Bridging Classification and Reconstruction: Cooperative Time Series Anomaly Detection

Anonymous Author(s)
Affiliation
Address
email

Abstract

Time series anomaly detection (TSAD) has been a perennially important topic in data science due to its wide-ranging applications. In recent years, numerous deep learning-based methods have been proposed for this task. However, latest benchmark studies reveal that deep learning-based methods generally underperform classical data mining methods, sparking skepticism about the practical effectiveness of deep learning in TSAD. This paper addresses these concerns by introducing a novel deep learning framework, COAD, which achieves effective and efficient anomaly detection, thereby reaffirming the potential of deep learning in time series anomaly detection. COAD overcomes the limitations of two promising paradigms by integrating them into a unified design. Extensive experiments on reliable datasets using rigorous evaluation protocols demonstrate that COAD significantly outperforms both deep learning and data mining based methods while achieving orders of magnitude faster inference speed. The code and datasets are available at `https://anonymous.4open.science/r/BRC-E46B` for reproducibility.

1 Introduction

Time series anomaly detection aims to identify patterns that deviate from expected behaviors within temporally sequential data and is crucial across numerous applications [1–3]. In recent years, with the flourish of deep learning, numerous deep learning-based methods have been proposed to tackle this problem [4–6]. However, latest studies [3, 7–10] indicates that deep learning-based methods may underperform classical data mining-based methods, especially in detecting subtle and prolonged anomalies [11, 12]. In response, Outlier Exposure (OE) [13] and Masked Autoencoders (MAE) [14] have emerged as prominent paradigms to solve above problems [15–21]. Nevertheless, both routes still possess inherent limitations that can impede their effectiveness in complex, real-world scenarios.

Limitations of OE-based (classification) methods: **L1. Heavy reliance on priori knowledge:** OE-based approaches fundamentally assume the existence of common anomalous patterns in the data, and leverage prior abnormal knowledge to generate pseudo-anomalous samples for training a classifier. While this approach can be effective when real-world anomalies align with the predefined types, it would fail to generalize to previously unseen or unexpected anomaly types. **L2. Improper classification granularity:** Current OE-based methods operate at either the "step level" or "window level", both of which present drawbacks. Step-level classification [15] attempts to assign anomaly scores to individual time steps by embedding the entire input window and producing predictions for each step in a single forward pass. This approach often struggles with longer input windows, which are necessary to capture sufficient contextual information. In contrast, window-level classification [16, 17] determines whether the entire input window contains anomalies and uses a sliding stride of one to achieve stepwise scores. While this can handle longer contexts, it risks obscuring short or subtle anomalies beneath predominantly normal patterns and introduces high computational overhead. **L3. Neglect of frequency-domain information:** Existing OE-based methods primarily operate in the time domain, overlooking the frequency domain where certain anomalies may be more pronounced. As a result, frequency-sensitive anomalies that are subtle or ambiguous in the time domain may go undetected.

Limitations of MAE-based (reconstruction) methods: **L4. Masking misalignment with anomaly locations:** MAE-based methods learn to model normal patterns by reconstructing masked patches from the unmasked ones, assigning anomaly scores based on reconstruction errors. Ideally, the masking strategy should mask potentially anomalous regions while leaving normal patches unmasked to ensure the model primarily use surrounding normal patterns to reconstruct masked areas. However, as illustrated in Figure 1, existing methods typically adopt either random [18, 19] or grating masking [22] without considering the semantic content of the patches and may mask both normal and anomalous regions indiscriminately. As a result, the reconstruction model is exposed to anomalous patterns, which can cause it to reproduce these patterns, thereby introducing false alarms in normal regions [18] (large reconstruction errors for normal regions in Figure 1(a)) or missed detections in anomalous regions [23] (good reconstruction for anomalous regions in Figure 1(b)).

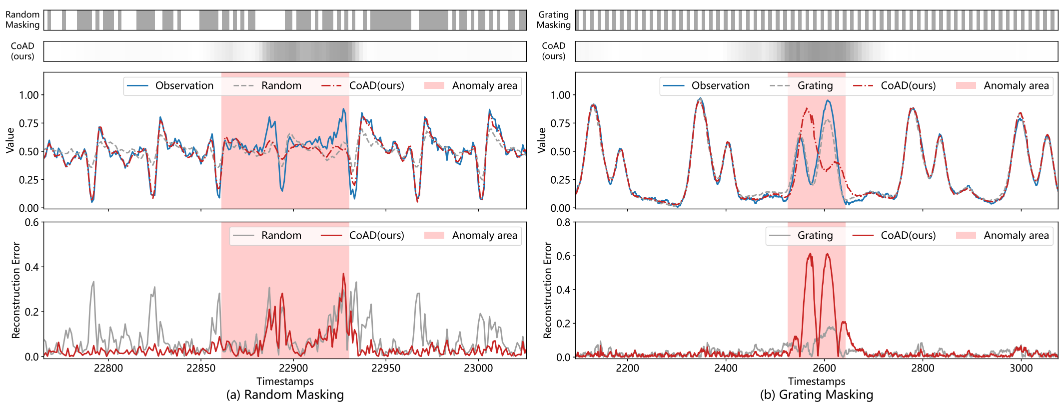

Figure 1: Comparison between our masking strategy and existing masking strategies. Complementary random masking refers to randomly selecting 50% patches within the patch sequence for masking, while grating masking refers to mask patches at regular intervals. The top bars denote the masked patches. For the random and grating masking strategies, the shaded regions indicate the masked areas. In the case of CoAD, darker colors signify patches that are masked more heavily.

To overcome the above limitations, we propose CoAD, a cooperative anomaly detection framework that integrates the strengths of both paradigms through a unified design. The core of CoAD is a guided soft masking mechanism, which leverages the OE classification to guide the masking process for the subsequent MAE-based reconstruction. Instead of masking patches randomly or uniformly, CoAD applies a probability-informed soft masking, where all patches are masked, but those deemed more likely to be anomalous are masked more heavily. This allows the model to suppress anomaly-related cues during reconstruction, leading to more accurate anomaly scoring (addressing **L4**). By jointly utilizing classification and reconstruction outputs, CoAD can both confirm known anomaly types and generalize to unseen ones (addressing **L1**). To support this cooperative strategy, CoAD introduces a patch-level, dual-branch time-frequency classification module. The input sequence is divided into non-overlapping patches, and each patch is independently classified based on features extracted from both the time and frequency domains. This patch-level structure offers two key advantages: (1) it enables the utilization of both long- and short-term contextual information from intra and inter-patch correlations with high efficiency (resolving **L2**), and (2) it captures frequency-domain patterns that are often overlooked by existing methods (solving **L3**).

Considering recent concerns about the experiment reliability in many existing studies [9, 24–26], we perform a rigorous and credible assessment with carefully selected benchmark datasets and evaluation metrics. Experimental results demonstrate that CoAD achieves significantly superior performance across all datasets from diverse sources, consistently surpassing both deep learning and data mining baselines. Moreover, qualitative analyses confirm that CoAD can successfully detect subtle

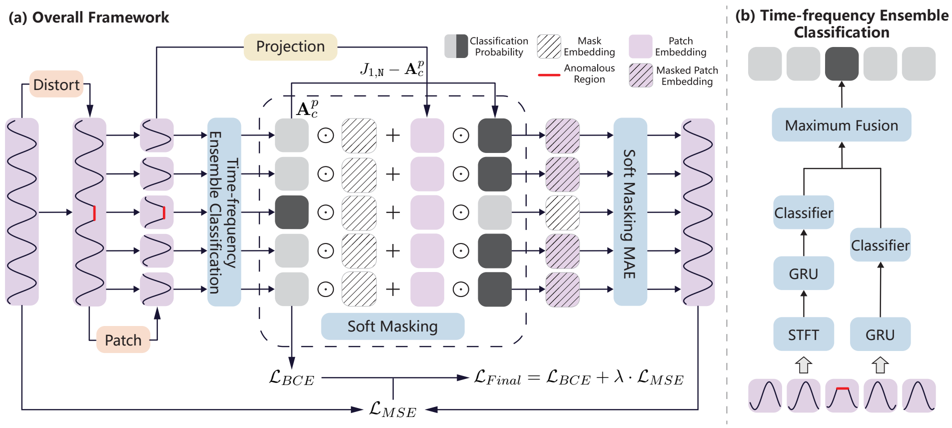

Figure 2: The overall framework of CoAD.

and hard anomalies that are often missed by existing methods, even those challenging for human experts.

2 Related work

OE-based methods assume that common anomalous patterns exist in time series and aim to train a binary classifier based on generated pseudo anomalies. AnomalyBERT [15] is a pioneering work that assumes four representative types of anomalies in time series and employs a BERT-like structure to extract features and perform anomaly classification. Although it achieves promising performance, it suffers from heavy reliance on pre-assumed anomalous patterns and often generalizes poorly to unseen anomaly types. Subsequent research attempts to address this issue by either replacing subsequences with random segments from the time series as pre-assumed anomalies [16] or leveraging a one-class classifier to additionally measure the distance to normal patterns [17]. Although these techniques appear promising, their advances do not lead to substantial improvements (see Table 1). Generalization remains the primary challenge for the OE-based methods.

MAE-based methods focus on learning from partially observed inputs and detecting anomalies specifically in the masked segments, in contrast to conventional reconstruction-based approaches that attempt to reconstruct the entire input indiscriminately. This strategy has been recently proposed and has led to significant performance improvements over traditional methods [18–21]. The masking strategy is the core technical component of MAE-based methods. The most intuitive and widely adopted approach is random masking [18, 19, 22]. However, purely random masking can excessively obscure normal patterns, making it difficult for the model to accurately reconstruct normal regions. An alternative is grating masking [22], which imposes a regular masking pattern. TFMAE [21] takes a preliminary step in this direction by masking subsequences with high variance within the input window; however, anomalies do not always correspond to high-variance areas.

3 Methodology

3.1 Task description

Let $\mathbf{S} = \{x_1, x_2, \ldots, x_{\mathtt{L}}\}$ denote a time series of length $\mathtt{L}$, where x_i represents the observation at time step i. The goal of time series anomaly detection is to assign an anomaly score $A(x_i) \in \mathbb{R}$ to each observation $x_i \in \mathbf{S}$, where a higher value of $A(x_i)$ indicates a greater likelihood that x_i is anomalous. Following conventions [4–6], we apply a sliding window to segment the raw time series $\mathbf{S}$ into a collection of windows, $\{\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_{\mathtt{K}}\}$, where each window $\mathbf{X}_n = \{x_{n,1}, x_{n,2}, \ldots, x_{n,\mathtt{T}}\}$ consists of $\mathtt{T}$ consecutive time steps and serves as a model input. For simplicity, we omit the window index n and denote a generic input window as $\mathbf{X} = \{x_1, x_2, \ldots, x_{\mathtt{T}}\}$.

3.2 Cooperative anomaly detection framework

We propose CoAD, a cooperative framework that integrates classification and reconstruction to leverage their complementary strengths and overcome their individual limitations. As shown in Figure 2, CoAD is built on the insight that the classification module can guide the reconstruction process, thereby improving detection accuracy, while the reconstruction module, in turn, enhances the generalizability of the classification module. The framework begins with an anomaly classifier, trained on pre-assumed anomaly patterns, to generate anomaly probabilities for each input patch, which are then used to guide the masking of the MAE module, enabling it to suppress the anomalous information as much as possible (Section 3.4). Both modules are jointly trained with a weighted loss combining binary cross-entropy (for classification) and mean squared error (for reconstruction), ensuring mutual optimization and synergy.

To further enhance performance, the classification module incorporates two key components. One is the patch-level classification, which is made feasible by the cooperative framework. Since the reconstruction module is responsible for generating fine-grained anomaly scores at the timestamp level, the classification module can operate on non-overlapping patches. This substantially reduces the input sequence length, easing computational burden and enabling the model to more effectively capture intra- and inter-patch contextual dependencies to find anomalies (see Section 3.3.1). The other is a time-frequency dual-branch ensemble that aggregates complementary temporal and spectral features, boosting the model's capacity to detect complex and subtle anomalies (see Section 3.3.1).

3.3 Mask generation via patch-level time-frequency classification

3.3.1 Distortion and patching.

Distortion. Classification-based anomaly detection methods rely on pre-defined anomalous patterns. Following prior work [15, 19], we simulate four common types of time series anomalies: Uniform Replacement, Mirror Flip, Length Scale, and Jittering. We randomly select a segment of the input window $\mathbf{X}$, with a length ranging from 0 to one dominant period, and inject one of the four types of anomalies. **Details of these distortion strategies are provided in Appendix A**. The resulting distorted series is denoted as $\tilde{\mathbf{X}}$.

Patching. As illustrated in Figure 3(a) and (b), existing classification-based methods typically adopt either "step-level" or "window-level" classification granularity, both of which have inherent limitations. Step-level methods encode the entire input window into a single latent representation, then use a decoder to produce anomaly scores for each individual time step [15]. However, producing accurate fine-grained scores from a single embedding is especially difficult for long sequences, which are necessary to capture sufficient contextual information [18, 27]. Window-level methods also use a single embedding for the entire input window but output only a binary label indicating whether any anomaly exists in the window [16, 17]. This coarse prediction can easily overlook short-duration or subtle anomalies that are masked by dominant normal patterns.

To address the above issues, we adopt a "patch-level" classification strategy. Specifically, an input window is segmented into non-overlapping patches $\tilde{\mathbf{X}}^p = [\tilde{x}_1^p, \tilde{x}_2^p, \ldots, \tilde{x}_{\mathtt{N}}^p] \in \mathbb{R}^{\mathtt{P}\times\mathtt{N}}$, where $\mathtt{N}$ is the number of patches and $\tilde{x}_i^p$ is a patch of length $\mathtt{P}$. Each patch is first embedded, and then all embeddings are integrated using a GRU module to extract long-term correlations, followed by a shared linear layer that classifies whether a patch contains anomalies (detailed in Section 3.3.2). This patch-level design strikes a balance between the granularity of step-level and window-level methods. Crucially, it is made feasible by our cooperative framework: since the reconstruction module provides fine-grained anomaly scores at the step level, the classification module does not need to produce scores for each individual time step. This decoupling enables non-overlapping patches, which substantially reduces the sequence length and enables the model to more effectively learn both intra- and inter-patch context with reduced computational overhead.

3.3.2 Time-frequency ensemble classification

Time-frequency analysis is widely used in time series research. As illustrated in Figure 4, certain anomalies that are difficult to distinguish in the time domain exhibit more salient patterns in the frequency domain. While existing classification-based anomaly detection methods generally over-

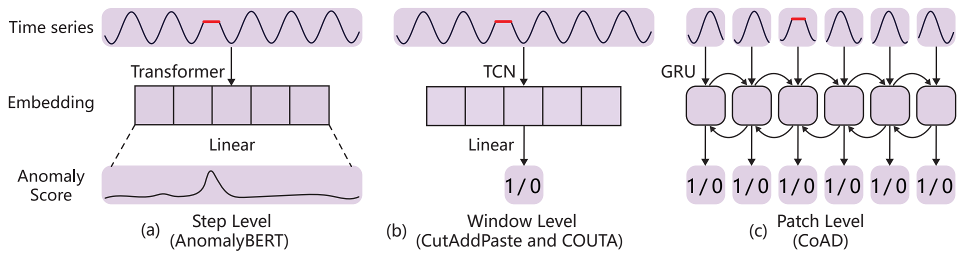

Figure 3: Comparison between different classification granularities.

look frequency features, some reconstruction-based approaches have incorporated time-frequency representations to improve detection performance [21, 28–30].

However, these reconstruction-based approaches face two limitations. (i) As observed in Figure 4 and corroborated by prior studies [28, 30], frequency amplitudes vary considerably across bands, typically high in low-frequency and low in high-frequency regions. This uneven distribution complicates accurate reconstruction across all bands, often resulting in large relative errors for high-frequency components. To mitigate this issue, we propose using frequency-domain features for classification rather than as reconstruction targets. (ii) Most existing work extracts frequency features via the Fast Fourier Transform (FFT), which provides fine-grained frequency resolution but only coarse (window-level) resolution in the time domain. To address this, we adopt the Short-Time Fourier Transform (STFT), which enables finer temporal localization of frequency patterns. By integrating both time- and frequency-domain representations, our two-branch classification module leverages complementary information to enhance anomaly detection robustness.

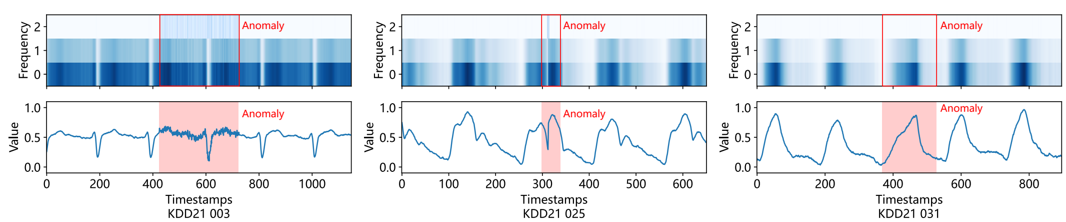

Figure 4: The comparison of frequency domain features between normal and anomalous regions. The upper part shows the spectrogram obtained through STFT.

Frequency branch. In the frequency classification branch, we apply the STFT to the entire input window $\tilde{\mathbf{X}}$ to obtain the frequency-domain representation $\tilde{\mathbf{X}}_f \in \mathbb{R}^{2\mathtt{K}\times\mathtt{T}}$, where $\mathtt{K}$ denotes the number of frequency bins. The real and imaginary parts of the complex-valued STFT output are concatenated. We then segment $\tilde{\mathbf{X}}_f$ into non-overlapping patches $\tilde{\mathbf{X}}_f^p = [\tilde{x}_{f,1}^p, \tilde{x}_{f,2}^p, \ldots, \tilde{x}_{f,\mathtt{N}}^p] \in \mathbb{R}^{2\mathtt{KP}\times\mathtt{N}}$, where $\mathtt{P}$ is the patch length and $\mathtt{N}$ is the number of patches.

To model inter-patch dependencies, each frequency patch is first linearly projected into a hidden space and then processed by a GRU encoder:

$$\tilde{\mathbf{H}}_f^p = \mathrm{GRU}\left(\mathbf{W_p}\tilde{\mathbf{X}}_f^p\right), \tag{1}$$

where $\mathbf{W_p} \in \mathbb{R}^{\mathtt{H}\times 2\mathtt{KP}}$ is a learnable projection matrix and $\mathtt{H}$ is the hidden dimension. The output $\tilde{\mathbf{H}}_f^p$ is then passed through a shared linear layer followed by a sigmoid activation to generate the anomaly probability for each patch:

$$\mathbf{A}_f^p = \sigma\left(\mathbf{W_f}\tilde{\mathbf{H}}_f^p\right), \tag{2}$$

where $\mathbf{W_f} \in \mathbb{R}^{1\times\mathtt{H}}$ is a learnable weight vector, and $\mathbf{A}_f^p = [a_{f,1}^p, a_{f,2}^p, \ldots, a_{f,\mathtt{N}}^p]$ denotes the predicted anomaly probabilities for the frequency patches $\tilde{X}_f^p$.

Time branch. The time classification branch directly feeds the patch set $\tilde{\mathbf{X}}^p$ into a GRU encoder, followed by a linear layer and sigmoid activation to produce patch-wise anomaly probabilities:

$$\mathbf{A}_t^p = \sigma\left(\mathbf{W_t}\ \mathrm{GRU}(\tilde{\mathbf{X}}^p)\right), \tag{3}$$

where $\mathbf{A}_t^p = [a_{t,1}^p, a_{t,2}^p, \ldots, a_{t,\mathtt{N}}^p]$, and $\mathbf{W_t} \in \mathbb{R}^{1\times\mathtt{H}}$ is a learnable projection.

Ensemble strategy. We adopt a maximum fusion strategy to combine anomaly probabilities from the two branches:

$$\mathbf{A}_c^p = \max(\mathbf{A}_f^p, \mathbf{A}_t^p) = [\max(a_{f,1}^p, a_{t,1}^p), \max(a_{f,2}^p, a_{t,2}^p), \ldots, \max(a_{f,\mathtt{N}}^p, a_{t,\mathtt{N}}^p)] \tag{4}$$

where $\mathbf{A}_c^p \in \mathbb{R}^{\mathtt{N}}$ represents the final anomalous probabilities for the patches $\tilde{\mathbf{X}}^p$. This strategy ensures that as long as either branch detects an anomaly, it will be retained. It helps reduce false alarms, since both branches must agree on normality. Moreover, it also avoids forcing either branch to fit on detecting anomalies it is less sensitive to, thus improving training stability. We also explore different ensemble strategies in Section 4.4.

3.4 Probability-informed Soft Masking MAE

Probability-informed Soft Masking. The core idea of COAD is to guide MAE reconstruction using prior classification probabilities. We propose a soft masking strategy, where each patch embedding is blended with a learnable mask embedding, weighted by the anomaly probability from the classification module. This enables a more nuanced suppression of potentially anomalous information, especially in borderline cases where a hard mask based on binary classification may be too rigid or error-prone. The resulting soft-masked embeddings are:

$$\mathbf{E_m} = \mathbf{A}_c^p \cdot \mathbf{E}_{\mathtt{mask}} + (J_{1,\mathtt{N}} - \mathbf{A}_c^p) \cdot \mathbf{W_m}\tilde{\mathbf{X}}^p, \tag{5}$$

where $\mathbf{E}_{mask} \in \mathbb{R}^{\mathtt{H}\times\mathtt{N}}$ represents learnable mask embedding and $\mathbf{W_m} \in \mathbb{R}^{\mathtt{H}\times\mathtt{P}}$ is a learnable projection matrix that maps raw patches into patch embeddings. $\mathbf{E_m}$ is subsequently input into a GRU encoder followed by a linear layer to reconstruct the original time series:

$$\mathbf{X_r} = \mathtt{Flat}(\mathbf{W_r}\,\mathrm{GRU}\,(\mathbf{E_m})), \text{where } \mathbf{W_r} \in \mathbb{R}^{\mathtt{P}\times\mathtt{H}}. \tag{6}$$

The MAE is trained to minimize the Mean Squared Error between $\tilde{\mathbf{X}}_\mathbf{r}$ and the original time series $\mathbf{X}$.

The probability-informed soft masking strategy offers fine-grained control over the masking strength, enabling adaptive suppression of anomalous information based on classification confidence. Unlike rigid hard masking, it handles uncertainty more gracefully, improving robustness and generalization, especially for unseen anomalies. In extreme cases where the anomalous patterns are completely different from pre-assumed anomalies, our guided masking process would approximate random masking. We provide the comparison experiments between soft masking and hard masking in Section 4.4. Additionally, soft masking ensures smoother gradient flow during training and fosters better synergy between the classification and reconstruction branches by aligning their objectives.

Training. The overall training objective combines the BCE loss for classification and MSE loss for reconstruction:

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N}(y_i \cdot \log(a_c^i) + (1-y_i)\cdot\log(1-a_c^i)) + \lambda \cdot \frac{1}{N}\sum_{i=1}^{N}\left\|x_{r,i}^p - x_i^p\right\|_2^2. \tag{7}$$

where a_c^i denotes the anomalous probability of patch $\tilde{x}_i^p$, and y_i is the corresponding ground truth label, with $y_i = 1$ if patch $\tilde{x}_i^p$ contains anomalies, and $y_i = 0$ otherwise. $x_{r,i}^p$ is the reconstructed patch and x_i^p is the original time series patch. λ is the weight to balance the two losses.

Anomaly scoring. During inference, the classification and reconstruction modules operate collaboratively to detect anomalies. The final anomaly score for each patch is computed as the sum of the classification probability and the reconstruction error:

$$a(x_i^p) = a_c^p \cdot J_{1,\mathtt{P}} + \left|x_i^p - x_{r,i}^p\right|, \tag{8}$$

where a_c^p is the classification probability for patch x_i^p, and $x_{r,i}^p$ is the corresponding reconstructed patch. The vector $J_{1,\mathtt{P}}$ ensures proper dimensional alignment for element-wise addition. Since the input time series is normalized using training set, both the classification probabilities and reconstruction errors lie on comparable scales, allowing them to be directly added into a unified anomaly score.

Table 1: Average results on KDD21 and TSB-AD datasets. All results are in %, the best results are in **bold**, and the second best results are with underline.

Model / Dataset		KDD21				TSB-AD			
Model Class	Model(Venue)	F1	AUC-PR	R-AUC-PR	VUS-PR	F1	AUC-PR	R-AUC-PR	VUS-PR
MAE	DADA (ICLR-25 [19])	3.49	1.39	2.04	2.05	17.36	12.30	14.92	14.53
	MOMENT (ICML-24 [20])	11.06	7.71	9.20	9.13	25.29	18.80	25.90	24.67
	MMA (VLDB-25 [18])	44.47	39.24	37.97	37.38	43.20	38.65	37.33	36.94
	TFMAE (ICDE-24 [21])	2.53	0.99	1.75	1.75	7.29	3.45	5.83	5.77
OE	AnomalyBERT (ICLR-23 [15])	23.42	15.92	14.16	13.79	19.14	9.85	13.21	13.85
	CutAddPaste (KDD-24 [16])	21.75	15.51	19.00	18.20	26.22	21.34	25.45	25.08
	TriAD (ICDE-24 [12])	16.78	22.37	28.09	27.56	N/A	N/A	N/A	N/A
	COUTA (TKDE-24 [17])	6.58	3.65	3.79	3.82	18.78	13.12	11.07	11.01
Time-Frequency Reconstruction	FCVAE (WWW-24 [29])	11.23	7.18	6.25	6.38	28.47	22.12	20.61	20.58
	TFAD (CIKM-22 [30])	1.85	0.86	1.59	1.58	6.17	3.34	6.21	5.99
	CATCH (ICLR-25 [28])	13.29	9.08	9.26	9.19	24.46	20.02	22.32	21.67
Reconstruction/ Prediction	TranAD (VLDB-22 [31])	11.23	7.78	7.94	7.89	18.07	13.06	12.23	12.05
	MAUT (ICASSP-23 [32])	30.20	23.84	23.94	23.65	19.82	14.67	14.91	14.75
	M2N2 (AAAI-24 [33])	5.57	2.70	3.15	3.18	16.98	10.38	9.41	9.25
Data Mining	KShapeAD (NeurIPS-24 [9])	43.27	39.37	39.48	39.02	35.51	32.67	31.87	31.27
	SAND (VLDB-21 [34])	39.43	34.72	34.23	33.78	34.74	31.52	31.85	31.05
	Sub-PCA (NeurIPS-24 [9])	15.45	11.32	14.06	13.17	32.49	27.59	23.88	23.85
	Series2Graph (VLDB-20 [35])	28.11	22.61	25.63	24.81	33.48	30.11	30.13	29.57
	KMeansAD (VLDB-22 [36])	37.97	34.33	33.49	33.20	41.42	37.26	37.64	36.81
	Matrix Profile (CIKM-16 [37])	28.00	18.50	25.37	24.13	35.10	27.93	29.88	28.94
Cooperative	**CoAD(ours)**	**52.82**	**48.10**	**46.35**	**45.67**	**49.13**	**43.66**	**40.50**	**39.83**

4 Experiments

4.1 Datasets and evaluation metrics

Current issues. Unreliable datasets and biased evaluation metrics have long plagued the field of time series anomaly detection [24, 9]. Many studies reported impressive performance based on flawed datasets and metrics, contributing to the problem of "Creating the Illusion of Progress" [24, 25]. Commonly used datasets such as SMD [38], PSM [39], SWAT [40], SMAP [41], MSL [41], and NAB [42], suffer from various issues including mislabeled ground truth, trivial anomalies, unrealistic anomaly densities, and the run-to-failure bias [24, 9, 25, 26]. Moreover, some evaluation metrics, such as F1 with point adjustment [38] and F1-Affiliation [43] tend to overestimate model performance, even awarding high scores to random predictions [18, 9, 44].

Our settings. To ensure the reliability of our evaluation, we adopt recently proposed high-quality datasets and rigorous metrics [24, 9, 45]. *For datasets*, we use the KDD21 (a.k.a. UCR Anomaly Archive) [24] and TSB-AD datasets [9]. i) KDD21 comprises 250 subsets across diverse domains, including healthcare, sports, industry, and robotics. ii) TSB-AD also covers various domains and partially overlaps with KDD21. However, some subsets in TSB-AD, such as NAB, WSD, and YAHOO, suffer from the aforementioned quality issues [26]. Therefore, adhering to the dataset quality criteria in prior research [24, 46, 3], we select only the high-quality, non-overlapping subsets from TSB-AD for our experiments. *For metrics*, we use those recommended by recent benchmarking studies, including Standard-F1, AUC-PR [47], Range-AUC-PR, and VUS-PR [45], which are recognized as the most reliable and precise measures in the recent benchmark paper [9]. **Details of the datasets and evaluation metrics are provided in Appendix B.**

4.2 Baselines and implementation details

To demonstrate the superiority of our method, we compared it against 21 SOTA baseline methods, including 14 deep learning-based methods and 7 best performing data mining-based methods in recent evaluation papers [9]. As shown in Table 1, the deep learning-based methods can be categorized into four groups: *i) MAE-based methods* detect anomalies by leveraging mask reconstruction errors, *ii) OE-based methods* incorporate prior abnormal knowledge to help detect anomalies, *iii) Time-Frequency Reconstruction methods* consider the reconstruction errors in both time and frequency domain to identify anomalies, *iv) Reconstruction- or prediction-based methods* detect anomalies using the reconstruction or prediction errors. The DADA [19] and MOMENT [20] methods are based on large foundation models, and TFMAE [21] employs MAEs in both the time and frequency domains.

Implementation details. We reproduce all deep learning-based models using their official open-source repositories. The non-deep learning models are implemented based on the TSB-UAD [48] and TSB-AD [9] libraries. Both our model and the baselines follow identical data preprocessing procedures in an integrated, unified pipeline. We strictly adhere to the original training and testing splits provided by the KDD21 and TSB-AD datasets, and employ the early stopping strategy for model selection. All deep learning-based methods are trained 5 times with different random seeds, and the averaged performance is reported. **More details are listed in Appendix C**

4.3 Comparison results

Effectiveness. The comparison results are summarized in Table 1. The following key observations can be made: (1) CoAD consistently outperforms all baseline models on both the KDD21 and TSB-AD datasets across all evaluation metrics. (2) Compared to approaches that rely exclusively on either MAE or OE, CoAD achieves substantial performance gains, demonstrating the effectiveness of the proposed cooperative framework in leveraging the complementary strengths of classification and reconstruction. (3) Notably, traditional data mining-based methods outperform many deep learning counterparts, consistent with prior findings [24], and highlighting the ongoing challenges of deep learning in time series anomaly detection. However, CoAD surpasses even the strongest data mining baselines, illustrating that with principled architectural design and rigorous evaluation, deep learning can achieve SOTA performance in this domain.

Efficiency. Figure 5 presents a comparison of model efficiency regarding inference time and parameter count on the KDD21 dataset (**The experiment settings and results on the TSB-AD dataset are in Appendix G**). The results highlight that CoAD not only achieves superior detection performance but also delivers remarkable efficiency. Specifically, CoAD completes the inference on all subsets of the KDD21 dataset, comprising more than 6 million data points, in just 6.89 seconds, outperforming most baselines by orders of magnitude in speed. This demonstrates the strong potential of CoAD for real-time anomaly detection in high-throughput data streams. Furthermore, the model maintains a compact size of only 2.04 million parameters, placing it on par with lightweight data mining-based methods and underscoring its practicality for deployment in resource-constrained environments.

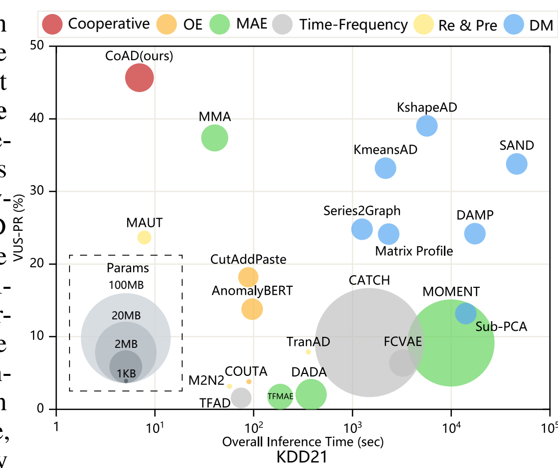

Figure 5: Model efficiency comparison.

4.4 Ablation and Design Choice Study

To evaluate the effectiveness of each component and design choice in CoAD, we conduct a comprehensive study with the model variants listed in Table 2. Specifically, we examine standalone models (OE or MAE alone), different cooperative strategies, classification granularities, time-frequency ensemble strategies and anomaly scoring methods. For clarification: *OE+MAE (Random/Grating)* directly combines detection results from OE and random/grating masking MAE without guidance; *OE+MAE (Guide w/ Hard Mask)* uses OE guidance with discrete hard masks; *OE+MAE (Guide w/o Score)* guides MAE using OE soft masking, but only uses MAE's reconstruction error for final anomaly scoring; *Feature_Gate* fuses time and frequency features using a gated mechanism [49]; and *Decision_Mean* averages the classification scores from both branches. **Implementation details and descriptions of all variants are available in Appendix E.**

Based on the results, we draw the following key conclusions: (1) CoAD achieves the best overall performance, validating the effectiveness of our design. (2) Cooperative models generally outperform standalone ones, with our guided masking framework outperforming even hard-masked or naive cooperative baselines. (3) Patch-level classification granularity yields markedly better results than step- or window-level approaches. (4) Frequency-domain information significantly enhances anomaly detection performance. **Qualitative ablation results are provided in Appendix H.**

Table 2: Ablation and design choice study. Best results are in **bold**.

Variants	OE		MAE				KDD21				TSB-AD			
	Time	Frequency	Random	Grating	Guide w/ Hard Mask	Guide w/ Soft Mask	F1	AUC-PR	R-AUC-PR	VUS-PR	F1	AUC-PR	R-AUC-PR	VUS-PR
OE alone	✓	–	–	–	–	–	27.84	24.12	24.81	24.38	29.79	22.36	27.14	26.28
	✓	✓	–	–	–	–	47.95	42.75	41.06	40.41	37.68	31.21	30.48	29.90
MAE alone	–	–	✓	–	–	–	35.65	30.23	30.96	30.24	32.30	25.74	23.55	23.35
	–	–	–	✓	–	–	38.99	32.31	32.76	31.99	35.01	29.08	28.51	28.03
Cooperative	✓	✓	✓	–	–	–	44.88	41.26	39.55	39.05	45.01	39.31	35.79	35.20
	✓	✓	–	✓	–	–	46.16	41.95	40.86	40.19	47.07	41.48	37.39	37.08
	✓	✓	–	–	✓	–	49.03	45.27	43.87	43.49	44.06	38.57	35.31	34.76
CoAD	✓	✓	–	–	–	✓	**52.82**	**48.10**	**46.35**	**45.67**	**49.13**	**43.66**	**40.50**	**39.83**

Design Choice

Variants	Classification Granularity		Fusion Strategy			Masking Method	KDD21				TSB-AD			
	Step Level	Window Level	Feature_Add	Feature_Gate	Decision_Mean	Guide w/o Score	F1	AUC-PR	R-AUC-PR	VUS-PR	F1	AUC-PR	R-AUC-PR	VUS-PR
Cooperative	✓	–	–	–	–	–	43.09	36.62	35.43	35.00	42.52	36.06	34.02	33.37
	–	✓	–	–	–	–	15.26	10.21	11.29	11.02	22.43	17.08	15.71	15.53
	–	–	✓	–	–	–	50.76	45.93	44.76	44.25	40.11	35.15	33.40	32.94
	–	–	–	✓	–	–	50.52	45.58	45.53	44.71	42.23	37.20	34.87	34.40
	–	–	–	–	✓	–	51.50	45.98	45.53	44.81	42.24	36.23	33.71	33.18
	–	–	–	–	–	✓	48.13	42.92	42.93	42.09	37.07	30.54	28.04	27.73
CoAD	**Patch Level**		**Decision_Max**			**Guide w/ Score**	**52.82**	**48.10**	**46.35**	**45.67**	**49.13**	**43.66**	**40.50**	**39.83**

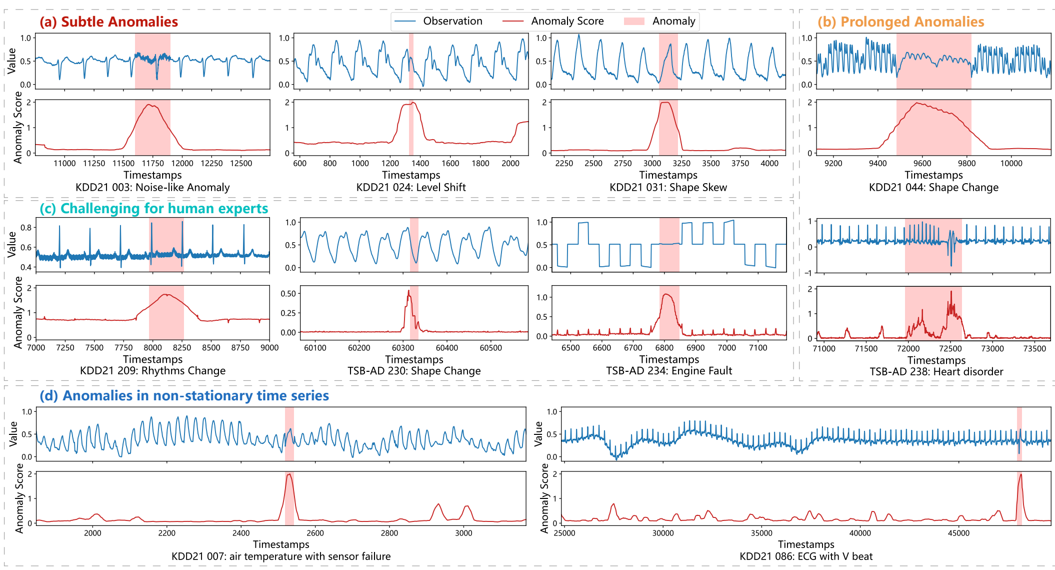

Figure 6: Visualization of detection results of challenging anomalies.

4.5 Visualization on challenging anomalies

Figure 6 visualizes the detection results of CoAD on several challenging cases. **a) Subtle anomalies:** KDD21 003, KDD21 024, and KDD21 031 contain subtle anomalies whose amplitudes are similar to those of normal values, making them easily overlooked by existing approaches [11, 12]. Despite their subtlety, these anomalies are effectively detected by CoAD. **b) Prolonged anomalies:** KDD21 044 and TSB-AD 238 include anomalies that persist over multiple periods, posing difficulties for many existing methods [6]. Nevertheless, CoAD successfully captures and localizes these long-duration deviations. **c) Anomalies challenging for human experts:** As emphasized by [26], a desirable anomaly detection model should be capable of identifying anomalies that are difficult even for human experts to recognize. KDD21 209, TSB-AD 230, and TSB-AD 234 exemplify such cases. CoAD consistently assigns high anomaly scores to these regions, highlighting its robustness in handling complex and ambiguous patterns. **d) Anomalies in non-stationary time series:** As highlighted in [50], existing deep learning-based methods often fail to detect anomalies in non-stationary time series. In contrast, CoAD demonstrates strong capability in handling real-world non-stationary data, accurately identifying anomalies in these complex scenarios.

5 Conclusion

This paper proposes a cooperative framework that bridges classification and reconstruction to address the limitations of existing OE and MAE based methods. The classification module leverages both time and frequency information to provide accurate masking proposals for the reconstruction module. The reconstruction module takes the soft masking strategy to fully leverage the guidance from the classification module. Extensive experiments on reliable datasets using rigorous evaluation metrics validate that our proposed framework significantly outperforming baselines in both detection performance and computational efficiency.

References

[1] A. Blázquez-García, A. Conde, U. Mori, and J. A. Lozano, "A review on outlier/anomaly detection in time series data," *ACM Computing Surveys*, vol. 54, no. 3, pp. 1–33, Apr. 2022.

[2] M. Gupta, J. Gao, C. C. Aggarwal, and J. Han, "Outlier detection for temporal data: A survey," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 9, pp. 2250–2267, Sep. 2014.

[3] S. Schmidl, P. Wenig, and T. Papenbrock, "Anomaly detection in time series: A comprehensive evaluation," *Proceedings of the VLDB Endowment*, vol. 15, no. 9, pp. 1779–1797, May 2022.

[4] G. Li and J. J. Jung, "Deep learning for anomaly detection in multivariate time series: Approaches, applications, and challenges," *Information Fusion*, vol. 91, pp. 93–102, Mar. 2023.

[5] Z. Zamanzadeh Darban, G. I. Webb, S. Pan, C. Aggarwal, and M. Salehi, "Deep learning for time series anomaly detection: A survey," *ACM Computing Surveys*, vol. 57, no. 1, pp. 1–42, Jan. 2025.

[6] N. Mejri, L. Lopez-Fuentes, K. Roy, P. Chernakov, E. Ghorbel, and D. Aouada, "Unsupervised anomaly detection in time-series: An extensive evaluation and analysis of state-of-the-art methods," *Expert Systems with Applications*, vol. 256, p. 124922, Dec. 2024.

[7] K.-H. Lai, D. Zha, Y. Zhao, G. Wang, J. Xu, and X. Hu, "Revisiting time series outlier detection: Definitions and benchmarks," in *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, 2021.

[8] F. Rewicki, J. Denzler, and J. Niebling, "Is it worth it? comparing six deep and classical methods for unsupervised anomaly detection in time series," *Applied Sciences*, vol. 13, no. 3, p. 1778, Jan. 2023.

[9] Q. Liu and J. Paparrizos, "The elephant in the room: Towards a reliable time-series anomaly detection benchmark," in *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024. [Online]. Available: https://openreview.net/forum?id=R6kJtWsTGy

[10] M. S. Sarfraz, M.-Y. Chen, L. Layer, K. Peng, and M. Koulakis, "Position: quo vadis, unsupervised time series anomaly detection?" in *Proceedings of the 41st International Conference on Machine Learning*, ser. ICML'24. JMLR.org, 2024.

[11] D. Lee, S. Malacarne, and E. Aune, "Explainable time series anomaly detection using masked latent generative modeling," *Pattern Recognition*, vol. 156, p. 110826, Dec. 2024.

[12] Y. Sun, G. Pang, G. Ye, T. Chen, X. Hu, and H. Yin, "Unraveling the 'anomaly' in time series anomaly detection: A self-supervised tri-domain solution," in *2024 IEEE 40th International Conference on Data Engineering (ICDE)*. Utrecht, Netherlands: IEEE, May 2024, pp. 981–994.

[13] D. Hendrycks, M. Mazeika, and T. Dietterich, "Deep anomaly detection with outlier exposure," in *International Conference on Learning Representations*, 2019. [Online]. Available: https://openreview.net/forum?id=HyxCxhRcY7

[14] K. He, X. Chen, S. Xie, Y. Li, P. Dollar, and R. Girshick, "Masked autoencoders are scalable vision learners," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. New Orleans, LA, USA: IEEE, Jun. 2022, pp. 15 979–15 988.

[15] Y. Jeong, E. Yang, J. H. Ryu, I. Park, and M. Kang, "Anomalybert: Self-supervised transformer for time series anomaly detection using data degradation scheme," in *The Eleventh International Conference on Learning Representations*, 2023.

[16] R. Wang, X. Mou, R. Yang, K. Gao, P. Liu, C. Liu, T. Wo, and X. Liu, "Cutaddpaste: Time series anomaly detection by exploiting abnormal knowledge," in *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, vol. 12. Barcelona Spain: ACM, Aug. 2024, pp. 3176–3187.

[17] H. Xu, Y. Wang, S. Jian, Q. Liao, Y. Wang, and G. Pang, "Calibrated one-class classification for unsupervised time series anomaly detection," *IEEE Transactions on Knowledge and Data Engineering*, pp. 1–14, 2024.

[18] Q. Tang, C. Dai, Y. Wu, and H. Zhou, "Mlp-mixer based masked autoencoders are effective, explainable and robust for time series anomaly detection," *Proc. VLDB Endow.*, vol. 18, no. 3, pp. 798–811, Apr. 2025. [Online]. Available: https://doi.org/10.14778/3712221.3712243

[19] Q. Shentu, B. Li, K. Zhao, Y. Shu, Z. Rao, L. Pan, B. Yang, and C. Guo, "Towards a general time series anomaly detector with adaptive bottlenecks and dual adversarial decoders," in *The Thirteenth International Conference on Learning Representations*, 2025. [Online]. Available: https://openreview.net/forum?id=aKcd7ImG5e

[20] M. Goswami, K. Szafer, A. Choudhry, Y. Cai, S. Li, and A. Dubrawski, "MOMENT: A family of open time-series foundation models," in *Forty-first International Conference on Machine Learning*, 2024. [Online]. Available: https://openreview.net/forum?id=FVvf69a5rx

[21] Y. Fang, J. Xie, Y. Zhao, L. Chen, Y. Gao, and K. Zheng, "Temporal-frequency masked autoencoders for time series anomaly detection," in *2024 IEEE 40th International Conference on Data Engineering (ICDE)*. Utrecht, Netherlands: IEEE, May 2024, pp. 1228–1241.

[22] Y. Chen, C. Zhang, M. Ma, Y. Liu, R. Ding, B. Li, S. He, S. Rajmohan, Q. Lin, and D. Zhang, "Imdiffusion: Imputed diffusion models for multivariate time series anomaly detection," *Proceedings of the VLDB Endowment*, vol. 17, no. 3, pp. 359–372, Nov. 2023.

[23] X. Yao, C. Zhang, R. Li, J. Sun, and Z. Liu, "One-for-all: Proposal masked cross-class anomaly detection," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 4, pp. 4792–4800, Jun. 2023.

[24] R. Wu and E. Keogh, "Current time series anomaly detection benchmarks are flawed and are creating the illusion of progress," *IEEE Transactions on Knowledge and Data Engineering*, pp. 1–1, 2021.

[25] E. Keogh, "Irrational exuberance: Why we should not believe 95% of papers on time series anomaly detection," Sep. 2021, https://kdd-milets.github.io/milets2021/slides/Irrational%20Exuberance_Eammon_Keogh.pdf.

[26] ——, "The fundamental problem in tsad research," 2024, https://lnkd.in/gP-H8w4i.

[27] Y. Nie, N. H. Nguyen, P. Sinthong, and J. Kalagnanam, "A time series is worth 64 words: Long-term forecasting with transformers," in *The Eleventh International Conference on Learning Representations*, 2023. [Online]. Available: https://openreview.net/forum?id=Jbdc0vTOcol

[28] X. Wu, X. Qiu, Z. Li, Y. Wang, J. Hu, C. Guo, H. Xiong, and B. Yang, "Catch: Channel-aware multivariate time series anomaly detection via frequency patching," in *ICLR*, 2025.

[29] Z. Wang, C. Pei, M. Ma, X. Wang, Z. Li, D. Pei, S. Rajmohan, D. Zhang, Q. Lin, H. Zhang, J. Li, and G. Xie, "Revisiting vae for unsupervised time series anomaly detection: A frequency perspective," in *Proceedings of the ACM on Web Conference 2024*. Singapore Singapore: ACM, May 2024, pp. 3096–3105.

[30] C. Zhang, T. Zhou, Q. Wen, and L. Sun, "Tfad: A decomposition time series anomaly detection architecture with time-frequency analysis," in *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. Atlanta GA USA: ACM, Oct. 2022, pp. 2497–2507.

[31] S. Tuli, G. Casale, and N. R. Jennings, "Tranad: Deep transformer networks for anomaly detection in multivariate time series data," *Proceedings of the VLDB Endowment*, vol. 15, no. 6, pp. 1201–1214, Feb. 2022.

[32] S. Qin, Y. Luo, and G. Tao, "Memory-augmented u-transformer for multivariate time series anomaly detection," in *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Rhodes Island, Greece: IEEE, Jun. 2023, pp. 1–5.

[33] D. Kim, S. Park, and J. Choo, "When model meets new normals: Test-time adaptation for unsupervised time-series anomaly detection," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 12, pp. 13 113–13 121, Mar. 2024.

[34] P. Boniol, J. Paparrizos, T. Palpanas, and M. J. Franklin, "Sand: Streaming subsequence anomaly detection," *Proceedings of the VLDB Endowment*, vol. 14, no. 10, pp. 1717–1729, Jun. 2021.

[35] P. Boniol and T. Palpanas, "Series2graph: : Graph-based subsequence anomaly detection for time series," *Proceedings of the VLDB Endowment*, vol. 13, no. 12, pp. 1821–1834, Aug. 2020.

[36] T. Yairi, Y. Kato, and K. Hori, "Fault detection by mining association rules from house-keeping data," in *proceedings of the 6th International Symposium on Artificial Intelligence, Robotics and Automation in Space*, vol. 18. Citeseer, 2001, p. 21.

[37] T. Nakamura, M. Imamura, R. Mercer, and E. Keogh, "Merlin: Parameter-free discovery of arbitrary length anomalies in massive time series archives," in *2020 IEEE International Conference on Data Mining (ICDM)*. Sorrento, Italy: IEEE, Nov. 2020, pp. 1190–1195.

[38] Y. Su, Y. Zhao, C. Niu, R. Liu, W. Sun, and D. Pei, "Robust anomaly detection for multivariate time series through stochastic recurrent neural network," *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, Jul. 2019.

[39] A. Abdulaal, Z. Liu, and T. 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*. Virtual Event Singapore: ACM, Aug. 2021, pp. 2485–2494.

[40] J. Goh, S. Adepu, K. N. Junejo, and A. Mathur, "A dataset to support research in the design of secure water treatment systems," in *Critical Information Infrastructures Security*, G. Havarneanu, R. Setola, H. Nassopoulos, and S. Wolthusen, Eds. Cham: Springer International Publishing, 2017, pp. 88–99.

[41] K. Hundman, V. Constantinou, C. Laporte, I. Colwell, and T. Soderstrom, "Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. London United Kingdom: ACM, Jul. 2018, pp. 387–395.

[42] S. Ahmad, A. Lavin, S. Purdy, and Z. Agha, "Unsupervised real-time anomaly detection for streaming data," *Neurocomputing*, vol. 262, pp. 134–147, 2017, online Real-Time Learning Strategies for Data Streams. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231217309864

[43] A. Huet, J. M. Navarro, and D. Rossi, "Local evaluation of time series anomaly detection algorithms," in *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. Washington DC USA: ACM, Aug. 2022, pp. 635–645.

[44] S. Kim, K. Choi, H.-S. Choi, B. Lee, and S. Yoon, "Towards a rigorous evaluation of time-series anomaly detection," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 7, pp. 7194–7201, 2022.

[45] J. Paparrizos, P. Boniol, T. Palpanas, R. S. Tsay, A. Elmore, and M. J. Franklin, "Volume under the surface: A new accuracy evaluation measure for time-series anomaly detection," *Proceedings of the VLDB Endowment*, vol. 15, no. 11, pp. 2774–2787, Jul. 2022.

[46] Y. Lu, R. Wu, A. Mueen, M. A. Zuluaga, and E. Keogh, "Damp: Accurate time series anomaly detection on trillions of datapoints and ultra-fast arriving data streams," *Data Mining and Knowledge Discovery*, vol. 37, no. 2, pp. 627–669, Mar. 2023.

[47] J. Davis and M. Goadrich, "The relationship between precision-recall and roc curves," in *Proceedings of the 23rd International Conference on Machine Learning - ICML '06*. Pittsburgh, Pennsylvania: ACM Press, 2006, pp. 233–240.

[48] J. Paparrizos, Y. Kang, P. Boniol, R. S. Tsay, T. Palpanas, and M. J. Franklin, "Tsb-uad: An end-to-end benchmark suite for univariate time-series anomaly detectiontsb-uad," *Proceedings of the VLDB Endowment*, vol. 15, no. 8, pp. 1697–1711, Apr. 2022.

[49] J. Arevalo, T. Solorio, M. Montes-y Gómez, and F. A. González, "Gated multimodal units for information fusion," *arXiv preprint arXiv:1702.01992*, 2017.

[50] R. Wu and E. Keogh, "Deep learning time series anomaly detection algorithms are brutally sensitive to concept drift," 2024, https://lnkd.in/g7qWVTpS.

[51] Y. Lu, R. Wu, A. Mueen, M. A. Zuluaga, and E. Keogh, "Matrix profile xxiv: Scaling time series anomaly detection to trillions of datapoints and ultra-fast arriving data streams," in *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. Washington DC USA: ACM, Aug. 2022, pp. 1173–1182.

[52] H. Si, J. Li, C. Pei, H. Cui, J. Yang, Y. Sun, S. Zhang, J. Li, H. Zhang, J. Han, D. Pei, and G. Xie, "Timeseriesbench: An industrial-grade benchmark for time series anomaly detection models," in *2024 IEEE 35th International Symposium on Software Reliability Engineering (ISSRE)*, 2024, pp. 61–72.

[53] J. Davis and M. Goadrich, "The relationship between precision-recall and roc curves," in *Proceedings of the 23rd International Conference on Machine Learning*, ser. ICML '06. Association for Computing Machinery, 2006, pp. 233–240.

[54] J. M. Lobo, "Auc: a misleading measure of the performance of predictive distribution models," *Global Ecology & Biogeography*, vol. 17, no. 2, pp. 145–151, 2010.

[55] A. Garg, W. Zhang, J. Samaran, R. Savitha, and C.-S. Foo, "An evaluation of anomaly detection and diagnosis in multivariate time series," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 6, pp. 2508–2517, Jun. 2022.

[56] D. Li, S. Zhang, Y. Sun, Y. Guo, Z. Che, S. Chen, Z. Zhong, M. Liang, M. Shao, M. Li, S. Liu, Y. Zhang, and D. Pei, "An empirical analysis of anomaly detection methods for multivariate time series," in *2023 IEEE 34th International Symposium on Software Reliability Engineering (ISSRE)*. Florence, Italy: IEEE, Oct. 2023, pp. 57–68.

[57] J. Chung, C. Gulcehre, K. H. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *Eprint Arxiv*, 2014.

[58] P. Tang and W. Zhang, "Unlocking the power of patch: Patch-based mlp for long-term time series forecasting," vol. 39, no. 12, pp. 12 640–12 648.

[59] E. Keogh, "Multi-dataset time series anomaly detection competition," 2021, https://compete.hexagon-ml.com/practice/competition/39/#evaluation.

A Simulated anomalies

We simulated anomalies by distorting a portion of the input time series window $\mathbf{X}$. In detail, we randomly select an interval $[t'_1, t'_2] \subset [t_1, t_2]$ in the input window $\mathbf{X} = \{x_{t_1} \dots x_{t_2}\}$, and replace the values $\mathbf{X}_{[t'_1, t'_2]} = \{x_{t'_1} \dots x_{t'_2}\}$ with one of the following anomalies (see Figure 7):

- **Uniform Replacement:** The original values are replaced with a constant sequence with values in the range $\{min(X'), max(X')\}$.
- **Mirror Flip:** The original values are flipped across the x-axis or the y-axis.
- **Length Scale:** The original sequences are substituted with lengthened or shortened versions of themselves.
- **Jittering:** The original values are added with random noise.

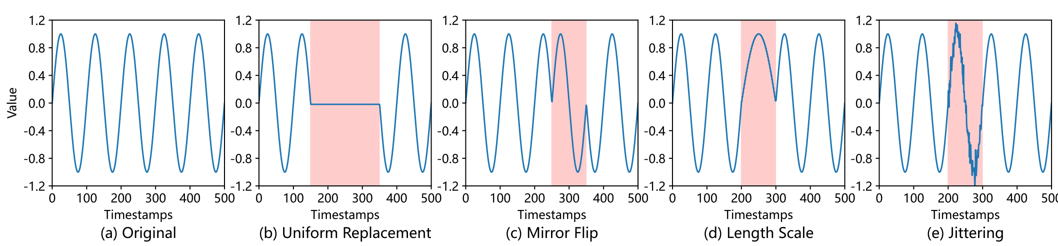

Figure 7: Illustration of the simulated anomalies.

The interval length $L_{inter} = t'_2 - t'_1$ is randomly and uniformly sampled from the range $[0.1 * L_{period}, L_{period}]$, where L_{period} is the length of the dominant period. Each input window is randomly injected with one of the four aforementioned types of anomalies.

B Datasets and metrics

B.1 Datasets

The KDD21 dataset [24], also known as the UCR Anomaly Archive, is widely acknowledged as the highest-quality benchmark in the field of time series anomaly detection.[18, 12, 51, 52]. It compromises 250 subsets drawn from diverse domains such as healthcare, sports, industry, and robotics. Notably, the anomalies in the subsets are challenge to detect, and could't be easily addressed by the "one-liner" approach [24]. In addition, the KDD21 dataset provides a document explaining why certain regions are labeled as anomalies. This further enhances the credibility and transparency of the dataset.

The TSB-AD dataset [9] is the recently proposed the largest scale dataset for time series anomaly detection. It curates and manually cleanses datasets from various sources. However, it has significantly overlap with the KDD21 dataset, and several subsets, such as NAB, WSD, YAHOO and Stock still suffer from issues including mislabeled ground truth, trivial anomalies, and unrealistic anomaly densities [26]. Therefore, we exclude the overlapping parts and adhere to the criteria outlined in prior research on dataset quality [24, 46, 3] to select several high-quality subsets from the TSB-AD dataset. The selected subsets are: MGAB, SED, SVDB, IOPS, and TODS.

The statistics of the datasets are summarized in Table 3. The anomaly ratio is calculated from the ratio between the sum of all anomaly points and sum of all test points.

Table 3: Statistics of datasets.

Datasets	Subsets Num	Avg Train Length	Avg Test Length	Avg Anomaly Length	Anomaly Ratio
KDD21	250	8953	24574	147	0.6%
TSB-AD	64	12163	88907	56	2.7%

B.2 Metrics

Recent studies [9, 18, 44] have pointed out that widely used evaluation metrics such as F1 with point adjustment and F1-Affiliation tend to significantly overestimate model performance, even assigning high evaluation scores to randomly generated predictions. In addition, anomaly detection datasets are highly class imbalanced, with normal points far outnumbering anomalous points. The AUC-ROC (Area Under the Receiver Operating Characteristic Curve) metric is biased towards the majority class, leading to inflated evaluation scores [53]. AUC-PR (Area Under the Precision-Recall Curve) has been advocated as a more informative alternative for imbalanced datasets [54]. Therefore, a recent benchmark evaluation paper has demonstrated that Standard-F1, AUC-PR [47], Range-AUC-PR [45] and VUS-PR [45] are the most reliable and accurate metrics for assessing model performance. The details of the evaluation metrics are described as follows:

- **Standard F1:** The standard F1 is calculated as the harmonic mean of precision and recall. To mitigate the influence of threshold selection methods on evaluation results, and keep consistency with prior works [9, 55, 44, 56], we use the maximum F1-score among all possible thresholds.
- **AUC-PR:** This metric measures the area under the precision-recall curve.
- **Range-AUC-PR:** This method addresses the nuances of labeling consistency and the impact of time lags on anomaly scores by adding a buffer to the boundaries of anomalies, thereby giving some credit to the high anomaly score in the vicinity of the anomaly boundary. According to the original Range-AUC-PR paper [45], we set the buffer length to be the average anomaly length in the dataset.
- **VUS-PR:** Since the selection of the buffer length can significantly affect the Range-AUC-PR results, the VUS-PR metric [45] address this limitation by incorporating the possible buffer region lengths $\mathcal{L} = [\ell_0, \ell_1, \ldots, \ell_L]$ with $0 = \ell_0 < \ell_1 < \ldots < \ell_L = \ell$. We set ℓ to be the average anomaly length in the dataset.

C Implementation details

C.1 Baseline settings

Deep learning based methods: we reimplement all the deep learning-based models based on their official open-source codes and follow the configurations recommended in their papers.

Non-deep learning models: All the non-deep learning models are implemented based on the TSB-UAD [48] and TSB-AD [9] library. The hyperparameters for these methods are set according to the TSB-AD [9] paper, where they are tuned on a large-scale validation set.

We also provide implementations of baseline methods in the repository `https://anonymous.4open.science/r/BRC-E46B` , enabling direct reproduction of baseline results.

C.2 Hyperparameters of CoAD

In all experiments, we use the Gate Recurrent Unit(GRU) [57] as the encoder, with a hidden size of 24 and 3 recurrent layers. The input window size T is set to 4 times the dominant period of the time series, and the patch size P is fixed to 8. We take the autocorrelation function (ACF) to find the dominant period of the time series. The weight λ is set to 10 and the number of frequency bands K is set to 4. **Appendix D provides an in-depth exploration of the hyperparameters.**

All experiments are conducted with the hardware configuration of an Intel i9-12900K CPU and 1 NVIDIA RTX 3090 GPU.

D Parameter study

The input window size T, patch size P, and the loss weight λ are most critical hyperparameters for the performance of CoAD. We explore the impact of these hyperparameters on the performance of CoAD on the KDD21 and TSB-AD datasets. As shown in Figure 8, increasing the input window size generally improves model performance. This is because larger windows allow the model to incorporate more contextual information to detection anomalies more effectively. However, when the window size exceeds 4 times the dominant period, the performance gains become marginal while incurring additional computational overhead. Therefore, we set the input window size to be 4 times the dominant period for all experiments. Figure 8 shows that our model performances do not vary significantly with different patch sizes. A patch size that is too small may fail to capture sufficient local information, while an overly large patch size may lead to excessive smoothing and compression of the data [58]. The ideal patch size may be dataset dependent, but patch sizes between $\{8, 12\}$ seem to be general good choices. Since the classification loss is significantly larger than the reconstruction loss, λ is used to balance the two losses. λ ranging from 5 to 10 generally provides good performance.

E Ablation study implementation details

We consider the following variants:

1. **OE and MAE variants works alone:**
   - *OE (Time):* Only the time domain features are used for classification.
   - *OE (Time + Frequency):* The time domain classification results and frequency domain classification results are ensembled using the maximum fusion strategy.
   - *MAE (Random):* The MAE takes the random masking strategy.
   - *MAE (Grating):* The MAE takes the grating masking strategy.
2. **Different cooperative strategies**
   - *OE+MAE (Random):* The anomaly score is obtained by adding the anomalous probability produced by OE and the reconstruction error generated by *MAE (Random)*.
   - *OE+MAE (Grating):* The anomaly score is obtained by adding the anomalous probability produced by OE and the reconstruction error generated by *MAE (Grating)*.

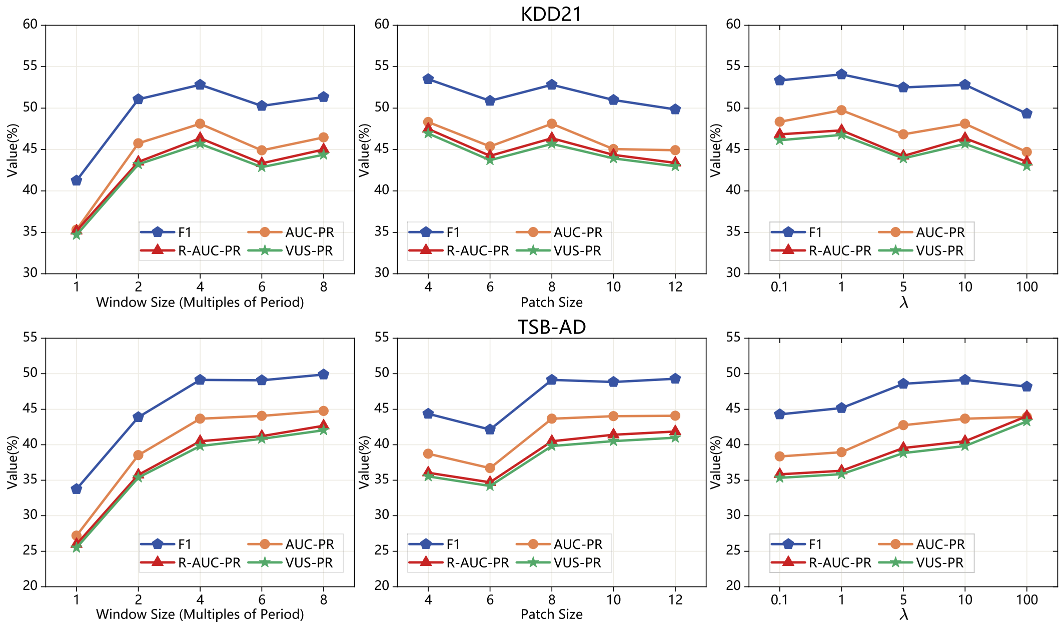

Figure 8: Parameter analysis.

- *OE+MAE (Guide w/ Hard Mask):* During training, the MAE is trained using a random masking strategy. During testing, we compute the anomalous probability of all patches in the training set and set a threshold as the mean plus three standard deviations. Patches with anomalous probabilities exceeding this threshold are then masked using hard (discrete) masks.

3. **Cooperation under different design choices:**
   - *OE (Step Level)+MAE:* The OE module takes step-level classification granularity like AnomalyBERT [15]. The mean anomalous probability of all points within a patch is taken as the anomalous probability of the patch and serves as its soft mask.
   - *OE (Window Level)+MAE:* The OE module takes window-level classification granularity like CutAddPaste [16]. We take the sliding window method with stride size 1 to obtain the anomalous probability for each point. After that, we compute the anomalous probability and soft mask for each patch using the same method as in *OE (Step Level)+MAE*.
   - *OE (Feature Add)+MAE:* The OE module takes the feature-level fusion strategy, where the features from both domains are directly added together. The fused features are then fed into the classifier.
   - *OE (Feature Gated)+MAE:* The OE module takes the feature-level fusion strategy, where the features from both domains are multiplied by a learnable gate and then added together.
   - *OE (Decision Mean)+MAE:* The OE module takes the decision-level fusion strategy, where the predictions from the time domain and frequency domain classifiers are averaged.
   - *OE+MAE (Guide w/o Score):* The MAE module is guided by the OE module, but the anomaly score is derived solely from the reconstruction error of MAE, ignoring the anomalous probability generated by OE.

F Evaluation results on KDD21 dataset using top-k accuracy scores.

The KDD21 dataset originates from the SIGKDD Cup 2021 competition, and we also adopt the official evaluation metric provided by the competition organizer [59]. Each of the 250 subsets in the KDD21 dataset contains a single anomaly, and each algorithm identifies the position with the highest anomaly score (top-1) as the predicted anomaly location. If this predicted location falls within ± 100 data points of the true location, it is considered correct and assigned an accuracy score of 1. Otherwise, if it lies outside this range, it is deemed incorrect and given a score of 0. The final accuracy score is the average of the accuracy scores across all subsets. In addition to top-1 accuracy, we also consider top-3 and top-5 accuracy scores, as real-world time series may contain multiple plausible anomalies. Thus, top-k accuracy provides a more comprehensive evaluation.

Table 4 presents the evaluation results on the KDD21 dataset using top-k accuracies (Acc.@k). The results show the outstanding performance of our proposed COAD framework.

Table 4: The evaluation results on the KDD21 dataset using top-k accuracies (Acc.@k). The best results are in **bold**, and the second best results are with underline.

Model / Dataset		**KDD21**		
Model Class	Model(Venue)	Acc.@1	Acc.@3	Acc.@5
MAE	DADA (ICLR-25)	7.47	11.62	17.84
	MOMENT (ICML-24)	15.70	21.49	25.21
	MMA (VLDB-25)	40.00	52.40	58.40
	TFMAE (ICDE-24)	3.48	7.39	12.61
OE	AnomalyBERT (ICLR-23)	9.64	15.66	57.43
	CutAddPaste (KDD-24)	55.20	64.80	71.20
	TriAD (ICDE-24)	25.60	34.00	34.00
	COUTA (TKDE-24)	15.48	24.27	28.45
Time-Frequency Reconstruction	FCVAE (WWW-24)	37.60	48.40	56.00
	TFAD (CIKM-22)	4.13	6.61	11.57
	CATCH (ICLR-25)	33.06	45.04	50.41
Reconstruction/ Prediction	TranAD (VLDB-22)	24.00	29.60	33.60
	MAUT (ICASSP-23)	30.40	39.20	43.60
	M2N2 (AAAI-24)	11.98	17.36	21.49
Data Mining	KshapeAD (NeurIPS-24)	46.00	58.80	65.60
	SAND (VLDB-21)	47.20	58.40	64.80
	Sub-PCA (NeurIPS-24)	21.68	28.92	34.14
	Series2Graph (VLDB-20)	36.40	50.00	55.20
	KmeansAD (VLDB-22)	40.00	52.80	59.60
	Matrix Profile (CIKM-16)	<u>56.80</u>	<u>72.40</u>	<u>78.40</u>
	DAMP (KDD-22)	38.40	44.40	51.20
Cooperative	**BCR(ours)**	**61.60**	**76.80**	**80.80**

G Efficiency results on TSB-AD dataset.

Experiment settings. We comprehensively compare the detection performance, inference speed, and model params of COAD against baseline methods. Following existing works [18, 51], we report the total inference time across all subsets as the overall inference time. To ensure a fair comparison, all deep learning-based models are evaluated with the same batch size of 128, the window size that yields the highest VUS-PR score, and are executed on the same NVIDIA RTX 3090 GPU. All Data Mining-based methods are tested on the same Intel i9-12900K CPU as they don't support GPU parallel processing.

Efficiency results. The model efficiency comparison on the TSB-AD dataset is shown in Figure 9. Our proposed framework COAD outperforms the state-of-the-art methods in terms of both efficiency and performance. While MMA and KmeansAD demonstrate performance levels comparable to COAD, COAD achieves inference speeds that are several orders of magnitude faster.

H Qualitative ablation results

We further provide visualization results to demonstrate the effectiveness of each module in COAD. Figure 10(a) shows that incorporating frequency domain features helps the classification module detect anomalies that are difficult to identify in the time domain. Figure 10(b) shows the importance of the cooperation between the reconstruction and classification modules. Some anomalies overlooked by one module can be effectively detected by the other module. Figure 10(c) and (d) highlight the superiority of our proposed guided masking strategy over existing random and grating masking methods. The random masking method yields high reconstruction errors even in normal regions, leading to severe false alarms. The grating mask method tends to overfit anomalies, resulting in false negatives. Our proposed guided masking strategy provides the most accurate detection results, ensuring the optimal detection performance of the reconstruction module.

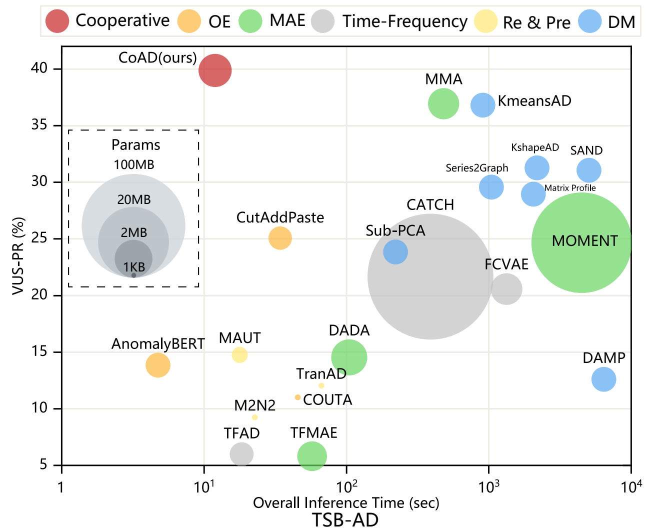

Figure 9: Model efficiency comparison on the TSB-AD dataset.

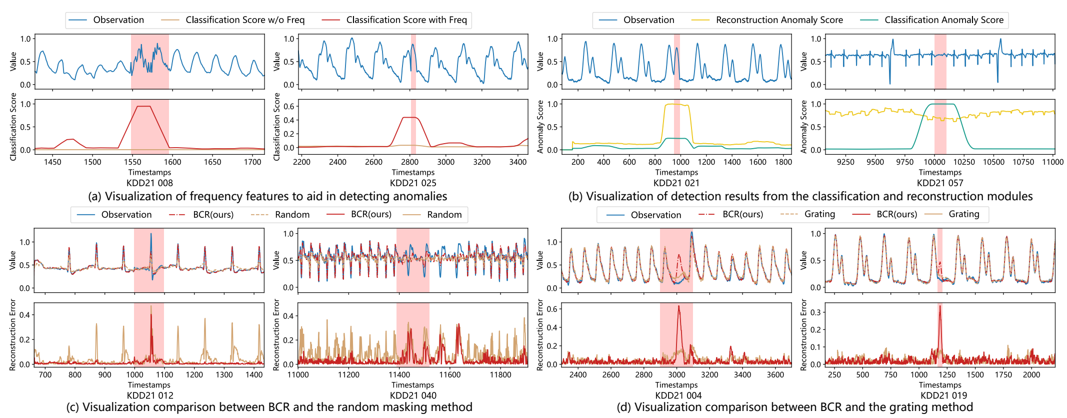

Figure 10: Visualization demonstration of the effectiveness of each module in CoAD.

I Limitation and broader impact

Limitation. The classification module and the reconstruction module are integrated in a sequential manner, which prevents parallel execution and consequently increases the overall time complexity.

Broader impact. As far as we know, our work is the first deep learning approach to outperform SOTA data mining-based methods on both the KDD21 and TSB-AD datasets, thereby reevaluating the potential of deep learning in time series anomaly detection.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS paper checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: **[TODO]**

   Justification: **[TODO]**

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: **[TODO]**

   Justification: **[TODO]**

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: **[TODO]**

   Justification: **[TODO]**

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: **[TODO]**

   Justification: **[TODO]**

   Guidelines:

   - The answer NA means that the paper does not include experiments.

- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: **[TODO]**

Justification: **[TODO]**

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so No is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).

- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: **[TODO]**

   Justification: **[TODO]**

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: **[TODO]**

   Justification: **[TODO]**

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: **[TODO]**

   Justification: **[TODO]**

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.

- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: **[TODO]**

   Justification: **[TODO]**

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: **[TODO]**

    Justification: **[TODO]**

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: **[TODO]**

    Justification: **[TODO]**

    Guidelines:

    - The answer NA means that the paper poses no such risks.

- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: **[TODO]**

    Justification: **[TODO]**

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: **[TODO]**

    Justification: **[TODO]**

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: **[TODO]**

    Justification: **[TODO]**

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: **[TODO]**

    Justification: **[TODO]**

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.