EXPLAINABLE TIME SERIES ANOMALY DETECTION USING MASKED LATENT GENERATIVE MODELING

Daesoo Lee
Norwegian University of Science and Technology

Sara Malacarne
Telenor Research

Erlend Aune
Norwegian University of Science and Technology
BI Norwegian Business School
HANCE

ABSTRACT

We present a novel time series anomaly detection method that achieves excellent detection accuracy while offering a superior level of explainability. Our proposed method, *TimeVQVAE-AD*, leverages masked generative modeling adapted from the cutting-edge time series generation method known as TimeVQVAE. The prior model is trained on the discrete latent space of a time-frequency domain. Notably, the dimensional semantics of the time-frequency domain are preserved in the latent space, enabling us to compute anomaly scores across different frequency bands, which provides a better insight into the detected anomalies. Additionally, the generative nature of the prior model allows for sampling likely normal states for detected anomalies, enhancing the explainability of the detected anomalies through *counterfactuals*. Our experimental evaluation on the UCR Time Series Anomaly archive demonstrates that TimeVQVAE-AD significantly surpasses the existing methods in terms of detection accuracy and explainability. We provide our implementation on GitHub: `https://github.com/ML4ITS/TimeVQVAE-AnomalyDetection`.

Keywords Time Series Anomaly Detection (TSAD) · TimeVQVAE-AD · TimeVQVAE · Masked Generative Modeling · Explainable AI (XAI) · Explainable Anomaly Detection

1 Introduction

Time series anomaly detection (TSAD) is a critical area of study in data analysis and machine learning, aiming to identify unusual patterns that deviate from expected behavior in fields such as finance, healthcare, and telecommunication. Various methods have been proposed for TSAD, leveraging techniques such as one-class classifications [1, 2], isolation forest [3, 4], discord discovery [5, 6], reconstruction [7, 8, 9, 10], forecasting [11, 12], and density estimation [13, 14, 15]. These methods have seemingly shown progress, primarily driven by deep learning methods. However, recent studies have exposed significant flaws in the popular benchmark datasets and evaluation protocols used in TSAD research [16, 17, 18]. The benchmark datasets often suffer from unrealistic anomaly density and mislabeled ground truth [16]. Additionally, these datasets may contain trivial problems that do not adequately assess the performance of TSAD methods. As for the evaluation protocol commonly used in TSAD, it is called Point Adjustment (PA), proposed by [19]. However, this protocol has been criticized by [18] due to its potential to inflate performance results. To address these issues, [16] released the UCR Time Series Anomaly (UCR-TSA) archive that contains 250 curated benchmark datasets to provide more accurate evaluations. [16] also suggested a simple scoring function to achieve a robust evaluation metric. Since the introduction of the UCR-TSA archive in the literature, multiple studies [17, 6, 20] have evaluated existing TSAD methods using this benchmark. The findings of these studies were quite stunning, as they revealed that most of existing deep learning-based TSAD methods exhibit noticeably lower detection accuracies than their non-deep learning-based counterparts, contradicting their claims of state-of-the-art (SOTA) performance. Furthermore, many papers plot few examples (as few as zero) even though time series analytics is inherently a visual domain [16].

In this paper, we present a novel TSAD method that achieves exceptional anomaly detection accuracy and offers a high level of explainability. The overview of the inference process of our method is illustrated in Fig. 1. Unlike most of the existing deep methods for TSAD, our approach leverages masked generative modeling. Masked generative modeling has demonstrated significant success in diverse fields, ranging from generative language modeling, such as BERT [21], to generative image modeling, such as DALL-E [22]. In this work, we introduce a novel approach in which we predict anomaly scores directly from a learned prior using a robust prior model from TimeVQVAE [23]. TimeVQVAE is a SOTA time series generative method that has demonstrated superior performance in the literature [24]. Once the prior model is trained and learns the prior distribution using a training dataset, it can assign high probabilities to likely normal subsequences and low probabilities to abnormal subsequences of the time series. These probabilities can then be employed to calculate anomaly scores through the negative log-likelihood. Our proposed method also benefits from TimeVQVAE for its explainability. The prior model of TimeVQVAE is trained on a time-frequency domain, rather than a time domain. This characteristic enables us to calculate anomaly scores across different frequency bands, consequently facilitating the factorization of anomalies in terms of anomaly types with respect to different frequency bands. Additionally, presentation of likely normal states can be effortlessly accomplished by masking anomalous segments and conducting sampling using the learned prior model since the prior model is fundamentally generative. This allows us to approach explainable AI (XAI) for TSAD within the *counterfactual* framework [25, 26]. The counterfactuals we generate verify, by definition, specific metrics adopted in this domain, such as 1) *validity*, 2) *plausibility* and 3) *low computability time* [25]. Finally, we call our proposed method *TimeVQVAE-AD*.
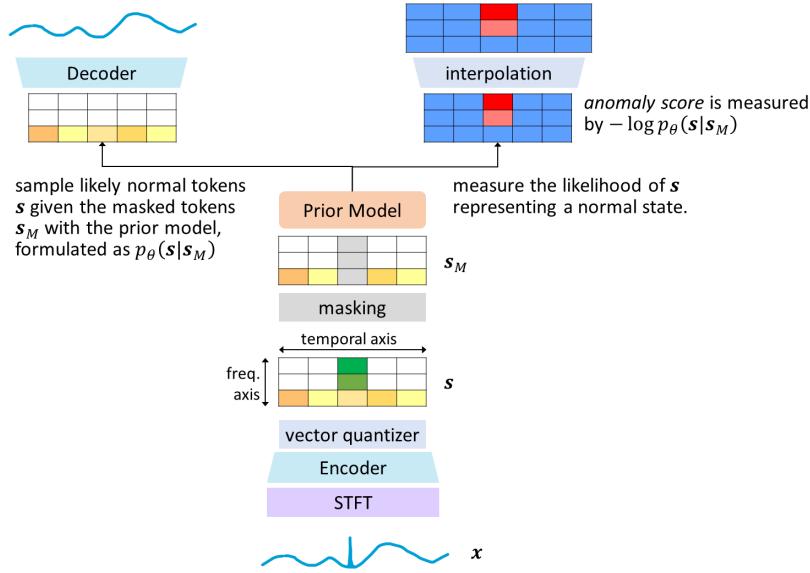

Figure 1: Overview of the inference process of our proposed method, TimeVQVAE-AD. In the figure, the time series x exhibits a high-frequency anomaly. Initially, x is processed by Short Time Fourier Transform (STFT), followed by processing through an encoder and a vector quantizer, resulting in s. Here, s represents a set of tokens derived from x, with color similarity indicating the Euclidean similarity between these tokens. The two axes of s correspond to time and frequency, respectively. Subsequently, a segment of s is masked to enable the prior model to sample likely normal tokens from the masked tokens (*explainable sampling*), and to compute the anomaly scores (*anomaly detection*) where low and high anomaly scores are depicted in blue and red, respectively. A noteworthy aspect of our method is its capability to address a broad spectrum of anomalies, due to the robust generative prior model that utilizes a learned prior to evaluate the likelihood of s representing a normal state.

The experimental evaluation is performed on the UCR-TSA archive, and the experimental results demonstrate that TimeVQVAE-AD surpasses the performance of existing methods in terms of detection accuracy. Furthermore, it provides a superior level of explainability that, to the best of our knowledge, has not been achieved by any previous method. To boost confidence and transparency of our proposed method, we provide visualizations and CSV files of predicted anomalies for all 250 UCR-TSA archive datasets in our GitHub repository. This visualization evaluation protocol has been strongly recommended by [16, 6] by pointing out at the issue that proper visualization practices have been significantly lacking in the literature despite the fact that TSAD belongs to the domain of time series analytics.

In summary, our contributions are

- TSAD via masked generative modeling,

- explainability via factorization of anomalies in terms frequency bands,
- explainability via counterfactuals,
- ground-breaking anomaly detection accuracy and explainability for TSAD,
- fair and robust evaluation on the UCR-TSA archive,
- availability of visualization for predicted anomalies across the UCR-TSA archive for transparency.

2   State of Time Series Anomaly Detection

Over the years, numerous methods have been proposed for TSAD utilizing a range of mechanisms such as one-class classifications, isolation forest, discord discovery, reconstruction, forecasting, and density estimation. Especially, a gradual improvement on the reported metrics has been observed, largely driven by the utilization of modern deep learning methods [9, 10]. This progress has seemed to indicate that the field has been advancing.

However, in recent years, a number of critical papers have emerged, highlighting significant flaws in the currently-popular benchmark datasets and evaluation protocol employed for TSAD methods [16, 17, 18]. These papers reveal that the literature has been misled by inaccurately assessed metric scores, generating a false perception of progress while actual advancements have been minimal.

The recent study conducted by [16] shed light on several critical issues regarding the currently-popular benchmark datasets employed to evaluate TSAD methods. The datasets that are commonly used now as benchmarks include Yahoo [27], Numenta [28], NASA [11], and OMNI [29]. [16] argued that those datasets often suffer from mislabeled data, containing both false positives and false negatives, thereby leading to an inaccurate measure of detection accuracy. In addition, [16] emphasized the presence of inconsistencies in labeling when dealing with similar instances of anomalies, resulting in an underestimation of the true positive rate for methods capable of detecting such anomalies. Furthermore, the study revealed that a substantial portion of the time series within those datasets can be easily solved through simple solutions, indicating that the trivial nature of the datasets fails to adequately assess the performance of TSAD methods. The flaws in these benchmark datasets have created a misleading perception of progress within the TSAD field, as methods that perform well on those datasets may not exhibit the same performance in real-world scenarios. To address these issues, [16] released a set of 250 carefully-curated benchmark datasets named UCR-TSA archive, where the datasets are free from these flaws and provide a more accurate evaluation of TSAD methods.

Another critical study [18] has pointed out the flaws of evaluating TSAD using the PA protocol. The idea behind PA is that if at least one moment in a contiguous anomaly segment is detected as an anomaly, the entire segment is then considered to be correctly predicted as an anomaly. The authors demonstrated that PA can lead to overestimation of the model's performance. For instance, even if the anomaly scores are randomly generated and cross the threshold only once within the ground truth segment, after applying PA, these predictions become indistinguishable from those of a well-trained model. This means that random anomaly scores can yield high F1 scores after PA, making it difficult to conclude that a model with a higher F1 score after PA performs better than others. To tackle the issue and establish a robust evaluation protocol, the author of the UCR-TSA archive suggested a simple yet fair evaluation protocol [30]. During the evaluation process, it is expected that a method provides a singular predicted anomaly location. If this predicted location falls within a range of ±100 data points from the true location, it is considered correct, thereby achieving an accuracy of 1.0. On the other hand, if the predicted location deviates outside this range, it is considered incorrect, resulting in an accuracy of 0.0. Then the accuracies over the 250 datasets are averaged, and that is reported for the comparative evaluation.

Experimental evidence supporting the flawed evaluation practices and the resultant misleading findings can be found in the literature [6, 17, 20], where the experiments were conducted on popular deep TSAD methods. For instance, TranAD [10] is a recent deep TSAD method built on a transformer model [31] and it claimed to be SOTA for TSAD. In its paper, TranAD outperformed its 10 competing methods, achieving F1 score of 0.94 on the Numenta dataset and 0.89 on the NASA dataset. However, when evaluated on the UCR-TSA archive, TranAD exhibited remarkably poor performance, achieving the averaged accuracy of merely 0.16 – even worse than the autoencoder (AE)-based method that achieved an accuracy of 0.28 [7].

One additional limitation we have observed in the existing methods is the absence of explainability. Given that TSAD operates within the domain of time series analytics, it is essential for models to provide users with diagnostic explanations. These explanations play a critical role in delivering valuable insights about the detected anomalies, thereby enabling users to gain a deeper understanding of the underlying reasons behind them. In our work, we provide two important perspectives for XAI for TSAD: 1) *factorization of anomalies in terms of anomaly types*, 2) *presentation of likely normal realizations, i.e., counterfactuals*. As for the first perspective, there exists a diverse range of anomalies in time series data, including local peaks, noise, steep increases, signal shifts, unusual patterns, and more [17]. These different types of anomalies can often be categorized based on their frequency characteristics. For instance, local peaks
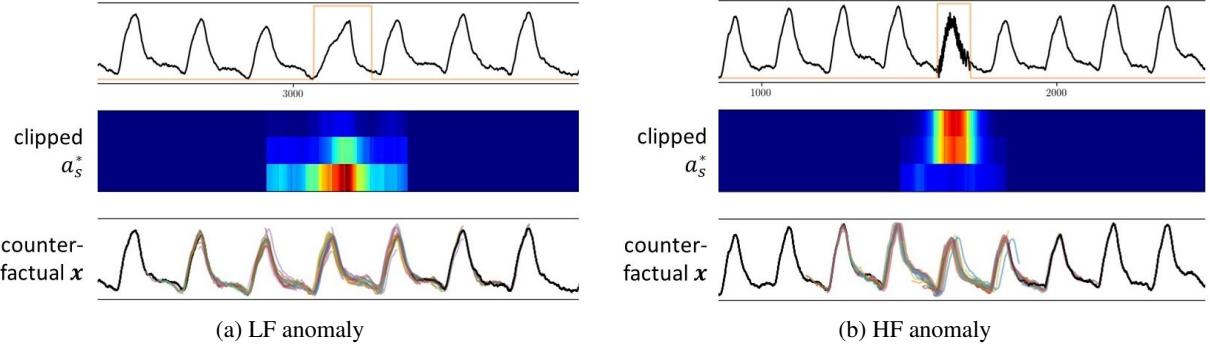
(a) LF anomaly

(b) HF anomaly

Figure 2: Example of the two perspectives of explainability: 1) factorization of anomalies in terms of anomaly types via frequency decomposition, 2) presentation of a corresponding normal state. The two subfigures show time series with different anomaly types: LF and HF anomalies, respectively. In each subfigure, the first figure presents a time series (black) with anomaly labels (orange), the second figure presents predicted anomaly scores with respect to different frequency bands using our proposed method, where the bottom row and the top row represent the lowest and highest frequency band, respectively (blue: low anomaly score, red: high anomaly score), and the third figure shows the likely normal states of the time series, in which the anomalous segments are resampled using our learned prior model. Note that the predicted anomaly scores are high in the low frequency band in (a), the scores are high in the high frequency band in (b), and the likely normal states are highly convincing when observed by human eyes.

and i.i.d noise can be considered high-frequency (HF) anomalies, while signal shifts and drifts can be classified as low-frequency (LF) anomalies. Additionally, unusual patterns can exhibit anomalous behavior in both LF and HF frequency bands. This anomaly type factorization in terms of frequency offers better interpretability of the detected anomalies, as depicted in Fig. 2, in which a HF anomaly may indicate a sensor noise and a LF anomaly can indicates a systematic change in the measured subject. The second perspective of explainability involves providing insights into what the data would look like in its non-anomalous state – counterfactuals in the XAI literature [25]. For example, a patient diagnosed with a brain tumor can better comprehend the doctor's assessment by comparing their brain image to that of a normal brain [32]. Similarly, this principle can be applied to TSAD, as depicted in Fig. 2. By offering a comparison between an anomalous time series and corresponding likely realizations over the anomaly window, an enhanced explainability of the detected anomalies can be achieved.

3 Related Work

3.1 Existing Anomaly Detection Methods

The existing methods for TSAD can be categorized into two groups: 1) Non-Deep Learning (DL)-based methods, and 2) DL-based methods.

3.1.1 Non-Deep Learning-based TSAD Methods

Non-DL-based TSAD methods utilize various mechanisms, including one-class classification [33, 2], isolation forest [3, 4], density estimation [13], matrix profile [34, 35], and discord discovery [5, 6]. Among these approaches, matrix profile techniques such as SCRIMP [34] and discord discovery methods such as MERLIN++ [6] demonstrate the highest detection accuracy. Furthermore, MERLIN++ offers greater computational efficiency. Both matrix profile and discord discovery methods calculate anomaly scores by measuring the distances between different subsequences of time series data. However, there is a difference in their computational requirements. Matrix profile methods can be computationally intensive due to the need to calculate pairwise distances between all pairs of subsequences. On the other hand, discord discovery focuses on identifying subsequences with large pairwise distances, resulting in a significant reduction in computational cost.

3.1.2 Deep Learning-based TSAD Methods

With the advancements in DL, many approaches for TSAD based on DL have emerged. The most common existing DL methods are based on reconstruction or forecasting tasks [36, 11]. Additionally, there have been attempts based on different approaches such as adversarial training [37, 38, 39], density estimation [14, 15], and (non-)contrastive learning [40, 20]. Moreover, some recent studies such as [41, 42] adopted masked modeling with anomaly score

measurement based on reconstruction. Although these studies utilize masked modeling, their approaches and our approach are fundamentally different in terms of anomaly detection because their anomaly scores are measured based on reconstruction while ours on density estimation. However, the critical papers by [17, 6] have demonstrated that the DL-based TSAD methods do not meaningfully outperform non-DL methods such as SCRIMP, MERLIN [5], and MERLIN++ when fairly evaluated on the UCR-TSA archive. We have identified the intrinsic limitations of DL-based methods as follows:

Limitation of Reconstruction or Forecasting-based TSAD Methods   The reconstruction or forecasting-based methods are trained by minimizing the error $\|x_{\text{train}} - \hat{x}\|$, where x_{train} represents the model input during training and $\hat{x}$ represents the model output for reconstruction or forecasting. Once the model is properly trained and well regularized, it should perform well on reconstructing x_{test}, resulting in a small test error and equivalently, a small anomaly score. Anomalies with unusually-high amplitude can be easily captured since their errors will be large due to the large amplitude of x_{test}. However, these methods inherently struggle to detect anomalies with small amplitude or subtle pattern differences, as depicted in Fig. 3. Experimental evidence aligned with this claim is provided in [17].

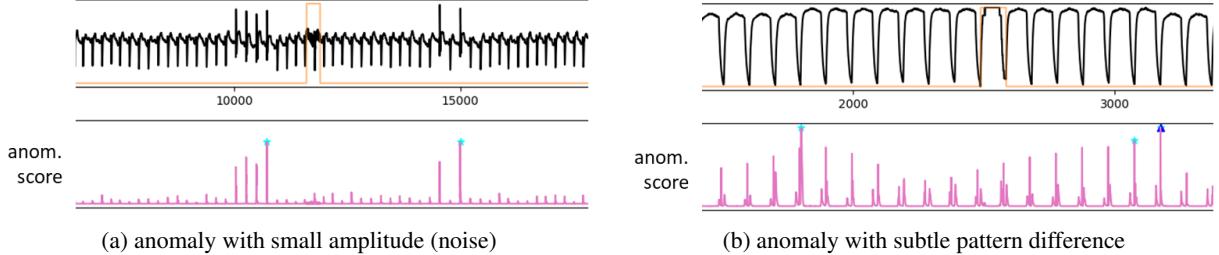| (a) anomaly with small amplitude (noise) | (b) anomaly with subtle pattern difference |

Figure 3: Examples of inevitable failure cases for reconstruction or forecasting-based TSAD methods. In each subfigure, the second figure presents predicted anomaly scores (pink). A reconstruction or forecasting error $\|x_{\text{train}} - \hat{x}\|$ tends to be larger on timesteps with large amplitudes, as they are more challenging to predict. This results in higher anomaly scores at peaks. In contrast, anomalies with small amplitudes inevitably yield low predicted anomaly scores due to their nature, as observed in (a) and (b).

Limitation of Adversarial Learning-based TSAD Methods   The adversarial learning or Generative Adversarial Network-based methods involve training a generator and discriminator, where the discriminator is trained to distinguish between real and generated samples. These methods measure anomaly scores by utilizing the discriminator score. However, using the discriminator score as an anomaly score can lead to potential issues, as the discriminator is primarily designed to distinguish between real and generated samples rather than specifically identifying anomalies – that is, there exists a definitive distinction between fake samples and anomalous samples. Furthermore, due to mode collapse in the generator, it is unlikely to generate a diverse range of anomalous patterns. As a result, the discriminator may not be able to accurately assign discriminator scores to unseen anomalous patterns.

Limitation of Existing Density Estimation-based TSAD Methods   The density estimation-based methods aim to measure anomaly scores via a learned prior distribution. DAGMM [14], which stands for Deep Autoencoding Gaussian Mixture Model, is a highly-cited method for anomaly detection, and [43] adopted it for time series. DAGMM consists of a compression network and an estimation network. The compression network compresses a sample into a latent vector, and the estimation network performs density estimation under the framework of Gaussian Mixture Modeling (GMM). The anomaly score is computed based on the likelihood of the latent vector in the Gaussian distribution modeled by GMM. However, DAGMM faces a similar challenge as the reconstruction or forecasting-based methods, as the latent vector obtained with the compression network retains the shape information of a sample, making it likely that similarly-shaped time series are located close in the Gaussian distribution [44]. Consequently, it is prone to failing in capturing anomalies with small amplitude or subtle pattern difference. Another recent proposal, GANF (Graph Augmented Normalizing Flow), measures anomaly scores based on its learned prior using normalizing flow. However, this approach is inherently limited by the properties of normalizing flow. [45] showed the failure of normalizing flows in detecting out-of-distribution data and investigated the reasons behind it. They found that the failure can be attributed to the model's inability to learn high-level semantic features and its shortcomings in global structure learning.

While most TSAD methods operate within a time domain, a study conducted by [46] proposed measuring anomaly scores across both time and frequency domains while using a distance metric for anomaly score measurement. The utilization of a frequency domain allows for capturing anomalous time series with a small magnitude in a time domain and a high magnitude in a frequency domain, at which the reconstruction-based methods fail. However, it only partially

resolves the limitation of the reconstruction-based methods because anomalies can only be captured when they carry a high magnitude either in a time or frequency domain due to the nature of a distance metric. Consequently, this limitation hinders its ability to detect subtle anomalies.

3.2 TimeVQVAE: A Powerful Time Series Generation Method

TimeVQVAE introduces a novel approach for time series generation, inspired by the success of masked image generative modeling [47, 48, 49]. TimeVQVAE focuses on producing high-quality synthetic samples through two key stages: tokenization (stage 1) and prior learning (stage 2). In the tokenization stage, TimeVQVAE utilizes vector quantization modeling in the time-frequency domain. This allows for effective representation and encoding of time series data. The subsequent prior learning stage involves training transformer models to learn a prior distribution of time series data. Notably, TimeVQVAE exhibits superior performance compared to other methods in both unconditional and class-conditional sampling, signifying a significant advancement in the field of time series generation. Because TimeVQVAE can learn a prior distribution, it can also perform anomaly detection, similar to DAGMM and GANF. However, unlike DAGMM and GANF, TimeVQVAE does not impose any bias on the shape of the prior distribution and learns the prior via masked generative modeling with a bidirectional transformer model. By doing so, TimeVQVAE overcomes the challenge of prior learning and can produce a learned prior distribution that robustly estimates a target prior distribution. In this work, we harness the powerful prior learning of TimeVQVAE and tailor its usage for anomaly detection, while incorporating explainability through its inherent generative nature.

3.3 Existing Approaches for XAI for TSAD

In [50], they adopt diverse counterfactuals for explaining anomalies in time series, and argue that a diverse set of counterfactuals are desirable to get the right intuition for why a detected anomaly is indeed anomalous. They elaborate on the importance of visualization for TSAD. In [51], they survey XAI for TSAD, focusing on multivariate industrial time series. They mention LIME and SHAP - two classical methods for feature attribution that may help explain how covariates contribute to an anomaly. In their review, they summarize the pros and cons of methods in a table (Table 1 in [51]). Common for the methodologies is the use of visualizations to explain an anomaly. For a general overview the taxonomy, opportunities and limitation of XAI, we refer to the survey [52].

4 Method

Before diving into the details of our method, it is important to establish the space in which the prior distribution resides and introduce relevant notations. TimeVQVAE follows the two-stage training approach by VQ-VAE [47]. In the first stage, an encoder E, vector quantizer VQ, and decoder D are trained by minimizing a reconstruction loss. In the context of TimeVQVAE, a given time series x undergoes a STFT preprocessing step and is subsequently encoded and quantized into a discrete latent vector z_q, expressed as $z_q = VQ(E(\text{STFT}(x)))$ where $x \in \mathbb{R}^T$, $\text{STFT}(x) \in \mathbb{R}^{2 \times H \times T'}$, and $z_q \in \mathbb{R}^{D \times H \times W}$. T denotes the time series length, 2, H, and T' denotes real and imaginary channels, the frequency dimension (height), and the temporal length of $\text{STFT}(x)$, respectively, and D and W denote the latent dimension size and the latent temporal length (width), respectively. Next, the second stage utilizes MaskGIT [53] to learn the prior of the discrete latent space $p(z_q)$ via masked modeling with a bidirectional transformer model (*i.e.,* prior model), and the learned prior distribution is denoted as $p_\theta(z_q)$ where θ denotes parameters of the prior model. In the following, we use the notation of tokens $s \in \mathbb{R}^{H \times W}$ instead of $z_q \in \mathbb{R}^{D \times H \times W}$ for brevity, where a token s refers to the codebook index of z_q, where $s \in s$ and $z_q \in z_q$ [53]. Using the notation, the target prior distribution is expressed as $p(s)$ instead of $p(z_q)$.

The prior distribution $p(s)$ is learned by $p_\theta(s)$ via masked modeling. Masked modeling is often adopted for language modeling [21], self-supervised learning [54], and generative modeling [53], and its objective is to maximize $p_\theta(s|s_M)$ where s_M denotes a masked version of s and is defined as $s_M = s \odot m + [\texttt{MASK}] \odot (1 - m)$ where m consists of 0 and 1 and $[\texttt{MASK}]$ represents a mask token. In the context of natural language processing, s and s_M can be $[\texttt{apple}, \texttt{is}, \texttt{red}]$ and $[\texttt{apple}, \texttt{is}, [\texttt{MASK}]]$, respectively, and the prior model can calculate the probability of $\texttt{red}$ given the other words $\texttt{apple}$ and $\texttt{is}$. The same approach can be applied to time series. For instance, TimeVQVAE adopted this masked modeling approach for time series generation and achieved SOTA performance. We redirect the use of the learned prior model of TimeVQVAE to perform TSAD instead of generation. In the training of masked modeling, a uniform-random portion of s is uniform-randomly masked, therefore the masking can be flexible in terms of size and locations during inference. Then, we can have a sliding masking window with an arbitrary window size and mask s and measure $p_\theta(s|s_M)$ iteratively along the temporal dimension. Finally, our anomaly score for the masked region M' can be calculated as $a_{M'} = a \odot (1 - m)$ where $a = -\log p_\theta(s|s_M)$ and $a_{M'}$ is the predicted anomaly score for the masked region in s_M.

As $p_\theta(\boldsymbol{s}|\boldsymbol{s}_M)$ is a crucial component in our TSAD, we delve deeper into the inner workings of it: a process of assigning higher probabilities to likely states of time series and lower probabilities to less and unlikely states.

The anomaly scores are measured using $p_\theta(\boldsymbol{s}|\boldsymbol{s}_M)$ and each element of the probability can be represented in the softmax form as

$$p_\theta(\boldsymbol{s}_i|\boldsymbol{s}_M) = \frac{e^{(\boldsymbol{u}_i)_{k^*}}}{\sum_{k=1}^{K} e^{(\boldsymbol{u}_i)_k}}, \tag{1a}$$

$$\boldsymbol{u} = f_\theta(\boldsymbol{s}_M), \tag{1b}$$

where the subscript i denotes an arbitrary element within the spatial dimension, k and K denote a codebook index and codebook size, respectively, f_θ denotes the prior model, $\boldsymbol{u} \in \mathbb{R}^{H \times W \times K}$, $\boldsymbol{u}_i \in \mathbb{R}^K$, and $(\boldsymbol{u}_i)_{k^*}$ denotes a prediction of u for $\boldsymbol{s}_i$, where k^* denotes a codebook index of $\boldsymbol{s}_i$. In the second stage, the prior model is trained by maximizing $p_\theta(\boldsymbol{s}_i|\boldsymbol{s}_M)$, leading to assigning higher values for $\boldsymbol{u}_i$ indexed by k^* while assigning lower values for $\boldsymbol{u}_i$ not indexed by k^*. The former corresponds to producing higher probabilities for likely normal states and the latter corresponds to producing lower probabilities for less likely and unlikely states. As a result, during inference, when given a time series containing an anomaly, the prior model can produce a low probability for the anomaly by performing the masked prediction with the mask on the anomaly.

In addition, $p_\theta(\boldsymbol{s}|\boldsymbol{s}_M)$ indicates that it is a stochastic generative process. As illustrated in Fig. 1, when given a token set with an anomalous segment masked, the prior model can stochastically generate likely normal states of the tokens. This enables the sampling of likely normal states of time series (counterfactual examples).

4.1 Training

Our prior model is an architectural variant of TimeVQVAE, therefore the training process remains the same. TimeVQ-VAE adopts the two-stage training approach from VQ-VAE. Mathematically, the first stage is formulated as minimizing

$$\mathbb{E}_{\boldsymbol{x} \sim X}\|\boldsymbol{x} - D(VQ(E(\text{STFT}(\boldsymbol{x}))))\| \tag{2}$$

and the second stage is formulated as maximizing

$$\mathbb{E}_{\boldsymbol{x} \sim X, z_q \sim VQ(E(\text{STFT}(\boldsymbol{x})))} \left[p_\theta(\boldsymbol{s}|\boldsymbol{s}_M) \right], \tag{3}$$

where E and D are trained in the first stage and set to be untrainable in the second stage. Fig. 4 illustrates the overview of the first and second stages.
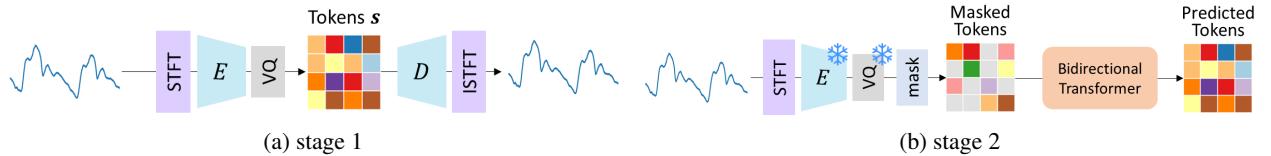

(a) stage 1   (b) stage 2

Figure 4: Overview of the first stage (stage 1) and the second stage (stage 2). The snowflakes indicate that the models are set to be untrainable, `mask` denotes random-masking, and the bidirectional transformer corresponds to the prior model.

4.2 Architecture

We propose the two architectural modifications to TimeVQVAE: 1) *LF-HF latent space merge*, 2) *dimensional semantics-preservative convolutional encoder*.

LF-HF Latent Space Merge   In TimeVQVAE, two sets of E, VQ, and D are used for LF and HF, respectively, to allow high compression on the LF latent space, leading to enhanced sampling performance. With the small LF latent space, the prior modeling $p_\theta(\boldsymbol{s}|\boldsymbol{s}_M)$ becomes easier, so as the sampling process $p_\theta(\boldsymbol{s}_{M,0:T_s}) = \prod_{t_s=0}^{T_s-1} p(\boldsymbol{s}_{M,t_s+1}|\boldsymbol{s})p_\theta(\boldsymbol{s}|\boldsymbol{s}_{M,t_s})$, in which the recursive product represents the iterative decoding proposed by [53], $\boldsymbol{s}_{M,0}$ is equivalent to a complete set of mask tokens `[MASK]`, and $\boldsymbol{s}_{M,T_s} = \boldsymbol{s}_{T_s}$ is equivalent to a set of fully-sampled tokens. We, however, do not need to solve $p_\theta(\boldsymbol{s}_{M,0:T_s})$ for anomaly detection. In cases where a time series exhibits anomalies, it is common to find that only a specific segment of the series is affected. Consequently, a particular segment of $\boldsymbol{s}$ carry tokens representing anomalous states. Therefore, we only need to solve a partial problem of $p_\theta(\boldsymbol{s}_{M,0:T_s})$ as $p_\theta(\boldsymbol{s}_{M,t:T})$ where $t > 0$ to compute the anomaly score of a certain-sized masked region $\boldsymbol{a}_{M'}$. Consequently, $p_\theta(\boldsymbol{s}_{M,t:T})$ is an easier problem on its own and we

no longer need to pose the high compression on the LF latent space. This allows us to merge the LF and HF latent spaces into a unified latent space. This not only simplifies the architecture but also enables us to capture fine-grained anomalies since s or z_q has a smaller receptive field due to the smaller compression rate. As a result, we only need a single set of E, VQ, and D instead of two sets, as shown in Fig. 4.

Dimensional Semantics-preservative Convolutional Encoder   Our anomaly scores are computed in the discrete latent space using our learned prior using $\boldsymbol{a} = -\log p_\theta(\boldsymbol{s}|\boldsymbol{s}_M)$. Hence, it is crucial to preserve the semantics of both the temporal and frequency dimensions of $\text{STFT}(\boldsymbol{x}) \in \mathbb{R}^{2 \times H \times T'}$ so that the predicted anomaly scores are temporally aligned with $\boldsymbol{x}$ while preserving the distinct frequency bands. However, the current configuration of the residual blocks in the encoder of TimeVQVAE, which use a kernel size of (3x3), and the downsampling blocks, which use a kernel size of (3x4), causes wide-mixture of information across the temporal and frequency dimensions. This wide-mixture significantly loosens the semantics of both dimensions. To address this limitation, we introduce a dimensional semantics-preservative convolutional encoder. This encoder is similar to the one used in TimeVQVAE, but with a key modification: the kernel sizes. Specifically, we propose replacing the kernel sizes of the residual block and downsampling block with (1x3) and (1x4), respectively. By making this adjustment, we ensure that z_q and its corresponding tokens $\boldsymbol{s}$ remain independent along the frequency axis and preserve sharper temporal semantics. Importantly, the decoder should also have the frequency-independent kernel sizes (*i.e.,* (1x4)). Otherwise, the decoder learns to predict i-th frequency information utilizing j-th frequency information, where $i \neq j$, allowing the encoder not to strictly encode the i-th frequency information, leading to ineffective discrete latent space modeling with respect to different frequencies, thereby sub-optimal performance.

4.3 Anomaly Score Prediction Process

Our anomaly scores reside in the discrete latent space. Yet, because the proposed convolutional encoder preserves the semantics of temporal and frequency dimensions, we can easily map the anomaly scores in the discrete latent space to the time series data space. We first discuss the anomaly score prediction in the discrete latent space and then mapping the scores onto the data space. Pseudocode for the anomaly score prediction process is presented in Algorithm 1.

4.3.1 Anomaly Score Prediction in the Discrete Latent Space

To systematically measure the anomaly scores, we propose to compute $\boldsymbol{a} = -\log p_\theta(\boldsymbol{s}|\boldsymbol{s}_M)$ using a *sliding masking latent window* along the temporal dimension. The masking latent window refers to masking a certain-sized temporal segment across the frequency dimension in the discrete latent space, and the sliding refers to the repetition of the latent window-masking process along the temporal dimension. To be precise, for $\boldsymbol{s} \in \mathbb{R}^{H \times W}$ and the certain temporal step w, we compute a predicted anomaly score as

$$\boldsymbol{a}_w = \mathbb{E}_w[-\log p_\theta(\boldsymbol{s}_{:,w-\alpha:w+\alpha}|s_{M(:,w-\alpha:w+\alpha)})], \tag{4}$$

where $\boldsymbol{a}_w \in \mathbb{R}^H$, $-\log p_\theta(\boldsymbol{s}_{:,w-\alpha:w+\alpha}|s_{M(:,w-\alpha:w+\alpha)}) \in \mathbb{R}^{H \times 2\alpha}$, the subscript $[:, w - \alpha : w + \alpha]$ indicates indexing the specified temporal (width) range from $w - \alpha$ to $w + \alpha$ across the frequency dimension H, α is an positive integer, and $M(:, w - \alpha : w + \alpha)$ indicates masking the specified temporal segment across the frequency dimension.

We adopt $-\log p_\theta(\boldsymbol{s}_{:,w-\alpha:w+\alpha}|s_{M(:,w-\alpha:w+\alpha)})$ instead of $-\log p_\theta(\boldsymbol{s}_{:,w}|s_{M(:,w-\alpha:w+\alpha)})$ because the latent temporal step w shares the information with the neighboring steps due to the receptive field of the convolutional encoder and we experimentally found that the former results in more robust anomaly score prediction.

We repeat the process along the temporal dimension W using a sliding latent window for all w, and obtain $\tilde{\boldsymbol{a}} \in \mathbb{R}^{H \times W}$ which is aligned with $\text{STFT}(\boldsymbol{x}) \in \mathbb{R}^{H \times T'}$ in terms of the dimensional semantics. We can obtain multiple $\tilde{\boldsymbol{a}}$ with a set of different α-s to incorporate the predicted anomaly scores with different latent window sizes, as α analogically specifies a kernel size. Then the multiple $\tilde{\boldsymbol{a}}$ can be summed to combine the effects of different latent window sizes.

We emphasize the importance of flexible α allowed by the masked generative modeling. Many of the existing anomaly detection methods require a fixed window size which in turn limits the performance as the smaller window size allows to capture short-ranged anomalies such as a peak but misses long-ranged anomalies such as a trend shift and vice versa. Our method, on the other hand, offers a flexible window size via α, enabling the anomaly detection in various aspects with respect to a window size without any further training.

4.3.2 Mapping the Anomaly Scores in the Discrete Latent Space to the Data Space

The predicted anomaly scores in the discrete latent space $\tilde{\boldsymbol{a}}$ can be simply mapped to the data space using a simple nearest-neighbor interpolation technique. The resulting anomaly score is denoted by $\tilde{\boldsymbol{a}}_m \in \mathbb{R}^{H \times T}$. This interpolation involves expanding the dimension from $(H \times W)$ to $(H \times T)$, where T represents the length of $\boldsymbol{x}$.

Algorithm 1 Pseudocode of the anomaly score prediction process using the learned prior model

---

Define a set of different α-s $\{\alpha_0, \alpha_1, ...\}$
Define an entire sequence of time series $\boldsymbol{x}^* \in \mathbb{R}^{T^*}$
Define a period length P of $\boldsymbol{x}^*$
Define multiple $\boldsymbol{a}^* \in \mathbb{R}^{H \times T^*}$ for $\{\alpha_0, \alpha_1, ...\}$, initialized with zeros.

for $\alpha \in \{\alpha_0, \alpha_1, ...\}$ **do** $\triangleright$ This can be replaced with parallel computation via multi-processing
    for $t \in [0, 1, 2, ...]$ **do** $\triangleright$ stride can be applied for faster computation
        $\boldsymbol{x} \leftarrow \boldsymbol{x}^*_{t:t+T}$ $\triangleright \boldsymbol{x} \in \mathbb{R}^T, T = 2P, [:, t : t + T]$ indexes $\boldsymbol{x}^*$ from t to T
        $\boldsymbol{z}_q \leftarrow VQ(E(\text{STFT}(\boldsymbol{x})))$ $\triangleright \boldsymbol{z}_q \in \mathbb{R}^{D \times H \times W}$
        $\boldsymbol{s} \leftarrow$ the codebook indices of $\boldsymbol{z}_q$ $\triangleright \boldsymbol{s} \in \mathbb{R}^{H \times W}$
        for $w \in [0, 1, ..., W]$ **do**
            $\boldsymbol{a}_w \leftarrow \mathbb{E}_w \left[-\log p_\theta(\boldsymbol{s}_{:, w-\alpha:w+\alpha} | \boldsymbol{s}_{M(:, w-\alpha:w+\alpha)}) \right] \triangleright \boldsymbol{a}_w \in \mathbb{R}^H$ denotes the predicted anomaly score at w
            Store $\boldsymbol{a}_w$
        end for
        $\tilde{\boldsymbol{a}} \leftarrow$ a collection of $\boldsymbol{a}_w$ $\triangleright \tilde{\boldsymbol{a}} \in \mathbb{R}^{H \times W}$
        $\tilde{\boldsymbol{a}}_m \leftarrow$ mapping $\tilde{\boldsymbol{a}}$ to the data space with simple nearest-neighbor interpolation $\triangleright \tilde{\boldsymbol{a}}_m \in \mathbb{R}^{H \times T}$
        $\boldsymbol{a}^*_{:, t:t+T} = \boldsymbol{a}^*_{:, t:t+T} + \tilde{\boldsymbol{a}}_m$ $\triangleright [:, t : t + T]$ indexes $\boldsymbol{a}^*$ from t to T across the frequency dimension
    end for
end for
$\boldsymbol{a}^*_s \leftarrow$ a summation of multiple $\boldsymbol{a}^*$-s obtained with $\{\alpha_0, \alpha_1, ...\}$ $\triangleright \boldsymbol{a}^*_s \in \mathbb{R}^{H \times T^*}$
$\bar{\boldsymbol{a}}^*_s \leftarrow \mathbb{E}_h [\boldsymbol{a}^*_s]$ $\triangleright \bar{\boldsymbol{a}}^*_s \in \mathbb{R}^{T^*}$
$\bar{\bar{\boldsymbol{a}}}^*_s \leftarrow$ moving-averaged $\bar{\boldsymbol{a}}^*_s$ with window size of T $\triangleright \bar{\bar{\boldsymbol{a}}}^*_s \in \mathbb{R}^{T^*}$
$\boldsymbol{a}_{\text{final}} \leftarrow (\bar{\boldsymbol{a}}^*_s + \bar{\bar{\boldsymbol{a}}}^*_s)/2$ $\triangleright$ final anomaly scores

---

4.4 Explainable Sampling

Explainable sampling refers to a process of masking anomalous segments in $\boldsymbol{s}$ and subsequently performing resampling with $p_\theta(\boldsymbol{s}|\boldsymbol{s}_M)$. To be more precise, we first compute the threshold as n-th quantile of $\boldsymbol{a}_{\text{final}}$ computed with a training dataset. n should be determined depending on the amount of present anomalies in your dataset, and the more anomalies present in the training dataset, the lower n should be. Typically, 0.9, 0.99, or 0.999 should be reasonable. Then, in the test dataset, we detect timesteps as anomalous if the anomaly scores $\boldsymbol{a}_{\text{final}}$ of the test dataset are above the threshold. We mask $\boldsymbol{s}$ for those anomalous timesteps across the frequency dimension, and perform $p_\theta(\boldsymbol{s}|\boldsymbol{s}_M)$ to predict the likely tokens for the masked regions. The resampling process is equivalent to the iterative decoding from TimeVQVAE. Fig. 2 presents an example of explainable sampling. It is important to emphasize that explainable sampling provides valuable insights and interpretations for users, enhancing confidence in the model's anomaly detection. Further details are described in Appendix A.2.

5 Experiments

5.1 Evaluation Metric

We follow the evaluation metric suggested by [30]. In the evaluation phase, the expected output of a method is a single predicted anomaly location. If this predicted location falls within a range of ±100 data points from the true location, it is considered accurate with a accuracy score of 1.0. Conversely, if the predicted location deviates outside this range, it is considered incorrect, resulting in an accuracy score of 0.0. To provide a comprehensive assessment, the accuracies across all 250 datasets are averaged, yielding a single accuracy score.

5.2 Experimental Setup

All datasets from the UCR-TSA archive are used in the experiments. A window size is set to $2 \times$ a period length P, following [6], and each input window is z-normalized. For our encoder and decoder, those from TimeVQVAE are adopted and modified according to our architectural proposals, and the vector quantizer and prior model from TimeVQVAE are adopted for TimeVQVAE-AD with a minor parameter change. Further details on the parameter choices and implementations are available in A.

5.3 Results

We have reviewed the existing literature and gathered accuracy scores of various TSAD methods evaluated on the UCR-TSA archive. Furthermore, we conducted our own evaluations and obtained results for several additional competing methods. The compiled accuracy scores are presented in Table 1. In addition to the top-1 accuracy, we report the top-k accuracies is important given that many real-world time series signals can involve multiple plausible anomalies, therefore the top-k accuracy can provide a more comprehensive evaluation of the model's performance. The top-3 and top-5 accuracies are measured using a simple local maxima-finding algorithm based on simple comparison of neighboring values such as the function named `find_peaks` from the SciPy library [55]. Note that the reported top-1 accuracy of COCA in its original paper is documented as 0.661. However, when we conducted experiments using the authors' GitHub codes without any modifications, our results yielded a significantly lower accuracy of 0.236. To ensure the validity of our finding, we compared our result with the visualization presented by the authors. Notably, the authors presented a visualization on a single specific dataset from the UCR-TSA archive. Our outcome closely aligned with their visualization.

Table 1 shows the outstanding superiority of TSAD accuracy achieved by TimeVQVAE-AD. To ensure the credibility of our results, we have made the visualizations and CSV files of the predicted anomaly scores on the UCR-TSA archive openly accessible in our GitHub repository. Fig. 5 displays visualizations of predicted anomaly scores by TimeVQVAE-AD on the various datasets.

Table 1: Evaluation of the various TSAD methods on the UCR-AD archive. *Acc.* denotes the accuracy metric. The evaluation of the model's performance is more comprehensive when considering the top-k accuracies.

	Method Mechanism	Method	Published Year	Top-1 Acc.	Top-3 Acc.	Top-5 Acc.
non-DL	one-class classification	OC-SVM [33]	1999	0.088 [20]		
non-DL	isolation forest	IF [3]	2008	0.376 [20]		
non-DL	isolation forest	RCF [4]	2016	0.387 [20]		
non-DL	matrix profile	Matrix Profile SCRIMP [34]	2016	0.416 [6]		
non-DL	density estimation	MDI [13]	2018	0.47 [17]		
non-DL	matrix profile	Matrix Profile STUMPY [35]	2019	0.512	0.684	0.744
non-DL	discord discovery	MERLIN [5]	2020	0.424 [6]		
non-DL	discord discovery	MERLIN++ [6]	2023	0.424 [6]		
DL	reconstruction	AE		0.236 [7]		
DL	reconstruction	Convolutional AE		0.352	0.412	0.448
DL	reconstruction	LSTM-ED [36]	2016	0.51 [20]		
DL	variational reconstruction	LSTM-VAE [8]	2018	0.198 [7]		
DL	forecasting	Telemanom [11]	2018	0.468 [6]		
DL	one-class classification	Deep SVDD [2]	2018	0.076 [20]		
DL	density estimation	DAGMM [14]	2018	0.061 [20]		
DL	spectral saliency map	SR-CNN [56]	2019	0.30 [20]		
DL	reconstruction, adversarial training	USAD [9]	2020	0.276 [7]		
DL	contrastive learning	CPC-AD [40]	2021	0.064 [20]		
DL	contrastive learning, one-class classification	TS-TCC-AD [57, 58]	2021	0.006 [20]		
DL	reconstruction	TranAD [10]	2022	0.19 [17]		
DL	density estimation	GANF [15]	2022	0.24 [17]		
DL	non-contrastive learning	COCA [20]	2023	0.236	0.328	0.408
DL	density estimation	TimeVQVAE-AD (ours)	2023	**0.708**	**0.776**	**0.824**

6 Discussion

How is TimeVQVAE-AD Significantly Better than the Others? We employ the strongest prior model for time series generation from TimeVQVAE. This prior model can generate synthetic time series with high fidelity, indicating its effectiveness in approximating a target prior. By utilizing this strong prior model in measuring anomaly scores, we are able to achieve more accurate anomaly assessments. Importantly, our anomaly scores do not rely on the error
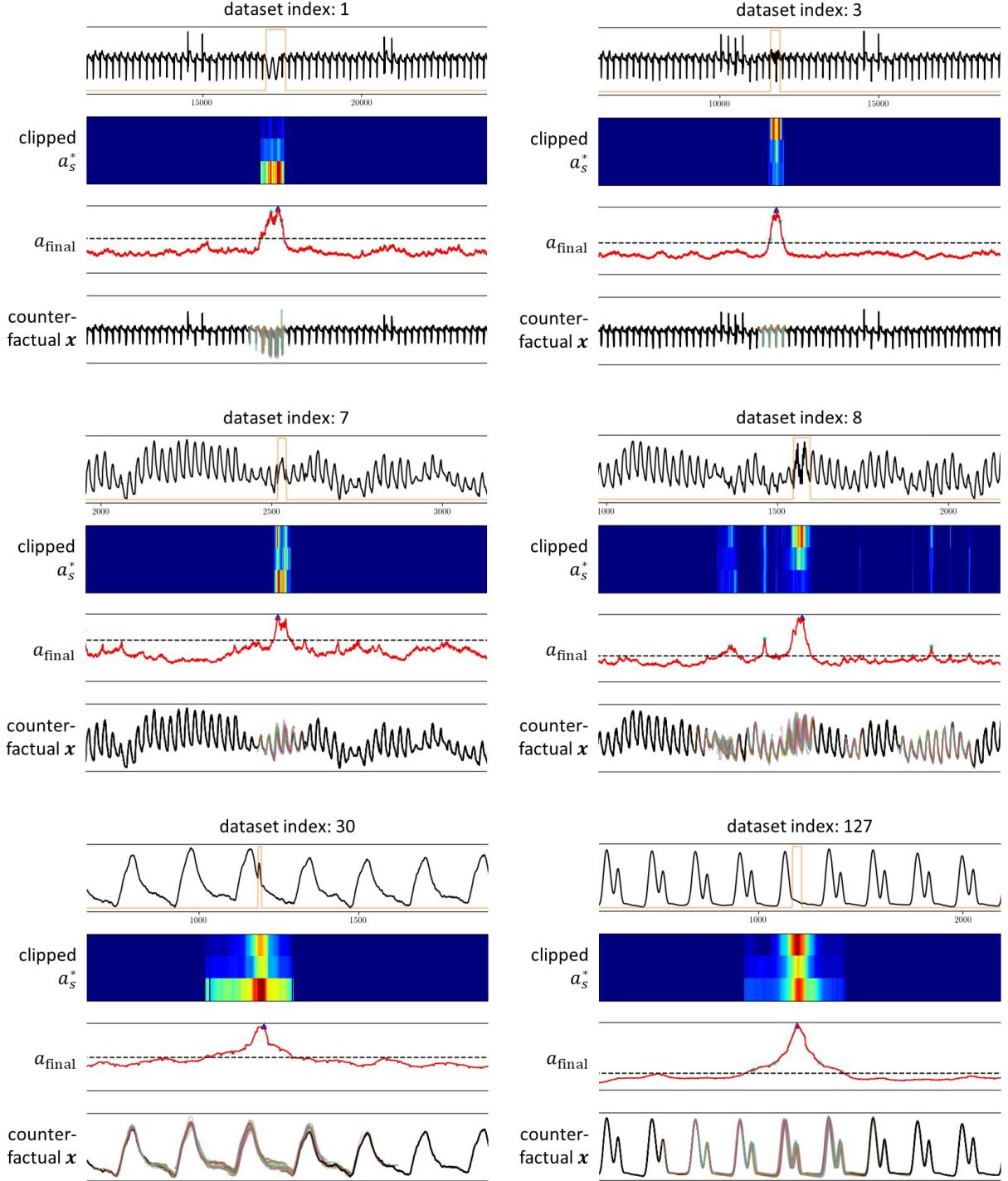
Figure 5: Examples of the visual results showcasing predicted anomaly scores by TimeVQVAE-AD on different datasets from the UCR-TSA archive. The first row shows a test time series (black) with the corresponding labels (orange), the second row presents the anomaly scores with the height representing the frequency dimension a_s^* clipped by the threshold, the third row presents the final anomaly scores a_{final}, and the last row presents the likely normal states achieved through explainable sampling. It should be noted that the scaling of the likely normal states depends on the scaling factors of the corresponding original time series segment. For instance, if an original time series segment has a low mean value, the likely normal states also have a low mean value. The same principle applies to the scaling of the variance.

between a target time series and predicted time series, unlike the majority of existing TSAD methods that do rely on such error. Typically, those methods are limited to detecting anomalies with abnormally-high amplitudes. In contrast, our anomaly scores are measured using the prior model, which captures the semantic relationships between different time series segments represented by distinct tokens. This approach enables us to effectively measure anomaly scores and detect anomalies beyond just amplitude abnormalities. Moreover, TimeVQVAE-AD enhances its detection accuracy by measuring anomaly scores in various aspects, especially across different frequency bands and various latent window sizes. Regarding explainability, our proposed encoder retains the semantics of both temporal and frequency dimensions, allowing the resulting anomaly scores to carry both dimensions. Notably, the inclusion of the frequency dimension is novel in TSAD and offers valuable diagnostic insights. Additionally, the generative nature of the prior model enables resampling anomalous segments and obtain corresponding likely normal states. These two properties significantly enhance the confidence in the detected anomalies.

Anomaly Scores Should Measure Magnitude of Anomalism   Anomalism refers to the quality of being anomalous. Anomalies can exhibit varying degrees of magnitude, ranging from slightly anomalous to moderately anomalous and completely anomalous. A robust TSAD method should be capable of effectively capturing this spectrum. In Fig. 6, we present an example showcasing a time series with a definite anomaly (labeled in orange) and subtle anomaly (yellow) with corresponding predicted anomaly scores by TimeVQVAE-AD, Matrix Profile STUMPY, and Convolutional AE. TimeVQVAE-AD and STUMPY successfully assign the highest scores to the labeled segment and sufficiently-high scores to the subtle anomaly, while TimeVQVAE-AD achieves a more precise capture of the anomaly. However, Convolutional AE fails. This is because Convolutional AE is regularized due to its bottleneck during training and can effectively reconstruct a time series from a test dataset that closely resembles those in the training dataset. Consequently, the reconstruction error remains low for the subtle anomaly due to the effective reconstruction. Regarding TimeVQVAE-AD, it is important to acknowledge that the prior model learns the prior distribution, which in turn allows for an effective capture of magnitude of anomalism.
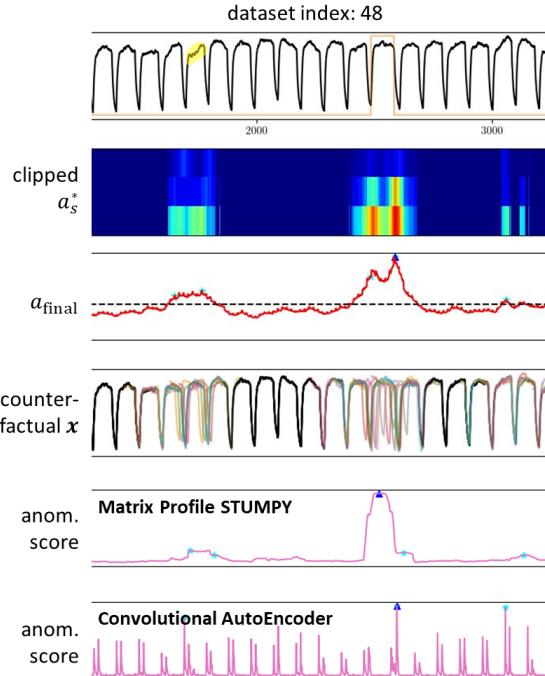

Figure 6: Example of a time series dataset with a definite anomaly (labeled in orange) and subtle anomaly (colored in yellow) with corresponding predicted anomaly scores by TimeVQVAE-AD, Matrix Profile STUMPY, and Convolutional AE. The labeled anomaly exhibits an abnormality in its period length, which is slightly longer than what is considered normal. It should be noted that TimeVQVAE-AD assigns the highest scores to the labeled segment and sufficiently-high scores to the subtle anomaly. Moreover, TimeVQVAE-AD demonstrates a more precise capture of the labeled anomaly in comparison to STUMPY. Given that the anomaly is characterized by an abnormal period, it is essential that both ends of the anomaly are accurately identified as the most anomalous timesteps, which is effectively accomplished by TimeVQVAE-AD.

Flexible Window Size Enabled by α Unlike the existing deep TSAD methods, TimeVQVAE-AD allows a flexible window size, enabled by α. It defines the temporal window range in the discrete latent space as $[w - \alpha, w + \alpha]$ which is analogically equivalent to a kernel size of a convolutional layer. The captured anomaly aspects vary depending on the latent window size. A narrow window size is effective in detecting short-range anomalies, while a wide window size is effective in capturing long-range anomalies. Fig. 7 presents examples of the predicted anomaly scores $\boldsymbol{a}^*$ with different latent window sizes. In practice, the latent window size is set as $r_w \times W$ where r_w denotes a latent window size rate and $r_w \in (0, 1)$, and we use r_w of $\{0.1, 0.3, 0.5\}$ in our experiments to cover from a narrow to wide window. We must emphasize that the incorporation of the effects of different window sizes enhances our detection accuracy.
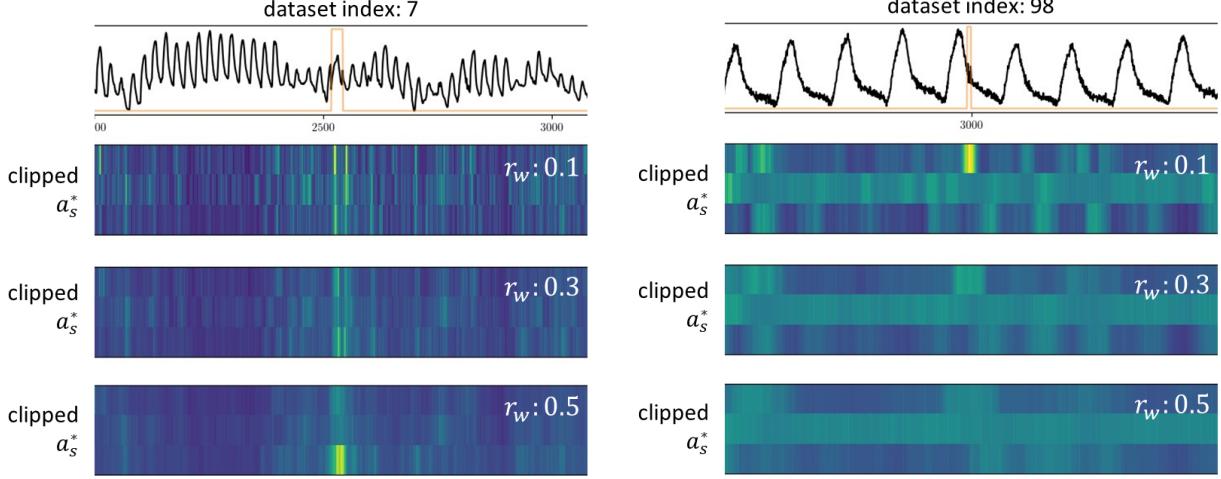

Figure 7: Examples of the predicted anomaly scores with different latent window sizes determined by the latent window size rate r_w. The first example involves a long-range anomaly and the second one involves a short-range anomaly. It is noticeable that the long-range anomaly is better captured with a wider latent window size rate, while the short-range anomaly is better detected with a narrower latent window size rate.

Where Do Matrix Profile and Discord Discovery Methods Fail?   Table 1 demonstrates that, in general, the matrix profile and discord discovery methods outperform the existing deep learning methods in terms of detection accuracy. Yet, the best accuracy achieved by those methods is 0.512, achieved by Matrix Profile STUMPY. This raises the question of where those methods fail. Both methods primarily rely on measuring anomaly scores by evaluating the distances between different subsequences of time series. Consequently, they face a similar challenge to the reconstruction and forecasting-based TSAD methods – that is, they can typically capture anomalies with abnormally-high amplitudes only since they measure anomaly scores based on the error between a target time series and its predicted counterpart. Fig. 8 presents an example of time series datasets where Matrix Profile STUMPY fails due to its limitation, whereas TimeVQVAE-AD succeeds.
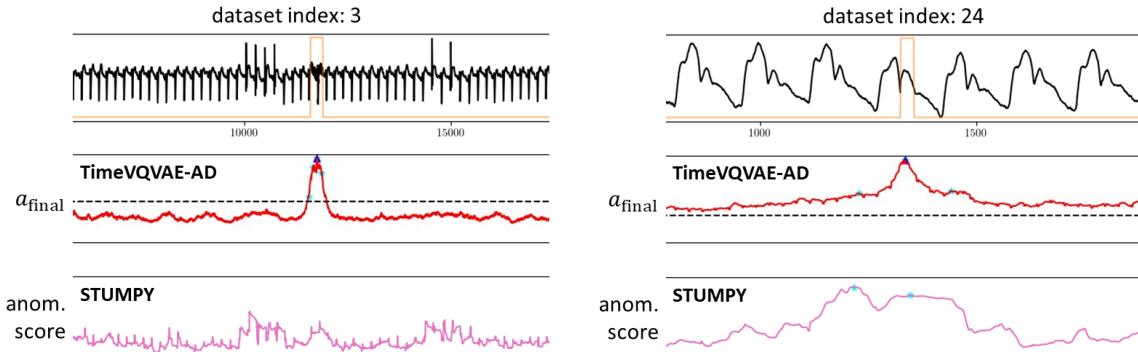

Figure 8: Example of time series datasets where Matrix Profile STUMPY fails. It highlights the inherent limitations associated with matrix profile and discord discovery methods for TSAD.

Presence of Hidden Anomalies   We have identified that there exist hidden anomalies, in other words, unlabeled anomalies. Perhaps that is natural to observe, referring to the argument from the author of the UCR-TSA archive, "In fact, perfect ground truth labels are impossible for anomaly detection" [59]. Fig. 9 presents examples of the datasets with hidden anomalies, detected by TimeVQVAE-AD. The predicted anomaly scores on such hidden anomalies can obscure the scores on labeled anomalies, leading to the accuracy of 0. However, it is worth considering that the model may have successfully detected the labeled anomalies with the second or third highest scores. Therefore, the evaluation of TSAD should also account for this capability. To accommodate for such scenarios, we suggest the adoption of top-k accuracies.
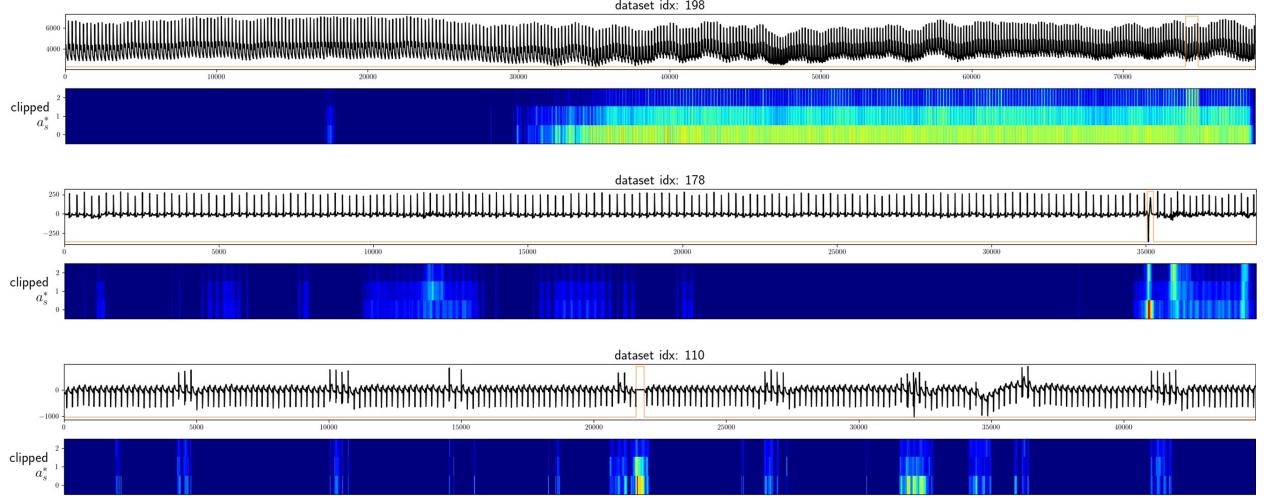

Figure 9: Examples of the datasets with hidden anomalies (*i.e.,* unlabeled anomalies), detected by TimeVQVAE-AD. In the first dataset, we notice an increase in density towards the right side. In the second dataset, upon closer examination, we can observe the anomalous patterns on the right side. The third dataset reveals hidden anomalies in the form of unusual amplitudes occurring around 32000.

Dilemma of Aggregation of Anomaly Scores   The predicted anomaly scores of TimeVQVAE-AD capture two novel aspects: 1) different frequency bands and 2) different temporal segment sizes determined by α. Therefore, we have n number of anomaly scores $\boldsymbol{a}_s^*$ before the aggregation process, where n is a number of α-s. These anomaly scores, however, must be aggregated to yield a single value for each timestep, as required for the computation of the evaluation metrics. Unfortunately, this aggregation process can sometimes compromise the metrics scores. For example, when an anomaly has a short duration, it is effectively captured by the anomaly score associated with a small α, but it is less likely to be detected by the score linked to a larger α. As a result, after the aggregation, the anomaly score with a small α may become too attenuated to emerge as the highest anomaly score. Another failure scenario arises when an anomaly score exhibits a distinctive peak in one frequency band but remains low in the other frequency bands. In such cases, the anomaly might fail to be identified as the top anomaly after the aggregation process for the same reason as above. We note, however, that a practitioner may choose to look at anomaly scores in certain frequency bands rather than at aggregated level depending on the application at hand. While the aggregated score may fail to detect an anomaly according to our proposed scheme, a practitioner has the opportunity to look at different frequency bands for further analysis. Examples of the first and second scenarios are shown in Fig. 10.

Dilemma of Long-range Medium-amplitude Anomaly Scores   In physics, there is a physical quantity called *impulse* J, defined as $J = F \times \Delta t$ where F and Δt denote force and duration. That indicates that J is large even if F is moderate as long as Δt is sufficiently large. The concept of J can be applied to anomaly scores. For instance, if an anomalous segment has a long range with medium-amplitude scores, it should still be regarded as highly anomalous due to its extensive duration. However, in the evaluation protocol, a method is typically required to return a single most-likely anomalous timestep, often obtained through an argmax operation. This approach does not adequately account for the significance of long-range medium-amplitude anomaly scores. To mitigate the limitation, we introduce the concept of Δt in calculating $\boldsymbol{a}_{\text{final}}$. As described in Algorithm 1, we propose applying moving-average with a window size of T to $\bar{\boldsymbol{a}}_s^*$, resulting in $\bar{\bar{\boldsymbol{a}}}_s^*$. Conceptually, $\bar{\bar{\boldsymbol{a}}}_s^*$ corresponds to $F \times \Delta t$, while $\bar{\boldsymbol{a}}_s^*$ corresponds to F. However, $\bar{\bar{\boldsymbol{a}}}_s^*$ loses locality, so we combine $\bar{\boldsymbol{a}}_s^*$ and $\bar{\bar{\boldsymbol{a}}}_s^*$ to obtain $\boldsymbol{a}_{\text{final}}$. Fig. 11 illustrates an example that highlights the dilemma.
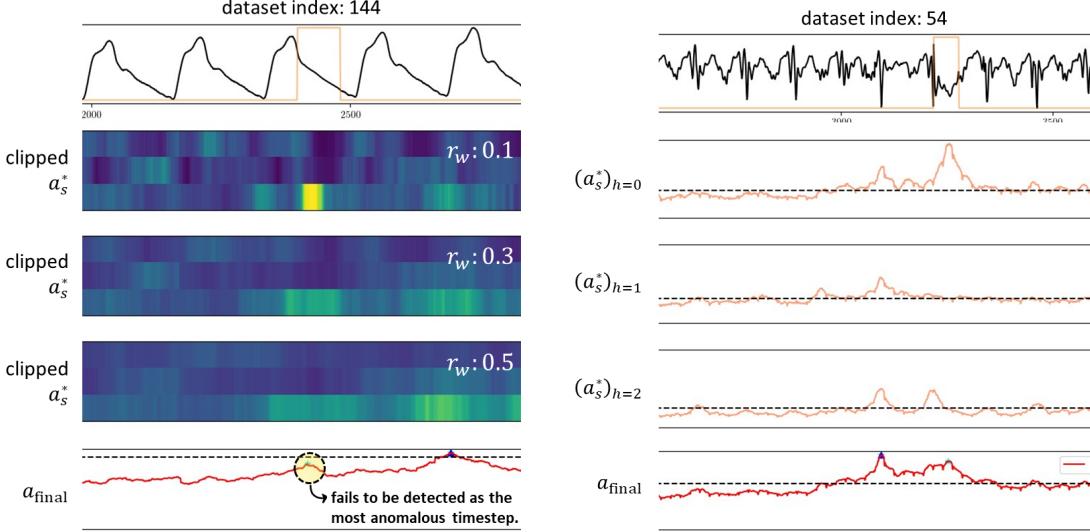
Figure 10: Examples of the dilemma regarding the aggregation of predicted anomaly scores. h denotes a frequency band, and $h = 0$, $h = 1$, and $h = 2$ denote the lowest, middle, and highest frequency bands, respectively. The left figure illustrates a scenario in which the anomaly is effectively captured using a small latent window size, but fails to be detected with larger latent window sizes, ultimately resulting in a detection failure. The right figure illustrates a scenario in which the anomaly score displays a prominent peak in one frequency band while remaining low in other frequency bands, leading to a detection failure despite the anomaly's evident presence.
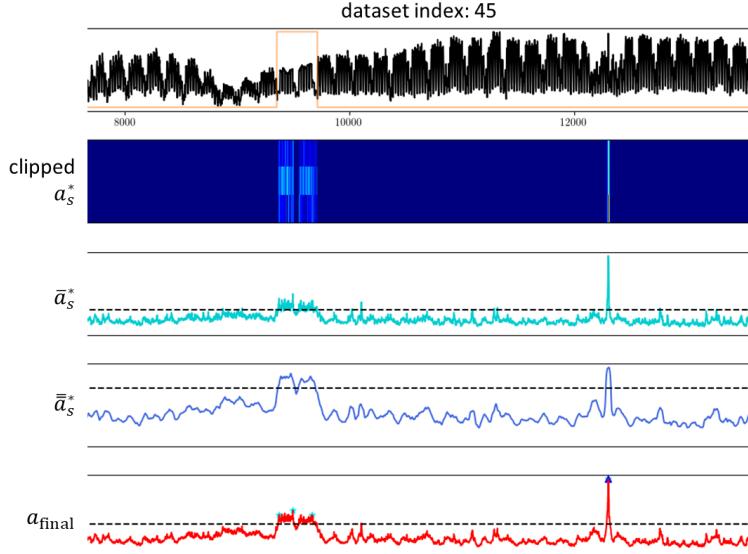

Figure 11: Example of the dilemma regarding long-range medium-amplitude anomaly scores. The anomaly segment at around 9600 has long-range medium-amplitude anomaly scores, while there are two other short-range high-amplitude anomaly segments. The dilemma is that the significance of long-range medium-amplitude anomaly scores is not properly accounted for. To mitigate the dilemma, moving-average is applied to $\bar{a}_s^*$, producing $\bar{\bar{a}}_s^*$ and it is noticeable that $\bar{\bar{a}}_s^*$ better captures the significance of long-range medium-amplitude anomaly scores.

7   Conclusion

In this paper, we touched upon significant flaws in the currently-popular benchmark datasets and evaluation protocol and limitations of the existing TSAD methods in terms of detection accuracy and explainability. Then we proposed TimeVQVAE-AD, a novel approach to TSAD that leverages a strong prior model from the SOTA time series generation method, TimeVQVAE, to learn the distributions of likely, less likely, and unlikely normal states of time series with

respect to different frequency bands. This distribution modeling enables the detection of a broad spectrum of anomalies across different frequency bands and temporal scales, which in turn allows for the detection of subtle and complex anomalies that may be overlooked by traditional TSAD methods. Moreover, our method's focus on explainability through the generation of counterfactual examples provides valuable insights into the detected anomalies. Our experiments were conducted on the UCR Time Series Anomaly archive for a fair evaluation. The experimental results showed that TimeVQVAE-AD achieves ground-breaking TSAD accuracy and an exceptional level of explainability through counterfactual samples.

Our work opens several paths for future research. Firstly, improving TimeVQVAE-AD involves overcoming the challenges in anomaly score aggregation and detecting long-range medium-amplitude anomalies. Secondly, applying our method to multivariate time series could greatly expand its use and impact. Additionally, exploring its potential across different sectors such as healthcare and finance could not only validate its effectiveness in real-world scenarios but also lead to new insights and improvements in the methodology. Overall, we believe TimeVQVAE-AD marks a major progress in the field of TSAD.

Acknowledgments

Ethical Statement

No conflicts of interest were present during the research process.

References

[1] Bernhard Schölkopf, Robert C Williamson, Alex Smola, John Shawe-Taylor, and John Platt. Support vector method for novelty detection. In S. Solla, T. Leen, and K. Müller, editors, *Advances in Neural Information Processing Systems*, volume 12. MIT Press, 1999.

[2] Lukas Ruff, Robert Vandermeulen, Nico Goernitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, Emmanuel Müller, and Marius Kloft. Deep one-class classification. In *International conference on machine learning*, pages 4393–4402. PMLR, 2018.

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

[4] Sudipto Guha, Nina Mishra, Gourav Roy, and Okke Schrijvers. Robust random cut forest based anomaly detection on streams. In *International conference on machine learning*, pages 2712–2721. PMLR, 2016.

[5] Takaaki Nakamura, Makoto Imamura, Ryan Mercer, and Eamonn Keogh. Merlin: Parameter-free discovery of arbitrary length anomalies in massive time series archives. In *2020 IEEE international conference on data mining (ICDM)*, pages 1190–1195. IEEE, 2020.

[6] Takaaki Nakamura, Ryan Mercer, Makoto Imamura, and Eamonn Keogh. Merlin++: parameter-free discovery of time series anomalies. *Data Mining and Knowledge Discovery*, pages 1–40, 2023.

[7] Julien Audibert, Sébastien Marti, Frédéric Guyard, and Maria A Zuluaga. From univariate to multivariate time series anomaly detection with non-local information. In *Advanced Analytics and Learning on Temporal Data: 6th ECML PKDD Workshop, AALTD 2021, Bilbao, Spain, September 13, 2021, Revised Selected Papers 6*, pages 186–194. Springer, 2021.

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

[9] 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*, pages 3395–3404, 2020.

[10] 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.

[11] 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*, pages 387–395, 2018.

[12] Axel Harstad and William Kvaale. Spatio-temporal graph attention network for anomaly detection in the telco domain. Master's thesis, NTNU, 2021.

[13] Björn Barz, Erik Rodner, Yanira Guanche Garcia, and Joachim Denzler. Detecting regions of maximal divergence for spatio-temporal anomaly detection. *IEEE transactions on pattern analysis and machine intelligence*, 41(5):1088–1101, 2018.

[14] 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.

[15] Enyan Dai and Jie Chen. Graph-augmented normalizing flows for anomaly detection of multiple time series. *arXiv preprint arXiv:2202.07857*, 2022.

[16] Renjie Wu and Eamonn Keogh. Current time series anomaly detection benchmarks are flawed and are creating the illusion of progress. *IEEE Transactions on Knowledge and Data Engineering*, 2021.

[17] Ferdinand Rewicki, Joachim Denzler, and Julia Niebling. Is it worth it? comparing six deep and classical methods for unsupervised anomaly detection in time series. *Applied Sciences*, 2023.

[18] Siwon Kim, Kukjin Choi, Hyun-Soo Choi, Byunghan Lee, and Sungroh Yoon. Towards a rigorous evaluation of time-series anomaly detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 7194–7201, 2022.

[19] Haowen Xu, Wenxiao Chen, Nengwen Zhao, Zeyan Li, Jiahao Bu, Zhihan Li, Ying Liu, Youjian Zhao, Dan Pei, Yang Feng, et al. Unsupervised anomaly detection via variational auto-encoder for seasonal kpis in web applications. In *Proceedings of the 2018 world wide web conference*, pages 187–196, 2018.

[20] Xudong Mou, Rui Wang, Tiejun Wang, Jie Sun, Bo Li, Tianyu Wo, and Xudong Liu. Deep autoencoding one-class time series anomaly detection. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2023.

[21] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[22] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pages 8821–8831. PMLR, 2021.

[23] Daesoo Lee, Sara Malacarne, and Erlend Aune. Vector quantized time series generation with a bidirectional prior model. In *International Conference on Artificial Intelligence and Statistics*, pages 7665–7693. PMLR, 2023.

[24] Yihao Ang, Qiang Huang, Yifan Bao, Anthony KH Tung, and Zhiyong Huang. Tsgbench: Time series generation benchmark. *arXiv preprint arXiv:2309.03755*, 2023.

[25] Riccardo Guidotti. Counterfactual explanations and how to find them: literature review and benchmarking. *Data Mining and Knowledge Discovery*, pages 1–55, 2022.

[26] Sahil Verma, Varich Boonsanong, Minh Hoang, Keegan E Hines, John P Dickerson, and Chirag Shah. Counterfactual explanations and algorithmic recourses for machine learning: A review. *arXiv preprint arXiv:2010.10596*, 2020.

[27] N Laptev, S Amizadeh, and Y Billawala. S5-a labeled anomaly detection dataset, version 1.0 (16m), 2015.

[28] Subutai Ahmad, Alexander Lavin, Scott Purdy, and Zuha Agha. Unsupervised real-time anomaly detection for streaming data. *Neurocomputing*, 262:134–147, 2017.

[29] 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*, pages 2828–2837, 2019.

[30] E. Keogh, T. Dutta Roy, U. Naik, and A. Agrawal. Multi-dataset time series anomaly detection competition, sigkdd 2021. https://compete.hexagon-ml.com/practice/competition/39/#evaluation, 2021.

[31] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[32] Sergio Naval Marimont and Giacomo Tarroni. Anomaly detection through latent space restoration using vector quantized variational autoencoders. In *2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI)*, pages 1764–1767. IEEE, 2021.

[33] 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.

[34] 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)*, pages 1317–1322. Ieee, 2016.

[35] Sean M. Law. STUMPY: A Powerful and Scalable Python Library for Time Series Data Mining. *The Journal of Open Source Software*, 4(39):1504, 2019.

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

[37] Dan Li, Dacheng Chen, Baihong Jin, Lei Shi, Jonathan Goh, and See-Kiong Ng. Mad-gan: Multivariate anomaly detection for time series data with generative adversarial networks. In *Artificial Neural Networks and Machine Learning–ICANN 2019: Text and Time Series: 28th International Conference on Artificial Neural Networks, Munich, Germany, September 17–19, 2019, Proceedings, Part IV*, pages 703–716. Springer, 2019.

[38] Alexander Geiger, Dongyu Liu, Sarah Alnegheimish, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. Tadgan: Time series anomaly detection using generative adversarial networks. In *2020 IEEE International Conference on Big Data (Big Data)*, pages 33–43. IEEE, 2020.

[39] Shyam Sundar Saravanan, Tie Luo, and Mao Van Ngo. Tsi-gan: Unsupervised time series anomaly detection using convolutional cycle-consistent generative adversarial networks. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 39–54. Springer, 2023.

[40] Puck de Haan and Sindy Löwe. Contrastive predictive coding for anomaly detection. *arXiv preprint arXiv:2107.07820*, 2021.

[41] Ah-Hyung Shin, Seong Tae Kim, and Gyeong-Moon Park. Time series anomaly detection using transformer-based gan with two-step masking. *IEEE Access*, 2023.

[42] Yiwei Fu and Feng Xue. Mad: Self-supervised masked anomaly detection task for multivariate time series. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2022.

[43] Aadyot Bhatnagar, Paul Kassianik, Chenghao Liu, Tian Lan, Wenzhuo Yang, Rowan Cassius, Doyen Sahoo, Devansh Arpit, Sri Subramanian, Gerald Woo, Amrita Saha, Arun Kumar Jagota, Gokulakrishnan Gopalakrishnan, Manpreet Singh, K C Krithika, Sukumar Maddineni, Daeki Cho, Bo Zong, Yingbo Zhou, Caiming Xiong, Silvio Savarese, Steven Hoi, and Huan Wang. Merlion: A machine learning library for time series. 2021.

[44] Thilo Spinner, Jonas Körner, Jochen Görtler, and Oliver Deussen. Towards an interpretable latent space: an intuitive comparison of autoencoders with variational autoencoders. In *IEEE VIS 2018*, 2018.

[45] Polina Kirichenko, Pavel Izmailov, and Andrew G Wilson. Why normalizing flows fail to detect out-of-distribution data. *Advances in neural information processing systems*, 33:20578–20589, 2020.

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

[47] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

[48] Jiahui Yu, Xin Li, Jing Yu Koh, Han Zhang, Ruoming Pang, James Qin, Alexander Ku, Yuanzhong Xu, Jason Baldridge, and Yonghui Wu. Vector-quantized image modeling with improved vqgan. *arXiv preprint arXiv:2110.04627*, 2021.

[49] Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, et al. Scaling autoregressive models for content-rich text-to-image generation. *arXiv preprint arXiv:2206.10789*, 2022.

[50] Deborah Sulem, Michele Donini, Muhammad Bilal Zafar, Francois-Xavier Aubet, Jan Gasthaus, Tim Januschowski, Sanjiv Das, Krishnaram Kenthapadi, and Cedric Archambeau. Diverse counterfactual explanations for anomaly detection in time series. *arXiv preprint arXiv:2203.11103*, 2022.

[51] Sarthak Manas Tripathy, Ashish Chouhan, Marcel Dix, Arzam Kotriwala, Benjamin Klöpper, and Ajinkya Prabhune. Explaining anomalies in industrial multivariate time-series data with the help of explainable ai. In *2022 IEEE International Conference on Big Data and Smart Computing (BigComp)*, pages 226–233. IEEE, 2022.

[52] Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador García, Sergio Gil-López, Daniel Molina, Richard Benjamins, et al. Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information fusion*, 58:82–115, 2020.

[53] Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T Freeman. Maskgit: Masked generative image transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11315–11325, 2022.

[54] Alexei Baevski, Wei-Ning Hsu, Qiantong Xu, Arun Babu, Jiatao Gu, and Michael Auli. Data2vec: A general framework for self-supervised learning in speech, vision and language. In *International Conference on Machine Learning*, pages 1298–1312. PMLR, 2022.

[55] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.

[56] Hansheng Ren, Bixiong Xu, Yujing Wang, Chao Yi, Congrui Huang, Xiaoyu Kou, Tony Xing, Mao Yang, Jie Tong, and Qi Zhang. Time-series anomaly detection service at microsoft. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 3009–3017, 2019.

[57] Emadeldeen Eldele, Mohamed Ragab, Zhenghua Chen, Min Wu, Chee Keong Kwoh, Xiaoli Li, and Cuntai Guan. Time-series representation learning via temporal and contextual contrasting. *arXiv preprint arXiv:2106.14112*, 2021.

[58] Kihyuk Sohn, Chun-Liang Li, Jinsung Yoon, Minho Jin, and Tomas Pfister. Learning and evaluating representations for deep one-class classification. *arXiv preprint arXiv:2011.02578*, 2020.

[59] Renjie Wu and Eamonn Keogh. Irrational Exuberance.pdf, supplemental material of the UCR Anomaly Archive. `https://www.cs.ucr.edu/~eamonn/time_series_data_2018/UCR_TimeSeriesAnomalyDatasets2021.zip`, 2021.

[60] Daesoo Lee, Erlend Aune, and Sara Malacarne. Masked generative modeling with enhanced sampling scheme. *arXiv preprint arXiv:2309.07945*, 2023.

[61] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

A Implementation Details

A.1 Dataset

All datasets from the UCR-TSA archive are utilized in our experiments. The window size is set to $2 \times P$ in accordance with [6], and each input window underwent z-normalization. To ensure comprehensive coverage of time series patterns, we set the window stride to 1, avoiding the possibility of missing any specific pattern. Regarding the parameter P, we manually measured its value for all 250 datasets and made the measurements available in our GitHub repository. While [6] employed an autocorrelation function to determine P, we observed that the periods of several datasets presented challenges when using autocorrelation. Consequently, in an effort to establish a standardized evaluation protocol, we conducted meticulous manual measurements of P for all datasets.

A.2 TimeVQVAE-AD

STFT, ISTFT STFT and ISTFT are implemented with `torch.stft` and `torch.istft`, respectively. Their main parameter – `n_fft`, size of Fourier transform – determines the frequency dimension size. The frequency dimension of $\text{STFT}(\boldsymbol{x})$ is determined by $\lfloor \texttt{n\_fft}/2 \rfloor + 1$. Consequently, a higher value of `n_fft` leads to anomaly scores with a larger frequency dimension, enabling the detection of anomalies with finer frequency resolution. Through our experiments, we found that an `n_fft` value of 4 was sufficient to capture various types of anomalies present in the UCR-TSA archive.

Encoder The encoder used in TimeVQVAE-AD shares the same structure as that of TimeVQVAE, with the exception of kernel size modifications and the utilization of a smaller compression rate, as detailed in Sect. 4.2. The compression rate can be alternatively specified by *downsampled width* [23], which refers to the width of $\boldsymbol{z}_q$. A smaller downsampled width corresponds to a higher compression rate, and increasing the number of downsampling blocks in the encoder achieves a higher compression rate. In TimeVQVAE-AD, we employ a downsampled width of 32. The selection of the downsampled width is important. When opting for a smaller downsampled width, it results in lower resolution in the prediction of AD scores since an element of z_q encompasses a broader temporal segment. Conversely, opting for a larger downsampled width can potentially lead to a reduction in the accuracy of AD predictions, as the number of tokens to predict increases for the same temporal segment. Thus, we chose 32 as a good trade-off between the prediction resolution and AD prediction accuracy in our experiments.

Vector Quantizer The same vector quantizer from TimeVQVAE is used except for the use of a bigger codebook size (128) to better capture different patterns in time series data. Also, in stage 2, we train the code embeddings fresh without re-initialization with the learned codes in stage 1 because we have a large sum of training data.

Decoder The same decoder architecture from TimeVQVAE is used.

Prior Model The prior model from TimeVQVAE is used with a slight modification in the output layer. The output of the prior model in TimeVQVAE is computed by forming a covariance matrix between the output from the transformer model and the codebook embeddings. In our implementation, we simply use a shallow network to project the transformer's output to logits over the codebook.

Anomaly Score Prediction In the anomaly score prediction, there exists one hyperparameter that needs to be determined, which is a set of α values. Each α determines a latent window size. As explained in Sect. 6, a latent window size can be alternatively defined as $r_w \times W$, where r_w represents a latent window size rate and $r_w \in (0, 1)$. In our experiments, we cover a range of latent window sizes from narrow to wide by using r_w values of $0.1, 0.3, 0.5$. To guarantee a minimum context for anomaly score prediction, we refrain from using α exceeding 0.7. Another parameter is stride size of a rolling window. A rolling window refers to a window from t to $t + T$, as specified in $\boldsymbol{x}^*_{t:t+T}$ in Algorithm 1. Thus, the size of the rolling window is T. In the pseudocode, the stride is set to 1, indicating the rolling window shifts from left to right by t of 1. For the faster inference, the stride can be larger than 1 such as $\lfloor r_{\text{rolling.window.stride}} \times T \rceil$ where $r_{\text{rolling.window.stride}}$ denotes a rolling window stride rate with its valid range between 0 and 1. Importantly, $r_{\text{rolling.window.stride}}$ involves a trade-off between the detection accuracy and inference speed-up. A large value of the rate leads to fast inference but can hinder the detection accuracy and vice versa. In our experiments, we set $r_{\text{rolling.window.stride}}$ to 0.1 unless specified otherwise.

Explainable Sampling To perform the explainable sampling, the prior model employs an iterative decoding sampling approach [53, 23, 60], which involves a hyperparameter known as the number of decoding steps denoted as T_s. In our implementation, we set T_s to 20, allowing for finer sampling compared to the default value of 10 used in TimeVQVAE. Additionally, we introduce another hyperparameter called *maximum masking rate for explainable sampling*. During

explainable sampling, we begin by masking anomalous segments in s. However, there is a possibility of completely masking s if all timesteps have anomaly scores above the threshold. In such cases, the prior model ends up performing unconditional sampling due to the absence of contextual information. To ensure a minimum context for explainable sampling, we limit the masking rate to a maximum of 90% of the temporal dimension, ensuring that a portion of s remains unmasked.

Optimizer   The AdamW optimizer [61] is used with specific settings: a batch size of 512 for stage 1 (256 if memory constraints arise) and a batch size of 512 for stage 2 (256 if memory constraints arise). The initial learning rate is set to 1e-3, and a cosine scheduler is employed as the learning rate scheduler. Additionally, a weight decay of 1e-5 is applied. Regarding the maximum number of epochs, 500 epochs for stage 1 and 1,000 epochs for stage 2 are used. Training ends if the duration exceeds 12 hours. The training is performed with GeForce GTX 1080 Ti.

A.3   Convolutional AutoEncoder

Convolutional AE is one of the competing methods presented in Table 1. It is implemented by utilizing the encoder and decoder architectures from TimeVQVAE-AD and replacing the existing layers with their corresponding one-dimensional versions.

B   Faster Inference

Inference speed is important from the operational perspective of the method. The inference runtime is determined by the time spent on running Algorithm 1. There are two lines in the pseudocode that can allow inference speed-up: 1) **for** $\alpha \in \{\alpha_0, \alpha_1, ...\}$ **do** and 2) **for** $t \in [0, 1, 2, ...]$ **do**. The former can be computed in parallel using multi-processing and the latter can enable the speed-up by having a stride size of $\lfloor r_{\text{rolling.window.stride}} \times T \rfloor$. While the multi-processing on α does not affect the detection accuracy, the stride size involves a trade-off between detection accuracy and inference speed. Table 2 presents the top-k accuracies and total inference runtime for the 250 datasets. Moreover, Fig. 12 presents the runtime for different sequence lengths from different datasets. To be more precise, sequence length and (sequence length / period P = number of periods in a sequence) are presented in the x-axis. These experiments were performed with Intel Core i9 11900K for CPU and GeForce RTX 3060 for GPU.

Table 2 demonstrates that the trade-off between the accuracy and computational efficiency is clearly observed. While the rate of 0.1 achieves the best accuracy, the rate of 0.5 achieves accuracy close to the best one with a significantly smaller runtime, which makes it perhaps more desirable in practice. Furthermore, Fig. 12 reveals a linear runtime increase for TimeVQVAE-AD whereas a quadratic increase for STUMPY. For a sequence with N windows (*i.e., N* subsequences), TimeVQVAE-AD runs the inference process of the prior model by N times only (*i.e.,* $O(N)$). On the other hand, a Matrix Profile method needs to compute the pairwise distances between all subsequences, resulting in $O(N^2)$ pairwise distance calculations.

Table 2: Accuracy and total inference runtime with respect to $r_{\text{rolling.window.stride}}$ values of 0.1, 0.5, and 1.0, respectively.

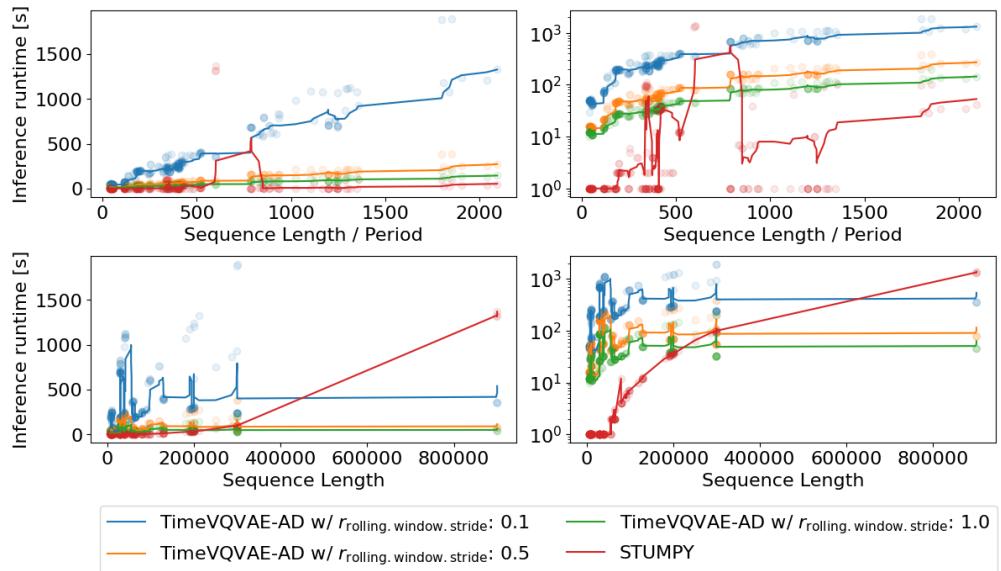| $r_{\text{rolling.window.stride}}$	Top-1 Acc.	Top-3 Acc.	Top-5 Acc.	Total inference runtime [s]
0.1	0.708	0.776	0.824	81609
0.5	0.7	0.784	0.8	18262
1.0	0.632	0.772	0.788	10352

Figure 12: Inference runtime vs sequence length (and sequence length/period). The first and second columns present a normal time scale and log time scale in the y-axis, respectively, and the first and second rows present (sequence length / period) and sequence length, respectively. TimeVQVAE-AD with different values of $r_{\text{rolling.window.stride}}$ and STUMPY, the second best performing model, are compared.