalized memory-distance outputs and reports the resulting dimensionless normalized gap. Figure 3 visualizes the corresponding full-run raw memory distances; its dotted vertical lines mark the median distances of audit-normal and fault samples. Raw and normalized gap values are therefore interpreted within their respective evaluation scales and are not numerically comparable.

Cross-mode false-positive-rate variation. For each mode m, the false-positive rate under a shared global threshold τ_q is

$$\mathrm{FPR}_m(q) = \frac{1}{|\mathcal{A}_m|}\sum_{i\in\mathcal{A}_m} \mathbf{1}(s_i > \tau_q).$$

The reported cross-mode FPR standard deviation is computed at $q = 0.95$. Let

$$\overline{\mathrm{FPR}}(q) = \frac{1}{|\mathcal{M}|}\sum_{m\in\mathcal{M}} \mathrm{FPR}_m(q).$$

Then

$$\text{FPR-Std} = \sqrt{\frac{1}{|\mathcal{M}|}\sum_{m\in\mathcal{M}} \left(\mathrm{FPR}_m(0.95) - \overline{\mathrm{FPR}}(0.95)\right)^2}.$$

The population standard deviation is used, consistent with the implementation.

Tail calibration error. Tail calibration error measures how close the mode-wise exceedance rate is to the ideal tail probability under a shared global threshold. For $q \in \{0.95, 0.99\}$, it is defined as

$$\mathrm{TailErr}_q = \frac{1}{|\mathcal{M}|}\sum_{m\in\mathcal{M}} \left|\mathrm{FPR}_m(q) - (1-q)\right|.$$

In particular, $\mathrm{TailErr}_{0.95}$ evaluates the average deviation from the ideal 5% tail probability, while $\mathrm{TailErr}_{0.99}$ evaluates the average deviation from the ideal 1% tail probability.

Dominant evidence consistency. Dominant evidence consistency measures whether sequences of the same fault type tend to be dominated by the same anomaly-evidence channel. Let

$$\mathcal{E} = \begin{Bmatrix} \texttt{memory_distance}, \texttt{state_novelty}, \\ \texttt{completion_scale8}, \texttt{completion_scale32} \end{Bmatrix}$$

be the set of evidence channels. For fault sequence ℓ of fault type f, let $\phi_{f,\ell,c}$ denote the calibrated sequence-level evidence score of channel c. This score is obtained by aggregating window-level evidence scores within the sequence using the 95th percentile. The dominant evidence channel of sequence ℓ is

$$e_{f,\ell} = \arg\max_{c \in \mathcal{E}} \phi_{f,\ell,c}.$$

For a fault type f with L_f valid sequences, its pairwise dominant-evidence agreement is

$$\mathrm{DEC}_f = \frac{2}{L_f(L_f-1)} \sum_{1 \le a < b \le L_f} \mathbf{1}(e_{f,a} = e_{f,b}).$$

Let $\mathcal{T}_{\mathrm{fault}}$ be the set of fault types with at least two valid dominant-evidence labels. The final dominant evidence consistency is

$$\mathrm{DEC} = \frac{1}{|\mathcal{T}_{\mathrm{fault}}|} \sum_{f \in \mathcal{T}_{\mathrm{fault}}} \mathrm{DEC}_f.$$

A higher value indicates stronger within-fault-type agreement on the dominant evidence source.

G TEP MECHANISM VISUALIZATION

Figure 3 shows the same mechanisms on the complete six-mode evaluation. Panel A shows six mode-colored groups in the learned state space, with the modes forming distinct clusters or separated subregions. This agrees with full-run SMC@10 = 1.0000 and SFR = 9.65, confirming that the learned state representation recovers operating-regime structure across all six modes. Panel B shows strong diagonal dominance in the query-mode versus retrieved-reference-mode correspondence matrix. The diagonal proportions are 89%, 75%, 85%, 84%, 78%, and 75% for Modes 1–6, respectively, yielding the full-run SMR@10 of 0.8357. Off-diagonal mass is not uniform: Mode 1 references most often spill into Mode 4, while Mode 6 has its largest off-diagonal share in Mode 3. This indicates predominantly compatible retrieval with some neighboring-mode ambiguity.

Panel C shows that the full-run fault distribution is shifted rightward relative to the audit-normal distribution across the logarithmic memory- distance axis. The raw full-run gap is 6606.68. Audit-normal samples concentrate at much smaller distances because retrieved references match both the operating regime and local temporal pattern, whereas fault samples retain larger discrepancies even when references come from the same regime. The full-run plot therefore provides direct visual support for the normalized fault-normal separation reported in Table 3.

Panel D displays all six mode-wise exceedance curves. The curves follow the theoretical $1 - q$ trend unevenly across modes: Mode 4 is the most elevated, while Modes 2 and 3 remain close to zero in the extreme tail. Across the full run, the resulting FPR standard deviation is 0.0731 and the 95% and 99% tail errors are 0.0572 and 0.0135. Thus calibration is broadly stable across modes, while the visualization also makes the remaining mode-specific tail variation visible.

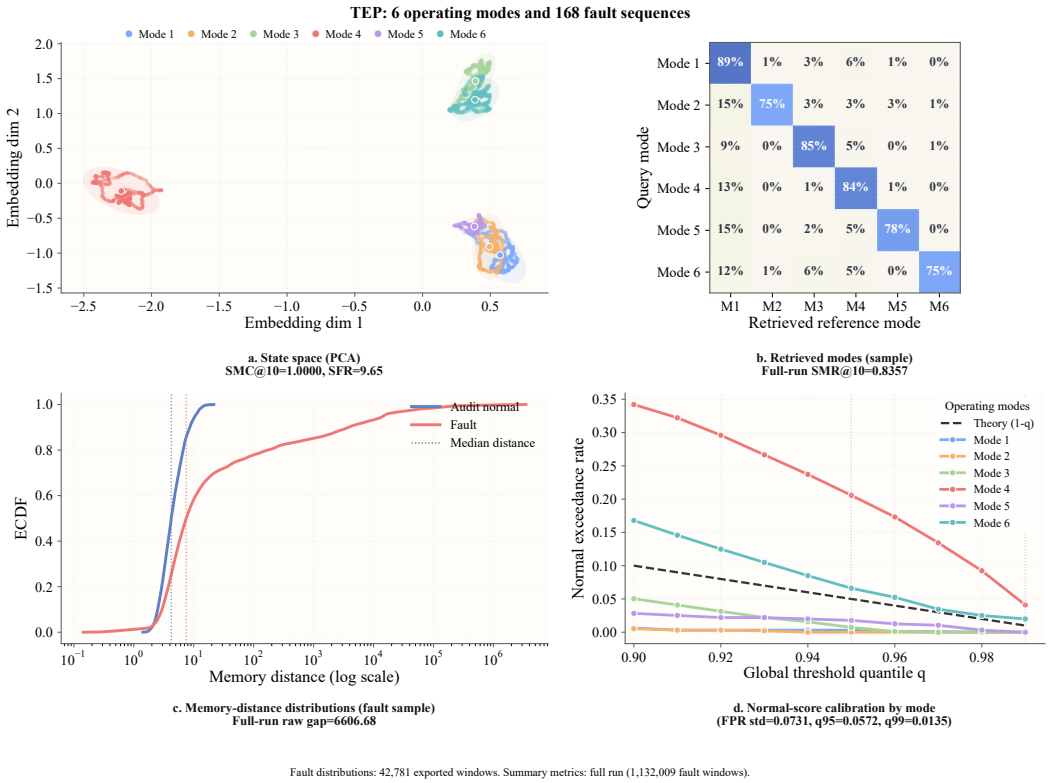


Figure 3: Mechanism visualization on the complete six-mode TEP evaluation (168 fault sequences). (A) PCA projection of audit-normal state representations, colored by true mode. (B) Query-mode versus retrieved-reference-mode proportions. (C) Raw memory-distance ECDFs for audit-normal and fault samples; dotted lines mark the medians. (D) Mode-wise normal-score exceedance versus global threshold quantile q, with the reference trend $1 - q$ shown dashed.

H GENERALIZATION TO THE TSB-AD BENCHMARK

We evaluate SCAR on the formal test collection of TSB-AD (Liu & Paparrizos, 2024), comprising 350 univariate and 180 multivariate series. Table 8 reports averages across each collection and across all 530 series. The primary CDF-mean variant from the main paper remains strong across ranking, thresholded, event-aware, and range-aware metrics. We also report SCAR-Z, a supplementary fusion variant that standardizes each diagnostic using fixed training statistics before aggregation. It improves AP, Point-F1, and VUS-PR, while CDF-mean remains the method used for the main five-dataset results.

Table 8: TSB-AD results in percent. U and M denote the univariate and multivariate formal evaluation collections. P-F1 is point-wise F1; A-P/A-R/A-F1 are affiliation metrics.

Fusion	Split	N	AUROC	AP	P-F1	PA-F1	A-P	A-R	A-F1	VUS-ROC	VUS-PR
CDF-mean	U	350	86.12	33.55	40.32	59.56	76.81	93.28	82.09	88.52	39.78
	M	180	76.18	27.32	33.44	55.51	71.19	93.90	78.89	78.92	32.18
	U+M	530	82.74	31.43	37.98	58.19	74.90	93.49	81.01	85.26	37.20
SCAR-Z	U	350	86.66	44.79	49.40	62.80	81.85	87.91	81.21	88.39	48.28
	M	180	76.80	37.84	43.59	57.05	77.81	86.46	77.77	77.96	40.55
	U+M	530	83.31	42.43	47.43	60.84	80.48	87.42	80.04	84.85	45.65

Across all 530 series, SCAR-Z raises AP from 31.43% to 42.43% and VUS-PR from 37.20% to 45.65%, but lowers affiliation recall from 93.49% to 87.42% and slightly lowers VUS-ROC. Thus the

fusion choice changes the precision–recall balance rather than improving every metric. Both variants perform better on the univariate collection than on the multivariate collection in AUROC, AP, and VUS-PR.

In the TSB-AD leaderboard snapshot dated July 1, 2026, SCAR-Z obtains 48.28% VUS-PR on the univariate collection (tied fifth overall and first among non-pretrained methods) and 40.55% on the multivariate collection (second overall and second among non-pretrained methods). These results extend the evidence beyond the five fixed multivariate benchmarks and show that condition-aware local references remain useful across diverse series collections.

I Retrieval behavior without regime labels

Ground-truth operating regimes are unavailable for the five point-wise benchmarks. We therefore examine retrieval behavior with three label-free diagnostics in Table 9. The state proxy consists of channel means, local trends, and low-frequency energies; each feature is standardized by the normal training memory, and proxy distance is one minus correlation. Temporal Jaccard@1 is the Jaccard overlap between the top-20 reference sets of adjacent queries. Reference-set Jaccard measures the overlap between an ablated retrieval set and the full two-level retrieval set. These quantities diagnose retrieval geometry without using anomaly labels. They are proxies rather than estimates of ground-truth regime accuracy.

Table 9: Label-free diagnostics of the retrieved normal references. State-proxy distance is lower when the retrieved references better match the query's coarse operating state. Jaccard values are percentages. "−State" and "−Context" remove the corresponding retrieval level. Reference-set overlap is measured against the full retrieval result.

	State-proxy distance ↓			Temporal Jaccard@1 ↑			Reference-set Jaccard	
Dataset	Full	−State	−Context	Full	−State	−Context	−State	−Context
MSL	0.703	0.711	0.707	48.8	25.3	50.4	5.1	38.4
PSM	0.736	0.622	0.739	92.0	87.8	89.0	1.6	50.7
SMAP	0.389	0.436	0.393	64.9	49.0	64.1	15.2	43.0
SMD	0.300	0.335	0.301	85.5	79.0	80.8	12.0	65.2
SWaT	0.704	0.720	0.703	91.6	96.8	91.2	0.8	81.5

Full retrieval gives the lowest state-proxy distance on MSL, SMAP, and SMD, whereas removing state filtering lowers this distance on PSM and removing context gives a marginally lower value on SWaT. The handcrafted proxy therefore does not uniformly favor the full model. Temporal overlap is highest for full retrieval on PSM, SMAP, and SMD, while MSL and SWaT favor different ablations. The very low overlap after removing state filtering, especially on PSM and SWaT, shows that the first retrieval level materially changes the candidate neighborhood rather than merely reranking the same references. Context removal retains more of the full set but still replaces a substantial fraction of references. Together with the detection ablation in Table 2, these diagnostics show that the two levels make distinct contributions to the local normal reference.

J Memory contamination, purification, and retention

SCAR assumes that the training memory is predominantly normal. We examine three aspects of this assumption: detection under contamination, sensitivity to purification strength, and retention of rare normal samples versus contaminating anomalies.

J.1 Detection under memory contamination

We test sensitivity by introducing increasing contamination into the candidate memory. Table 10 reports controlled runs in percentage units. Average AUROC changes by less than 1.2 points through 10% contamination, whereas AP is more sensitive, especially on SWaT. The difference is consistent with AP placing greater emphasis on the highest-ranked samples.

Table 10: Detection performance under memory contamination. Entries are AUROC/AP in percent; Avg. is the macro average over datasets.

Dataset	0%	1%	5%	10%
MSL	76.25/19.60	75.87/19.47	76.11/19.66	75.97/19.42
PSM	75.54/23.44	75.36/23.29	76.98/24.56	77.79/25.33
SMAP	65.76/10.31	66.51/10.44	66.53/10.24	66.44/10.55
SMD	80.16/7.26	79.64/7.11	79.38/7.30	79.76/8.09
SWaT	89.03/80.98	87.08/51.10	81.84/26.64	84.60/31.89
Avg.	77.35/28.32	76.89/22.28	76.17/17.68	76.91/19.06

J.2 Sensitivity to purification strength

Table 11 varies the amount removed by quantile purification while holding memory contamination at 10%. No purification gives the highest average AUROC/AP (77.91/25.32), compared with 76.91/19.06 at 2% removal and 76.11/19.75 at 10%. The effect is dataset-dependent: 2% removal is nearly neutral on MSL and SMAP and slightly improves PSM, but reduces AP on SMD and SWaT. More aggressive removal is not uniformly beneficial either. These results delimit purification's role: removing high-error candidates does not by itself guarantee better detection.

Table 11: Purification under 10% memory contamination. Columns denote the fraction removed from the candidate memory; entries are AUROC/AP in percent.

Dataset	None	2%	10%
MSL	76.09/19.55	75.97/19.42	72.99/17.09
PSM	77.70/25.21	77.79/25.33	75.85/23.67
SMAP	66.50/10.56	66.44/10.55	66.51/10.58
SMD	80.04/11.76	79.76/8.09	79.18/5.81
SWaT	89.21/59.53	84.60/31.89	86.01/41.62
Avg.	77.91/25.32	76.91/19.06	76.11/19.75

J.3 Rare-normal retention and anomaly survival

Tables 12 and 13 examine what remains in the memory. Average rare-normal retention is 94.65% under mild stress and 81.07% under strong stress, showing that stronger filtering also removes some legitimate normal diversity. Purification retains 99.55% of low-error anomalies on average, versus 40.87% of high-error anomalies. After final memory construction, these survival rates fall to 13.43% and 8.45%, respectively. The contrast shows why a reconstruction-error filter cannot remove all contamination: anomalous candidates with low error can pass the filter. Retention and survival must therefore be interpreted together with detection accuracy.

Table 12: Rare-normal retention under mild and strong stress settings. All values are percentages.

Dataset	Mild	Strong
MSL	95.88	82.93
PSM	93.57	78.61
SMAP	96.76	87.44
SMD	93.15	79.21
SWaT	93.87	77.16
Avg.	94.65	81.07

Table 13: Survival of contaminating anomalies after purification and after final memory construction. Low- and high-error groups are reported separately; all values are percentages.

	Low-error anomalies		High-error anomalies	
Dataset	Purified	Final memory	Purified	Final memory
MSL	100.00	33.03	81.67	26.11
PSM	97.74	13.81	6.44	1.73
SMAP	100.00	13.63	95.59	13.46
SMD	100.00	2.84	0.00	0.00
SWaT	100.00	3.84	20.63	0.97
Avg.	99.55	13.43	40.87	8.45

K SENSITIVITY ANALYSIS

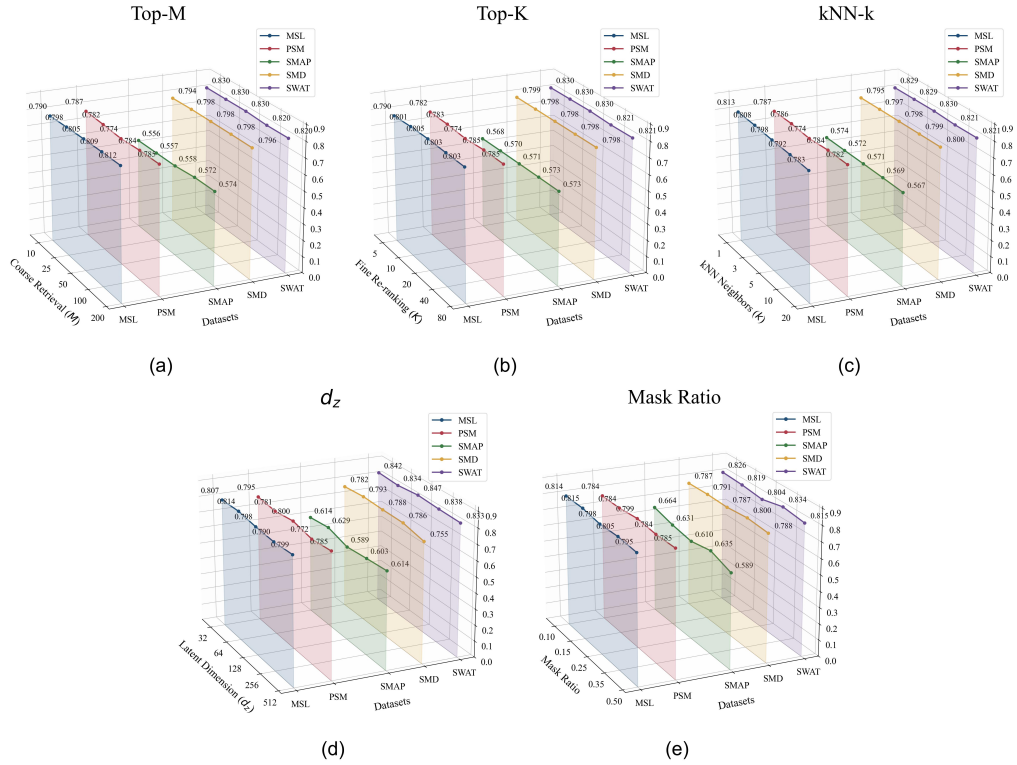


Figure 4: Sensitivity analysis of SCAR on five benchmark datasets. Panels (a)–(e) vary Top-M (state-level candidate windows), Top-K (context-level retrieved patches), kNN-k (content-nearest neighbors), d_z (latent dimension), and mask ratio, respectively. The vertical axis reports AUROC.

Table 14: Hyperparameter search ranges for sensitivity analysis. "Reuse Stage-A" means that the Stage-A encoder is kept fixed while retrieval- or memory-related settings are varied. "Full training" means that SCAR is retrained under the corresponding hyperparameter setting.

Hyperparameter	Candidate values	Default	Run mode
State-level candidate windows	{10, 25, 50, 100, 200}	50	Reuse Stage-A
Context-level retrieved patches	{5, 10, 20, 40, 80}	20	Reuse Stage-A
Content-nearest neighbors	{1, 3, 5, 10, 20}	5	Reuse Stage-A
Latent dimension	{32, 64, 128, 256, 512}	128	Full training
Mask ratio	{0.10, 0.15, 0.25, 0.35, 0.50}	0.25	Full training

K.1 SCALE CONTROLS

The sensitivity curves vary one hyperparameter around the default. We additionally run paired controls that change the input or patch scale for both full condition-aware retrieval and a global-retrieval counterpart. Table 15 compares look-back lengths $L \in \{128, 512\}$. Table 16 fixes the patch size to $p = 64$. These are controlled paired runs with the remaining settings held fixed; they are not restatements of the independently selected main-table configurations.

Table 15: Look-back scale control. AUROC/AP in percent.

	$L = 128$		$L = 512$	
Dataset	Full	Global	Full	Global
MSL	71.54/23.31	67.59/20.47	72.46/24.52	67.82/20.25
PSM	80.01/58.78	80.01/58.78	77.81/51.40	77.81/51.40
SMAP	62.11/19.00	62.18/19.85	69.49/21.05	68.13/21.00
SMD	81.24/20.12	81.27/20.12	80.82/18.44	80.84/18.43
SWaT	83.40/72.85	78.11/33.52	84.49/74.66	79.08/55.60
Avg.	75.66/38.81	73.83/30.55	77.02/38.01	74.74/33.33

Table 16: Patch control ($p = 64$). AUROC/AP in percent.

Dataset	Full	Global
MSL	81.47/33.58	81.70/33.40
PSM	77.84/49.20	77.85/49.21
SMAP	49.85/15.59	49.21/15.30
SMD	81.58/17.05	80.41/15.44
SWaT	79.50/70.03	43.21/12.14
Avg.	74.05/37.09	66.48/25.10

The full model retains its clearest advantage on the heterogeneous SWaT benchmark under both controls, while the two variants are nearly tied on PSM and SMD. The average gains at both look-back lengths and at $p = 64$ indicate that the benefit of condition-aware retrieval is not tied to a single temporal scale.

L COMPUTATIONAL EFFICIENCY AND RESOURCE USE

Table 17 reports end-to-end training and inference measurements on the RTX 4090 system described in Appendix C. Model size varies slightly with the number of input channels. Training time grows with sequence length, while inference remains below 0.6 ms per point on every dataset.

Table 17: SCAR resource use by dataset. Memory columns report peak allocated memory.

	Training					Inference			
Dataset	Points	Params.	Min.	GPU GiB	CPU GiB	ms/point	point/s	Total s	GPU GiB
MSL	73,729	1.67M	17.29	0.89	16.80	0.587	1,704	43.27	2.60
PSM	87,841	1.27M	24.56	0.86	16.64	0.563	1,778	49.42	2.65
SMAP	427,617	1.27M	50.59	0.86	17.29	0.456	2,195	194.84	2.65
SMD	708,420	1.44M	67.00	0.90	19.81	0.532	1,880	376.66	2.68
SWaT	449,919	1.62M	59.32	0.89	20.89	0.322	3,108	144.78	2.73

Table 18: Efficiency comparison under the same measurement protocol. Training time gives the cross-dataset mean followed by the range in parentheses.

Method	Parameters	Train min.	Train GPU GiB	ms/point	point/s	Infer. GPU GiB
SCAR	1.27–1.67M	43.75 (17.29–67.00)	0.86–0.90	0.4630	2,160	∼2.67
PaAno	0.39–0.42M	176.29 (1.37–441.99)	2.08–2.12	0.0365	27,364	∼2.12
PUAD	13.05–19.24M	18.28 (3.62–39.23)	2.63–2.83	0.0472	21,197	∼2.83
PGRF-Net	1.28–1.56M	94.53 (10.19–241.92)	0.94–0.98	0.0508	19,683	∼0.98

SCAR uses fewer parameters than PUAD and less training GPU memory than all three retrieval-oriented comparators, but its two-level memory search makes inference slower. The cost is therefore concentrated at query time, matching the method's use of a query-specific rather than globally shared normal reference.

M ADDITIONAL QUALITATIVE CASE STUDIES

This section presents qualitative case studies on all five benchmark datasets, illustrating how SCAR detects representative anomalous events and decomposes the final anomaly score into its evidence sources. Each panel displays the selected multivariate signals, the final anomaly score, the contribution and dynamic fusion weight of each scoring branch, and the relation heatmaps computed from the base and event regions. Representative windows are selected by clustering all test windows. The characterizations below reflect observed score patterns rather than ground-truth anomaly-type labels.

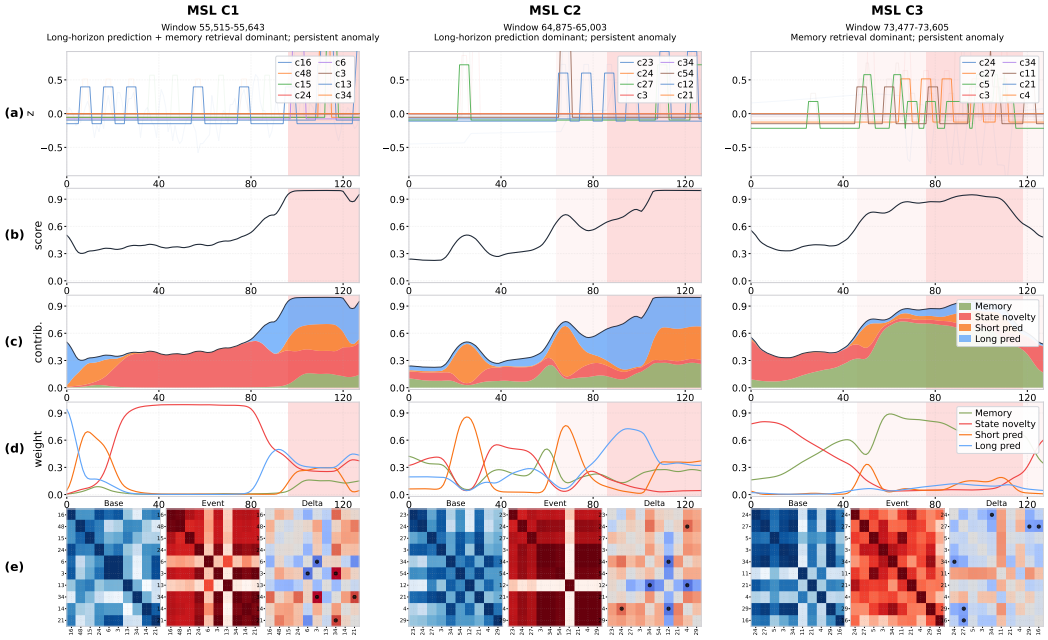


Figure 5: Qualitative case studies on MSL.

MSL. Figure 5 presents three representative anomalous cases from MSL. All three cases exhibit persistent anomaly responses: the scores remain elevated within or near the labeled event intervals and show clear temporal correspondence with the highlighted regions. This suggests that SCAR captures sustained abnormal evidence at the event level. Specifically, `MSL C1` is jointly driven by the long-horizon prediction branch and the memory retrieval branch, consistent with the event deviating from both long-term temporal evolution and the normal memory references. `MSL C2` is dominated by long-horizon prediction evidence, consistent with a sustained temporal prediction mismatch. In contrast, `MSL C3` is primarily driven by the memory retrieval branch, suggesting that although the local trajectory shows no abrupt short-term disruption, its overall pattern deviates from the normal memory manifold. The relation heatmaps further suggest that, compared with cases explicitly characterized by cross-channel relation shifts, these MSL cases are better interpreted as persistent temporal-pattern deviations, with relation changes appearing more localized rather than globally reorganized.

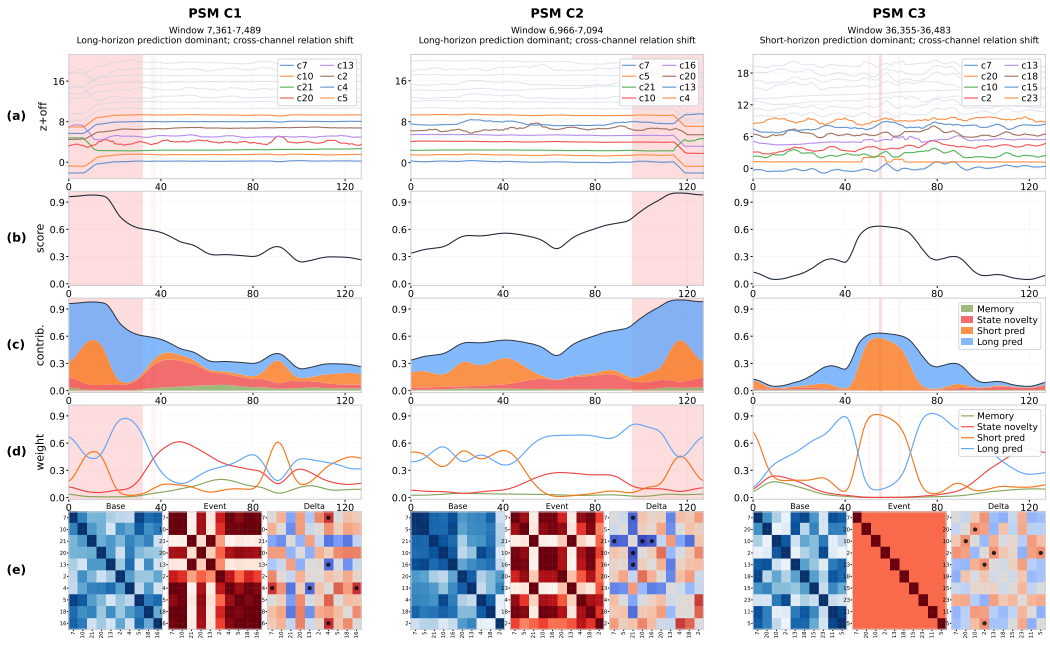


Figure 6: Qualitative case studies on PSM.

PSM. Figure 6 illustrates three representative cases from PSM, showing that SCAR distinguishes among different sources of anomalous evidence. Both `PSM C1` and `PSM C2` are dominated by the long-horizon prediction branch and are accompanied by visible cross-channel relation changes. This is consistent with the evidence arising not from isolated point-wise fluctuations but from multivariate mismatch over a longer temporal scale. In `PSM C1`, the anomaly score remains high during the early labeled event interval and gradually decreases afterwards, whereas in `PSM C2`, the score increases substantially in the later event interval. Both cases show event-level responses broadly aligned with the annotated regions. By contrast, `PSM C3` is driven by the short-horizon prediction branch, and its anomaly response is more localized and concentrated, suggesting a rapid deviation at a shorter temporal scale. The PSM cases demonstrate that SCAR not only detects anomalous events but also differentiates whether the dominant evidence reflects long-range temporal mismatch or short-term disturbance. The relation heatmaps further suggest that these events are often accompanied by changes in cross-channel dependencies.

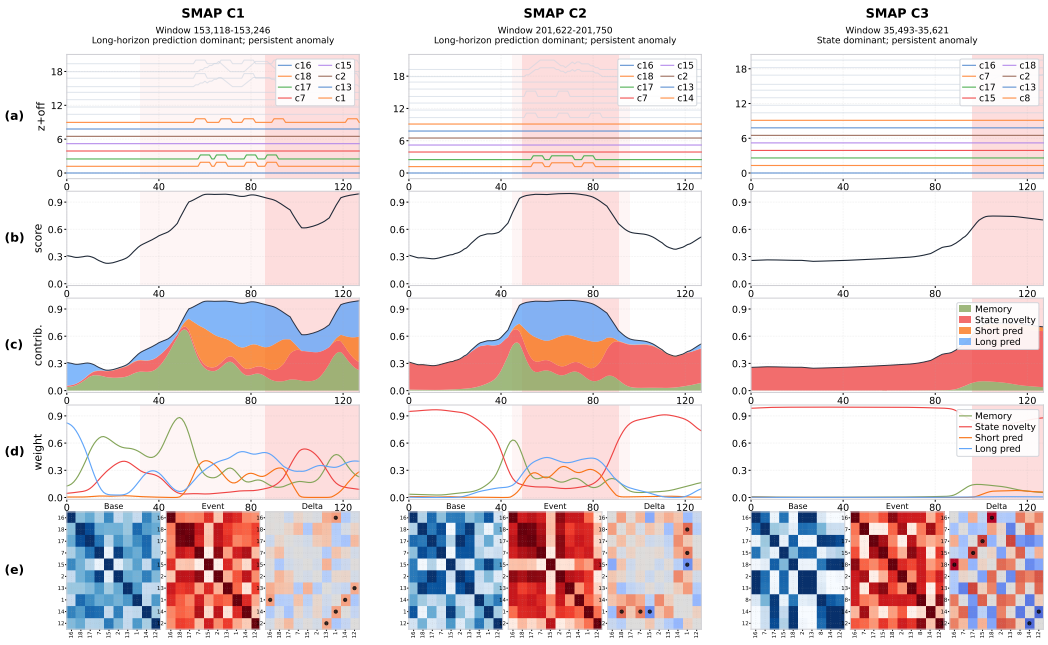


Figure 7: Qualitative case studies on SMAP.

SMAP. Figure 7 shows three SMAP cases that all exhibit persistent anomaly characteristics, while their dominant evidence sources differ. This suggests that SCAR provides fine-grained mechanistic explanations even within a single dataset. `SMAP C1` and `SMAP C2` are dominated by the long-horizon prediction branch, and their anomaly scores form relatively stable plateaus around the event regions. This is consistent with identification primarily through long-term temporal prediction mismatch. In comparison, `SMAP C3` is driven by the state novelty branch, suggesting that the anomalous evidence does not primarily arise from a sudden increase in prediction error but from an inconsistency between the current state representation and the normal state references. Together, these cases show that SCAR does not rely on a single prediction-error signal but distinguishes long-term temporal mismatch from state-level novelty as separate evidence sources.

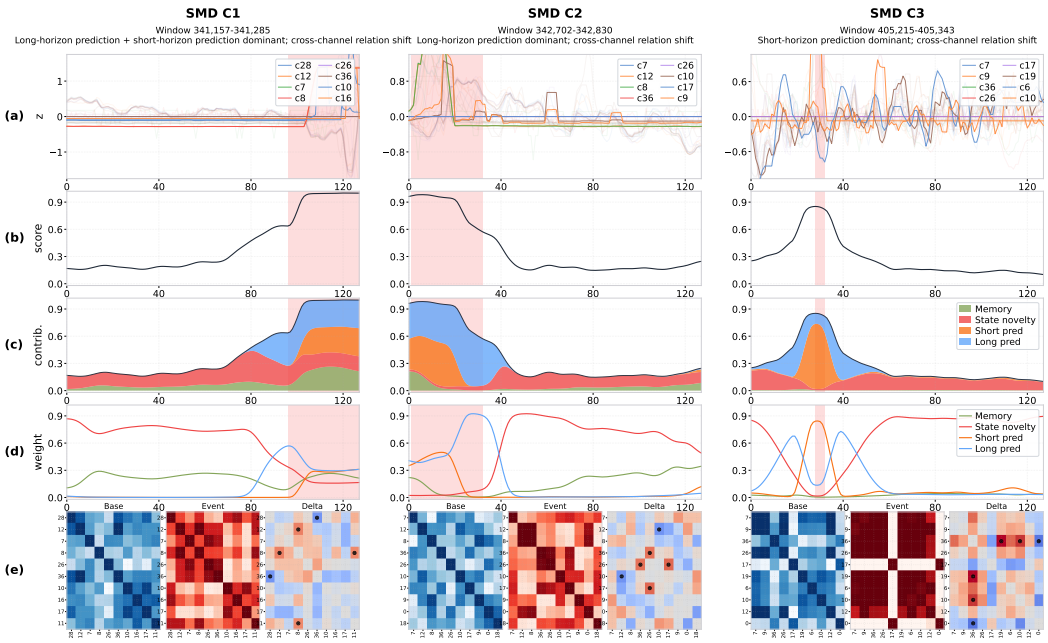


Figure 8: Qualitative case studies on SMD.

SMD. Figure 8 presents three representative anomalous cases from the server monitoring dataset SMD. The anomaly scores align clearly with the highlighted event intervals, and the relation heatmaps reveal visible differences between base and event regions. Specifically, `SMD C1` is jointly dominated by the long-horizon and short-horizon prediction branches, consistent with the event disrupting normal behavior at both coarse and fine temporal scales. `SMD C2` is driven by the long-horizon prediction branch, consistent with a longer-scale temporal context mismatch. In contrast, `SMD C3` is dominated by the short-horizon prediction branch and shows a more localized and concentrated anomaly response. Although the three cases differ in their dominant evidence sources, all exhibit cross-channel relation changes to varying degrees. This suggests that anomalies in SMD are not always isolated single-variable deviations but may also involve changes in inter-channel interaction structure. These cases illustrate SCAR's ability to capture compound anomalous evidence involving both multi-scale prediction mismatch and channel-relation changes.

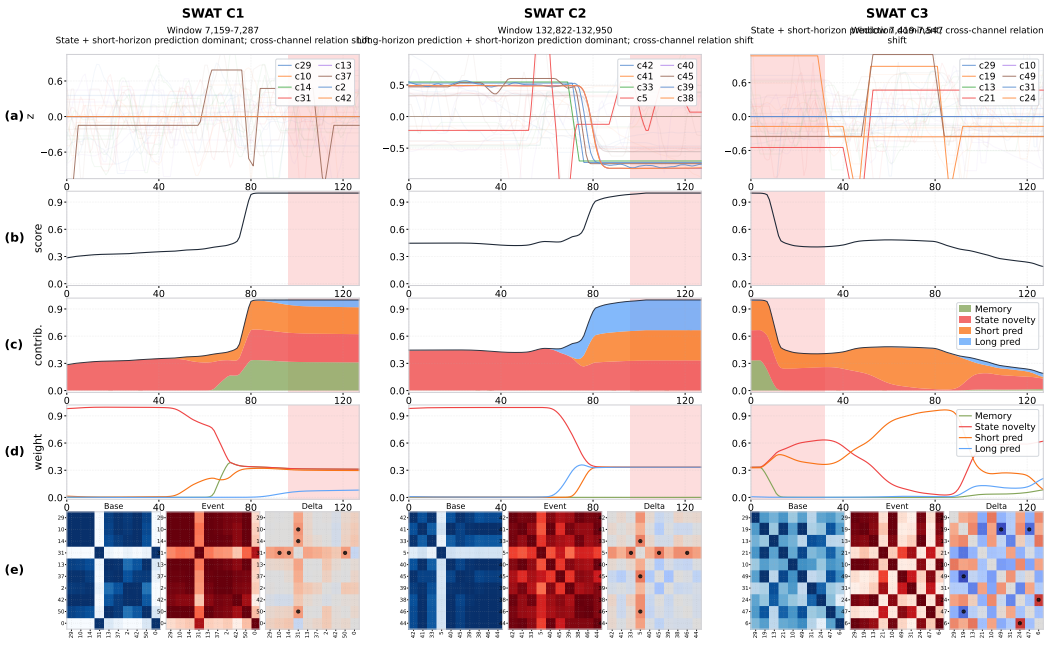


Figure 9: Qualitative case studies on SWaT.

SWaT. Figure 9 highlights the compound nature of anomalous events in the industrial control system SWaT. Both `SWAT C1` and `SWAT C3` are driven by the state novelty and short-horizon prediction branches, with relation heatmaps showing visible cross-channel structural changes between base and event regions. This is consistent with these events involving rapid local temporal deviations, state-level inconsistency, and shifts in multivariate coupling. `SWAT C2` is jointly dominated by the long-horizon and short-horizon prediction branches and also shows cross-channel relation changes, suggesting that the event combines short-term disturbances, longer-scale temporal pattern shifts, and altered channel dependencies. Overall, the SWaT cases demonstrate that SCAR can produce stable responses to anomalous events in industrial control systems while further distinguishing whether the dominant evidence arises from state novelty, short-term prediction mismatch, or multi-scale temporal mismatch.