A Locally Tokenized Generative Model for Robust Time-Series Watermarking

Dongbin Kim[1 †], **Geonwoo Shin**[1 †], **Yujin Choi**[1 2], **Soyeon Park**[1], **Jaewook Lee**[1 *]

[1]Seoul National University [2]Nanyang Technological University

{dongbin413,shin0621,uznhigh,soyeon2,jaewook}@snu.ac.kr

Abstract

Watermarking is a central tool for provenance in generative models, yet its application to multivariate time series remains hindered by reliability failures under post-editing attacks. We show that existing detectors, which rely on globally coupled re-encoding, suffer from bidirectional drift of the null distribution: post-editing attacks can shift the z-score of non-watermarked samples in either direction, invalidating clean-calibrated thresholds. We argue that this instability is a property of the re-encoding, and that reliable detection requires each recovered unit to depend only on a bounded temporal neighborhood. Guided by this principle, we propose L-VQVAE, a generative model in which each discrete token is produced from a short contiguous window, and LVQMark, a watermarking method over this token space that combines logit-bias injection with robust re-encoding for attack-time detection. Experiments on four benchmarks spanning finance, energy, and neuroimaging show that our approach preserves generation quality while stabilizing both detection power and false-positive behavior under post-editing attacks.

1 Introduction

Reliable provenance verification is essential as generative models increasingly produce high-quality synthetic data [2, 15, 25]. Among existing approaches—provenance metadata, forensic detection, model fingerprinting, and watermarking [5, 14, 24]—generation-time watermarking is particularly suited to synthetic time series [9, 18]: it embeds verification signals during sampling, requires no retraining or external metadata, and remains detectable under downstream transformations [20, 22].

Despite these advances, most watermarking methods assess robustness through detectability of watermarked samples after post-editing attack, typically reporting true positive rate (TPR) at a fixed false positive rate (FPR) (TPR@X%FPR) calibrated on clean negative samples [27, 20, 18, 16]. Yet for provenance verification, false positives are also critical: attacked non-watermarked samples may be misdetected as watermarked, leading to erroneous attribution and reduced reliability of the provenance guarantee. While recent image watermarking and forgery-aware studies discuss such false-attribution risks [13, 1], they remain largely unexplored in time-series watermarking.

We address this gap by enforcing *locality* in the recovered token representation. We first introduce **L-VQVAE**, a generative model for multivariate time series in which each discrete token is recovered from a bounded temporal neighborhood of the observed signal, rather than through global inversion. L-VQVAE comprises a local tokenizer, a global decoder, and an autoregressive transformer, all sharing a single codebook. We then build **LVQMark**, a generation-time watermarking method designed over this locally recoverable token interface: it embeds a red–green logit bias during autoregressive sampling, where green-set tokens receive a positive sampling bias at each position,

[†]Equal contribution.

[*]Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

and introduces a robust encoder that maps attacked continuous signals back to their clean token assignments for detection. Because generation, insertion, and detection are all defined over the same locally recoverable token representation, perturbations affect only a bounded subset of recovered evidence, preserving both detection power and false-positive control.

We evaluate LVQMark on four time-series datasets under varying sequence lengths and post-editing attacks. Experiments show that LVQMark preserves high generation quality while achieving robust watermark detection and stable false-positive behavior on attacked non-watermarked samples.

Our main contributions are summarized as follows:

- We formulate false-positive reliability in time-series watermarking as a *re-encoding stability* problem. We show that when the re-encoding map is globally coupled, post-editing attacks can shift the null distribution of the detection statistic for non-watermarked samples, invalidating clean-calibrated thresholds.
- We introduce L-VQVAE, a generative model for multivariate time series in which each token depends only on a short temporal neighborhood, preventing perturbations from inducing global drift in the recovered representation.
- We build LVQMark, a generation-time watermarking method over the L-VQVAE token space, combining red–green logit-bias insertion with scheduling modifications tailored to L-VQVAE and robust re-encoding for attack-time detection.

2 Watermarking in Time Series and Robustness

Generalized detection framework. Generation-time watermarks [9, 7, 18] share a common statistical template. Let $\mathbf{x} \in \mathbb{R}^{T\times D}$ denote an observed sample, let $\mathcal{X}$ denote the space of recovered units (e.g. a finite codebook index set), and let $\Phi(\mathbf{x}) \in \mathcal{X}^N$ denote the internal representation recovered by the detector. For each position $n \in \{1,\dots,N\}$, a secret key κ specifies a position-wise target region $\mathcal{T}_n \subset \mathcal{X}$. Under the null hypothesis H_0, the recovered units $\{\Phi(\mathbf{x})_n\}_{n=1}^N$ are independent across positions and each lies in $\mathcal{T}_n$ with probability μ_0; watermarking biases the generative process so that this event occurs more frequently. The detector computes the agreement rate and its standardized score

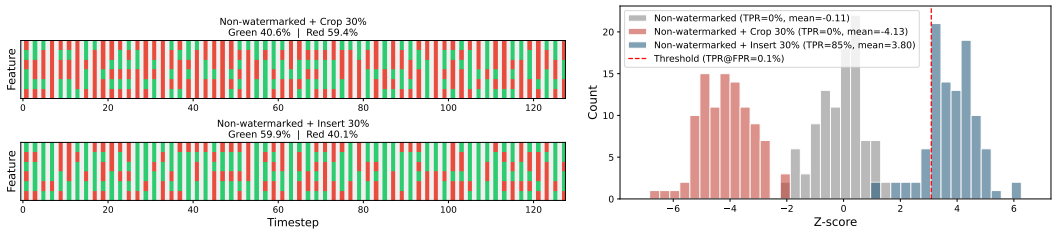
$$\hat{g}(\mathbf{x}) = \frac{1}{N}\sum_{n=1}^{N} \mathbb{1}[\Phi(\mathbf{x})_n \in \mathcal{T}_n], \qquad z(\mathbf{x}) = \frac{\hat{g}(\mathbf{x}) - \mu_0}{\sigma_0}, \qquad \sigma_0 = \sqrt{\tfrac{\mu_0(1-\mu_0)}{N}}, \tag{1}$$

and declares $\mathbf{x}$ watermarked whenever $z(\mathbf{x}) > \eta$ for a fixed threshold $\eta > 0$. In the balanced setting considered throughout this paper, $\mu_0 = 1/2$, so that $\sigma_0 = \frac{1}{2\sqrt{N}}$. This formulation subsumes both token-level generation-time watermarks for language models, such as KGW [9], and time-series methods such as TimeWak [18]: although these approaches adopt different choices of the recovered representation Φ and target regions $\mathcal{T}_n$, they share an identical statistical structure.

Red-green watermarking and time-series departure. KGW [9] recovers $\Phi(\mathbf{x})$ trivially from the observed token sequence: detection simply reproduces the green list G_n from the secret seed κ and applies Equation (1) directly. Time-series watermarking [18] adheres to the same statistical logic but departs in one essential respect: the detector must first re-encode the continuous signal into $\Phi(\mathbf{x})$. This re-encoding step is benign under clean conditions, but, as we show next, becomes the primary source of robustness degradation under post-editing attacks.

Table 1: Intended interpretation of z under a clean-calibrated detector. The validity of this interpretation requires re-encoding stability of Φ under the attack family; in its absence, null drift bidirectionally invalidates the threshold.

	$z \gg 0$	$z \approx 0$	$z \ll 0$
Watermarked	Correct detection	Miss	Statistical anomaly
Non-watermarked	False positive	Correct rejection	Statistical anomaly



(a) Recovered green/red assignments.

(b) z-score distributions.

Figure 1: Bidirectional drift of the null on *non-watermarked* samples (TimeWak [18], Stocks). Crop and insert push the recovered tokens (a) and the resulting z-score (b) in opposite directions, with insert crossing the clean-calibrated threshold and yielding an 85% false-positive rate.

Re-encoding stability. To formalize this failure mode, we call a re-encoding map Φ *stable* under an attack family $\mathcal{F}$ when, for every non-watermarked $\mathbf{x}$ and every $\mathcal{A} \in \mathcal{F}$, the distribution of $\Phi(\mathcal{A}(\mathbf{x}))$ stays close to that of $\Phi(\mathbf{x})$, keeping $\mathbb{E}_{\mathcal{H}_0}[\hat{g}(\mathcal{A}(\mathbf{x}))]$ near μ_0. We refer to violations of this property as *re-encoding instability*. Its observable symptom is the drift of $\mathbb{E}[\hat{g}(\mathcal{A}(\mathbf{x}))]$ away from μ_0 on non-watermarked samples, which we call *null drift*. We refer to the property that the clean-calibrated threshold continues to control the FPR under attack as *false-positive reliability*. Under this view, false-positive reliability is a direct consequence of re-encoding stability.

Bidirectional null drift under post-editing attacks. Existing methods fail to satisfy this property. As Figure 1 illustrates on a representative method and Table 17 confirms across baselines, $\hat{g}$ concentrates near $\mu_0 = 1/2$ in the clean setting (as Equation (1) requires) but drifts away under attack in a method- and attack-dependent manner. Since the drift direction is governed by the interaction between Φ and $\mathcal{A}$ rather than by the watermark, attacks can either trigger false positives ($z \gg 0$) or induce severe *statistical anomalies* ($z \ll 0$), and the clean-calibrated threshold loses its intended meaning under both regimes of Table 1.

A representation-level interpretation. We attribute null drift to Φ rather than to the statistic itself: existing methods instantiate Φ through *globally coupled* computations, where each recovered position depends on essentially every time step of $\mathbf{x}$, so a localized perturbation propagates to every position. We address this in Section 3 by making each unit of Φ depend only on a bounded temporal neighborhood—a property we call *locality*—confining attacks to a bounded subset of recovered units.

3 Proposed Method

Motivated by the observation that globally coupled representations propagate post-editing attacks beyond the attacked region, we design a generative model with a deliberately *local* internal representation (**L-VQVAE**) and a watermarking method that exploits this locality (**LVQMark**). A schematic is shown in Figure 2.

3.1 L-VQVAE: Locally Tokenized Generative Model

L-VQVAE generates multivariate time series through three stages: a local tokenizer encodes each short temporal window into a discrete codebook entry, a global decoder reconstructs a full time series from the resulting token sequence, and an autoregressive transformer models the distribution over token sequences for sampling.

3.1.1 Local Tokenization

Let $\mathbf{x} \in \mathbb{R}^{T \times D}$ denote a multivariate time series with T time steps and D variables. The local tokenizer produces a discrete token sequence $\mathbf{s} = (s_1, \ldots, s_N) \in \{1, \ldots, K\}^N$ from $\mathbf{x}$ in three steps: sliding-window decomposition, per-window summarization, and vector quantization against a shared codebook $\mathcal{C} = \{\mathbf{e}_k\}_{k=1}^{K}$.

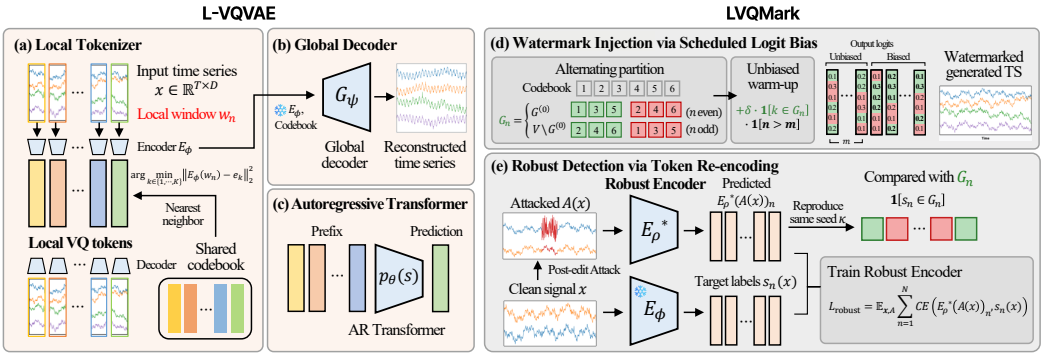


Figure 2: Overview of L-VQVAE and LVQMark. **(a)** The local tokenizer encodes each temporal window into a discrete VQ token. **(b)** The global decoder reconstructs the time series from the token sequence. **(c)** An autoregressive transformer models the token distribution. **(d)** Watermark injection via scheduled logit bias with alternating partition and unbiased warm-up. **(e)** Robust detection via token re-encoding, where E_ρ^* recovers clean token assignments from the attacked signal for the green-ratio test.

Sliding-window decomposition. We partition $\mathbf{x}$ into overlapping local windows with receptive field R and stride s (with s chosen so that $(T-R)/s$ is an integer):

$$\mathbf{w}_n = \mathbf{x}_{t_n+1:\,t_n+R} \in \mathbb{R}^{R\times D}, \qquad t_n = (n-1)\,s, \qquad n = 1,\dots,N, \tag{2}$$

where $N = (T-R)/s + 1$ is the number of windows. By construction, $\mathbf{w}_n$ depends only on the time steps within $[t_n+1,\, t_n+R]$, so a temporal post-editing attack overlaps at most $\lceil R/s \rceil + 1$ consecutive windows and can therefore alter at most that many tokens downstream.

Window summarization and quantization. Each window is summarized into a single latent vector by a shared cross-attention encoder E_ϕ and then quantized against the codebook. We linearly project $\mathbf{w}_n$ to dimension d and add learnable positional embeddings $\mathbf{P} \in \mathbb{R}^{R\times d}$ (shared across windows) to obtain $\mathbf{H}_n \in \mathbb{R}^{R\times d}$. A single learnable query vector $\mathbf{q}_0 \in \mathbb{R}^{1\times d}$, shared across all windows, is then refined by a stack of L cross-attention blocks: for $\ell = 1,\dots,L$,

$$\tilde{\mathbf{q}}_n^{(\ell)} = \mathrm{LN}\Big(\mathbf{q}_n^{(\ell-1)} + \mathrm{CrossAttn}(\mathbf{q}_n^{(\ell-1)}, \mathbf{H}_n, \mathbf{H}_n)\Big), \tag{3}$$

$$\mathbf{q}_n^{(\ell)} = \mathrm{LN}\Big(\tilde{\mathbf{q}}_n^{(\ell)} + \mathrm{FFN}(\tilde{\mathbf{q}}_n^{(\ell)})\Big), \tag{4}$$

with $\mathbf{q}_n^{(0)} = \mathbf{q}_0$ for all n. The output $\mathbf{q}_n^{(L)} \in \mathbb{R}^{1\times d}$ is squeezed and linearly projected to obtain the window latent $E_\phi(\mathbf{w}_n) \in \mathbb{R}^{d_c}$, which is mapped to its nearest codebook entry:

$$s_n = \arg\min_{k\in\{1,\dots,K\}} \|E_\phi(\mathbf{w}_n) - \mathbf{e}_k\|_2^2, \qquad \mathbf{z}_n^q = \mathbf{e}_{s_n}, \tag{5}$$

with gradients propagated through the $\arg\min$ by the straight-through estimator [19]. Because keys and values are derived exclusively from $\mathbf{w}_n$, any post-editing attack of $\mathbf{x}$ that does not overlap $\mathbf{w}_n$ leaves $E_\phi(\mathbf{w}_n)$ and s_n unchanged. For brevity, we write $\mathbf{z}_n := E_\phi(\mathbf{w}_n)$ for the continuous (pre-quantization) encoding of window $\mathbf{w}_n$, and $\mathbf{z}_n^q$ for its quantized counterpart.

Local pretraining. Training $(E_\phi, \mathcal{C})$ jointly with the global decoder from scratch leads to poor codebook utilization: the global decoder can compensate for inaccurate codes by drawing on neighboring tokens, weakening the gradient signal to the codebook and undermining the per-token locality on which detection-time recovery relies. We therefore first pretrain $(E_\phi, \mathcal{C})$ together with a *local* decoder $D_{\tilde{\psi}}^{\mathrm{loc}}$ that reconstructs $\mathbf{w}_n$ from the single code $\mathbf{z}_n^q$ alone. The encoder and local decoder are trained by minimizing

$$\mathcal{L}_{\mathrm{VQ}} = \mathbb{E}_n\left[\underbrace{\|\mathbf{w}_n - D_{\tilde{\psi}}^{\mathrm{loc}}(\mathbf{z}_n^q)\|_2^2}_{\text{reconstruction}} + \beta\underbrace{\|\mathbf{z}_n - \mathrm{sg}[\mathbf{z}_n^q]\|_2^2}_{\text{commitment}}\right], \tag{6}$$

where $\mathrm{sg}[\cdot]$ denotes the stop-gradient operator and $\mathbf{z}_n = E_\phi(\mathbf{w}_n)$. The codebook entries $\{\mathbf{e}_k\}_{k=1}^K$ are not updated by gradient descent; instead, each $\mathbf{e}_k$ is maintained as an exponential moving average of the encoder outputs $\mathbf{z}_n$. Under this local bottleneck, each code must fully describe its window without relying on neighboring context. After pretraining, $D_{\tilde{\psi}}^{\mathrm{loc}}$ is discarded, and $(E_\phi, \mathcal{C})$ are frozen for the subsequent training of the global decoder.

3.1.2 Global Decoder and Autoregressive Transformer

Locality is imposed only on the encoding side, which is the sole component re-executed at detection time and therefore exposed to post-editing attacks. Since the decoder and AR Transformer are invoked only during generation, they need not be local and can model long-range dependencies without compromising robustness.

Global decoder. The decoder G_ψ maps the full token sequence to a time series in a single forward pass. Let $\mathbf{E} \in \mathbb{R}^{N\times d}$ denote the embedded token sequence with positional embeddings, and let $\mathbf{Q}_t \in \mathbb{R}^{T\times d}$ be a learnable set of time-step queries. A stack of cross-attention blocks updates $\mathbf{Q}_t$ using $\mathbf{E}$ as keys and values,

$$\mathbf{Q}_t \leftarrow \mathrm{LN}(\mathbf{Q}_t + \mathrm{CrossAttn}(\mathbf{Q}_t, \mathbf{E}, \mathbf{E})), \qquad \mathbf{Q}_t \leftarrow \mathrm{LN}(\mathbf{Q}_t + \mathrm{FFN}(\mathbf{Q}_t)), \tag{7}$$

and a linear projection yields $\hat{\mathbf{x}} = G_\psi(\mathbf{s}) \in \mathbb{R}^{T\times D}$. G_ψ is trained with E_ϕ and the codebook frozen, under $\mathcal{L}_{\mathrm{dec}} = \|\mathbf{x} - G_\psi(\mathbf{s}(\mathbf{x}))\|_2^2$.

Autoregressive Transformer. The token sequence is modeled by a decoder-only autoregressive transformer, $p_\theta(\mathbf{s}) = \prod_{n=1}^N p_\theta(s_n \mid s_{<n})$, trained with next-token cross-entropy. Causal self-attention is restricted to a fixed lookback window, consistent with the locality principle of the tokenizer. Together, the local tokenizer, the global decoder, and the autoregressive transformer constitute L-VQVAE as a standalone generative model: sampling from p_θ and decoding through G_ψ produces a synthetic time series without any watermark.

3.2 LVQMark: Watermarking over Local VQ Tokens

LVQMark embeds a watermark into the L-VQVAE generation process by biasing the autoregressive sampling step, and provides a detection mechanism based on robust re-encoding.

3.2.1 Watermark Injection via Scheduled Logit Bias

Logit-bias watermarking. At each autoregressive position n, a secret seed κ partitions the codebook $\mathcal{V} = \{1, \ldots, K\}$ into a green subset $G_n \subseteq \mathcal{V}$ and its complementary red subset $R_n = \mathcal{V} \setminus G_n$, with $|G_n| = |R_n| = K/2$. Let $\boldsymbol{\ell}_n \in \mathbb{R}^K$ denote the logits produced by the autoregressive transformer at position n. A positive bias $\delta > 0$ is added to the logits of green tokens before sampling:

$$\tilde{\ell}_{n,k} = \ell_{n,k} + \delta\, \mathbb{1}[k \in G_n], \qquad s_n \sim \mathrm{Categorical}(\mathrm{softmax}(\tilde{\boldsymbol{\ell}}_n)), \tag{8}$$

This adapts the red–green scheme of KGW [9] (Section 2) to the L-VQVAE codebook. At detection time, the observed time series is re-encoded into tokens by the robust encoder $E_\rho^\star$ (Section 3.2.2), the green sets $\{G_n\}$ are reconstructed from the same seed κ, and the green-ratio test of Section 2 is applied with target regions $\mathcal{T}_n = G_n$.

Context-independent partition. In KGW for language models, the green set G_n typically depends on the preceding token s_{n-1}, which diversifies the partition across positions. We instead fix the partition independently of the recovered token context. The reason is locality: if G_n depended on recovered tokens, a single re-encoding error at position n would propagate to the partition at position $n{+}1$, coupling re-encoding instability with watermark scheduling. A context-independent rule confines each re-encoding error to the position where it occurs.

Failure modes of a naïve fixed partition. A context-independent partition interacts with two structural properties of L-VQVAE token sequences. First, adjacent windows largely overlap, so the AR Transformer frequently emits runs of the same token; a position-invariant G_n then applies $+\delta$ uniformly across the run, driving the green ratio toward 1 or 0 regardless of watermark presence

(*bias accumulation*). Second, at the first few autoregressive steps the context is empty and logits are nearly uniform, so a fixed δ causes near-deterministic selection of the same green token, producing a stereotyped prefix that conditions all subsequent steps (*bias dominance*).

Alternating partition. To address bias accumulation, we make the partition depend on position parity. Let $G^{(0)} \subset \mathcal{V}$ be a fixed green set determined by the secret seed κ, with $|G^{(0)}| = K/2$, and define

$$G_n = \begin{cases} G^{(0)}, & n \text{ even}, \\ \mathcal{V} \setminus G^{(0)}, & n \text{ odd}, \end{cases} \qquad R_n = \mathcal{V} \setminus G_n. \tag{9}$$

Under this rule, a token k is biased ($+\delta$) at one parity and unbiased at the other. Consequently, when the autoregressive model emits a run of identical tokens, the bias is applied at only half of the positions in the run, rather than uniformly across the entire run. This bounds the contribution of any single token value to the green-ratio statistic and prevents repeated-token runs from saturating it toward 1 or 0. Since the detector reconstructs G_n from the same seed and parity rule, no detection power is lost.

Unbiased warm-up. To address bias dominance at early steps, we suppress the logit bias for the first m positions (we use $m = 3$). Combining this with the alternating partition above, the watermarked logits become

$$\tilde{\ell}_{n,k} = \ell_{n,k} + \delta \cdot \mathbb{1}[k \in G_n] \cdot \mathbb{1}[n > m], \qquad s_n \sim \text{Categorical}(\text{softmax}(\tilde{\boldsymbol{\ell}}_n)), \tag{10}$$

so that positions $n \leq m$ are sampled from the unmodified autoregressive distribution. The detector excludes these same positions when computing the green-ratio statistic, so no detection signal is lost: at steps where the autoregressive context is empty, δ would have caused near-deterministic selection of a green token regardless of input, providing no usable evidence for the test.

3.2.2 Robust Detection via Token Re-encoding

Local tokenization confines the effect of a post-editing attack to a bounded set of tokens, but does not guarantee that those tokens are recovered to their clean codebook entries: the clean encoder E_ϕ may reassign perturbed windows to different codes. We therefore introduce a *robust encoder* $E^\star_\rho$ whose objective is to recover, from an attacked input $A(\mathbf{x})$, the token assignments that E_ϕ would have produced on $\mathbf{x}$. The two components are complementary: without local tokenization, a globally coupled representation propagates the post-editing attack to every recovered token, leaving $E^\star_\rho$ no stable reference to recover (Table 5); without $E^\star_\rho$, locally affected tokens remain misclassified.

$E^\star_\rho$ shares the windowed cross-attention architecture of E_ϕ but replaces vector quantization with a per-token classifier over the codebook vocabulary. It is trained on pairs $(\mathbf{x}, A(\mathbf{x}))$ with $\mathbf{x}$ sampled from L-VQVAE and A drawn from a fixed family of attacks (including the identity), excluding pairs in which A removes the entire span of some window:

$$\mathcal{L}_{\text{robust}} = \mathbb{E}_{\mathbf{x},A} \sum_{n=1}^{N} \text{CE}\big(E^\star_\rho(A(\mathbf{x}))_n,\ s_n(\mathbf{x})\big)\,, \tag{11}$$

where $s_n(\mathbf{x})$ are the codes produced by the frozen E_ϕ on the clean input. Joint training over multiple attack types yields a single encoder that recovers clean assignments without specializing to any one perturbation. At detection time, $E^\star_\rho$ replaces E_ϕ in the green-ratio test of Section 2.

4 Experiments

4.1 Experimental Setup

Datasets and baselines. We evaluate LVQMark on four multivariate time-series datasets: Stocks [23], ETTh [28], Energy [3], and fMRI [17], allowing us to assess the generality of the proposed method under heterogeneous temporal patterns. We compare against representative watermarking baselines, including Tree-Ring (TR) [20], Gaussian Shading (GS) [22], and TimeWak [18], which cover both context-independent and context-dependent watermarking strategies. To analyze watermarking performance together with backbone choice, we consider diffusion- and VQ-based

Table 2: Results of synthetic time series watermark detection and quality. Watermarked (TPR) and Non-watermarked (FPR, mean z-score) detections are evaluated under 30% attacks. Quality metrics are for 64-length. Best results are in bold, and second-best are underlined. z-scores with $|z| > 3.09$ are marked in red.

Setting			Watermark (TPR ↑)			Non-watermarked (FPR ↓ | z-score)						Quality Metric (↓)			
Dataset	Model	Method	Offset	Crop	Insert	Offset		Crop		Insert		C-FID	Corr.	Disc.	Pred.
Stocks	DiffusionTS	TR	0.00	**1.00**	**1.00**	**0.00**	+0.21	1.00	+69.43	1.00	+8.11	1.52	0.07	0.15	**0.04**
		GS	**1.00**	0.00	**1.00**	**0.00**	−1.40	**0.00**	−14.87	0.05	+1.52	1.50	0.02	0.23	**0.04**
		TimeWak	**1.00**	**1.00**	**1.00**	**0.00**	−0.19	**0.00**	−4.32	0.05	+1.35	0.29	**0.01**	0.13	**0.04**
	SDformer	LVQMark	0.00	0.01	0.00	**0.00**	−0.16	0.02	+1.19	**0.00**	+0.31	0.08	**0.01**	**0.01**	**0.04**
	L-VQVAE	LVQMark	**1.00**	**1.00**	**1.00**	**0.00**	−0.12	0.01	−0.05	0.01	+0.35	**0.07**	0.02	0.06	**0.04**
ETTh	DiffusionTS	TR	**1.00**	**1.00**	**1.00**	**0.00**	+0.89	1.00	+17.16	1.00	+38.55	2.17	0.22	0.29	0.14
		GS	**1.00**	**1.00**	**1.00**	**0.00**	−1.53	**0.00**	−5.44	**0.00**	−5.82	3.43	0.25	0.36	0.16
		TimeWak	**1.00**	0.61	**1.00**	0.04	+1.17	**0.00**	−0.28	0.03	+1.20	0.37	0.13	0.11	**0.12**
	SDformer	LVQMark	0.00	0.00	0.00	**0.00**	−0.94	0.02	+0.09	**0.00**	−0.09	0.04	**0.05**	**0.01**	**0.12**
	L-VQVAE	LVQMark	**1.00**	**1.00**	**1.00**	0.01	−0.12	**0.00**	−0.22	**0.00**	+0.35	**0.03**	0.06	**0.01**	**0.12**
Energy	DiffusionTS	TR	0.00	**1.00**	**1.00**	**0.00**	+1.18	1.00	+53.73	1.00	+58.55	0.58	1.98	0.43	0.28
		GS	**1.00**	0.38	**1.00**	1.00	+14.51	**0.00**	+0.87	1.00	+11.77	1.78	2.72	0.48	0.31
		TimeWak	**1.00**	0.97	**1.00**	**0.00**	−9.69	**0.00**	−0.42	**0.00**	−1.57	0.14	1.52	0.14	**0.25**
	SDformer	LVQMark	**1.00**	0.02	0.58	**0.00**	−0.02	**0.00**	+0.34	**0.00**	+0.34	**0.04**	1.05	**0.08**	**0.25**
	L-VQVAE	LVQMark	**1.00**	**1.00**	**1.00**	**0.00**	−0.06	**0.00**	−0.25	0.01	+0.04	**0.04**	**0.95**	0.15	**0.25**
fMRI	DiffusionTS	TR	**1.00**	0.02	**1.00**	**0.00**	+0.05	1.00	+2.27	1.00	+12.14	3.63	12.83	0.40	0.14
		GS	**1.00**	**1.00**	**1.00**	**0.00**	−1.78	**0.00**	−32.25	0.06	−0.01	0.74	8.31	0.50	0.11
		TimeWak	**1.00**	**1.00**	**1.00**	**0.00**	+0.08	**0.00**	−2.74	**0.00**	+0.18	0.45	1.87	0.25	0.10
	SDformer	LVQMark	0.25	0.02	0.04	**0.00**	+0.03	0.02	+0.51	**0.00**	−0.40	**0.13**	1.19	**0.12**	**0.09**
	L-VQVAE	LVQMark	**1.00**	**1.00**	**1.00**	**0.00**	−0.03	**0.00**	−0.24	**0.00**	−0.17	**0.13**	**1.14**	0.27	**0.09**

generative backbones: DiffusionTS [25] for diffusion-based watermarking methods, and TimeVQ-VAE [10] and SDFormer [4] as VQ-based generative baselines. TimeVQVAE uses discrete latent tokens with a transformer prior, while SDformer is a recent VQ-based model designed for strong time-series generation quality.

Attack setting and metrics. We evaluate robustness under three post-editing attacks—*offset*, *crop*, and *insertion*—at two strengths, 5% and 30%. The main paper reports results for the stronger 30% setting, while the 5% results are deferred to Appendix F. We assess generation quality using Context-FID [8], Correlational [11], Discriminative [23], and Predictive score [23]. For watermark reliability, we treat watermarked samples as the positive class and report TPR@0.1%FPR, where the threshold is calibrated on clean non-watermarked samples to yield 0.1% FPR, and TPR is measured on watermarked samples at test time. To directly evaluate false-positive stability under post-editing attack, we additionally report the empirical FPR on *attacked* non-watermarked samples under the same clean-calibrated detector.

All reported results are averaged over five runs conducted with fixed random seeds to ensure reproducibility. Additional details on datasets, preprocessing, attack implementation, metrics, and hyperparameters are deferred to Appendix B.

4.2 Watermark Detection and Generation Quality

Stable false-positive behavior under post-editing attacks. Table 2 reveals two distinct failure modes among existing baselines. *Spurious false positives.* Tree-Ring under crop produces mean z values of $+69.4$ on Stocks, $+17.2$ on ETTh, and $+53.7$ on Energy, driving the non-watermarked FPR to 1.00; Gaussian Shading shows the same pattern under offset on Energy ($z = +14.5$, FPR $= 1.00$). The attacked null has drifted past the clean-calibrated threshold η, causing non-watermarked samples to be systematically misattributed. *Severe negative drift.* Other baselines drift in the opposite direction: Gaussian Shading reaches $z = -32.2$ on fMRI under crop, and TimeWak reaches $z = -9.7$ on Energy under offset. These scores stay below η and yield zero empirical FPR, but the drift magnitude shows the null has been distorted well beyond its nominal scale—calibration here is accidental, not principled. FPR alone is therefore insufficient: a detector can appear calibrated when drift falls on the negative side while remaining unreliable in absolute terms. By contrast, L-VQVAE with LVQMark keeps the mean attacked z within $[-0.25, +0.35]$ across all twelve (dataset, attack) configurations, with FPR ≤ 0.01 and TPR $= 1.00$ in eleven of twelve cases.

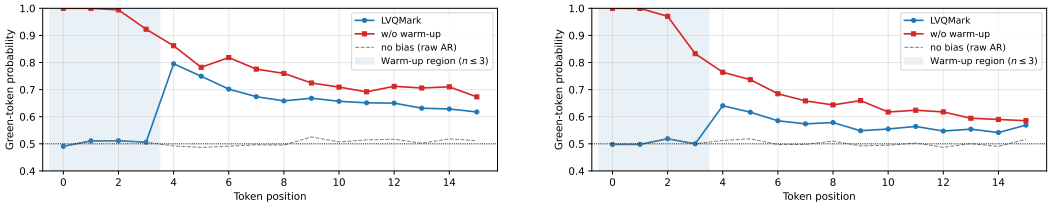
L-VQVAE achieves generation quality comparable to state-of-the-art backbones. Table 3 shows that L-VQVAE remains highly competitive in generation quality, achieving the best or second-best result on most metrics across the four datasets. By comparison, TimeVQVAE performs substantially worse on several datasets. TimeVQVAE uses separate pathways to model coarse temporal structure and fine local detail. Although this helps represent different scales of variation, cross-variable interactions are handled less explicitly, which may limit generation quality on datasets with strong inter-variable dependencies, such as Energy and fMRI.

Table 3: Results of synthetic time series quality for 64-length sequences. Best results are in bold, and second-best are underlined.

	Stocks				Energy				ETTh				fMRI			
Model	**C-FID**	**Corr.**	**Disc.**	**Pred.**	**C-FID**	**Corr.**	**Disc.**	**Pred.**	**C-FID**	**Corr.**	**Disc.**	**Pred.**	**C-FID**	**Corr.**	**Disc.**	**Pred.**
TimeVQVAE	0.439	0.079	0.368	0.056	5.423	7.189	0.466	0.293	2.432	0.238	0.453	0.244	18.643	38.410	0.275	0.255
DiffusionTS	0.294	0.013	0.126	0.037	0.143	1.519	0.135	0.251	0.370	0.134	0.114	0.117	0.446	1.869	0.253	0.100
SDformer	0.067	**0.007**	0.043	**0.036**	**0.029**	1.009	**0.081**	**0.248**	0.030	**0.052**	**0.008**	**0.112**	**0.089**	1.129	**0.114**	0.084
L-VQVAE	**0.039**	0.008	**0.018**	**0.036**	0.033	**0.955**	0.127	**0.248**	**0.024**	0.062	0.010	0.115	0.096	**1.118**	0.163	**0.083**

4.3 Component Analysis and Ablations

Both scheduling components are necessary. We ablate the two scheduling modifications introduced in Section 3.2.1: the alternating partition and the unbiased warm-up. Figure 3 shows the per-position green-token probability with and without the warm-up. Without it, the first few positions are sampled near-deterministically from the green set, creating a stereotyped prefix that propagates into later steps. The warm-up suppresses the bias during positions $n \leq m$, keeping early tokens close to the unbiased baseline and allowing the autoregressive context to diversify before the watermark signal is introduced.



(a) ETTh (b) Energy

Figure 3: Per-position green-token probability (T=64, δ=10). Shaded region: warm-up positions ($n \leq 3$) where the logit bias is suppressed.

Table 4 quantifies the impact on generation quality. Removing the warm-up alone degrades Context-FID moderately; removing both modifications together leads to further deterioration across most datasets, confirming that the two components address complementary failure modes.

Table 4: Ablation of watermark scheduling components for 64-length sequences (LVQMark). NO-WARMUP: unbiased warm-up removed. NO-ALT-WARMUP: both alternating partition and unbiased warm-up removed. Best results are in bold, and second-best are underlined.

	Stocks				Energy				ETTh				fMRI			
Variant	**C-FID**	**Corr.**	**Disc.**	**Pred.**	**C-FID**	**Corr.**	**Disc.**	**Pred.**	**C-FID**	**Corr.**	**Disc.**	**Pred.**	**C-FID**	**Corr.**	**Disc.**	**Pred.**
NO-ALT-WARMUP	0.222	**0.007**	0.094	**0.036**	0.114	1.060	0.203	0.250	0.172	0.098	0.027	0.119	0.324	1.362	0.199	0.093
NO-WARMUP	0.194	0.020	0.107	**0.036**	0.071	**0.924**	0.181	**0.248**	0.061	**0.060**	**0.007**	**0.114**	0.255	1.186	**0.170**	0.089
LVQMark	**0.068**	0.015	**0.055**	**0.036**	**0.035**	0.945	**0.151**	**0.248**	**0.028**	0.061	0.010	0.116	**0.133**	**1.135**	0.267	**0.086**

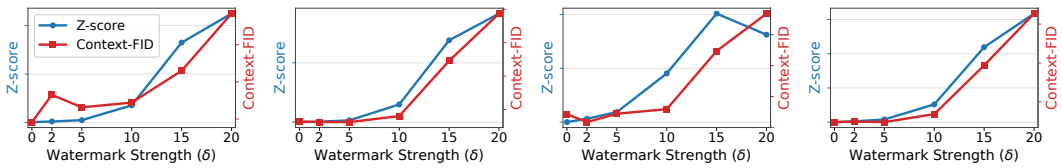
Across the four datasets, LVQMark consistently achieves the best or second-best quality on most metrics. Removing the unbiased warm-up alone degrades Context-FID and Correlational scores moderately, while removing both modifications together leads to further deterioration, particularly on ETTh, Energy, and fMRI. This incremental pattern confirms that the two modifications address complementary failure modes: the alternating partition mitigates the statistical artifact induced by repeated-token runs, while the unbiased warm-up prevents the stereotyped prefix from distorting

downstream generation. We adopt the full configuration as the default in all main experiments. Additional ablations for other sequence lengths are provided in Appendix E.2.

Table 5: Results of watermark detection under attack for 64-length sequences. LVQMark is applied to different variants, and watermark detection performance (TPR) is evaluated under 30% attacks. Best results are in bold, and second-best are underlined.

		Stocks			Energy			ETTh			fMRI		
Model	**Type**	**Offset**	**Crop**	**Insert**	**Offset**	**Crop**	**Insert**	**Offset**	**Crop**	**Insert**	**Offset**	**Crop**	**Insert**
SDformer	w/o Robust	0.00	0.01	0.00	**1.00**	0.02	0.58	0.00	0.00	0.00	**1.00**	0.02	**1.00**
	w/ Robust	0.00	0.00	0.00	**1.00**	0.66	0.80	0.13	0.08	0.03	0.05	0.28	0.03
L-VQVAE	w/o Robust	**1.00**	0.00	**1.00**	**1.00**	0.00	**1.00**	**1.00**	0.01	**1.00**	**1.00**	**1.00**	**1.00**
	w/ Robust	**1.00**	**1.00**	0.98	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**

Robust encoder requires local tokenization. Table 5 compares SDformer and L-VQVAE with and without the attack-augmented robust encoder used in our framework. The robust encoder is itself effective: paired with L-VQVAE, it lifts Crop detection from near-zero to near-perfect across all datasets while preserving Offset and Insert. With SDformer, however, gains are uneven—substantial on Energy, marginal on ETTh, absent on Stocks—and on fMRI the encoder actively *degrades* detection, collapsing Offset from 1.00 to 0.05 and Insert from 1.00 to 0.03. Robust detection thus hinges not on attack-aware training alone, but on a local representation that supports stable per-region recovery. Additional experiments for other sequence lengths are provided in Appendix E.1.



(a) Stocks (b) Energy (c) ETTh (d) fMRI

Figure 4: z-score and Context-FID across watermark strength δ for 64-length

Detectability–quality trade-off in δ. Figure 4 shows how watermark strength δ affects the trade-off between watermark detectability and generation quality under our LVQMark, where $\delta = 0$ denotes the non-watermarked baseline. As δ increases, watermark evidence becomes substantially stronger across Energy, ETTh, and fMRI, while Context-FID remains relatively stable at moderate strengths and increases more clearly only at larger values. This pattern indicates a clear trade-off: stronger bias improves detectability, but excessive bias can harm sample quality.

Table 6: Effect of the receptive field R of L-VQVAE under the 30% crop attack for 64-length sequences. Windows are non-overlapping ($s = R$) and the codebook size and token count are fixed; $R = 4$ is the default. Best results are in bold. z-scores with $|z| > 3.09$ are marked in red.

	WM (TPR ↑)					No-WM (FPR ↓ | z)									
Dataset	R=4	R=8	R=16	R=32	R=64	R=4		R=8		R=16		R=32		R=64	
Stocks	**1.00**	0.00	0.00	0.00	**1.00**	0.01	−0.05	**0.00**	−7.65	**0.00**	−1.48	**0.00**	+0.04	1.00	+8.93
ETTh	**1.00**	**1.00**	**1.00**	**1.00**	0.00	**0.00**	−0.22	0.17	+2.06	**0.00**	+0.54	**0.00**	+0.71	**0.00**	−6.01
Energy	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**0.00**	−0.25	**0.00**	−0.43	**0.00**	−1.63	0.03	+1.49	0.96	+4.91
fMRI	**1.00**	**1.00**	**1.00**	0.00	0.01	**0.00**	−0.24	0.95	+4.76	**0.00**	−0.32	**0.00**	−2.52	**0.00**	−0.11

Sensitivity to the receptive field. Table 6 varies the receptive field R from the default of 4 to 64 within L-VQVAE, with non-overlapping windows, a fixed codebook size and token count, and all other settings unchanged (Appendix D), so that locality is the only design axis that moves. Two degradation patterns appear as R grows: the attacked null drifts away from zero, in several settings far enough to raise the false-positive rate, and the watermark evidence weakens until detection fails. Only $R = 4$ satisfies both objectives on all four datasets, and the practical rule is to keep R small relative to T: locality binds only the encoder, while long-range structure is handled by the global decoder and the autoregressive transformer at generation time.

4.4 Deployment Cost

Table 7 compares the recurring cost of LVQMark with TimeWak on ETTh; the full measurements, including the higher one-time training cost of LVQMark, are in Appendix B.6. Autoregressive sampling generates an order of magnitude faster than diffusion sampling, and verification is a single forward pass of the local encoder followed by a counting test, three orders of magnitude faster than the sequential diffusion inversion of TimeWak. This profile suits the intended deployment, in which generation is a one-time operation at the data provider and verification is the recurring one, performed at high volume and often on modest hardware.

Table 7: Generation and verification cost on ETTh ($T = 64$), single GPU, batch size 1 (memory at 64).

Metric	TimeWak	Ours
Gen. throughput (samples/s)	350	**4,922**
Verif. FLOPs (GFLOP)	9.59	**1.83**
Verif. GPU latency (ms)	1,010	**0.72**
Verif. peak memory (MB)	781	**259**

4.5 Robustness Beyond Local Edits

The attacks of Table 2 are shift, mask, and replace operations. Table 8 evaluates two transformations that alter every window at once, Gaussian noise and smoothing (Appendix B.3), with the original robust encoder and with one retrained on both attacks. The original encoder transfers to every setting except Gaussian noise on Stocks, with a small loss on ETTh that retraining the encoder alone closes. The non-watermarked z-score stays near zero and the false-positive rate at or below 0.01 throughout, so null stability extends to attacks that touch every window. Stocks under Gaussian noise fails with either encoder: financial series have a low signal-to-noise ratio [12], so additive noise obscures the local patterns on which token recovery relies, while false positives remain controlled.

Table 8: Results of watermark detection under non-local attacks for 64-length sequences. Gaussian noise ($\alpha = 0.3$) and smoothing ($w = 3$) perturb every time step. Orig.: the robust encoder of the main experiments. Retr.: retrained with both attacks added to its augmentation set, with the generator and the watermark schedule unchanged. Best results are in bold.

		WM (TPR ↑)		No-WM (FPR ↓ | z)				WM (TPR ↑)		No-WM (FPR ↓ | z)			
Enc.	**Dataset**	**Noise**	**Smooth**	**Noise**		**Smooth**		**Dataset**	**Noise**	**Smooth**	**Noise**		**Smooth**	
Orig.	Stocks	**0.02**	**1.00**	**0.00**	−0.83	**0.00**	−0.06	ETTh	0.93	**1.00**	**0.00**	−0.07	**0.00**	+0.05
Retr.		0.00	**1.00**	**0.00**	−0.24	**0.00**	−0.53		**1.00**	**1.00**	0.01	−0.17	**0.00**	−0.22
Orig.	Energy	**1.00**	**1.00**	**0.00**	−0.07	**0.00**	−0.21	fMRI	**1.00**	**1.00**	**0.00**	−0.02	**0.00**	−0.06
Retr.		**1.00**	**1.00**	**0.00**	−0.06	**0.00**	+0.08		**1.00**	**1.00**	**0.00**	−0.01	**0.00**	−0.02

5 Conclusion and Limitations

In this work, we argued that false-positive reliability in time-series watermarking is fundamentally a re-encoding stability problem. Under post-editing attacks, watermark detection must first re-encode the observed continuous signal into an internal token representation, and when this re-encoding is globally coupled, post-editing attacks can distort the recovered evidence far beyond the attacked region. Motivated by this failure mode, we introduced L-VQVAE, a generative model in which each token depends only on a short temporal neighborhood, and built LVQMark on top of this locally recoverable interface with robust re-encoding and generation-time watermark injection. Across four datasets, L-VQVAE with LVQMark preserves competitive generation quality while substantially improving false-positive stability under post-editing attacks.

Limitations. Our experiments cover four multivariate time-series datasets and three sequence lengths; extending the evaluation to additional domains and longer horizons is a natural direction for future work. Robustness is assessed under three local post-editing attacks at two strength levels and two non-local transformations. Finally, the multi-stage training pipeline involves more steps than single-stage baselines, though each stage is standard and can be executed independently.

Acknowledgments

This work was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (No. RS-2025-25420430). This work was also partly supported by the NRF grant funded by the Korean government (MSIT) (No. RS-2024-00338859), and by the Rural Development Administration of Korea (No. RS-2026-25504956).

References

[1] Kasra Arabi, R Teal Witter, Chinmay Hegde, and Niv Cohen. Seal: Semantic aware image watermarking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16196–16205, 2025.

[2] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[3] Luis Candanedo. Appliances energy prediction. *UCI Machine Learning Repository*, 10: C5VC8G, 2017.

[4] Zhicheng Chen, Shibo Feng, Zhong Zhang, Xi Xiao, Xingyu Gao, and Peilin Zhao. Sdformer: Similarity-driven discrete transformer for time series generation. *Advances in Neural Information Processing Systems*, 37:132179–132207, 2024.

[5] Paul England, Henrique S Malvar, Eric Horvitz, Jack W Stokes, Cédric Fournet, Rebecca Burke-Aguero, Amaury Chamayou, Sylvan Clebsch, Manuel Costa, John Deutscher, et al. Amp: Authentication of media via provenance. In *Proceedings of the 12th ACM Multimedia Systems Conference*, pages 108–121, 2021.

[6] Zhengmian Hu, Lichang Chen, Xidong Wu, Yihan Wu, Hongyang Zhang, and Heng Huang. Unbiased watermark for large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=uWVC5FVidc`.

[7] Mingjia Huo, Sai Ashish Somayajula, Youwei Liang, Ruisi Zhang, Farinaz Koushanfar, and Pengtao Xie. Token-specific watermarking with enhanced detectability and semantic coherence for large language models. In *Proceedings of the 41st International Conference on Machine Learning*, pages 20746–20767, 2024.

[8] Paul Jeha, Michael Bohlke-Schneider, Pedro Mercado, Shubham Kapoor, Rajbir Singh Nirwan, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. Psa-gan: Progressive self attention gans for synthetic time series. In *The Tenth International Conference on Learning Representations*, 2022.

[9] John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. A watermark for large language models. In *International conference on machine learning*, pages 17061–17084. PMLR, 2023.

[10] 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.

[11] Shujian Liao, Hao Ni, Lukasz Szpruch, Magnus Wiese, Marc Sabate-Vidales, and Baoren Xiao. Conditional sig-wasserstein gans for time series generation. *arXiv preprint arXiv:2006.05421*, 2020.

[12] Xiao-Yang Liu, Ziyi Xia, Jingyang Rui, Jiechao Gao, Hongyang Yang, Ming Zhu, Christina Wang, Zhaoran Wang, and Jian Guo. Finrl-meta: Market environments and benchmarks for data-driven financial reinforcement learning. *Advances in Neural Information Processing Systems*, 35:1835–1849, 2022.

[13] Andreas Müller, Denis Lukovnikov, Jonas Thietke, Asja Fischer, and Erwin Quiring. Black-box forgery attacks on semantic watermarks for diffusion models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 20937–20946, 2025.

[14] Jonas Ricker, Denis Lukovnikov, and Asja Fischer. Aeroblade: Training-free detection of latent diffusion images using autoencoder reconstruction error. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9130–9140, 2024.

[15] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[16] Tom Sander, Pierre Fernandez, Alain Oliviero Durmus, Teddy Furon, and Matthijs Douze. Watermark anything with localized messages. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=IkZVDzdC8M.

[17] Stephen M Smith, Karla L Miller, Gholamreza Salimi-Khorshidi, Matthew Webster, Christian F Beckmann, Thomas E Nichols, Joseph D Ramsey, and Mark W Woolrich. Network modelling methods for fmri. *Neuroimage*, 54(2):875–891, 2011.

[18] Zhi Wen Soi, Chaoyi Zhu, Fouad Abiad, Aditya Shankar, Jeroen M. Galjaard, Huijuan Wang, and Lydia Y. Chen. Timewak: Temporal chained-hashing watermark for time series data. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=akhhwQh6UV.

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

[20] Yuxin Wen, John Kirchenbauer, Jonas Geiping, and Tom Goldstein. Tree-rings watermarks: Invisible fingerprints for diffusion images. *Advances in Neural Information Processing Systems*, 36:58047–58063, 2023.

[21] Yihan Wu, Ruibo Chen, Georgios Milis, and Heng Huang. An ensemble framework for unbiased language model watermarking. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=iZ7i2y1YxO.

[22] Zijin Yang, Kai Zeng, Kejiang Chen, Han Fang, Weiming Zhang, and Nenghai Yu. Gaussian shading: Provable performance-lossless image watermarking for diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12162–12171, 2024.

[23] Jinsung Yoon, Daniel Jarrett, and Mihaela Van der Schaar. Time-series generative adversarial networks. *Advances in neural information processing systems*, 32, 2019.

[24] Ning Yu, Vladislav Skripniuk, Sahar Abdelnabi, and Mario Fritz. Artificial fingerprinting for generative models: Rooting deepfake attribution in training data. In *Proceedings of the IEEE/CVF International conference on computer vision*, pages 14448–14457, 2021.

[25] Xinyu Yuan and Yan Qiao. Diffusion-TS: Interpretable diffusion for general time series generation. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=4h1apFjO99.

[26] Zhihan Yue, Yujing Wang, Juanyong Duan, Tianmeng Yang, Congrui Huang, Yunhai Tong, and Bixiong Xu. Ts2vec: Towards universal representation of time series. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pages 8980–8987, 2022.

[27] Xuandong Zhao, Prabhanjan Vijendra Ananth, Lei Li, and Yu-Xiang Wang. Provable robust watermarking for AI-generated text. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=SsmT8aO45L.

[28] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 11106–11115, 2021.

A Related Work

Discrete interfaces for time-series generation. Vector Quantization (VQ) [19] provides a discrete interface for continuous signals by mapping encoder outputs to entries in a learned codebook. The resulting code indices can be decoded back to the signal domain or modeled by a sequence prior. TimeVQVAE [10] and SDformer [4] show that such discrete token-based generators can be competitive with continuous-domain time-series generators [25] and preserve information useful for downstream classification or forecasting. However, these representations are not designed for re-encoding stability, or robustness to post-processing corruptions. These properties are essential for watermark detection, where a modified continuous signal must be re-encoded into a consistent token sequence before the watermark statistic can be computed.

Token-level generation-time watermarking. Generation-time watermarking is attractive because it embeds the watermark by steering the sampling process itself, rather than by directly perturbing the generated sample. This distinction is important because post-processing-based watermarking can directly alter generated samples and degrade quality [22]; in time-series generation, such perturbations may further distort temporal or cross-variate dependencies that are central to synthetic-data utility [23, 25]. The red-green watermarking framework [9] is a standard template for token-level generation-time watermarking in language models. At each sampling step, it partitions the vocabulary into a *green* set and its complement *red* set using a secret key, and adds a positive logit bias to the tokens in green set. This alters the sampling distribution, so that generated token sequences contain a detectable statistical bias. Subsequent methods [7, 6, 21] refine this mechanism by adapting the bias to token-level context or by designing unbiased rules that better preserve the original sampling distribution. We build on this distributional view, but instantiate it for time series over a local VQ token space. In contrast, our work focuses on detector-side re-encoding stability and false-positive behavior on attacked non-watermarked samples.

B Experiment Details

B.1 Implementation Details

Tables 9–10 summarize the hyperparameters used for each model. All models share the same data splits and preprocessing across datasets for fair comparison. Regarding the random replacement probability, during AR Transformer training each input token is independently replaced with a random codebook index with probability p, while ground-truth labels remain unchanged. This forces the model to predict the next token correctly even under corrupted context, preventing over-reliance on preceding tokens and improving generalization. For the standard VQ-VAE, which attends over the full token history, we set $p = 0.3$ for ETTh and Stocks — datasets with stronger temporal autocorrelation — and $p = 0.1$ for Energy and fMRI. For Local VQ-VAE, the AR Transformer restricts attention to a fixed lookback window of 6 tokens, which structurally limits long-range dependency; accordingly, $p = 0.1$ suffices across all datasets.

B.2 Dataset Details

We use four multivariate time-series datasets spanning diverse domains. All datasets are segmented into overlapping windows of size $T \in \{24, 64, 128\}$ with stride 1, yielding a set of fixed-length samples. Each sample is normalized per-variable to $[-1, 1]$ via MinMax scaling. The resulting samples split 80%/20% into train and test sets. Table 11 summarizes the dataset statistics.

B.3 Attack Settings

We evaluate detection robustness under three post-editing attacks: offset, insert, and crop. Each attack is applied with strength factor $\alpha \in \{0.05, 0.30\}$.

The **offset** attack shifts each variable by $\alpha \times \bar{x}_d$, where $\bar{x}_d$ is the temporal mean of variable d in the sample. The **insert** attack replaces $\lfloor \alpha T \rfloor$ randomly selected time steps with uniform random values drawn from the observed range of each variable. The **crop** attack removes a random rectangular region in the (T, D) space, retaining a window of size $(1 - \alpha)T \times (1 - \alpha)D$ at a randomly chosen position. The removed region is filled with the per-variable midpoint, defined as $(\min + \max)/2$

Table 9: L-VQVAE hyperparameter settings. Stride 2 is used for window T=24; stride 4 for T=64, 128. † Robust encoder is trained with **crop**$_{\text{var}}$.

Parameter	Energy	ETTh	fMRI	Stocks
Encoder / Decoder				
Feature size D	28	7	50	6
d_{model}	512	512	512	128
Attention heads	8	8	8	4
Encoder layers	5	5	5	5
Local decoder layers	5	5	5	5
Global decoder layers	5	5	5	5
Receptive field r	4	4	4	4
Stride (T=24)	2	2	2	2
Stride (T=64, 128)	4	4	4	4
Codebook size K	4096	512	1024	1024
Codebook dim	512	512	512	128
Commitment weight	0.25	0.25	0.25	0.25
Dropout	0.0	0.0	0.0	0.0
AR Transformer				
n_{embd}	1024	1024	1024	1024
n_{head}	8	8	8	8
n_{layer}	2	2	2	2
Random replace prob	0.1			
Lookback	6			
Watermark δ	10	10	10	20
Training				
Optimizer	AdamW, β=(0.9, 0.99)			
Batch size	128			
Learning rate	3×10^{-4}			
Gradient clip	1.0			
Pretrain steps (stage 1)	30,000			
Decoder steps (stage 2)	10,000			
Transformer steps (stage 3)	30,000			
Robust Encoder†				
d_{model}	1024	1024	1024	256
Attention heads	16	16	16	4
Layers	3			
Receptive field r	4			
Optimizer	AdamW, β=(0.9, 0.99)			
Learning rate	1×10^{-4}			
Gradient clip	1.0			
Training steps	50,000			
Val interval	5,000			
Attack augmentation (robust encoder training)				
Clean (no attack)	-			
Offset	factor $\in \{0.05, 0.30\}$			
crop$_{\text{var}}$ †	factor $\in \{0.05, 0.30\}$			
Insertion	factor $\in \{0.05, 0.30\}$			
Gaussian noise (retrained $E^{\star}_{\rho}$ only)	$\alpha = 0.3$			
Smoothing (retrained $E^{\star}_{\rho}$ only)	$w = 3$			

computed over the surviving window. Under standard min-max normalization followed by rescaling to $[-1, 1]$, this midpoint reduces to zero, making the fill semantically neutral. All model families use the same randomly drawn crop window at evaluation time.

Table 10: SDformer hyperparameter settings.

Parameter	Energy	ETTh	fMRI	Stocks
Encoder / Decoder				
Feature size D	28	7	50	6
Hidden dim	512	512	512	512
Residual blocks	2	2	2	2
Downsample rate	4	4	4	4
EMA decay	0.99	0.99	0.99	0.99
Embed loss weight	0.01	0.5	0.01	2.0
Quantizer type	cosine			
Codebook size K	512	512	512	512
Codebook dim	512	512	512	256
AR Transformer				
n_{embd}	1024	1024	1024	1024
n_{head}	8	8	8	8
n_{layer}	2	6	2	2
Random replace prob	0.1	0.3	0.1	0.3
Watermark δ	10			
Training				
Optimizer	AdamW, β=(0.9, 0.99)			
Batch size	64	128	64	128
Learning rate	3×10^{-4}			
Gradient clip	1.0			
VQ-VAE steps (stage 1)	50,000			
Transformer steps (stage 2)	50,000			

Table 11: Details of datasets used in experiments.

Dataset	# Timesteps	# Features	Source
Stocks	3,685	6	`https://finance.yahoo.com`
ETTh	17,420	7	`https://github.com/zhouhaoyi/ETDataset`
Energy	19,735	28	`https://archive.ics.uci.edu`
fMRI	10,000	50	`https://www.fmrib.ox.ac.uk/datasets`

For robust encoder training, we use an additional augmentation, $\textbf{crop}_{\text{var}}$, which retains a contiguous block of $(1-\alpha)D$ variables over the full temporal axis and fills the remaining $\lfloor \alpha D \rfloor$ variables with the per-variable midpoint. $\textbf{crop}_{\text{var}}$ differs from **crop** in that it preserves the temporal dimension and perturbs only the variable axis, thereby exposing the encoder to feature-level corruption without removing local temporal context entirely.

For the non-local evaluation in Table 8, the Gaussian noise attack adds zero-mean Gaussian noise to every observation with standard deviation α times the within-sample standard deviation of each variable, using $\alpha = 0.3$. The smoothing attack replaces each observation by the average of a centered window of $w = 3$ consecutive time steps. The retrained robust encoder of Table 8 uses the augmentation set of Table 9 extended with these two attacks, and only the robust-encoder stage is retrained. The local tokenizer, the global decoder, the autoregressive transformer, and the watermark schedule are those of the main experiments.

B.4 Evaluation Metrics

Context-FID Context-FID is an evaluation metric proposed by Jeha et al. [8] to measure the distributional gap between real and generated time series. It can be viewed as a time-series-oriented extension of the conventional Fréchet Inception Distance (FID), which is widely used in image generation tasks.

The key difference from image-based FID lies in the feature representation. While standard FID extracts image embeddings using an Inception network, Context-FID employs TS2Vec [26], a representation model specifically designed for time series. Since TS2Vec captures temporal characteristics of sequential data, the resulting distance is more suitable for assessing synthetic time series quality.

Yue et al. [26] further observed that generative models achieving lower Context-FID scores often show stronger downstream forecasting performance. This suggests that Context-FID reflects not only superficial similarity but also useful temporal distributional alignment. Accordingly, a smaller Context-FID value indicates that the generated time series more closely match the real data distribution.

Correlational score The correlational score evaluates whether synthetic time series reproduce the inter-variable dependency patterns observed in real data. To this end, we first compute the covariance between the i-th and j-th variables over the temporal dimension. Following [11], the covariance is defined as:

$$\mathrm{Cov}_{i,j} = \frac{1}{W}\sum_{t=1}^{W} K_i^t K_j^t - \left(\frac{1}{W}\sum_{t=1}^{W} K_i^t\right)\left(\frac{1}{W}\sum_{t=1}^{W} K_j^t\right).$$

Here, W is the sequence length, and K_i^t and K_j^t denote the values of the i-th and j-th features at time step t. The covariance is obtained by subtracting the product of the two feature means from the mean of their element-wise products.

After computing covariance matrices for both real and synthetic data, we compare their normalized correlation structures using the following discrepancy measure [25]:

$$\frac{1}{10}\sum_{i,j}^{d}\left|\frac{\mathrm{Cov}_{i,j}^{R}}{\sqrt{\mathrm{Cov}_{i,i}^{R}\mathrm{Cov}_{j,j}^{R}}} - \frac{\mathrm{Cov}_{i,j}^{S}}{\sqrt{\mathrm{Cov}_{i,i}^{S}\mathrm{Cov}_{j,j}^{S}}}\right|.$$

In the equation above, superscripts R and S indicate real and synthetic data, respectively, and d denotes the number of features. The summation is computed over all feature pairs. A smaller correlational score means that the synthetic data better preserves the pairwise correlation structure of the real data.

Discriminative score The discriminative score measures the distinguishability between real and synthetic time series. Specifically, a classifier is trained to predict whether each sample comes from the real dataset or the generated dataset, and the score is calculated as $|\mathrm{accuracy} - 0.5|$.

If the generated samples are highly realistic, the classifier should not be able to reliably distinguish them from real samples, and its accuracy should approach 0.5. Therefore, lower discriminative scores indicate better generation quality. Following the evaluation protocol of TimeGAN [23], we use a two-layer GRU classifier for this evaluation.

Predictive score The predictive score assesses whether generated time series preserve temporal patterns that are useful for forecasting. Rather than directly measuring visual or statistical similarity, this metric evaluates the utility of generated sequences in a prediction task.

Following the evaluation protocol of TimeGAN [23], we use a one-layer GRU-based predictor. We report the mean absolute error (MAE) between the predicted values and the ground-truth values on the test set. A lower predictive score indicates that the synthetic data better preserves the temporal dynamics required for sequence prediction.

z-score The z-score is used to evaluate the detectability of watermarks. It measures how far the detection statistic of watermarked samples deviates from that of non-watermarked samples. This metric is naturally connected to hypothesis testing. The null hypothesis H_0 assumes that the sample does not contain a watermark generated by the corresponding watermarking method.

When the z-score is close to zero, the watermarked and non-watermarked samples are statistically difficult to separate. A sufficiently large positive z-score, in contrast, provides evidence against H_0 and supports the conclusion that a watermark is present.

We consider methods that produce a scalar per-sample detection statistic x_i, such as bit accuracy for TimeWak [18] or green-list ratio for LVQMark. For these methods, the z-score is computed at the population level via a subsampled-mean construction. Let μ_{NW} and σ_{NW} denote the mean and standard deviation of x_i over a non-watermarked, attack-free reference set generated by the same model. Following TimeWak [18], we draw $B = 100$ subsamples of size $n = 1000$ without replacement. For each replicate b, we compute

$$Z_b = \frac{\bar{x}^{(b)} - \mu_{NW}}{\sigma_{NW}/\sqrt{n}}.$$

We then report $\overline{Z} = \mathrm{mean}(Z_b)$, $\sigma_Z = \mathrm{std}(Z_b)$, and the empirical decision threshold $\Pr\big(Z_b > \Phi^{-1}(0.999)\big) \approx 3.0902$.

TPR@X%FPR TPR@X%FPR is the true positive rate at a threshold calibrated to yield an X% false positive rate on clean non-watermarked samples. It measures detection power under a fixed false-positive constraint, which is particularly important in provenance verification, where false positives can lead to incorrect attribution of synthetic content. In our setting, this metric allows us to compare how reliably different methods detect watermarked samples when they are required to satisfy the same nominal false-positive budget.

B.5 Compute Resources

All experiments were conducted on a server with three GPUs: one NVIDIA H100 PCIe (80 GB) and two NVIDIA RTX PRO 6000 Blackwell Server Edition (96 GB each), an Intel Xeon Gold 6444Y dual-socket CPU (64 logical cores, up to 4.0 GHz).

B.6 Computational Cost Measurement

All cost measurements are taken on ETTh with $T = 64$ on a single NVIDIA RTX PRO 6000 GPU, with TimeWak over DiffusionTS as the baseline. Training time is the wall-clock time to train each model as configured for the main experiments, with all four stages of LVQMark included (Table 9). Generation throughput is the number of samples produced per second on the GPU, where TimeWak runs its diffusion sampler and L-VQVAE samples 16 tokens autoregressively and decodes them with the global decoder in one pass.

The verification stage is measured end to end, from the observed series to the detection decision. For LVQMark it is one forward pass of the robust encoder $E^{\star}_{\rho}$ (three windowed cross-attention layers, $d_{\text{model}} = 1024$, 38.3M parameters for ETTh) followed by the green-ratio count, without the generator or the autoregressive transformer. For TimeWak it is the 249-step BDIA inversion of the DiffusionTS model (0.38M parameters, applied sequentially) followed by its value-domain statistical test. FLOPs per sample count the dense operations of each pipeline, 114 MFLOP per window over 16 windows for LVQMark and 38.5 MFLOP per diffusion step over 249 steps for TimeWak, whose statistical test adds a negligible amount. GPU latency is reported at batch size 1, GPU throughput and peak memory at batch size 64. CPU measurements fix a single thread (`torch.set_num_threads(1)`) as a core-count-independent proxy for edge devices. Table 12 lists all measurements.

Table 12: Full cost measurements on ETTh ($T = 64$) on a single GPU. Best results are in bold.

Stage	Metric	TimeWak	LVQMark (ours)
Training (once)	Wall-clock time (min)	**9.3**	41.8
Generation	Throughput, GPU (samples/s)	350	**4,922**
Verification	FLOPs per sample (GFLOP)	9.59	**1.83**
	Latency, GPU, batch 1 (ms/sample)	1,010	**0.72**
	Throughput, GPU, batch 64 (samples/s)	47	**19,090**
	Peak GPU memory, batch 64 (MB)	781	**259**
	Latency, 1 CPU thread, batch 1 (ms/sample)	892	**23.4**
	Throughput, 1 CPU thread, batch 64 (samples/s)	3.8	**85.9**

C Effect of Partition Rules on Robust Detection

Tables 13 and 14 compare the context-dependent KGW partition with our context-independent partition under 5% and 30% attacks, respectively. Under the KGW rule, the green set at position n is determined by the recovered token at $n-1$, so a single re-encoding error can affect the partition at the next position and propagate a local perturbation beyond the attacked region. In contrast, our context-independent partition depends only on the position index and the secret key, eliminating this cross-position coupling and confining each re-encoding error to its originating position. As a result, our partition consistently yields higher watermark z-scores across datasets and attack types under both attack strengths. This shows that the robustness benefit of preserving locality is systematic rather than specific to the stronger 30% setting. Combined with parity alternation and the bias-free warm-up (Section 3.2.1), the proposed design leads to more stable recovered evidence and stronger overall watermark detectability.

Table 13: Partition rule comparison. z-scores under 30% attacks. Best results are in bold.

	Stocks			Energy			ETTh			fMRI		
Partition	**Offset**	**Crop**	**Insert.**	**Offset**	**Crop**	**Insert.**	**Offset**	**Crop**	**Insert.**	**Offset**	**Crop**	**Insert.**
KGW	14.96	5.38	2.76	14.73	10.06	9.98	10.51	8.14	7.26	12.71	13.12	9.31
Ours	**15.39**	**16.19**	**9.16**	**17.30**	**13.77**	**13.14**	**17.10**	**16.09**	**13.31**	**14.55**	**16.72**	**11.88**

Table 14: Partition rule comparison. z-scores under 5% attacks. Best results are in bold.

	Stocks			Energy			ETTh			fMRI		
Partition	**Offset**	**Crop**	**Insert**	**Offset**	**Crop**	**Insert**	**Offset**	**Crop**	**Insert**	**Offset**	**Crop**	**Insert**
KGW	12.95	8.62	8.44	14.80	13.30	14.27	11.09	9.02	10.55	12.77	13.14	12.66
Ours	**19.51**	**15.80**	**16.37**	**17.35**	**15.74**	**16.98**	**17.65**	**16.53**	**17.80**	**14.68**	**15.19**	**14.49**

D Receptive-Field Sweep

The models with $R \in \{8, 16, 32, 64\}$ in Table 6 use stride $s = R$, so the windows tile the sequence without overlap, and each window is summarized by $R/4$ codes from the same codebook, so that the token sequence keeps 16 tokens at $T = 64$ and the codebook size of Table 9 is unchanged. Each model is trained from scratch with the four-stage procedure of the main experiments, and detection uses a robust encoder trained for that R with the augmentation set of Table 9. Table 15 reports clean detection and Context-FID across R. At $R = 8$ and 16 on Stocks and at $R \geq 32$ on fMRI the watermark is not recoverable on clean samples, so the zeros in Table 6 at those settings reflect the representation rather than the attack. Context-FID rises with R for both non-watermarked and watermarked generation, gradually at moderate fields and sharply at $R = 64$.

Table 15: Clean detection and generation quality across the receptive field R for 64-length sequences. No-WM / WM: Context-FID of non-watermarked and watermarked generation.

Dataset	**Metric**	R=4	R=8	R=16	R=32	R=64
Stocks	Clean TPR	1.00	0.00	0.00	1.00	1.00
	C-FID (No-WM / WM)	0.04 / 0.07	0.03 / 0.15	0.03 / 0.32	0.08 / 0.39	0.10 / 0.22
ETTh	Clean TPR	1.00	1.00	1.00	1.00	0.03
	C-FID (No-WM / WM)	0.02 / 0.03	0.03 / 0.06	0.03 / 0.08	0.07 / 0.09	11.44 / 11.90
Energy	Clean TPR	1.00	1.00	1.00	1.00	0.10
	C-FID (No-WM / WM)	0.03 / 0.04	0.07 / 0.14	0.20 / 0.28	0.54 / 0.59	1.39 / 1.52
fMRI	Clean TPR	1.00	1.00	1.00	0.00	0.00
	C-FID (No-WM / WM)	0.10 / 0.13	0.15 / 0.37	3.24 / 3.22	10.51 / 10.74	26.82 / 27.33

E Experimental Results For Other Lengths

This section provides additional results for other sequence lengths ($T = 24, 128$) and complements the main analysis at $T = 64$. The overall trend is consistent with the main results.

E.1 Watermark Detection and Generation Quality

False-positive behavior under post-editing attacks Tables 16 and 17 extend the analysis of bidirectional null shift in Section 4.2 to other sequence lengths. Consistent with the main results, the same qualitative pattern persists across other sequence lengths, further supporting that the attacked null remains stable under L-VQVAE with LVQMark while existing baselines exhibit substantial drift.

Table 16: Results of synthetic time series watermark detection and quality. Watermarked (TPR) and Non-watermarked (FPR, mean z-score) detections are evaluated under 30% attacks. Quality metrics are for 24-length. Best results are in bold, and second-best are underlined. z-scores with $|z| > 3.09$ are marked in red.

Setting			Watermark (TPR ↑)			Non-watermarked (FPR ↓ | z-score)						Quality Metric (↓)			
Dataset	Model	Method	Offset	Crop	Insert	Offset		Crop		Insert		C-FID	Corr.	Disc.	Pred.
Stocks	DiffusionTS	TR	0.00	**1.00**	0.00	**0.00**	+0.19	1.00	+63.26	**0.00**	+1.91	0.96	0.10	0.20	**0.04**
		GS	**1.00**	0.45	**1.00**	**0.00**	−1.15	**0.00**	−6.32	**0.00**	+0.16	8.88	0.09	0.43	**0.04**
		TimeWak	**1.00**	**1.00**	**1.00**	**0.00**	−0.99	0.02	+1.39	**0.00**	−1.77	0.33	0.02	0.16	**0.04**
	SDformer	LVQMark	0.13	0.81	0.22	**0.00**	+0.51	0.86	+4.49	0.01	+1.33	0.11	**0.01**	**0.02**	**0.04**
	L-VQVAE	LVQMark	**1.00**	0.99	0.98	**0.00**	−0.25	0.04	+1.10	0.02	+0.38	**0.07**	**0.01**	0.11	**0.04**
ETTh	DiffusionTS	TR	**1.00**	**1.00**	**1.00**	**0.00**	+0.95	1.00	+10.84	1.00	+15.97	1.56	0.18	0.27	0.14
		GS	**1.00**	**1.00**	**1.00**	1.00	+12.01	1.00	+18.58	1.00	+6.96	4.66	0.42	0.38	0.19
		TimeWak	**1.00**	0.48	**1.00**	**0.00**	−0.34	0.02	+1.05	**0.00**	−1.68	0.23	0.21	0.08	**0.12**
	SDformer	LVQMark	0.98	0.03	0.07	0.01	+0.79	**0.01**	+0.18	**0.00**	+0.70	0.13	0.08	0.04	**0.12**
	L-VQVAE	LVQMark	**1.00**	**1.00**	**1.00**	**0.00**	+0.03	**0.01**	+0.41	**0.00**	−0.15	**0.05**	**0.05**	**0.02**	**0.12**
Energy	DiffusionTS	TR	0.00	**1.00**	**1.00**	**0.00**	+0.43	1.00	+39.25	1.00	+26.77	0.43	2.66	0.41	0.30
		GS	**1.00**	**1.00**	**1.00**	1.00	+13.17	1.00	+65.49	1.00	+14.31	1.58	3.37	0.49	0.33
		TimeWak	**1.00**	**1.00**	**1.00**	**0.00**	+0.33	0.99	+6.03	**0.00**	−1.22	0.09	1.53	0.14	**0.25**
	SDformer	LVQMark	**1.00**	0.76	**1.00**	**0.00**	+0.29	0.07	+1.48	**0.00**	+0.32	0.12	1.27	0.23	**0.25**
	L-VQVAE	LVQMark	**1.00**	**1.00**	**1.00**	**0.00**	−0.16	**0.00**	−0.19	**0.00**	−0.03	**0.03**	**1.02**	**0.09**	**0.25**
fMRI	DiffusionTS	TR	0.04	0.00	0.00	**0.00**	+0.42	**0.00**	+1.00	1.00	+4.89	2.26	13.42	0.50	0.15
		GS	**1.00**	**1.00**	**1.00**	0.24	+2.33	**0.00**	−2.26	**0.00**	−6.16	0.71	15.21	0.50	0.11
		TimeWak	**1.00**	**1.00**	**1.00**	**0.00**	+0.19	0.22	+2.39	**0.00**	+0.26	**0.18**	1.98	**0.09**	**0.10**
	SDformer	LVQMark	**1.00**	0.99	**1.00**	**0.00**	+0.14	0.01	−0.41	0.01	+0.01	0.94	3.25	0.21	**0.10**
	L-VQVAE	LVQMark	**1.00**	**1.00**	**1.00**	**0.00**	+0.11	0.02	−0.23	**0.00**	−0.05	0.20	**1.92**	0.23	**0.10**

Generation quality Tables 18 and 19 show the generation quality results for sequence lengths 24 and 128, respectively. L-VQVAE remains competitive in quality while improving watermark reliability under attack.

E.2 Component Analysis and Ablations

Effect of watermark scheduling Tables 20 and 21 extend the scheduling ablation in Section 4.3 to other sequence lengths. We compare the default configuration against variants without unbiased warm-up (NO-WARMUP) and without both alternating partition and unbiased warm-up (NO-ALT-WARMUP). Overall, the default configuration yields the most consistent quality across datasets, and the degradation is typically larger when both components are removed. These trends are broadly consistent with the main results at 64-length.

Effect of local tokenization on robust detection Tables 22 and 23 extend the analysis of Section 4.3 to other sequence lengths by comparing models with and without robust encoder. These results show that robust re-encoding alone is insufficient without a locally recoverable token representation.

F Experimental Results Under Mild Attack

While the main paper reports results under the stronger 30% attack setting, this section provides the corresponding results under milder 5% attacks. We include the full detection and quality results,

Table 17: Results of synthetic time series watermark detection and quality. Watermarked (TPR) and Non-watermarked (FPR, mean z-score) detections are evaluated under 30% attacks. Quality metrics are for 128-length. Best results are in bold, and second-best are underlined. z-scores with $|z| > 3.09$ are marked in red.

Setting			Watermark (TPR ↑)			Non-watermarked (FPR ↓ | z-score)						Quality Metric (↓)			
Dataset	Model	Method	Offset	Crop	Insert	Offset		Crop		Insert		C-FID	Corr.	Disc.	Pred.
Stocks	DiffusionTS	TR	0.00	**1.00**	**1.00**	**0.00**	+0.30	1.00	+84.66	1.00	+20.60	3.05	0.09	0.24	**0.04**
		GS	**1.00**	0.04	**1.00**	**0.00**	−4.40	**0.00**	−10.04	**0.00**	−5.22	2.63	0.04	0.19	**0.04**
		TimeWak	**1.00**	**1.00**	**1.00**	**0.00**	−0.16	**0.00**	−4.18	0.69	+3.74	0.34	**0.01**	0.15	**0.04**
	SDformer	LVQMark	0.00	0.00	0.00	**0.00**	−0.54	**0.00**	−0.89	**0.00**	−1.20	0.12	**0.01**	**0.03**	**0.04**
	L-VQVAE	LVQMark	**1.00**	**1.00**	**1.00**	**0.00**	+0.09	**0.00**	−0.09	**0.00**	+0.23	**0.08**	**0.01**	0.18	**0.04**
ETTh	DiffusionTS	TR	0.00	**1.00**	**1.00**	**0.00**	+1.52	1.00	+12.19	1.00	+36.77	2.52	0.26	0.30	0.13
		GS	**1.00**	**1.00**	**1.00**	**0.00**	−6.09	**0.00**	−13.71	**0.00**	−6.37	5.59	0.23	0.39	0.15
		TimeWak	**1.00**	0.92	**1.00**	0.02	+1.09	**0.00**	−1.63	**0.00**	+0.77	1.08	0.18	0.15	**0.11**
	SDformer	LVQMark	0.14	0.04	0.04	**0.00**	+0.53	0.01	+0.76	**0.00**	+1.01	**0.04**	**0.05**	0.03	**0.11**
	L-VQVAE	LVQMark	**1.00**	**1.00**	**1.00**	**0.00**	−0.04	**0.00**	−0.05	**0.00**	−0.30	**0.04**	**0.05**	**0.02**	**0.11**
Energy	DiffusionTS	TR	0.00	**1.00**	**1.00**	**0.00**	+2.20	1.00	+61.47	1.00	+98.65	0.50	1.65	0.49	0.28
		GS	**1.00**	**1.00**	**1.00**	1.00	+15.43	1.00	+48.30	1.00	+27.89	3.34	3.59	0.48	0.29
		TimeWak	**1.00**	0.10	**1.00**	**0.00**	−1.57	**0.00**	−0.78	**0.00**	+0.66	0.17	1.50	0.24	0.25
	SDformer	LVQMark	**1.00**	0.00	0.32	**0.00**	+0.06	**0.00**	−0.65	**0.00**	−0.08	0.04	**0.67**	**0.05**	0.25
	L-VQVAE	LVQMark	**1.00**	0.99	**1.00**	**0.00**	−0.03	0.02	+0.00	**0.00**	+0.13	**0.03**	0.74	0.19	**0.24**
fMRI	DiffusionTS	TR	0.00	**1.00**	**1.00**	**0.00**	+0.02	1.00	+6.33	1.00	+21.25	4.57	14.15	0.45	0.17
		GS	**1.00**	**1.00**	**1.00**	**0.00**	−0.70	**0.00**	−57.42	0.07	+0.52	1.05	5.97	0.50	0.11
		TimeWak	**1.00**	**1.00**	**1.00**	**0.00**	−0.07	**0.00**	−3.03	**0.00**	−1.95	0.81	1.81	0.36	0.10
	SDformer	LVQMark	0.10	0.02	0.03	**0.00**	−0.03	0.01	−0.25	**0.00**	−0.15	**0.09**	**0.84**	0.16	**0.08**
	L-VQVAE	LVQMark	**1.00**	**1.00**	**1.00**	**0.00**	−0.09	**0.00**	+0.16	**0.00**	−0.01	0.13	0.90	**0.06**	**0.08**

Table 18: Results of synthetic time series quality for 24-length sequences. Best results are in bold, and second-best are underlined.

	Stocks				Energy				ETTh				fMRI			
Model	C-FID	Corr.	Disc.	Pred.	C-FID	Corr.	Disc.	Pred.	C-FID	Corr.	Disc.	Pred.	C-FID	Corr.	Disc.	Pred.
TimeVQVAE	0.507	0.049	0.361	0.047	2.527	6.632	0.476	0.383	4.417	0.263	0.415	0.199	12.532	61.599	0.459	0.156
DiffusionTS	0.329	0.022	0.160	**0.037**	0.087	1.534	0.137	0.253	0.231	0.212	0.078	0.121	0.181	1.976	0.085	0.100
SDformer	**0.028**	**0.009**	**0.023**	**0.037**	0.030	1.088	0.093	**0.252**	0.036	0.066	**0.010**	0.120	0.114	**1.833**	**0.056**	**0.092**
L-VQVAE	0.034	0.010	0.140	**0.037**	**0.018**	**1.008**	**0.033**	**0.252**	**0.029**	**0.052**	0.018	**0.117**	**0.105**	1.875	0.209	0.095

Table 19: Results of synthetic time series quality for 128-length sequences. Best results are in bold, and second-best are underlined.

	Stocks				Energy				ETTh				fMRI			
Model	C-FID	Corr.	Disc.	Pred.	C-FID	Corr.	Disc.	Pred.	C-FID	Corr.	Disc.	Pred.	C-FID	Corr.	Disc.	Pred.
TimeVQVAE	0.293	0.087	0.135	0.039	7.738	6.631	0.498	0.344	2.205	0.251	0.256	0.143	30.672	45.692	0.250	0.149
DiffusionTS	0.344	0.014	0.152	0.037	0.169	1.505	0.235	0.249	1.075	0.175	0.145	0.113	0.810	1.812	0.357	0.100
SDformer	0.128	**0.013**	**0.032**	**0.036**	0.032	**0.708**	**0.057**	**0.244**	**0.032**	**0.050**	0.062	0.113	**0.064**	**0.801**	0.176	**0.080**
L-VQVAE	**0.038**	**0.013**	0.035	**0.036**	**0.031**	0.733	0.155	0.246	0.039	0.052	**0.025**	**0.107**	0.096	0.895	**0.054**	0.082

Table 20: Ablation of watermark scheduling components for 24-length sequences (LVQMark). NO-WARMUP: unbiased warm-up removed. NO-ALT-WARMUP: both alternating partition and unbiased warm-up removed. Best results are in bold, and second-best are underlined.

	Stocks				Energy				ETTh				fMRI			
Variant	C-FID	Corr.	Disc.	Pred.	C-FID	Corr.	Disc.	Pred.	C-FID	Corr.	Disc.	Pred.	C-FID	Corr.	Disc.	Pred.
NO-ALT-WARMUP	0.211	0.014	0.141	**0.037**	0.081	1.172	0.151	0.254	0.180	0.113	0.066	0.123	0.419	2.350	0.298	0.101
NO-WARMUP	0.160	0.013	0.224	**0.037**	0.060	1.063	0.130	0.253	0.093	0.059	0.039	0.123	0.346	2.067	0.296	0.098
LVQMark	**0.074**	**0.007**	**0.110**	**0.037**	**0.028**	**1.022**	**0.090**	**0.252**	**0.046**	**0.048**	**0.021**	**0.117**	**0.200**	**1.920**	**0.229**	**0.097**

the robust-encoder ablations in order to examine whether the main observations are specific to strong perturbations. Overall, the results are broadly consistent with the main findings: although the attack-induced distortion is often smaller than in the 30% setting, the relative pattern across methods remains largely unchanged.

Table 21: Ablation of watermark scheduling components for 128-length sequences (LVQMark). NO-WARMUP: unbiased warm-up removed. NO-ALT-WARMUP: both alternating partition and unbiased warm-up removed. Best results are in bold, and second-best are underlined.

	Stocks				Energy				ETTh				fMRI			
Variant	**C-FID**	**Corr.**	**Disc.**	**Pred.**	**C-FID**	**Corr.**	**Disc.**	**Pred.**	**C-FID**	**Corr.**	**Disc.**	**Pred.**	**C-FID**	**Corr.**	**Disc.**	**Pred.**
NO-ALT-WARMUP	0.185	0.008	0.148	**0.036**	0.093	0.869	0.235	0.247	0.166	0.083	0.035	0.111	0.257	1.023	0.134	0.084
NO-WARMUP	0.095	**0.007**	**0.120**	**0.036**	0.066	0.774	0.200	0.246	0.073	0.063	0.027	**0.108**	0.198	0.900	0.086	0.084
LVQMark	**0.079**	0.009	0.179	**0.036**	**0.033**	**0.740**	**0.190**	**0.244**	**0.042**	**0.050**	**0.021**	**0.108**	**0.127**	**0.896**	**0.060**	**0.082**

Table 22: Results of watermark detection under attack for 24-length sequences. LVQMark is applied to different variants, and watermark detection performance (TPR) is evaluated under 30% attacks. Best results are in bold, and second-best are underlined.

		Stocks			Energy			ETTh			fMRI		
Model	**Type**	**Offset**	**Crop**	**Insert**	**Offset**	**Crop**	**Insert**	**Offset**	**Crop**	**Insert**	**Offset**	**Crop**	**Insert**
SDformer	w/o Robust	0.13	0.81	0.22	**1.00**	0.76	**1.00**	0.98	0.03	0.07	**1.00**	0.99	**1.00**
	w/ Robust	0.07	0.20	0.08	**1.00**	0.68	**1.00**	**1.00**	0.57	0.96	**1.00**	**1.00**	**1.00**
L-VQVAE	w/o Robust	**1.00**	0.21	0.35	**1.00**	0.00	**1.00**	**1.00**	0.01	**1.00**	**1.00**	**1.00**	**1.00**
	w/ Robust	**1.00**	**0.99**	**0.98**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**

Table 23: Results of watermark detection under attack for 128-length sequences. LVQMark is applied to different variants, and watermark detection performance (TPR) is evaluated under 30% attacks. Best results are in bold, and second-best are underlined.

		Stocks			Energy			ETTh			fMRI		
Model	**Type**	**Offset**	**Crop**	**Insert**	**Offset**	**Crop**	**Insert**	**Offset**	**Crop**	**Insert**	**Offset**	**Crop**	**Insert**
SDformer	w/o Robust	0.00	0.00	0.00	**1.00**	0.00	0.32	0.14	0.04	0.04	0.10	0.02	0.03
	w/ Robust	0.00	0.00	0.00	**1.00**	0.20	0.85	0.41	0.11	0.29	0.05	0.01	0.01
L-VQVAE	w/o Robust	**1.00**	0	0.99	1.00	0.00	0.81	1.00	0.00	1.00	1.00	1.00	0.62
	w/ Robust	**1.00**	**1.00**	**1.00**	**1.00**	**0.99**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**

F.1 Watermark Detection

Tables 24–26 show that the qualitative conclusions of the main paper remain unchanged under the milder 5% attack setting. Compared with the 30% setting, the attack-induced distortion is generally smaller in magnitude, but the relative pattern across methods remains largely the same. Across sequence lengths, L-VQVAE with LVQMark continues to achieve strong watermark detection on watermarked samples while keeping the false-positive rate on attacked non-watermarked samples consistently low. The corresponding non-watermarked z-scores also remain close to the intended null region, indicating that the detector also stays well calibrated under mild post-editing attacks. By contrast, the baseline methods still exhibit post-editing-dependent instability, showing that the calibration problem is not limited to severe attacks but can already arise under small perturbations. Overall, these results suggest that mild attacks reduce, but do not eliminate, the robustness gap observed in the 30% setting.

F.2 Effect of Local Tokenization on Robust Detection

Tables 27–29 further examine whether robust re-encoding alone is sufficient in the mild-attack regime. The results are consistent with the main-paper ablation: adding a robust encoder can partially improve detection in some settings, but the gains remain limited and inconsistent when the underlying representation is not locally recoverable. By contrast, when robust re-encoding is combined with the local tokenization of L-VQVAE, detection performance remains uniformly strong across datasets, attacks, and sequence lengths. This again supports our central claim that robust recovery is not solely a matter of attack-aware training, but depends fundamentally on a representation whose recovered units remain locally stable under perturbation.

Table 24: Results of synthetic time series watermark detection and quality. Watermarked (TPR) and Non-watermarked (FPR, mean z-score) detections are evaluated under 5% attacks. Quality metrics are for 24-length. Best results are in bold, and second-best are underlined. z-scores with $|z| > 3.09$ are marked in red.

Setting			Watermark (TPR ↑)			Non-watermarked (FPR ↓ | z-score)						Quality Metric (↓)			
Dataset	Model	Method	Offset	Crop	Insert	Offset		Crop		Insert		C-FID	Corr.	Disc.	Pred.
Stocks	DiffusionTS	TR	0.00	**1.00**	0.00	**0.00**	+0.06	1.00	+52.15	**0.00**	+0.41	0.96	0.10	0.20	**0.04**
		GS	**1.00**	**1.00**	**1.00**	**0.00**	−0.27	1.00	+22.14	0.43	+2.88	8.88	0.09	0.43	**0.04**
		TimeWak	**1.00**	**1.00**	**1.00**	**0.00**	−0.20	1.00	+9.36	**0.00**	+0.58	0.33	0.02	0.16	**0.04**
	SDformer	LVQMark	0.19	0.44	0.72	**0.00**	+0.19	0.53	+3.11	**0.00**	+0.22	0.11	**0.01**	**0.02**	**0.04**
	L-VQVAE	LVQMark	**1.00**	**1.00**	**1.00**	**0.00**	+0.11	**0.02**	+0.43	**0.00**	+0.09	**0.07**	**0.01**	0.11	**0.04**
ETTh	DiffusionTS	TR	**1.00**	**1.00**	**1.00**	**0.00**	+0.19	1.00	+10.11	1.00	+4.78	1.56	0.18	0.27	0.14
		GS	**1.00**	**1.00**	**1.00**	0.18	+2.17	**0.00**	−10.53	0.14	+2.02	4.66	0.42	0.38	0.18
		TimeWak	**1.00**	**1.00**	**1.00**	**0.00**	−0.26	**0.00**	+0.77	**0.00**	−0.34	0.23	0.21	0.08	**0.12**
	SDformer	LVQMark	**1.00**	0.01	**1.00**	**0.00**	+0.15	**0.00**	+0.10	**0.00**	+0.29	0.13	0.08	0.04	**0.12**
	L-VQVAE	LVQMark	**1.00**	**1.00**	**1.00**	**0.00**	+0.28	**0.00**	+0.13	**0.00**	−0.14	**0.05**	**0.05**	**0.02**	**0.12**
Energy	DiffusionTS	TR	0.00	**1.00**	**1.00**	**0.00**	+0.03	1.00	+36.66	1.00	+9.01	0.43	2.66	0.41	0.30
		GS	**1.00**	**1.00**	**1.00**	0.59	+3.33	1.00	+43.84	0.48	+3.17	1.58	3.37	0.49	0.33
		TimeWak	**1.00**	**1.00**	**1.00**	**0.00**	−1.43	1.00	+5.79	**0.00**	−0.44	0.09	1.53	0.14	**0.25**
	SDformer	LVQMark	**1.00**	**1.00**	**1.00**	**0.00**	−0.22	**0.00**	+0.15	**0.00**	+0.42	0.12	1.27	0.23	**0.25**
	L-VQVAE	LVQMark	**1.00**	**1.00**	**1.00**	**0.00**	−0.11	0.01	−0.02	**0.00**	−0.05	**0.03**	**1.02**	**0.09**	**0.25**
fMRI	DiffusionTS	TR	**1.00**	0.00	0.00	**0.00**	+0.06	**0.00**	+0.20	**0.00**	+0.70	2.26	13.42	0.50	0.15
		GS	**1.00**	**1.00**	**1.00**	**0.00**	+0.58	0.20	+2.37	**0.00**	−0.42	0.71	15.21	0.50	0.11
		TimeWak	**1.00**	**1.00**	**1.00**	**0.00**	−0.09	**0.00**	−1.07	**0.00**	−0.28	**0.18**	1.98	**0.08**	**0.10**
	SDformer	LVQMark	**1.00**	**1.00**	**1.00**	**0.00**	−0.08	**0.00**	−0.36	**0.00**	−0.07	0.94	3.24	0.21	**0.10**
	L-VQVAE	LVQMark	**1.00**	**1.00**	**1.00**	**0.00**	+0.07	**0.00**	−0.11	**0.00**	−0.14	0.20	**1.92**	0.23	**0.10**

Table 25: Results of synthetic time series watermark detection and quality. Watermarked (TPR) and Non-watermarked (FPR, mean z-score) detections are evaluated under 5% attacks. Quality metrics are for 64-length. Best results are in bold, and second-best are underlined. z-scores with $|z| > 3.09$ are marked in red.

Setting			Watermark (TPR ↑)			Non-watermarked (FPR ↓ | z-score)						Quality Metric (↓)			
Dataset	Model	Method	Offset	Crop	Insert	Offset		Crop		Insert		C-FID	Corr.	Disc.	Pred.
Stocks	DiffusionTS	TR	0.00	**1.00**	0.00	**0.00**	+0.06	1.00	+55.65	**0.00**	+2.06	1.52	0.07	0.15	**0.04**
		GS	**1.00**	**1.00**	**1.00**	**0.00**	−0.35	**0.00**	−4.62	0.05	+1.55	1.50	0.02	0.22	**0.04**
		TimeWak	**1.00**	**1.00**	**1.00**	**0.00**	−0.12	0.29	+2.56	**0.00**	+0.11	0.29	**0.01**	0.13	**0.04**
	SDformer	LVQMark	0.01	0.03	0.01	**0.00**	−0.17	**0.00**	+0.58	0.01	+0.32	0.08	**0.01**	**0.01**	**0.04**
	L-VQVAE	LVQMark	**1.00**	**1.00**	**1.00**	**0.00**	−0.33	**0.00**	+0.08	**0.00**	+0.30	**0.07**	**0.01**	0.06	**0.04**
ETTh	DiffusionTS	TR	**1.00**	**1.00**	**1.00**	**0.00**	+0.21	1.00	+14.18	1.00	+12.70	2.17	0.22	0.29	0.14
		GS	**1.00**	**1.00**	**1.00**	**0.00**	−0.29	**0.00**	−29.58	**0.00**	−1.60	3.43	0.25	0.36	0.16
		TimeWak	**1.00**	**1.00**	**1.00**	0.01	+0.19	0.90	+4.08	**0.00**	+0.23	0.37	0.13	0.11	**0.12**
	SDformer	LVQMark	0.27	0.00	0.06	0.01	+0.25	**0.00**	−0.27	**0.00**	−0.11	0.04	**0.05**	**0.00**	**0.12**
	L-VQVAE	LVQMark	**1.00**	**1.00**	**1.00**	**0.00**	−0.17	**0.00**	−0.07	0.01	+0.22	**0.03**	0.06	0.01	**0.12**
Energy	DiffusionTS	TR	0.00	**1.00**	**1.00**	**0.00**	+0.05	1.00	+50.50	1.00	+21.61	0.58	1.98	0.43	0.28
		GS	**1.00**	**1.00**	**1.00**	0.07	+1.21	1.00	+11.15	0.37	+2.71	1.78	2.72	0.48	0.31
		TimeWak	**1.00**	**1.00**	**1.00**	**0.00**	−1.26	**0.00**	−1.42	**0.00**	−0.84	0.14	1.52	0.14	**0.25**
	SDformer	LVQMark	**1.00**	0.46	**1.00**	**0.00**	+0.06	**0.00**	+0.23	**0.00**	+0.07	**0.04**	1.04	**0.08**	**0.25**
	L-VQVAE	LVQMark	**1.00**	**1.00**	**1.00**	**0.00**	−0.06	0.01	+0.02	0.01	+0.17	**0.04**	**0.94**	0.15	**0.25**
fMRI	DiffusionTS	TR	**1.00**	**1.00**	**1.00**	**0.00**	+0.03	**0.00**	+0.30	**0.00**	+2.25	3.63	12.83	0.40	0.14
		GS	**1.00**	**1.00**	**1.00**	**0.00**	−0.55	1.00	+20.44	1.00	+8.32	0.74	8.31	0.50	0.10
		TimeWak	**1.00**	**1.00**	**1.00**	**0.00**	+0.09	0.13	+1.96	0.01	+0.60	0.45	1.87	0.25	0.10
	SDformer	LVQMark	0.24	0.29	0.24	**0.00**	+0.04	0.01	+0.43	**0.00**	+0.10	**0.13**	1.19	**0.12**	**0.09**
	L-VQVAE	LVQMark	**1.00**	**1.00**	**1.00**	**0.00**	−0.18	**0.00**	+0.09	**0.00**	−0.04	**0.13**	**1.14**	0.27	**0.09**

G Detailed Watermark Detection and Quality Statistics

Tables 30–32 complement the main results by reporting both the mean and the standard deviation of the detection and quality metrics. For detection, the reported standard deviations summarize the spread of sample-level z-scores under each setting, showing how tightly the detection statistics concentrate around their mean values. For quality, the reported standard deviations reflect variability across the same five-run protocol used in the main experiments. These additional statistics support the same qualitative conclusions as the mean-only results. In particular, L-VQVAE with LVQMark keeps

Table 26: Results of synthetic time series watermark detection and quality. Watermarked (TPR) and Non-watermarked (FPR, mean z-score) detections are evaluated under 5% attacks. Quality metrics are for 128-length. Best results are in bold, and second-best are underlined. z-scores with $|z| > 3.09$ are marked in red.

Setting			Watermark (TPR ↑)			Non-watermarked (FPR ↓ | z-score)						Quality Metric (↓)			
Dataset	Model	Method	Offset	Crop	Insert	Offset		Crop		Insert		C-FID	Corr.	Disc.	Pred.
Stocks	DiffusionTS	TR	0.00	**1.00**	**1.00**	**0.00**	+0.09	1.00	+66.42	1.00	+5.55	3.05	0.09	0.24	**0.04**
		GS	**1.00**	**1.00**	**1.00**	**0.00**	−1.32	**0.00**	−3.91	**0.00**	−0.47	2.63	0.04	0.19	**0.04**
		TimeWak	**1.00**	**1.00**	**1.00**	**0.00**	−0.17	1.00	+11.97	0.16	+2.05	0.34	**0.01**	0.15	**0.04**
	SDformer	LVQMark	0.00	0.00	0.00	**0.00**	−0.82	**0.00**	−0.22	**0.00**	−0.02	0.12	**0.01**	**0.03**	**0.04**
	L-VQVAE	LVQMark	**1.00**	**1.00**	**1.00**	**0.00**	+0.02	**0.00**	+0.16	0.01	+0.15	**0.08**	**0.01**	0.18	**0.04**
ETTh	DiffusionTS	TR	0.00	**1.00**	**1.00**	**0.00**	+0.28	1.00	+11.00	1.00	+11.88	2.52	0.26	0.30	0.13
		GS	**1.00**	**1.00**	**1.00**	**0.00**	−0.93	**0.00**	−13.63	**0.00**	−1.08	5.59	0.23	0.39	0.15
		TimeWak	**1.00**	**1.00**	**1.00**	**0.00**	+0.03	0.02	+0.94	**0.00**	+0.40	1.08	0.17	0.15	**0.11**
	SDformer	LVQMark	0.52	0.03	0.10	**0.00**	−0.07	0.03	+1.19	**0.00**	+0.53	**0.04**	**0.05**	0.03	**0.11**
	L-VQVAE	LVQMark	**1.00**	**1.00**	**1.00**	**0.00**	−0.10	**0.00**	+0.16	**0.00**	+0.04	**0.04**	**0.05**	**0.02**	**0.11**
Energy	DiffusionTS	TR	0.00	**1.00**	**1.00**	**0.00**	+0.04	1.00	+58.72	1.00	+37.10	0.50	1.65	0.49	0.28
		GS	**1.00**	**1.00**	**1.00**	**0.00**	+0.84	1.00	+32.87	1.00	+6.63	3.34	3.59	0.48	0.29
		TimeWak	**1.00**	**1.00**	**1.00**	**0.00**	−0.82	**0.00**	−1.89	**0.00**	+0.58	0.17	1.50	0.24	0.25
	SDformer	LVQMark	**1.00**	0.14	**1.00**	**0.00**	−0.06	**0.00**	−0.61	**0.00**	+0.28	0.04	**0.67**	**0.05**	0.25
	L-VQVAE	LVQMark	**1.00**	**1.00**	**1.00**	**0.00**	+0.15	**0.00**	−0.09	**0.00**	−0.01	**0.03**	0.74	0.19	**0.24**
fMRI	DiffusionTS	TR	0.00	0.00	**1.00**	**0.00**	+0.02	**0.00**	+1.37	1.00	+3.98	4.57	14.15	0.45	0.17
		GS	**1.00**	**1.00**	**1.00**	**0.00**	+0.04	**0.00**	−38.23	0.29	+2.54	1.05	5.97	0.50	0.11
		TimeWak	**1.00**	**1.00**	**1.00**	**0.00**	−0.08	**0.00**	−0.90	**0.00**	−0.47	0.81	1.81	0.36	0.10
	SDformer	LVQMark	0.10	0.12	0.11	**0.00**	+0.06	**0.00**	+0.02	**0.00**	−0.00	**0.09**	**0.84**	0.16	**0.08**
	L-VQVAE	LVQMark	**1.00**	**1.00**	**1.00**	**0.00**	−0.10	**0.00**	+0.25	**0.00**	−0.03	0.13	0.90	**0.06**	**0.08**

Table 27: Results of watermark detection under attack for 24-length sequences. LVQMark is applied to different variants, and watermark detection performance (TPR) is evaluated under 5% attacks. Best results are in bold, and second-best are underlined.

		Stocks			Energy			ETTh			fMRI		
Model	Type	Offset	Crop	Insert	Offset	Crop	Insert	Offset	Crop	Insert	Offset	Crop	Insert
SDformer	w/o Robust	0.19	0.44	0.72	**1.00**	**1.00**	**1.00**	**1.00**	0.01	**1.00**	**1.00**	**1.00**	**1.00**
	w/ Robust	0.08	0.57	0.24	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
L-VQVAE	w/o Robust	**1.00**	0.78	0.86	**1.00**	0.76	**1.00**	**1.00**	0.90	**1.00**	**1.00**	**1.00**	**1.00**
	w/ Robust	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**

Table 28: Results of watermark detection under attack for 64-length sequences. LVQMark is applied to different variants, and watermark detection performance (TPR) is evaluated under 5% attacks. Best results are in bold, and second-best are underlined.

		Stocks			Energy			ETTh			fMRI		
Model	Type	Offset	Crop	Insert	Offset	Crop	Insert	Offset	Crop	Insert	Offset	Crop	Insert
SDformer	w/o Robust	0.01	0.03	0.01	**1.00**	0.46	**1.00**	0.27	0.00	0.06	0.24	0.29	0.24
	w/ Robust	0.00	0.00	0.00	**1.00**	0.95	**1.00**	0.14	0.09	0.10	0.04	0.12	0.10
L-VQVAE	w/o Robust	**1.00**	0.00	**1.00**	**1.00**	0.03	**1.00**	**1.00**	0.93	**1.00**	**1.00**	**1.00**	**1.00**
	w/ Robust	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**

Table 29: Results of watermark detection under attack for 128-length sequences. LVQMark is applied to different variants, and watermark detection performance (TPR) is evaluated under 5% attacks. Best results are in bold, and second-best are underlined.

		Stocks			Energy			ETTh			fMRI		
Model	Type	Offset	Crop	Insert	Offset	Crop	Insert	Offset	Crop	Insert	Offset	Crop	Insert
SDformer	w/o Robust	0.00	0.00	0.00	**1.00**	0.14	**1.00**	0.52	0.03	0.10	0.10	0.12	0.11
	w/ Robust	0.00	0.00	0.00	**1.00**	0.85	**1.00**	0.34	0.31	0.35	0.06	0.04	0.02
L-VQVAE	w/o Robust	**1.00**	0.94	**1.00**	**1.00**	0.02	**1.00**	**1.00**	0.50	**1.00**	**1.00**	**1.00**	**1.00**
	w/ Robust	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**

the non-watermarked z-scores centered near the intended null region across attacks and sequence lengths, while the corresponding standard deviations remain moderate, indicating stable calibration at the sample level rather than a mean obtained from offsetting extreme cases. At the same time, the watermark z-scores remain clearly positive and well separated from zero, showing that strong detectability is preserved together with null stability. By contrast, several baselines continue to exhibit large mean shifts on attacked non-watermarked samples, and these shifts are typically much larger in magnitude than their corresponding standard deviations, indicating that the observed null drift is systematic rather than attributable to a small subset of atypical samples. Taken together, these results reinforce the main finding of the paper: the advantage of L-VQVAE with LVQMark lies not only in achieving favorable mean performance, but also in maintaining a more stable and interpretable detection behavior under attack.

Table 30: Results of synthetic time series watermark detection and quality. Watermark and Non-watermarked detections (z-score) are evaluated under 30% attacks. Detection and quality metrics (mean $\pm$ std) are for 24-length.

Setting			Watermark (z-score ↑)			Non-watermarked (z-score)			Quality Metric (↓)			
Dataset	**Model**	**Method**	**Offset**	**Crop**	**Insert**	**Offset**	**Crop**	**Insert**	**C-FID**	**Corr.**	**Disc.**	**Pred.**
Stocks	DiffusionTS	TR	$0.59_{\pm 0.03}$	$51.54_{\pm 0.66}$	$0.22_{\pm 0.04}$	$0.19_{\pm 0.03}$	$63.26_{\pm 0.69}$	$1.91_{\pm 0.09}$	$0.96_{\pm 0.15}$	$0.10_{\pm 0.01}$	$0.20_{\pm 0.02}$	$0.04_{\pm 0.00}$
		GS	$85.77_{\pm 0.80}$	$3.11_{\pm 0.67}$	$52.58_{\pm 0.80}$	$-1.15_{\pm 1.22}$	$-6.32_{\pm 0.69}$	$0.16_{\pm 0.81}$	$8.88_{\pm 0.62}$	$0.09_{\pm 0.01}$	$0.43_{\pm 0.01}$	$0.04_{\pm 0.00}$
		TimeWak	$180.18_{\pm 0.86}$	$11.24_{\pm 0.94}$	$56.48_{\pm 1.13}$	$-0.99_{\pm 0.99}$	$1.39_{\pm 0.86}$	$-1.77_{\pm 1.03}$	$0.33_{\pm 0.03}$	$0.02_{\pm 0.01}$	$0.16_{\pm 0.02}$	$0.04_{\pm 0.00}$
	SDformer	LVQMark	$2.14_{\pm 0.96}$	$4.21_{\pm 1.19}$	$2.39_{\pm 0.94}$	$0.51_{\pm 0.91}$	$4.49_{\pm 1.28}$	$1.33_{\pm 0.82}$	$0.11_{\pm 0.02}$	$0.01_{\pm 0.00}$	$0.02_{\pm 0.02}$	$0.04_{\pm 0.00}$
	L-VQVAE	LVQMark	$11.16_{\pm 1.12}$	$6.97_{\pm 1.40}$	$5.45_{\pm 0.88}$	$-0.25_{\pm 0.85}$	$1.10_{\pm 1.21}$	$0.38_{\pm 1.18}$	$0.07_{\pm 0.01}$	$0.01_{\pm 0.01}$	$0.11_{\pm 0.09}$	$0.04_{\pm 0.00}$
ETTh	DiffusionTS	TR	$3.70_{\pm 0.10}$	$11.62_{\pm 0.25}$	$18.23_{\pm 0.32}$	$0.95_{\pm 0.04}$	$10.84_{\pm 0.22}$	$15.97_{\pm 0.28}$	$1.56_{\pm 0.16}$	$0.18_{\pm 0.02}$	$0.27_{\pm 0.01}$	$0.14_{\pm 0.00}$
		GS	$53.08_{\pm 1.14}$	$23.49_{\pm 1.00}$	$36.63_{\pm 1.00}$	$12.01_{\pm 0.79}$	$18.58_{\pm 1.11}$	$6.96_{\pm 0.94}$	$4.66_{\pm 0.47}$	$0.42_{\pm 0.02}$	$0.38_{\pm 0.02}$	$0.18_{\pm 0.01}$
		TimeWak	$115.86_{\pm 0.96}$	$3.07_{\pm 0.97}$	$19.44_{\pm 1.06}$	$-0.34_{\pm 0.97}$	$1.05_{\pm 0.96}$	$-1.68_{\pm 1.00}$	$0.23_{\pm 0.01}$	$0.21_{\pm 0.04}$	$0.08_{\pm 0.03}$	$0.12_{\pm 0.00}$
	SDformer	LVQMark	$5.15_{\pm 0.94}$	$0.50_{\pm 1.26}$	$1.82_{\pm 0.89}$	$0.79_{\pm 0.96}$	$0.18_{\pm 1.22}$	$0.70_{\pm 0.85}$	$0.13_{\pm 0.02}$	$0.08_{\pm 0.03}$	$0.04_{\pm 0.02}$	$0.12_{\pm 0.00}$
	L-VQVAE	LVQMark	$20.66_{\pm 1.12}$	$17.01_{\pm 1.24}$	$13.48_{\pm 0.95}$	$0.03_{\pm 1.10}$	$0.41_{\pm 1.45}$	$-0.15_{\pm 1.01}$	$0.05_{\pm 0.00}$	$0.05_{\pm 0.01}$	$0.02_{\pm 0.01}$	$0.12_{\pm 0.00}$
Energy	DiffusionTS	TR	$0.84_{\pm 0.03}$	$37.42_{\pm 0.25}$	$26.12_{\pm 0.16}$	$0.43_{\pm 0.03}$	$39.25_{\pm 0.32}$	$26.77_{\pm 0.16}$	$0.43_{\pm 0.03}$	$2.66_{\pm 0.30}$	$0.41_{\pm 0.01}$	$0.30_{\pm 0.00}$
		GS	$53.58_{\pm 1.00}$	$75.56_{\pm 1.24}$	$49.04_{\pm 1.07}$	$13.17_{\pm 1.19}$	$65.49_{\pm 1.36}$	$14.31_{\pm 1.14}$	$1.58_{\pm 0.06}$	$3.37_{\pm 0.13}$	$0.49_{\pm 0.00}$	$0.33_{\pm 0.00}$
		TimeWak	$192.68_{\pm 1.64}$	$10.87_{\pm 0.89}$	$55.08_{\pm 1.08}$	$0.33_{\pm 0.94}$	$6.03_{\pm 1.07}$	$-1.22_{\pm 1.01}$	$0.09_{\pm 0.01}$	$1.53_{\pm 0.19}$	$0.14_{\pm 0.02}$	$0.25_{\pm 0.00}$
	SDformer	LVQMark	$24.84_{\pm 0.84}$	$3.99_{\pm 1.27}$	$14.32_{\pm 0.90}$	$0.29_{\pm 1.02}$	$1.48_{\pm 0.99}$	$0.32_{\pm 0.95}$	$0.12_{\pm 0.00}$	$1.27_{\pm 0.13}$	$0.23_{\pm 0.01}$	$0.25_{\pm 0.00}$
	L-VQVAE	LVQMark	$28.18_{\pm 1.09}$	$21.94_{\pm 1.19}$	$21.55_{\pm 1.00}$	$-0.16_{\pm 0.97}$	$-0.19_{\pm 1.21}$	$-0.03_{\pm 0.99}$	$0.03_{\pm 0.00}$	$1.02_{\pm 0.10}$	$0.09_{\pm 0.01}$	$0.25_{\pm 0.00}$
fMRI	DiffusionTS	TR	$2.99_{\pm 0.06}$	$2.47_{\pm 0.05}$	$1.74_{\pm 0.07}$	$0.42_{\pm 0.03}$	$0.55_{\pm 0.04}$	$4.89_{\pm 0.04}$	$2.26_{\pm 0.10}$	$13.42_{\pm 0.14}$	$0.50_{\pm 0.00}$	$0.15_{\pm 0.00}$
		GS	$418.07_{\pm 1.19}$	$173.30_{\pm 0.84}$	$262.45_{\pm 1.32}$	$2.33_{\pm 0.88}$	$-3.89_{\pm 0.87}$	$-6.16_{\pm 1.01}$	$0.71_{\pm 0.05}$	$15.21_{\pm 0.06}$	$0.50_{\pm 0.00}$	$0.11_{\pm 0.00}$
		TimeWak	$380.87_{\pm 0.92}$	$77.01_{\pm 0.92}$	$134.32_{\pm 0.87}$	$0.19_{\pm 1.07}$	$2.39_{\pm 0.91}$	$0.26_{\pm 1.04}$	$0.18_{\pm 0.01}$	$1.98_{\pm 0.07}$	$0.08_{\pm 0.02}$	$0.10_{\pm 0.00}$
	SDformer	LVQMark	$22.63_{\pm 0.95}$	$6.01_{\pm 1.23}$	$9.43_{\pm 0.97}$	$0.14_{\pm 0.98}$	$-0.41_{\pm 1.22}$	$0.01_{\pm 0.91}$	$0.94_{\pm 0.07}$	$3.24_{\pm 0.15}$	$0.21_{\pm 0.03}$	$0.10_{\pm 0.00}$
	L-VQVAE	LVQMark	$20.28_{\pm 0.91}$	$24.23_{\pm 1.32}$	$17.16_{\pm 0.96}$	$0.11_{\pm 0.91}$	$-0.23_{\pm 1.32}$	$-0.05_{\pm 1.02}$	$0.20_{\pm 0.01}$	$1.92_{\pm 0.03}$	$0.23_{\pm 0.01}$	$0.10_{\pm 0.00}$

Table 31: Results of synthetic time series watermark detection and quality. Watermark and Non-watermarked detections (z-score) are evaluated under 30% attacks. Detection and quality metrics (mean $\pm$ std) are for 64-length.

Setting			Watermark (z-score ↑)			Non-watermarked (z-score)			Quality Metric (↓)			
Dataset	**Model**	**Method**	**Offset**	**Crop**	**Insert**	**Offset**	**Crop**	**Insert**	**C-FID**	**Corr.**	**Disc.**	**Pred.**
Stocks	DiffusionTS	TR	$0.88_{\pm 0.03}$	$56.36_{\pm 0.84}$	$4.60_{\pm 0.17}$	$0.21_{\pm 0.03}$	$69.43_{\pm 0.77}$	$8.11_{\pm 0.21}$	$1.52_{\pm 0.11}$	$0.07_{\pm 0.02}$	$0.15_{\pm 0.05}$	$0.04_{\pm 0.00}$
		GS	$143.09_{\pm 1.43}$	$-14.28_{\pm 0.94}$	$64.00_{\pm 0.80}$	$-1.40_{\pm 0.78}$	$-14.87_{\pm 0.92}$	$1.52_{\pm 1.01}$	$1.50_{\pm 0.32}$	$0.02_{\pm 0.01}$	$0.22_{\pm 0.03}$	$0.04_{\pm 0.00}$
		TimeWak	$335.49_{\pm 0.81}$	$16.23_{\pm 0.92}$	$79.44_{\pm 1.28}$	$-0.19_{\pm 0.95}$	$-4.32_{\pm 0.91}$	$1.35_{\pm 0.91}$	$0.29_{\pm 0.03}$	$0.01_{\pm 0.00}$	$0.13_{\pm 0.03}$	$0.04_{\pm 0.00}$
	SDformer	LVQMark	$0.59_{\pm 0.81}$	$0.99_{\pm 0.87}$	$0.42_{\pm 0.93}$	$-0.16_{\pm 0.89}$	$1.19_{\pm 1.05}$	$0.31_{\pm 1.04}$	$0.08_{\pm 0.01}$	$0.01_{\pm 0.00}$	$0.01_{\pm 0.01}$	$0.04_{\pm 0.00}$
	L-VQVAE	LVQMark	$15.21_{\pm 1.05}$	$28.43_{\pm 1.33}$	$8.23_{\pm 0.85}$	$-0.12_{\pm 0.95}$	$-0.05_{\pm 1.28}$	$0.32_{\pm 1.06}$	$0.07_{\pm 0.01}$	$0.01_{\pm 0.01}$	$0.06_{\pm 0.02}$	$0.04_{\pm 0.00}$
ETTh	DiffusionTS	TR	$4.69_{\pm 0.09}$	$17.54_{\pm 0.29}$	$39.34_{\pm 0.52}$	$0.89_{\pm 0.04}$	$17.16_{\pm 0.27}$	$38.55_{\pm 0.60}$	$2.17_{\pm 0.10}$	$0.22_{\pm 0.01}$	$0.29_{\pm 0.01}$	$0.14_{\pm 0.00}$
		GS	$160.44_{\pm 2.50}$	$56.44_{\pm 1.67}$	$127.53_{\pm 1.28}$	$-1.53_{\pm 0.85}$	$-5.44_{\pm 1.43}$	$-5.82_{\pm 1.42}$	$3.43_{\pm 0.24}$	$0.25_{\pm 0.02}$	$0.36_{\pm 0.01}$	$0.16_{\pm 0.00}$
		TimeWak	$194.83_{\pm 1.53}$	$3.42_{\pm 0.81}$	$25.92_{\pm 1.13}$	$1.17_{\pm 1.00}$	$-0.28_{\pm 0.85}$	$1.20_{\pm 0.88}$	$0.37_{\pm 0.02}$	$0.13_{\pm 0.01}$	$0.11_{\pm 0.00}$	$0.12_{\pm 0.00}$
	SDformer	LVQMark	$-0.16_{\pm 0.92}$	$0.03_{\pm 1.16}$	$0.07_{\pm 0.97}$	$-0.94_{\pm 0.83}$	$0.09_{\pm 1.17}$	$-0.09_{\pm 0.98}$	$0.04_{\pm 0.00}$	$0.05_{\pm 0.02}$	$0.00_{\pm 0.01}$	$0.12_{\pm 0.01}$
	L-VQVAE	LVQMark	$16.81_{\pm 0.99}$	$17.48_{\pm 1.13}$	$14.46_{\pm 0.97}$	$-0.12_{\pm 1.06}$	$-0.22_{\pm 1.08}$	$0.35_{\pm 0.84}$	$0.03_{\pm 0.00}$	$0.06_{\pm 0.03}$	$0.01_{\pm 0.01}$	$0.12_{\pm 0.01}$
Energy	DiffusionTS	TR	$2.14_{\pm 0.04}$	$48.74_{\pm 0.34}$	$58.22_{\pm 0.28}$	$1.18_{\pm 0.04}$	$53.73_{\pm 0.38}$	$58.55_{\pm 0.29}$	$0.58_{\pm 0.04}$	$1.98_{\pm 0.09}$	$0.43_{\pm 0.02}$	$0.28_{\pm 0.00}$
		GS	$31.04_{\pm 0.88}$	$2.78_{\pm 1.06}$	$37.14_{\pm 1.38}$	$14.51_{\pm 0.86}$	$0.87_{\pm 1.08}$	$11.77_{\pm 1.18}$	$1.78_{\pm 0.14}$	$2.72_{\pm 0.15}$	$0.48_{\pm 0.01}$	$0.31_{\pm 0.00}$
		TimeWak	$167.49_{\pm 2.09}$	$5.02_{\pm 0.93}$	$28.04_{\pm 1.01}$	$-9.69_{\pm 0.82}$	$-0.42_{\pm 0.93}$	$-1.57_{\pm 0.98}$	$0.14_{\pm 0.01}$	$1.52_{\pm 0.24}$	$0.14_{\pm 0.01}$	$0.25_{\pm 0.00}$
	SDformer	LVQMark	$7.41_{\pm 0.98}$	$1.14_{\pm 0.87}$	$3.29_{\pm 0.88}$	$-0.02_{\pm 1.10}$	$0.34_{\pm 0.87}$	$0.34_{\pm 0.90}$	$0.04_{\pm 0.00}$	$1.04_{\pm 0.15}$	$0.08_{\pm 0.02}$	$0.25_{\pm 0.00}$
	L-VQVAE	LVQMark	$15.24_{\pm 1.16}$	$13.22_{\pm 1.12}$	$11.31_{\pm 1.02}$	$-0.06_{\pm 1.09}$	$-0.25_{\pm 1.14}$	$0.04_{\pm 1.07}$	$0.04_{\pm 0.00}$	$0.94_{\pm 0.34}$	$0.15_{\pm 0.01}$	$0.25_{\pm 0.00}$
fMRI	DiffusionTS	TR	$5.25_{\pm 0.05}$	$3.36_{\pm 0.06}$	$20.02_{\pm 0.11}$	$0.05_{\pm 0.03}$	$2.27_{\pm 0.04}$	$12.14_{\pm 0.05}$	$3.63_{\pm 0.41}$	$12.83_{\pm 0.08}$	$0.40_{\pm 0.17}$	$0.14_{\pm 0.00}$
		GS	$543.72_{\pm 1.11}$	$275.99_{\pm 1.72}$	$421.92_{\pm 3.12}$	$-1.78_{\pm 0.91}$	$-38.77_{\pm 1.70}$	$-0.01_{\pm 2.12}$	$0.74_{\pm 0.05}$	$8.31_{\pm 0.03}$	$0.50_{\pm 0.00}$	$0.10_{\pm 0.00}$
		TimeWak	$574.43_{\pm 0.98}$	$84.45_{\pm 0.98}$	$185.86_{\pm 0.95}$	$0.08_{\pm 0.97}$	$-2.74_{\pm 0.99}$	$0.18_{\pm 0.90}$	$0.45_{\pm 0.01}$	$1.87_{\pm 0.05}$	$0.25_{\pm 0.11}$	$0.10_{\pm 0.00}$
	SDformer	LVQMark	$2.39_{\pm 0.97}$	$1.13_{\pm 1.09}$	$1.48_{\pm 0.95}$	$0.03_{\pm 0.86}$	$0.51_{\pm 1.16}$	$-0.40_{\pm 0.85}$	$0.13_{\pm 0.01}$	$1.19_{\pm 0.04}$	$0.12_{\pm 0.02}$	$0.09_{\pm 0.00}$
	L-VQVAE	LVQMark	$12.90_{\pm 0.88}$	$15.19_{\pm 1.23}$	$10.45_{\pm 0.97}$	$-0.03_{\pm 0.99}$	$-0.24_{\pm 1.15}$	$-0.17_{\pm 0.99}$	$0.13_{\pm 0.00}$	$1.14_{\pm 0.06}$	$0.27_{\pm 0.11}$	$0.09_{\pm 0.00}$

H Visualization of Generation Results

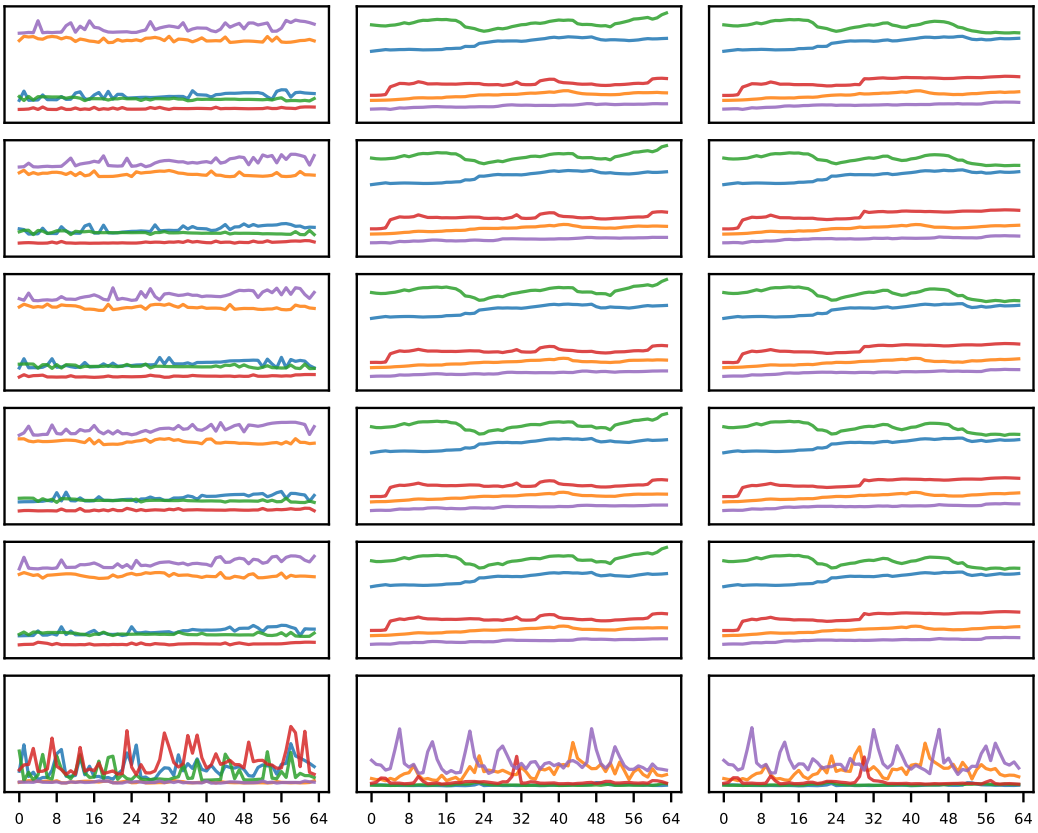
We provide qualitative visualizations of generation-time watermarked time series data on Stocks, ETTh, Energy, and fMRI for 64-length sequences. Figures 5-8 compare original samples with

Table 32: Results of synthetic time series watermark detection and quality. Watermark and Non-watermarked detections (z-score) are evaluated under 30% attacks. Detection and quality metrics (mean $\pm$ std) are for 128-length.

Setting			Watermark (z-score ↑)			Non-watermarked (z-score)			Quality Metric (↓)			
Dataset	**Model**	**Method**	**Offset**	**Crop**	**Insert**	**Offset**	**Crop**	**Insert**	**C-FID**	**Corr.**	**Disc.**	**Pred.**
Stocks	DiffusionTS	TR	$0.96_{\pm 0.03}$	$60.85_{\pm 0.68}$	$19.13_{\pm 0.39}$	$0.30_{\pm 0.03}$	$84.66_{\pm 0.65}$	$20.60_{\pm 0.43}$	$3.05_{\pm 0.78}$	$0.09_{\pm 0.01}$	$0.24_{\pm 0.05}$	$0.04_{\pm 0.00}$
		GS	$158.85_{\pm 1.94}$	$1.08_{\pm 0.95}$	$56.98_{\pm 0.95}$	$-4.40_{\pm 1.00}$	$-10.04_{\pm 0.81}$	$-5.22_{\pm 1.01}$	$2.63_{\pm 0.16}$	$0.04_{\pm 0.02}$	$0.19_{\pm 0.03}$	$0.04_{\pm 0.00}$
		TimeWak	$466.00_{\pm 1.19}$	$10.37_{\pm 1.01}$	$76.52_{\pm 1.29}$	$-0.16_{\pm 0.90}$	$-4.18_{\pm 0.93}$	$3.74_{\pm 1.03}$	$0.34_{\pm 0.08}$	$0.01_{\pm 0.00}$	$0.15_{\pm 0.04}$	$0.04_{\pm 0.00}$
	SDformer	LVQMark	$-0.57_{\pm 0.94}$	$-0.77_{\pm 0.66}$	$-0.78_{\pm 0.95}$	$-0.54_{\pm 0.87}$	$-0.89_{\pm 0.98}$	$-1.20_{\pm 1.10}$	$0.12_{\pm 0.01}$	$0.01_{\pm 0.01}$	$0.03_{\pm 0.03}$	$0.04_{\pm 0.00}$
	L-VQVAE	LVQMark	$19.79_{\pm 1.51}$	$13.17_{\pm 1.64}$	$13.82_{\pm 1.20}$	$0.09_{\pm 0.92}$	$-0.09_{\pm 1.24}$	$0.23_{\pm 0.96}$	$0.08_{\pm 0.01}$	$0.01_{\pm 0.01}$	$0.18_{\pm 0.12}$	$0.04_{\pm 0.00}$
ETTh	DiffusionTS	TR	$0.61_{\pm 0.04}$	$10.99_{\pm 0.23}$	$35.91_{\pm 0.46}$	$1.52_{\pm 0.04}$	$12.19_{\pm 0.23}$	$36.77_{\pm 0.38}$	$2.52_{\pm 0.12}$	$0.26_{\pm 0.01}$	$0.30_{\pm 0.01}$	$0.13_{\pm 0.00}$
		GS	$229.36_{\pm 2.44}$	$51.50_{\pm 1.15}$	$155.83_{\pm 1.52}$	$-6.09_{\pm 0.98}$	$-13.71_{\pm 0.85}$	$-6.37_{\pm 1.18}$	$5.59_{\pm 0.36}$	$0.23_{\pm 0.01}$	$0.39_{\pm 0.00}$	$0.15_{\pm 0.00}$
		TimeWak	$257.53_{\pm 2.12}$	$4.62_{\pm 1.03}$	$40.33_{\pm 0.98}$	$1.09_{\pm 0.91}$	$-1.63_{\pm 0.86}$	$0.77_{\pm 0.97}$	$1.08_{\pm 0.08}$	$0.17_{\pm 0.02}$	$0.15_{\pm 0.01}$	$0.11_{\pm 0.00}$
	SDformer	LVQMark	$2.01_{\pm 1.01}$	$1.12_{\pm 1.15}$	$1.43_{\pm 0.85}$	$0.53_{\pm 0.85}$	$0.76_{\pm 1.11}$	$1.01_{\pm 0.87}$	$0.04_{\pm 0.00}$	$0.05_{\pm 0.03}$	$0.03_{\pm 0.01}$	$0.11_{\pm 0.01}$
	L-VQVAE	LVQMark	$13.82_{\pm 0.93}$	$13.76_{\pm 1.07}$	$11.98_{\pm 1.02}$	$-0.04_{\pm 0.81}$	$-0.05_{\pm 1.09}$	$-0.30_{\pm 0.83}$	$0.04_{\pm 0.00}$	$0.05_{\pm 0.01}$	$0.02_{\pm 0.01}$	$0.11_{\pm 0.01}$
Energy	DiffusionTS	TR	$2.34_{\pm 0.04}$	$56.95_{\pm 0.34}$	$95.70_{\pm 0.35}$	$2.20_{\pm 0.03}$	$61.47_{\pm 0.39}$	$98.65_{\pm 0.37}$	$0.50_{\pm 0.05}$	$1.65_{\pm 0.19}$	$0.49_{\pm 0.00}$	$0.28_{\pm 0.00}$
		GS	$56.04_{\pm 0.78}$	$61.55_{\pm 1.05}$	$65.92_{\pm 1.13}$	$15.43_{\pm 1.09}$	$48.30_{\pm 0.98}$	$27.89_{\pm 1.32}$	$3.34_{\pm 0.25}$	$3.59_{\pm 0.12}$	$0.48_{\pm 0.01}$	$0.29_{\pm 0.01}$
		TimeWak	$172.41_{\pm 1.57}$	$1.82_{\pm 0.90}$	$18.91_{\pm 0.96}$	$-1.57_{\pm 0.89}$	$-0.78_{\pm 0.88}$	$0.66_{\pm 0.86}$	$0.17_{\pm 0.01}$	$1.50_{\pm 0.25}$	$0.24_{\pm 0.10}$	$0.25_{\pm 0.00}$
	SDformer	LVQMark	$7.48_{\pm 1.03}$	$-0.29_{\pm 0.86}$	$2.74_{\pm 0.89}$	$0.06_{\pm 0.96}$	$-0.65_{\pm 0.69}$	$-0.08_{\pm 0.90}$	$0.04_{\pm 0.00}$	$0.67_{\pm 0.19}$	$0.05_{\pm 0.01}$	$0.25_{\pm 0.00}$
	L-VQVAE	LVQMark	$8.48_{\pm 1.01}$	$6.16_{\pm 1.26}$	$6.57_{\pm 1.08}$	$-0.03_{\pm 1.02}$	$0.00_{\pm 1.26}$	$0.13_{\pm 0.82}$	$0.03_{\pm 0.00}$	$0.74_{\pm 0.22}$	$0.19_{\pm 0.02}$	$0.24_{\pm 0.00}$
fMRI	DiffusionTS	TR	$0.22_{\pm 0.04}$	$4.73_{\pm 0.05}$	$25.33_{\pm 0.12}$	$0.02_{\pm 0.02}$	$6.33_{\pm 0.04}$	$21.25_{\pm 0.06}$	$4.57_{\pm 0.51}$	$14.15_{\pm 0.05}$	$0.45_{\pm 0.09}$	$0.17_{\pm 0.00}$
		GS	$801.62_{\pm 0.81}$	$148.15_{\pm 1.57}$	$499.87_{\pm 2.71}$	$-0.70_{\pm 0.97}$	$-57.42_{\pm 0.88}$	$0.52_{\pm 1.78}$	$1.05_{\pm 0.06}$	$5.97_{\pm 0.05}$	$0.50_{\pm 0.00}$	$0.11_{\pm 0.00}$
		TimeWak	$739.70_{\pm 1.24}$	$97.66_{\pm 0.98}$	$204.66_{\pm 0.96}$	$-0.07_{\pm 0.96}$	$-3.03_{\pm 0.95}$	$-1.95_{\pm 1.00}$	$0.81_{\pm 0.04}$	$1.81_{\pm 0.03}$	$0.36_{\pm 0.10}$	$0.10_{\pm 0.00}$
	SDformer	LVQMark	$1.95_{\pm 0.94}$	$0.73_{\pm 1.15}$	$0.93_{\pm 0.90}$	$-0.03_{\pm 0.88}$	$-0.25_{\pm 1.11}$	$-0.15_{\pm 0.86}$	$0.09_{\pm 0.01}$	$0.84_{\pm 0.03}$	$0.16_{\pm 0.03}$	$0.08_{\pm 0.00}$
	L-VQVAE	LVQMark	$10.47_{\pm 0.97}$	$10.75_{\pm 1.25}$	$8.05_{\pm 0.96}$	$-0.09_{\pm 0.87}$	$0.16_{\pm 1.24}$	$-0.01_{\pm 0.96}$	$0.13_{\pm 0.01}$	$0.90_{\pm 0.02}$	$0.06_{\pm 0.05}$	$0.08_{\pm 0.00}$

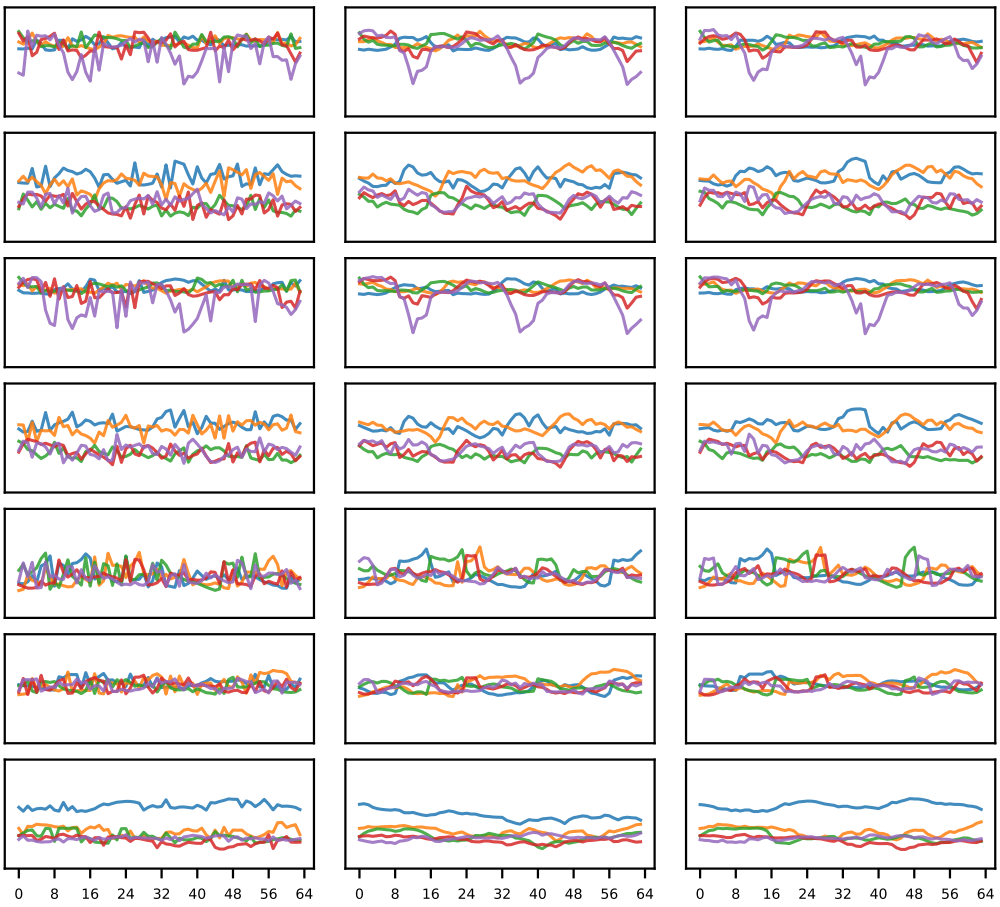
LVQMark generations. The left column shows original samples, while the middle and right columns show watermarked generations produced with LVQMark and non-watermarked generations produced without watermarking, respectively.

Each row corresponds to a different channel, with five representative samples shown in each column. The generated samples capture dataset-specific temporal patterns, and the watermarked samples remain qualitatively similar to the non-watermarked ones, suggesting that LVQMark does not introduce noticeable qualitative degradation. This observation is also consistent with the low Context-FID values reported in Table 2.



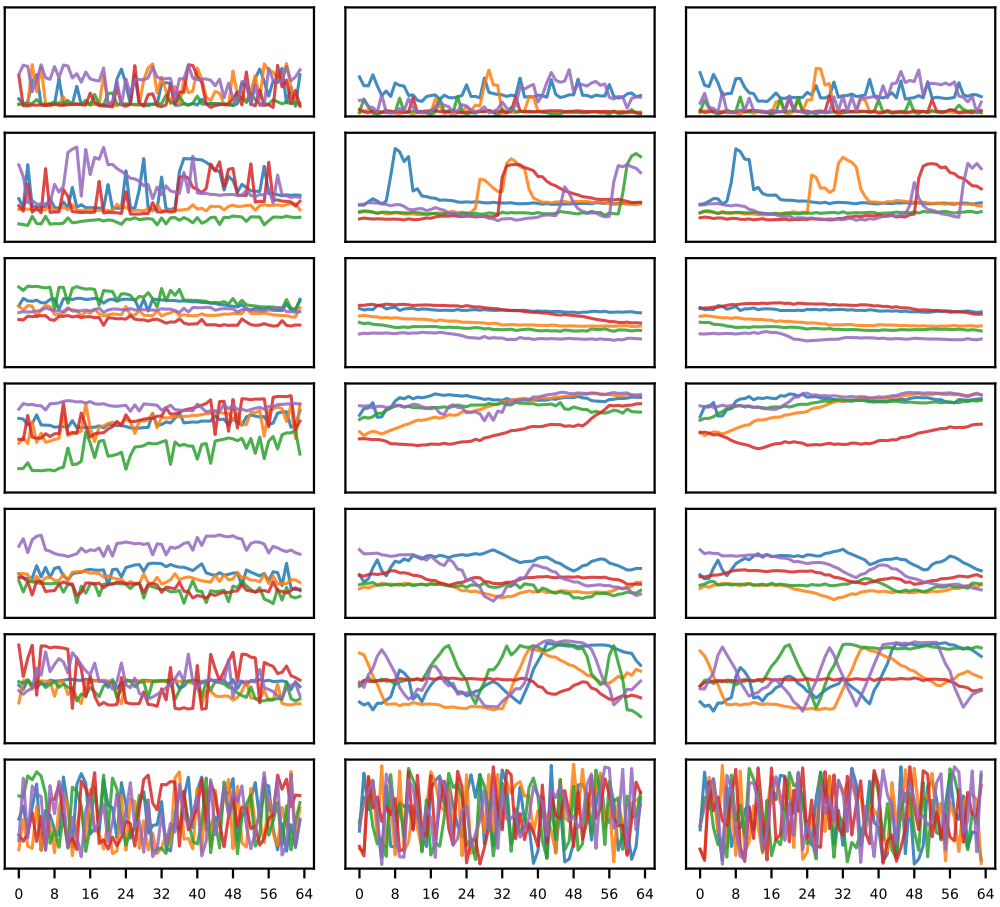
(a) Original (b) Watermarked (c) Non-watermarked

Figure 5: Visualization of original and generated samples on the Stocks dataset with length 64.



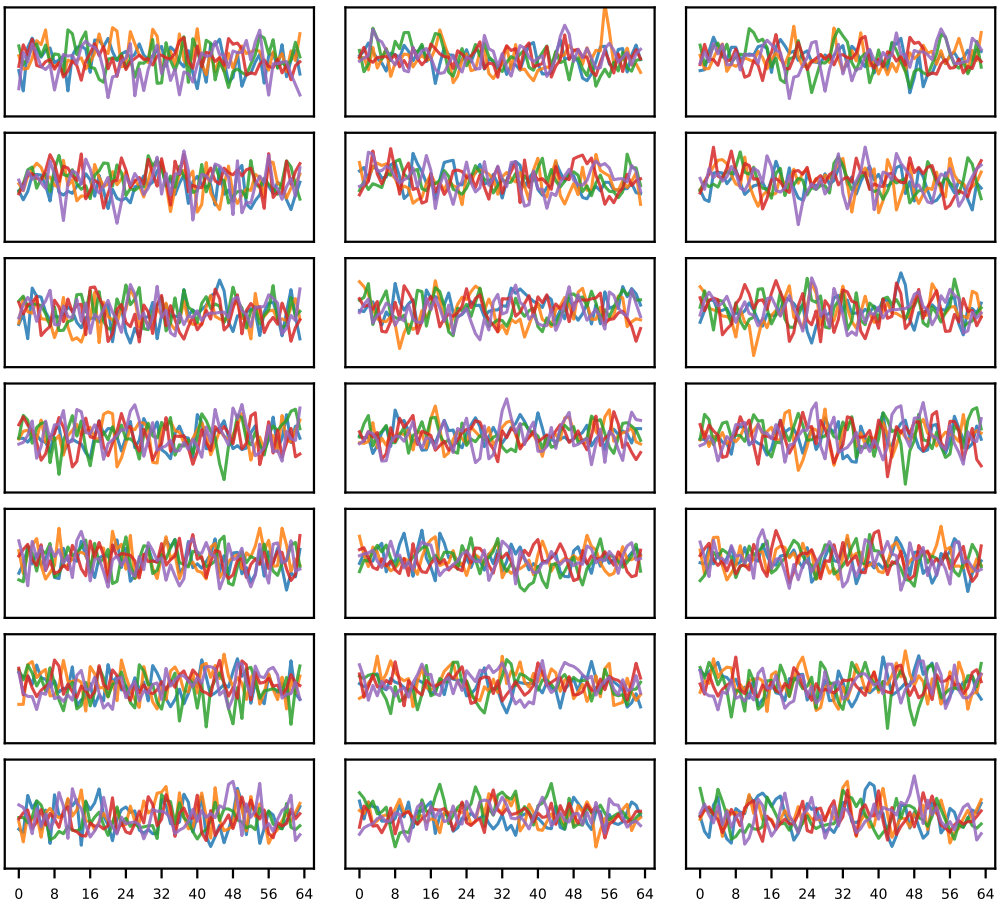
(a) Original (b) Watermarked (c) Non-watermarked

Figure 6: Visualization of original and generated samples on the ETTh dataset with length 64.



(a) Original (b) Watermarked (c) Non-watermarked

Figure 7: Visualization of original and generated samples on the Energy dataset with length 64.



(a) Original (b) Watermarked (c) Non-watermarked

Figure 8: Visualization of original and generated samples on the fMRI dataset with length 64.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The claims in the abstract and introduction accurately reflect the paper's contributions. The proposed L-VQVAE and LVQMark are described with their scope and assumptions, and the experimental results in Sections 3–4 support the stated claims.

   Guidelines:

   - The answer [N/A] 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 [N/A] 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: [Yes]

   Justification: Section 5 contains an explicit Limitations paragraph discussing the restricted number of datasets, sequence lengths, attack types, and the increased complexity of the multi-stage training pipeline compared to single-stage baselines.

   Guidelines:

   - The answer [N/A] 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: [N/A]

Justification: The paper does not include formal theorems or proofs; it is an empirical contribution.

Guidelines:

- The answer [N/A] 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: [Yes]

Justification: Appendix B provides full details on datasets, preprocessing, hyperparameters (Tables 9–10), attack configurations, evaluation metrics, and cost measurement (Appendix B.6). All results are averaged over five runs with fixed random seeds.

Guidelines:

- The answer [N/A] 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: [Yes]

   Justification: The code is publicly available at `https://github.com/dongbeank/LVQMark`. All datasets used are publicly available.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.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://neurips.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) necessary to understand the results?

   Answer: [Yes]

   Justification: Appendix B specifies data splits (80%/20%), optimizer, batch sizes, learning rates, training steps per stage, and attack parameters in full detail.

   Guidelines:

   - The answer [N/A] 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: [Yes]

   Justification: All reported results are averaged over five runs with fixed random seeds. Mean and standard deviation of all detection and quality metrics are reported in Appendix G (Tables across sequence lengths 24, 64, and 128).

   Guidelines:

   - The answer [N/A] 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: [Yes]

   Justification: Compute resources are described in Appendix B.5, including GPU models (NVIDIA H100 PCIe 80 GB, RTX PRO 6000 Blackwell 96 GB $\times 2$) and CPU. Per-stage training steps are provided in Tables 9–10, and the training time, generation throughput, and verification cost of LVQMark and TimeWak are measured in Appendix B.6 on a single RTX PRO 6000.

   Guidelines:

   - The answer [N/A] 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: [Yes]

   Justification: The research uses only publicly available datasets, involves no human subjects or sensitive data collection, and conforms to the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] 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: [Yes]

Justification: The proposed method supports provenance verification of synthetic time series, which has a positive societal impact in combating misattribution of AI-generated content. A potential negative impact is that advances in watermarking robustness may motivate stronger evasion attacks; this is acknowledged in Section 4.5.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] 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., pre-trained language models, image generators, or scraped datasets)?

Answer: [Yes]

Justification: The paper presents a detection framework and does not release high-risk pretrained generative models or scraped datasets that would require additional safeguards.

Guidelines:

- The answer [N/A] 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: [Yes]

Justification: All datasets (Stocks, ETTh, Energy, fMRI) and baseline methods (DiffusionTS, TimeVQVAE, SDformer, TimeWak, KGW, Tree-Ring, Gaussian Shading) are cited with their original papers. Dataset sources are listed in Table 11.

Guidelines:

- The answer [N/A] 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: [N/A]

Justification: The paper does not release new public datasets or pretrained model checkpoints.

Guidelines:

- The answer [N/A] 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: [N/A]

Justification: The paper does not involve crowdsourcing or research with human subjects.

Guidelines:

- The answer [N/A] 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: [N/A]

Justification: No human subjects research is conducted, so IRB approval is not applicable.

Guidelines:

- The answer [N/A] 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.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: LLMs are not used as an important, original, or non-standard component of the core methodology in this work.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.