LEARN-TS: LLM-ENHANCED ALIGNMENT AND RECONSTRUCTION WITH NORMALITY GUIDANCE FOR MULTIVARIATE TIME-SERIES ANOMALY DETECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Reconstruction errors in multivariate time-series anomaly detection may not reliably distinguish abnormal behavior from benign deviations. Language-derived semantics offer complementary context, but existing multimodal approaches may rely on time associated paired textual information that is difficult to obtain consistently and is not provided by standard multivariate time-series anomaly detection benchmarks. This setting poses two challenges: (1) conditioning masked reconstruction on window-specific semantics without exposing exact numerical targets or anomaly specific cues, and (2) using a window-independent concept of normality as a complementary semantic reference rather than an independent anomaly detector. We propose LLM-Enhanced Alignment and Reconstruction with Normality Guidance for Time Series (LEARN-TS), which uses a frozen language model to construct two role separated semantic representations without requiring temporally paired external text. Window-specific observation semantics encode temporal and cross-variable context without exact numerical values to guide channel-shared patch-masked reconstruction. A fixed, dataset-agnostic normality prompt provides a window-independent semantic reference for aligning normal representations and estimating normality discrepancy. At inference, masking each temporal patch once yields timestamp level reconstruction evidence, conditionally modulated by discrepancy from a separate unmasked view. Across four benchmarks, LEARN-TS achieves the highest mean performance in 13 of 16 dataset–metric comparisons. Controlled ablations examine observation conditioning, joint normality alignment and scoring, and reference content, showing dataset-dependent ranking benefits and modest average gains from semantic over random references.

1 INTRODUCTION

Multivariate time-series anomaly detection (MTSAD) is essential for monitoring industrial processes, computing infrastructures, application services, and spacecraft systems. Because anomalous events are rare and costly to annotate, MTSAD is commonly formulated under unsupervised or one-class settings, where normal operation patterns are learned and deviations are identified from reconstruction, prediction, or representation based evidence. A fundamental difficulty, however, is that numerical deviation alone does not fully characterize whether an observation is behaviorally abnormal. Expressive models may reconstruct anomalous inputs accurately, while benign regime changes or distribution shifts may produce large residuals. Consequently, anomaly evidence derived only from numerical discrepancies can remain ambiguous without context about what constitutes coherent normal behavior, motivating the use of complementary semantic information.

Recent language model and multimodal time-series methods have begun to introduce such semantic information into time-series analysis. In multimodal anomaly detection, semantic context may be provided through textual descriptions, reports, logs, or other information associated with particular time periods or events. This form of *time associated paired textual context* can provide rich behavioral cues, but it is difficult to assume in many industrial monitoring settings. Operational logs and reports are often sparse, irregularly recorded, written at a different temporal granularity from sensor streams, or unavailable for large portions of the monitored timeline. Moreover, standard MTSAD

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.

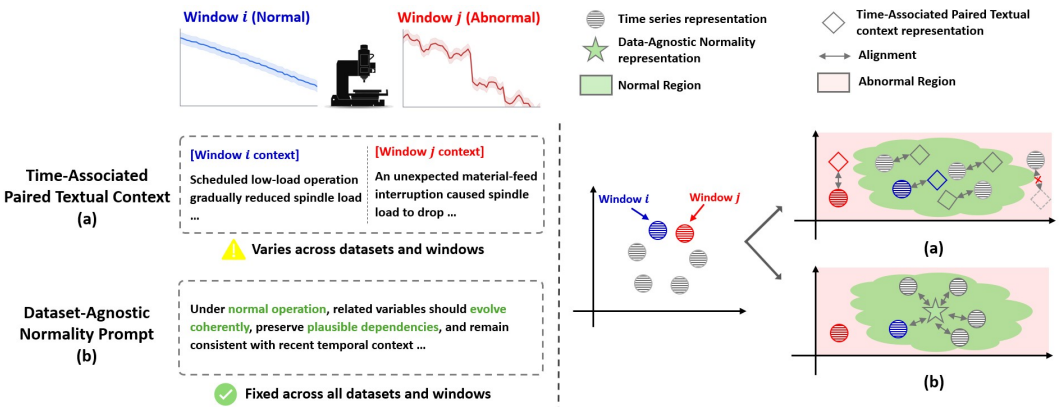

Figure 1: (a) Time-associated paired external textual context may vary across datasets and monitored intervals, providing context-dependent semantic information. (b) LEARN-TS instead derives a shared, window-independent normality reference from a fixed, dataset-agnostic prompt applied unchanged across all datasets and windows.

benchmarks such as SWaT, SMD, PSM, and MSL do not provide paired textual observations for individual windows. Requiring such context therefore changes the information available to the detector and limits applicability when comprehensive temporal pairing cannot be obtained. This raises a practical question: **can language-derived semantics provide useful guidance without requiring temporally paired external text?**

We address this question by assigning distinct roles to two complementary semantic representations. First, *window-specific observation semantics* summarize coarse temporal and cross-variable characteristics of each input window without including exact numerical values or explicit anomaly labels. Second, a *window-independent normality reference* is derived from a fixed, dataset-agnostic prompt shared across all datasets and windows. As illustrated in Figure 1, panel (a) assumes time-associated paired external text, whereas panel (b) uses the fixed normality prompt without requiring interval-specific external textual context.

Building on this formulation, we propose **LLM-Enhanced Alignment and Reconstruction with Normality Guidance for Time Series (LEARN-TS)**. It uses a frozen language model to construct two complementary semantic representations: window-specific observation semantics that guide numerical reconstruction, and a shared semantic normality reference that supports normality-aware representation learning and anomaly scoring. Across SWaT, SMD, PSM, and MSL, LEARN-TS achieves the best result in 13 of 16 dataset–metric comparisons, demonstrating strong performance across both ranking-based and event-level evaluation criteria. Controlled analyses examine cross-modal fusion, observation–window correspondence, the roles of normality alignment and scoring, and the value of structured normality semantics relative to a random semantic-free reference.

Our contributions are summarized as follows:

- We introduce a semantic guidance framework for MTSAD that assigns distinct roles to window-specific observation descriptions and a shared normality prompt, without requiring temporally paired external text.
- We develop observation-conditioned patch-masked reconstruction and normality-guided representation alignment, combining exhaustive patch-wise reconstruction evidence with discrepancy-based score modulation from a separate unmasked view.
- Through five-seed benchmark evaluations and controlled ablations on four datasets, we examine observation conditioning, joint alignment and scoring, and reference content. The results show dataset-dependent ranking benefits and modest average gains from semantic over random references.

2 Related Work

2.1 Reconstruction- and Normality-Based Time-Series Anomaly Detection

Unsupervised MTSAD typically models normal behavior through reconstruction or prediction. Representative methods use stochastic latent dynamics (Su et al., 2019), adversarially trained autoencoders (Audibert et al., 2020), or association discrepancy alongside reconstruction evidence (Xu et al., 2022). Recent work addresses distribution drift, reconstruction over generalization, and cross-dataset generalization (Wang et al., 2023; Sun et al., 2025; Shentu et al., 2025), while memory-based and one-class methods encode normal patterns using memory items, prototypes, latent centers, or hierarchical hyperspheres (Song et al., 2023; Li et al., 2023; Ruff et al., 2018; Shen et al., 2020). These methods derive normality primarily from numerical data; LEARN-TS adds a language-derived semantic normality reference.

2.2 Language Models for Time-Series Analysis

Pretrained language models have been adapted as general time-series backbones in GPT4TS (Zhou et al., 2023), through numerical patch reprogramming in Time-LLM (Jin et al., 2024), and with multiscale language-derived context in LLM-Mixer (Kowsher et al., 2025). However, removing or replacing the language model may preserve forecasting performance in some settings (Tan et al., 2024). LEARN-TS instead uses a frozen language model to encode explicitly constructed semantic descriptions rather than primarily as a numerical backbone.

2.3 Multimodal and Semantic Alignment for Time-Series Anomaly Detection

Multimodal time-series methods augment numerical observations with textual or other semantic information and learn correspondences across modalities. CALF applies cross-modal alignment and fine-tuning to time-series forecasting (Liu et al., 2025). For anomaly detection, MindTS combines endogenous time-series descriptions with externally collected background text through fine-grained semantic alignment, condensed interaction, and cross-modal reconstruction (Hu et al., 2026). MindTS uses external background text and supports loose or sparse window-level matching rather than requiring strict temporal alignment.

LEARN-TS does not require a corpus of external text associated with monitored intervals. Instead of associating external textual context with individual windows, LEARN-TS derives observation semantics from each numerical input and uses a window-independent normality reference constructed from a fixed, dataset-agnostic prompt. The two semantic representations are assigned distinct roles in reconstruction conditioning, normality alignment, and anomaly scoring.

3 Methodology

Let $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_T) \in \mathbb{R}^{T \times C}$ denote a multivariate input sequence with T timestamps and C variables, where $\mathbf{x}_t \in \mathbb{R}^C$. LEARN-TS operates on N sliding windows $\mathbf{X}_i \in \mathbb{R}^{L \times C}$, $i \in \{1, \ldots, N\}$, where L denotes the window length and N the total number of windows extracted from the input sequence. At inference, it produces a timestamp-level anomaly score S_t and classifies timestamp t as anomalous when $\widehat{y}_t = \mathbb{I}[S_t > \tau]$, where τ is selected according to the evaluation protocol in Section 4.1.

3.1 Overall Framework

As illustrated in Figure 2, LEARN-TS comprises three stages. First, a frozen language model encodes a window-specific observation prompt and a fixed, dataset-agnostic normality prompt shared unchanged across all datasets and windows. Second, the observation representation conditions channel shared patch masked reconstruction through gated cross-modal fusion, while fused representations of normal training windows are aligned with the normality representation. Finally, inference combines reconstruction evidence obtained by masking each temporal patch once with normality discrepancy computed from a separate unmasked pass. The calibrated discrepancy conditionally modulates reconstruction evidence to produce timestamp-level anomaly scores.

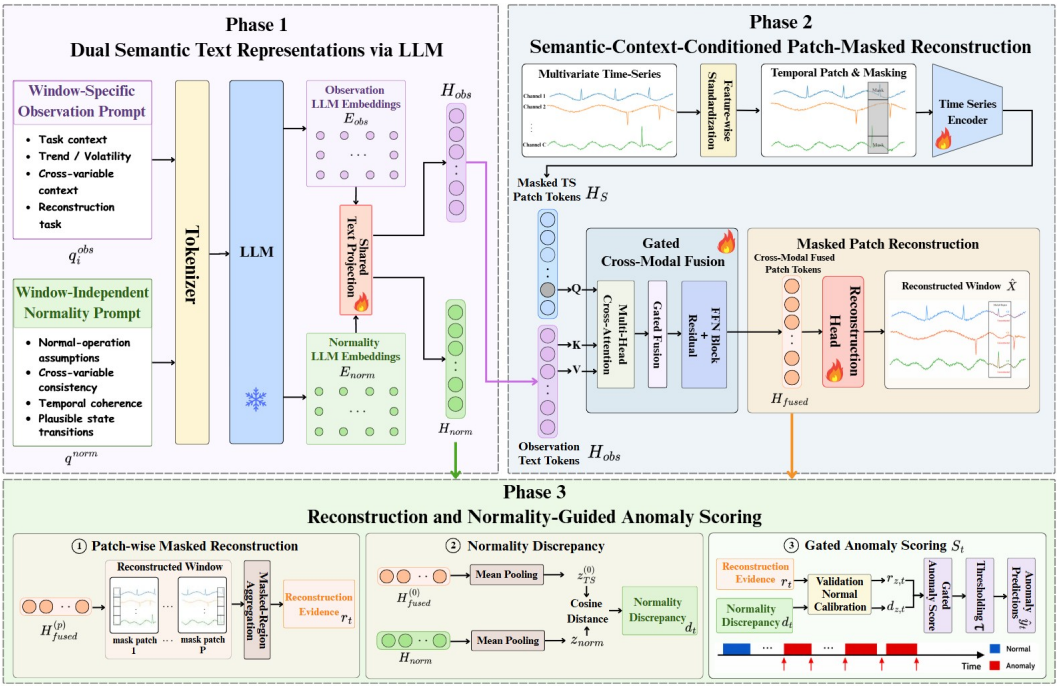

Figure 2: Overview of the three-phase LEARN-TS framework.

3.2 Dual Semantic Text Representations via LLM

Window-specific observation representation. For each window $\mathbf{X}_i$, the observation prompt q_i^{obs} combines deterministic descriptors of the fully observed window with fixed dataset-level system descriptions, sensor groups, and qualitative operational rules. Descriptors summarize trends, volatility, and temporal and cross-variable relationships without exposing exact measurements, timestamps, change-point locations, local deviation magnitudes, anomaly labels, or explicit anomaly judgments. Only the normality prompt is dataset-agnostic. Construction details appear in Appendix F.2.

Window-independent normality representation. To construct a window-independent semantic reference, we define a fixed, dataset-agnostic normality prompt q^{norm} describing broad properties of normal multivariate temporal behavior, including temporal coherence, consistent changes among related variables, plausible state transitions, and context supported recovery. The prompt contains no dataset-specific sensor identities, anomaly labels, or test related information and is shared unchanged across all datasets and input windows. It therefore induces a window-independent semantic notion of normality rather than context associated with a particular observed interval.

Frozen text encoding and shared projection. A frozen language model E_{text} encodes both prompts, and a shared trainable projection $\phi(\cdot)$ maps their token embeddings to the time-series model dimension:

$$\mathbf{H}_{\text{obs},i} = \phi\big(E_{\text{text}}(q_i^{\text{obs}})\big)\,, \qquad \mathbf{H}_{\text{norm}} = \phi(E_{\text{text}}(q^{\text{norm}}))\,. \tag{1}$$

Here, $\mathbf{H}_{\text{obs},i} \in \mathbb{R}^{T_o \times d}$ and $\mathbf{H}_{\text{norm}} \in \mathbb{R}^{T_n \times d}$, where T_o and T_n denote the respective token lengths and d the projected time-series model dimension. The frozen embeddings are precomputed and cached, while the shared projection adapts them to the learned time-series space. For a given backbone, the normality prompt and its frozen embedding are fixed across datasets and windows; the projected normality reference is learned per model and shared across its windows. Model configurations and prompt construction details are provided in Appendices B and F, respectively.

3.3 SEMANTIC-CONTEXT-CONDITIONED PATCH MASKED RECONSTRUCTION

LEARN-TS employs an asymmetric text-conditioned reconstruction formulation. The numerical branch masks the same temporal region across all variables to suppress direct numerical copying, whereas the semantic branch retains coarse context derived from the fully observed window. Numerical masking is therefore intended to suppress direct identity mapping rather than to formulate information free missing value imputation.

Channel shared temporal patch masking. Let $\overline{\mathbf{X}}_i \in \mathbb{R}^{L\times C}$ denote a feature-wise standardized input window. We divide it into $P = 1 + \lfloor (L-\ell)/s \rfloor$ temporal patches of length ℓ and stride s, indexed by $p \in \{1,\dots,P\}$. Each patch contains all C variables over its temporal interval and is flattened and linearly projected with a positional embedding to form $\mathbf{z}_{i,p} \in \mathbb{R}^d$.

During training, one temporal patch m_i is sampled uniformly and replaced with a trainable mask token:

$$m_i \sim \mathrm{Uniform}\{1,\dots,P\}, \qquad \widetilde{\mathbf{z}}_{i,p}^{(m_i)} = \begin{cases} \mathbf{z}^{\mathrm{mask}}, & p = m_i, \\ \mathbf{z}_{i,p}, & p \neq m_i. \end{cases} \tag{2}$$

Because each patch token jointly represents all variables, this operation masks the same temporal region across every variable and prevents cross variable copying at the held out timestamps. A time-series encoder maps the masked token sequence to numerical patch representations $\mathbf{H}_{S,i}^{(m_i)} \in \mathbb{R}^{P\times d}$. Here, S denotes the time-series branch, i indexes the input window, and P is the number of temporal patches.

Selective semantic conditioning. Starting from $\mathbf{U}_i^{[0]} = \mathbf{H}_{S,i}^{(m_i)}$, where $\mathbf{U}_i^{[0]}$ denotes the initial numerical representation, each fusion block obtains $\mathbf{A}_i^{[k]}$ through multi-head cross-attention using $\mathbf{U}_i^{[k-1]}$ as queries and $\mathbf{H}_{\mathrm{obs},i}$ as keys and values. Because the relevance of semantic context may vary across patches, $\mathbf{A}_i^{[k]}$ is selectively integrated through gated residual updates:

$$\begin{aligned} \mathbf{G}_i^{[k]} &= \sigma\Big(\mathrm{MLP}([\mathbf{U}_i^{[k-1]}; \mathbf{A}_i^{[k]}])\Big), \\ \mathbf{V}_i^{[k]} &= \mathbf{G}_i^{[k]} \odot \mathbf{A}_i^{[k]} + (\mathbf{1} - \mathbf{G}_i^{[k]}) \odot \mathbf{U}_i^{[k-1]}, \\ \mathbf{R}_i^{[k]} &= \mathrm{LN}_1(\mathbf{U}_i^{[k-1]} + \mathbf{V}_i^{[k]}), \qquad \mathbf{U}_i^{[k]} = \mathrm{LN}_2(\mathbf{R}_i^{[k]} + \mathrm{FFN}(\mathbf{R}_i^{[k]})). \end{aligned} \tag{3}$$

Here, $\mathbf{A}_i^{[k]}$ denotes the observation-conditioned cross-attention output of the k-th fusion block, $[\cdot;\cdot]$ denotes concatenation along the feature dimension, σ is the sigmoid function, $\odot$ denotes element-wise multiplication, and LN_1 and LN_2 denote the two LayerNorm operations. After K fusion blocks, the final fused representation is $\mathbf{H}_{\mathrm{fused},i}^{(m_i)} = \mathbf{U}_i^{[K]}$, where K denotes the number of fusion blocks.

Patch-wise reconstruction. A lightweight LayerNorm–linear reconstruction head maps each fused token back to its standardized multivariate patch. The reconstructed patches are combined to form $\widehat{\mathbf{X}}_i^{(m_i)} \in \mathbb{R}^{L\times C}$. The same time-series encoder, fusion blocks, and reconstruction head are reused under the patch mask conditions employed at inference. Model and training configurations are provided in Appendix B.

3.4 JOINT OPTIMIZATION AND NORMALITY-GUIDED SCORING

Training objective. LEARN-TS jointly optimizes masked-patch reconstruction, auxiliary full-window reconstruction, and semantic normality alignment. Let Ω_{mask} denote the masked timestamp–variable positions in a training batch and let Ω_{full} denote all positions. For $a \in \{\mathrm{mask}, \mathrm{full}\}$, the reconstruction losses are

$$\mathcal{L}_a = \frac{1}{|\Omega_a|} \sum_{(i,t,c)\in\Omega_a} \left(\widehat{X}_{i,t,c}^{(m_i)} - \overline{X}_{i,t,c}\right)^2. \tag{4}$$

Here, $c \in \{1,\dots,C\}$ indexes the variables. For a token sequence $\mathbf{H}$, define $\rho(\mathbf{H}) = \mathrm{norm}(\mathrm{MeanPool}(\mathbf{H}))$, with $\mathrm{norm}(\mathbf{v}) = \mathbf{v}/(\|\mathbf{v}\|_2 + \epsilon)$, where $\epsilon > 0$ is a small constant for numerical stability.

The masked fused representation and normality representation are summarized as

$$\mathbf{z}_{\mathrm{TS},i}^{(m_i)} = \rho(\mathbf{H}_{\mathrm{fused},i}^{(m_i)}), \qquad \mathbf{z}_{\mathrm{norm}} = \rho(\mathbf{H}_{\mathrm{norm}}). \tag{5}$$

Because the training windows represent normal operation, their masked fused representations are encouraged to remain close to the shared semantic normality reference, where $\mathcal{B}$ denotes the current mini-batch:

$$\mathcal{L}_{\mathrm{norm}} = \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \left[1 - \cos\left(\mathbf{z}_{\mathrm{TS},i}^{(m_i)}, \mathbf{z}_{\mathrm{norm}} \right) \right]. \tag{6}$$

Applying normality alignment to randomly masked views encourages robustness to stochastic patch corruption. The complete training objective is

$$\mathcal{L} = \mathcal{L}_{\mathrm{mask}} + \lambda_{\mathrm{full}} \mathcal{L}_{\mathrm{full}} + \lambda_{\mathrm{norm}} \mathcal{L}_{\mathrm{norm}}. \tag{7}$$

Here, λ_{full} and λ_{norm} weight the full-window reconstruction and normality-alignment losses, respectively. Although no direct alignment loss is imposed on $\mathbf{H}_{\mathrm{obs},i}$, the observation projection and fusion modules are jointly optimized through the reconstruction and normality-alignment objectives. The language model remains frozen.

Exhaustive patch-wise reconstruction evidence. At inference, LEARN-TS masks each temporal patch once. Let $\widehat{\mathbf{X}}_i^{(p)}$ denote the reconstruction obtained when only patch p is masked. Under the non-overlapping patch configuration used in our experiments, let $p(t)$ denote the unique patch containing timestamp t. The reconstruction evidence is

$$r_{i,t} = \mathcal{G}_{\mathcal{D}} \left(\left\{ \left(\widehat{X}_{i,t,c}^{(p(t))} - \overline{X}_{i,t,c} \right)^2 \right\}_{c=1}^{C} \right), \tag{8}$$

where $\mathcal{G}_{\mathcal{D}}$ is the dataset-specific channel error aggregation operator defined in Appendix B. Thus, $r_{i,t}$ uses only the pass in which $p(t)$ is masked; contributions from overlapping sliding windows are aggregated on the original timeline to obtain r_t.

Unmasked normality discrepancy. A separate unmasked inference pass produces $\mathbf{H}_{\mathrm{fused},i}^{(0)}$, from which the window-level normality discrepancy is computed as

$$\mathbf{z}_{\mathrm{TS},i}^{(0)} = \rho(\mathbf{H}_{\mathrm{fused},i}^{(0)}), \qquad d_i = 1 - \cos\left(\mathbf{z}_{\mathrm{TS},i}^{(0)}, \mathbf{z}_{\mathrm{norm}} \right). \tag{9}$$

Masked-view alignment during training encourages robustness to patch corruption, whereas the unmasked inference view makes d_i independent of a particular mask position. Each d_i is assigned to the timestamps covered by its window, and overlapping contributions are aggregated to obtain d_t. The dataset-specific aggregation rules are reported in Table 6.

Normality-gated anomaly score. Because reconstruction evidence and normality discrepancy have different scales, we independently calibrate them using normal validation score distributions, obtaining $r_{z,t}$ and $d_{z,t}$, which denote the calibrated reconstruction evidence and calibrated normality discrepancy, respectively. The calibration procedure is provided in Appendix B. The final anomaly score is

$$S_t = r_{z,t} \left[1 + \lambda_{\mathrm{gate}} \, \mathrm{ReLU}(d_{z,t}) \right]. \tag{10}$$

Here, λ_{gate} is a weighting hyperparameter controlling the strength of discrepancy based modulation.

A positive discrepancy amplifies positive reconstruction deviations while pushing negative deviations farther below the decision threshold, whereas a nonpositive discrepancy leaves the calibrated reconstruction evidence unchanged. Thus, discrepancy from the shared normality reference acts as a conditional modulation signal rather than an independent anomaly score. The final prediction is $\widehat{y}_t = \mathbb{I}[S_t > \tau]$.

4 Experiments

We evaluate LEARN-TS on four real-world MTSAD benchmarks and analyze its detection performance, major components, and sensitivity to key design choices.

Table 1: Performance comparison on four real-world multivariate time-series anomaly detection datasets. Results are averaged over five runs; standard deviations are reported in Table 7. For each dataset–metric pair, the best and second-best results are shown in **bold** and underlined, respectively.

Dataset	Metric	Omni	A.T.	Times	Patch	iTrans	G4TS	TLLM	LMixer	CALF	DADA	LEARN-TS
SWaT	A-PR	0.1492	0.6829	0.1341	0.0890	0.0931	0.0894	0.0846	0.0866	0.0819	0.5385	**0.7578**
	V-PR	0.1371	0.4508	0.1656	0.0982	0.1008	0.0991	0.0930	0.0951	0.0907	0.4543	**0.4859**
	R-F1	0.1718	0.1602	0.1593	0.1097	0.1277	0.1344	0.1190	0.1201	0.1137	0.0990	**0.3102**
	Aff-F1	0.7137	0.7413	**0.7940**	0.6376	0.6621	0.6826	0.6658	0.6737	0.6570	0.7119	0.7277
SMD	A-PR	0.3713	0.2575	0.4640	0.4893	0.4726	**0.4947**	0.4514	0.4307	0.4578	0.4754	0.4930
	V-PR	0.3752	0.3214	0.6184	0.6376	0.6250	**0.6455**	0.5839	0.5662	0.5910	0.5912	0.5398
	R-F1	0.1178	0.1930	0.2175	0.2388	0.2375	0.2427	0.2611	0.2177	0.2725	0.2683	**0.3203**
	Aff-F1	0.6510	0.6818	0.7313	0.7297	0.7284	0.7153	0.7507	0.7156	0.7553	0.7543	**0.8121**
PSM	A-PR	0.3401	0.3886	0.4013	0.4307	0.4250	0.4269	0.4169	0.4209	0.3999	0.4295	**0.5700**
	V-PR	0.3622	0.4243	0.5421	0.5667	0.5863	0.5490	0.5021	0.5137	0.5136	0.5198	**0.6025**
	R-F1	0.3091	0.2168	0.4451	0.4685	0.4712	0.4896	0.4665	0.4690	0.4354	0.4495	**0.5256**
	Aff-F1	0.5980	0.4356	0.7145	0.6545	0.7005	0.7092	0.6873	0.6852	0.7021	0.6404	**0.7765**
MSL	A-PR	0.1706	0.1353	0.2823	0.2696	0.3116	0.2799	0.2851	0.2826	0.2818	0.2851	**0.3472**
	V-PR	0.2139	0.1742	0.3474	0.3195	0.3256	0.3253	0.3254	0.3248	0.3302	0.3259	**0.3752**
	R-F1	0.1231	0.1427	0.2206	0.1584	0.1255	0.2178	0.2581	0.2844	0.1392	0.1962	**0.2951**
	Aff-F1	0.4981	0.5765	0.6363	0.4137	0.3429	0.6232	0.6666	0.6746	0.4943	0.5961	**0.7481**

4.1 Experimental Setting

Datasets. We evaluate LEARN-TS on SWaT (Goh et al., 2017), SMD (Su et al., 2019), PSM (Abdulaal et al., 2021), and MSL (Hundman et al., 2018), covering industrial control, server monitoring, application services, and spacecraft telemetry. SMD is evaluated entity-wise, whereas MSL uses a shared model across its telemetry entities, with windows constructed separately within each entity. Dataset statistics, splits, and entity-handling protocols are provided in Appendix A.

Baselines. We compare LEARN-TS with 10 representative baselines: (1) dedicated TSAD methods, OmniAnomaly (Omni) (Su et al., 2019) and Anomaly Transformer (A.T.) (Xu et al., 2022); (2) general time-series backbones, TimesNet (Times) (Wu et al., 2023), PatchTST (Patch) (Nie et al., 2023), and iTransformer (iTrans) (Liu et al., 2024); (3) language model-based methods, GPT4TS (G4TS) (Zhou et al., 2023), Time-LLM (TLLM) (Jin et al., 2024), LLM-Mixer (LMixer) (Kowsher et al., 2025), and CALF (Liu et al., 2025); and (4) the pretrained general anomaly detector DADA (Shentu et al., 2025). We omit MindTS because matching its external-text setting on these benchmarks would require additional text curation. We use public implementations and recommended configurations when available, with a common downstream metric and thresholding pipeline for all methods. Unless otherwise stated, LEARN-TS uses GPT-2 (Radford et al., 2019) as the frozen language backbone; backbone robustness is evaluated in Appendix E.

Metrics. We report two threshold-dependent event metrics, Range-based F1 (R-F1) (Tatbul et al., 2018) and Affiliation F1 (Aff-F1) (Huet et al., 2022), and two threshold-independent ranking metrics, Area under the Precision–Recall curve (A-PR) (Saito & Rehmsmeier, 2015) and Volume under the Precision–Recall surface (VUS-PR, denoted V-PR) (Paparrizos et al., 2022). Our overlap-aware R-F1 accounts for temporal coverage and fragmented detections. Together, these metrics assess event-level localization and anomaly score discrimination without point adjustment (Kim et al., 2022). Model hyperparameters and calibration parameters are fixed using validation data. For threshold-dependent metrics, we report the R-F1-maximizing operating point over normal-validation quantile candidates; Aff-F1 uses the same threshold. Further details appear in Appendices A and B.

4.2 Main Results

Table 1 presents the performance comparison on four real-world multivariate time-series anomaly detection datasets. The results of LEARN-TS are averaged over five random seeds. It achieves the best performance in 13 of 16 dataset–metric combinations, including R-F1 on all four datasets and

Table 2: Ablation analysis on the four-dataset macro average. Results are five-seed mean ± sample standard deviation; the best result within each panel is shown in **bold**.

(a) Cross-modal fusion

Variant	A-PR	V-PR	R-F1	Aff-F1
w/o Obs.	0.5119 ± 0.0153	0.4746 ± 0.0041	0.3333 ± 0.0064	0.7636 ± 0.0045
Concat fusion	0.5287 ± 0.0284	0.4867 ± 0.0151	0.3500 ± 0.0054	**0.7688 ± 0.0037**
Full	**0.5420 ± 0.0060**	**0.5008 ± 0.0105**	**0.3628 ± 0.0119**	0.7661 ± 0.0066

(b) Normality alignment and scoring

Variant	Align.	Score	A-PR	V-PR	R-F1	Aff-F1
No normality	✗	✗	0.3865 ± 0.0062	0.4108 ± 0.0036	**0.3654 ± 0.0109**	**0.7711 ± 0.0022**
Scoring only	✗	✓	0.3801 ± 0.0110	0.4136 ± 0.0078	0.3531 ± 0.0133	0.7644 ± 0.0058
Alignment only	✓	✗	0.3831 ± 0.0047	0.4106 ± 0.0023	0.3579 ± 0.0089	0.7705 ± 0.0023
Full	✓	✓	**0.5420 ± 0.0060**	**0.5008 ± 0.0105**	0.3628 ± 0.0119	0.7661 ± 0.0066

(c) Normality-reference control

Normality reference	A-PR	V-PR	R-F1	Aff-F1
Random semantic-free	0.5360 ± 0.0113	0.4965 ± 0.0104	0.3596 ± 0.0052	0.7623 ± 0.0045
Dataset-agnostic semantic	**0.5420 ± 0.0060**	**0.5008 ± 0.0105**	**0.3628 ± 0.0119**	**0.7661 ± 0.0066**

A-PR on three. The gains are most consistent on PSM and MSL, where LEARN-TS ranks first across all four metrics. On SWaT, it leads in A-PR, V-PR, and R-F1, while TimesNet attains the highest Aff-F1. On SMD, LEARN-TS achieves the best R-F1 and Aff-F1, whereas GPT4TS obtains the best A-PR and V-PR. Overall, the results indicate that LEARN-TS performs competitively across complementary ranking- and event-level metrics, although the gains vary across datasets.

4.3 Model Analysis

Table 2 separates the effects of observation fusion, normality guidance, and reference content using five-seed macro averages across four datasets. Dataset-wise results appear in Appendix C.

Cross-modal fusion. Table 2(a) compares removing observation conditioning (w/o Obs.), concatenating pooled observation embeddings with numerical representations (Concat fusion), and the proposed gated cross-modal fusion (Full). Removing observation conditioning reduces macro-averaged A-PR, V-PR, and R-F1, while concatenation partially recovers performance. Full achieves the highest A-PR, V-PR, and R-F1, whereas Concat fusion yields slightly higher Aff-F1. These results support the utility of the observation pathway and selective cross-modal fusion.

Normality alignment and scoring. Under the semantic-reference configuration, Table 2(b) shows that jointly applying normality alignment and discrepancy-guided scoring improves macro-averaged A-PR and V-PR over either component alone or no normality guidance. Neither component alone reproduces the ranking gains of their joint use, supporting their complementary roles. Mean R-F1 and Aff-F1 remain slightly below the no-normality variant, indicating that the joint mechanism primarily benefits score ranking rather than uniformly improving event-level detection.

Normality-reference control. With the alignment-and-scoring pipeline held fixed, Table 2(c) compares the dataset-agnostic normality reference with a random semantic-free reference. The semantic reference achieves slightly higher means across all four metrics, suggesting a modest additional benefit from reference content. Reference construction details are provided in Appendix F.1.

Observation–window correspondence. Figure 3 compares matched observations with representations shuffled across windows while preserving the architecture. Using seed 42, matched observations improve reconstruction-only A-PR and final A-PR and V-PR across all four datasets, supporting the importance of correspondence between observation semantics and the underlying window.

Qualitative analysis. We compare with GPT4TS, the top SMD baseline in A-PR and V-PR. Figure 4 compares event-level predictions on three SMD entities with different anomaly densities. Here, LEARN-TS predictions often overlap more closely with labeled events, while GPT4TS misses some events in the sparse case and produces detections outside the main event cluster in the dense case.

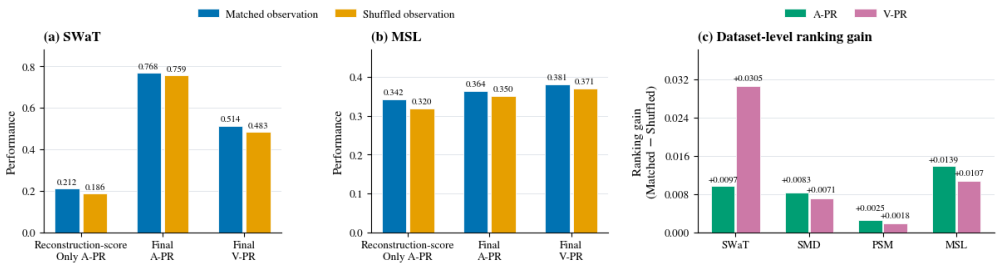

Figure 3: Effect of observation–window correspondence: (a)–(b) matched vs. shuffled observations on SWaT and MSL; (c) A-PR and V-PR gains across all four datasets.

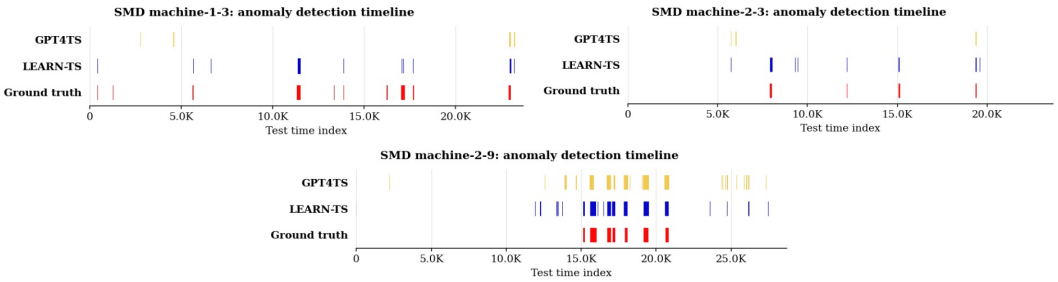

Figure 4: Event-level predictions on three SMD entities: Machine 1-3 with intermediate event density, Machine 2-3 with sparse events, and Machine 2-9 with dense events.

Sensitivity analysis. Figure 5 summarizes fixed-seed sensitivity trends. Removing the full-window objective or normality guidance reduces macro ranking performance, whereas performance is generally more stable across positive weight values. A nonzero discrepancy gate improves A-PR and V-PR over reconstruction-only scoring, although its effects vary across datasets and metrics. Single-patch masking yields the most consistent aggregate performance among the evaluated masking configurations. Overall, the selected values provide favorable aggregate trade-offs across the four reported metrics. Dataset-wise results and validation-selected settings are provided in Appendices D and B, respectively.

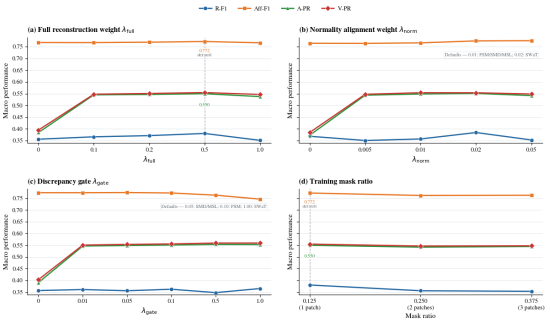

Figure 5: Sensitivity to (a) λ_{full}, (b) λ_{norm}, (c) λ_{gate}, and (d) the number of masked training patches. Results are four-dataset macro averages using a fixed seed.

5 Conclusion

We studied MTSAD without requiring time-associated paired external text. We proposed LEARN-TS, a semantically guided framework that combines observation-conditioned patch-masked reconstruction with a window-independent normality reference. Across four real-world benchmarks, LEARN-TS ranks first in 13 of 16 primary dataset–metric comparisons. Controlled analyses support the importance of observation–window correspondence and show that normality alignment and discrepancy-guided scoring benefit anomaly-score ranking, with modest gains from the dataset-agnostic semantic reference over a random semantic-free alternative. Future work will investigate expert-validated sensor-level normality constraints and reduce the cost of patch-wise inference.

AI USE STATEMENT

We used AI assistants for feedback on experimental design, interpretation of results, literature discovery, and drafting and editing manuscript text and LaTeX tables. The authors reviewed the AI-assisted material, made the final research decisions, and take responsibility for the content.

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*, KDD '21, pp. 2485–2494, New York, NY, USA, 2021. Association for Computing Machinery. doi: 10.1145/3447548.3467174. URL https://doi.org/10.1145/3447548.3467174.

Julien Audibert, Pietro Michiardi, Frédéric Guyard, Sébastien Marti, and Maria A. Zuluaga. USAD: Unsupervised anomaly detection on multivariate time series. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '20, pp. 3395–3404, New York, NY, USA, 2020. Association for Computing Machinery. doi: 10.1145/3394486.3403392. URL https://doi.org/10.1145/3394486.3403392.

Jonathan Goh, Sridhar Adepu, Khurum Nazir Junejo, and Aditya Mathur. A dataset to support research in the design of secure water treatment systems. In Grigore Havarneanu, Roberto Setola, Hypatia Nassopoulos, and Stephen Wolthusen (eds.), *Critical Information Infrastructures Security*, volume 10242 of *Lecture Notes in Computer Science*, pp. 88–99, Cham, 2017. Springer International Publishing. doi: 10.1007/978-3-319-71368-7_8. URL https://doi.org/10.1007/978-3-319-71368-7_8.

Shiyan Hu, Jianxin Jin, Yang Shu, Peng Chen, Bin Yang, and Chenjuan Guo. Towards multimodal time series anomaly detection with semantic alignment and condensed interaction. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=fNFbGqu6Rg.

Alexis Huet, Jose Manuel Navarro, and Dario Rossi. Local evaluation of time series anomaly detection algorithms. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 635–645. Association for Computing Machinery, 2022. doi: 10.1145/3534678.3539339. URL https://doi.org/10.1145/3534678.3539339.

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*, KDD '18, pp. 387–395, New York, NY, USA, 2018. Association for Computing Machinery. doi: 10.1145/3219819.3219845. URL https://doi.org/10.1145/3219819.3219845.

Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y. Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, and Qingsong Wen. Time-LLM: Time series forecasting by reprogramming large language models. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=Unb5CVPtae.

Siwon Kim, Kukjin Choi, Hyun-Soo Choi, Byunghan Lee, and Sungroh Yoon. Towards a rigorous evaluation of time-series anomaly detection. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(7):7194–7201, 2022. doi: 10.1609/aaai.v36i7.20680. URL https://doi.org/10.1609/aaai.v36i7.20680.

Md Kowsher, Md. Shohanur Islam Sobuj, Nusrat Jahan Prottasha, E. Alejandro Alanis, Ozlem Garibay, and Niloofar Yousefi. LLM-Mixer: Multiscale mixing in LLMs for time series forecasting. In *Proceedings of the 4th Table Representation Learning Workshop*, pp. 156–165, Vienna, Austria, 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.trl-1.12. URL https://aclanthology.org/2025.trl-1.12/.

Yuxin Li, Wenchao Chen, Bo Chen, Dongsheng Wang, Long Tian, and Mingyuan Zhou. Prototype-oriented unsupervised anomaly detection for multivariate time series. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 19407–19424. PMLR, 2023. URL https://proceedings.mlr.press/v202/li23d.html.

Peiyuan Liu, Hang Guo, Tao Dai, Naiqi Li, Jigang Bao, Xudong Ren, Yong Jiang, and Shu-Tao Xia. CALF: Aligning LLMs for time series forecasting via cross-modal fine-tuning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(18):18915–18923, 2025. doi: 10.1609/aaai.v39i18.34082. URL https://doi.org/10.1609/aaai.v39i18.34082.

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. iTransformer: Inverted transformers are effective for time series forecasting. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=JePfAI8fah.

Meta. Llama 3.2 model card. https://github.com/meta-llama/llama-models/blob/main/models/llama3_2/MODEL_CARD.md, 2024. Accessed: 2026-09-09.

Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=Jbdc0vTOcol.

John Paparrizos, Paul Boniol, Themis Palpanas, Ruey S. Tsay, Aaron Elmore, and Michael J. Franklin. Volume under the surface: A new accuracy evaluation measure for time-series anomaly detection. *Proceedings of the VLDB Endowment*, 15(11):2774–2787, 2022. doi: 10.14778/3551793.3551830. URL https://doi.org/10.14778/3551793.3551830.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. OpenAI technical report, 2019. URL https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf.

Lukas Ruff, Robert Vandermeulen, Nico Goernitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, Emmanuel Müller, and Marius Kloft. Deep one-class classification. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 4393–4402. PMLR, 2018. URL https://proceedings.mlr.press/v80/ruff18a.html.

Takaya Saito and Marc Rehmsmeier. The precision–recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. *PLOS ONE*, 10(3):e0118432, 2015. doi: 10.1371/journal.pone.0118432. URL https://doi.org/10.1371/journal.pone.0118432.

Lifeng Shen, Zhuocong Li, and James T. Kwok. Timeseries anomaly detection using temporal hierarchical one-class network. In *Advances in Neural Information Processing Systems*, volume 33, pp. 13016–13026. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/hash/97e401a02082021fd24957f852e0e475-Abstract.html.

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. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/ca7998666c2e53cc1e882b7268414d8a-Abstract-Conference.html.

Junho Song, Keonwoo Kim, Jeonglyul Oh, and Sungzoon Cho. MEMTO: Memory-guided transformer for multivariate time series anomaly detection. In *Advances in Neural Information Processing Systems*, volume 36, pp. 57947–57963, 2023. doi: 10.52202/075280-2525. URL https://doi.org/10.52202/075280-2525.

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. Association for Computing Machinery, 2019. doi: 10.1145/3292500.3330672. URL https://doi.org/10.1145/3292500.3330672.

Xin Sun, Heng Zhou, and Chao Li. Multivariate time series anomaly detection with idempotent reconstruction. In *Advances in Neural Information Processing Systems*, volume 38. Curran Associates, Inc., 2025. doi: 10.52202/085713-5374. URL https://papers.nips.cc/paper_files/paper/2025/hash/ebba26f4dbcfe7318e4a263c54e9cf42-Abstract-Conference.html.

Mingtian Tan, Mike A. Merrill, Vinayak Gupta, Tim Althoff, and Thomas Hartvigsen. Are language models actually useful for time series forecasting? In *Advances in Neural Information Processing Systems*, volume 37, pp. 60162–60191, 2024. doi: 10.52202/079017-1922. URL https://doi.org/10.52202/079017-1922.

Nesime Tatbul, Tae Jun Lee, Stan Zdonik, Mejbah Alam, and Justin Gottschlich. Precision and recall for time series. In *Advances in Neural Information Processing Systems*, volume 31, pp. 1924–1934, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/8f468c873a32bb0619eaeb2050ba45d1-Abstract.html.

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. In *Advances in Neural Information Processing Systems*, volume 36. Curran Associates, Inc., 2023. doi: 10.52202/075280-0473. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/22f5d8e689d2a011cd8ead552ed59052-Abstract-Conference.html.

Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. TimesNet: Temporal 2D-variation modeling for general time series analysis. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=ju_Uqw384Oq.

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. URL https://openreview.net/forum?id=LzQQ89U1qm_.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024. URL https://arxiv.org/abs/2412.15115.

Tian Zhou, Peisong Niu, Xue Wang, Liang Sun, and Rong Jin. One fits all: Power general time series analysis by pretrained LM. In *Advances in Neural Information Processing Systems*, volume 36, pp. 43322–43355, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/86c17de05579cde52025f9984e6e2ebb-Abstract-Conference.html.

A Dataset Statistics and Evaluation Protocol

Table 3 summarizes the datasets used in our experiments. The reported train and test sizes correspond to the official splits before our internal training–validation division. For SMD and MSL, sequence lengths are summed over all entities, and the anomaly ratio denotes the percentage of anomalous timestamps in the official test split.

Preprocessing and validation split. We preserve the official test splits without modification. Each official training sequence is chronologically divided into the first 80% for training and the remaining 20% for validation before window construction. For SMD and MSL, this split is performed independently within each entity. Feature-wise standardization statistics are estimated exclusively from the resulting training portions and applied unchanged to validation and test data. Windows have length $L = 128$, and no window crosses an entity boundary. Normal validation data are used for score calibration, hyperparameter selection, and quantile-based threshold candidate construction. For threshold-dependent metrics, we report the operating point that maximizes R-F1 over these candidates and evaluate Aff-F1 at the same threshold.

Entity-wise protocol. For SMD, a separate model is trained and evaluated for each of the 28 machines, and machine-level metrics are macro-averaged. For MSL, a shared model is trained on windows constructed within each of the 27 telemetry entities. Inference is performed separately for each entity, after which scores and labels are concatenated. Concatenated MSL processing is also used in the public OmniAnomaly implementation (Su et al., 2019). Smoothing and metric computation are applied to the pooled sequence without explicit boundary handling. SWaT and PSM are each treated as a single multivariate sequence.

Timeline aggregation and evaluation. Window-level outputs are mapped back to the original timeline. Reconstruction errors are retained only from inference passes in which the corresponding temporal patch is masked. Window-level normality discrepancies are broadcast to their covered timestamps, and overlapping contributions are aggregated using mean aggregation for SWaT and MSL and maximum aggregation for SMD and PSM. All methods are evaluated using the same score-processing, metric, and thresholding pipeline.

Baseline evaluation. For all baselines, we preserve each method's original anomaly-scoring procedure and evaluate the resulting anomaly-score sequences using the same downstream score-processing, metric, and threshold selection pipeline adopted for LEARN-TS. For DADA, we use the officially released pretrained checkpoint in its zero-shot setting without target domain fine tuning, while preserving its official preprocessing and inference procedures.

Metric implementation. Before evaluation, score smoothing is applied uniformly to all methods using a window of 30 timestamps for SWaT and 10 for the other datasets. R-F1 uses an overlap-aware range configuration that accounts for temporal coverage and fragmented detections, while Aff-F1 evaluates affiliation between predicted and ground-truth events.

A-PR is computed from continuous anomaly scores and the original point-wise labels. VUS-PR is computed using the reference implementation of Paparrizos et al. (2022) (`RangeAUC_volume_opt`), with 250 score thresholds and temporal buffer lengths up to 200 timestamps. A-PR and VUS-PR are threshold-independent and do not use the selected operating point. No point adjustment is applied to any metric. Additional evaluator settings are provided in the anonymized code.

B Implementation Details

Exact implementation constants and complete configuration files are provided in the anonymized code release.

Data preprocessing. Feature-wise standardization statistics are computed exclusively from the normal training split and subsequently applied without modification to the validation and test splits.

Table 3: Dataset statistics before the internal training–validation split. For SMD and MSL, sequence lengths are summed over all entities.

Dataset	Domain	Entities	Variables	Official train	Official test	Anomaly (%)
SWaT	Industrial control	1	51	496,800	449,919	11.98
SMD	Server monitoring	28	38	708,405	708,420	4.16
PSM	Application server	1	25	132,481	87,841	27.75
MSL	Spacecraft telemetry	27	55	58,317	73,729	10.72

Table 4: Common model and optimization settings used across all datasets.

Hyperparameter	Value
Input window length L	128
Patch length ℓ	16
Number of patches P	8
Hidden dimension d	768
Time-series encoder layers	3
Cross-modal fusion layers	2
Attention heads	8
Batch size	32
Optimizer	AdamW
Weight decay	1×10^{-4}
Gradient clipping	1.0
Initial learning rate	$5 \times 10^{-5\dagger}$
Minimum learning rate	1×10^{-6}
Early-stopping patience	5 epochs
λ_{full}	0.5

† PSM uses an initial learning rate of 2×10^{-5}.

Each multivariate time series is segmented into input windows of length $L = 128$. Each window is divided into $P = 8$ non-overlapping temporal patches of length $\ell = 16$.

Patch masking. During training and validation, one temporal patch is sampled uniformly and masked at the same temporal location across all variables. At inference, stochastic masking is replaced by deterministic exhaustive patch-wise masking. Specifically, each of the P patches is masked once, resulting in eight single patch masked forward passes for each input window. Reconstruction errors are retained only for timestamps belonging to the currently masked patch. The retained errors from all forward passes are then combined to construct dense reconstruction evidence over the complete input window.

Model configuration. The hidden dimension is set to $d = 768$. The time-series encoder consists of three layers, and the gated cross-modal fusion module consists of two layers. Each encoder and fusion layer uses eight attention heads. The text encoder is kept frozen, whereas the time-series encoder, text projection module, cross-modal fusion module, and reconstruction head are optimized during training. The common model and optimization settings used across all datasets are summarized in Table 4.

Optimization. All trainable components are optimized using AdamW with a batch size of 32, weight decay of 10^{-4}, and gradient clipping with a maximum norm of 1.0. The initial learning rate is 5×10^{-5} for SWaT, SMD, and MSL, and 2×10^{-5} for PSM. We apply cosine annealing with a minimum learning rate of 10^{-6}. Training is terminated when the validation loss does not improve for five consecutive epochs. All experiments were conducted on an NVIDIA L40S GPU with 48 GB of GPU memory.

Table 5: Dataset-specific loss and scoring configurations.

Dataset	λ_{norm}	λ_{gate}
SWaT	0.02	1.00
SMD	0.01	0.05
PSM	0.01	0.1
MSL	0.01	0.05

Table 6: Dataset-specific channel error and overlapping-window aggregation rules.

Dataset	Channel aggregation $\mathcal{G}_{\mathcal{D}}$	Window overlap
SWaT	Mean over all variables	Mean
SMD	Top-k mean	Maximum
PSM	Top-k mean	Maximum
MSL	Target telemetry channel	Mean

Loss configuration. We use $\lambda_{\text{full}} = 0.5$ for all datasets. The normality-alignment weight is set to $\lambda_{\text{norm}} = 0.02$ for SWaT and $\lambda_{\text{norm}} = 0.01$ for SMD, PSM, and MSL. The dataset-specific configurations are summarized in Table 5.

Dataset-specific score aggregation. The operator $\mathcal{G}_{\mathcal{D}}$ converts the channel-wise reconstruction errors in Equation 8 into a scalar score using the dataset-specific rule in Table 6. For top-k aggregation, the k largest channel errors are averaged. Scores contributed by overlapping windows are aggregated using either their mean or maximum. The same window-overlap rule is used to aggregate the broadcast normality discrepancies.

Score calibration and anomaly prediction. Reconstruction evidence and normality discrepancy are calibrated independently using their timestamp-level distributions on normal validation data. For $a \in \{r, d\}$, let $\mathcal{Q}_N^{(a)}$ denote the corresponding normal validation distribution. The calibrated score is

$$a_{z,t} = \text{clip}\left(\frac{a_t - \text{median}(\mathcal{Q}_N^{(a)})}{s_N^{(a)}}, -\kappa, \kappa\right), \qquad a \in \{r, d\}, \tag{11}$$

where $s_N^{(a)}$ is a positive robust scale computed from $\mathcal{Q}_N^{(a)}$. In implementation, the scale takes the maximum of normalized median absolute deviation, central quantile range, standard deviation, and positive-floor estimates. This composite definition prevents near-degenerate validation score distributions from producing excessively large calibrated values.

The calibrated scores are combined according to Equation 10. The discrepancy-gating coefficient is selected using validation data. Quantile-based threshold candidates are constructed from normal validation scores, with the final operating point selected as described in Appendix A.

Random seeds and reporting. The main results are averaged over five independent runs with different random seeds. For each trainable method, the same five-seed evaluation protocol is used. The module ablations in Table 2 use five seeds, whereas sensitivity analyses and the observation correspondence control use a single fixed seed (42).

Dataset-level results are reported as the mean and sample standard deviation over five random seeds. For trainable methods, the seeds control model initialization and stochastic training. DADA uses its fixed officially released pretrained checkpoint and is independently evaluated under five random seeds, which control the stochastic components of its inference pipeline. For four-dataset macro results, the four datasets are first averaged with equal weight within each seed, after which the final mean and sample standard deviation are computed over the five seed-level averages. Table 7 reports five-run summary statistics for each method, dataset, and primary metric. Per-seed raw results are provided in the supplementary result files. Evaluation labels are not used for model selection or model hyperparameter tuning. For metric reporting, they select the R-F1-maximizing threshold from validation-derived candidates; Aff-F1 uses the same threshold.

Table 7: Performance of all methods on the four datasets over five independent runs. Results are reported as mean ± sample standard deviation. For each dataset–metric pair, the best and second-best mean results are shown in **bold** and underlined, respectively.

Method	A-PR	V-PR	R-F1	Aff-F1
(a) SWaT				
OmniAnomaly	0.1492 ± 0.0559	0.1371 ± 0.0321	0.1718 ± 0.1140	0.7137 ± 0.0041
Anomaly Transformer	0.6829 ± 0.0193	0.4508 ± 0.0077	0.1602 ± 0.0187	0.7413 ± 0.0149
TimesNet	0.1341 ± 0.0018	0.1656 ± 0.0021	0.1593 ± 0.0057	**0.7940 ± 0.0091**
PatchTST	0.0890 ± 0.0002	0.0982 ± 0.0002	0.1097 ± 0.0062	0.6376 ± 0.0026
iTransformer	0.0931 ± 0.0010	0.1008 ± 0.0005	0.1277 ± 0.0029	0.6621 ± 0.0025
GPT4TS	0.0894 ± 0.0002	0.0991 ± 0.0003	0.1344 ± 0.0015	0.6826 ± 0.0056
Time-LLM	0.0846 ± 0.0003	0.0930 ± 0.0003	0.1190 ± 0.0021	0.6658 ± 0.0016
LLM-Mixer	0.0866 ± 0.0004	0.0951 ± 0.0004	0.1201 ± 0.0033	0.6737 ± 0.0009
CALF	0.0819 ± 0.0003	0.0907 ± 0.0004	0.1137 ± 0.0016	0.6570 ± 0.0062
DADA	0.5385 ± 0.0042	0.4543 ± 0.0031	0.0990 ± 0.0061	0.7119 ± 0.0076
LEARN-TS	**0.7578 ± 0.0117**	**0.4859 ± 0.0346**	**0.3102 ± 0.0436**	0.7277 ± 0.0069
(b) SMD				
OmniAnomaly	0.3713 ± 0.0123	0.3752 ± 0.0083	0.1178 ± 0.0060	0.6510 ± 0.0087
Anomaly Transformer	0.2575 ± 0.0110	0.3214 ± 0.0050	0.1930 ± 0.0060	0.6818 ± 0.0183
TimesNet	0.4640 ± 0.0048	0.6184 ± 0.0027	0.2175 ± 0.0049	0.7313 ± 0.0052
PatchTST	0.4893 ± 0.0009	0.6376 ± 0.0009	0.2388 ± 0.0022	0.7297 ± 0.0017
iTransformer	0.4726 ± 0.0008	0.6250 ± 0.0010	0.2375 ± 0.0028	0.7284 ± 0.0014
GPT4TS	**0.4947 ± 0.0006**	**0.6455 ± 0.0008**	0.2427 ± 0.0066	0.7153 ± 0.0039
Time-LLM	0.4514 ± 0.0008	0.5839 ± 0.0013	0.2611 ± 0.0026	0.7507 ± 0.0020
LLM-Mixer	0.4307 ± 0.0486	0.5662 ± 0.0445	0.2177 ± 0.0216	0.7156 ± 0.0231
CALF	0.4578 ± 0.0013	0.5910 ± 0.0009	0.2725 ± 0.0010	0.7553 ± 0.0005
DADA	0.4754 ± 0.0030	0.5912 ± 0.0034	0.2683 ± 0.0015	0.7543 ± 0.0016
LEARN-TS	0.4930 ± 0.0056	0.5398 ± 0.0037	**0.3203 ± 0.0033**	**0.8121 ± 0.0056**
(c) PSM				
OmniAnomaly	0.3401 ± 0.0392	0.3622 ± 0.0224	0.3091 ± 0.0976	0.5980 ± 0.0703
Anomaly Transformer	0.3886 ± 0.0527	0.4243 ± 0.0449	0.2168 ± 0.0308	0.4356 ± 0.0466
TimesNet	0.4013 ± 0.0046	0.5421 ± 0.0089	0.4451 ± 0.0097	0.7145 ± 0.0046
PatchTST	0.4307 ± 0.0027	0.5667 ± 0.0031	0.4685 ± 0.0045	0.6545 ± 0.0162
iTransformer	0.4250 ± 0.0024	0.5863 ± 0.0027	0.4712 ± 0.0075	0.7005 ± 0.0047
GPT4TS	0.4269 ± 0.0021	0.5490 ± 0.0014	0.4896 ± 0.0072	0.7092 ± 0.0133
Time-LLM	0.4169 ± 0.0011	0.5021 ± 0.0032	0.4665 ± 0.0023	0.6873 ± 0.0037
LLM-Mixer	0.4209 ± 0.0007	0.5137 ± 0.0022	0.4690 ± 0.0029	0.6852 ± 0.0036
CALF	0.3999 ± 0.0005	0.5136 ± 0.0031	0.4354 ± 0.0031	0.7021 ± 0.0001
DADA	0.4295 ± 0.0025	0.5198 ± 0.0024	0.4495 ± 0.0089	0.6404 ± 0.0103
LEARN-TS	**0.5700 ± 0.0100**	**0.6025 ± 0.0094**	**0.5256 ± 0.0064**	**0.7765 ± 0.0103**
(d) MSL				
OmniAnomaly	0.1706 ± 0.0170	0.2139 ± 0.0253	0.1231 ± 0.0189	0.4981 ± 0.0435
Anomaly Transformer	0.1353 ± 0.0062	0.1742 ± 0.0056	0.1427 ± 0.0084	0.5765 ± 0.0201
TimesNet	0.2823 ± 0.0044	0.3474 ± 0.0046	0.2206 ± 0.0081	0.6363 ± 0.0360
PatchTST	0.2696 ± 0.0090	0.3195 ± 0.0095	0.1584 ± 0.0257	0.4137 ± 0.0568
iTransformer	0.3116 ± 0.0048	0.3256 ± 0.0035	0.1255 ± 0.0088	0.3429 ± 0.0185
GPT4TS	0.2799 ± 0.0024	0.3253 ± 0.0024	0.2178 ± 0.0237	0.6232 ± 0.0361
Time-LLM	0.2851 ± 0.0003	0.3254 ± 0.0002	0.2581 ± 0.0028	0.6666 ± 0.0010
LLM-Mixer	0.2826 ± 0.0057	0.3248 ± 0.0029	0.2844 ± 0.0199	0.6746 ± 0.0157
CALF	0.2818 ± 0.0004	0.3302 ± 0.0004	0.1392 ± 0.0009	0.4943 ± 0.0031
DADA	0.2851 ± 0.0084	0.3259 ± 0.0085	0.1962 ± 0.0164	0.5961 ± 0.0331
LEARN-TS	**0.3472 ± 0.0184**	**0.3752 ± 0.0053**	**0.2951 ± 0.0023**	**0.7481 ± 0.0282**

C Dataset-Wise Ablation Results

The main paper reports five-seed macro averaged ablation results for a compact comparison of the major components. Here, we provide dataset-wise results for cross-modal fusion, normality alignment and scoring, and semantic-reference construction to examine whether their effects are consistent across benchmarks.

C.1 Cross-Modal Fusion

Table 8 reports the dataset-wise cross-modal fusion ablation. We compare three architectural configurations: *w/o Obs.*, which removes observation-text conditioning; *Concat fusion*, which replaces gated cross-attention with pooled text concatenation; and *LEARN-TS (Full)*, which uses gated cross-modal fusion with the corresponding observation representation.

Table 8: Dataset-wise cross-modal fusion ablation. Results are reported as mean ± sample standard deviation over five random seeds. The best result in each dataset–metric column is shown in **bold**.

Dataset	Variant	A-PR	V-PR	R-F1	Aff-F1
SWaT	w/o Obs.	0.6552 ± 0.0652	0.3727 ± 0.0193	0.2446 ± 0.0110	0.7047 ± 0.0012
	Concat fusion	0.7001 ± 0.1014	0.4170 ± 0.0508	0.2911 ± 0.0189	0.7270 ± 0.0090
	LEARN-TS (Full)	**0.7578 ± 0.0117**	**0.4859 ± 0.0346**	**0.3102 ± 0.0436**	**0.7277 ± 0.0069**
SMD	w/o Obs.	0.4650 ± 0.0074	0.5266 ± 0.0047	0.3222 ± 0.0070	0.8137 ± 0.0102
	Concat fusion	0.4829 ± 0.0122	0.5380 ± 0.0098	**0.3262 ± 0.0057**	**0.8174 ± 0.0084**
	LEARN-TS (Full)	**0.4930 ± 0.0056**	**0.5398 ± 0.0037**	0.3203 ± 0.0033	0.8121 ± 0.0056
PSM	w/o Obs.	0.5767 ± 0.0047	**0.6203 ± 0.0049**	0.4613 ± 0.0138	**0.7783 ± 0.0110**
	Concat fusion	**0.5809 ± 0.0063**	0.6186 ± 0.0035	0.4768 ± 0.0102	0.7724 ± 0.0077
	LEARN-TS (Full)	0.5700 ± 0.0100	0.6025 ± 0.0094	**0.5256 ± 0.0064**	0.7765 ± 0.0103
MSL	w/o Obs.	**0.3509 ± 0.0126**	**0.3788 ± 0.0095**	0.3053 ± 0.0064	0.7576 ± 0.0103
	Concat fusion	**0.3509 ± 0.0065**	0.3733 ± 0.0074	**0.3057 ± 0.0048**	**0.7583 ± 0.0097**
	LEARN-TS (Full)	0.3472 ± 0.0184	0.3752 ± 0.0053	0.2951 ± 0.0023	0.7481 ± 0.0282

The full model improves A-PR, V-PR, and R-F1 substantially on SWaT and yields the highest ranking performance on SMD. On PSM, the full model provides a clear improvement in R-F1, although the simpler fusion variants obtain higher ranking metrics. The differences are smaller on MSL, where the three configurations perform comparably. These results indicate that the benefit of gated cross-modal fusion is dataset dependent, with its strongest gains appearing on benchmarks where observation semantics provide complementary information for reconstruction. The observation–window correspondence control is analyzed separately in Figure 3.

C.2 Normality Alignment and Scoring

The macro results in Table 2(b) are computed by first averaging the four datasets equally within each seed and then reporting the mean and sample standard deviation across the five seed-level averages. Table 9 reports the corresponding dataset-wise results.

The ablation compares four configurations. *No normality* removes both alignment training and discrepancy-guided scoring; *Scoring only* applies discrepancy-guided scoring without alignment; *Alignment only* retains alignment training but uses reconstruction evidence alone at inference; and *LEARN-TS (Full)* uses both components.

The joint configuration substantially improves A-PR and V-PR on SWaT and provides consistent ranking gains on SMD, whereas neither component alone reproduces these improvements. Its effects on PSM and MSL are limited or metric-dependent, and the threshold-dependent metrics do not exhibit uniform gains. These results indicate that normality guidance primarily benefits anomaly-score ranking on datasets where reconstruction evidence alone provides insufficient separation, rather than uniformly improving event localization across all benchmarks.

C.3 Normality-Reference Control

Table 10 examines the contribution of semantic content in the normality reference. The random semantic-free variant replaces the normality-prompt embedding with a fixed random reference while preserving the alignment and scoring pipeline.

The semantic reference improves all four metrics on SWaT and SMD, and A-PR, V-PR, and Aff-F1 on MSL, whereas the random reference achieves higher R-F1 on MSL. The two variants perform comparably on PSM.

Together with the alignment-and-scoring ablation, these results distinguish the benefits of the guidance mechanism from those of reference content. Joint alignment and scoring yield substantial ranking gains on SWaT and SMD, while semantic rather than random references provide modest, dataset-dependent improvements. These findings support the utility of semantic references without establishing their necessity.

Table 9: Dataset-wise ablation of normality alignment and discrepancy-guided scoring. Results are reported as mean ± sample standard deviation over five random seeds. The best result in each dataset–metric column is shown in **bold**.

Variant	Align.	Score	A-PR	V-PR	R-F1	Aff-F1
(a) SWaT						
No normality	✗	✗	0.2088 ± 0.0076	0.1748 ± 0.0010	0.2971 ± 0.0611	**0.7407 ± 0.0157**
Scoring only	✗	✓	0.1460 ± 0.0531	0.1530 ± 0.0264	0.2711 ± 0.0635	0.7103 ± 0.0271
Alignment only	✓	✗	0.2004 ± 0.0094	0.1766 ± 0.0008	0.2646 ± 0.0211	0.7299 ± 0.0097
LEARN-TS (Full)	✓	✓	**0.7578 ± 0.0117**	**0.4859 ± 0.0346**	**0.3102 ± 0.0436**	0.7277 ± 0.0069
(b) SMD						
No normality	✗	✗	0.4138 ± 0.0043	0.4895 ± 0.0037	0.3393 ± 0.0041	0.8067 ± 0.0078
Scoring only	✗	✓	0.4514 ± 0.0222	0.5200 ± 0.0158	0.3340 ± 0.0033	0.8088 ± 0.0129
Alignment only	✓	✗	0.4144 ± 0.0031	0.4900 ± 0.0003	**0.3412 ± 0.0019**	0.8099 ± 0.0071
LEARN-TS (Full)	✓	✓	**0.4930 ± 0.0056**	**0.5398 ± 0.0037**	0.3203 ± 0.0033	**0.8121 ± 0.0056**
(c) PSM						
No normality	✗	✗	**0.5743 ± 0.0091**	0.6056 ± 0.0075	0.5194 ± 0.0199	0.7759 ± 0.0115
Scoring only	✗	✓	0.5741 ± 0.0099	**0.6083 ± 0.0079**	0.5006 ± 0.0132	0.7765 ± 0.0129
Alignment only	✓	✗	0.5700 ± 0.0101	0.6025 ± 0.0094	0.5249 ± 0.0119	**0.7808 ± 0.0079**
LEARN-TS (Full)	✓	✓	0.5700 ± 0.0100	0.6025 ± 0.0094	**0.5256 ± 0.0064**	0.7765 ± 0.0103
(d) MSL						
No normality	✗	✗	**0.3491 ± 0.0122**	0.3733 ± 0.0066	0.3058 ± 0.0044	0.7612 ± 0.0077
Scoring only	✗	✓	0.3489 ± 0.0125	0.3732 ± 0.0064	**0.3066 ± 0.0028**	**0.7619 ± 0.0077**
Alignment only	✓	✗	0.3476 ± 0.0162	0.3734 ± 0.0045	0.3010 ± 0.0061	0.7613 ± 0.0087
LEARN-TS (Full)	✓	✓	0.3472 ± 0.0184	**0.3752 ± 0.0053**	0.2951 ± 0.0023	0.7481 ± 0.0282

Table 10: Dataset-wise comparison between random semantic-free and dataset-agnostic semantic normality references. Results are reported as mean ± sample standard deviation over five random seeds. The best result in each dataset–metric column is shown in **bold**; values tied at the displayed precision are both highlighted.

Normality reference	A-PR	V-PR	R-F1	Aff-F1
(a) SWaT				
Random semantic-free	0.7431 ± 0.0426	0.4758 ± 0.0364	0.2933 ± 0.0182	0.7276 ± 0.0017
Dataset-agnostic semantic	**0.7578 ± 0.0117**	**0.4859 ± 0.0346**	**0.3102 ± 0.0436**	**0.7277 ± 0.0069**
(b) SMD				
Random semantic-free	0.4919 ± 0.0052	0.5388 ± 0.0052	0.3153 ± 0.0058	0.8022 ± 0.0143
Dataset-agnostic semantic	**0.4930 ± 0.0056**	**0.5398 ± 0.0037**	**0.3203 ± 0.0033**	**0.8121 ± 0.0056**
(c) PSM				
Random semantic-free	**0.5700 ± 0.0089**	**0.6031 ± 0.0080**	**0.5293 ± 0.0102**	0.7764 ± 0.0090
Dataset-agnostic semantic	**0.5700 ± 0.0100**	0.6025 ± 0.0094	0.5256 ± 0.0064	**0.7765 ± 0.0103**
(d) MSL				
Random semantic-free	0.3390 ± 0.0117	0.3683 ± 0.0028	**0.3004 ± 0.0055**	0.7429 ± 0.0082
Dataset-agnostic semantic	**0.3472 ± 0.0184**	**0.3752 ± 0.0053**	0.2951 ± 0.0023	**0.7481 ± 0.0282**

D Dataset-Wise Sensitivity Analysis

The sensitivity plots in the main paper present macro averaged trends across the four datasets. Tables 11 and 12 report the corresponding dataset-wise results for the three weighting hyperparameters and the number of masked training patches. All sensitivity experiments use a single fixed random seed and are therefore intended to characterize response trends rather than statistical variability. For each weighting hyperparameter, the zero setting removes the corresponding contribution while leaving the remaining architecture and evaluation pipeline unchanged. Bold rows indicate the configurations adopted in the main experiments for the corresponding dataset.

Table 11: Dataset-wise sensitivity to (a) the full-reconstruction weight λ_{full} and (b) the normality-alignment weight λ_{norm}. Results are obtained using a fixed random seed. Bold rows denote the configurations used in the main experiments.

(a) Full-reconstruction weight λ_{full}

Dataset	Value	A-PR	V-PR	R-F1	Aff-F1
SWaT	0.0	0.1766	0.1738	0.2920	0.7381
	0.1	0.7654	0.5004	0.2793	0.7121
	0.2	0.7662	0.5122	0.3293	0.7332
	0.5	**0.7684**	**0.5139**	**0.3784**	**0.7343**
	1.0	0.7665	0.5023	0.2655	0.7276
SMD	0.0	0.4552	0.5277	0.3361	0.8128
	0.1	0.4924	0.5394	0.3308	0.8122
	0.2	0.4918	0.5383	0.3233	0.8151
	0.5	**0.4868**	**0.5376**	**0.3226**	**0.8163**
	1.0	0.4877	0.5367	0.3105	0.7976
PSM	0.0	0.5939	0.6198	0.4891	0.7631
	0.1	0.5693	0.5989	0.5520	0.7873
	0.2	0.5792	0.6106	0.5379	0.7628
	0.5	**0.5793**	**0.6136**	**0.5256**	**0.7659**
	1.0	0.5675	0.6031	0.5294	0.7716
MSL	0.0	0.3122	0.3549	0.3075	0.7566
	0.1	0.3532	0.3765	0.3032	0.7580
	0.2	0.3496	0.3734	0.2952	0.7654
	0.5	**0.3644**	**0.3815**	**0.2957**	**0.7717**
	1.0	0.3285	0.3736	0.2995	0.7692

(b) Normality-alignment weight λ_{norm}

Dataset	Value	A-PR	V-PR	R-F1	Aff-F1
SWaT	0.000	0.1167	0.1502	0.3440	0.7298
	0.005	0.7558	0.4773	0.2599	0.7303
	0.010	0.7635	0.5025	0.2848	0.7114
	0.020	**0.7684**	**0.5139**	**0.3784**	**0.7343**
	0.050	0.7610	0.4963	0.2876	0.7486
SMD	0.000	0.4323	0.5044	0.3310	0.7877
	0.005	0.4924	0.5345	0.3165	0.7926
	0.010	**0.4868**	**0.5376**	**0.3226**	**0.8163**
	0.020	0.4985	0.5402	0.3251	0.8179
	0.050	0.4824	0.5373	0.3180	0.8103
PSM	0.000	0.5845	0.6173	0.4928	0.7742
	0.005	0.5788	0.6134	0.5218	0.7667
	0.010	**0.5793**	**0.6136**	**0.5256**	**0.7659**
	0.020	0.5778	0.6127	0.5282	0.7758
	0.050	0.5708	0.6050	0.5066	0.7873
MSL	0.000	0.3602	0.3800	0.3064	0.7662
	0.005	0.3476	0.3790	0.3044	0.7668
	0.010	**0.3644**	**0.3815**	**0.2957**	**0.7717**
	0.020	0.3592	0.3801	0.3065	0.7705
	0.050	0.3537	0.3810	0.2995	0.7559

D.1 Loss and Scoring Weights

Table 11 reports sensitivity to the full-reconstruction weight λ_{full} and the normality-alignment weight λ_{norm}, while Table 12(a) reports sensitivity to the discrepancy-gating coefficient λ_{gate}. The main experiments use $\lambda_{\text{full}} = 0.5$ for all datasets, $\lambda_{\text{norm}} = 0.02$ for SWaT and 0.01 for SMD, PSM, and MSL, and $\lambda_{\text{gate}} \in \{1.0, 0.05, 0.1, 0.05\}$ for SWaT, SMD, PSM, and MSL, respectively.

SWaT is particularly sensitive to removing the full-window reconstruction objective, normality alignment, or discrepancy-guided modulation, with pronounced decreases in A-PR and V-PR when the corresponding weight is set to zero. Performance is generally more stable across positive weighting values, although the preferred operating range remains dataset dependent. PSM is comparatively

Table 12: Dataset-wise sensitivity to (a) the discrepancy-gating coefficient λ_{gate} and (b) the number of masked training patches. Results are obtained using a fixed random seed. Bold rows denote the configurations used in the main experiments.

(a) Discrepancy-gating coefficient λ_{gate}

Dataset	Value	A-PR	V-PR	R-F1	Aff-F1
SWaT	0.00	0.2132	0.1763	0.2607	0.7387
	0.01	0.7622	0.5137	0.2786	0.7330
	0.05	0.7640	0.5138	0.2861	0.7338
	0.10	0.7670	0.5153	0.3121	0.7337
	0.50	0.7686	0.5138	0.3299	0.7394
	1.00	**0.7684**	**0.5139**	**0.3784**	**0.7343**
SMD	0.00	0.4091	0.4897	0.3422	0.8107
	0.01	0.4825	0.5399	0.3381	0.8154
	0.05	**0.4868**	**0.5376**	**0.3226**	**0.8163**
	0.10	0.4902	0.5369	0.3053	0.8134
	0.50	0.4997	0.5290	0.2740	0.7811
	1.00	0.5040	0.5231	0.2657	0.7673
PSM	0.00	0.5794	0.6136	0.5198	0.7742
	0.01	0.5794	0.6136	0.5198	0.7743
	0.05	0.5794	0.6136	0.5202	0.7749
	0.10	**0.5793**	**0.6136**	**0.5256**	**0.7659**
	0.50	0.5791	0.6136	0.5035	0.7713
	1.00	0.5789	0.6137	0.5444	0.7628
MSL	0.00	0.3619	0.3772	0.3055	0.7705
	0.01	0.3627	0.3782	0.3087	0.7709
	0.05	**0.3644**	**0.3815**	**0.2957**	**0.7717**
	0.10	0.3662	0.3820	0.3072	0.7759
	0.50	0.3656	0.3807	0.2853	0.7601
	1.00	0.3590	0.3695	0.2737	0.7201

(b) Number of masked training patches

Dataset	# Mask	A-PR	V-PR	R-F1	Aff-F1
SWaT	**1**	**0.7684**	**0.5139**	**0.3784**	**0.7343**
	2	0.7423	0.4644	0.2901	0.7110
	3	0.7456	0.4769	0.2673	0.7108
SMD	**1**	**0.4868**	**0.5376**	**0.3226**	**0.8163**
	2	0.4901	0.5342	0.3147	0.8004
	3	0.4998	0.5468	0.3205	0.8015
PSM	**1**	**0.5793**	**0.6136**	**0.5256**	**0.7659**
	2	0.5687	0.6040	0.5158	0.7752
	3	0.5703	0.6043	0.5188	0.7726
MSL	**1**	**0.3644**	**0.3815**	**0.2957**	**0.7717**
	2	0.3656	0.3804	0.3040	0.7603
	3	0.3626	0.3824	0.3079	0.7655

insensitive to λ_{gate} in A-PR and V-PR, whereas SMD and MSL exhibit clearer trade-offs between ranking and threshold-dependent event metrics as the gate strength increases.

D.2 Training Masking

The main configuration masks one of the eight temporal patches during each training forward pass. Figure 5(d) presents the corresponding four-dataset macro trend, while Table 12(b) provides the complete dataset-wise results. Masking additional patches generally degrades macro performance, with the strongest effect observed on SWaT. Although SMD and MSL exhibit small metric-dependent improvements under heavier masking, these gains are not consistent across datasets or metrics. Single-patch masking therefore provides the most balanced overall behavior while matching the exhaustive single-patch masking scheme used at inference.

Table 13: Backbone efficiency and detection performance on four datasets. Detection results are mean ± sample standard deviation over five seeds. Embedding throughput measures prompt encoding; scoring throughput excludes text encoding and uses cached embeddings. Bold indicates the best result within each dataset and column.

(a) Computational efficiency

Dataset	Backbone	Trainable Params ↓	Peak Train GPU (GiB) ↓	Embed. (prompts/s) ↑	Scoring (windows/s) ↑
SWaT	GPT-2	**41.16M**	**0.795**	**80.90**	**1111.46**
	Qwen2.5-1.5B	41.75M	0.811	38.81	932.19
	Llama-3.2-1B	42.14M	0.822	53.18	958.34
PSM	GPT-2	**40.79M**	**0.792**	**80.15**	**1194.91**
	Qwen2.5-1.5B	41.38M	0.803	40.17	1152.70
	Llama-3.2-1B	41.77M	0.814	55.89	1097.99
MSL	GPT-2	**40.96M**	**0.794**	**80.12**	**1212.53**
	Qwen2.5-1.5B	41.55M	0.805	40.33	1168.39
	Llama-3.2-1B	41.95M	0.815	56.41	1116.43
SMD	GPT-2	**40.84–41.06M**	**0.794**	**82.11**	**1185.91**
	Qwen2.5-1.5B	41.43–41.65M	0.805	40.29	1098.63
	Llama-3.2-1B	41.82–42.04M	0.816	56.14	1108.09

(b) Anomaly detection performance

Dataset	Backbone	A-PR	V-PR	R-F1	Aff-F1
SWaT	GPT-2	0.7578 ± 0.0117	0.4859 ± 0.0346	**0.3102 ± 0.0436**	**0.7277 ± 0.0069**
	Qwen2.5-1.5B	0.7540 ± 0.0376	0.4883 ± 0.0320	0.2677 ± 0.0237	0.7204 ± 0.0078
	Llama-3.2-1B	**0.7654 ± 0.0050**	**0.4955 ± 0.0117**	0.2626 ± 0.0191	0.7199 ± 0.0166
PSM	GPT-2	0.5700 ± 0.0100	0.6025 ± 0.0094	0.5256 ± 0.0064	0.7765 ± 0.0103
	Qwen2.5-1.5B	**0.5748 ± 0.0088**	**0.6029 ± 0.0072**	0.5136 ± 0.0252	0.7783 ± 0.0145
	Llama-3.2-1B	0.5716 ± 0.0118	0.6027 ± 0.0096	**0.5266 ± 0.0096**	**0.7787 ± 0.0056**
MSL	GPT-2	**0.3472 ± 0.0184**	**0.3752 ± 0.0053**	0.2951 ± 0.0023	0.7481 ± 0.0282
	Qwen2.5-1.5B	0.3421 ± 0.0077	0.3717 ± 0.0107	0.3005 ± 0.0035	0.7552 ± 0.0246
	Llama-3.2-1B	0.3361 ± 0.0075	0.3688 ± 0.0046	**0.3006 ± 0.0057**	**0.7620 ± 0.0045**
SMD	GPT-2	**0.4930 ± 0.0056**	**0.5398 ± 0.0037**	**0.3203 ± 0.0033**	0.8121 ± 0.0056
	Qwen2.5-1.5B	0.4874 ± 0.0121	0.5342 ± 0.0030	0.3173 ± 0.0030	**0.8125 ± 0.0058**
	Llama-3.2-1B	0.4897 ± 0.0070	0.5341 ± 0.0047	0.3150 ± 0.0069	0.8105 ± 0.0123

E Language-Backbone Robustness and Efficiency

We evaluate whether LEARN-TS remains effective across different frozen language backbones while comparing their computational costs. Table 13 reports efficiency and anomaly detection performance across all four datasets for GPT-2, Qwen2.5-1.5B (Yang et al., 2024), and Llama-3.2-1B (Meta, 2024).

As shown in Table 13, the three frozen language backbones yield broadly comparable ranking performance across all four datasets. GPT-2 achieves the highest embedding and cached scoring throughput throughout, while detection performance varies by dataset and metric. No backbone consistently outperforms the others, and larger backbones do not yield uniform gains.

All efficiency measurements were performed on an NVIDIA L40S GPU with 48 GB of GPU memory. For SMD, detection metrics are macro-averaged across 28 machines within each seed. Parameter counts report the range across machines, peak training GPU memory reports the maximum, and scoring throughput is computed as total windows divided by total scoring time.

F Prompt Templates

LEARN-TS uses a window-specific observation prompt and a fixed, dataset-agnostic normality prompt with distinct semantic roles. Both are encoded by the same frozen language model and use neither anomaly labels nor test-set-wide statistics. Figure 6 shows the fixed normality prompt, while

Dataset-Agnostic Normality Prompt

[System / Domain]
The monitored system is represented by multivariate time-series signals whose exact domain semantics are intentionally unspecified.

[Normal-operation assumptions]
Under normal operation, related variables should evolve coherently, preserve plausible dependencies, and remain consistent with recent temporal context.

[Cross-variable constraints]
[R1] Changes in one variable should be supported by plausible changes in related variables.
[R2] Correlated variables should exhibit coherent co-movement rather than isolated persistent excursions.
[R3] System-state changes should produce temporally consistent responses across affected variables.
[R4] Persistent deviations should be explainable by a coherent operating-state transition.

[Temporal constraints]
[R5] Short transients may occur, but they should follow a plausible temporal evolution and recovery.
[R6] Abrupt isolated spikes or drops without supporting context violate normal temporal consistency.
[R7] Normal trajectories should preserve local continuity while allowing coherent regime transitions.

[Expected behavior]
A normal window should be temporally coherent and mutually consistent across related variables.

[Output focus] Return a latent representation emphasizing generic normal-operation constraints, cross-variable consistency, temporal coherence, and plausible state transitions.

Figure 6: Dataset-agnostic normality prompt used in LEARN-TS. The prompt is fixed before training and shared unchanged across all datasets and input windows.

Figure 7 illustrates an abridged observation template with placeholders for window-dependent fields. LEARN-TS extracts token-level hidden states rather than generating or parsing textual responses.

F.1 Normality Reference Construction

LEARN-TS uses the dataset-agnostic semantic reference as its default normality reference. To determine whether normality guidance benefits from semantic content rather than merely from an additional reference input, we compare it with a random semantic-free reference of the same input shape.

F.1.1 Dataset-Agnostic Semantic Reference

The normality prompt is fixed before training and shared unchanged across SWaT, SMD, PSM, and MSL. It describes broad properties of normal multivariate temporal behavior without using dataset-specific sensor identities, operating rules, anomaly labels, or window-specific observations.

Figure 6 shows the complete prompt used to construct the shared normality reference. The prompt encodes generic assumptions about cross variable consistency, temporal coherence, plausible state transitions, and context-supported recovery. Because the same prompt is used for every dataset and input window, its language model representation serves as a window-independent semantic reference rather than as context associated with a particular observation.

F.1.2 Random Semantic-Free Reference

The random semantic-free reference evaluates whether normality guidance benefits from semantic normality content rather than merely from an additional reference input. For each experimental run, we generate a seeded random tensor $\mathbf{E}_{\text{rand}} \in \mathbb{R}^{T_n \times d_e}$ with the same shape as the frozen normality-prompt embedding $\mathbf{E}_{\text{norm}} = E_{\text{text}}(q^{\text{norm}})$. The random tensor replaces the frozen language model

Abridged SWaT Observation Prompt Template

[Dataset]
SWaT: A scaled water treatment system with multi-stage processing, including tank levels, flow sensors, motorized valves, and pumps.

[Task]
Given the observed history and partially masked current window, reconstruct the masked segments using the observed temporal context and the operational consistency rules described below.

[Window statistics]
Trend: [window trend]
Volatility: [window volatility]
Exact values, timestamps, explicit change-point locations, and local deviation magnitudes are omitted.

[Sensor-group summary]
Group 1 (level and storage tanks): [group 1 trend] and [group 1 volatility]
Group 2 (flow and routing pipes): [group 2 trend] and [group 2 volatility]
...

[Operational context]
Apply dataset-level normal-process constraints without anomaly-specific wording. Emphasize consistency between actuator states, sensor responses, and local mass-balance relationships.

[Logical rules]
[R1] If a tank level is above a high threshold, the inflow valve should close or the downstream pump should activate within a short delay.
[R2] If a pump is on, the corresponding flow sensor should exhibit a positive response after a short delay.
...

[Patch-local observations]
Patches [patch indices]: [patch-group trend] and [patch-group volatility]
Patches [patch indices]: [patch-group trend] and [patch-group volatility]
...

[Reconstruction target]
Maintain local temporal continuity and compatibility with the qualitative operational rules.

[Output focus]
Emphasize temporal coherence, operational consistency, and qualitative cross-sensor dependencies.

Figure 7: Abridged SWaT observation prompt template. Blue placeholders denote window-dependent descriptors and patch-index lists; dataset-level descriptions and rules remain fixed. Exact measurements and explicit anomaly labels are excluded. Repeated entries are omitted for readability.

embedding before the shared text projection:

$$\mathbf{H}_{\text{rand}} = \phi(\mathbf{E}_{\text{rand}}) \in \mathbb{R}^{T_n \times d}. \tag{12}$$

The random input is sampled once for each experimental seed and remains fixed throughout training and inference. The shared projection and all downstream normality-alignment and discrepancy-scoring components are otherwise identical to the dataset-agnostic semantic-reference configuration. This control therefore preserves the input shape and downstream architecture while removing language-derived normality content.

F.2 Window-Specific Observation Prompt

The observation prompt combines a common template structure with fixed dataset-specific profiles and deterministic window descriptors. Within each dataset, the same profile and descriptor-construction procedure are used for training, validation, and test windows. Unlike the dataset-agnostic normality prompt, observation prompts include dataset-level system descriptions, sensor-group definitions, and qualitative operational rules alongside window-derived descriptors. They require no temporally paired external text and use no anomaly labels. Exact numerical measurements,

timestamps, explicit change-point locations, and local deviation magnitudes are excluded. Figure 7 shows an abridged SWaT observation prompt template, retaining all information types used by the implementation. The same prompt structure is used for the other datasets, with their corresponding system descriptions, sensor groups, operational rules, and window-derived descriptors.

The descriptors are computed from the fully observed window before numerical masking. Consequently, they may retain coarse qualitative information about a subsequently masked patch, such as its trend or volatility, without exposing its exact numerical target. The masking objective therefore suppresses direct numerical identity mapping rather than defining information-free missing-value imputation.

Prompt structure and construction. Each prompt contains: (i) a dataset-level system description and reconstruction task; (ii) global trend and volatility descriptors; (iii) sensor-group-level temporal summaries; (iv) qualitative operational context and logical rules; (v) patch-local trend and variability descriptors; and (vi) the reconstruction objective.

All descriptors are computed deterministically using statistics from the current window. Volatility levels are assigned using fixed normalized cutoffs of 0.3 and 0.8, and trend categories follow the deterministic rule implemented in the released code. Dataset-level descriptions are based on the original dataset papers (Goh et al., 2017; Su et al., 2019; Abdulaal et al., 2021; Hundman et al., 2018). Where available, accompanying public sensor or channel metadata are additionally used to define channel-group keywords. For SWaT, qualitative operational rules are manually specified based on public descriptions of the testbed and its sensor metadata, covering coarse relationships among tank levels, valve and pump states, flow responses, and mass-balance consistency. Channel groups are constructed by predefined keyword matching, with unmatched channels assigned to auxiliary groups in their original order. All dataset profiles are specified independently of anomaly labels and test-set events and remain fixed across windows. Complete keyword lists, operational rules, descriptor formulas, and exception-handling procedures are provided in the released code.

Text encoding. The prompt is encoded once by the frozen language model, and its final-layer token hidden states are cached for downstream use. Padding tokens are excluded from representation pooling. The tokenizer configuration and input-length handling are provided in the released code.