CARE: A CASCADED FRAMEWORK FOR EFFICIENT AND RELIABLE TIME SERIES ANOMALY DETECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Deep learning models have achieved state-of-the-art performance in time series anomaly detection, yet their complex architectures incur substantial inference overhead. Existing methods uniformly apply costly inference to all inputs, ignoring that anomalies are inherently scarce and most temporal data exhibit predictable patterns. To address this inefficiency, we propose CARE, a model-agnostic cascaded inference framework that pairs a Lightweight Pre-filter Model (LPM) with a high-capacity Complex Detection Model (CDM). The LPM rapidly filters high-confidence normal samples using a Residual MLP-based AutoEncoder and a Normality-Conditioned Gating mechanism. The gating mechanism incorporates Structure Attention and Risk-State Memory to capture channel-wise anomaly contributions and temporal risk evolution. A budget-aware gating objective learns normality confidence rankings while aligning CDM allocation with the available computational budget. Extensive experiments across eight real-world benchmarks demonstrate that CARE effectively isolates high-confidence normal samples. By prioritizing low-confidence samples for CDM processing, our framework achieves inference speedups of up to $4.1\times$ compared to the most accurate SOTA approaches, while maintaining competitive detection quality. The source code is available at `https://anonymous.4open.science/r/CCARE-C711`.

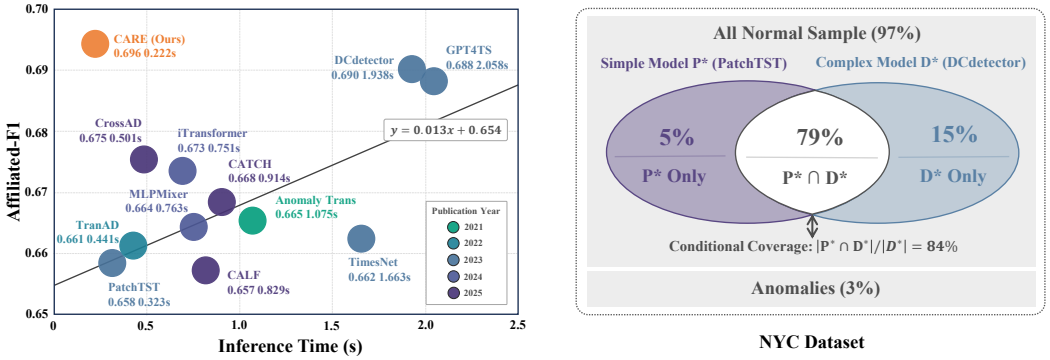
1 INTRODUCTION

Time series anomalies serve as critical early indicators of underlying faults and emerging risks in modern cyber-physical systems (Zamanzadeh Darban et al., 2024; Hawkins, 1980; Blázquez-García et al., 2021). Consequently, Time Series Anomaly Detection (TSAD) has emerged as a crucial research frontier in data mining, driving essential real-world applications such as fraud detection (Yu et al., 2024), healthcare monitoring (Roy et al., 2023), and urban management (Dou et al., 2024). Although traditional methods (Breunig et al., 2000; Liu et al., 2008; Schölkopf et al., 1999) laid the groundwork for identifying basic outliers, they generally struggle to capture the highly complex, non-linear temporal dependencies pervasive across real-world data streams (Li et al., 2025b).

Deep learning has since transformed TSAD by enabling automated feature extraction and the modeling of intricate temporal dynamics. To capture the complex dynamics of real-world data, recent methodologies have increasingly relied on sophisticated, high-capacity architectures, such as Transformer-based models for long-term dependencies (Dai et al., 2024; Li et al., 2025a), contrastive learning for latent normal patterns (Tang et al., 2024; Zhuang et al., 2024), and frequency-domain analysis for multi-view relationships (Fang et al., 2024; Wu et al., 2024). However, this continuous pursuit of superior detection quality via architectural complexity comes at a steep computational price. Consequently, deploying these deep TSAD models in practical, efficiency-sensitive scenarios reveals critical limitations.

Limitation 1: Severe inference bottlenecks in quality-leading detection models. While increasing model complexity generally yields superior detection quality, it inherently introduces substantial inference overhead. As depicted in Figure 1a, state-of-the-art (SOTA) methods (e.g., DCdetector and GPT4TS) exhibit prohibitively high inference costs (Qiu et al., 2025). Conversely, lightweight models (e.g., TranAD and PatchTST) offer low-latency inference but sacrifice detection accuracy. Unfortunately, TSAD methodologies that explicitly improve inference efficiency while preserving

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.



(a) Model performance comparison on NYC.

(b) Normal coverage overlap between models.

Figure 1: Motivation of the proposed CARE framework. (a) The trade-off between detection performance and inference cost among existing methods. CARE explicitly bridges this gap, delivering state-of-the-art accuracy with substantially lower latency. (b) Over 84% of the normal samples identified by the high-capacity model (DCdetector) are also successfully captured by the lightweight model (PatchTST), indicating that most instances can be identified by simple models.

detection quality remain insufficiently explored, leaving a critical gap in deploying these models at scale.

Limitation 2: Uniform complex inference on predominantly normal patterns. Due to the inherent scarcity of anomalous events (Aggarwal, 2016; Schmidl et al., 2022), real-world temporal data is dominated by normal samples. A substantial portion of these normal instances exhibits stable, predictable patterns that can be effectively captured by relatively simple architectures. As illustrated in Figure 1b, simple architectures (e.g., PatchTST) can accurately identify the vast majority of normal temporal patterns alongside high-capacity models, indicating that most data points are sufficiently regular to be processed without complex inference. Nevertheless, most modern deep TSAD backbones are still evaluated through uniform inference, applying their complete models to every input window. This forces straightforward normal samples to undergo costly and redundant inference procedures, leading to unnecessary overhead.

To address these limitations, we propose a **CA**scaded framework for **R**eliable and **E**fficient time series anomaly detection (CARE), which rethinks the inference process by seamlessly integrating a Lightweight Pre-filter Model (LPM) with a high-capacity Complex Detection Model (CDM). Operating as an efficient front-end, the LPM employs a Residual MLP-based AutoEncoder coupled with a Normality-Conditioned Gating Module. This gating mechanism rapidly assesses the normality confidence of incoming temporal windows: high-confidence normal instances are retained by the lightweight LPM, while low-confidence samples are prioritized for refined CDM evaluation. To ensure reliable routing without compromising detection quality, the gating network combines Structure Attention with Risk-State Memory to capture both channel-wise anomaly contributions and temporal risk evolution across consecutive windows. It is further optimized with a budget-aware gating objective that learns routing priorities under a specified computational budget. The proposed approach thus mitigates the conventional performance-efficiency trade-off, enabling efficient and reliable anomaly detection. Our main contributions are:

- We introduce CARE, a selective cascaded inference framework for efficient time series anomaly detection. By rethinking the uniform complex inference paradigm for TSAD, CARE serves as a flexible, model-agnostic architecture that seamlessly integrates with, and substantially accelerates, existing high-capacity anomaly detectors.
- We design a Lightweight Pre-filter Model (LPM) equipped with a Normality-Conditioned Gating (NCG) mechanism that models both channel-wise structural contributions and temporal risk evolution across consecutive windows. Furthermore, we introduce a budget-aware gating objective that learns routing priorities from label-free CDM supervision, enabling efficient and reliable selective inference under specified computational budgets.

- Extensive experiments across eight diverse real-world benchmarks demonstrate that CARE effectively isolates straightforward normal samples, breaking the traditional performance-efficiency bottleneck. Notably, CARE achieves inference speedups of up to $4.1\times$ while maintaining, and in several cases even surpassing, state-of-the-art detection quality.

2 Related Work

Recent advancements in deep learning have significantly improved Time Series Anomaly Detection (TSAD). State-of-the-art methodologies, including forecasting-based, reconstruction-based, and representation-based models, as well as recent adaptations of Large Language Models (LLMs), have achieved remarkable detection accuracy (Tang et al., 2024; Fang et al., 2024; Liu et al., 2024; Zhou et al., 2023). However, this continuous pursuit of accuracy frequently relies on highly complex architectures (e.g., Transformers or multi-view fusions). While effective, these methods incur prohibitive inference latency, rendering them computationally expensive and impractical for latency-sensitive deployments.

To balance performance and efficiency, cascaded inference models have been extensively explored in domains such as computer vision and natural language processing (Han et al., 2021; Chen et al., 2024b). The core principle is to process straightforward samples with lightweight models while reserving high-capacity architectures for complex cases. Similar strategies exist in anomaly detection: STARE (Yoon et al., 2020b) skips density updates in stationary stream regions, while Chang et al. (2025) develops a GMM and VAE cascade under inference cost constraints for edge TSAD. CARE instead targets label-free routing for pretrained deep TSAD models whose raw anomaly scores are not calibrated across models. It learns an efficient gate from relative CDM score rankings and structural and temporal risk, retaining confident normal samples at the LPM and allocating CDM computation to uncertain samples under a specified budget.

Due to space constraints, an extended and comprehensive review of the related literature is provided in Appendix A.

3 Methodology

Given an unlabeled time series $X = \{x_1, x_2, \ldots, x_T\} \in \mathbb{R}^{T\times C}$, where T denotes the sequence length and C represents the number of feature channels, the objective of TSAD is to assign a real-valued anomaly score $AS(x_t) \in [0, 1]$ to each observation, with higher scores indicating a greater likelihood of being anomalous.

Cascaded Time Series Anomaly Detection. Given a pre-trained, high-capacity, complex anomaly detection model CDM along with its inferred anomaly scores AS_c^{train} on the training set X^{train}, we introduce a lightweight pre-filter model LPM, which is trained using X^{train} and guided by the prior knowledge AS_c^{train}. The objective of the LPM is to generate a coarse-grained anomaly score $AS_l(x_t) \in [0, 1]$, derived from the reconstruction error of x_t, and estimate a normality confidence score $P(x_t) \in [0, 1]$ for any observation $x_t \in X^{test}$.

During inference, the evaluation path of x_t is dynamically routed based on $P(x_t)$. A high $P(x_t)$ indicates a high-confidence normal sample, allowing the final anomaly score $AS(x_t)$ to be derived directly from AS_l. Conversely, uncertain samples are forwarded to the CDM for a refined, albeit computationally more expensive, evaluation.

3.1 Overall Architecture of CARE

The proposed CARE framework cascades a Lightweight Pre-filter Model (LPM) with a Complex Detection Model (CDM). The LPM rapidly identifies predictable normal samples and routes uncertain instances to the CDM, significantly enhancing efficiency without compromising detection quality. Since CARE aims to accelerate existing high-capacity detectors, it assumes only that the backend CDM provides deterministic continuous anomaly scores, a standard property of most TSAD models. Consequently, we focus on the architectural design of the LPM, which consists of three principal components. The overall CARE framework is illustrated in Figure 2.

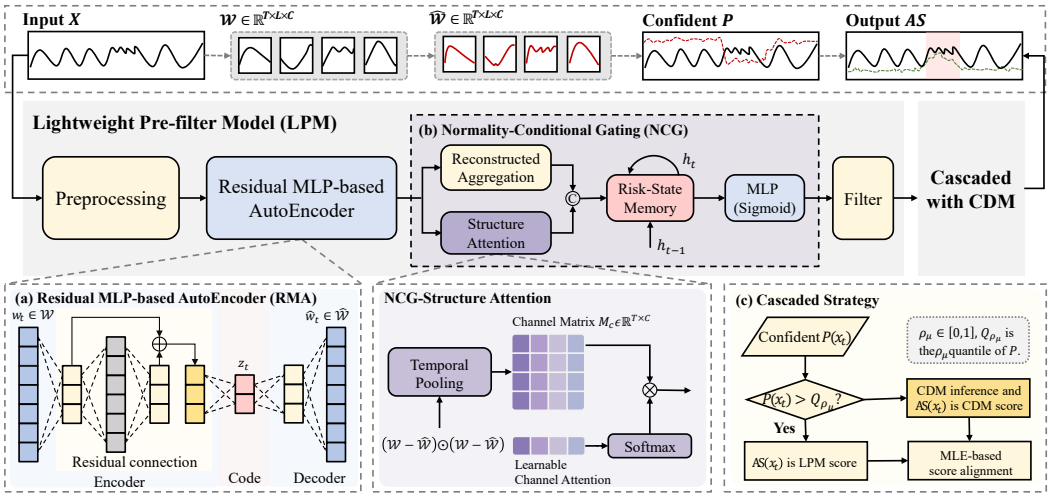


Figure 2: The overall architecture of the proposed CARE framework. (a) The Residual AutoEncoder performs lightweight and rapid input reconstruction. (b) The Normality-Conditioned Gating network combines Structure Attention and Risk-State Memory with reconstruction deviations to estimate a stateful normality confidence score for each window. (c) The cascaded inference strategy leverages the confidence score to bypass trivial normal samples and allocate the available CDM computation to uncertain samples, improving inference efficiency while preserving detection reliability.

Time Series Preprocessing. We apply standard Z-score normalization to mitigate distribution shifts (Li et al., 2025b). The normalized sequence is then partitioned into fixed-length sliding windows (using prefix padding for initial steps) to serve as model inputs. Specifically, the input at time step t is defined as $w_t \in \mathbb{R}^{L\times C}$, encompassing the most recent L observations of the time series. This yields the complete window set $\mathcal{W} = \{w_1, \ldots, w_T\} \in \mathbb{R}^{T\times L\times C}$ for all T time steps.

Residual MLP-based AutoEncoder (RMA). To enable efficient anomaly inference, the LPM adopts an MLP-based architecture for time series reconstruction. The preprocessed windows $\mathcal{W} = \{w_1, \ldots, w_T\}$ are fed into the encoder to capture underlying normal patterns, where residual connections are introduced to enhance feature representation. The resulting latent embeddings are subsequently passed to a decoder to reconstruct the original sequence.

Normality-Conditioned Gating (NCG). NCG is designed to identify high-confidence normal samples and prevent them from undergoing unnecessary complex inference. To achieve this, we incorporate a risk-state gating network equipped with a Structure Attention mechanism. This module extracts multi-granular reconstruction errors to characterize deviations from established normal patterns, while Structure Attention explicitly captures channel-wise anomaly contributions. Subsequently, these deviation-based and structural features are fused and further integrated through Risk-State Memory to model temporal risk evolution across consecutive windows. Ultimately, the resulting stateful representation is projected into a normalized confidence score through the gating network, which directly guides the downstream routing decision in the cascaded framework.

3.2 Residual MLP-based AutoEncoder (RMA)

To enable rapid and computationally efficient anomaly inference, the RMA adopts a Multilayer Perceptron (MLP) autoencoder architecture, deliberately avoiding costly self-attention mechanisms. Specifically, the encoder maps flattened sliding windows through an initial MLP layer, followed by a residual block to effectively capture complex temporal dynamics without heavy parameterization. The decoder then reconstructs the sequence to obtain the reconstructed window set $\hat{\mathcal{W}} \in \mathbb{R}^{T\times L\times C}$ via two consecutive MLP layers and a reshaping operation. The formal definition of the RMA is provided in Appendix B.

3.3 NORMALITY-CONDITIONED GATING (NCG)

As the core decision component of the LPM, the Normality-Conditioned Gating (NCG) identifies high-confidence normal samples using the reconstruction outputs from Section 3.2. To improve routing reliability, NCG combines Structure Attention with Risk-State Memory to jointly model structural and temporal risk information.

Structure Attention. In practice, sensor channels exhibit varying sensitivity to anomalies, with certain channels inherently more fault-prone (Tuli et al., 2022). However, standard MLP-based feature fusion does not explicitly expose the contribution of each channel to the routing decision. To address this, CARE introduces a Structure Attention mechanism that explicitly quantifies channel-wise anomaly contributions.

Specifically, for each window w_t, we compute the squared reconstruction deviation and apply temporal average pooling to obtain channel-wise anomaly contributions. To adaptively scale these contributions, we introduce a learnable **channel attention vector** $\mathbf{v} \in \mathbb{R}^C$, normalized via a softmax function. The channel-wise deviations are then weighted and aggregated to yield the **structural anomaly feature** $\mathbf{f}_{\mathrm{st},t}$, which serves as an auxiliary signal for the subsequent gating network:

$$\mathbf{f}_{\mathrm{st},t} = \mathrm{TemporalPool}\left((w_t - \hat{w}_t) \odot (w_t - \hat{w}_t)\right) \cdot \mathrm{softmax}(\mathbf{v}) \tag{1}$$

where $\mathbf{f}_{\mathrm{st},t} \in \mathbb{R}^1$ denotes the structural anomaly feature of window w_t, $\odot$ denotes the Hadamard product, and $\mathrm{TemporalPool}(\cdot)$ performs average pooling along the window-length dimension L.

Risk-State Gating Network. In the final stage of the LPM, NCG estimates normality confidence by combining current reconstruction evidence with historical risk states. Specifically, we extract window-level and point-level reconstruction features, $\mathbf{f}_{\mathrm{win},t}$ and $\mathbf{f}_{\mathrm{pot},t}$, and concatenate them with the structural anomaly feature $\mathbf{f}_{\mathrm{st},t}$ from Structure Attention to form the current risk representation:

$$\begin{aligned}
\mathbf{f}_{\mathrm{win},t} &= \mathrm{WindowPool}\left((w_t - \hat{w}_t) \odot (w_t - \hat{w}_t)\right), \\
\mathbf{f}_{\mathrm{pot},t} &= \mathrm{ChannelPool}\left((w_{t,L,:} - \hat{w}_{t,L,:}) \odot (w_{t,L,:} - \hat{w}_{t,L,:})\right), \\
\mathbf{f}_t &= \mathrm{Concat}\left(\mathbf{f}_{\mathrm{win},t}, \mathbf{f}_{\mathrm{pot},t}, \mathbf{f}_{\mathrm{st},t}\right)
\end{aligned} \tag{2}$$

where $\mathbf{f}_{\mathrm{win},t}, \mathbf{f}_{\mathrm{pot},t} \in \mathbb{R}^1$, while $\mathbf{f}_t \in \mathbb{R}^3$ denotes their concatenated risk representation. $\mathrm{WindowPool}(\cdot)$ denotes the global average pooling across both the temporal and channel dimensions, while $\mathrm{ChannelPool}(\cdot)$ computes the average over channels at any time step.

Risk-State Memory. Rather than making an independent gating decision for each window, Risk-State Memory recursively integrates the current risk representation $\mathbf{f}_t$ with the preceding state $\mathbf{h}_{t-1}$, thereby modeling the temporal evolution of anomaly risk across consecutive windows. The resulting risk state is then projected into the normality confidence:

$$\mathbf{h}_t = \mathrm{GRUCell}\left(\mathbf{f}_t, \mathbf{h}_{t-1}\right), \quad P(x_t) = 1 - \sigma\left(\mathbf{W}_g \mathbf{h}_t + b_g\right) \tag{3}$$

where $\mathbf{h}_t \in \mathbb{R}^{D_g}$ is the recurrent risk state, and D_g denotes the hidden dimension of the gating network, and $\mathbf{W}_g \in \mathbb{R}^{1 \times D_g}$ and $b_g \in \mathbb{R}$ are learnable parameters.

3.4 CASCADED MODEL TRAINING AND INFERENCE

This section describes the training of the Lightweight Pre-filter Model (LPM) and the cascaded inference process of CARE. The pseudocode of CARE is provided in Appendix D.

Two-stage LPM Training. To prevent the gating objective from distorting the underlying temporal representations (i.e., semantic drift), we decouple the optimization process into two stages.

In Stage 1, the RMA is optimized solely with the sequence reconstruction loss. This encourages the RMA to capture the dominant temporal patterns in the training data independently of the downstream routing objective. The loss function is formulated as:

$$\mathcal{L}_{recon} = \|\mathcal{W} - \hat{\mathcal{W}}\|_2^2 \tag{4}$$

In Stage 2, both the RMA trained in Stage 1 and the pre-trained CDM are frozen to provide stable prior knowledge. The NCG is optimized over chronologically ordered window chunks so that Risk-State Memory can capture temporal risk evolution across consecutive observations.

Computational Budget. To coordinate routing decisions under different resource constraints, we introduce a user-specified normalized computational budget μ. Let C_L denote the cost of the always executed lightweight stage and C_C the additional cost of invoking the CDM. For $\mu \in [C_L/(C_L + C_C), 1]$, we define the corresponding target CDM allocation ratio $\rho_\mu \in [0, 1]$ as

$$\rho_\mu = \frac{\mu(C_L + C_C) - C_L}{C_C}, \tag{5}$$

which specifies the target fraction of samples assigned to the CDM and serves as the shared budget reference for training and inference.

To train the gating network, we use the anomaly scores produced by the CDM on the training set (AS_c^{train}) as supervisory signals. A fundamental challenge in unsupervised TSAD, however, is that raw anomaly scores lack inherent physical interpretability and exhibit severe scale variations across datasets. Consequently, directly optimizing the network NCG against absolute AS_c values is prone to yielding unreliable routing decisions. To circumvent this, we exploit the robust relative ranking preserved within AS_c. Guided by the intrinsic prior that normal patterns dominate the data distribution, we partition the training samples using the budget-induced risk threshold $Q_{\text{risk}} = Q_{1-\rho_\mu}(AS_c)$. Samples below this threshold are regarded as high-confidence safe instances, whereas those exceeding it are treated as high-risk. The gating optimization is formalized as a Quantile-Anchored Margin Loss, composed of the following principal components:

Bipartite Loss. This component explicitly enforces a margin-based separation, ensuring that designated safe samples consistently yield higher normality confidences than their high-risk counterparts. By penalizing ranking violations, it robustly preserves the global relative ordering between the two disjoint subsets. Formally, it is defined as:

$$\mathcal{L}_{bipartite} = \mathbb{E}_{x_i, x_j \sim X^{train}} \Big[\mathbb{I}_{\text{risk}}(x_i)(1 - \mathbb{I}_{\text{risk}}(x_j)) \cdot \max\big(0, \xi_1 - (P(x_j) - P(x_i))\big) \Big] \tag{6}$$

where $\mathbb{I}_{\text{risk}}(x_t) = \mathbb{I}\big(AS_c(x_t) > Q_{\text{risk}}\big)$ denotes the high-risk indicator. ξ_1 is a predefined margin hyperparameter controlling the separation between the two subsets.

Boundary Loss. To resolve ambiguity near the risk boundary, we enforce localized hinge constraints on samples immediately adjacent to Q_{risk}. This explicitly penalizes incorrect local orderings within the most uncertain region. It is expressed as:

$$\mathcal{L}_{boundary} = \mathbb{E}_{x_i, x_j \sim X^{train}} \Big[\mathbb{I}^{\text{high}}_{\text{risk}}(x_i) \mathbb{I}^{\text{low}}_{\text{risk}}(x_j) \cdot \max\big(0, \xi_2 - (P(x_j) - P(x_i))\big) \Big] \tag{7}$$

where $\mathbb{I}^{\text{high}}_{\text{risk}}(x_t) = \mathbb{I}\big(Q_{\text{risk}} < AS_c(x_t) \le Q_{\text{risk}} + w\big)$ and $\mathbb{I}^{\text{low}}_{\text{risk}}(x_t) = \mathbb{I}\big(Q_{\text{risk}} - w \le AS_c(x_t) \le Q_{\text{risk}}\big)$ denote the samples immediately above and below the risk boundary, respectively. Here, w denotes the predefined boundary width, and ξ_2 is the margin hyperparameter for this localized constraint.

Budget-Alignment Loss. The ranking objectives above encourage reliable separation between low- and high-risk windows, but do not explicitly align the expected CDM allocation with the available computational budget. We therefore introduce a budget-alignment loss that matches the expected normalized inference cost to the budget μ:

$$\mathcal{L}_{\text{budget}} = \left[\frac{C_L + C_C \frac{1}{N} \sum_{t=1}^{N} (1 - P(x_t))}{C_L + C_C} - \mu \right]^2 \tag{8}$$

Combining these components, the overall training objective for the gating network is formalized as:

$$\mathcal{L}_{\text{gate}} = \mathcal{L}_{\text{bipartite}} + \lambda \mathcal{L}_{\text{boundary}} + \alpha \mathcal{L}_{\text{budget}} \tag{9}$$

Cascaded CARE Inference. As illustrated in Figure 2, CARE processes each inference sequence while maintaining the recurrent risk state. For each observation x_t, the NCG produces a normality confidence score $P(x_t)$. Given the target CDM allocation ratio ρ_μ induced by the computational budget μ, the routing threshold Q_{ρ_μ} is set as the ρ_μ quantile of P. The framework then partitions the input into two mutually exclusive subsets: a high-confidence normal set $\mathcal{N}$ and an uncertain set $\mathcal{U}$. Specifically, if $P(x_t) \ge Q_{\rho_\mu}$, $x_t \in \mathcal{N}$ and bypasses the CDM; conversely, if $P(x_t) < Q_{\rho_\mu}$, $x_t \in \mathcal{U}$ and is routed to the CDM for refined evaluation.

Since the LPM and CDM anomaly scores, AS_l and AS_c, lie on heterogeneous scales, we align them before aggregation, as detailed in Appendix C.

4 EXPERIMENTS

Datasets We evaluate the performance of CARE on eight real-world datasets. The datasets span diverse application scenarios, such as spacecraft, server machines, and water treatment systems, and include CalIt2 (Asuncion & Newman, 2007), NYC (Cui et al., 2016), GECCO (Moritz et al., 2018), MSL (Hundman et al., 2018), PSM (Abdulaal et al., 2021), ECG (Yoon et al., 2020a), SMAP (Hundman et al., 2018), and KDD21 (Qiu et al., 2025). More details about the datasets are provided in Appendix E.1.

Baselines CARE is systematically compared with 14 baselines, covering representative methods and recent advances in TSAD. Specifically, these baselines include the classical methods: OCSVM (Schölkopf et al., 1999), Autoencoder (AE) (Bhatnagar et al., 2021), LSTMED (LSTM) (Sakurada & Yairi, 2014); the recent representative deep learning methods: Anomaly Transformer (ATrans) (Xu et al., 2022), TranAD (Tuli et al., 2022), DCdetector (DC) (Yang et al., 2023), MLP-Mixer (Mixer) (Tang et al., 2024), CATCH (Li et al., 2025b), CrossAD (Cross) (Li et al., 2025a); the approaches adapted from general time series analysis: TimesNet (TsNet) (Wu et al., 2023), iTransformer (iTrans) (Liu et al., 2024), PatchTST (Patch) (Nie et al., 2023); and the large language model-based methods: GPT4TS (GS) (Zhou et al., 2023), CALF (Liu et al., 2025). Among them, CrossAD, MLP-Mixer, and TranAD are regarded as representative state-of-the-art methods in TSAD.

Setup To demonstrate the effectiveness of CARE, we evaluate it from two perspectives: detection quality and inference efficiency. For detection quality, we adopt the label-based metric Affiliated-F1 (Aff-F) and the ranking-based metric Area Under the Precision-Recall Curve (A-P) as primary evaluation metrics. For inference efficiency, we use average inference time (Time) as the core metric. In addition, we report comprehensive results for all metrics in Appendix M. To ensure a strong and up-to-date backbone, we employ the SOTA CrossAD model as the CDM. Notably, CARE is model-agnostic, and the CDM can be replaced with any existing method. Experimental results in Section 4.4 corroborate this. More implementation details are presented in Appendix E.3.

4.1 MAIN RESULTS

We conduct a comprehensive evaluation of CARE against 14 baselines on the eight datasets (Table 1). For detection quality, CARE achieves SOTA performance in both Aff-F and A-P across 5 datasets. On most remaining datasets, it ranks second, with only minor deviations from the best results. This result indicates that NCG effectively retains high-confidence normal samples on the lightweight path while preserving overall detection quality. In terms of efficiency, CARE attains the lowest inference time on all datasets. The substantial inference speedup demonstrates the effectiveness of CARE's routing strategy in reducing unnecessary backend computation. By assigning only uncertain samples to the CDM, the gating mechanism effectively reduces CDM invocations and lowers the overall inference cost. Overall, CARE achieves significant gains in inference efficiency while maintaining high detection quality.

4.2 ABLATION STUDY

Given that NCG is the core component of CARE, we compare it with random and RMA-based routing under the same target CDM allocation ratio, and further ablate Structure Attention, Risk-State Memory, optimization objectives, and the two-stage training strategy. Based on Table 2, we obtain the following observations: **First**, replacing NCG with random or RMA-based routing consistently degrades detection quality, demonstrating the effectiveness of learned routing priorities under the same inference budget. **Second**, removing Structure Attention or Risk-State Memory reduces overall performance, particularly for Risk-State Memory, highlighting the importance of both structural and temporal risk modeling. **Third**, removing any component of $\mathcal{L}_{\text{gate}}$ degrades average performance, while replacing the complete objective with either BCE or percentile regression loss also yields inferior results, confirming the effectiveness of the proposed composite optimization objective. **Finally**, training without the two-stage strategy reduces detection quality, supporting the decoupling of reconstruction learning from gating optimization.

Table 1: Average Aff-F (Affiliated-F1), A-P (AUC-PR), and inference time across 8 real-world datasets. The best results are in bold, while the second-best results are underlined.

Dataset	Metric	CARE	Cross	CATCH	Mixer	DC	TranAD	ATrans	CALF	GS	Patch	iTrans	TsNet	LSTM	AE	OCSVM
CalIt2	Aff-F	0.787	**0.788**	0.780	0.779	0.673	0.780	0.673	0.755	0.783	0.782	0.761	0.747	0.782	0.786	**0.788**
	A-P	**0.100**	0.099	0.094	0.097	0.029	0.052	0.030	0.048	0.087	0.093	0.094	0.079	0.078	0.081	0.095
	Time	**0.122**	0.312	0.583	0.252	0.877	0.265	0.713	0.522	1.016	0.213	0.222	0.572	0.474	0.472	0.259
NYC	Aff-F	**0.696**	0.675	0.668	0.664	0.690	0.661	0.665	0.657	0.689	0.658	0.673	0.662	0.670	0.688	0.667
	A-P	**0.090**	0.086	0.048	0.024	0.041	0.027	0.026	0.051	0.034	0.032	0.040	0.033	0.047	0.043	0.020
	Time	**0.222**	0.501	0.914	0.763	1.938	0.441	1.075	0.829	2.058	0.323	0.751	1.663	0.834	0.557	3.622
GECCO	Aff-F	**0.931**	0.884	0.863	0.379	0.669	0.166	0.669	0.783	0.908	0.844	0.820	0.908	0.351	0.392	0.468
	A-P	**0.607**	0.596	0.331	0.310	0.011	0.191	0.011	0.321	0.295	0.347	0.154	0.479	0.284	0.277	0.039
	Time	**4.506**	11.98	23.64	15.16	22.79	9.520	17.03	15.22	26.01	6.121	4.654	17.44	17.78	9.712	291.6
MSL	Aff-F	0.768	0.761	**0.769**	0.768	0.679	0.763	0.679	0.699	0.768	0.759	0.762	0.760	0.470	0.689	0.688
	A-P	0.262	**0.272**	0.228	0.247	0.108	0.188	0.098	0.178	0.225	0.201	0.213	0.207	0.193	0.144	0.153
	Time	**3.628**	6.862	11.56	7.111	24.80	7.267	17.71	14.75	27.51	6.009	4.292	18.31	17.56	10.73	114.7
ECG	Aff-F	**0.819**	0.799	0.773	0.786	0.728	0.663	0.728	0.754	0.700	0.789	0.741	0.709	0.790	0.793	0.790
	A-P	**0.511**	0.503	0.505	0.495	0.147	0.272	0.157	0.486	0.506	0.490	0.483	0.412	0.500	0.478	0.480
	Time	**8.768**	32.59	71.86	38.31	29.72	9.050	15.04	24.55	21.20	9.693	8.823	14.12	14.18	8.786	145.5
PSM	Aff-F	0.748	**0.764**	0.699	0.729	0.694	0.715	0.694	0.694	0.694	0.696	0.699	0.705	0.693	0.694	0.693
	A-P	**0.547**	0.465	0.381	0.438	0.278	0.380	0.290	0.344	0.379	0.370	0.377	0.394	0.433	0.476	0.418
	Time	**10.96**	30.56	72.60	35.56	36.82	13.50	24.06	32.78	33.33	11.86	11.37	23.45	23.81	14.86	537.0
SMAP	Aff-F	**0.725**	0.712	0.704	0.680	0.680	0.682	0.680	0.679	0.692	0.706	0.680	0.703	0.681	0.683	0.680
	A-P	0.137	0.132	0.124	0.133	0.136	0.110	0.124	0.109	0.124	**0.140**	0.114	0.134	0.103	0.109	0.101
	Time	**18.13**	38.46	67.94	40.98	152.4	42.33	101.7	80.23	165.0	35.88	22.67	108.6	187.6	155.7	1423
KDD21	Aff-F	**0.717**	0.712	0.693	0.714	0.669	0.714	0.695	0.715	0.709	0.696	0.707	0.683	0.698	0.708	0.704
	A-P	0.049	0.044	0.042	0.046	0.009	0.033	0.040	**0.058**	0.023	0.025	0.042	0.021	0.023	0.030	0.028
	Time	**2.354**	7.101	13.06	4.162	11.27	6.108	10.73	8.330	23.53	3.396	2.957	18.52	9.861	8.409	38.39
Avg	Aff-F	**0.774**	0.762	0.744	0.687	0.685	0.643	0.685	0.717	0.743	0.741	0.730	0.735	0.642	0.679	0.685
	A-P	**0.288**	0.275	0.219	0.224	0.095	0.157	0.097	0.199	0.209	0.212	0.190	0.220	0.208	0.205	0.167
	Time	**6.086**	16.05	32.77	17.81	35.08	11.06	23.51	22.15	37.45	9.186	6.966	25.34	34.02	26.15	319.3

Table 2: Ablation studies for CARE. Aff-F and A-P denote Affiliated-F1 and AUC-PR, respectively; higher values are better. All variants use the same target CDM allocation ratio ρ_μ induced by $\mu = 0.3$ for fair comparison. The best values are highlighted in bold.

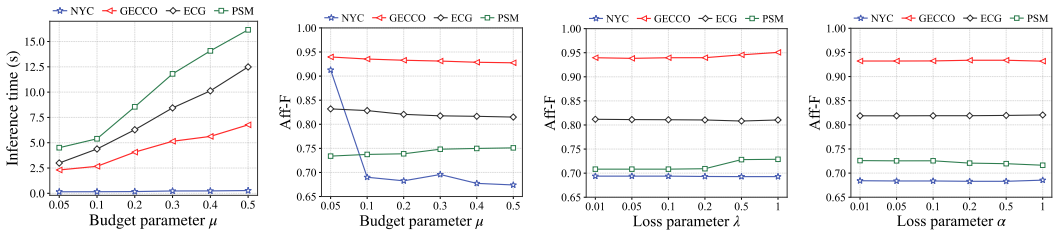
Variations		NYC		GECCO		ECG		PSM		Avg	
		Aff-F	A-P	Aff-F	A-P	Aff-F	A-P	Aff-F	A-P	Aff-F	A-P
Model Components	w/o NCG (Random routing)	0.667	0.069	0.865	0.335	0.812	0.505	0.728	0.505	0.768	0.356
	w/o NCG (RMA routing)	0.693	0.087	0.772	0.466	0.817	0.508	0.717	0.388	0.750	0.365
	w/o Structure Attention	0.686	0.089	0.923	0.604	0.815	0.508	0.737	0.541	0.790	0.438
	w/o Risk-State Memory	0.697	0.085	0.814	0.501	0.812	0.492	0.714	0.388	0.759	0.369
Optimization Objectives	w/o $\mathcal{L}_{\text{bipartite}}$	0.682	0.079	0.908	0.593	0.782	0.495	0.740	0.546	0.778	0.431
	w/o $\mathcal{L}_{\text{boundary}}$	0.694	0.088	0.912	0.601	0.818	0.510	0.748	0.544	0.793	0.438
	w/o $\mathcal{L}_{\text{budget}}$	0.692	0.076	0.911	0.609	0.817	0.501	0.745	0.543	0.792	0.435
	BCE Loss	0.695	0.088	0.903	0.596	0.813	0.510	0.729	0.536	0.785	0.435
	Percentile Regression Loss	0.684	0.086	0.893	0.553	0.802	0.490	0.737	0.546	0.779	0.421
Training w/o Two-stage Strategy		0.681	0.080	0.923	0.602	0.812	0.463	0.746	0.507	0.791	0.416
CARE (ours)		**0.696**	**0.090**	**0.931**	**0.607**	**0.819**	**0.511**	**0.748**	**0.547**	**0.799**	**0.439**

4.3 Parameter Sensitivity

The budget μ, which determines the target CDM allocation ratio ρ_μ, is the most critical parameter, significantly affecting both inference efficiency and detection quality. As shown in Figure 3a, increasing μ consistently increases the inference time because more samples are processed by the CDM. However, the detection performance does not improve monotonically with μ, and excessive CDM invocations may provide only limited gains while incurring substantially higher inference cost, as illustrated in Figure 3b. Overall, setting μ between 0.1 and 0.3 achieves a favorable trade-off between stability and efficiency. We also analyze λ and α in the budget-aware gating objective (Eq. 9), which respectively control the influence of the boundary constraint and the budget-alignment term on NCG training. As depicted in Figures 3c and 3d, λ is stable and easily tunable in the range of 0.5 to 1, while α performs best within the range of 0.05 to 0.1. Additional analyses are provided in Appendix H.

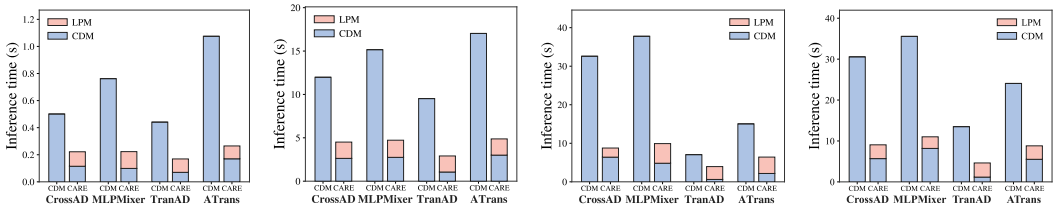
4.4 Adaptability of CARE to Diverse Detection Models

In this subsection, we demonstrate the versatility of CARE by instantiating the CDM with different existing detectors. As shown in Table 3, we use CrossAD, MLP-Mixer, TranAD, and ATrans as CDMs



(a) Inference time with μ (b) Affiliated-F1 with μ (c) Loss parameter λ (d) Loss parameter α

Figure 3: Parameter sensitivity studies of the main hyperparameters in CARE.



(a) NYC (b) GECCO (c) ECG (d) PSM

Figure 4: Inference time comparison of CARE with various CDM backbones.

within CARE and compare their detection results with those of the original models. CARE improves Aff-F in 11 of the 16 model-dataset combinations and does not degrade it in 13, with a maximum relative gain of 25.8%. This improvement suggests that the lightweight pre-filtering mechanism effectively filters normal temporal patterns. By intercepting instances that complex models might otherwise over-interpret, CARE can reduce false positives and enhance overall detection reliability. Meanwhile, CARE substantially improves inference efficiency on all datasets, with a maximum speedup of $4.1\times$. Figure 4 illustrates this improvement by decomposing the computational cost of the LPM and CDM stages: under the current settings, the minimal decision overhead of the LPM substantially reduces the number of CDM invocations. These results validate the generality of CARE: it can be combined with diverse TSAD models to achieve efficient and reliable anomaly detection across different datasets and scenarios. Additional adaptability analyses are provided in Appendix J.

Table 3: The generality of CARE with different CDM models. Aff-F, Time, and $S\times$ denote Affiliated F1-score, inference time, and speedup, respectively. The best results are in bold.

Method	NYC			GECCO			ECG			PSM			Avg		
	Aff-F	Time	S×	Aff-F	Time	S×	Aff-F	Time	S×	Aff-F	Time	S×	Aff-F	Time	S×
CrossAD	0.675	0.501	1.000	0.884	11.981	1.000	0.799	32.589	1.000	**0.764**	30.562	1.000	0.781	18.908	1.000
CARE (CrossAD as CDM)	**0.696**	**0.222**	**2.251**	**0.931**	**4.506**	**2.659**	**0.819**	**8.768**	**3.717**	0.748	**10.956**	**2.790**	**0.799**	**6.113**	**2.854**
MLP-Mixer	0.664	0.763	1.000	0.379	15.158	1.000	0.786	38.310	1.000	**0.729**	35.561	1.000	0.639	22.448	1.000
CARE (MLP-Mixer as CDM)	0.659	**0.224**	**3.412**	**0.392**	**4.724**	**3.209**	**0.829**	**9.920**	**3.862**	0.727	**13.226**	**2.689**	**0.652**	**7.023**	**3.293**
TranAD	0.661	0.441	1.000	0.166	9.520	1.000	0.663	9.050	1.000	0.715	13.497	1.000	0.551	8.127	1.000
CARE (TranAD as CDM)	**0.669**	**0.169**	**2.609**	**0.166**	**2.907**	**3.275**	**0.834**	**3.952**	**2.290**	**0.727**	**4.361**	**3.095**	**0.599**	**2.847**	**2.817**
ATrans	0.665	1.075	1.000	0.669	17.030	1.000	0.728	15.040	1.000	0.694	24.060	1.000	0.689	14.301	1.000
CARE (ATrans as CDM)	**0.672**	**0.265**	**4.061**	**0.669**	**4.875**	**3.494**	**0.834**	**6.420**	**2.343**	**0.731**	**8.298**	**2.899**	**0.726**	**4.965**	**3.199**

5 Conclusion

This work introduces CARE, a cascaded framework that formulates efficient time series anomaly detection as budget-aware selective inference. Its lightweight pre-filter combines Structure Attention and Risk-State Memory with a budget-aware gating objective to retain high-confidence normal samples and prioritize uncertain samples for CDM inference. Extensive experiments on real-world datasets demonstrate the effectiveness and generality of CARE. Future work will investigate adaptive routing under changing resource constraints and evolving time series distributions.

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.

Charu C Aggarwal. An introduction to outlier analysis. In *Outlier analysis*, pp. 1–34. Springer, 2016.

Arthur Asuncion and David Newman. Uci machine learning repository, 2007.

Julien Audibert, Pietro Michiardi, Frédéric Guyard, Sébastien Marti, and Maria A Zuluaga. Usad: Unsupervised anomaly detection on multivariate time series. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 3395–3404, 2020.

Aadyot Bhatnagar, Paul Kassianik, Chenghao Liu, Tian Lan, Wenzhuo Yang, Rowan Cassius, Doyen Sahoo, Devansh Arpit, Sri Subramanian, Gerald Woo, et al. Merlion: A machine learning library for time series. *arXiv preprint arXiv:2109.09265*, 2021.

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

Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6154–6162, 2018.

Cheng-Hsun Chang, Adarsh Prasad Behera, Sophia Zhang Pettersson, and James Gross. A cost-aware hierarchical cascade for anomaly detection at the edge in connected vehicles. In *Proceedings of the Tenth ACM/IEEE Symposium on Edge Computing*, 2025. doi: 10.1145/3769102.3774634.

Dong Chen, Yueting Zhuang, Shuo Zhang, Jinfeng Liu, Su Dong, and Siliang Tang. Data shunt: Collaboration of small and large models for lower costs and better performance. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 11249–11257, 2024a.

Lingjiao Chen, Matei Zaharia, and James Zou. Frugalgpt: How to use large language models while reducing cost and improving performance. *Trans. Mach. Learn. Res.*, 2024b.

Shuhao Chen, Weisen Jiang, Baijiong Lin, James Kwok, and Yu Zhang. Routerdc: Query-based router by dual contrastive learning for assembling large language models. *Advances in Neural Information Processing Systems*, 37:66305–66328, 2024c.

Yuwei Cui, Chetan Surpur, Subutai Ahmad, and Jeff Hawkins. A comparative study of htm and other neural network models for online sequence learning with streaming data. In *2016 International joint conference on neural networks (IJCNN)*, pp. 1530–1538. IEEE, 2016.

Zhihao Dai, Ligang He, Shuang-Hua Yang, and Matthew Leeke. Sarad: Spatial association-aware anomaly detection and diagnosis for multivariate time series. *Advances in Neural Information Processing Systems*, 37:48371–48410, 2024.

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

Shaoyu Dou, Kai Yang, Yang Jiao, Chengbo Qiu, and Kui Ren. Anomaly detection in event-triggered traffic time series via similarity learning. *IEEE Transactions on Dependable and Secure Computing*, 22(2):888–902, 2024.

Yuchen Fang, Jiandong Xie, Yan Zhao, Lu Chen, Yunjun Gao, and Kai Zheng. Temporal-frequency masked autoencoders for time series anomaly detection. In *2024 IEEE 40th international conference on data engineering (ICDE)*, pp. 1228–1241. IEEE, 2024.

Yizeng Han, Gao Huang, Shiji Song, Le Yang, Honghui Wang, and Yulin Wang. Dynamic neural networks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 44(11): 7436–7456, 2021.

D. M. Hawkins. *Identification of Outliers*. Monographs on Applied Probability and Statistics. Springer, 1980. ISBN 978-94-015-3996-8. doi: 10.1007/978-94-015-3994-4. URL https://doi.org/10.1007/978-94-015-3994-4.

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.

Youhe Jiang, Fangcheng Fu, Wanru Zhao, Stephan Rabanser, Nicholas D Lane, and Binhang Yuan. Cascadia: A cascade serving system for large language models. *arXiv e-prints*, pp. arXiv–2506, 2025.

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. *arXiv preprint arXiv:2510.12489*, 2025a.

Mengxuan Li, Ke Liu, Hongyang Chen, Jiajun Bu, Hongwei Wang, and Haishuai Wang. Tsinr: capturing temporal continuity via implicit neural representations for time series anomaly detection. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*, pp. 671–682, 2025b.

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.

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. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 18915–18923, 2025.

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

S Moritz, F Rehbach, S Chandrasekaran, M Rebolledo, and T Bartz-Beielstein. Gecco industrial challenge 2018 dataset. *Tech. Rep.*, 2018.

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

Xiangfei Qiu, Zhe Li, Wanghui Qiu, Shiyan Hu, Lekui Zhou, Xingjian Wu, Zhengyu Li, Chenjuan Guo, Aoying Zhou, Zhenli Sheng, et al. Tab: Unified benchmarking of time series anomaly detection methods. *Proceedings of the VLDB Endowment*, 18(9):2775–2789, 2025.

Moumita Roy, Sukanta Majumder, Anindya Halder, and Utpal Biswas. Ecg-net: A deep lstm autoencoder for detecting anomalous ecg. *Engineering Applications of Artificial Intelligence*, 124: 106484, 2023.

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.

Sebastian Schmidl, Phillip Wenig, and Thorsten Papenbrock. Anomaly detection in time series: a comprehensive evaluation. *Proceedings of the VLDB Endowment*, 15(9):1779–1797, 2022.

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.

Mei-Ling Shyu, Shu-Ching Chen, Kanoksri Sarinnapakorn, and LiWu Chang. A novel anomaly detection scheme based on principal component classifier. 2003.

Haotian Si, Changhua Pei, Zhihan Li, Yadong Zhao, Jingjing Li, Haiming Zhang, Zulong Diao, Jianhui Li, Gaogang Xie, and Dan Pei. Beyond sharing: Conflict-aware multivariate time series anomaly detection. In *Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, pp. 1635–1645, 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.

Qideng Tang, Chaofan Dai, Yahui Wu, and Haohao Zhou. Mlp-mixer based masked autoencoders are effective, explainable and robust for time series anomaly detection. *Proceedings of the VLDB Endowment*, 18(3):798–811, 2024.

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

Paul Viola and Michael J Jones. Robust real-time face detection. *International journal of computer vision*, 57(2):137–154, 2004.

Yuxuan Wang, Haixu Wu, Jiaxiang Dong, Guo Qin, Haoran Zhang, Yong Liu, Yunzhong Qiu, Jianmin Wang, and Mingsheng Long. Timexer: Empowering transformers for time series forecasting with exogenous variables. *Advances in Neural Information Processing Systems*, 37:469–498, 2024.

Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. Timesnet: Temporal 2d-variation modeling for general time series analysis. In *The Eleventh International Conference on Learning Representations (ICLR)*. OpenReview.net, 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. *arXiv preprint arXiv:2410.12261*, 2024.

Jiehui Xu, Haixu Wu, Jianmin Wang, and Mingsheng Long. Anomaly transformer: Time series anomaly detection with association discrepancy. In *The Tenth International Conference on Learning Representations (ICLR)*. OpenReview.net, 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.

Susik Yoon, Jae-Gil Lee, and Byung Suk Lee. Ultrafast local outlier detection from a data stream with stationary region skipping. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 1181–1191, 2020a.

Susik Yoon, Jae-Gil Lee, and Byung Suk Lee. Ultrafast local outlier detection from a data stream with stationary region skipping. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 1181–1191, 2020b.

Chang Yu, Yongshun Xu, Jin Cao, Ye Zhang, Yixin Jin, and Mengran Zhu. Credit card fraud detection using advanced transformer model. In *2024 IEEE international conference on metaverse computing, networking, and applications (MetaCom)*, pp. 343–350. IEEE, 2024.

Zahra Zamanzadeh Darban, Geoffrey I Webb, Shirui Pan, Charu Aggarwal, and Mahsa Salehi. Deep learning for time series anomaly detection: A survey. *ACM Computing Surveys*, 57(1):1–42, 2024.

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

Dingyuan Zhang, Dingkang Liang, Zichang Tan, Xiaoqing Ye, Cheng Zhang, Jingdong Wang, and Xiang Bai. Make your vit-based multi-view 3d detectors faster via token compression. In *European Conference on Computer Vision*, pp. 56–72. Springer, 2024.

Tian Zhou, Peisong Niu, Liang Sun, Rong Jin, et al. One fits all: Power general time series analysis by pretrained lm. *Advances in neural information processing systems*, 36:43322–43355, 2023.

Zixuan Zhou, Xuefei Ning, Ke Hong, Tianyu Fu, Jiaming Xu, Shiyao Li, Yuming Lou, Luning Wang, Zhihang Yuan, Xiuhong Li, et al. A survey on efficient inference for large language models. *arXiv preprint arXiv:2404.14294*, 2024.

Zhihao Zhuang, Yingying Zhang, Kai Zhao, Chenjuan Guo, Bin Yang, Qingsong Wen, and Lunting Fan. Noise matters: Cross contrastive learning for flink anomaly detection. *Proceedings of the VLDB Endowment*, 18(4):1159–1168, 2024.

A Extended Related Work

A.1 Deep Time Series Anomaly Detection

Deep learning-based TSAD utilizes neural networks to learn feature representations or anomaly scores, outperforming traditional methods in many complex scenarios (Zamanzadeh Darban et al., 2024). Existing approaches can be broadly categorized into three paradigms. Forecasting-based methods (Bhatnagar et al., 2021; Deng & Hooi, 2021; Si et al., 2023) detect anomalies by measuring deviations between predicted and observed future steps. Reconstruction-based methods, predominantly relying on AutoEncoders and Transformers (Tang et al., 2024; Fang et al., 2024; Sakurada & Yairi, 2014; Su et al., 2019; Audibert et al., 2020; Xu et al., 2022), identify anomalies through reconstruction errors, with recent advancements incorporating adversarial training (Tuli et al., 2022) or channel fusion (Li et al., 2025b). Alternatively, representation-based approaches (Yang et al., 2023; Wu et al., 2023) focus on learning discriminative temporal representations in the latent space. Beyond dedicated TSAD models, general time series forecasting architectures (Liu et al., 2024; Zeng et al., 2023; Nie et al., 2023; Wang et al., 2024) and recent pre-trained large language models (Zhou et al., 2023; Liu et al., 2025) have also been adapted for anomaly detection.

Despite their architectural diversity, a common limitation characterizes these methodologies: a predominant focus on maximizing detection quality (e.g., accuracy or precision) through increasingly complex networks. This continuous pursuit of accuracy rarely incorporates explicit designs for inference efficiency, rendering these models computationally expensive for latency-sensitive deployments.

A.2 Cascaded Inference Models

Cascaded inference models optimize the adaptive allocation of computational resources by integrating models of varying complexity (Han et al., 2021). The fundamental principle is to process straightforward samples with lightweight, low-cost models, strategically reserving high-capacity architectures for complex cases. These models have been extensively explored across domains such as computer vision (Viola & Jones, 2004; Cai & Vasconcelos, 2018; Zhang et al., 2024) and natural language processing (Chen et al., 2024b;c; Jiang et al., 2025; Zhou et al., 2024). The core mechanism typically uses the prediction confidence of a smaller model to route uncertain samples to a larger one (Chen et al., 2024a). This strategy significantly reduces inference latency without sacrificing overall performance.

In the domain of time series anomaly detection, anomalous events are inherently scarce, resulting in highly skewed data distributions dominated by normal patterns (Aggarwal, 2016; Schmidl et al., 2022). Consequently, the vast majority of normal samples can be accurately identified without invoking complex neural architectures. This intrinsic data characteristic naturally motivates our adoption of a cascaded inference framework to circumvent redundant computations, enabling highly efficient yet reliable anomaly detection.

B Formal Definition of RMA

This section presents a formal representation of the proposed Residual MLP-based AutoEncoder (RMA), providing a structured complement to the preceding model description. Following its architectural design, we describe the model from both encoder and decoder perspectives.

Specifically, the encoding process can be formally stated as:

$$\begin{aligned}\mathbf{h} &= \mathrm{GELU}((\mathrm{Flatten}(\mathcal{W}))\mathbf{W}_1^{(e)} + \mathbf{b}_1^{(e)}) \\ \mathbf{r} &= \mathrm{GELU}((\mathrm{GELU}(\mathbf{h}\mathbf{W}_2^{(e)} + \mathbf{b}_2^{(e)})\mathbf{W}_3^{(e)}) + \mathbf{b}_3^{(e)}) \\ \mathbf{z} &= (\mathbf{h} + \mathbf{r})\mathbf{W}_4^{(e)} + \mathbf{b}_4^{(e)}\end{aligned} \tag{10}$$

where $\mathbf{W}_1^{(e)} \in \mathbb{R}^{(LC)\times D_h}$, $\mathbf{W}_2^{(e)} \in \mathbb{R}^{D_h \times 2D_h}$, $\mathbf{W}_3^{(e)} \in \mathbb{R}^{2D_h \times D_h}$, and $\mathbf{W}_4^{(e)} \in \mathbb{R}^{D_h \times D_z}$. $\mathbf{h}, \mathbf{r} \in \mathbb{R}^{T \times D_h}$ and $\mathbf{z} \in \mathbb{R}^{T \times D_z}$ are the hidden representations. D_h and D_z denote the hidden dimensions of the encoder.

The decoding process can be formally expressed as:

$$\begin{aligned}\hat{\mathbf{h}} &= \mathrm{GELU}(\mathbf{z}\mathbf{W}_1^{(d)} + \mathbf{b}_1^{(d)}) \\ \hat{\mathcal{W}} &= \mathrm{Reshape}(\hat{\mathbf{h}}\mathbf{W}_2^{(d)} + \mathbf{b}_2^{(d)})\end{aligned} \tag{11}$$

where $\mathbf{W}_1^{(d)} \in \mathbb{R}^{D_z \times D_h}$, $\mathbf{W}_2^{(d)} \in \mathbb{R}^{D_h \times (LC)}$ are learnable weight matrices, and $\mathbf{b}_1^{(d)} \in \mathbb{R}^{D_h}$ and $\mathbf{b}_2^{(d)} \in \mathbb{R}^{LC}$ are their corresponding bias terms. $\hat{\mathbf{h}} \in \mathbb{R}^{T \times D_h}$ and $\hat{\mathcal{W}} \in \mathbb{R}^{T \times L \times C}$ denote the hidden representations of the decoder and the set of reconstruction windows, respectively.

C MLE-BASED SCORE ALIGNMENT

Since the raw anomaly scores from the LPM (AS_l) and the CDM (AS_c) operate on heterogeneous scales, they cannot be directly unified. Motivated by the empirically observed right-skewed distributions of anomaly scores, we employ Maximum Likelihood Estimation (MLE) on the training set to fit log-normal distributions for both models: $AS_l^{train} \sim \mathrm{LogNormal}(\mu_l, \sigma_l^2)$ and $AS_c^{train} \sim \mathrm{LogNormal}(\mu_c, \sigma_c^2)$. During inference, we project the raw scores into a unified probabilistic domain using their respective cumulative distribution functions (CDFs, denoted as F_l and F_c). The analysis of this score alignment strategy is provided in Appendix I. The statistically aligned final score is:

$$AS(x_t) = \begin{cases} -\log\big(1 - F_l(AS_l(x_t))\big), & x_t \in \mathcal{N}, \\ -\log\big(1 - F_c(AS_c(x_t))\big), & x_t \in \mathcal{U}. \end{cases} \tag{12}$$

D DETAILED ALGORITHM OF CARE

E EXPERIMENTAL DETAILS

E.1 DATASETS

To verify the effectiveness of the proposed framework, we conduct extensive experiments on several publicly available real-world datasets, which are described in detail below:

- **CalIt2** (Asuncion & Newman, 2007) is a visitor flow rate dataset that records building entry and exit counts over 15 weeks to detect event-related anomalies.
- **NYC** (Cui et al., 2016) is a transportation dataset that records passenger counts from New York City taxi rides at 30-minute intervals.
- **GECCO** (Moritz et al., 2018) is a water quality dataset published in the GECCO Industrial Challenge.
- **MSL** (Hundman et al., 2018) is a spacecraft telemetry dataset that contains incident and anomaly data from the Mars Science Laboratory rover, Curiosity.
- **PSM** (Abdulaal et al., 2021) is a server metrics dataset that records key performance indicators from servers on an online shopping platform.
- **ECG** (Yoon et al., 2020a) is a medical dataset containing features extracted from electrocardiogram signals, where abnormal heartbeats are labeled as anomalies.
- **SMAP** (Hundman et al., 2018) is a spacecraft telemetry dataset that contains soil moisture measurements and telemetry information from the Soil Moisture Active Passive satellite.
- **KDD21** (Qiu et al., 2025) is a multi-domain dataset comprising time-series entities from diverse areas, including healthcare, sports, industry, and robotics.

These datasets span multiple application domains and are highly diverse and representative. Specifically, the feature dimensionality ranges from 1 to 32, the average length of the test sequences ranges from 2,520 to 427,617, and the anomaly ratio ranges from 0.57% to 16.27%. Such diversity in data characteristics provides a comprehensive and reliable basis for evaluating the performance of CARE. More detailed statistics are reported in Table 4.

Algorithm 1 CARE - Overall Architecture.

Require: Input time series $X = \{x_1, x_2, \ldots, x_T\} \in \mathbb{R}^{T \times C}$; input length T; window size L; computational budget μ; costs C_L and C_C.
Ensure: Anomaly Score *AS*.

▷ The input time series is processed into sliding windows.
$\mathcal{W} \leftarrow \text{Windowing}(\text{ZScore}(X))$ ▷ $\mathcal{W} \in \mathbb{R}^{T \times L \times C}$
▷ The RMA is applied to reconstruct the windows.
$\mathbf{z} \leftarrow \text{Encoder}(\mathcal{W})$ ▷ $\mathbf{z} \in \mathbb{R}^{T \times D_z}$
$\hat{\mathcal{W}} \leftarrow \text{Decoder}(\mathbf{z})$ ▷ $\hat{\mathcal{W}} \in \mathbb{R}^{T \times L \times C}$
▷ The NCG generates confidence scores.
$\mathbf{P}, AS_l \leftarrow$ Normality-Conditioned Gating$(\mathcal{W}, \hat{\mathcal{W}})$
$\rho_\mu \leftarrow \frac{\mu(C_L + C_C) - C_L}{C_C}$
$Q_{\rho_\mu} \leftarrow \text{Quantile}(\mathbf{P}, \rho_\mu)$
$\mathcal{U} \leftarrow \text{Filtering}(\mathbf{P}, X, Q_{\rho_\mu})$
if $\mathcal{U}$ is empty **then**
    $AS \leftarrow AS_l$
else
    ▷ Anomaly scores for uncertain samples are computed by the CDM.
    $AS_c \leftarrow \text{CDM}(\mathcal{U})$ ▷ $AS_c \in \mathbb{R}^{|\mathcal{U}| \times 1}$
▷ Anomaly scores are aligned via MLE.
$AS \leftarrow \text{Aligner}(AS_l, AS_c)$ ▷ $AS \in \mathbb{R}^{T \times 1}$
return *AS* ▷ Return the anomaly score.

procedure NORMALITY-CONDITIONED GATING($\mathcal{W}, \hat{\mathcal{W}}$)
    $\mathbf{h}_0 \leftarrow \mathbf{0}$ ▷ Initialize the risk state.
    for $t = 1, \ldots, T$ **do**
        $\mathbf{e}_t \leftarrow (w_t - \hat{w}_t) \odot (w_t - \hat{w}_t)$ ▷ $\mathbf{e}_t \in \mathbb{R}^{L \times C}$
        $\mathbf{f}_{\text{st},t} \leftarrow \text{TemporalPool}(\mathbf{e}_t) \cdot \text{softmax}(\mathbf{v})$ ▷ $\mathbf{f}_{\text{st},t} \in \mathbb{R}^1$
        $\mathbf{f}_{\text{win},t} \leftarrow \text{WindowPool}(\mathbf{e}_t)$ ▷ $\mathbf{f}_{\text{win},t} \in \mathbb{R}^1$
        $\mathbf{f}_{\text{pot},t} \leftarrow \text{ChannelPool}(\mathbf{e}_t[L, :])$ ▷ $\mathbf{f}_{\text{pot},t} \in \mathbb{R}^1$
        $\mathbf{f}_t \leftarrow \text{Concat}(\mathbf{f}_{\text{win},t}, \mathbf{f}_{\text{pot},t}, \mathbf{f}_{\text{st},t})$ ▷ $\mathbf{f}_t \in \mathbb{R}^3$
        $\mathbf{h}_t \leftarrow \text{GRUCell}(\mathbf{f}_t, \mathbf{h}_{t-1})$ ▷ $\mathbf{h}_t \in \mathbb{R}^{D_g}$
        $P(x_t) \leftarrow 1 - \text{Sigmoid}(\mathbf{W}_g \mathbf{h}_t + b_g)$ ▷ $P(x_t) \in [0, 1]$
    return $\mathbf{P}, AS_l$ ▷ Return the confidence and anomaly scores.

procedure FILTERING($\mathbf{P}, X, Q_{\rho_\mu}$)
    for $t = 1, \ldots, T$ **do**
        if $\mathbf{P}_t \geq Q_{\rho_\mu}$ **then**
            Add x_t to $\mathcal{N}$ ▷ $\mathcal{N}$ is the set of high-confidence normal samples.
        else
            Add x_t to $\mathcal{U}$ ▷ $\mathcal{U}$ is the set of uncertain samples.
    return $\mathcal{U}$ ▷ $\mathcal{N}$ is filtered out; only $\mathcal{U}$ is returned.

E.2 METRICS

Since CARE aims to improve inference efficiency while maintaining detection quality, we categorize the evaluation metrics into two groups: detection-quality metrics and inference-efficiency metrics.

Table 4: Statistics of the datasets. (AR: anomaly ratio)

Dataset	Domain	Dimension	Avg Train Length	Avg Val Length	Avg Test Length	Avg AR (%)
CalIt2	Visitors flowrate	2	2016	504	2520	4.09
NYC	Transport	3	14016	3504	4416	0.57
GECCO	Water treatment	9	55408	13852	69261	1.25
MSL	Spacecraft	1	46653	11663	73730	5.88
ECG	Health	32	44883	11221	56105	16.27
PSM	Server Machine	25	105984	26497	87841	11.07
SMAP	Spacecraft	1	108146	27037	427617	9.72
KDD21	Multiple	1	14886	3721	47294	0.58

The detection-quality metrics include Precision (P), Recall (R), F1-score (F1), Affiliated Precision (Aff-P), Affiliated Recall (Aff-R), Affiliated-F1 (Aff-F), the Area Under the Precision-Recall Curve (A-P), the Area Under the Receiver Operating Characteristic Curve (A-R), the Range-based Area Under the Precision-Recall Curve (R-A-P), the Range-based Area Under the Receiver Operating Characteristic Curve (R-A-R), the Volume Under the Precision-Recall Surface (V-PR), and the Volume Under the Receiver Operating Characteristic Surface (V-ROC).

For inference efficiency, we report the average inference time (Time), per-sample latency (PSL), and throughput (Thr.). Specifically, Time is computed as the average elapsed time over three complete inference runs on each dataset, PSL denotes the average inference time required to process a single sample, and Thr. measures the number of windows processed per unit time under the same experimental settings.

E.3 Implementation Details

In our experiments, the dimension of hidden states (D_h) and the dimension of latent states (D_z) of the RMA are set to 16 and 5, respectively. For the NCG, the dimension of hidden states is set to 5. The RMA and the NCG are trained for 10 and 20 epochs, respectively. All models are trained using the Adam optimizer with an initial learning rate of 10^{-4} and a fixed batch size of 64. By default, the sliding-window size matches that of the cascaded CDM to ensure semantic consistency. During the optimization of the NCG with the budget-aware gating objective, ξ_1 and ξ_2 are both set to 0.01, the regularization coefficient α is set to 0.05, the boundary constraint coefficient λ is set to 1.0, and the fixed width w is set to 0.1. The computational budget μ is set to 0.3 by default, with the corresponding CDM allocation ratio ρ_μ computed from Eq. 5. Before formal timing, we perform three batches of warm-up forward passes to reduce cold-start effects and improve the stability of time measurements. All experiments are implemented in PyTorch with Python 3.8 and conducted on an NVIDIA RTX A6000 GPU with 48 GB of memory.

E.4 Budget Allocation Details

Table 5 reports the per-window FLOPs used to determine the CDM allocation ratio at the default budget $\mu = 0.3$. Here, C_L includes the always-executed lightweight stage, and C_C denotes the additional FLOPs required when the CDM is invoked. The target ratio $\rho_{0.3}$ is computed using Eq. 5. We also report the fraction of test windows actually routed to the CDM. End-to-end wall-clock inference times are reported separately in Table 1.

F Analysis of Routing Behavior

We further analyze the routing behavior of CARE on NYC, GECCO, ECG, and PSM under $\mu = 0.3$. Figure 5 shows the proportions of normal and anomalous observations that bypass the CDM, and Table 6 compares the detection results of CARE with those of its CDM counterpart, CrossAD. All rate differences in the table are calculated as CARE minus CrossAD and reported in percentage points. Ground-truth labels are used only for this analysis, not for routing. As shown in Figure 5, CARE bypasses CDM inference for a large proportion of normal observations on all four datasets.

Table 5: Budget allocation at $\mu = 0.3$. Costs are in MFLOPs per window; ratios are percentages.

Dataset	C_L	C_C	Target $\rho_{0.3}$	Observed CDM ratio
CalIt2	0.014	108.57	30.00	30.00
NYC	0.020	162.83	30.00	29.98
GECCO	0.029	342.19	30.00	30.00
MSL	0.005	30.355	30.00	30.00
ECG	0.051	2076.2	30.00	30.00
PSM	0.155	1288.7	30.00	30.01
SMAP	0.008	54.278	30.00	30.00
KDD21	0.004	30.355	30.00	30.00

On NYC, GECCO, and ECG, anomalous observations are more likely than normal observations to be forwarded to the CDM. This indicates that CARE can reduce computation on normal inputs while retaining CDM inference for more anomalous inputs. Table 6 further shows that, on NYC and GECCO, the false-positive rate among bypassed normal observations is lower than that of full CrossAD inference by 4.72 and 0.83 percentage points, respectively. The overall false-positive rates also decrease, and Aff-F improves on both datasets. This suggests that the lightweight stage can help reduce false positives in addition to saving computation. On ECG, CARE reduces the overall false-negative rate by 5.45 percentage points, with only a slight increase in the false-positive rate. On PSM, however, the false-positive rate among bypassed normal observations increases, and detection quality decreases slightly. This result suggests that the effectiveness of pre-filtering also depends on the quality of the lightweight predictions, which we aim to improve in future work.

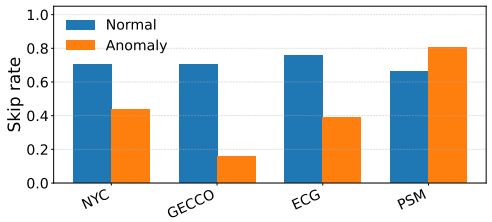


Figure 5: Fractions of normal and anomalous observations that bypass the CDM, calculated separately within each ground-truth class.

Dataset	ΔFPR_N	ΔFPR	ΔFNR	ΔAff-F
NYC	-4.72	-1.99	$+6.06$	$+0.021$
GECCO	-0.83	-0.49	$+4.66$	$+0.047$
ECG	$+0.05$	$+0.16$	-5.45	$+0.019$
PSM	$+31.5$	$+16.5$	-17.5	-0.019

Table 6: Changes relative to full CrossAD inference. Rate differences are in percentage points; ΔAff-F is the absolute change in Affiliated-F1.

G Visualization of Anomaly Scores

We conduct a comparative visualization analysis of CARE against its CDM counterpart, CrossAD, to demonstrate the effectiveness of the cascaded framework. As illustrated in Figure 6, four anomalous subsequences are selected from the KDD21 dataset. The raw time series and the anomaly scores produced by both methods are shown. The results show that CARE and CrossAD exhibit similar sensitivity to anomalous regions. This indicates that CARE largely preserves the anomaly detection capability of the CDM while introducing cascaded inference. This is because CARE bypasses CDM inference for high-confidence normal samples and forwards low-confidence samples to CrossAD for further evaluation. For normal regions, the anomaly score distributions of the two methods differ to some extent. This is mainly because the scores of normal samples in CARE are partially produced by the LPM, and the score alignment between the LPM and the CDM further affects the final score distribution. As a result, CARE and CrossAD maintain similar anomaly sensitivity and ranking behavior, while their absolute score values may differ. Moreover, the pre-filtering mechanism of CARE can also help reduce false positives. As shown in the visualization on KDD120 and KDD119, CrossAD incorrectly identifies a normal segment as anomalous, whereas CARE alleviates this issue. This suggests that CARE may achieve more robust detection performance in certain scenarios, in addition to improving inference efficiency.

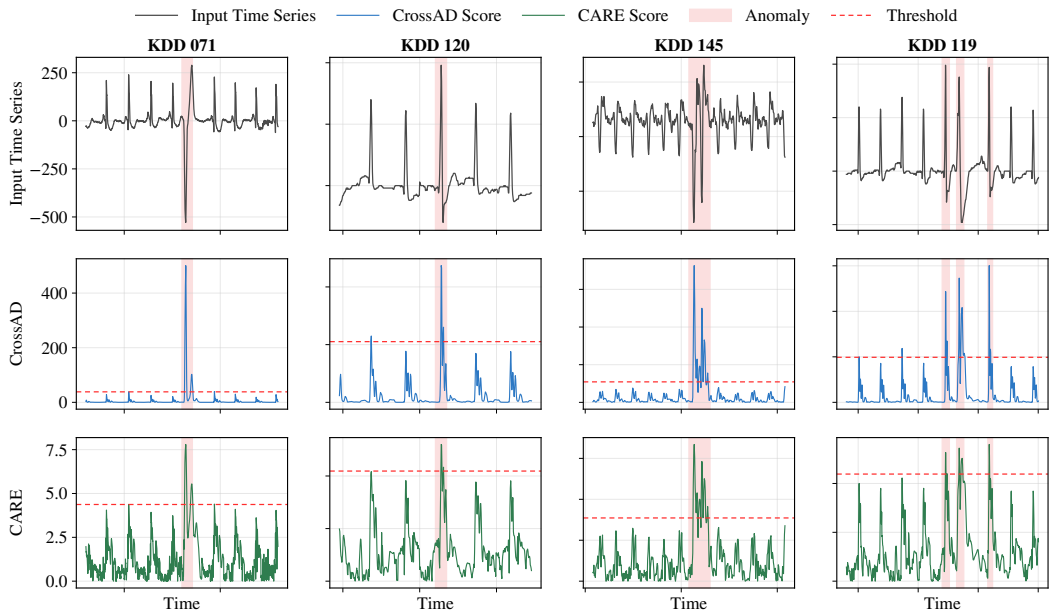
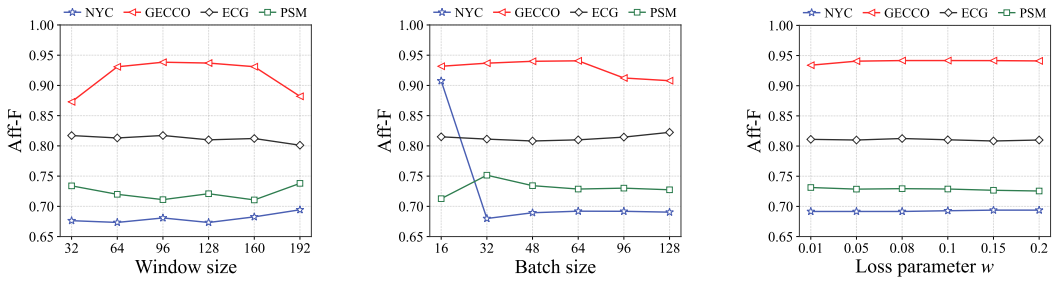


Figure 6: Visualization of anomalies detected by CARE and CrossAD on the KDD21 dataset.



(a) Window size (b) Batch size (c) Loss parameter w

Figure 7: Additional parameter sensitivity studies of main hyperparameters in CARE.

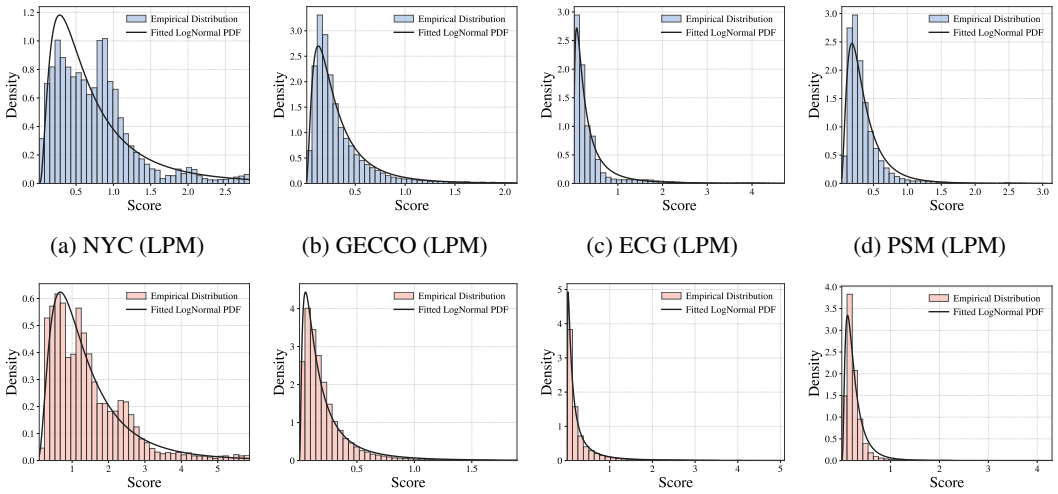
H Additional Hyperparameter Sensitivity Analysis

We analyze the sensitivity of CARE to the boundary width w in the budget-aware gating objective. The boundary width w determines the range of samples involved in fine-grained ranking near the risk threshold. As shown in Figure 7c, model performance gradually stabilizes when w exceeds 0.08. A smaller w covers only a limited number of boundary samples within each batch, making it difficult to effectively supplement the main loss, while appropriately increasing w introduces more threshold-adjacent samples, thereby improving the stability of boundary learning.

Furthermore, we analyze the impact of window size and batch size on model performance. Window size is a fundamental parameter in time series analysis. This paper primarily focuses on the window size of the LPM, while the window size of the CDM follows that of the original model. In our experiments, the CDM adopts the SOTA model CrossAD, with window sizes of 192, 96, 48, and 192 for the NYC, GECCO, ECG, and PSM datasets, respectively. As shown in Figure 7a, when the window size of the LPM matches that of the CDM, the model typically achieves stable and competitive detection performance, suggesting that consistent window sizes improve the semantic alignment between the LPM and the CDM. Regarding batch size, values between 32 and 64 yield stable performance, as shown in Figure 7b.

I Analysis of MLE-based Score Alignment

To assess the score alignment strategy, we visualize the training anomaly scores generated by the LPM and the CDM in Figure 8. Across all four datasets, the scores exhibit pronounced right-skewness and long tails. Despite local irregularities, the fitted log-normal PDFs capture their overall trends, providing a reasonable parametric approximation for score alignment. We further compare the proposed log-normal MLE alignment strategy with no alignment, Gaussian MLE, and empirical CDF. As shown in Table 7, all three aligned variants achieve better average performance than the unaligned variant, demonstrating the benefit of mapping the heterogeneous LPM and CDM scores into a comparable probability space. Log-normal MLE achieves the best average Aff-F and AUC-PR and obtains the strongest results on GECCO and ECG. Meanwhile, Gaussian MLE and empirical CDF remain competitive, indicating that CARE does not critically depend on a specific distributional assumption. Together with the observed right-skewed score distributions, these results support log-normal MLE as an effective choice for score alignment.



(a) NYC (LPM) (b) GECCO (LPM) (c) ECG (LPM) (d) PSM (LPM)

(e) NYC (CDM) (f) GECCO (CDM) (g) ECG (CDM) (h) PSM (CDM)

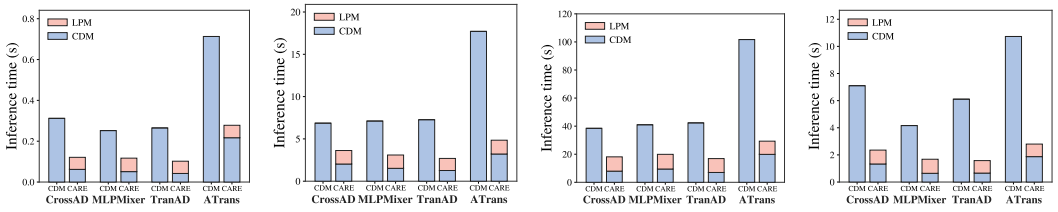
Figure 8: Empirical distributions of training anomaly scores and their fitted log-normal PDFs on representative datasets. For readability, each subplot displays scores up to the 99th percentile of the corresponding training distribution.

Table 7: Comparison of different score alignment strategies in CARE. All variants use the same trained models and evaluation protocol, with only the score alignment module replaced. All alignment mappings are estimated exclusively from training anomaly scores and remain fixed during testing. The best result in each column is highlighted in bold.

Alignment method	NYC		GECCO		ECG		PSM		Avg	
	Aff-F	A-P	Aff-F	A-P	Aff-F	A-P	Aff-F	A-P	Aff-F	A-P
No Alignment	0.692	0.085	0.907	0.598	0.810	0.504	0.747	0.544	0.789	0.433
Gaussian MLE	0.698	0.088	0.920	0.605	0.815	0.508	**0.748**	0.546	0.795	0.437
Empirical CDF	**0.699**	0.086	0.927	0.606	0.813	0.506	**0.748**	**0.547**	0.797	0.436
Log-normal MLE (Ours)	0.696	**0.090**	**0.931**	**0.607**	**0.819**	**0.511**	**0.748**	**0.547**	**0.799**	**0.439**

J Additional Adaptability Analysis

Due to space limitations, we provide an additional generality analysis of CARE on the remaining four datasets in this section. As shown in Table 8, CARE consistently reduces inference time while maintaining competitive detection performance across different CDM models. Although the detection performance is not always superior to the original CDM, the degradation remains within 1% in terms of Aff-F, demonstrating that CARE can effectively preserve detection quality under substantially



(a) CalIt2 (b) MSL (c) SMAP (d) KDD21

Figure 9: Additional inference time comparison of CARE with various CDM backbones.

reduced inference costs. The corresponding inference time comparisons are presented in Figure 9. These results further support the adaptability of CARE as a cascaded inference framework for diverse TSAD models.

Table 8: Additional analysis of CARE generality with different CDM models. Aff-F, Time, and $S\times$ denote Affiliated F1-score, inference time, and speedup, respectively. The best results are in bold.

Method	CalIt2			MSL			SMAP			KDD21			Avg		
	Aff-F	Time	S×	Aff-F	Time	S×	Aff-F	Time	S×	Aff-F	Time	S×	Aff-F	Time	S×
CrossAD	**0.788**	0.312	1.000	0.761	6.862	1.000	0.712	38.460	1.000	**0.712**	7.101	1.000	0.743	13.184	1.000
CARE (CrossAD as CDM)	0.787	**0.122**	**2.566**	**0.768**	**3.628**	**1.892**	**0.725**	**18.128**	**2.122**	**0.717**	**2.354**	**3.017**	**0.749**	**6.058**	**2.399**
MLP-Mixer	0.779	0.252	1.000	**0.768**	7.111	1.000	0.680	40.980	1.000	0.714	4.162	1.000	0.735	13.126	1.000
CARE (MLP-Mixer as CDM)	**0.798**	**0.117**	**2.160**	0.763	**3.096**	**2.297**	**0.708**	**19.922**	**2.057**	**0.716**	**1.680**	**2.477**	**0.746**	**6.204**	**2.248**
TranAD	0.780	0.265	1.000	**0.763**	7.267	1.000	**0.682**	42.330	1.000	**0.714**	6.108	1.000	0.735	13.993	1.000
CARE (TranAD as CDM)	**0.785**	**0.102**	**2.597**	0.758	**2.692**	**2.700**	**0.723**	**16.894**	**2.506**	0.707	**1.580**	**3.866**	**0.743**	**5.317**	**2.917**
ATrans	0.673	0.713	1.000	0.679	17.710	1.000	0.679	101.70	1.000	**0.695**	10.729	1.000	0.682	32.713	1.000
CARE (ATrans as CDM)	**0.684**	**0.278**	**2.569**	**0.734**	**4.846**	**3.654**	**0.680**	**29.306**	**3.470**	0.689	**2.798**	**3.835**	**0.696**	**9.307**	**3.382**

K Stability Analysis

We further evaluate the stability of CARE on representative datasets using different random seeds. We report the mean and standard deviation for both detection quality and inference efficiency metrics, as shown in Table 9. The results demonstrate that CARE maintains stable detection quality and inference efficiency across repeated runs.

Table 9: Stability analysis of CARE over repeated runs with different random seeds. Each value is reported as mean $\pm$ standard deviation.

Metric	NYC	GECCO	ECG	PSM	Avg
Aff-F	0.696 ± 0.003	0.931 ± 0.005	0.819 ± 0.005	0.748 ± 0.003	0.799 ± 0.004
A-P	0.090 ± 0.004	0.607 ± 0.001	0.511 ± 0.004	0.547 ± 0.008	0.439 ± 0.004
Time	0.222 ± 0.014	4.506 ± 0.085	8.768 ± 0.143	10.96 ± 0.063	6.114 ± 0.076
PSL	0.050 ± 0.003	0.065 ± 0.001	0.156 ± 0.002	0.125 ± 0.001	0.099 ± 0.002

L Limitations

In this paper, we propose CARE, a cascaded framework for time series anomaly detection. CARE introduces a lightweight gating model to identify high-confidence normal samples and prioritize low-confidence samples for CDM inference, thereby avoiding uniform complex processing across all samples. Our current experiments evaluate CARE under a fixed normalized budget μ. In practical deployments, resource availability may change over time, while distribution shifts may also affect routing priorities. Extending CARE with dynamic budget selection and online gate adaptation under such evolving conditions is an important direction for future work.

M FULL EXPERIMENTAL RESULTS

Due to space limitations in the main text, this section presents the complete experimental results for CARE. Table 10 reports the full inference-efficiency results in terms of inference time, per-sample latency, and throughput. Tables 11-14 present the detection-quality results from different perspectives, including Precision, Recall, and F1-score; Affiliated Precision, Affiliated Recall, and Affiliated F1-score; A-P, R-A-P, and V-PR; and A-R, R-A-R, and V-ROC, respectively. Furthermore, we include the results of several classical anomaly detection methods, including Isolation Forest (IF) (Liu et al., 2008), Local Outlier Factor (LOF) (Breunig et al., 2000), and Principal Component Analysis (PCA) (Shyu et al., 2003). These results further demonstrate that CARE significantly improves inference efficiency while maintaining competitive detection quality.

Table 10: Average Time (Inference Time, s), PSL (Per-Sample Latency, ms) and Thr. (Throughput, windows/s) across all datasets. The best results are in bold, while the second-best results are underlined.

Dataset	Metric	CARE	Cross	CATCH	Mixer	DC	TranAD	ATrans	CALF	GS	Patch	iTrans	TsNet	LSTM	AE	OCSVM
CalIt2	Time	**0.122**	0.312	0.583	0.252	0.877	0.265	0.713	0.522	1.016	0.213	0.222	0.572	0.474	0.472	0.259
	PSL	**0.048**	0.124	0.231	0.100	0.348	0.105	0.283	0.207	0.403	0.085	0.088	0.227	0.188	0.187	0.103
	Thr.	**20722**	11413	4578.1	10878	3103.3	10275	3744.8	5431.1	2673.2	13984	12701	4766.9	5324.9	5343.5	9740.5
NYC	Time	**0.222**	0.501	0.914	0.763	1.938	0.441	1.075	0.829	2.058	0.323	0.751	1.663	0.834	0.557	3.622
	PSL	**0.050**	0.113	0.207	0.173	0.439	0.100	0.243	0.188	0.466	0.073	0.170	0.377	0.189	0.126	0.820
	Thr.	**19852**	9634.5	5034.7	6512.7	2370.5	10575	4300.5	6162.3	2652.6	14903	15871	5270.6	5302.7	7949.5	1219.2
GECCO	Time	**4.506**	11.98	23.64	15.16	22.79	9.520	17.03	15.22	26.01	6.121	4.654	17.44	17.78	9.712	291.6
	PSL	**0.065**	0.173	0.341	0.219	0.329	0.137	0.246	0.220	0.375	0.088	0.067	0.252	0.257	0.140	4.210
	Thr.	**15370**	5805.8	2935.4	4569.4	3048.1	7306.9	4197.9	4575.1	2671.7	11385	14312	3983.7	3898.2	7136.6	237.5
MSL	Time	**3.628**	6.862	11.56	7.111	24.80	7.267	17.71	14.75	27.51	6.009	4.292	18.31	17.56	10.73	114.7
	PSL	**0.049**	0.093	0.157	0.096	0.336	0.099	0.240	0.200	0.373	0.082	0.058	0.248	0.238	0.145	1.555
	Thr.	**20325**	10791	6397.8	11351	2978.2	10167	4197.4	5018.1	2687.7	12319	17281	4049.3	4199.3	6874.3	643.0
ECG	Time	**8.768**	32.59	71.86	38.31	29.72	9.050	15.04	24.55	21.20	9.693	8.823	14.12	14.18	8.786	145.5
	PSL	**0.156**	0.581	1.281	0.683	0.530	0.1613	0.268	0.438	0.378	0.1728	0.157	0.252	0.253	0.157	2.594
	Thr.	**6399.0**	1767.6	782.1	1485.4	1919.8	6018.1	4015.0	2300.7	2658.5	5745.1	6201.9	4061.0	3969.5	6299.9	385.5
PSM	Time	**10.96**	30.56	72.60	35.56	36.82	13.50	24.06	32.78	33.33	11.86	11.37	23.45	23.81	14.86	537.0
	PSL	**0.125**	0.348	0.827	0.406	0.419	0.154	0.274	0.373	0.379	0.135	0.129	0.267	0.271	0.169	6.114
	Thr.	**8017.8**	2883.7	1211.3	2470.2	2396.4	6632.9	3932.5	2688.4	2643.3	7236.9	7945.0	3757.6	3693.2	5918.1	163.6
SMAP	Time	**18.13**	38.46	67.94	40.98	152.4	42.33	101.7	80.23	165.0	35.88	22.67	108.6	187.6	155.7	1423
	PSL	**0.042**	0.090	0.159	0.096	0.356	0.099	0.238	0.188	0.386	0.084	0.053	0.254	0.439	0.364	3.328
	Thr.	**23589**	11137	6353.5	10443	2808.1	10118	4216.2	5364.2	2593.3	12116	18920	3939.6	2279.1	2746.8	300.5
KDD21	Time	**2.354**	7.101	13.06	4.162	11.27	6.108	10.73	8.330	23.53	3.396	2.957	18.52	9.861	8.409	38.39
	PSL	**0.057**	0.173	0.318	0.101	0.274	0.149	0.261	0.203	0.573	0.083	0.072	0.451	0.240	0.205	0.935
	Thr.	**19782**	6248.7	3872.1	10036	3721.4	7198.7	3842.9	5365.5	1795.8	12557	14633	2430.4	4970.8	6189.6	7777.7
Avg	Time	**6.085**	16.05	32.77	17.79	35.08	11.06	23.51	22.15	37.45	9.186	6.966	25.34	34.02	26.15	319.3
	PSL	**0.074**	0.212	0.440	0.234	0.379	0.125	0.257	0.252	0.417	0.100	0.099	0.291	0.259	0.187	2.457
	Thr.	**16757**	7460.1	3895.6	7218.2	2793.2	8536.3	4055.9	4613.2	2547.0	11281	13483	4032.4	4204.7	6057.3	2558.4
1st count		**27**	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 11: Average P (Precision), R (Recall), and F1 (F1-score) across all datasets. For datasets containing multiple subsets (e.g., KDD21), metrics are macro-averaged over individual subsets before aggregation. The best results are in bold, while the second-best results are underlined.

Dataset	Metric	CARE	Cross	CATCH	Mixer	DC	TranAD	ATrans	CALF	GS	Patch	iTrans	TsNet	LSTM	AE	OCSVM	IF	LOF	PCA
CalIt2	P	0.105	0.106	**0.120**	0.114	0.029	0.089	0.029	0.096	0.103	0.108	0.114	0.097	0.104	0.102	0.114	0.029	0.056	0.096
	R	0.608	0.568	0.432	0.473	**1.000**	0.284	**1.000**	0.149	0.324	0.365	0.324	0.432	0.527	0.604	0.324	**1.000**	0.149	0.568
	F1	0.180	0.179	**0.188**	0.183	0.057	0.136	0.057	0.116	0.156	0.166	0.168	0.158	0.174	0.174	0.169	0.057	0.081	0.164
NYC	P	0.086	0.082	0.055	0.029	0.045	0.046	0.051	**0.117**	0.080	0.045	0.061	0.035	0.078	0.052	0.022	0.022	0.024	0.070
	R	0.434	0.495	0.212	0.828	0.121	0.162	0.081	0.152	0.091	0.293	0.253	0.465	0.192	0.343	**1.000**	0.998	0.525	0.192
	F1	**0.143**	0.141	0.087	0.056	0.066	0.072	0.062	0.132	0.085	0.078	0.099	0.064	0.111	0.091	0.044	0.044	0.046	0.103
GECCO	P	0.641	0.473	0.322	0.672	0.011	0.556	0.011	0.369	0.296	0.350	0.229	0.472	0.763	0.777	0.047	0.011	0.131	**0.779**
	R	0.725	0.728	0.481	0.267	**1.000**	0.321	**1.000**	0.426	0.388	0.497	0.290	0.534	0.268	0.267	0.430	**1.000**	0.299	0.262
	F1	**0.680**	0.573	0.386	0.382	0.021	0.407	0.021	0.396	0.336	0.411	0.256	0.501	0.397	0.398	0.085	0.021	0.182	0.392
MSL	P	0.260	0.211	0.200	0.228	0.105	0.305	0.105	0.158	0.212	0.176	0.186	0.187	**0.331**	0.166	0.168	0.192	0.136	0.328
	R	0.374	0.740	0.821	0.432	0.999	0.261	**1.000**	0.621	0.639	0.692	0.745	0.665	0.216	0.455	0.439	0.398	0.518	0.212
	F1	0.307	**0.329**	0.322	0.299	0.191	0.281	0.191	0.251	0.319	0.281	0.298	0.292	0.262	0.243	0.243	0.259	0.216	0.257
ECG	P	**0.475**	0.458	0.465	0.471	0.152	0.294	0.152	0.404	0.463	0.434	0.396	0.371	0.450	0.430	0.447	0.400	0.184	0.344
	R	0.726	0.671	0.613	0.630	0.999	0.678	**1.000**	0.628	0.496	0.680	0.632	0.598	0.685	0.701	0.661	0.699	0.687	0.659
	F1	**0.574**	0.545	0.529	0.539	0.263	0.410	0.263	0.491	0.479	0.530	0.487	0.458	0.543	0.533	0.533	0.509	0.291	0.452
PSM	P	0.384	**0.422**	0.291	0.362	0.278	0.325	0.278	0.278	0.278	0.279	0.286	0.296	0.302	0.315	0.295	0.278	0.404	0.319
	R	0.743	0.564	0.884	0.596	**1.000**	0.852	0.998	0.998	0.999	0.982	0.904	0.843	0.870	0.959	0.952	0.999	0.843	0.957
	F1	0.506	0.483	0.438	0.451	0.435	0.471	0.435	0.435	0.435	0.435	0.435	0.438	0.448	0.474	0.451	0.435	**0.546**	0.479
SMAP	P	0.158	0.166	0.162	0.128	0.128	0.129	0.128	0.128	0.139	0.152	0.128	0.161	0.130	0.134	0.128	0.128	**0.213**	0.131
	R	0.879	0.841	0.804	**1.000**	0.999	0.969	0.998	0.999	0.691	0.922	**1.000**	0.685	0.978	0.891	0.999	0.998	0.436	0.918
	F1	0.268	0.278	0.269	0.227	0.227	0.228	0.227	0.227	0.232	0.261	0.227	0.261	0.229	0.233	0.227	0.227	**0.286**	0.230
KDD21	P	0.129	0.119	0.080	**0.133**	0.017	0.112	0.075	0.104	0.090	0.079	0.093	0.078	0.096	0.107	0.083	0.102	0.057	0.065
	R	0.619	0.624	0.583	0.533	**0.692**	0.641	0.645	0.514	0.603	0.622	0.511	0.607	0.670	0.618	0.612	0.491	0.571	0.632
	F1	0.074	0.069	0.065	0.077	0.018	0.055	0.056	**0.089**	0.039	0.041	0.068	0.038	0.044	0.049	0.053	0.053	0.054	0.034
Avg	P	0.280	0.255	0.212	0.267	0.096	0.232	0.103	0.207	0.208	0.203	0.187	0.212	**0.282**	0.260	0.163	0.145	0.151	0.267
	R	0.639	0.654	0.604	0.595	**0.851**	0.521	0.840	0.561	0.529	0.632	0.582	0.604	0.551	0.605	0.677	0.823	0.503	0.550
	F1	**0.341**	0.325	0.285	0.277	0.160	0.257	0.164	0.267	0.260	0.275	0.255	0.276	0.276	0.274	0.226	0.201	0.213	0.264
1st count		**5**	2	2	2	5	0	4	2	0	0	1	0	2	0	1	2	3	1

Table 12: Average Aff-P (Affiliated-Precision), Aff-R (Affiliated-Recall), and Aff-F (Affiliated-F1-score) across all datasets. For datasets containing multiple subsets (e.g., KDD21), metrics are macro-averaged over individual subsets before aggregation. The best results are in bold, while the second-best results are underlined.

Dataset	Metric	CARE	Cross	CATCH	Mixer	DC	TranAD	ATrans	CALF	GS	Patch	iTrans	TsNet	LSTM	AE	OCSVM	IF	LOF	PCA
CalIt2	Aff-P	0.651	0.658	0.649	0.666	0.507	0.650	0.507	0.641	0.657	0.648	0.622	0.610	0.644	0.651	0.662	0.507	0.598	**0.671**
	Aff-R	0.994	0.984	0.978	0.938	1.000	0.975	**1.000**	0.918	0.968	0.988	0.982	0.964	0.993	0.994	0.976	**1.000**	0.957	0.895
	Aff-F	0.787	**0.788**	0.780	0.779	0.673	0.780	0.673	0.755	0.783	0.782	0.761	0.747	0.782	0.786	**0.788**	0.673	0.736	0.767
NYC	Aff-P	**0.535**	0.510	0.502	0.497	0.528	0.494	0.499	0.489	0.526	0.491	0.508	0.495	0.505	0.525	0.500	0.500	0.490	0.525
	Aff-R	0.998	0.999	0.997	**1.000**	0.995	0.997	0.993	0.999	0.996	0.998	0.997	0.998	0.996	0.998	**1.000**	0.999	0.999	0.996
	Aff-F	**0.696**	0.675	0.668	0.664	0.690	0.661	0.665	0.657	0.689	0.658	0.673	0.662	0.670	0.688	0.667	0.667	0.658	0.687
GECCO	Aff-P	0.873	0.809	0.767	0.560	0.502	**0.983**	0.502	0.762	0.838	0.752	0.706	0.844	0.696	0.700	0.484	0.502	0.602	0.802
	Aff-R	0.997	0.974	0.988	0.286	**1.000**	0.091	**1.000**	0.806	0.990	0.960	0.978	0.981	0.235	0.272	0.453	**1.000**	0.354	0.146
	Aff-F	**0.931**	0.884	0.863	0.379	0.669	0.166	0.669	0.783	0.908	0.844	0.820	0.908	0.351	0.392	0.468	0.669	0.446	0.248
MSL	Aff-P	**0.647**	0.629	0.626	0.636	0.514	0.644	0.514	0.540	0.626	0.612	0.617	0.613	0.536	0.527	0.527	0.624	0.539	0.602
	Aff-R	0.945	0.961	0.996	0.970	0.999	0.935	**1.000**	0.992	0.994	0.998	0.997	0.997	0.419	0.994	0.993	0.993	0.868	0.355
	Aff-F	0.768	0.761	**0.769**	0.768	0.679	0.763	0.679	0.699	0.768	0.759	0.762	0.760	0.470	0.689	0.688	0.766	0.665	0.446
ECG	Aff-P	**0.815**	0.766	0.723	0.769	0.572	0.495	0.572	0.738	0.674	0.745	0.633	0.592	0.767	0.766	0.764	0.732	0.552	0.682
	Aff-R	0.823	0.836	0.830	0.803	**1.000**	**1.000**	**1.000**	0.772	0.728	0.838	0.892	0.884	0.814	0.831	0.817	0.891	0.772	0.801
	Aff-F	**0.819**	0.799	0.773	0.786	0.728	0.663	0.728	0.754	0.700	0.789	0.741	0.709	0.790	0.793	0.790	0.804	0.643	0.737
PSM	Aff-P	0.598	**0.619**	0.538	0.574	0.532	0.557	0.532	0.532	0.532	0.533	0.538	0.545	0.533	0.534	0.532	0.532	0.597	0.538
	Aff-R	0.997	0.998	0.998	0.998	**1.000**	0.997	0.999	**1.000**	**1.000**	**1.000**	**1.000**	0.998	0.991	0.992	0.996	0.997	0.974	0.986
	Aff-F	0.748	**0.764**	0.699	0.729	0.694	0.715	0.694	0.694	0.694	0.696	0.699	0.705	0.693	0.694	0.693	0.694	0.740	0.696
SMAP	Aff-P	**0.569**	0.554	0.544	0.515	0.515	0.517	0.515	0.515	0.529	0.546	0.515	0.544	0.517	0.519	0.515	0.515	0.494	0.521
	Aff-R	0.999	0.996	0.996	**1.000**	0.999	1.000	0.997	0.998	0.998	0.998	**1.000**	0.996	**1.000**	0.999	**1.000**	0.999	0.739	0.998
	Aff-F	**0.725**	0.712	0.704	0.680	0.680	0.682	0.680	0.679	0.692	0.706	0.680	0.703	0.681	0.683	0.680	0.680	0.592	0.685
KDD21	Aff-P	0.566	0.560	0.538	**0.570**	0.503	0.564	0.539	0.563	0.557	0.540	0.552	0.524	0.544	0.557	0.548	0.554	0.546	0.553
	Aff-R	0.999	0.997	0.997	0.996	0.997	0.996	0.997	**0.999**	0.995	0.993	0.995	0.994	0.998	0.998	0.997	0.996	0.999	0.992
	Aff-F	**0.717**	0.712	0.693	0.714	0.669	0.714	0.695	0.715	0.709	0.696	0.707	0.683	0.698	0.708	0.704	0.708	0.702	0.707
Avg	Aff-P	**0.657**	0.638	0.611	0.598	0.519	0.613	0.523	0.593	0.617	0.608	0.586	0.591	0.593	0.599	0.567	0.558	0.565	0.612
	Aff-R	0.969	0.968	0.973	0.874	**0.999**	0.874	0.998	0.936	0.963	0.967	0.980	0.976	0.806	0.885	0.829	0.973	0.770	0.772
	Aff-F	**0.774**	0.762	0.744	0.687	0.685	0.643	0.685	0.717	0.743	0.741	0.730	0.735	0.642	0.679	0.685	0.707	0.648	0.622
1st count		**10**	3	1	2	4	3	5	2	1	1	1	0	1	0	3	2	1	1

Table 13: Average A-P (AUC-PR), R-A-P (R-AUC-PR), and V-PR (VUS-PR) across all datasets. For datasets containing multiple subsets (e.g., KDD21), metrics are macro-averaged over individual subsets before aggregation. The best results are in bold, while the second-best results are underlined.

Dataset	Metric	CARE	Cross	CATCH	Mixer	DC	TranAD	ATrans	CALF	GS	Patch	iTrans	TsNet	LSTM	AE	OCSVM	IF	LOF	PCA
CalIt2	A-P	**0.100**	0.099	0.094	0.097	0.029	0.052	0.030	0.048	0.087	0.093	0.094	0.079	0.078	0.081	0.095	0.029	0.035	0.073
	R-A-P	0.112	**0.121**	0.117	0.113	0.028	0.059	0.056	0.065	0.093	0.105	0.112	0.092	0.091	0.104	0.109	0.120	0.062	0.106
	V-PR	0.110	0.119	0.114	0.112	0.028	0.057	0.059	0.063	0.091	0.101	0.109	0.093	0.090	0.101	0.109	**0.121**	0.060	0.103
NYC	A-P	**0.090**	0.086	0.048	0.024	0.041	0.027	0.026	0.051	0.034	0.032	0.040	0.033	0.047	0.043	0.020	0.022	0.020	0.046
	R-A-P	0.080	0.075	0.063	0.042	0.056	0.052	0.038	0.075	0.049	0.055	0.058	0.051	0.076	0.102	**0.333**	0.121	0.053	0.076
	V-PR	0.080	0.075	0.064	0.041	0.059	0.052	0.037	0.072	0.051	0.053	0.056	0.048	0.072	0.092	**0.318**	0.121	0.049	0.072
GECCO	A-P	**0.607**	0.596	0.331	0.310	0.011	0.191	0.011	0.321	0.295	0.347	0.154	0.479	0.284	0.277	0.039	0.011	0.086	0.234
	R-A-P	0.491	**0.522**	0.376	0.134	0.508	0.038	0.011	0.294	0.388	0.343	0.162	0.442	0.040	0.053	0.098	0.508	0.058	0.047
	V-PR	0.500	**0.528**	0.363	0.136	0.508	0.038	0.011	0.294	0.385	0.343	0.159	0.441	0.040	0.053	0.101	0.508	0.058	0.046
MSL	A-P	0.262	**0.272**	0.228	0.247	0.108	0.188	0.098	0.178	0.225	0.201	0.213	0.207	0.193	0.144	0.153	0.173	0.128	0.183
	R-A-P	0.330	**0.335**	0.300	0.305	0.186	0.237	0.111	0.241	0.295	0.281	0.287	0.274	0.179	0.194	0.185	0.226	0.188	0.172
	V-PR	0.320	**0.329**	0.296	0.300	0.186	0.236	0.111	0.238	0.291	0.277	0.284	0.271	0.179	0.193	0.185	0.224	0.187	0.172
ECG	A-P	**0.511**	0.503	0.505	0.495	0.147	0.272	0.157	0.486	0.506	0.490	0.483	0.412	0.500	0.478	0.480	0.478	0.202	0.347
	R-A-P	**0.460**	0.451	0.452	0.445	0.134	0.265	0.108	0.412	0.407	0.433	0.405	0.357	0.450	0.432	0.433	0.423	0.181	0.305
	V-PR	**0.490**	0.485	0.489	0.480	0.153	0.300	0.121	0.453	0.447	0.467	0.448	0.404	0.470	0.464	0.466	0.465	0.209	0.343
PSM	A-P	**0.547**	0.465	0.381	0.438	0.278	0.380	0.290	0.344	0.379	0.370	0.377	0.394	0.433	0.476	0.418	0.337	0.462	0.477
	R-A-P	**0.500**	0.465	0.382	0.434	0.440	0.378	0.282	0.344	0.381	0.370	0.379	0.395	0.402	0.429	0.369	0.337	0.422	0.412
	V-PR	**0.500**	0.467	0.383	0.435	0.440	0.379	0.283	0.344	0.382	0.371	0.380	0.397	0.402	0.430	0.370	0.337	0.423	0.413
SMAP	A-P	0.137	0.132	0.124	0.133	0.136	0.110	0.124	0.109	0.124	0.140	0.114	0.134	0.103	0.109	0.101	0.129	**0.183**	0.109
	R-A-P	0.162	0.155	0.147	0.159	0.166	0.127	0.134	0.127	0.144	0.168	0.137	0.159	0.120	0.162	0.152	0.147	**0.191**	0.129
	V-PR	0.162	0.155	0.147	0.158	0.166	0.127	0.135	0.127	0.144	0.167	0.137	0.158	0.121	0.162	0.152	0.147	**0.191**	0.129
KDD21	A-P	0.049	0.044	0.042	0.046	0.009	0.033	0.040	**0.058**	0.023	0.025	0.042	0.021	0.023	0.030	0.028	0.029	0.024	0.019
	R-A-P	0.039	0.034	0.026	0.049	0.017	0.029	0.052	**0.063**	0.025	0.026	0.038	0.023	0.022	0.026	0.028	0.027	0.031	0.021
	V-PR	0.049	0.042	0.032	0.050	0.017	0.030	0.048	**0.063**	0.033	0.033	0.045	0.032	0.022	0.027	0.029	0.031	0.031	0.021
Avg	A-P	**0.288**	0.275	0.219	0.224	0.095	0.157	0.097	0.199	0.209	0.212	0.190	0.220	0.208	0.205	0.167	0.151	0.143	0.186
	R-A-P	**0.272**	0.270	0.233	0.210	0.192	0.148	0.099	0.203	0.223	0.223	0.197	0.224	0.173	0.188	0.213	0.239	0.148	0.158
	V-PR	**0.276**	0.275	0.236	0.214	0.195	0.152	0.101	0.207	0.228	0.227	0.202	0.230	0.175	0.190	0.216	0.244	0.151	0.162
1st count		**12**	6	0	0	0	0	0	3	0	0	0	0	0	0	2	1	3	0

Table 14: Average A-R (AUC-ROC), R-A-R (R-AUC-ROC), and V-ROC (VUS-ROC) across all datasets. For datasets containing multiple subsets (e.g., KDD21), metrics are macro-averaged over individual subsets before aggregation. The best results are in bold, while the second-best results are underlined.

Dataset	Metric	CARE	Cross	CATCH	Mixer	DC	TranAD	ATrans	CALF	GS	Patch	iTrans	TsNet	LSTM	AE	OCSVM	IF	LOF	PCA
CalIt2	A-R	0.817	**0.830**	0.775	0.787	0.474	0.500	0.505	0.567	0.735	0.765	0.783	0.741	0.674	0.727	0.804	0.500	0.494	0.790
	R-A-R	0.848	**0.856**	0.814	0.821	0.465	0.527	0.499	0.652	0.774	0.812	0.813	0.770	0.722	0.788	0.840	0.503	0.546	0.796
	V-ROC	0.838	**0.845**	0.799	0.808	0.465	0.519	0.500	0.627	0.762	0.797	0.806	0.762	0.710	0.772	0.828	0.503	0.519	0.786
NYC	A-R	**0.766**	0.710	0.625	0.563	0.524	0.556	0.522	0.664	0.549	0.541	0.623	0.579	0.667	0.681	0.456	0.500	0.464	0.666
	R-A-R	**0.761**	0.742	0.676	0.656	0.523	0.647	0.506	0.739	0.634	0.692	0.707	0.672	0.731	0.736	0.612	0.505	0.604	0.744
	V-ROC	**0.754**	0.732	0.674	0.637	0.527	0.633	0.505	0.721	0.640	0.667	0.690	0.651	0.716	0.725	0.589	0.505	0.581	0.730
GECCO	A-R	0.964	0.969	0.949	0.871	0.500	0.594	0.476	0.923	0.911	0.959	0.793	**0.974**	0.774	0.724	0.804	0.500	0.796	0.711
	R-A-R	0.911	**0.982**	0.970	0.810	0.501	0.463	0.478	0.936	0.955	0.955	0.853	0.972	0.665	0.586	0.757	0.501	0.777	0.598
	V-ROC	0.947	**0.980**	0.967	0.818	0.501	0.457	0.479	0.934	0.950	0.955	0.846	0.971	0.665	0.588	0.756	0.501	0.767	0.595
MSL	A-R	0.731	**0.759**	0.755	0.685	0.514	0.562	0.397	0.619	0.730	0.709	0.725	0.723	0.541	0.529	0.524	0.600	0.551	0.508
	R-A-R	0.785	**0.800**	0.795	0.748	0.536	0.630	0.434	0.693	0.782	0.768	0.787	0.774	0.563	0.601	0.594	0.657	0.608	0.539
	V-ROC	0.776	**0.793**	0.790	0.742	0.535	0.625	0.434	0.685	0.775	0.762	0.780	0.769	0.558	0.597	0.590	0.653	0.598	0.532
ECG	A-R	**0.866**	0.865	0.861	0.859	0.479	0.750	0.451	0.838	0.826	0.858	0.837	0.811	0.858	0.860	0.858	0.842	0.589	0.784
	R-A-R	0.799	**0.810**	0.803	0.805	0.434	0.652	0.420	0.773	0.757	0.796	0.771	0.738	0.792	0.800	0.801	0.800	0.419	0.647
	V-ROC	0.818	**0.829**	0.825	0.825	0.444	0.676	0.427	0.799	0.783	0.816	0.798	0.767	0.805	0.818	0.820	0.823	0.470	0.672
PSM	A-R	**0.721**	0.665	0.595	0.633	0.500	0.647	0.500	0.569	0.579	0.574	0.587	0.605	0.634	0.660	0.619	0.545	0.720	0.667
	R-A-R	**0.698**	0.662	0.595	0.629	0.501	0.611	0.480	0.565	0.580	0.571	0.585	0.603	0.577	0.597	0.530	0.547	0.639	0.573
	V-ROC	**0.696**	0.662	0.596	0.629	0.501	0.611	0.480	0.563	0.580	0.572	0.586	0.605	0.578	0.598	0.532	0.545	0.641	0.576
SMAP	A-R	0.565	0.526	0.504	0.493	0.509	0.418	0.477	0.420	0.485	0.526	0.447	0.551	0.396	0.426	0.359	0.508	**0.619**	0.423
	R-A-R	**0.612**	0.592	0.552	0.543	0.520	0.457	0.483	0.469	0.520	0.590	0.502	0.595	0.432	0.455	0.396	0.522	0.533	0.437
	V-ROC	**0.608**	0.580	0.548	0.541	0.520	0.455	0.483	0.466	0.520	0.586	0.498	0.591	0.431	0.454	0.395	0.523	0.535	0.435
KDD21	A-R	0.570	0.586	0.606	0.657	0.478	0.590	0.530	**0.667**	0.598	0.605	0.631	0.581	0.582	0.622	0.609	0.498	0.592	0.575
	R-A-R	0.635	0.633	0.656	0.708	0.492	0.634	0.541	**0.713**	0.665	0.664	0.690	0.652	0.649	0.672	0.663	0.582	0.637	0.627
	V-ROC	0.631	0.633	0.650	0.699	0.494	0.627	0.538	**0.719**	0.658	0.661	0.685	0.648	0.647	0.669	0.654	0.579	0.637	0.625
Avg	A-R	**0.750**	0.739	0.709	0.694	0.497	0.577	0.482	0.658	0.677	0.692	0.678	0.696	0.641	0.654	0.629	0.562	0.603	0.641
	R-A-R	0.756	**0.760**	0.733	0.715	0.497	0.578	0.480	0.693	0.708	0.731	0.713	0.722	0.641	0.655	0.649	0.577	0.595	0.620
	V-ROC	**0.759**	0.757	0.731	0.712	0.498	0.575	0.481	0.689	0.708	0.727	0.711	0.720	0.639	0.653	0.646	0.579	0.594	0.619
1st count		**11**	**11**	0	0	0	0	0	3	0	0	0	1	0	0	0	0	1	0