TEDA: TIME-FREQUENCY ENTROPY-AWARE DISTRIBUTION MODELING FOR TIME SERIES ANOMALY DETECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

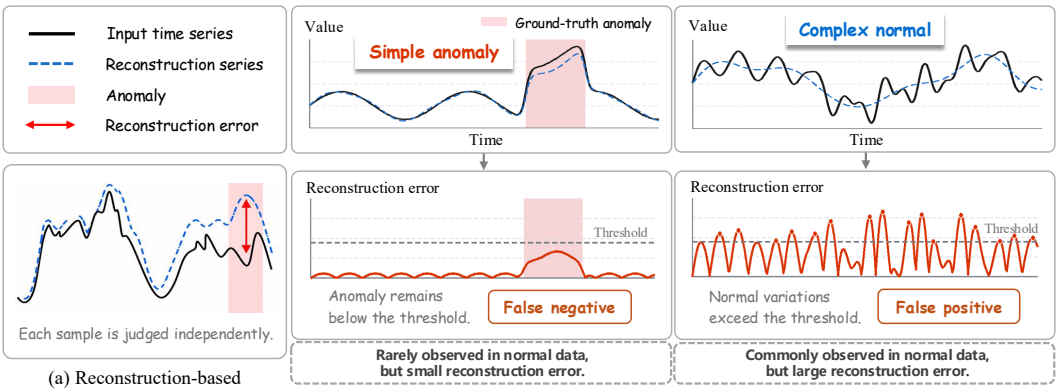
Time series anomaly detection plays an important role in many real-world systems. Existing advanced unsupervised methods mainly rely on prediction or reconstruction errors to identify abnormal patterns. However, such a paradigm only focuses on sample-wise errors and fails to capture whether a sample conforms to the distribution of whole data. As a result, simple anomalies but with low-probability may be well reconstructed, while complex but normal patterns may be failed. In this paper, we propose TEDA, a time-frequency distribution-aware framework for unsupervised time series anomaly detection. TEDA learns distributional centroids and introduces optimal transport into the reconstruction-based paradigm to model normal patterns in the latent distributional space. Specifically, a patch-based branch captures local temporal distributions, while a frequency-domain branch models long-term periodicity, trends, and global distributions. To improve the stability and discriminative ability of optimal transport, TEDA uses entropy regularization to preserve smooth transport plans during training, and removes it to perform sharp assignment during inference. The final anomaly score combines reconstruction errors and distribution-aware optimal transport costs. Extensive experiments on six real-world datasets show that TEDA achieves better performance compared with classical and recent state-of-the-art methods. The results demonstrate the effectiveness of enhancing time series anomaly detection with the distribution of whole data.

1 INTRODUCTION

As society becomes increasingly digitalized, time series analysis is widely used in industrial monitoring, cloud services, intelligent transportation, healthcare, and financial risk control (Cook et al., 2019). In these scenarios, abnormal patterns may indicate system failures, performance degradation, cyber attacks, or other risks. Due to the scarcity and high cost of anomaly annotations, unsupervised time series anomaly detection has attracted increasing attention. Reconstruction-based deep learning methods are widely used for unsupervised time series anomaly detection. They typically learn representations of normal patterns with an encoder-decoder architecture and detect anomalies using reconstruction errors. The underlying assumption is that normal patterns can be well reconstructed, whereas anomalous patterns tend to yield larger reconstruction errors.

Although this reconstruction paradigm is simple and effective and has achieved good performance in many anomaly detection scenarios, it still has a fundamental limitation, as shown in Fig. 1: reconstruction error evaluates each sample independently and cannot explicitly capture distributional abnormality (Zhang et al., 2020; Bouman & Heskes, 2025). Specifically, some anomalous samples exhibit temporal patterns that are rarely observed in normal data. However, due to their relatively simple structures, these samples may still be accurately reconstructed, resulting in small reconstruction errors and potential false negatives(Zhang et al., 2020; Gong et al., 2019). Conversely, some normal samples exhibit patterns that are commonly observed in normal data but contain complex local variations or periodic structures that are difficult to reconstruct precisely, potentially leading to large reconstruction errors and false positives (Lan et al., 2025; Chen et al., 2024; Goodge et al., 2022). Therefore, reconstruction error alone may not reliably indicate whether a sample conforms to the normal data distribution.

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) Reconstruction-based Anomaly Detection

(b) Reconstruction Error Fails to Capture Distributional Normality

Figure 1: Limitations of Reconstruction-based Time Series Anomaly Detection from a Distribution Perspective.

These limitations suggest that time series anomaly detection models need to be equipped with distribution awareness. That is, the model should not only reconstruct normal samples, but also characterize their distributional structure in the latent space and determine whether a test sample is consistent with this normal distribution. However, two challenges still remain: **First,** time series data exhibit complex and heterogeneous distributional characteristics: patch-level local time series segments reflect distributions of short-term temporal patterns (Nie et al., 2022; Park & Kang, 2026), whereas window-level frequency-domain features reflect distributions of long-term periodicity, trends, and cross-patches temporal dependencies (Wu et al., 2022; 2025). A single time-domain distribution learning is insufficient to fully capture such complex and heterogeneous distributional characteristics. **Second,** even though some recent works (Montesuma et al., 2025; Villani et al., 2009) have introduced optimal transport (OT) for distribution modeling, these naive OTs cannot be stably trained under modern deep neural networks, making them inferior to the deep reconstruction-based anomaly detection models (Li et al., 2025; Wu et al., 2025).

To address these challenges, we propose a time-frequency distribution-aware model for time series anomaly detection. **For the first challenge,** we incorporate deep optimal transport and learnable distributional centroids into the reconstruction-based anomaly detection framework, to characterize the heterogeneous distribution of normal time series through a dual-branch time-frequency structure. Specifically, the time-domain branch divides the input time series into local patches and learns patch-based local temporal features to capture distributions of short-term and local patterns. The frequency-domain branch maps the input time series into the frequency space through the Fourier transform and learns long-term distributional structures based on frequency components, thereby capturing periodicity, trends, and global cross-patch temporal patterns. Then, the two branches use deep optimal transports to quantify how a sample conforms to the normal data distribution by matching the relationship between sample representations and the learned distributional centroids, and jointly provide distribution-aware representations for the reconstruction-based decoder. **For the second challenge,** we design novel and differentiated assignment strategies to learn optimal transport plans. During training, we utilize entropy regularization to ensure smooth transport plans, allowing sample representations to be diversely assigned to multiple distributional centroids and stabilizing the learning of diverse normal patterns in both the temporal and frequency domains. During inference, we remove the entropy smoothing regularization and adopt a sharp assignment strategy, so that the representation of each test time series is directly assigned with its one nearest normal distributional centroid, yielding a sharper measurement of distributional deviation that helps to distinguish normal and abnormal samples.

Our main contributions are summarized as follows:

- We propose a time-frequency distribution-aware optimal transports framework for time series anomaly detection. We use learnable centroids to characterize typical distributional patterns of normal time series, and model local temporal distributions and long-term frequency-domain distributions through dual time-domain and frequency-domain branches, to capture the heterogeneous distributional structure of normal data.

- We investigate the role of entropy regularization in optimal transports for anomaly detection and propose training-inference differentiated optimal transport assignment strategies. During training, entropy regularization stabilizes model learning, while entropy smoothing is removed during inference so that the transport costs can be used as sharper anomaly scores.
- We conduct experiments on multiple real-world time series anomaly detection benchmark datasets. Experimental results demonstrate that our method consistently achieves superior performance across different datasets and evaluation metrics.

2 Related Work

2.1 Time Series Anomaly Detection

Time series anomaly detection has been extensively studied. Traditional methods mainly include statistical density-based, distance-based, boundary-based, and clustering-based anomaly detection methods, such as LOF (Breunig et al., 2000), OC-SVM (Schölkopf et al., 1999), Isolation Forest (Liu et al., 2008), PCA (Shyu et al., 2003), and Matrix Profile (Yeh et al., 2016). Recent deep learning methods can be broadly categorized into forecasting-based, reconstruction-based, and representation-based approaches. Forecasting methods, such as GDN (Deng & Hooi, 2021), identify anomalies through prediction errors. Reconstruction-based methods learn representations of normal sequences and use reconstruction discrepancies for detection (Su et al., 2019; Audibert et al., 2020; Xu et al., 2021; Tuli et al., 2022; Zhou et al., 2023; Luo & Wang, 2024; Shentu et al., 2025) Contrastive methods, such as DCdetector (Yang et al., 2023), distinguish normal and anomalous patterns through representation discrepancies. In addition, recent studies have further explored more well-designed or more efficient modules for the deep reconstruction-based anomaly detection framework, such as TFMAE (Fang et al., 2024) uses temporal window masking and amplitude-frequency masking for temporal-frequency representation learning, CATCH (Wu et al., 2025) based on frequency patching representation learning, PaAno (Park & Kang, 2026) based on patch representation learning, CrossAD based on cross-patch learning (Li et al., 2025), and ALoRa (Shimillas et al., 2026) based on efficient low-rank attention modeling. Unlike these error-based methods, this paper further focuses on whether a sample conforms to the latent distribution structure of normal time series.

2.2 Optimal Transport-based Anomaly Detection

Optimal Transport (OT) provides a principled framework for measuring discrepancies between probability distributions by minimizing mass transportation costs (Villani et al., 2009; Peyré & Cuturi, 2019; Cuturi, 2013). Existing studies have applied OT to anomaly detection in two main ways. Some directly compute transport distances between time series or signals to identify anomalous samples (Alaoui-Belghiti et al., 2019; 2020). Others combine OT with prototypes or deep representations. For example, PUAD (Li et al., 2023) models normal dynamics using global and local prototypes and employs transport distances to retrieve normal patterns for reconstruction, while its anomaly score remains based on reconstruction errors. MROT (Montesuma et al., 2025) formulates detection as a self-transport problem: an exclusion zone prevents local transport, causing samples in low-density regions to incur larger transportation costs. However, existing methods mainly model normal time series in the original data space, a single time-domain latent space, or through distance measures, without learning the probabilities and the heterogeneous distributional characteristics of local temporal patterns and long-term frequency structures in time series. Moreover, these naive OTs are unaware of entropy and suffer from unstable training and inferior performance. In contrast, this paper introduces entropy-aware learnable distributional centroids, proposes deep optimal transport to learn distributions for both patch representations and frequency representations, and proposes novel distribution-aware anomaly criteria.

3 Methodology

Given a multivariate time series, $X = (x_1, x_2, \ldots, x_T) \in \mathbb{R}^{T \times C}$, where T is the temporal length and C is the number of variables. Time series anomaly detection aims to learn a detection model from unlabeled training data and assign an anomaly score $S(t)$ and an anomaly label $\hat{y}_t \in \{0, 1\}$ to each timestamp in an unseen test sequence $X^{\text{test}} \in \mathbb{R}^{T' \times C}$, where 1 denotes an anomaly and 0

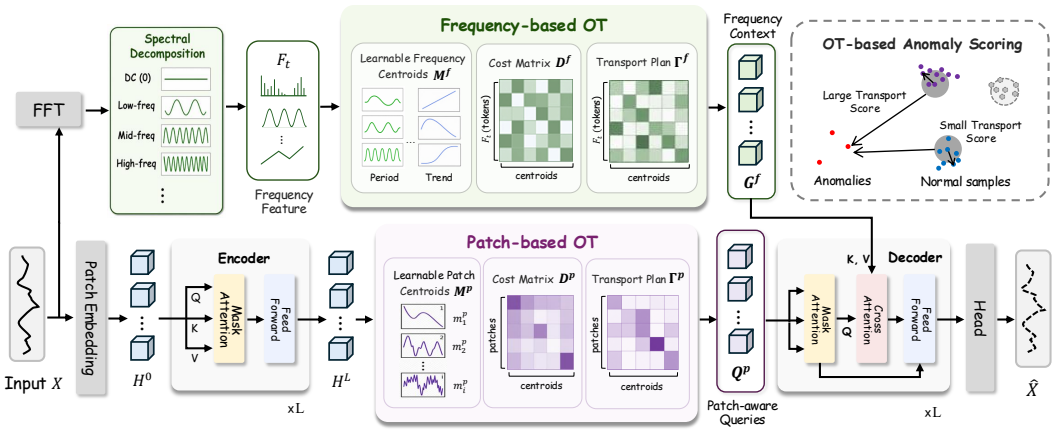


Figure 2: Overall framework of **TEDA**. Given an input X, the model extracts complementary representations from temporal and frequency views. The temporal branch applies patch embedding and an encoder to obtain patch representations, which are matched with learnable patch centroids by the patch-based OT module. The frequency branch applies FFT to obtain frequency tokens, which are matched with learnable frequency centroids by the frequency-based OT module. The resulting patch-aware queries (Q^p) and frequency context (G^f) are fed into the decoder for distribution-aware reconstruction. TEDA employs entropy-regularized transport plans to stabilize centroid learning during training and entropy-free sharp nearest-centroid assignment to measure distributional deviations during inference. The final anomaly score combines reconstruction errors and OT-induced distributional costs.

denotes a normal timestamp. Following channel-independent modeling, each variable is treated as an individual univariate sequence, while the same model parameters generalize across all time series variables. For a sliding window $X_{i:i+W-1} \in \mathbb{R}^{W\times C}$ of length W, we reshape it into C univariate windows $\{x^{(c)}_{i:i+W-1}\}_{c=1}^{C}$, where $x^{(c)}_{i:i+W-1} \in \mathbb{R}^{W}$.

3.1 Overall Architecture

The overall architecture of TEDA is shown in Figure 2. For an input time series segment X, the model first extracts representations from two complementary views. The temporal branch applies patch embedding and an encoder to obtain temporal representations H^L, where L denotes the number of encoder layers. These representations capture local patch patterns and contextual temporal dependencies. The frequency branch applies Fast Fourier Transform (FFT) and feature projection to obtain frequency tokens F, which characterize periodicity, trends, and global variations within the input window. Each branch then uses OT to match its representations with learnable distributional centroids representing normal temporal or frequency patterns, producing the transport plans Γ^p and Γ^f. Based on these transport plans, the model constructs the patch-aware queries Q^p and frequency context G^f. During decoding, the model uses the patch-aware queries Q^p as queries and the frequency context G^f as keys and values in cross-attention. The decoder reconstructs the input sequence under the joint constraints of local temporal distributions and global frequency distributions. During training, the model jointly optimizes the reconstruction loss and two OT-based distributional losses using entropy-regularized transport plans to stabilize centroid learning. During inference, it combines reconstruction errors with entropy-free sharp-assignment costs to compute a variable-wise anomaly score. The resulting variable-wise scores are then aggregated across variables and thresholded to produce the final timestamp-level anomaly predictions.

3.2 Time-Frequency Distribution Modeling

Temporal representation. Given an input window $X \in \mathbb{R}^{W\times C}$, where W is the window length and C is the number of variables, the model adopts channel-independent modeling by reshaping the multivariate sequence into multiple univariate sequences. For simplicity, we omit the variable index in this subsection and denote one univariate input window as $x \in \mathbb{R}^{W}$.

Let P denote the patch length and N_p denote the number of patches where $N_p = \lceil \frac{W}{P} \rceil$. The input window x is divided into N_p patches and projected into a d-dimensional latent space. The patch representations are then encoded as

$$H^L = \text{Encoder}\left(\text{PatchEmbed}(x) + E^p_{\text{pos}}\right), \tag{1}$$

where L is the number of encoder layers, d is the hidden dimension, $E^p_{\text{pos}} \in \mathbb{R}^{N_p \times d}$ is the learnable patch positional embedding, and $H^L \in \mathbb{R}^{N_p \times d}$ captures local patch patterns and their contextual temporal dependencies.

Frequency representation. To capture long-term periodicity, trends, and global variations, we further introduce a frequency-domain distribution modeling branch. For a channel-independent input window $x \in \mathbb{R}^W$, the model applies real FFT to obtain non-redundant frequency-domain coefficients. Since the resulting Fourier coefficients are complex-valued, we concatenate their real and imaginary parts as frequency features and further project them into the d-dimensional latent space:

$$F = \text{Proj}\left([\text{Re}(\text{rFFT}(x)), \text{Im}(\text{rFFT}(x))]\right) + E^f_{\text{pos}}, \tag{2}$$

where $F, E^f_{\text{pos}} \in \mathbb{R}^{N_f \times d}$, and $N_f = \lfloor W/2 \rfloor + 1$.

OT-based distribution matching. Optimal transport measures the discrepancy between two distributions by finding a transport plan with minimum total cost (Villani et al., 2009; Peyré & Cuturi, 2019). In TEDA, the source distribution is formed by temporal patch representations or frequency tokens, while the target distribution is formed by learnable distributional centroids representing normal temporal or frequency patterns. The formal OT formulation and Sinkhorn optimization are provided in Appendix A.

To characterize normal patterns in the temporal and frequency domains, we maintain two learnable distributional centroid tables:

$$M^p = \{m^p_1, m^p_2, \ldots, m^p_K\} \in \mathbb{R}^{K \times d}, \quad M^f = \{m^f_1, m^f_2, \ldots, m^f_K\} \in \mathbb{R}^{K \times d}. \tag{3}$$

Here M^p represents typical normal local temporal patterns and M^f represents typical normal frequency-domain patterns. K denotes the number of distributional centroids in each branch.

Let h_i and f_i denote the i-th temporal patch representation in H^L and the i-th frequency token in F, respectively. We compute their distances to the corresponding distributional centroids as

$$D^p_{ij} = d(h_i, m^p_j), \quad D^f_{ij} = d(f_i, m^f_j), \tag{4}$$

where $D^p \in \mathbb{R}^{N_p \times K}$ and $D^f \in \mathbb{R}^{N_f \times K}$. Here, $d(\cdot, \cdot)$ denotes the distance function, which can be cosine distance or normalized Euclidean distance.

Based on the distance matrices, the two OT modules compute the corresponding transport plans:

$$\Gamma^p = \text{OT}(D^p), \quad \Gamma^f = \text{OT}(D^f), \tag{5}$$

where $\Gamma^p \in \mathbb{R}^{N_p \times K}$ and $\Gamma^f \in \mathbb{R}^{N_f \times K}$. The centroid-induced representations are then obtained as

$$G^p = \Gamma^p M^p, \quad G^f = \Gamma^f M^f. \tag{6}$$

Finally, the temporal representation is combined with its centroid-induced representation to construct the patch-aware queries:

$$Q^p = H^L + G^p. \tag{7}$$

The patch-aware queries Q^p and frequency context G^f are then used for distribution-aware reconstruction.

3.3 Training-Inference Differentiated Optimal Transport Assignment

The OT branches have different objectives during training and inference. During training, the model requires smooth transport plans to stably learn diverse normal distributional centroids. During inference, the learned centroids are fixed, and the objective changes to measuring whether a test

representation can be matched with a normal centroid at a low cost. We therefore retain entropy regularization during training and adopt entropy-free sharp assignment during inference.

For each branch $b \in \{p, f\}$, let D^b denote the corresponding representation-to-centroid distance matrix. During training, the entropy-regularized transport plan is computed as

$$\Gamma^b_{\text{smooth}} = \underset{\Gamma \in \Pi(\boldsymbol{\mu}^b, \boldsymbol{\nu}^b)}{\arg\min} \left\langle \Gamma, D^b \right\rangle - \epsilon H(\Gamma), \tag{8}$$

where $\Pi(\boldsymbol{\mu}^b, \boldsymbol{\nu}^b)$ denotes the set of transport plans satisfying the source and target marginal constraints, and $\epsilon > 0$ is the entropy-regularization coefficient. The entropy of a transport plan is

$$H(\Gamma) = -\sum_{i,j} \Gamma_{ij} \log \Gamma_{ij}. \tag{9}$$

Entropy regularization produces smooth assignments between representations and multiple centroids, preventing premature sharp matching and enabling stable learning of diverse normal temporal and frequency patterns.

During inference, retaining entropy regularization may over-smooth the matching relationships and average the distances to multiple centroids, thereby weakening the distributional deviations of anomalous representations. We therefore use entropy-free sharp nearest-centroid assignment.

Let $z^p_i = h_i$ and $z^f_i = f_i$ denote the i-th temporal patch representation and frequency token, respectively. For each representation z^b_i, its nearest normal distributional centroid is defined as

$$j^{b,\star}_i = \underset{1 \le j \le K}{\arg\min} D^b_{ij}. \tag{10}$$

The corresponding sharp-assignment matrix and distributional cost are

$$\Gamma^b_{\text{sharp},ij} = \mathbb{I}\left(j = j^{b,\star}_i\right), \quad o^b_i = \sum_{j=1}^{K} \Gamma^b_{\text{sharp},ij} D^b_{ij} = D^b_{i,j^{b,\star}_i}. \tag{11}$$

This assignment directly measures the distance from each representation to its nearest normal distributional centroid without averaging over multiple centroids, thereby producing more discriminative distributional anomaly scores. Since Γ^b_{sharp} does not generally satisfy the target marginal constraints, it is not a solution to the balanced OT problem in Eq. (8); the resulting o^b_i is used as an OT-induced distributional cost during inference.

3.4 Distribution-Aware Decoding and Training Objective

After obtaining the patch-aware queries Q^p and the frequency context G^f, the model performs distribution-aware reconstruction through a lightweight decoder. Specifically, the decoder first uses masked self-attention to model the temporal dependencies among the patch-aware queries, and then uses $\tilde{Q}^p$ as queries and G^f as keys and values for cross-attention.

$$\tilde{Q}^p = \text{MaskedSelfAttn}(Q^p, Q^p, Q^p), \quad U = \text{CrossAttn}(\tilde{Q}^p, G^f, G^f). \tag{12}$$

The resulting representation is passed through a fully connected output head to reconstruct the input sequence:

$$\hat{x} = \text{Head}(U). \tag{13}$$

This decoding process integrates local temporal and global frequency distributional information to guide the reconstruction of the input sequence. The reconstruction loss is defined as

$$\mathcal{L}_{rec} = \|x - \hat{x}\|_2^2. \tag{14}$$

The final training objective jointly optimizes reconstruction and the two distributional constraints:

$$\mathcal{L} = \mathcal{L}_{\text{rec}} + \lambda_p \left\langle \Gamma^p_{\text{smooth}}, D^p \right\rangle + \lambda_f \left\langle \Gamma^f_{\text{smooth}}, D^f \right\rangle, \tag{15}$$

where λ_p and λ_f control the contributions of the temporal and frequency OT-based distributional losses, respectively. Entropy regularization is incorporated into training through Γ^p_{smooth} and Γ^f_{smooth}, rather than introduced as an additional independent loss. This objective enables the model to reconstruct normal time series while learning stable matching relationships with normal temporal and frequency distributional centroids.

3.5 ANOMALY SCORING

The final anomaly score combines reconstruction errors with OT-induced distributional scores. Since the model follows channel-independent modeling, we first compute anomaly scores for each variable and then aggregate them across variables. For the c-th variable, the reconstruction anomaly score at timestamp t is defined as the point-wise discrepancy between the input value and the reconstructed value, and the OT-induced distributional score combines the temporal and frequency deviations:

$$S_{\text{rec}}^{(c)}(t) = \left| x_t^{(c)} - \hat{x}_t^{(c)} \right|^2, \quad S_{\text{OT}}^{(c)} = \alpha_p \sum_{r=1}^{N_p} o_r^{p,(c)} + \alpha_f \sum_{r=1}^{N_f} o_r^{f,(c)}, \tag{16}$$

where $o_r^{p,(c)}$ and $o_r^{f,(c)}$ are the sharp-assignment costs defined in Eq. (11) for the r-th temporal patch representation and frequency token, respectively. The weights α_p and α_f balance the local temporal and global frequency distributional deviations.

The variable-wise anomaly score combines the reconstruction and distributional deviations. Since $S_{\text{OT}}^{(c)}$ describes the entire input window, it is assigned to all timestamps within that window. The variable-wise score, final timestamp-level score, and prediction are

$$S^{(c)}(t) = S_{\text{rec}}^{(c)}(t) + \beta S_{\text{OT}}^{(c)}, \quad S(t) = \sum_{c=1}^{C} S^{(c)}(t), \quad \hat{y}_t = \mathbb{I}(S(t) > \tau), \tag{17}$$

where β controls the weight of the OT-induced distributional score and τ is the anomaly threshold.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETTINGS

Datasets. We evaluate TEDA on six public time series anomaly detection datasets: MSL (Hundman et al., 2018b), PSM (Abdulaal et al., 2021), SWaT (Mathur & Tippenhauer, 2016), GECCO and SWAN from NeurIPS-TS (Lai et al., 2021), and UCR (Wu & Keogh, 2021). These datasets cover spacecraft telemetry, server monitoring, industrial control, space weather, and diverse natural signals. Detailed dataset descriptions and statistics are provided in Appendix B.1.

Baselines. We extensively compare TEDA against 18 competitive baselines, including the linear transformation-based method: OCSVM (Schölkopf et al., 1999); the outlier-based method: LODA (Pevnỳ, 2016); the density estimation-based methods: HBOS (Goldstein & Dengel, 2012), LOF (Breunig et al., 2000); the neural network-based methods: AutoEncoder (AE) (Sakurada & Yairi, 2014), DAGMM (Zong et al., 2018), LSTM (Hundman et al., 2018a), OmniAnomaly (Omni) (Su et al., 2019), CAE-Ensemble (CAE) (Campos et al., 2021), Anomaly Transformer (AT) (Xu et al., 2021), DCdetector (DC) (Yang et al., 2023), GPT4TS (Zhou et al., 2023), ModernTCN (Luo & Wang, 2024), CATCH (Wu et al., 2025), MROT (Montesuma et al., 2025), ALoRa (Shimillas et al., 2026), PaAno (Park & Kang, 2026), and CrossAD (Li et al., 2025). Further details concerning baselines are available in Appendix B.2.

Metrics. We adopt Label-based metric: Accuracy (Acc), standard F1-score (Std-F1), and Affiliation-F1 (Aff-F1) (Huet et al., 2022); and Score-based metrics: VUS-ROC (VUS-R), VUS-PR (VUS-P), AUC-ROC (AUC-R), AUC-PR (AUC-P), Range-AUC-ROC (R-A-R), and Range-AUC-PR (R-A-P) (Paparrizos et al., 2022a). Detailed definitions are provided in Appendix B.3.

4.2 DETECTION RESULTS

Main results. We extensively compare our proposed TEDA method with 18 competitive baselines on six real-world datasets, as shown in Table 1. Our method TEDA achieves superior performance under the VUS-PR and AUC-ROC metrics across all datasets. This shows that our method can distinguish normal samples from abnormal samples well. It also keeps stable detection performance under different thresholds, which is important for real-world anomaly detection. These results show the effectiveness of our time-frequency distribution-aware design. With learnable distribution centers and OT-based anomaly scoring, the model can measure how far a test sample deviates from normal distribution patterns. This shows the value of modeling time series anomalies from a distribution perspective.

Table 1: Results on the six real-world datasets. All results are averaged over five independent runs with different random seeds. The VUS-P and AUC-R are the VUS-PR and AUC-ROC, respectively, where higher values indicate better performance. The best results are in **bold**, and the second-best results are underlined.

Dataset	SWAT		SWAN		GECCO		PSM		MSL		UCR	
Metric	VUS-P	AUC-R	VUS-P	AUC-R	VUS-P	AUC-R	VUS-P	AUC-R	VUS-P	AUC-R	VUS-P	AUC-R
OCSVM	0.3705	0.8123	0.8796	0.6085	0.0370	0.7286	0.5022	0.7151	0.1660	0.5035	0.0804	0.6257
LODA	0.4777	0.8396	0.8670	0.5708	0.0588	0.6665	0.4715	0.6818	0.1419	0.5773	0.0997	0.7622
LOF	0.1261	0.4219	0.8738	0.6957	0.0378	0.6405	0.4553	0.6952	0.1479	0.5742	0.0695	0.7088
HBOS	0.4491	0.8418	0.9013	0.7073	0.0358	0.6465	0.4392	0.6769	0.1669	0.5987	0.0646	0.7319
AE	0.4158	0.8227	0.9142	0.6649	0.5653	0.9534	0.5237	0.7160	0.2782	0.7782	0.0491	0.8547
DAGMM	0.0746	0.1748	0.8882	0.7410	0.0149	0.2914	0.3711	0.6216	0.1324	0.4997	0.0170	0.6858
LSTM	0.3998	0.8247	0.9018	0.6762	0.0868	0.8325	0.4826	0.6691	0.3008	0.7746	0.0609	0.7347
Omni	0.1792	0.6715	0.9130	0.7436	0.3314	0.9236	0.4395	0.6952	0.2788	0.7447	0.0623	0.7299
CAE	0.3624	0.8170	0.9173	0.6601	0.1320	0.7189	0.4901	0.6758	0.2454	0.7098	0.1135	0.7944
AT	0.4534	0.7849	0.8092	0.5084	0.0218	0.4228	0.3272	0.5254	0.1250	0.3683	0.0279	0.5621
DC	0.1232	0.5219	0.8300	0.5237	0.0215	0.3133	0.3075	0.4941	0.1284	0.4787	0.0034	0.3856
GPT4TS	0.1581	0.3180	0.9093	0.6069	0.5372	0.9743	0.4370	0.5666	0.2884	0.7746	0.0639	0.8905
ModernTCN	0.1188	0.2923	0.9313	0.7307	0.4680	0.9840	0.5293	0.6396	0.3057	0.7786	0.0768	0.9108
CATCH	0.3697	0.7583	0.9112	0.5259	0.3820	0.9314	0.4616	0.6075	0.2992	0.7734	0.1350	0.9629
MROT	0.4303	0.8391	0.8983	0.7231	0.0453	0.7057	0.3455	0.5583	0.1867	0.5732	0.1290	0.7508
CrossAD	0.4768	0.8464	0.9171	0.6011	0.6213	0.9866	0.5596	0.6810	0.3138	0.7838	0.1631	0.8405
PaAno	0.4673	0.8451	0.8310	0.4723	0.0276	0.5572	0.3207	0.5364	0.1438	0.5903	0.0034	0.6799
ALoRa	0.4516	0.8224	0.9118	0.7399	0.0879	0.7864	0.4583	0.6598	0.1615	0.5468	0.1155	0.7379
TEDA	**0.5046**	**0.8484**	**0.9414**	**0.7512**	**0.6340**	**0.9872**	**0.5979**	**0.7216**	**0.3176**	**0.7839**	**0.2964**	**0.9871**

Multi-metrics. For a comprehensive comparison, we also evaluate our proposed method, TEDA, using more metrics, as shown in Table 2. Specifically, we compare TEDA with three recognized advanced methods: CATCH, CrossAD, and ALoRa. It is shown that TEDA achieves consistently superior results across multiple metrics, demonstrating its excellent anomaly detection capabilities.

Table 2: Multi-metric results in the three real-world datasets. The higher values for all metrics represent better performance. The best ones are in **bold**, and the second ones are underlined.

Dataset	Method	Acc	Std-F1	Aff-F1	VUS-R	VUS-P	AUC-R	AUC-P	R-A-R	R-A-P
PSM	**CATCH**	0.4496	0.4374	0.8254	0.6659	0.4616	0.6075	0.3892	0.6995	0.5158
	CrossAD	0.6853	0.4974	0.8450	0.7302	0.5596	0.6810	0.4906	0.7564	0.6032
	ALoRa	0.4182	0.4750	0.7367	0.6390	0.4583	0.6598	0.4706	0.6459	0.4798
	TEDA	**0.6923**	**0.5207**	**0.8476**	**0.7643**	**0.5979**	**0.7216**	**0.5258**	**0.7852**	**0.6421**
SWAN	**CATCH**	0.3249	0.4905	0.7105	0.9443	0.9112	0.5259	0.4267	0.9480	0.9464
	CrossAD	0.3252	0.4906	0.7105	0.9499	0.9171	0.6011	0.5024	0.9408	0.9408
	ALoRa	0.6565	0.5736	0.7137	**0.9542**	0.9118	0.7399	0.6340	0.9253	0.9253
	TEDA	**0.7658**	**0.5937**	**0.7145**	0.9503	**0.9414**	**0.7512**	**0.6580**	**0.9634**	**0.9676**
GECCO	**CATCH**	0.9817	0.3539	0.8579	0.9840	0.3820	0.9314	0.2518	0.9658	0.3871
	CrossAD	0.9877	0.5705	0.9150	0.9948	0.6213	0.9866	0.5515	**0.9888**	**0.6381**
	ALoRa	**0.9914**	0.4183	0.8964	0.7822	0.0879	0.7864	0.3156	0.6450	0.0725
	TEDA	0.9890	**0.5918**	**0.9314**	**0.9951**	**0.6340**	**0.9872**	**0.5679**	0.9872	0.6293

4.3 Model Analysis

Ablation studies. To verify the effectiveness of each key design in TEDA, we conduct ablation studies on three selected datasets, as reported in Table 3. Row 1 removes OT-based distribution modeling and uses only the reconstruction objective and score. Compared with Row 1, TEDA (Row 8) improves average VUS-PR/AUC-ROC by 4.80%/8.89%. Rows 2 and 3 evaluate the differentiated OT strategy by removing entropy regularization during training or retaining it during inference. Compared with them, TEDA improves the average VUS-PR/AUC-ROC by 1.33%/1.16% and 1.53%/3.07%, respectively, supporting the use of entropy-regularized training and entropy-free inference. Rows 4 and 5 use only the reconstruction and OT-induced scores at inference, respectively. TEDA improves upon them by 3.50%/8.80% and 35.35%/9.07%, confirming the complementarity of point-wise reconstruction errors and distributional deviations. Rows 6 and 7 remove the frequency and temporal OT components, respectively. TEDA further improves upon them by 1.26%/2.10% and 1.49%/1.85%, demonstrating the complementarity of temporal and frequency distribution modeling.

Table 3: Ablation study results. All variants include the reconstruction loss during training.

	Variant	T-OT	F-OT	Train OT	Test OT	AS Score	PSM		GECCO		SWAN		Avg.	
							VUS-P	AUC-R	VUS-P	AUC-R	VUS-P	AUC-R	VUS-P	AUC-R
1	**w/o OT**	×	×	–	–	Rec.	0.5477	0.6722	0.6095	0.9859	0.9169	0.6009	0.6913	0.7530
2	**w/o Training Entropy**	✓	✓	w/o Ent.	w/o Ent.	Rec.+T-OT+F-OT	0.5826	0.7012	0.6265	0.9866	0.9355	0.7442	0.7149	0.8106
3	**w/ Inference Entropy**	✓	✓	w/ Ent.	w/ Ent.	Rec.+T-OT+F-OT	0.5801	0.6976	0.6229	0.9868	0.9375	0.7024	0.7135	0.7956
4	**Only Rec. Score**	✓	✓	w/ Ent.	–	Rec.	0.5590	0.6792	0.6237	0.9795	0.9170	0.6025	0.6999	0.7537
5	**Only OT Score**	✓	✓	w/ Ent.	w/o Ent.	T-OT+F-OT	0.4837	0.6624	0.2086	0.9310	0.9131	0.6618	0.5352	0.7518
6	**w/o Frequency OT**	✓	×	w/ Ent.	w/o Ent.	Rec.+T-OT	0.5868	0.7069	0.6291	0.9856	0.9302	0.7168	0.7154	0.8031
7	**w/o Temporal OT**	×	✓	w/ Ent.	w/o Ent.	Rec.+F-OT	0.5789	0.6998	0.6274	0.9849	0.9351	0.7306	0.7138	0.8051
8	**TEDA**	✓	✓	w/ Ent.	w/o Ent.	Rec.+T-OT+F-OT	**0.5979**	**0.7216**	**0.6340**	**0.9872**	**0.9414**	**0.7512**	**0.7244**	**0.8200**

Parameter sensitivity. We also study the parameter sensitivity of the TEDA. We examine the inverse entropy-regularization coefficient $\lambda = 1/\epsilon$, which controls the sharpness of the Sinkhorn transport plan. A smaller λ produces smoother transport plans, whereas a larger λ leads to sharper assignments. We vary $\lambda \in \{1, 5, 10, 20, 50\}$ on MSL and PSM. As shown in Fig. 3, MSL achieves its best VUS-PR and AUC-ROC of 0.3131 and 0.7835 at $\lambda = 10$, while PSM obtains its best results of 0.5480 and 0.6781 at $\lambda = 5$. Performance declines when the assignments become overly smooth or sharp; for example, VUS-PR on PSM decreases to 0.4886 at $\lambda = 50$. These results show that moderate entropy smoothing balances stable centroid learning and discriminative distributional matching. More sensitivity analyses are provided in Appendix C.4.

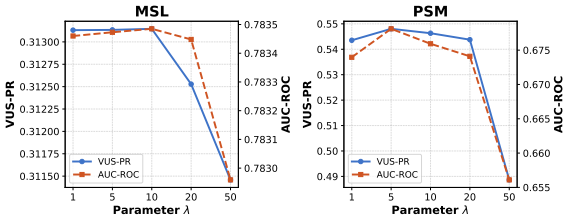


Figure 3: Sensitivity to the inverse entropy-regularization parameter λ.

Visual analysis. We visualize the anomaly scores of TEDA on five datasets. As shown in Fig. 4, TEDA assigns higher anomaly scores to the ground-truth anomaly regions and keeps low scores in most normal regions. It captures sustained level or state changes on MSL and SWaT and produces localized score peaks for abrupt anomalies on PSM, GECCO, and UCR. The anomaly score increases near the abnormal transition and remains high within the abnormal region. These results demonstrate that TEDA can distinguish diverse anomalous patterns from complex normal variations.

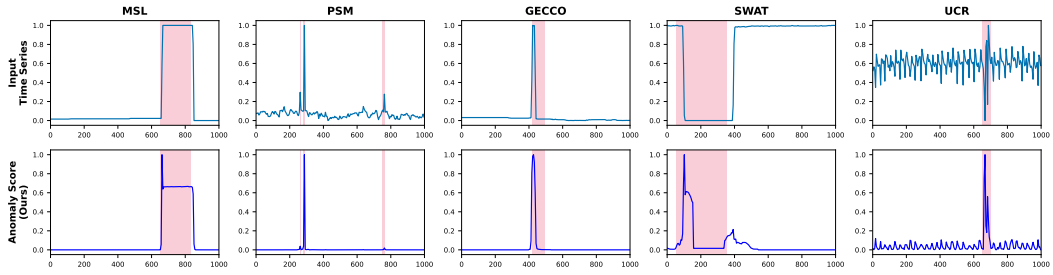


Figure 4: Visualization of TEDA anomaly scores. The top row shows the input series with ground-truth anomaly intervals (highlighted), and the bottom row shows the corresponding anomaly scores.

5 Conclusion

In this paper, we proposed TEDA, a time-frequency distribution-aware method for unsupervised time series anomaly detection. Unlike reconstruction-based methods that mainly rely on reconstruction errors, TEDA models the latent distribution of normal time series. It learns distributional centroids through complementary temporal and frequency branches and uses optimal transport to measure deviations from normal patterns. In addition, it introduces a training-inference differentiated OT strategy, where entropy regularization preserves smooth transport plans during training, and entropy-free sharp assignment produces more discriminative anomaly scores during inference. Extensive experiments demonstrate strong performance against competitive baselines, while ablation studies and visualizations validate the effectiveness of its key designs. Our code is available at `https://anonymous.4open.science/r/TEDA-8E82/`.

AI USE STATEMENT

In this work, we used generative AI tools to assist with language polishing. All AI-assisted text were carefully reviewed and verified by the authors. The research questions, methodology, experimental design, analysis, and scientific claims were developed independently by the authors. We take full responsibility for the final content of this work.

ETHICS STATEMENT

This work focuses on methodological research for unsupervised time series anomaly detection. Our experiments use existing public benchmark datasets, without involving human subjects or the intentional collection of personally identifiable information. The proposed method is intended for time series anomaly detection research, and we have not identified any direct ethical concerns specific to the experiments conducted in this study.

REPRODUCIBILITY STATEMENT

The model architecture and training strategy of TEDA are described in Section 3 , with experimental settings, dataset details, and implementation provided in Appendix B. We report the configurations and evaluation protocols required to reproduce the main experiments. The source code is available at `https://anonymous.4open.science/r/TEDA-8E82/` and necessary scripts will be released.

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.

Amina Alaoui-Belghiti, Sylvain Chevallier, and Eric Monacelli. Unsupervised anomaly detection using optimal transport for predictive maintenance. In *International Conference on Artificial Neural Networks*, pp. 686–697. Springer, 2019.

Amina Alaoui-Belghiti, Sylvain Chevallier, Eric Monacelli, Guillaume Bao, and Eric Azabou. Semi-supervised optimal transport methods for detecting anomalies. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2997–3001. IEEE, 2020.

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.

Roel Bouman and Tom Heskes. Autoencoders for anomaly detection are unreliable. *arXiv preprint arXiv:2501.13864*, 2025.

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.

David Campos, Tung Kieu, Chenjuan Guo, Feiteng Huang, Kai Zheng, Bin Yang, and Christian S Jensen. Unsupervised time series outlier detection with diversity-driven convolutional ensembles. *Proceedings of the VLDB Endowment*, 15(3):611–623, 2021.

Feiyi Chen, Yingying Zhang, Zhen Qin, Lunting Fan, Renhe Jiang, Yuxuan Liang, Qingsong Wen, and Shuiguang Deng. Learning multi-pattern normalities in the frequency domain for efficient time series anomaly detection. In *2024 IEEE 40th International Conference on Data Engineering (ICDE)*, pp. 747–760. IEEE, 2024.

Andrew A Cook, Göksel Mısırlı, and Zhong Fan. Anomaly detection for iot time-series data: A survey. *IEEE Internet of Things Journal*, 7(7):6481–6494, 2019.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26, 2013.

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.

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.

Markus Goldstein and Andreas Dengel. Histogram-based outlier score (hbos): A fast unsupervised anomaly detection algorithm. *KI-2012: poster and demo track*, 1:59–63, 2012.

Dong Gong, Lingqiao Liu, Vuong Le, Budhaditya Saha, Moussa Reda Mansour, Svetha Venkatesh, and Anton van den Hengel. Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1705–1714, 2019.

Adam Goodge, Bryan Hooi, See Kiong Ng, and Wee Siong Ng. Ares: Locally adaptive reconstruction-based anomaly scoring. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 193–208. Springer, 2022.

Alexis Huet, Jose Manuel Navarro, and Dario Rossi. Local evaluation of time series anomaly detection algorithms. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 635–645, 2022.

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, 2018a.

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, 2018b.

Kwei-Herng Lai, Daochen Zha, Junjie Xu, Yue Zhao, Guanchu Wang, and Xia Hu. Revisiting time series outlier detection: Definitions and benchmarks. In *Thirty-fifth conference on neural information processing systems datasets and benchmarks track (round 1)*, 2021.

Tian Lan, Hao Duong Le, Jinbo Li, Wenjun He, Meng Wang, Chenghao Liu, and Chen Zhang. Towards foundation models for zero-shot time series anomaly detection: Leveraging synthetic data and relative context discrepancy. *arXiv preprint arXiv:2509.21190*, 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. *Advances in Neural Information Processing Systems*, 38:137959–137985, 2025.

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

Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. In *2008 eighth ieee international conference on data mining*, pp. 413–422. IEEE, 2008.

Qinghua Liu and John Paparrizos. The elephant in the room: Towards a reliable time-series anomaly detection benchmark. *Advances in Neural Information Processing Systems*, 37:108231–108261, 2024.

Donghao Luo and Xue Wang. Moderntcn: A modern pure convolution structure for general time series analysis. In *ICLR*, 2024.

Aditya P Mathur and Nils Ole Tippenhauer. Swat: A water treatment testbed for research and training on ics security. In *2016 international workshop on cyber-physical systems for smart water networks (CySWater)*, pp. 31–36. IEEE, 2016.

Eduardo Fernandes Montesuma, Adel El Habazi, and Fred Ngole Mboula. Unsupervised anomaly detection through mass repulsing optimal transport. *arXiv preprint arXiv:2502.12793*, 2025.

Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. *arXiv preprint arXiv:2211.14730*, 2022.

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

John Paparrizos, Yuhao Kang, Paul Boniol, Ruey S Tsay, Themis Palpanas, and Michael J Franklin. Tsb-uad: An end-to-end benchmark suite for univariate time-series anomaly detection. *Proc. VLDB Endow.*, 15(8):1697–1711, 2022b.

Jinju Park and Seokho Kang. Paano: Patch-based representation learning for time-series anomaly detection. *ICLR*, 2026.

Tomáš Pevnỳ. Loda: Lightweight on-line detector of anomalies. *Machine Learning*, 102(2):275–304, 2016.

Gabriel Peyré and Marco Cuturi. Computational optimal transport with applications to data sciences. *Foundations and trends® in machine learning*, 11(5-6):355–607, 2019.

Xiangfei Qiu, Zhe Li, Wanghui Qiu, Shiyan Hu, Lekui Zhou, Xingjian Wu, Zhengyu Li, Chenjuan Guo, Aoying Zhou, Zhenli Sheng, Jilin Hu, Christian S. Jensen, and Bin Yang. TAB: unified benchmarking of time series anomaly detection methods. *Proc. VLDB Endow.*, 18(9):2775–2789, 2025.

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.

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.

Qichao Shentu, Beibu Li, Kai Zhao, Yang Shu, Zhongwen Rao, Lujia Pan, Bin Yang, and Chenjuan Guo. Towards a general time series anomaly detector with adaptive bottlenecks and dual adversarial decoders. In *International Conference on Learning Representations*, volume 2025, pp. 81358–81381, 2025.

Charalampos Shimillas, Kleanthis Malialis, Konstantinos Fokianos, and Marios M Polycarpou. Low rank transformer for multivariate time series anomaly detection and localization. *ICLR*, 2026.

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

Alban Siffer, Pierre-Alain Fouque, Alexandre Termier, and Christine Largouet. Anomaly detection in streams with extreme value theory. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1067–1075, 2017.

Ya Su, Youjian Zhao, Chenhao Niu, Rong Liu, Wei Sun, and Dan Pei. Robust anomaly detection for multivariate time series through stochastic recurrent neural network. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 2828–2837, 2019.

Shreshth Tuli, Giuliano Casale, and Nicholas R Jennings. Tranad: Deep transformer networks for anomaly detection in multivariate time series data. *arXiv preprint arXiv:2201.07284*, 2022.

Cédric Villani et al. *Optimal transport: old and new*, volume 338. Springer, 2009.

Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. Timesnet: Temporal 2d-variation modeling for general time series analysis. *arXiv preprint arXiv:2210.02186*, 2022.

Renjie Wu and Eamonn J Keogh. Current time series anomaly detection benchmarks are flawed and are creating the illusion of progress. *IEEE transactions on knowledge and data engineering*, 35 (3):2421–2429, 2021.

Xingjian Wu, Xiangfei Qiu, Zhengyu Li, Yihang Wang, Jilin Hu, Chenjuan Guo, Hui Xiong, and Bin Yang. Catch: Channel-aware multivariate time series anomaly detection via frequency patching. In *International conference on learning representations*, volume 2025, pp. 17017–17045, 2025.

Jiehui Xu, Haixu Wu, Jianmin Wang, and Mingsheng Long. Anomaly transformer: Time series anomaly detection with association discrepancy. *arXiv preprint arXiv:2110.02642*, 2021.

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.

Chin-Chia Michael Yeh, Yan Zhu, Liudmila Ulanova, Nurjahan Begum, Yifei Ding, Hoang Anh Dau, Diego Furtado Silva, Abdullah Mueen, and Eamonn Keogh. Matrix profile i: all pairs similarity joins for time series: a unifying view that includes motifs, discords and shapelets. In *2016 IEEE 16th international conference on data mining (ICDM)*, pp. 1317–1322. Ieee, 2016.

Chunkai Zhang, Wei Zuo, and Xuan Wang. Reconstruct anomaly to normal: Adversarial learned and latent vector-constrained autoencoder for time-series anomaly detection. *arXiv preprint arXiv:2010.06846*, 2020.

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.

Bo Zong, Qi Song, Martin Renqiang Min, Wei Cheng, Cristian Lumezanu, Daeki Cho, and Haifeng Chen. Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In *International conference on learning representations*, 2018.

A Optimal Transport

Optimal Transport (OT) studies how to transport mass from a source distribution to a target distribution with minimum total cost. Intuitively, if a distribution is viewed as a set of weighted points, OT seeks a transportation strategy that determines how much mass should be moved from each source point to each target point such that the overall transportation cost is minimized.

In the continuous setting, let P and Q be two probability distributions defined on a space $\mathcal{X}$, and let $c(x, y)$ denote the cost of transporting one unit of mass from x to y. The Kantorovich formulation represents the transportation strategy as a joint distribution γ, also called a transport plan, whose two marginals are P and Q. The OT problem is written as

$$\gamma^\star = \arg \inf_{\gamma \in \Gamma(P,Q)} \int_{\mathcal{X}\times\mathcal{X}} c(x, y)\, d\gamma(x, y), \tag{18}$$

where $\Gamma(P, Q)$ denotes the set of all transport plans with marginals P and Q. Unlike the Monge formulation, where each source point is mapped to a single target point, the Kantorovich formulation allows the mass of a source point to be distributed to multiple target points. This makes it more suitable for flexible distribution matching in deep neural networks.

In practice, the analytical forms of P and Q are usually unavailable, and we only have access to samples drawn from them. Therefore, empirical distributions are used to approximate the true distributions. Given source samples $\{x_i\}_{i=1}^n$ and target samples $\{y_j\}_{j=1}^m$, with masses p_i and q_j, we have

$$\hat{P}(x) = \sum_{i=1}^{n} p_i \delta(x - x_i), \qquad \hat{Q}(y) = \sum_{j=1}^{m} q_j \delta(y - y_j). \tag{19}$$

Substituting the empirical distributions into the OT problem leads to a discrete linear program:

$$\gamma^\star = \arg \min_{\gamma \in \Gamma(p,q)} \langle \gamma, \mathbf{C} \rangle = \arg \min_{\gamma \in \Gamma(p,q)} \sum_{i=1}^{n} \sum_{j=1}^{m} \gamma_{ij} \mathbf{C}_{ij}, \tag{20}$$

where $\mathbf{C}_{ij} = c(x_i, y_j)$ is the cost matrix, and γ_{ij} denotes the amount of mass transported from the source sample x_i to the target sample y_j. A valid transport plan satisfies the marginal constraints

$$\gamma \mathbf{1} = p, \qquad \gamma^\top \mathbf{1} = q, \qquad \gamma_{ij} \geq 0. \tag{21}$$

That is, the total mass sent out by each source point must equal its own mass, and the total mass received by each target point must equal its target mass. The resulting OT cost $\langle \gamma, \mathbf{C} \rangle$ can be interpreted as the matching cost between two discrete distributions.

Directly solving the above linear program is computationally expensive. To improve efficiency, an entropy regularization term is commonly added to the OT objective, yielding entropic OT:

$$\gamma_\epsilon^\star = \arg \min_{\gamma \in \Gamma(p,q)} \langle \gamma, \mathbf{C} \rangle + \epsilon \sum_{i=1}^{n} \sum_{j=1}^{m} \gamma_{ij} (\log \gamma_{ij} - 1), \tag{22}$$

where ϵ controls the regularization strength. Since the total transport mass is fixed, the regularization term is equivalent to $-\epsilon H(\gamma)$ up to a constant, where $H(\gamma)$ denotes the entropy of the transport plan.The entropy regularization smooths the transport plan and enables efficient optimization through Sinkhorn iterations. Specifically, a transport kernel is first constructed as

$$K_{ij} = \exp(-\mathbf{C}_{ij}/\epsilon), \tag{23}$$

and row and column normalizations are alternately applied so that the transport plan gradually satisfies the marginal constraints. In our implementation, we adopt a Sinkhorn-style procedure that constructs an exponential kernel from the distance matrix and performs row-column normalization to obtain an approximate transport plan.

In our framework, the source distribution is not formed by raw samples, but by time series representations learned by the model. The target distribution is not another set of external samples, but a set of learnable normal distributional centroids. Specifically, for patch-based OT, the source samples are temporal patch representations and the target samples are learnable patch centroids. For

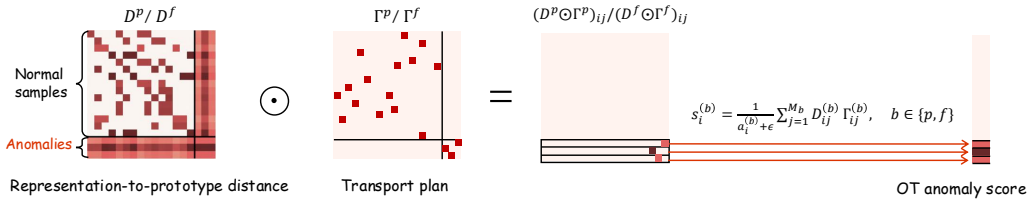


Figure 5: Illustration of OT-based distribution anomaly scoring. For each branch $b \in \{p, f\}$, the representation-to-prototype distance matrix $D^{(b)}$ and the OT transport plan $\Gamma^{(b)}$ are combined to compute the weighted transport cost. Here, $\Gamma^{(b)}$ is used to illustrate the transport-based distribution matching mechanism. During inference, entropy regularization is removed and $\Gamma^{(b)}$ becomes an entropy-free sharp assignment matrix, as detailed in Section 3.3.

frequency-based OT, the source samples are frequency tokens and the target samples are learnable frequency centroids. Let $Z = \{z_i\}_{i=1}^n$ denote source representations and $M = \{m_j\}_{j=1}^K$ denote learnable distributional centroids. The cost matrix is defined as

$$D_{ij} = d(z_i, m_j), \tag{24}$$

where $d(\cdot, \cdot)$ can be cosine distance or normalized Euclidean distance. Based on this cost matrix, the OT module computes a transport plan Γ and the OT loss:

$$\mathcal{L}_{OT} = \langle \Gamma, D \rangle = \sum_{i=1}^{n} \sum_{j=1}^{K} \Gamma_{ij} D_{ij}. \tag{25}$$

As illustrated in Fig. 5, for each branch $b \in \{p, f\}$, the assignment matrix weights the corresponding representation-to-centroid distances in D^b, which are then aggregated to obtain the OT-based distributional cost. Normal representations can be matched with the learned normal distributional centroids at low costs, whereas anomalous representations generally have larger distances from their nearest normal centroids and therefore produce higher distributional scores. Thus, OT provides distributional constraints during training and distributional deviation signals during inference.

B Experimental Details

B.1 Datasets

We conduct experiments on six widely used multivariate time series anomaly detection datasets, including MSL, PSM, SWaT, GECCO, SWAN, and UCR. These datasets cover spacecraft telemetry, server monitoring, industrial control, space weather, and diverse natural signals. Their statistical information is summarized in Table 4.

MSL (Mars Science Laboratory) is a spacecraft telemetry dataset released by NASA. It contains measurements collected from sensors and actuators monitoring the operational status of the Mars rover. Following previous studies, we retain the first continuous dimension for evaluation.

PSM (Pooled Server Metrics) contains multiple system metrics collected from eBay server machines. Its labeled anomalies correspond to abnormal server behavior and service failures.

SWaT (Secure Water Treatment) contains multivariate sensor and actuator measurements collected from a continuously operating water-treatment testbed. The anomalies represent attacks and abnormal operating states in the industrial control system.

GECCO and **SWAN** are obtained from the NeurIPS 2021 Time Series Benchmark. GECCO contains multivariate measurements from a water-treatment system, while SWAN contains telemetry measurements from a space-weather monitoring system. Both datasets contain diverse temporal patterns and anomaly scenarios.

UCR refers to the complete UCR Time Series Anomaly Archive, which contains 250 univariate time series from diverse domains. Each time series contains one labeled anomalous interval.

Table 4: Statistics of the datasets.

Dataset	Domain	Dimension	Window	Training	Validation	Test (labeled)	AR (%)
MSL	Spacecraft	1	96	46,653	11,664	73,729	10.50
PSM	Server Machine	25	192	105,984	26,497	87,841	27.80
SWaT	Water Treatment	31	192	396,000	99,000	449,919	12.10
GECCO	Water Treatment	9	128	55,408	13,852	69,261	1.25
SWAN	Space Weather	38	192	48,000	12,000	60,000	23.80
UCR	Natural	1	–	1,790,680	447,670	6,143,541	0.60

B.2 BASELINES

We compare TEDA with 18 baselines covering classical anomaly detection, deep reconstruction and representation learning, and recent state-of-the-art (SOTA) time series anomaly detection methods.

Classical anomaly detection methods. OCSVM (Schölkopf et al., 1999) learns a one-class decision boundary to separate normal observations from potential anomalies. LODA (Pevnỳ, 2016) combines multiple one-dimensional histogram estimators obtained through random projections. HBOS (Goldstein & Dengel, 2012) estimates feature-wise histograms and combines their outlier scores, while LOF (Breunig et al., 2000) identifies anomalies according to deviations in local data density.

Deep anomaly detection methods. AE Sakurada & Yairi (2014) detects anomalies using reconstruction errors produced by an autoencoder. DAGMM (Zong et al., 2018) combines representation learning with Gaussian-mixture-based density estimation. LSTM (Hundman et al., 2018a) models temporal dynamics through recurrent prediction and nonparametric thresholding. Omni-Anomaly (Su et al., 2019) employs a stochastic recurrent latent-variable model for multivariate time series. CAE-Ensemble (Campos et al., 2021) combines multiple convolutional autoencoders with diversity-driven training. Anomaly Transformer (Xu et al., 2021) identifies anomalies using association discrepancy, whereas DCdetector (Yang et al., 2023) learns discriminative representations through dual-attention contrastive learning. GPT4TS (Zhou et al., 2023) adapts a pretrained language-model backbone to time series analysis, and ModernTCN (Luo & Wang, 2024) uses a purely convolutional architecture to model temporal and channel dependencies.

Recent time series anomaly detection methods. CATCH (Wu et al., 2025) models inter-channel dependencies through frequency-domain patching. MROT (Montesuma et al., 2025) performs unsupervised anomaly detection using mass-repulsing optimal transport. ALoRa (Shimillas et al., 2026) adopts a low-rank Transformer architecture for multivariate anomaly detection and localization. PaAno (Park & Kang, 2026) learns patch-based representations for time series anomaly detection. CrossAD (Li et al., 2025) captures cross-scale associations and models dependencies across adjacent windows. These methods represent the most relevant recent comparisons for evaluating TEDA's time-frequency and distribution-aware designs.

B.3 METRICS

Recent studies have shown that point adjustment (PA) (Yang et al., 2023; Xu et al., 2021), which is widely used in time series anomaly detection, may lead to overly optimistic results, and even random anomaly scores can achieve competitive performance under this protocol (Li et al., 2025; Huet et al., 2022). Recent benchmark studies, such as TSB-UAD (Paparrizos et al., 2022b) and TAB (Qiu et al., 2025), further highlight the importance of standardized evaluation protocols and fair comparison for time series anomaly detection. To reduce the bias introduced by PA, some recent methods adopt Affiliation-based metrics (Huet et al., 2022) and VUS-based metrics (Paparrizos et al., 2022a), which evaluates anomaly detection from an event-level perspective by measuring the directed distance between predicted anomaly events and ground-truth anomaly events. In addition, TSB-AD (Liu & Paparrizos, 2024) demonstrates that VUS-PR is the most robust (insensitive to lag), accurate (unbiased and effective across scenarios), and fair (consistent in similar situations) evaluation metric for the time series anomaly detection task, since it is less sensitive to detection delay and can provide stable evaluation across different anomaly scenarios. Therefore, in our main results, we report

VUS-PR and AUC-ROC as the primary evaluation metrics. Meanwhile, to ensure a comprehensive comparison, we also evaluate the SOTA methods using more metrics, including Accuracy (Acc), Standard-F1 (Std-F1), Affiliation-F1 (Aff-F1), VUS-ROC (VUS-R), VUS-PR (VUS-P), AUC-ROC (AUC-R), AUC-PR (AUC-P), Range-AUC-ROC (R-A-R), and Range-AUC-PR (R-A-P) (Paparrizos et al., 2022a).

B.4 Data Processing and Implementation Details

We use the predefined training and test splits of each benchmark. For MSL, PSM, SWaT, GECCO, and SWAN, the final 20% of each original training sequence is chronologically reserved as the validation set for early stopping. This chronological split preserves the temporal order of the observations. Anomaly labels are used only for final evaluation and are not used for model optimization, checkpoint selection, or anomaly-threshold estimation.

In our experiments, we set the dimension of hidden states and representations as 128, the number of attention heads as 4, and each attention dimension as 32. Both the encoder and the decoder consist of two attention layers. The dropout rates of attention, projection, and feed-forward networks are all set to 0.1. The number of learnable distribution centroids in both the time-domain and frequency-domain OT branches is set to 32. We use a sliding-window strategy to process long time series. The stride is set to 1 during training. In the validation and testing stages, we use non-overlapping windows, where the stride is equal to the window size. For all methods, the window sizes of MSL, GECCO, PSM, SWAN, and SWaT are set to 96, 128, 192, 192, and 192, following previous works (Li et al., 2025; Liu & Paparrizos, 2024; Wu et al., 2025) to ensure fair comparison. We use the Adam optimizer with an initial learning rate of 10^{-4} and set the batch size to 128. We do not use drop-last in any data loading stage to ensure that all data are evaluated. All experiments are repeated five times with different random seeds $\{2021, 2022, 2023, 2024, 2025\}$. We use identical data splits and hyperparameter settings across all runs and report the average performance. After obtaining the anomaly scores, we use unsupervised SPOT method (Siffer et al., 2017) to determine the anomaly threshold following previous works (Li et al., 2025; Shentu et al., 2025). All experiments are implemented in PyTorch and run on an NVIDIA GeForce RTX 3090 24GB GPU.

C Additional Experiments Results

C.1 TSB benchmark

TSB-AD benchmark (Liu & Paparrizos, 2024) comprises 40 diverse, publicly available datasets. We provide a comparison with the state-of-the-art methods from TSB-AD. In Table 5, we present a comparison of the accuracy between TEDA and other detection algorithms on TSB-AD-U and TSB-AD-M. From the results, it can be found that TEDA demonstrates excellent detection performance across all metrics, whether in TSB-AD-U or TSB-AD-M. For more details about the baselines and metrics can be seen in TSB-AD benchmark.

C.2 UCR benchmark

The UCR Anomaly Archive consists of 250 univariate time series datasets, each containing a single labeled anomalous interval in the test sequence (Wu & Keogh, 2021). We further evaluate TEDA on this benchmark to assess its anomaly localization capability. For each dataset, TEDA computes an anomaly score for every timestamp in the test set, and the timestamps are ranked in descending order according to their anomaly scores. Following the evaluation protocol adopted in prior works such as Timer and DADA, detection is considered successful if the timestamp corresponding to a given quantile α falls within the ground-truth anomalous interval. As shown in Fig. 6, we report the detection performance of TEDA under different quantile thresholds. The left panel shows the number of UCR datasets on which the anomalous interval is successfully detected at quantile levels of 3% and 10%, where a larger number indicates stronger anomaly localization capability. The right panel presents the cumulative distribution of the normalized localization rank α across all UCR datasets, together with the average localization ranks of different methods. A lower average quantile indicates that the ground-truth anomaly tends to appear earlier in the anomaly-score ranking, reflecting better anomaly localization performance.

Table 5: Comparison of results on TSB-AD-U and TSB-AD-M under different metrics. The best ones are in bold and the second ones are underlined.

	Method	AUC-PR	AUC-ROC	VUS-PR	VUS-ROC	Standard-F1	PA-F1	Event-based-F1	R-based-F1	Affiliation-F1
TSB-AD-U	TEDA	**0.42**	**0.77**	**0.43**	**0.82**	**0.44**	**0.81**	**0.68**	**0.41**	**0.89**
	Sub-PCA	0.37	0.71	0.42	0.76	0.42	0.56	0.49	0.41	0.85
	SubShapeAD	0.35	0.74	0.40	0.76	0.39	0.58	0.46	0.40	0.83
	POLY	0.31	0.73	0.39	0.76	0.37	0.53	0.45	0.35	0.85
	Series2Graph (FT)	0.33	0.79	0.39	0.80	0.38	0.65	0.50	0.35	0.85
	MOMENTAD (ZS)	0.30	0.68	0.38	0.75	0.35	0.61	0.49	0.36	0.86
	KMeansAD	0.32	0.74	0.37	0.76	0.37	0.56	0.44	0.38	0.82
	USAD	0.32	0.66	0.36	0.71	0.37	0.50	0.43	0.40	0.84
	Sub-KNN	0.27	0.76	0.35	0.79	0.34	0.61	0.43	0.32	0.84
	MatrixProfile	0.26	0.73	0.35	0.76	0.33	0.63	0.44	0.32	0.84
	SAND	0.29	0.73	0.34	0.76	0.35	0.56	0.42	0.36	0.81
	CNN	0.33	0.71	0.34	0.79	0.38	0.78	0.66	0.35	0.88
	LSTMAD	0.31	0.68	0.33	0.76	0.37	0.71	0.59	0.34	0.86
	SR	0.32	0.74	0.32	0.81	0.38	0.87	0.67	0.35	0.89
	TimesFM	0.28	0.67	0.30	0.74	0.34	0.84	0.63	0.34	0.89
	IForest	0.29	0.71	0.30	0.78	0.35	0.73	0.56	0.30	0.84
	OmniAnomaly	0.27	0.65	0.29	0.72	0.31	0.59	0.46	0.29	0.83
	Lag-Llama	0.25	0.65	0.27	0.72	0.30	0.77	0.59	0.31	0.88
	Chronos	0.26	0.66	0.27	0.73	0.32	0.83	0.61	0.33	0.88
	TimesNet	0.18	0.61	0.26	0.72	0.24	0.67	0.47	0.21	0.86
	AutoEncoder	0.19	0.63	0.26	0.69	0.25	0.54	0.36	0.28	0.82
	TranAD	0.20	0.57	0.26	0.68	0.25	0.58	0.43	0.25	0.83
	FITS	0.17	0.61	0.26	0.73	0.23	0.65	0.42	0.20	0.86
	Sub-LOF	0.16	0.68	0.25	0.73	0.24	0.57	0.35	0.25	0.82
	OFA	0.16	0.59	0.24	0.71	0.22	0.67	0.45	0.20	0.86
	Sub-MCD	0.15	0.67	0.24	0.72	0.23	0.54	0.32	0.24	0.81
	Sub-HBOS	0.18	0.61	0.23	0.67	0.23	0.60	0.35	0.27	0.79
	Sub-OCSVM	0.16	0.65	0.23	0.73	0.22	0.55	0.32	0.23	0.79
	Sub-IForest	0.16	0.63	0.22	0.72	0.22	0.63	0.34	0.23	0.80
	Donut	0.14	0.56	0.20	0.68	0.20	0.57	0.38	0.20	0.82
	LOF	0.14	0.58	0.17	0.68	0.21	0.62	0.41	0.22	0.79
	AnomalyTransformer	0.08	0.50	0.12	0.56	0.12	0.53	0.34	0.14	0.77
TSB-AD-M	TEDA	**0.36**	**0.74**	**0.35**	**0.76**	**0.39**	**0.79**	0.64	0.37	0.86
	CNN	0.32	0.73	0.31	0.76	0.37	0.78	**0.65**	0.37	0.87
	OmniAnomaly	0.27	0.65	0.31	0.69	0.32	0.55	0.41	0.37	0.81
	PCA	0.31	0.70	0.31	0.74	0.37	0.79	0.59	0.29	0.85
	LSTMAD	0.31	0.70	0.31	0.74	0.36	0.79	0.64	**0.38**	**0.87**
	USAD	0.26	0.64	0.30	0.68	0.31	0.53	0.40	0.37	0.80
	AutoEncoder	0.30	0.67	0.30	0.69	0.34	0.60	0.44	0.28	0.80
	KMeansAD	0.25	0.69	0.29	0.73	0.31	0.68	0.49	0.33	0.82
	CBLOF	0.28	0.67	0.27	0.70	0.32	0.65	0.45	0.31	0.81
	MCD	0.27	0.65	0.27	0.69	0.33	0.46	0.33	0.20	0.76
	OCSVM	0.23	0.61	0.26	0.67	0.28	0.48	0.41	0.30	0.80
	Donut	0.20	0.64	0.26	0.71	0.28	0.52	0.36	0.21	0.81
	RobustPCA	0.24	0.58	0.24	0.61	0.29	0.60	0.42	0.33	0.81
	FITS	0.15	0.58	0.21	0.66	0.22	0.72	0.32	0.16	0.81
	OFA	0.15	0.55	0.21	0.63	0.21	0.72	0.41	0.17	0.83
	EIF	0.19	0.67	0.21	0.71	0.26	0.74	0.44	0.26	0.81
	COPOD	0.20	0.65	0.20	0.69	0.27	0.72	0.41	0.24	0.80
	IForest	0.19	0.66	0.20	0.69	0.26	0.68	0.41	0.24	0.80
	HBOS	0.16	0.63	0.19	0.67	0.24	0.67	0.40	0.24	0.80
	TimesNet	0.13	0.56	0.19	0.64	0.20	0.68	0.32	0.17	0.82
	KNN	0.14	0.51	0.18	0.59	0.19	0.69	0.45	0.21	0.79
	TranAD	0.14	0.59	0.18	0.65	0.21	0.68	0.40	0.21	0.79
	LOF	0.10	0.53	0.14	0.60	0.15	0.57	0.32	0.14	0.76
	AnomalyTransformer	0.07	0.52	0.12	0.57	0.12	0.53	0.33	0.14	0.74

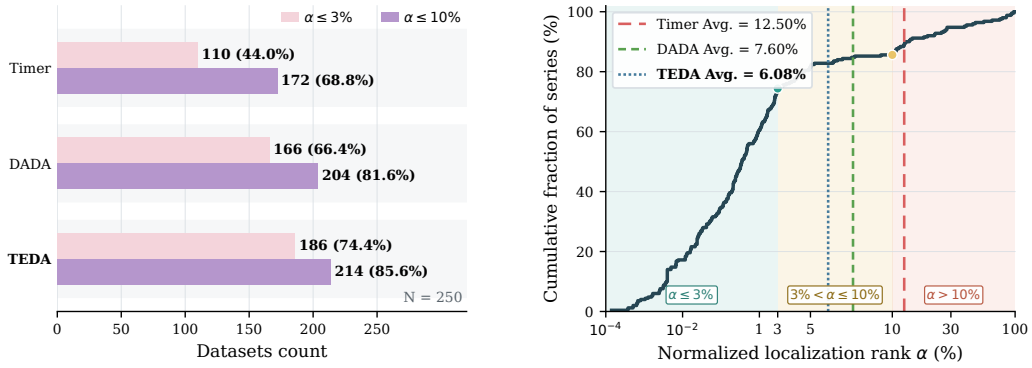


Figure 6: Anomaly localization results on the UCR Anomaly Archive. Left: numbers of datasets successfully localized within $\alpha \leq 3\%$ and $\alpha \leq 10\%$. Right: cumulative distribution of the normalized localization rank α, with the average ranks of different methods.

C.3 Model Efficiency

We compare TEDA with ModernTCN, CATCH, and CrossAD in terms of training time, inference time, model size, and detection performance on the GECCO dataset. As shown in Table 6, ModernTCN achieves the highest computational efficiency but with lower detection performance, while CATCH introduces substantially more parameters. In comparison, TEDA maintains a compact model size and competitive computational efficiency while achieving the best VUS-PR, demonstrating a favorable balance between efficiency and detection performance.

Table 6: Efficiency comparison of different methods on the GECCO dataset. Training time represents the time cost to train the model for 5 epochs with the same batch size 128.

Method	Training Time (s)	Inference Time (s)	Total Params (M)	VUS-PR
ModernTCN	26.30	1.50	0.12	0.4680
CATCH	112.17	16.88	89.93	0.3820
CrossAD	193.22	8.74	1.33	0.6213
TEDA	167.63	8.80	0.94	**0.6340**

C.4 Parameter Sensitivity

Effect of OT loss weight. We study the effect of OT loss weight, which controls the contribution of OT-based distribution learning in the training objective. We vary the OT loss weight and report the results on MSL and PSM. As shown in Fig. 7, the best weight is different across datasets. On MSL, performance generally improves as the weight increases: the best VUS-PR of 0.3163 is obtained at 2.5, while the best AUC-ROC of 0.7843 is achieved at 2.0. This indicates that OT loss helps the model learn more discriminative normal distribution representations. But when the weight further increases to 3.0, VUS-PR no longer improves. On PSM, a smaller weight of 0.05 achieves the best VUS-PR and AUC-ROC of 0.5638 and 0.6809, respectively, whereas larger weights degrade performance. This may be because an excessive OT loss weight overemphasizes distributional centroid matching at the expense of reconstruction and local temporal modeling. These results suggest that OT loss weight should be properly balanced. A too small weight cannot fully use the distribution modeling ability of OT. A too large weight may disturb the balance between reconstruction learning and OT-based distribution learning.

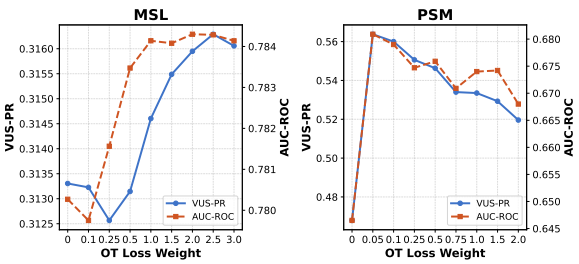


Figure 7: Sensitivity to the OT Loss Weight.

Effect of OT memory size. We investigate the effect of OT memory size, which controls the number of learnable distributional prototypes in the OT module. We vary the memory size among 4, 8, 16, 32, 64, and 128, and report the results on MSL and PSM. As shown in Fig. 8, TEDA achieves the best overall performance when the memory size is set to 32. Specifically, memory size 32 obtains the best VUS-PR and AUC-ROC on MSL, with scores of 0.3124 and 0.7777, respectively. It also achieves the best AUC-ROC on PSM. When the memory size is too small, the model may not have enough capacity to cover diverse normal patterns. In contrast, an overly large memory size may introduce redundant prototypes and weaken the compactness of the learned normal distribution. This can make abnormal samples easier to match with some nearby prototypes. These results suggest that a moderate memory size is important for effective OT-based distribution modeling.

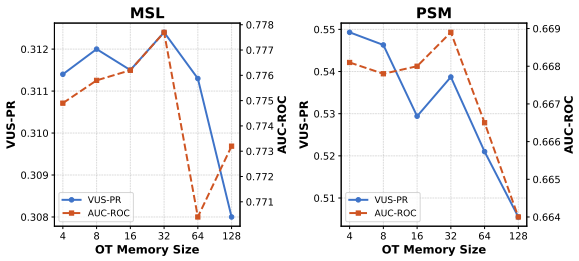


Figure 8: Sensitivity to the OT Memory Size.

Effect of batch size. We investigate the effect of training batch size, which influences the stability of model optimization and distributional centroid learning. We vary the batch size among 16, 32, 64, 128, and 256, and report the results on GECCO and PSM. As shown in Fig. 9, the optimal batch size differs across datasets. GECCO achieves the best VUS-PR and AUC-ROC of 0.6241 and 0.9855 at a batch size of 64, while PSM reaches its best results of 0.5387 and 0.6689 at 128. Performance drops noticeably at 256 on both datasets, suggesting that overly large batches may reduce sensitivity to diverse local distributional patterns. Overall, moderate batch sizes are more suitable for stable distribution-aware learning.

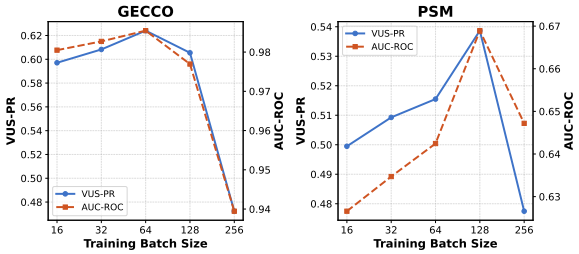


Figure 9: Sensitivity to the training batch size.

C.5 Visualization of complex normal patterns

We visualize a complex normal segment from the GECCO dataset (label = 0) in Fig. 10.

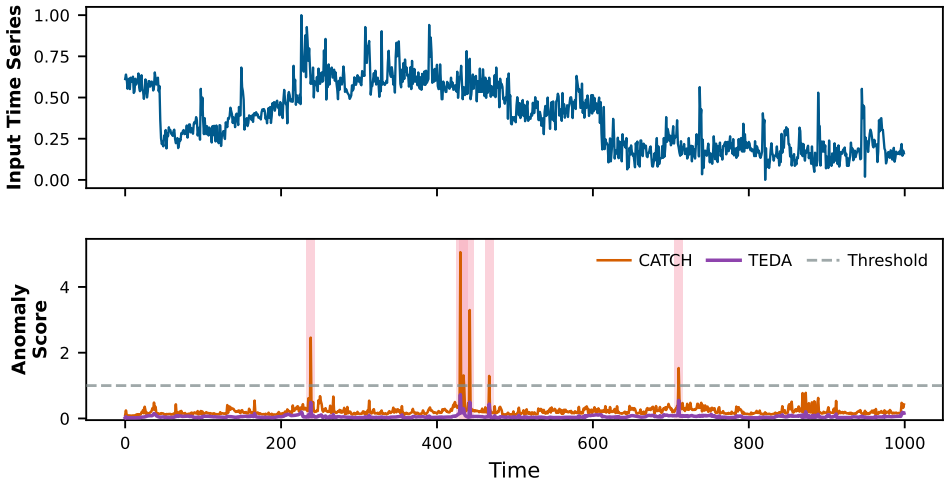


Figure 10: Visualization of a complex normal time series segment. The whole segment is normal according to the ground truth labels. The anomaly scores are normalized by each method's threshold, so the gray dashed line at 1 denotes the decision boundary (threshold) of each method.

Although this segment has complex temporal variations, it still belongs to the normal class. The first row shows the input time series, and the second row shows the anomaly scores of the baseline method (selecting CATCH) and TEDA. As shown in the figure, CATCH produces high anomaly scores in this normal segment, and several scores exceed the threshold. This indicates that CATCH may misclassify complex normal patterns as anomalies. In contrast, TEDA keeps its anomaly score low over the whole segment. Most scores are close to zero and remain below the threshold. This shows that TEDA does not produce false alarms on this complex normal segment. The result supports the core motivation of this paper: complex but high-probability normal patterns should not be misclassified as anomalies. By combining reconstruction error and OT-based distribution deviation scoring, TEDA can better distinguish complex normal patterns from true anomalies.

C.6 LIMITATIONS

Although TEDA demonstrates strong performance across multiple time series anomaly detection benchmarks, our evaluation primarily follows standard offline protocols. While we analyze sensitivity to the number of distributional centroids, alternative frequency representations and adaptive selection of the centroid number remain underexplored. The framework's ability to adapt to continuously evolving normal patterns also warrants further investigation. Systematic studies of these aspects would clarify the effects of different design choices and extend TEDA to a broader range of deployment settings.