# PHASELOCK: PHASE-STRUCTURED TOKENIZATION FOR WATERMARKING PERIODIC TIME SERIES

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Generation-time watermarks for time series bias a discrete token sequence toward a secret set of codes and detect the watermark by re-encoding the observed signal. Existing methods tokenize windows starting at arbitrary positions and treat the codebook as an unstructured set, so patches at different phases of the period are represented and marked in the same way. The bias can then favor codes used at other phases, and the detector does not know the phase of the observed window. We take the phase of a periodic series as the organizing principle for the codebook, the watermark, and the detector. PHASELOCK partitions the codebook into phase groups and quantizes each patch using the group corresponding to its phase. The tokenizer and generator are trained only on windows aligned to the patch grid of the period. PHASELOCK outputs each generated window with a random temporal shift, allowing samples to start at arbitrary times as in real data, and it injects the watermark with a phase-gated bias, so that each phase is marked with its own codes. At detection, a single re-encoding network recovers the grid alignment, phase, and codes directly from the observed signal. On ten periodic benchmarks, PHASELOCK remains detectable under post-editing attacks at a lower Context-FID than the same generator with an unstructured codebook of equal size. It also yields fewer false positives under wrong keys and preserves the periodicity of the data more closely than existing watermarks.

## 1 INTRODUCTION

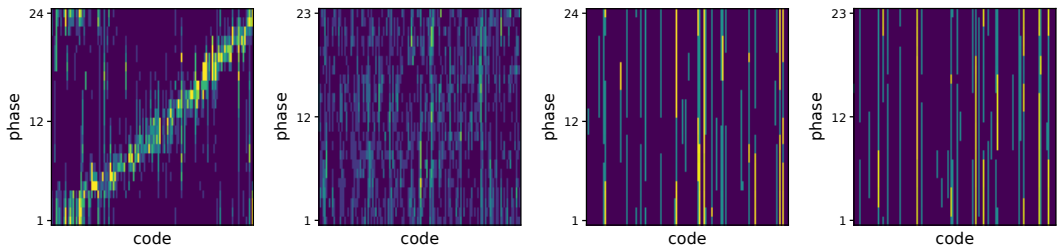
Generative models of time series support data augmentation and sharing in domains where observations are scarce or subject to privacy constraints, including energy, healthcare, and finance (Yuan & Qiao, 2024; Naiman et al., 2024; Narasimhan et al., 2024). Generators based on vector quantization (VQ) encode continuous signals into discrete codes and learn sequence priors to synthesize new samples (Lee et al., 2023; Chen et al., 2024b). As the fidelity of synthetic series improves, reliable provenance verification becomes increasingly important, motivating generation-time watermarking (Soi et al., 2026; Kim et al., 2026). In VQ-based time-series watermarking, the generator biases sampling toward key-dependent codes, and the detector recovers the codes from the observed signal, which may have been edited after generation (Kim et al., 2026). Codebook organization therefore underlies generation, watermark insertion, and token recovery.

In time series, unlike images or language, observations are often periodic, and local patterns depend on their phase, the position within the cycle (Lin et al., 2024b). Forecasting models have exploited this dependence (Lai et al., 2018; Wu et al., 2023; Lin et al., 2024a). We observe that VQ codebooks capture the same structure without phase supervision. In a pretrained VQ tokenizer on ETTh, subsets of codes are preferentially used at specific daily phases, and this preference weakens under a mismatched period (Figure 1). In contrast, Exchange, which has no daily cycle, shows no comparable phase dependence at the tested periods. Nevertheless, existing VQ-based watermarking does not exploit phase: green sets are drawn from the entire codebook irrespective of phase, and the detector recovers codes without estimating the phase or the patch alignment of the observed window. For edited windows whose alignment is unknown, phase could serve as an additional constraint on token recovery.

Phase matters for the watermark at two stages that the existing methods do not address. At generation, a green set drawn from the whole codebook lets the bias raise codes that the tokenizer uses



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.




(a) ETTh, $\tau = 24$ (b) ETTh, $\tau = 23$ (c) Exchange, $\tau = 24$ (d) Exchange, $\tau = 23$

Figure 1: **A plain VQ codebook remembers the period of its data.** Each panel shows the conditional code distribution $p(c \mid \varphi)$ of a VQ tokenizer trained under the standard protocol, in which windows start at every time step. The phase of a token is $\varphi = t \bmod \tau$, where $t$ is the time step at which the token starts and $\tau$ is a candidate period in time steps. For hourly ETTh a distinct diagonal band appears at $\tau = 24$, the daily cycle (a), and no comparable band appears at $\tau = 23$ (b). Exchange, sampled daily, shows no band at either period (c, d). Columns are ordered by the preferred phase of each code, estimated on alternating blocks of 96 tokens, and the displayed distributions are computed from the remaining blocks.

at other phases, which can pull a sample away from the periodic pattern of the data. At detection, the phase of an observed window is unknown, since real windows start at arbitrary times and a post-editing attack can shift them further. We propose PHASELOCK, which organizes the codebook, the watermark, and the detector by phase. It quantizes each patch within its corresponding phase group, draws green codes within each group, and biases only the group of the current phase. Generated windows are output from a random shift of the patch grid, and a phase-aware re-encoder recovers the alignment, the phase, and the codes of an observed window from the signal alone before the key is used to count green tokens.

Our contributions are as follows.

- We show that a VQ codebook trained without phase supervision uses different codes at different phases of a periodic series (Figure 1).
- We propose PHASELOCK, a generation-time watermark for periodic time series that uses phase in the tokenizer, the green set, and the detector, and recovers the alignment and phase of an edited window without the key.
- On ten periodic datasets, PHASELOCK remains detectable under post-editing attacks without false positives, at a lower Context-FID than the same generator with an unstructured codebook of equal size, and with fewer false positives under wrong keys.

## 2 RELATED WORKS

### 2.1 WATERMARKING IN TIME SERIES

Generation-time watermarking embeds statistical signals by modifying the sampling process (Dathathri et al., 2024; Zhao et al., 2024). A representative approach is KGW (Kirchenbauer et al., 2023), a token-level watermarking framework for language models that partitions the vocabulary into pseudorandom green and red sets. At each generation step, KGW adds a positive bias to the logits of green tokens, increasing their probability of being sampled. Detection then re-encodes the observed token sequence and tests whether the proportion of green tokens significantly exceeds its non-watermarked expectation. Diffusion-based methods embed watermarks into the initial noise: Tree-Ring (Wen et al., 2023) uses a key-dependent Fourier pattern, while Gaussian Shading (Yang et al., 2024) encodes bits while preserving the Gaussian distribution. Both detect watermarks by recovering the initial noise via sampler inversion.

Recent work extends this generation-time watermarking framework to synthetic time series. TimeWak (Soi et al., 2026) employs a recovered representation $\Phi(\mathbf{x})$ and position-wise target regions $\mathcal{T}_n$, while retaining the agreement-based statistical structure of KGW. Detection from continuous

time series requires re-encoding the observed signal $\mathbf{x}$ into $\Phi(\mathbf{x})$ and evaluating the proportion of recovered units satisfying $\Phi(\mathbf{x})_n \in \mathcal{T}_n$. Watermark presence is determined by testing whether this agreement rate significantly exceeds the non-watermarked baseline. The test statistic and how a post-editing attack can shift it on non-watermarked data are described in Appendix A. LVQMark (Kim et al., 2026) adopts a discrete-token formulation that combines local time-series tokenization with generation-time logit bias. Its detector incorporates robust re-encoding to recover tokens from perturbed observations and evaluates green-list membership in the recovered sequence for detection under post-editing attacks.

### 2.2 Structured Codebooks and Token Grouping

Discrete token models based on vector quantization (VQ) increasingly structure the codebook rather than treating it as a single flat set (Tian et al., 2024). In visual generation, STSCQ (Wang et al., 2026) combines image-level codebook selection with token-specific sub-codebooks, and IAR (Hu et al., 2025) groups codes through balanced $k$-means clustering and trains the autoregressive prior with a cluster-oriented objective. In time-series generation, TimeVQVAE (Lee et al., 2023) assigns dedicated codebooks to low- and high-frequency components, and MSDformer (Feng et al., 2026) cascades scale-specific VQ-VAEs with independent codebooks. The codebook is thus organized by token position, embedding geometry, frequency band, or temporal scale.

Grouping tokens by shared characteristics has also been applied to the green–red partition of generation-time watermarks. In language models, WatME (Chen et al., 2024a) partitions clusters of lexically redundant tokens into green and red subsets, and Guo et al. (2024) construct semantic sets for the partition by locality-sensitive hashing of token embeddings. SIR (Liu et al., 2024) derives the watermark logits from a semantic embedding of the preceding text, so that semantically similar contexts receive similar partitions. In visual generation, ClusterMark (Lukovnikov et al., 2026) assigns green and red labels to embedding clusters of the codebook, so that a substitution within a cluster during re-encoding does not change the label of a token. In these methods, tokens are grouped by semantic or embedding similarity. For time series, periodicity provides another basis for grouping. Similar local patterns appear at the same phase of each cycle, so tokens can be grouped by phase, their position within the cycle. This criterion differs from the frequency bands and temporal scales above, and it has not been used to structure codebooks or the watermarks built on them.

## 3 Proposed Method

We propose PhaseLock, which takes phase, the position of a patch within the period, as the organizing principle of the codebook and builds the watermark and the detector around it. The key marks each phase with its own set of codes, and the detector recovers the phase of an observed window from the signal alone before it counts green tokens.

PhaseLock consists of three stages (Fig. 2). We first construct a phase-partitioned tokenizer by partitioning the codebook into phase groups and training on windows aligned to the patch grid (§3.1), then perform watermarked generation with a phase-gated bias and a random sub-patch shift of the output (§3.2), and finally recover the alignment, the phase, and the codes of an observed window for detection (§3.3). The generator, the network used for re-encoding, and the green-ratio statistic are inherited from LVQMark (Kim et al., 2026).

### 3.1 Phase-partitioned tokenizer

**Partitioning the codebook by phase.** Let $\mathbf{x} \in \mathbb{R}^{T \times D}$ be a window of $T$ time steps and $D$ variables. The tokenizer divides it into non-overlapping patches of $s$ time steps, $\mathbf{w}_n = \mathbf{x}_{(n-1)s+1:ns}$ for $n = 1, \ldots, T/s$, and the encoder $E_\phi$ maps each patch to a vector that is quantized to the nearest code of the codebook $\mathcal{C} = \{\mathbf{e}_k\}_{k=1}^{K}$. The index $s_n$ of that code is the token of patch $n$. Let $\tau$ be the period of the series in time steps. It follows from the sampling interval and the cycle of interest, for example $\tau = 24$ for hourly data with a daily cycle and $\tau = 96$ for 15-minute data, and when it is not given it can be estimated from the autocorrelation function (Appendix B.3). A cycle thus contains

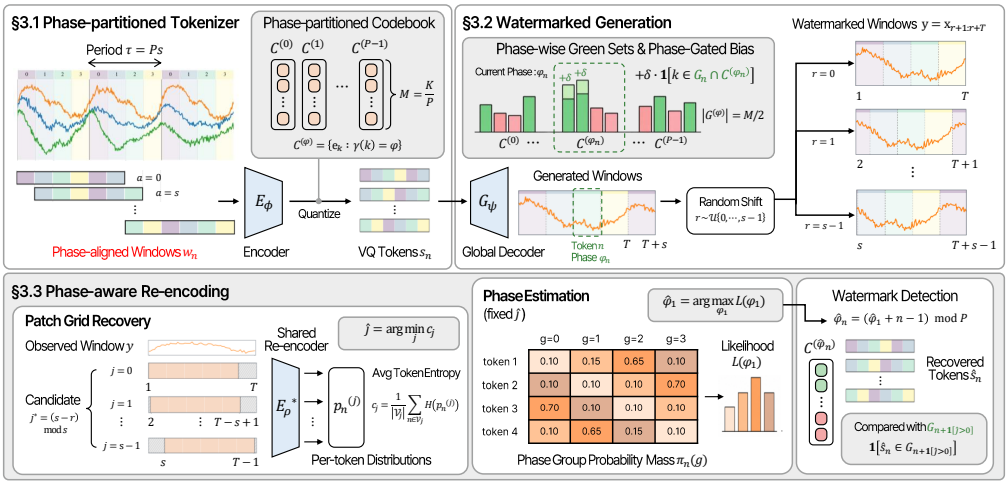



Figure 2: Overview of PHASELOCK. (§3.1) The codebook is partitioned into $P$ phase groups and the tokenizer is trained on windows aligned to the patch grid, so that each token is quantized within the group of its phase. (§3.2) Generation adds the bias $\delta$ only to the green codes of the current phase group, and each generated window is output from a random shift $r$. (§3.3) Detection re-encodes an observed window with a single network, recovers the patch grid from the entropy of the per-token distributions, the phase from their mass per group, and the codes within the recovered group, and then counts green tokens.

$P = \tau/s$ patches. We partition the codebook into $P$ groups of equal size $M = K/P$,

$$\mathcal{C}^{(\varphi)} = \{\mathbf{e}_k : \gamma(k) = \varphi\}, \qquad \varphi \in \{0, \ldots, P-1\}, \tag{1}$$

where $\gamma(k) = \lfloor (k-1)/M \rfloor$ assigns code $k$ to its group, so that the groups are consecutive blocks of $M$ codes. We call $\varphi$ the phase. With time step 0 at the start of a cycle, the phase of a patch is its position within its cycle, 0 for the first patch of the cycle through $P-1$ for the last, and the codes of $\mathcal{C}^{(\varphi)}$ are used for patches of phase $\varphi$.

**Aligning training windows to the grid.** The patches of a window that starts at time step $a$ lie on this grid only when $a$ is a multiple of $s$. For any other $a$ the phase labels of its patches depend on the distance of $a$ from the grid, which a detector that only observes the window cannot know. We therefore train the tokenizer, and the generator on top of it, only on windows that start at a multiple of $s$. For such windows, the phase advances by exactly one per token,

$$\varphi_n = (\varphi_1 + n - 1) \bmod P, \tag{2}$$

where $\varphi_1$ is the phase of the first patch, so that the whole phase sequence of a window is determined by the phase of its first token.

**Quantizing within the phase group.** Quantization is restricted to the phase group of the patch,

$$s_n = \arg \min_{k \,:\, \mathbf{e}_k \in \mathcal{C}^{(\varphi_n)}} \|E_\phi(\mathbf{w}_n) - \mathbf{e}_k\|_2^2, \tag{3}$$

implemented as a masked nearest-neighbor search. The training objective and the codebook updates are unchanged, except that an unused code is re-seeded from its own group so that every group keeps $M$ live codes. The global decoder $G_\psi$ and the autoregressive prior $p_\theta$ are trained on the resulting token sequences. The prior operates on code indices and receives no phase input. Because each code belongs to exactly one phase group, the phase of a generated token is recoverable from its code as $\gamma(s_n)$.

### 3.2 WATERMARKED GENERATION

Generation with PHASELOCK differs from unbiased sampling from $p_\theta$ in two respects. The watermark is injected through a green set stratified by phase and a logit bias applied only to the green

codes of the current phase group, where the phase of the window is determined by its first token. Each window is then generated one token longer than needed and output from a random shift, so that samples start at all possible alignments rather than on the grid.

**Stratifying the green set by phase.** A green-ratio watermark draws a fixed green set $G^{(0)}$ of size $K/2$ from the codes $V = \{1, \dots, K\}$ using the key $\kappa$ (Kirchenbauer et al., 2023). PHASELOCK draws it within each group: for every $\varphi$ exactly $M/2$ codes of $\mathcal{C}^{(\varphi)}$ are green, so that $|G^{(0)}| = K/2$ and the green fraction is the same as for an unstructured codebook. The green set $G_n$ at position $n$ and the unbiased warm-up of $m$ positions at the start of each window follow the inherited scheme (Appendix A), where $G_n$ alternates between $G^{(0)}$ and its complement with the parity of $n$.

**Gating the bias by phase.** The first token is sampled from $p_\theta$ with neither bias nor mask. Its group defines the phase of the window, $\varphi_1 = \gamma(s_1)$, and the phases of the remaining positions follow Eq. 2. For $n > m$ the bias $\delta > 0$ is added to the logit $\ell_{n,k}$ that $p_\theta$ assigns to code $k$ at position $n$, only for green codes of the current phase group,

$$\tilde{\ell}_{n,k} = \ell_{n,k} + \delta \, \mathbf{1}\Big[k \in G_n \cap \mathcal{C}^{(\varphi_n)}\Big]. \tag{4}$$

**Shifting the output window.** Because training windows start on the grid, a generator that emitted such windows directly would produce samples that all start at phase boundaries, which real windows do not. We therefore generate a window $\mathbf{x}$ of $T + s$ time steps, one token longer than needed, and output the window $\mathbf{y}$ of $T$ time steps that starts at a random shift $r$,

$$\mathbf{y} = \mathbf{x}_{r+1:r+T}, \qquad r \sim \mathrm{Uniform}\{0, \dots, s-1\}, \tag{5}$$

drawn independently per window. The $s$ values of $r$ cover every alignment of the grid exactly once, so the output windows start at arbitrary alignments and the detector has $s$ alignments to consider. Everything downstream, quality metrics included, uses only the output windows of length $T$.

### 3.3 PHASE-AWARE RE-ENCODING

The detector receives a window $\mathbf{y} \in \mathbb{R}^{T \times D}$ of $D$ variables that may have been post-edited by an attack, and knows neither the shift $r$ nor the phase $\varphi_1$. Its re-encoding map $\Phi(\mathbf{y})$ therefore proceeds in three steps, each fixing one unknown for the next: it first aligns the window to the patch grid, then estimates the phase of the aligned tokens, and finally reads the codes under the group restriction of the estimated phase and counts green tokens. All three steps use a single network that outputs, for every token, a distribution over the $K$ codes: the alignment is read from the entropy of that distribution, the phase from its mass per group, and the code from its maximum within the estimated group.

**Training the network.** The re-encoding network $E^\star_\rho$ outputs, for every token of a window, a distribution over the $K$ codes (Appendix A). PHASELOCK changes its training in two respects. Training inputs replay the output step: generated windows are truncated at a random shift, attacked, and re-aligned to the grid with the true shift, so that training inputs are prepared by the same procedure as detection inputs. Training targets are the group-restricted codes of Eq. 3 under the true phase. These targets coincide with the sampled tokens for around $98\%$ of patches. The network receives no phase input, and the phase is obtained from its output as described below.

**Recovering the patch grid.** The first step locates the patch grid inside $\mathbf{y}$, which can be at one of $s$ positions. Candidate $j \in \{0, \dots, s-1\}$ discards the first $j$ time steps of $\mathbf{y}$ and keeps the rest up to the last complete patch, so $\mathbf{y}^{(0)}$ has $T/s$ tokens and every other candidate one fewer. The true candidate is $j^* = (s - r) \bmod s$. Each candidate is passed through the network once, giving per-token distributions $\mathbf{p}_n^{(j)}$. Let $\mathcal{V}_j$ be the set of valid tokens of $\mathbf{y}^{(j)}$: time steps at either end of the window that are constant across all variables are trimmed, and a token is valid when all of its $s$ time steps lie inside the remaining range. We select the candidate whose tokens are, on average, predicted with the least uncertainty,

$$\hat{j} = \arg\min_j \; \frac{1}{|\mathcal{V}_j|} \sum_{n \in \mathcal{V}_j} H\Big(\mathbf{p}_n^{(j)}\Big), \tag{6}$$

where $H$ is the entropy, with ties going to the smallest $j$. The criterion uses neither the key nor the green sets, so searching over $s$ candidates per window does not bias the green ratio of non-watermarked windows. We write $\mathbf{p}_n$ and $\mathcal{V}$ for the distributions and valid tokens of the selected candidate, with its tokens indexed $n = 1, 2, \ldots$ from its first complete patch.

**Estimating the phase.** The second step fixes the phase of the aligned tokens by maximum likelihood. For each token $n$, let $\pi_n(g) = \sum_{k:\gamma(k)=g} p_{n,k}$ denote the probability that $\mathbf{p}_n$ assigns to phase group $g$, where $p_{n,k}$ is the probability of code $k$. By Eq. 2, the phase $\varphi_1$ of the first scored token fixes the phase $\varphi_n$ of every other token, so a hypothesis is a value of $\varphi_1$ and there are $P$ of them. Under a hypothesis, token $n$ carries a code of group $\varphi_n$, and $\pi_n(\varphi_n)$ is the likelihood of that token. Treating tokens as independent, the likelihood of the hypothesis is the product over valid tokens, and we select its maximizer,

$$\hat{\varphi}_1 = \arg \max_{\varphi_1 \in \{0,\ldots,P-1\}} \prod_{n \in \mathcal{V}} \pi_n(\varphi_n), \tag{7}$$

with $\varphi_n$ given by Eq. 2, computed as a sum of logarithms with the mass floored at a small $\epsilon$, and ties going to the smallest $\varphi_1$.

**Reading the codes and counting green tokens.** The third step reads one code per token within the recovered phase group and tests it against the green set. With $\hat{\varphi}_1$ fixed, token $n$ has phase $\hat{\varphi}_n$ by Eq. 2 and its code $\hat{s}_n$ is the highest-scoring code of $\mathcal{C}^{(\hat{\varphi}_n)}$ under $\mathbf{p}_n$. This is the group restriction of Eq. 3 applied at detection, with the estimated phase in place of the true one. The sequence $\hat{\mathbf{s}}$ together with $\hat{j}$ and $\hat{\varphi}_1$ is the output of $\Phi(\mathbf{y})$. Because $G_n$ depends on the position $n$, positions are counted from the first generated token: if $\hat{j} = 0$ the first scored token is generation position 1, otherwise the window starts off the grid and its first complete patch is position 2. We therefore test token $n$ against $G_{n+o}$ with $o = \mathbf{1}[\hat{j} > 0]$. The green ratio is

$$\hat{g}(\mathbf{y}) = \frac{1}{|\mathcal{V}|} \sum_{n \in \mathcal{V}} \mathbf{1}[\hat{s}_n \in G_{n+o}], \tag{8}$$

with $\hat{g} = 1/2$ when $\mathcal{V}$ is empty. All valid tokens are scored, warm-up positions included. The detection statistic computed from $\hat{g}$ over a set of windows is described in Appendix B.5.

# 4 Experiments

## 4.1 Setup

We evaluate on ten multivariate datasets with a daily cycle: ETTh, ETTm, the six air quality stations AirQ-1 to AirQ-6, Weather, and Energy. The period is one day, which gives $P = 4$ phase groups on ETTh and AirQ, 16 on ETTm, and 24 on Weather and Energy. Exchange has no cycle and serves as the aperiodic control of Appendix D.1. LVQMark (Kim et al., 2026) is the same generator with an unstructured codebook under the standard protocol, run at the codebook size of PhaseLock and at one larger size. TimeWak (Soi et al., 2026), Tree-Ring (TR) (Wen et al., 2023), and Gaussian Shading (GS) (Yang et al., 2024) are run on Diffusion-TS (Yuan & Qiao, 2024).

Every method is evaluated clean and under the post-editing attacks offset, crop, and insert at strength 0.30, with the green-ratio $z$-test of Appendix B.5 applied to watermarked and non-watermarked pools. We report the true positive rate (TPR) and mark a non-watermarked pool as TN, FP, or AN when it is not detected, falsely detected, or detected with a negative $z$. LVQMark and PhaseLock are compared at their minimal strength $\delta^\star$, the smallest $\delta$ at which the TPR is 1.00 in every condition, and sample quality is measured at $\delta^\star$ by Context-FID (C-FID) and the correlational score (Corr.). Datasets, hyperparameters, attacks, and the full protocol are given in Appendix B.

## 4.2 Watermark detection and sample quality

Table 1 reports detection and sample quality on the ten periodic datasets. The two L-VQVAE methods, LVQMark and PhaseLock, are detected in every condition at their $\delta^\star$ and never fire on non-watermarked windows. The Diffusion-TS methods do not reach this standard. TR misses the offset attack on most datasets (eight of ten) and produces false positives under crop and insert on all of

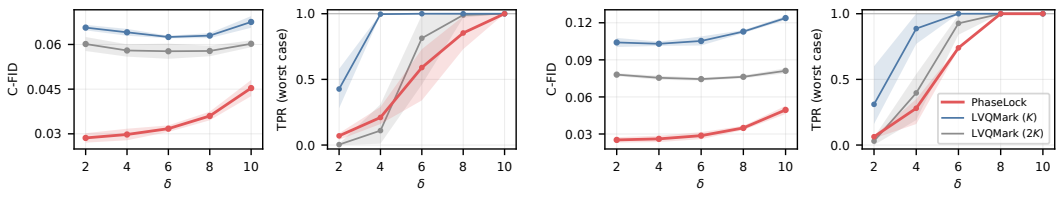
Table 1: Watermark detection and sample quality on ten datasets (mean over three seeds). LVQMark and **PhaseLock** rows are reported at their minimal strength $\delta^\star$: the smallest $\delta$ with TPR 1.00 on the clean pool and under all three 30% attacks. WM: TPR under offset / crop / insert. Non-WM: TN = not detected, **FP** = false positive, **AN** = anomaly. Best in bold, second-best underlined.

| Setting | | | | | WM (TPR ↑) | | | Non-WM | | | Quality (↓) | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Data | Method | P | K | $\delta$ | Off | Crop | Ins | Off | Crop | Ins | C-FID | Corr. |
| ETTh | TR | – | – | – | **1.00** | **1.00** | **1.00** | TN | FP | FP | 2.108 | 0.265 |
| | GS | – | – | – | **1.00** | **1.00** | **1.00** | TN | AN | AN | 5.510 | 0.291 |
| | TimeWak | – | – | – | **1.00** | **1.00** | **1.00** | TN | TN | TN | 0.691 | 0.158 |
| | LVQMark | – | 256 | 8 | **1.00** | **1.00** | **1.00** | TN | TN | TN | 0.051 | 0.052 |
| | LVQMark | – | 512 | 8 | **1.00** | **1.00** | **1.00** | TN | TN | TN | **0.037** | 0.045 |
| | **PhaseLock** | 4 | 256 | 10 | **1.00** | **1.00** | **1.00** | TN | TN | TN | 0.039 | **0.044** |
| ETTm | TR | – | – | – | 0.00 | **1.00** | **1.00** | TN | FP | FP | 3.222 | 0.791 |
| | GS | – | – | – | **1.00** | **1.00** | **1.00** | FP | FP | FP | 10.068 | 0.976 |
| | TimeWak | – | – | – | **1.00** | **1.00** | **1.00** | TN | FP | FP | 0.463 | 0.195 |
| | LVQMark | – | 1024 | 6 | **1.00** | **1.00** | **1.00** | TN | TN | TN | 0.063 | 0.029 |
| | LVQMark | – | 2048 | 10 | **1.00** | **1.00** | **1.00** | TN | TN | TN | 0.060 | **0.028** |
| | **PhaseLock** | 16 | 1024 | 10 | **1.00** | **1.00** | **1.00** | TN | TN | TN | **0.045** | 0.035 |
| AirQ-1 | TR | – | – | – | 0.00 | **1.00** | **1.00** | TN | FP | FP | 3.222 | 0.791 |
| | GS | – | – | – | **1.00** | **1.00** | **1.00** | FP | FP | FP | 10.068 | 0.976 |
| | TimeWak | – | – | – | **1.00** | **1.00** | **1.00** | TN | FP | FP | 0.463 | 0.195 |
| | LVQMark | – | 512 | 8 | **1.00** | **1.00** | **1.00** | TN | TN | TN | 0.109 | 0.132 |
| | LVQMark | – | 1024 | 8 | **1.00** | **1.00** | **1.00** | TN | TN | TN | 0.064 | 0.129 |
| | **PhaseLock** | 4 | 512 | 10 | **1.00** | **1.00** | **1.00** | TN | TN | TN | **0.025** | **0.115** |
| AirQ-2 | TR | – | – | – | 0.00 | **1.00** | **1.00** | TN | FP | FP | 2.855 | 0.658 |
| | GS | – | – | – | **1.00** | **1.00** | **1.00** | TN | FP | FP | 9.461 | 1.135 |
| | TimeWak | – | – | – | **1.00** | **1.00** | **1.00** | TN | FP | FP | 0.550 | 0.231 |
| | LVQMark | – | 512 | 6 | **1.00** | **1.00** | **1.00** | TN | TN | TN | 0.114 | 0.202 |
| | LVQMark | – | 1024 | 8 | **1.00** | **1.00** | **1.00** | TN | TN | TN | 0.078 | 0.101 |
| | **PhaseLock** | 4 | 512 | 10 | **1.00** | **1.00** | **1.00** | TN | TN | TN | **0.026** | **0.093** |
| AirQ-3 | TR | – | – | – | 0.00 | **1.00** | **1.00** | TN | FP | FP | 3.411 | 1.101 |
| | GS | – | – | – | **1.00** | **1.00** | **1.00** | TN | TN | FP | 9.297 | 1.099 |
| | TimeWak | – | – | – | **1.00** | **1.00** | **1.00** | TN | FP | TN | 0.490 | 0.296 |
| | LVQMark | – | 512 | 6 | **1.00** | **1.00** | **1.00** | TN | TN | TN | 0.074 | 0.120 |
| | LVQMark | – | 1024 | 6 | **1.00** | **1.00** | **1.00** | TN | TN | TN | 0.058 | 0.136 |
| | **PhaseLock** | 4 | 512 | 10 | **1.00** | **1.00** | **1.00** | TN | TN | TN | **0.036** | **0.093** |
| AirQ-4 | TR | – | – | – | 0.00 | **1.00** | **1.00** | TN | FP | FP | 3.278 | 0.486 |
| | GS | – | – | – | **1.00** | **1.00** | **1.00** | TN | TN | FP | 9.703 | 0.715 |
| | TimeWak | – | – | – | **1.00** | **1.00** | **1.00** | TN | TN | TN | 0.676 | 0.223 |
| | LVQMark | – | 512 | 8 | **1.00** | **1.00** | **1.00** | TN | TN | TN | 0.110 | 0.162 |
| | LVQMark | – | 1024 | 8 | **1.00** | **1.00** | **1.00** | TN | TN | TN | 0.079 | 0.138 |
| | **PhaseLock** | 4 | 512 | 10 | **1.00** | **1.00** | **1.00** | TN | TN | TN | **0.025** | **0.110** |
| AirQ-5 | TR | – | – | – | 0.00 | **1.00** | **1.00** | TN | FP | FP | 3.771 | 0.654 |
| | GS | – | – | – | **1.00** | **1.00** | **1.00** | TN | FP | FP | 11.343 | 0.934 |
| | TimeWak | – | – | – | **1.00** | **1.00** | **1.00** | TN | TN | TN | 0.553 | 0.198 |
| | LVQMark | – | 512 | 8 | **1.00** | **1.00** | **1.00** | TN | TN | TN | 0.128 | 0.172 |
| | LVQMark | – | 1024 | 8 | **1.00** | **1.00** | **1.00** | TN | TN | TN | 0.071 | 0.107 |
| | **PhaseLock** | 4 | 512 | 10 | **1.00** | **1.00** | **1.00** | TN | TN | TN | **0.023** | **0.093** |
| AirQ-6 | TR | – | – | – | 0.00 | **1.00** | **1.00** | TN | FP | FP | 3.187 | 0.495 |
| | GS | – | – | – | **1.00** | **1.00** | **1.00** | TN | FP | FP | 10.189 | 1.122 |
| | TimeWak | – | – | – | **1.00** | **1.00** | **1.00** | TN | TN | FP | 0.428 | 0.201 |
| | LVQMark | – | 512 | 12 | **1.00** | **1.00** | **1.00** | TN | TN | TN | 0.129 | 0.253 |
| | LVQMark | – | 1024 | 6 | **1.00** | **1.00** | **1.00** | TN | TN | TN | 0.054 | 0.111 |
| | **PhaseLock** | 4 | 512 | 10 | **1.00** | **1.00** | **1.00** | TN | TN | TN | **0.023** | **0.087** |
| Energy | TR | – | – | – | **1.00** | **1.00** | **1.00** | FP | FP | FP | 0.497 | 1.521 |
| | GS | – | – | – | **1.00** | **1.00** | **1.00** | AN | TN | AN | 6.438 | 4.854 |
| | TimeWak | – | – | – | **1.00** | **1.00** | **1.00** | FP | FP | TN | 0.155 | 1.621 |
| | LVQMark | – | 1152 | 4 | **1.00** | **1.00** | **1.00** | TN | TN | TN | 0.506 | 0.853 |
| | LVQMark | – | 2048 | 6 | **1.00** | **1.00** | **1.00** | TN | TN | TN | 0.325 | **0.802** |
| | **PhaseLock** | 24 | 1152 | 12 | **1.00** | **1.00** | **1.00** | TN | TN | TN | **0.070** | 0.876 |
| Weather | TR | – | – | – | 0.00 | 0.00 | **1.00** | TN | FP | FP | 3.006 | 1.370 |
| | GS | – | – | – | **1.00** | **1.00** | **1.00** | FP | AN | FP | 6.858 | 1.523 |
| | TimeWak | – | – | – | **1.00** | **1.00** | **1.00** | TN | AN | FP | 0.670 | 0.758 |
| | LVQMark | – | 1152 | 6 | **1.00** | **1.00** | **1.00** | TN | TN | TN | 0.105 | 0.345 |
| | LVQMark | – | 2048 | 8 | **1.00** | **1.00** | **1.00** | TN | TN | TN | 0.076 | **0.272** |
| | **PhaseLock** | 24 | 1152 | 8 | **1.00** | **1.00** | **1.00** | TN | TN | TN | **0.035** | 0.350 |

them, GS has a false positive or an anomaly on every dataset, and TimeWak, which is detected everywhere, is flagged on seven of the ten datasets. Where a non-watermarked pool is flagged, the TPR in that column is not evidence of detection. The accuracy with which the alignment and the phase of a watermarked window are recovered, on the clean pool and under each attack, is given in Appendix C.4.

Between the two L-VQVAE methods, PHASELOCK retains the better sample quality at the strength where it is detectable. Its Context-FID is below that of LVQMark at the same codebook size on every dataset, by a factor of 1.3 to 7, and below that of LVQMark with a larger codebook on nine of the ten. ETTh is the exception, where LVQMark at the larger codebook size is lowest (0.037 against 0.039). The correlational score of PHASELOCK is the lowest of the block on ETTh and the six AirQ stations and higher than LVQMark on ETTm, Weather, and Energy.

### 4.3 DETECTION AND QUALITY ACROSS WATERMARK STRENGTHS




(a) ETTm (b) Weather

Figure 3: Watermark strength sweep. In each panel, left: Context-FID of the watermarked pool, right: worst-case TPR, the smallest TPR over the clean pool and the three attacks. PHASELOCK is compared with LVQMark at the same codebook size ($K$) and about twice the size. Shaded bands are the range over three generation seeds.

Figure 3 sweeps the watermark strength on ETTm and Weather. PHASELOCK adds the bias to the green codes of a single phase group, one $P$-th of the codes that LVQMark biases, and a token is counted as green only when its phase is recovered as well, so the same $\delta$ yields a smaller green excess and PHASELOCK reaches TPR 1.00 at a larger strength (10 against 6 on ETTm, 8 against 6 on Weather). Its sample quality is nevertheless the better one at every strength: the Context-FID of PHASELOCK at its own $\delta^\star$ remains below that of LVQMark at its smaller $\delta^\star$, and below that of LVQMark with a codebook twice as large. The sweeps on the other datasets are given in Appendix C.

### 4.4 FALSE POSITIVES UNDER WRONG KEYS

Table 2: Detection with wrong keys. Each watermarked pool of Table 1 is scored with ten other keys under the three attacks. Mean and max are taken over the thirty combinations of attack and key. LVQMark is at the same codebook size as PHASELOCK. Better value in bold.

| **Method** | **Dataset** | $\delta^\star$ | $\text{FPR}_{\text{mean}}$ | $\text{FPR}_{\text{max}}$ | $z_{\text{max}}$ | **Dataset** | $\delta^\star$ | $\text{FPR}_{\text{mean}}$ | $\text{FPR}_{\text{max}}$ | $z_{\text{max}}$ |
|---|---|---|---|---|---|---|---|---|---|---|
| LVQMark | ETTh | 8 | 0.008 | 0.08 | 1.61 | ETTm | 6 | 0.028 | 0.12 | 1.96 |
| PhaseLock | | 10 | **0.004** | **0.02** | **1.18** | | 10 | **0.014** | **0.08** | **1.86** |
| LVQMark | AirQ-1 | 8 | 0.012 | 0.18 | 1.79 | AirQ-2 | 6 | 0.011 | 0.08 | 1.61 |
| PhaseLock | | 10 | **0.001** | **0.02** | **0.60** | | 10 | **0.006** | **0.05** | **1.32** |
| LVQMark | AirQ-3 | 6 | 0.005 | 0.05 | 1.10 | AirQ-4 | 8 | 0.009 | 0.07 | **1.44** |
| PhaseLock | | 10 | **0.004** | **0.02** | **1.09** | | 10 | **0.005** | **0.05** | 1.54 |
| LVQMark | AirQ-5 | 8 | 0.008 | 0.05 | 1.36 | AirQ-6 | 12 | 0.031 | 0.27 | 2.52 |
| PhaseLock | | 10 | **0.004** | **0.02** | **0.87** | | 10 | **0.010** | **0.10** | **1.86** |
| LVQMark | Energy | 4 | 0.010 | 0.08 | 1.48 | Weather | 6 | 0.018 | 0.13 | 1.90 |
| PhaseLock | | 12 | **0.003** | **0.03** | **1.12** | | 8 | **0.003** | **0.04** | **1.03** |

A watermark must not be detected with a key other than the one used at generation. Table 2 scores each watermarked pool with ten wrong keys under the three attacks. PHASELOCK has the lower false positive rate on every dataset, both on average over attacks and keys (0.001 to 0.014 against 0.005 to 0.031) and in the worst combination (at most 0.10 against 0.27), and its largest wrong-key $z$ is lower on nine of the ten datasets. Under a wrong key the green count of both methods stays near its null level, and the residual false positives of PHASELOCK are fewer and smaller.

### 4.5 ABLATION OF THE PHASE PARTITION

Table 3: Ablation of the phase partition. Context-FID at $\delta^\star$ and TPR at $\delta = 10$ on the clean pool and under the three attacks. PHASELOCK w/o partition keeps the grid-aligned training and the shifted output of PHASELOCK on an unstructured codebook of the same size.

| **Method** | **Data** | $K$ | $\delta^\star$ | C-FID | Clean | Offset | Crop | Insert | **Data** | $K$ | $\delta^\star$ | C-FID | Clean | Offset | Crop | Insert |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| LVQMark | AirQ-1 | 1024 | 8 | 0.064 | 1.00 | 1.00 | 1.00 | 1.00 | Energy | 2048 | 6 | 0.325 | 1.00 | 1.00 | 1.00 | 1.00 |
| LVQMark | | 512 | 8 | 0.109 | 1.00 | 1.00 | 1.00 | 1.00 | | 1152 | 4 | 0.506 | 1.00 | 1.00 | 1.00 | 1.00 |
| PHASELOCK w/o partition | | 512 | 12 | 0.032 | 1.00 | 0.97 | 0.98 | 0.88 | | 1152 | 14 | 0.099 | 0.67 | 0.69 | 0.52 | 0.64 |
| PHASELOCK | | 512 | 10 | **0.025** | 1.00 | 1.00 | 1.00 | 1.00 | | 1152 | 12 | **0.070** | 1.00 | 1.00 | 0.99 | 1.00 |

PHASELOCK differs from LVQMark in the phase partition of the codebook and in the shifted output with grid-aligned training. Table 3 removes the partition alone on AirQ-1 and Energy. PHASELOCK without the partition keeps the grid-aligned training and the shifted output on an unstructured codebook of the same size, and its detector recovers the shift and then reads the codes directly, since without groups there is no phase to estimate. Removing the partition improves the Context-FID over LVQMark by a large margin but leaves it above that of PHASELOCK on both datasets. The larger difference is in detection. At the same $\delta$ the TPR of the configuration without the partition falls below 1.00 under every attack on AirQ-1 and in every condition on Energy, whereas PHASELOCK stays at or near 1.00, and its $\delta^\star$ is larger than that of PHASELOCK on both datasets. The phase partition therefore carries much of the detection, and it also contributes to the sample quality.

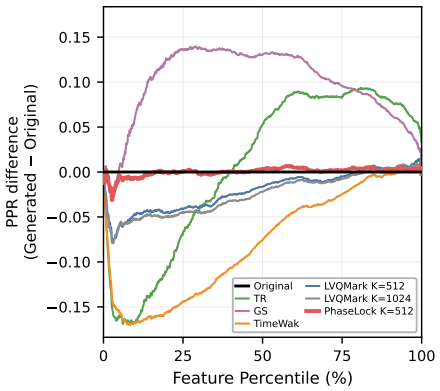



Figure 4: Generated minus real PPR at matching percentiles on AirQ-2 (TEMP). Closer to zero is better.

Table 4: Mean Wasserstein-1 distance between the real and the generated PPR distributions. Lower is better. Best in bold, second-best underlined.

| Method | K | ETTh | ETTm | AirQ-1 | AirQ-2 | AirQ-3 |
|---|---|---|---|---|---|---|
| TR | – | 0.0893 | 0.0284 | 0.0590 | 0.0793 | 0.0773 |
| GS | – | 0.1881 | 0.1676 | 0.0566 | 0.0994 | 0.0891 |
| TimeWak | – | 0.0813 | 0.0243 | 0.0678 | 0.0790 | 0.0718 |
| LVQMark | $K_1$ | **0.0102** | 0.0411 | 0.0465 | 0.0231 | 0.0278 |
| LVQMark | $K_2$ | 0.0167 | 0.0267 | 0.0212 | 0.0265 | 0.0186 |
| **PhaseLock** | $K_1$ | 0.0103 | **0.0175** | **0.0051** | **0.0039** | **0.0050** |

| Method | K | AirQ-4 | AirQ-5 | AirQ-6 | Energy | Weather |
|---|---|---|---|---|---|---|
| TR | – | 0.0803 | 0.0869 | 0.0611 | 0.0112 | 0.0375 |
| GS | – | 0.0387 | 0.1028 | 0.0716 | 0.0646 | 0.0627 |
| TimeWak | – | 0.0704 | 0.0615 | 0.0770 | 0.0068 | 0.0371 |
| LVQMark | $K_1$ | 0.0208 | 0.0357 | 0.0494 | 0.0096 | 0.0065 |
| LVQMark | $K_2$ | 0.0187 | 0.0235 | 0.0148 | 0.0154 | 0.0122 |
| **PhaseLock** | $K_1$ | **0.0067** | **0.0047** | **0.0030** | **0.0060** | **0.0063** |

$K_1$ / $K_2$: ETTh 256 / 512, ETTm 1024 / 2048.
AirQ-1–6 512 / 1024, Energy and Weather 1152 / 2048.

### 4.6 Preservation of periodicity under watermarking

A watermark that biases the token sequence can distort the daily cycle of the series that carries it. Table 4 reports how closely the watermarked series of each method reproduce the distribution of a daily periodic feature on the ten datasets. The periodic power ratio (PPR) of a window is the fraction of its variance that lies at the daily frequency, so the distribution of PPR over windows describes how the strength of the daily cycle varies. Because periodicity is a property of individual variables, and several variables of each dataset carry no daily cycle, we evaluate each dataset on its most periodic variable, the one with the highest mean PPR over the real training windows (Appendix E.1). We compare the PPR distributions of the real test windows and of the watermarked pools of Table 1, without attack, by the one-dimensional Wasserstein distance, computed per generation seed and averaged. Definitions are given in Appendix E.2.

The PPR distance of PhaseLock is the lowest on nine of the ten datasets, including all six AirQ stations, where it lies between 0.0030 and 0.0067, a factor of 2.8 to 5.9 below the next best method. The closest competitor to PhaseLock is LVQMark on ETTh, the six AirQ stations, and Weather, and TimeWak on ETTm and Energy. Figure 4 shows the comparison on AirQ-2. Each curve shows generated minus real PPR at matching percentiles, averaged over generation seeds. The curve of PhaseLock stays close to zero over most percentiles, consistent with its distance of 0.0039 against 0.0231 for the closest competitor, LVQMark at $K = 512$.

## 5 Conclusion and Limitations

We introduced PhaseLock, which takes the phase of a periodic series as the organizing principle of the codebook, the watermark, and the detector. The codebook is partitioned into phase groups, the bias is gated by phase, each generated window is output from a random shift, and a single re-encoding network recovers the alignment, the phase, and the codes of an observed window. PhaseLock is a method for series with a period. On a series without one the watermark is hardly recovered, one period is used even when a series has several, and the codebook grows with the number of patches in the period. Its detector runs the network once per candidate alignment, and edits that change the length of the window are not evaluated. Following previous work, the network is trained with the same attack families that are used at evaluation, and attacks outside these families are not evaluated. Within this scope, on ten periodic datasets PhaseLock remains detectable under post-editing attacks at a lower Context-FID than LVQMark with the same codebook size, produces fewer false positives under wrong keys, and reproduces the daily periodic strength of the data more closely than the compared methods on nine of the ten datasets. Training the tokenizer on the patch grid of the period and partitioning its codebook by phase together yield a generator whose samples are of higher quality and whose watermark is detected more reliably than with an unstructured codebook of the same size.

### AI USE STATEMENT

We used generative AI tools to implement methods and to edit the text for readability. We have not used them for any other task with required disclosure, and translation and qualitative or thematic data analysis are not applicable to this work. All AI-assisted code and text were reviewed and verified by the authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

### ETHICS STATEMENT

This work concerns the provenance of synthetic time series. All datasets are public benchmarks without personal identifiers, listed with their sources in Appendix B.2, and no human subjects are involved. A generation-time watermark supports the verification of synthetic data and does not by itself prevent the generation of misleading series. Detection requires the key, so the method does not enable the identification of data that was generated without it. The authors declare no conflicts of interest.

### REPRODUCIBILITY STATEMENT

Section 3 specifies the tokenizer, the watermark, and the detector with the equations used in the implementation. Appendix B lists the hyperparameters of every model (Tables 7 to 9), the datasets with their sources and preprocessing (Table 10), the attacks (Appendix B.4), and the evaluation protocol with all seeds and keys (Appendix B.5). Appendix E defines the periodicity measures. The source code, the configuration files, and the scripts that generate the tables are provided as supplementary material.

## REFERENCES

Luis M Candanedo, Véronique Feldheim, and Dominique Deramaix. Data driven prediction models of energy use of appliances in a low-energy house. *Energy and buildings*, 140:81–97, 2017.

Liang Chen, Yatao Bian, Yang Deng, Deng Cai, Shuaiyi Li, Peilin Zhao, and Kam-Fai Wong. Watme: Towards lossless watermarking through lexical redundancy. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9166–9180, 2024a.

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, 2024b.

Sumanth Dathathri, Abigail See, Sumedh Ghaisas, Po-Sen Huang, Rob McAdam, Johannes Welbl, Vandana Bachani, Alex Kaskasoli, Robert Stanforth, Tatiana Matejovicova, et al. Scalable watermarking for identifying large language model outputs. *Nature*, 634(8035):818–823, 2024.

Shibo Feng, Zhicheng Chen, Xi Xiao, Zhong Zhang, Qing Li, Xingyu Gao, and Peilin Zhao. Msdformer: Multi-scale discrete transformer for time series generation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2026.

Yuxuan Guo, Zhiliang Tian, Yiping Song, Tianlun Liu, Liang Ding, and Dongsheng Li. Context-aware watermark with semantic balanced green-red lists for large language models. In *Proceedings of the 2024 conference on empirical methods in natural language processing*, pp. 22633–22646, 2024.

Teng Hu, Jiangning Zhang, Ran Yi, Jieyu Weng, Yabiao Wang, Xianfang Zeng, Zhucun Xue, and Lizhuang Ma. Improving autoregressive visual generation with cluster-oriented token prediction. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9351–9360. IEEE, 2025.

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 *International Conference on Learning Representations*, 2022.

Dongbin Kim, Geonwoo Shin, Yujin Choi, Soyeon Park, and Jaewook Lee. A locally tokenized generative model for robust time-series watermarking. *arXiv preprint arXiv:2608.19727*, 2026.

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*, pp. 17061–17084. PMLR, 2023.

Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. Modeling long-and short-term temporal patterns with deep neural networks. In *The 41st international ACM SIGIR conference on research & development in information retrieval*, pp. 95–104, 2018.

Daesoo Lee, Sara Malacarne, and Erlend Aune. Vector quantized time series generation with a bidirectional prior model. In Francisco Ruiz, Jennifer Dy, and Jan-Willem van de Meent (eds.), *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pp. 7665–7693. PMLR, 25–27 Apr 2023.

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.

Shengsheng Lin, Weiwei Lin, Xinyi Hu, Wentai Wu, Ruichao Mo, and Haocheng Zhong. Cyclenet: Enhancing time series forecasting through modeling periodic patterns. *Advances in Neural Information Processing Systems*, 37:106315–106345, 2024a.

Shengsheng Lin, Weiwei Lin, Wentai Wu, Haojun Chen, and Junjie Yang. SparseTSF: Modeling long-term time series forecasting with *1k* parameters. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 30211–30226. PMLR, 21–27 Jul 2024b.

Aiwei Liu, Leyi Pan, Xuming Hu, Shiao Meng, and Lijie Wen. A semantic invariant robust watermark for large language models. In *International Conference on Learning Representations*, volume 2024, pp. 6499–6519, 2024.

Denis Lukovnikov, Andreas Müller, Erwin Quiring, and Asja Fischer. Clustermark: Towards robust watermarking for autoregressive image generators with visual token clustering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9213–9222, 2026.

Ilan Naiman, Nimrod Berman, Itai Pemper, Idan Arbiv, Gal Fadlon, and Omri Azencot. Utilizing image transforms and diffusion models for generative modeling of short and long time series. *Advances in Neural Information Processing Systems*, 37:121699–121730, 2024.

Sai Shankar Narasimhan, Shubhankar Agarwal, Oguzhan Akcin, sujay sanghavi, and Sandeep P. Chinchali. Time weaver: A conditional time series generation model. In *Forty-first International Conference on Machine Learning*, 2024.

Zhi Wen Soi, Chaoyi Zhu, Fouad Abiad, Aditya Shankar, Jeroen Galjaard, Huijuan Wang, and Lydia Chen. Timewak: Temporal chained-hashing watermark for time series data. *Advances in Neural Information Processing Systems*, 38:137917–137958, 2026.

Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction. *Advances in neural information processing systems*, 37:84839–84865, 2024.

Yongbo Wang, Haonan Wang, Guodong Mu, Ruixin Zhang, Jiaqi Chen, Jingyun Zhang, Jun Wang, Yuan Xie, Shouhong Ding, et al. Switchable token-specific codebook quantization for face image compression. *Advances in Neural Information Processing Systems*, 38:43486–43507, 2026.

Yuxin Wen, John Kirchenbauer, Jonas Geiping, and Tom Goldstein. Tree-rings watermarks: Invisible fingerprints for diffusion images. In *Advances in Neural Information Processing Systems*, volume 36, pp. 58047–58063, 2023. doi: 10.52202/075280-2529.

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in neural information processing systems*, 34:22419–22430, 2021.

Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. Timesnet: Temporal 2d-variation modeling for general time series analysis. In *The Eleventh International Conference on Learning Representations*, 2023.

Zijin Yang, Kai Zeng, Kejiang Chen, Han Fang, Weiming Zhang, and Nenghai Yu. Gaussian shading: Provable performance-lossless image watermarking for diffusion models. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12162–12171. IEEE, 2024.

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

Xinyu Yuan and Yan Qiao. Diffusion-ts: Interpretable diffusion for general time series generation. In *International Conference on Learning Representations*, volume 2024, pp. 41582–41610, 2024.

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, pp. 8980–8987, 2022.

Shuyi Zhang, Bin Guo, Anlan Dong, Jing He, Ziping Xu, and Song Xi Chen. Cautionary tales on air-quality improvement in beijing. *Proceedings. Mathematical, physical, and engineering sciences*, 473(2205):20170457, 2017.

Xuandong Zhao, Prabhanjan Ananth, Lei Li, and Yu-Xiang Wang. Provable robust watermarking for ai-generated text. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 43738–43772, 2024.

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, pp. 11106–11115, 2021.

# A Background on Time-series Watermarking

**Generation-time watermarking.** A generation-time watermark modifies the sampling process of a generative model so that its outputs carry a statistical signal tied to a secret key. In KGW (Kirchenbauer et al., 2023), the key selects a green subset $G_n$ of the vocabulary at each step $n$, and a bias $\delta > 0$ is added to the logit $\ell_{n,k}$ of every green token $k$,

$$\tilde{\ell}_{n,k} = \ell_{n,k} + \delta\, \mathbf{1}[k \in G_n]\,. \tag{9}$$

Watermarked sequences therefore contain more green tokens than expected by chance. For an observed sequence $\mathbf{x} = (x_1, \ldots, x_N)$, the detector reproduces the green sets from the key and computes the green ratio and its standardized score,

$$\hat{g}(\mathbf{x}) = \frac{1}{N}\sum_{n=1}^{N} \mathbf{1}[x_n \in G_n]\,, \qquad z(\mathbf{x}) = \frac{\hat{g}(\mathbf{x}) - \mu_0}{\sigma_0}, \tag{10}$$

where $\mu_0$ and $\sigma_0$ are the mean and standard deviation of $\hat{g}$ under the null hypothesis $H_0$ that $\mathbf{x}$ is not watermarked. The sequence is declared watermarked when $z(\mathbf{x}) > \eta$ for a fixed threshold $\eta > 0$. Under $H_0$, $z$ is approximately standard normal, so $\eta$ sets the false positive rate.

**Re-encoding in time series.** In text, the detector observes the tokens directly. A time-series generator outputs continuous values, so the detector must first re-encode the observed signal into discrete units before applying Eq. 10. VQ-based methods assign a code to each patch with a re-encoding network, and diffusion-based methods recover the initial noise by inverting the sampler (Wen et al., 2023; Yang et al., 2024; Soi et al., 2026). Re-encoding is the step that post-editing attacks affect: an attack changes the signal, and the recovered units may differ from those that were generated.

**Null drift and its two outcomes.** The threshold $\eta$ controls the false positive rate only if $\hat{g}$ of non-watermarked data stays at $\mu_0$ after the attack. When an attack changes the recovered units systematically, $\hat{g}$ of non-watermarked data drifts away from $\mu_0$, and the direction of the drift depends on the attack and the re-encoding, not on the watermark. An upward drift produces false positives: non-watermarked data is detected as watermarked. A downward drift produces a statistical anomaly: $z$ falls far below zero, a value that $H_0$ does not predict. An anomaly causes no false detection, but it shows that the null distribution no longer holds, so $\eta$ no longer corresponds to its nominal false positive rate. The same drift also acts on watermarked data, so a detection under such an attack does not reliably indicate the watermark. Table 5 summarizes the cases. The evaluation protocol used in our experiments is given in Appendix B.5.

Table 5: Interpretation of $z$ for watermarked and non-watermarked data.

| | $z \gg 0$ | $z \approx 0$ | $z \ll 0$ |
|---|---|---|---|
| Watermarked | Correct detection | Miss | Anomaly |
| Non-watermarked | False positive (FP) | Correct rejection (TN) | Anomaly (AN) |

**Inherited framework.** The green set $G^{(0)}$ of size $K/2$ is drawn from the key $\kappa$ and alternates along the sequence, $G_n = G^{(0)}$ at odd positions and $G_n = V \setminus G^{(0)}$ at even positions, and no bias is applied at the first $m$ positions of a window. Re-encoding uses a network $E^{\star}_{\rho}$ with the cross-attention trunk of the encoder $E_\phi$ and a $K$-way classifier per token, trained to predict the generated tokens from attacked windows. The score $z$ is standardized by the mean and standard deviation of $\hat{g}$ on non-watermarked windows under the same attack and the same detector.

# B EXPERIMENT DETAILS

## B.1 IMPLEMENTATION

Table 7 lists the hyperparameters of the L-VQVAE backbone shared by LVQMark and PHASELOCK, and Table 8 the settings of PHASELOCK. Two settings differ from L-VQVAE (Kim et al., 2026) for every method in this paper: the patch length equals the stride, $s = 6$, so that a token covers a whole number of hours on every dataset, and the output window is $T = 96$ time steps on the hourly datasets and 192 on the sub-hourly ones. LVQMark is trained at the codebook size of PHASELOCK and at about twice that size. TimeWak, TR, and GS are run on a Diffusion-TS generator trained by us on the same windows with the public configurations (Table 9), at their default setting with watermark key 217. All models were trained and evaluated on a single NVIDIA H100 (80 GB) or RTX PRO 6000 Blackwell (96 GB) GPU, and all methods of one dataset on the same GPU model.

Table 6 reports the throughput of generation and detection. Detection with PHASELOCK passes each window through the network once per candidate alignment, so its throughput is 0.35 to 0.45 of that of LVQMark, and its generation throughput is 0.68 to 0.95 of that of LVQMark. Both remain in the hundreds to thousands of windows per second, whereas Diffusion-TS is 19 to 26 times slower than PHASELOCK at generation and 49 to 54 times slower at detection.

Table 6: Throughput in windows per second (higher is faster): generation of a 10,000-window pool and detection of one 10,000-window condition, on one GPU. Diffusion-TS uses the TimeWak sampler and detector.

| | | **Generation** | | | **Detection** | | |
|---|---|---|---|---|---|---|---|
| **Dataset** | $T$ | PhaseLock | LVQMark | Diffusion-TS | PhaseLock | LVQMark | Diffusion-TS |
| ETTh | 96 | 1,302 | 1,370 | 50 | 2,299 | 5,102 | 43 |
| ETTm | 192 | 455 | 674 | 24 | 975 | 2,793 | 20 |
| AirQ-1 | 96 | 1,125 | 1,362 | 50 | 2,237 | 5,155 | 43 |
| AirQ-2 | 96 | 1,217 | 1,379 | 48 | 2,217 | 5,102 | 42 |
| AirQ-3 | 96 | 1,121 | 1,241 | 50 | 2,151 | 5,181 | 43 |
| AirQ-4 | 96 | 1,131 | 1,295 | 51 | 2,179 | 5,128 | 43 |
| AirQ-5 | 96 | 1,120 | 1,332 | 50 | 2,096 | 5,181 | 43 |
| AirQ-6 | 96 | 1,155 | 1,241 | 50 | 2,066 | 5,155 | 42 |
| Energy | 192 | 404 | 541 | 16 | 925 | 2,268 | 17 |
| Weather | 192 | 444 | 534 | 20 | 923 | 2,358 | 17 |

## B.2 DATASETS

Table 10 lists the datasets. ETTh and ETTm are the files ETTh1 and ETTm1 of ETDataset (Zhou et al., 2021), Energy is the appliances energy data of the UCI repository (Candanedo et al., 2017), and Weather is the Jena weather record distributed with the long-term forecasting benchmarks (Wu et al., 2021). AirQ-1 to AirQ-6 are the six stations Aotizhongxin, Changping, Gucheng, Huairou, Nongzhanguan, and Wanshouxigong of the Beijing Multi-Site Air Quality data (Zhang et al., 2017), one dataset per station. Exchange is the daily exchange rate data of LSTNet (Lai et al., 2018) and has no daily or weekly cycle.

For AirQ we drop the non-numeric columns and fill missing values from the previous or the next valid value. For Weather we remove the rows that break the 10-minute grid and replace the sentinel value $-9999$ by the previous valid value. Every variable is scaled to $[-1, 1]$ by its minimum and maximum over the series. Windows are formed at stride 1 and split 80/20 into training and test windows, and the test windows are the reference set of the quality metrics. The grid-aligned models use the training windows of $T + s$ time steps that start at a multiple of $s$. Time step 0 is the first record of the series.

Table 7: Hyperparameters of the L-VQVAE backbone shared by LVQMark and PHASELOCK. AirQ denotes each of the six stations.

| Parameter | ETTh | AirQ | ETTm | Energy | Weather | Exchange |
|---|---|---|---|---|---|---|
| *Data* | | | | | | |
| Number of variables $D$ | 7 | 11 | 7 | 28 | 21 | 8 |
| Sampling interval | 1 h | 1 h | 15 min | 10 min | 10 min | 1 day |
| Output window $T$ (time steps) | 96 | 96 | 192 | 192 | 192 | 96 |
| Tokens per output window | 16 | 16 | 32 | 32 | 32 | 16 |
| *Encoder and decoders* | | | | | | |
| $d_{\text{model}}$ | 512 | | | | | |
| Attention heads | 8 | | | | | |
| Encoder / local decoder / global decoder layers | 5 / 5 / 5 | | | | | |
| Patch length and stride $s$ | 6 | | | | | |
| Code dimension | 512 | | | | | |
| Codebook size $K$ (LVQMark) | 256, 512 | 512, 1024 | 1024, 2048 | 1152, 2048 | 1152, 2048 | 512 |
| Codebook size $K$ (PHASELOCK) | 256 | 512 | 1024 | 1152 | 1152 | 512 |
| Commitment weight | 0.25 | | | | | |
| Codebook EMA decay / dead-code re-seed | 0.8 / 100 steps | | | | | |
| Dropout | 0.0 | | | | | |
| *Autoregressive prior* | | | | | | |
| $n_{\text{embd}}$ / heads / layers | 1024 / 8 / 2 | | | | | |
| Attention / residual dropout | 0.1 / 0.1 | | | | | |
| Random replacement probability | 0.1 | | | | | |
| Lookback (tokens) | 6 | | | | | |
| Unbiased warm-up positions $m$ | 4 | | | | | |
| *Training* | | | | | | |
| Optimizer | AdamW, $\beta = (0.9,\ 0.99)$ | | | | | |
| Batch size | 128 | | | | | |
| Learning rate | $3 \times 10^{-4}$ | | | | | |
| Gradient clipping | 1.0 | | | | | |
| Steps: pretraining / global decoder / prior | 30,000 / 10,000 / 30,000 | | | | | |
| *Network* $E^{\star}_{\rho}$ | | | | | | |
| $d_{\text{model}}$ / heads / layers | 1024 / 16 / 3 | | | | | |
| Optimizer, learning rate | AdamW, $\beta = (0.9,\ 0.99)$, $1 \times 10^{-4}$ | | | | | |
| Training steps / validation interval | 50,000 / 5,000 | | | | | |
| Training pools | 4 generation seeds $\times$ 10,000 windows | | | | | |
| Attack augmentation | clean, offset, variable-axis crop, insert at $\alpha \in \{0.05,\ 0.30\}$ | | | | | |

Table 8: PHASELOCK settings. One phase group per patch of the period. For Exchange a period of 24 time steps is declared although the series has no such cycle (Appendix D.1).

| | ETTh | AirQ | ETTm | Energy | Weather | Exchange |
|---|---|---|---|---|---|---|
| Period $\tau$ (time steps) | 24 | 24 | 96 | 144 | 144 | 24 |
| Phase groups $P = \tau/s$ | 4 | 4 | 16 | 24 | 24 | 4 |
| Group size $M$ | 64 | 128 | 64 | 48 | 48 | 128 |
| Codebook size $K = PM$ | 256 | 512 | 1024 | 1152 | 1152 | 512 |
| Model window $T + s$ (time steps / tokens) | 102 / 17 | 102 / 17 | 198 / 33 | 198 / 33 | 198 / 33 | 102 / 17 |
| Output window $T$ (time steps) | 96 | 96 | 192 | 192 | 192 | 96 |
| Tokens scored at detection | 16 or 15 | 16 or 15 | 32 or 31 | 32 or 31 | 32 or 31 | 16 or 15 |
| Candidate alignments | 6 | | | | | |

### B.3 PERIOD AND PHASE

The period is one day for every periodic dataset, $\tau = 24$ for hourly, 96 for 15-minute, and 144 for 10-minute data, and the phase of a token whose first time step is $t$ is $\lfloor t/s \rfloor \bmod P$. When the cycle is not known in advance, $\tau$ can be read from the autocorrelation function as the lag of its first dominant peak. The periodicity of each dataset is examined in Appendix E.

### B.4 ATTACKS

The offset attack shifts each variable by $\alpha \bar{x}_d$, where $\bar{x}_d$ is the temporal mean of variable $d$ in the window. The insert attack replaces $\lfloor \alpha T \rfloor$ random time steps by uniform random values in the observed range of each variable. The crop attack keeps a rectangle of size $(1-\alpha)T \times (1-\alpha)D$ at a

Table 9: Diffusion-TS hyperparameters (generator of TimeWak, TR, and GS).

| Parameter | ETTh, ETTm, AirQ | Energy, Weather |
|---|---|---|
| Encoder / decoder layers | 3 / 2 | 4 / 3 |
| $d_{\text{model}}$ / heads | 64 / 4 | 96 / 4 |
| Diffusion steps / sampling steps | 500 / 250 | 1000 / 500 |
| Noise schedule / loss | cosine / $\ell_1$ | |
| Batch size | 128 | 64 |
| Training steps | 18,000 | 25,000 |
| Learning rate | $10^{-5}$, warm-up to $8 \times 10^{-4}$ in 500 steps, reduce on plateau | |
| Gradient accumulation / EMA decay | 2 / 0.995 | |
| Window $T$ | as for L-VQVAE (96 or 192) | |

Table 10: Datasets. $\tau$ is the period in time steps used to define the phase. Windows are counted at the model window $T + s$ on the patch grid (training / all).

| Dataset | Time steps | $D$ | Interval | $\tau$ | Grid windows | Source |
|---|---|---|---|---|---|---|
| ETTh | 17,420 | 7 | 1 h | 24 | 2,310 / 2,887 | https://github.com/zhouhaoyi/ETDataset |
| ETTm | 69,680 | 7 | 15 min | 96 | 9,265 / 11,581 | https://github.com/zhouhaoyi/ETDataset |
| AirQ-1 to 6 | 35,064 each | 11 | 1 h | 24 | 4,663 / 5,828 | https://archive.ics.uci.edu/dataset/501 |
| Energy | 19,735 | 28 | 10 min | 144 | 2,606 / 3,257 | https://archive.ics.uci.edu/dataset/374 |
| Weather | 52,417 | 21 | 10 min | 144 | 6,937 / 8,671 | https://github.com/thuml/Autoformer |
| Exchange | 7,588 | 8 | 1 day | 24 (declared) | 999 / 1,248 | https://github.com/laiguokun/multivariate-time-series-data |

random position and fills the rest with the per-variable midpoint of the retained part. All methods are attacked with the same random draws, and for the models with shifted output the attack is applied to the output window before the alignment search. The network $E_\rho^\star$ is trained with a variable-axis crop in place of the crop, which keeps a contiguous block of $(1 - \alpha)D$ variables over the full time axis.

### B.5 Evaluation protocol

**Pools.** One model is trained per dataset and method with training seed 12345. Every cell is computed from a non-watermarked and a watermarked pool of 10,000 windows with the watermark key 12345, repeated for the generation seeds 12345, 12346, and 12347 with all other seeds fixed. TR and GS do not depend on the generation seed and are reported for one seed. For Table 12, PhaseLock is retrained with the training seeds 12346 and 12347 and evaluated with the generation seed fixed to 12345 and the result averaged over the watermark keys 12345, 12346, and 12347.

**Detection.** Following previous papers (Soi et al., 2026; Kim et al., 2026), a draw of 1,000 windows is sampled without replacement from the watermarked pool, its mean green ratio is compared with the mean and the standard deviation of the non-watermarked pool by a one-sided $z$-test, and the draw is detected when $z > 3.09$. TPR is the fraction of 100 such draws that are detected and $z$ their mean. The same procedure on the non-watermarked pool, passed through the same output step, attack, and detector, gives the FPR and the non-watermarked $z$. A non-watermarked cell is AN when its mean $z$ is below $-3.09$, FP when its FPR exceeds 0.05 or its mean $z$ exceeds 3.09, and TN otherwise. For TR the statistic is the distance to the key pattern in the inverted noise, and both of its pools drift under crop and insert, so a TR row supports a detection claim only when its FPR is low. The strength $\delta^\star$ of a row is found by increasing $\delta$ in steps of 2 from 2 until the TPR is 1.00 on the clean pool and under the three attacks for all three generation seeds. For Table 2 the watermarked pools are scored with the keys 1 to 10, and the mean and the maximum are taken over the thirty combinations of attack and key.

**Sample quality.** Each metric is averaged over five evaluation runs with fixed seeds. Context-FID (Jeha et al., 2022) is the Fréchet distance between generated and reference windows in the embedding space of TS2Vec (Yue et al., 2022). The correlational score (Liao et al., 2020) is the absolute difference between the correlation matrices of generated and reference windows summed over variable pairs, with the discrepancy measure of Diffusion-TS (Yuan & Qiao, 2024), and is compared within a dataset only. The discriminative and the predictive score follow TimeGAN (Yoon et al., 2019), with a two-layer GRU classifier and a one-layer GRU predictor. On Weather and AirQ the discriminative score is computed after removing the variables whose real values are zero in

Table 11: Full results of Table 1: mean ± standard deviation over three generation seeds, with the FPR and the mean $z$ of the non-watermarked pool under each attack and all four quality metrics. $|z| > 3.09$ in red. Best in bold, second best underlined.

| Setting | | | | | | WM (TPR ↑) | | | Non-WM (FPR ↓ \| Z-score) | | | | | | Quality (↓) | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Data | Model | Method | P | K | $\delta$ | Offset | Crop | Insert | Offset | | Crop | | Insert | | C-FID ↓ | Corr. ↓ | Disc. ↓ | Pred. ↓ |
| ETTh | DiffusionTS | TR | – | – | – | **1.00** | **1.00** | **1.00** | **0.00** | +2.11 | 1.00 | +20.82 | 1.00 | +49.21 | 2.108 | 0.265 | 0.259 | 0.134 |
| | | GS | – | – | – | **1.00** | **1.00** | **1.00** | **0.00** | −3.04 | **0.00** | −19.42 | **0.00** | −9.75 | 5.510 | 0.291 | 0.363 | 0.174 |
| | | TimeWak | – | – | – | **1.00** | **1.00** | **1.00** | **0.00** | −0.43±0.53 | **0.00** | +0.45±0.65 | **0.00** | +0.84±0.36 | 0.691±0.021 | 0.158±0.013 | 0.138±0.002 | 0.115±0.000 |
| | L-VQVAE | LVQMark | – | 256 | 8 | **1.00** | **1.00** | **1.00** | **0.00** | −0.04±0.12 | **0.00** | −0.39±0.30 | **0.00** | +0.10±0.24 | 0.051±0.005 | 0.052±0.015 | 0.052±0.063 | 0.112±0.006 |
| | | LVQMark | – | 512 | 8 | **1.00** | **1.00** | **1.00** | **0.00** | +0.06±0.29 | **0.00** | −0.04±0.37 | **0.00** | +0.11±0.25 | **0.037**±0.005 | 0.045±0.008 | 0.221±0.142 | **0.111**±0.006 |
| | | **PhaseLock** | 4 | 256 | 10 | **1.00** | **1.00** | **1.00** | **0.00** | −0.02±0.09 | **0.00** | +0.01±0.06 | **0.00** | −0.01±0.19 | 0.039±0.005 | **0.044**±0.008 | **0.048**±0.056 | 0.112±0.007 |
| ETTm | DiffusionTS | TR | – | – | – | 0.00 | **1.00** | **1.00** | **0.00** | +0.78 | 1.00 | +18.89 | 1.00 | +76.95 | 2.359 | 0.253 | 0.259 | 0.133 |
| | | GS | – | – | – | **1.00** | **1.00** | **1.00** | 1.00 | +8.80 | 1.00 | +17.43 | **0.00** | −11.94 | 2.150 | 0.196 | 0.320 | 0.142 |
| | | TimeWak | – | – | – | **1.00** | **1.00** | **1.00** | **0.00** | −0.90±0.18 | 0.76 | +3.91±0.43 | **0.00** | −0.85±0.20 | 0.443±0.015 | 0.182±0.004 | 0.107±0.005 | 0.120±0.000 |
| | L-VQVAE | LVQMark | – | 1024 | 6 | **1.00** | **1.00** | **1.00** | **0.00** | +0.14±0.17 | 0.01 | −0.08±0.31 | **0.00** | +0.31±0.27 | 0.063±0.005 | 0.029±0.007 | **0.012**±0.006 | **0.117**±0.007 |
| | | LVQMark | – | 2048 | 10 | **1.00** | **1.00** | **1.00** | **0.00** | −0.00±0.03 | **0.00** | −0.32±0.67 | **0.00** | −0.42±0.20 | 0.060±0.005 | **0.028**±0.006 | 0.078±0.096 | 0.118±0.005 |
| | | **PhaseLock** | 16 | 1024 | 10 | **1.00** | **1.00** | **1.00** | **0.00** | −0.03±0.09 | **0.00** | +0.14±0.18 | **0.00** | −0.05±0.18 | **0.045**±0.002 | 0.035±0.008 | 0.109±0.142 | **0.117**±0.007 |
| AirQ-1 | DiffusionTS | TR | – | – | – | 0.00 | **1.00** | **1.00** | **0.00** | +0.15 | 1.00 | +11.76 | 1.00 | +7.10 | 3.222 | 0.791 | 0.408 | 0.066 |
| | | GS | – | – | – | **1.00** | **1.00** | **1.00** | 0.06 | +1.73 | 1.00 | +28.87 | 1.00 | +38.02 | 10.068 | 0.976 | 0.441 | 0.072 |
| | | TimeWak | – | – | – | **1.00** | **1.00** | **1.00** | **0.00** | −0.44±0.14 | 1.00 | +7.44±0.36 | 0.54 | +3.22±0.20 | 0.463±0.005 | 0.195±0.006 | 0.159±0.003 | **0.060**±0.000 |
| | L-VQVAE | LVQMark | – | 512 | 8 | **1.00** | **1.00** | **1.00** | **0.00** | −0.15±0.19 | **0.00** | +0.16±0.38 | **0.00** | −0.22±0.33 | 0.109±0.018 | 0.132±0.033 | 0.239±0.074 | **0.060**±0.000 |
| | | LVQMark | – | 1024 | 8 | **1.00** | **1.00** | **1.00** | **0.00** | −0.17±0.19 | **0.00** | −0.04±0.25 | **0.00** | +0.03±0.16 | 0.064±0.008 | 0.129±0.020 | 0.169±0.046 | **0.060**±0.000 |
| | | **PhaseLock** | 4 | 512 | 10 | **1.00** | **1.00** | **1.00** | **0.00** | +0.03±0.05 | **0.00** | −0.10±0.15 | **0.00** | −0.08±0.13 | **0.025**±0.003 | **0.115**±0.016 | **0.071**±0.026 | **0.060**±0.001 |
| AirQ-2 | DiffusionTS | TR | – | – | – | 0.00 | **1.00** | **1.00** | **0.00** | +0.15 | 1.00 | +7.71 | 1.00 | +4.67 | 2.855 | 0.658 | 0.423 | 0.076 |
| | | GS | – | – | – | **1.00** | **1.00** | **1.00** | **0.00** | −2.74 | 1.00 | +5.08 | 1.00 | +26.17 | 9.461 | 1.135 | 0.443 | 0.087 |
| | | TimeWak | – | – | – | **1.00** | **1.00** | **1.00** | **0.00** | −0.26±0.11 | 0.48 | +2.95±0.99 | 0.02 | +0.63±0.91 | 0.550±0.009 | 0.231±0.015 | 0.186±0.004 | 0.070±0.000 |
| | L-VQVAE | LVQMark | – | 512 | 6 | **1.00** | **1.00** | **1.00** | **0.00** | −0.02±0.22 | **0.00** | −0.24±0.20 | **0.00** | −0.19±0.19 | 0.114±0.008 | 0.202±0.032 | 0.222±0.092 | **0.069**±0.002 |
| | | LVQMark | – | 1024 | 8 | **1.00** | **1.00** | **1.00** | **0.00** | −0.23±0.18 | 0.01 | +0.33±0.40 | **0.00** | −0.07±0.33 | 0.078±0.007 | 0.101±0.018 | 0.282±0.100 | **0.069**±0.001 |
| | | **PhaseLock** | 4 | 512 | 10 | **1.00** | **1.00** | **1.00** | 0.01 | +0.01±0.11 | **0.00** | −0.19±0.18 | **0.00** | −0.09±0.12 | **0.026**±0.003 | **0.093**±0.026 | **0.061**±0.019 | **0.069**±0.001 |
| AirQ-3 | DiffusionTS | TR | – | – | – | 0.00 | **1.00** | **1.00** | **0.00** | +0.11 | 1.00 | +10.71 | 1.00 | +5.74 | 3.411 | 1.101 | 0.376 | 0.055 |
| | | GS | – | – | – | **1.00** | **1.00** | **1.00** | **0.00** | −0.09 | **0.00** | +0.71 | 1.00 | +27.19 | 9.297 | 1.099 | 0.428 | 0.074 |
| | | TimeWak | – | – | – | **1.00** | **1.00** | **1.00** | **0.00** | −1.44±0.23 | 0.06 | +1.45±0.42 | 0.03 | +1.20±0.58 | 0.490±0.005 | 0.296±0.022 | 0.144±0.002 | 0.051±0.000 |
| | L-VQVAE | LVQMark | – | 512 | 6 | **1.00** | **1.00** | **1.00** | **0.00** | −0.01±0.13 | 0.01 | +0.18±0.02 | **0.00** | −0.12±0.52 | 0.074±0.010 | 0.120±0.033 | 0.152±0.066 | **0.050**±0.001 |
| | | LVQMark | – | 1024 | 6 | **1.00** | **1.00** | **1.00** | **0.00** | −0.14±0.04 | **0.00** | +0.02±0.42 | **0.00** | +0.06±0.41 | 0.058±0.003 | 0.136±0.033 | 0.106±0.056 | **0.050**±0.001 |
| | | **PhaseLock** | 4 | 512 | 10 | **1.00** | **1.00** | **1.00** | **0.00** | −0.07±0.16 | 0.01 | −0.01±0.07 | **0.00** | −0.01±0.53 | **0.036**±0.005 | **0.093**±0.027 | **0.088**±0.008 | **0.050**±0.001 |
| AirQ-4 | DiffusionTS | TR | – | – | – | 0.00 | **1.00** | **1.00** | **0.00** | +0.11 | 1.00 | +15.40 | 1.00 | +7.93 | 3.278 | 0.486 | 0.404 | 0.053 |
| | | GS | – | – | – | **1.00** | **1.00** | **1.00** | **0.00** | −2.71 | **0.00** | −0.21 | 1.00 | +20.06 | 9.703 | 0.715 | 0.394 | 0.060 |
| | | TimeWak | – | – | – | **1.00** | **1.00** | **1.00** | **0.00** | −0.53±0.16 | **0.00** | −2.87±0.22 | 0.04 | +1.36±0.37 | 0.676±0.004 | 0.223±0.009 | 0.187±0.003 | **0.050**±0.000 |
| | L-VQVAE | LVQMark | – | 512 | 8 | **1.00** | **1.00** | **1.00** | **0.00** | −0.13±0.30 | **0.00** | −0.19±0.17 | **0.00** | −0.12±0.22 | 0.110±0.013 | 0.162±0.020 | 0.279±0.071 | **0.050**±0.001 |
| | | LVQMark | – | 1024 | 8 | **1.00** | **1.00** | **1.00** | **0.00** | −0.06±0.27 | **0.00** | −0.37±0.35 | 0.01 | +0.23±0.36 | 0.079±0.013 | 0.138±0.025 | 0.212±0.041 | **0.050**±0.000 |
| | | **PhaseLock** | 4 | 512 | 10 | **1.00** | **1.00** | **1.00** | **0.00** | +0.07±0.08 | **0.00** | −0.15±0.06 | **0.00** | −0.07±0.25 | **0.025**±0.003 | **0.110**±0.020 | **0.075**±0.009 | **0.050**±0.001 |
| AirQ-5 | DiffusionTS | TR | – | – | – | 0.00 | **1.00** | **1.00** | **0.00** | +0.11 | 1.00 | +12.31 | 1.00 | +8.52 | 3.771 | 0.654 | 0.404 | 0.071 |
| | | GS | – | – | – | **1.00** | **1.00** | **1.00** | **0.00** | −1.36 | 0.94 | +4.67 | 1.00 | +27.83 | 11.343 | 0.934 | 0.447 | 0.082 |
| | | TimeWak | – | – | – | **1.00** | **1.00** | **1.00** | **0.00** | +0.08±0.13 | **0.00** | −0.29±0.71 | 0.02 | +0.63±0.90 | 0.553±0.005 | 0.198±0.005 | 0.168±0.004 | **0.067**±0.000 |
| | L-VQVAE | LVQMark | – | 512 | 8 | **1.00** | **1.00** | **1.00** | **0.00** | −0.12±0.13 | **0.00** | +0.06±0.43 | **0.00** | −0.06±0.15 | 0.128±0.011 | 0.172±0.026 | 0.233±0.028 | 0.068±0.000 |
| | | LVQMark | – | 1024 | 8 | **1.00** | **1.00** | **1.00** | **0.00** | −0.03±0.07 | **0.00** | −0.08±0.21 | **0.00** | −0.01±0.30 | 0.071±0.009 | 0.107±0.019 | 0.246±0.070 | 0.068±0.000 |
| | | **PhaseLock** | 4 | 512 | 10 | **1.00** | **1.00** | **1.00** | **0.00** | −0.11±0.17 | 0.01 | +0.11±0.32 | **0.00** | −0.13±0.16 | **0.023**±0.003 | **0.093**±0.033 | **0.124**±0.012 | **0.067**±0.001 |
| AirQ-6 | DiffusionTS | TR | – | – | – | 0.00 | **1.00** | **1.00** | **0.00** | +0.12 | 1.00 | +11.79 | 1.00 | +6.66 | 3.187 | 0.495 | 0.396 | 0.056 |
| | | GS | – | – | – | **1.00** | **1.00** | **1.00** | **0.00** | −2.66 | 1.00 | +7.69 | 1.00 | +29.37 | 10.189 | 1.122 | 0.436 | 0.059 |
| | | TimeWak | – | – | – | **1.00** | **1.00** | **1.00** | **0.00** | −1.51±0.40 | **0.00** | −2.73±0.93 | 0.14 | +1.99±0.59 | 0.428±0.006 | 0.201±0.008 | 0.159±0.004 | 0.051±0.000 |
| | L-VQVAE | LVQMark | – | 512 | 12 | **1.00** | **1.00** | **1.00** | **0.00** | +0.03±0.27 | **0.00** | +0.08±0.22 | **0.00** | −0.10±0.31 | 0.129±0.009 | 0.253±0.031 | 0.240±0.033 | 0.051±0.000 |
| | | LVQMark | – | 1024 | 6 | **1.00** | **1.00** | **1.00** | **0.00** | −0.10±0.04 | 0.01 | +0.06±0.38 | **0.00** | −0.06±0.27 | 0.054±0.005 | 0.111±0.024 | 0.195±0.022 | **0.050**±0.000 |
| | | **PhaseLock** | 4 | 512 | 10 | **1.00** | **1.00** | **1.00** | **0.00** | −0.01±0.10 | **0.00** | +0.19±0.12 | **0.00** | −0.15±0.04 | **0.023**±0.002 | **0.087**±0.018 | **0.101**±0.008 | **0.050**±0.001 |
| Energy | DiffusionTS | TR | – | – | – | **1.00** | **1.00** | **1.00** | 1.00 | +10.39 | 1.00 | +87.68 | 1.00 | +170.18 | 0.497 | 1.521 | 0.485 | 0.273 |
| | | GS | – | – | – | **1.00** | **1.00** | **1.00** | **0.00** | −25.06 | **0.00** | −1.83 | **0.00** | −6.01 | 6.438 | 4.854 | 0.448 | 0.267 |
| | | TimeWak | – | – | – | **1.00** | **1.00** | **1.00** | 0.10 | +1.65±0.40 | 0.82 | +3.93±0.37 | 0.02 | +1.02±0.52 | 0.155±0.006 | 1.621±0.210 | 0.223±0.025 | 0.248±0.000 |
| | L-VQVAE | LVQMark | – | 1152 | 4 | **1.00** | **1.00** | **1.00** | **0.00** | −0.04±0.08 | **0.00** | +0.27±0.16 | **0.00** | +0.14±0.22 | 0.506±0.021 | 0.853±0.205 | 0.500±0.000 | 0.341±0.006 |
| | | LVQMark | – | 2048 | 6 | **1.00** | **1.00** | **1.00** | **0.00** | −0.08±0.14 | **0.00** | +0.19±0.14 | **0.00** | −0.25±0.28 | 0.325±0.014 | **0.802**±0.216 | 0.500±0.001 | 0.308±0.005 |
| | | **PhaseLock** | 24 | 1152 | 12 | **1.00** | **1.00** | **1.00** | **0.00** | +0.02±0.06 | **0.00** | +0.20±0.13 | **0.00** | +0.00±0.12 | **0.070**±0.004 | 0.876±0.203 | **0.217**±0.025 | **0.244**±0.001 |
| Weather | DiffusionTS | TR | – | – | – | 0.00 | 0.00 | **1.00** | **0.00** | +0.18 | 0.98 | +3.13 | 1.00 | +5.71 | 3.006 | 1.370 | 0.354 | 0.030 |
| | | GS | – | – | – | **1.00** | **1.00** | **1.00** | 1.00 | +5.85 | **0.00** | −12.03 | 1.00 | +20.32 | 6.858 | 1.523 | 0.427 | 0.038 |
| | | TimeWak | – | – | – | **1.00** | **1.00** | **1.00** | **0.00** | −0.36±0.13 | **0.00** | −3.35±0.22 | 0.31 | +2.62±0.34 | 0.670±0.016 | 0.758±0.043 | 0.148±0.005 | **0.026**±0.000 |
| | L-VQVAE | LVQMark | – | 1152 | 6 | **1.00** | **1.00** | **1.00** | **0.00** | +0.25±0.05 | 0.01 | −0.14±0.22 | **0.00** | −0.06±0.24 | 0.105±0.006 | 0.345±0.058 | 0.385±0.028 | **0.026**±0.000 |
| | | LVQMark | – | 2048 | 8 | **1.00** | **1.00** | **1.00** | **0.00** | −0.07±0.43 | 0.02 | +0.24±0.67 | **0.00** | −0.09±0.09 | 0.076±0.004 | **0.272**±0.048 | 0.324±0.080 | **0.026**±0.001 |
| | | **PhaseLock** | 24 | 1152 | 8 | **1.00** | **1.00** | **1.00** | **0.00** | +0.08±0.10 | **0.00** | +0.21±0.13 | **0.00** | +0.18±0.17 | **0.035**±0.002 | 0.350±0.076 | **0.143**±0.038 | **0.026**±0.001 |

almost every window. In Table 11 the ± of a quality column is the standard deviation over the fifteen evaluation runs and that of a $z$ column over the three generation seeds.

# C Additional Results

## C.1 Full results

Table 11 extends Table 1 with the standard deviation of every entry, the FPR and the mean $z$ of the non-watermarked pool under each attack, and the discriminative and the predictive score. The rows, the strengths $\delta^\star$, and the ranking rules are those of the main table (Appendix B.5).

### C.2 WATERMARK STRENGTH SWEEP ON ALL DATASETS

Figure 5 shows the sweep of Figure 3 on all ten datasets. The $\delta^\star$ of a LVQMark or PHASELOCK row in Table 1 is the first strength at which the worst-case TPR of the corresponding curve reaches 1.00 on all three generation seeds. The sweep covers $\delta \leq 10$. The $\delta^\star$ of PHASELOCK on Energy (12) lies beyond the sweep and comes from the search of Appendix B.5.

### C.3 ROBUSTNESS TO THE TRAINING SEED

Table 12 reports PHASELOCK retrained with three training seeds on every dataset, with every training stage repeated per seed and everything else fixed, evaluated at the $\delta^\star$ of its row in Table 1 under the protocol of Appendix B.5. The TPR is 1.00 under every attack for every training seed on every dataset, and the clean $z$ lies between $+7.78$ and $+15.39$ with a standard deviation of at most 1.18 over seeds. The standard deviation of the Context-FID of the non-watermarked pool over training seeds is at most 0.006, which is smaller than the gap in Context-FID between PHASELOCK and LVQMark at the same codebook size in Table 1 on every dataset.

Table 12: Robustness to the training seed. PHASELOCK retrained with three training seeds on every dataset, where each seed repeats the training of the tokenizer, the global decoder, the autoregressive prior, and the network $E^\star_\rho$, and everything else follows Appendix B.5. Each cell is the mean $\pm$ standard deviation over training seeds after averaging over the three watermark keys, at the $\delta^\star$ of the row in Table 1. Quality is the Context-FID and the correlational score of the non-watermarked pool. TPR is under the three attacks at strength 0.30, and $z$ is the score of the clean watermarked pool.

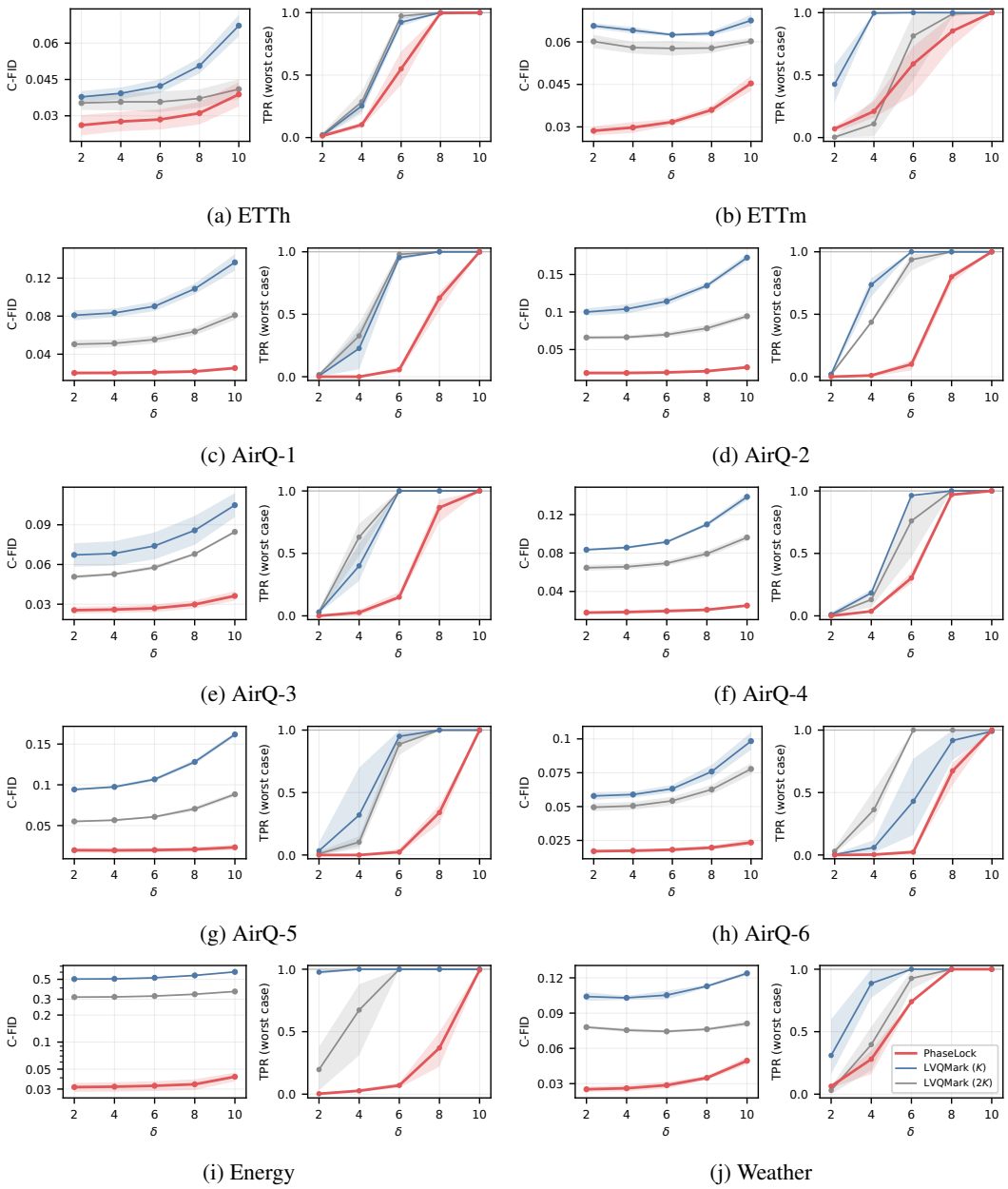
| Setting | | | NoWM Quality (↓) | | Watermarked (TPR ↑) | | | Clean |
|---|---|---|---|---|---|---|---|---|
| **Dataset** | **P** | **K** | **C-FID** | **Corr.** | **Offset** | **Crop** | **Insert** | $z$ |
| ETTh | 4 | 256 | $0.037_{\pm 0.006}$ | $0.043_{\pm 0.004}$ | $1.00_{\pm 0.00}$ | $1.00_{\pm 0.00}$ | $1.00_{\pm 0.00}$ | $+13.86_{\pm 0.52}$ |
| ETTm | 16 | 1024 | $0.025_{\pm 0.003}$ | $0.029_{\pm 0.007}$ | $1.00_{\pm 0.00}$ | $1.00_{\pm 0.00}$ | $1.00_{\pm 0.00}$ | $+13.47_{\pm 1.18}$ |
| AirQ-1 | 4 | 512 | $0.020_{\pm 0.003}$ | $0.090_{\pm 0.017}$ | $1.00_{\pm 0.00}$ | $1.00_{\pm 0.00}$ | $1.00_{\pm 0.00}$ | $+8.64_{\pm 0.26}$ |
| AirQ-2 | 4 | 512 | $0.026_{\pm 0.006}$ | $0.083_{\pm 0.009}$ | $1.00_{\pm 0.00}$ | $1.00_{\pm 0.00}$ | $1.00_{\pm 0.00}$ | $+9.85_{\pm 0.84}$ |
| AirQ-3 | 4 | 512 | $0.022_{\pm 0.002}$ | $0.100_{\pm 0.010}$ | $1.00_{\pm 0.00}$ | $1.00_{\pm 0.00}$ | $1.00_{\pm 0.00}$ | $+9.25_{\pm 0.64}$ |
| AirQ-4 | 4 | 512 | $0.022_{\pm 0.006}$ | $0.096_{\pm 0.014}$ | $1.00_{\pm 0.00}$ | $1.00_{\pm 0.00}$ | $1.00_{\pm 0.00}$ | $+10.04_{\pm 0.39}$ |
| AirQ-5 | 4 | 512 | $0.023_{\pm 0.000}$ | $0.084_{\pm 0.011}$ | $1.00_{\pm 0.00}$ | $1.00_{\pm 0.00}$ | $1.00_{\pm 0.00}$ | $+8.48_{\pm 0.47}$ |
| AirQ-6 | 4 | 512 | $0.018_{\pm 0.001}$ | $0.083_{\pm 0.008}$ | $1.00_{\pm 0.00}$ | $1.00_{\pm 0.00}$ | $1.00_{\pm 0.00}$ | $+8.60_{\pm 0.63}$ |
| Energy | 24 | 1152 | $0.035_{\pm 0.006}$ | $0.709_{\pm 0.059}$ | $1.00_{\pm 0.00}$ | $1.00_{\pm 0.00}$ | $1.00_{\pm 0.00}$ | $+15.39_{\pm 1.01}$ |
| Weather | 24 | 1152 | $0.024_{\pm 0.005}$ | $0.294_{\pm 0.046}$ | $1.00_{\pm 0.00}$ | $1.00_{\pm 0.00}$ | $1.00_{\pm 0.00}$ | $+7.78_{\pm 0.54}$ |

### C.4 ALIGNMENT AND PHASE RECOVERY

Table 13 reports how often the phase-aware re-encoding recovers the shift and the phase of a watermarked window in the configurations of Table 1. Alignment accuracy is the fraction of windows whose candidate alignment $\hat{j}$ equals the true candidate $j^*$, with chance $1/s$. Phase recovery accuracy is the fraction of windows whose estimated phase $\hat{\varphi}_1$ equals the phase of the first generated token, with chance $1/P$. On the clean pool the alignment accuracy lies between 0.82 (ETTm) and 0.99 (Energy) and the phase recovery accuracy between 0.93 and 0.96. Offset and crop leave both nearly unchanged. Insert lowers the alignment accuracy on the hourly and 15-minute datasets, to 0.50 on ETTm and to 0.70 to 0.81 on ETTh and the AirQ stations, while the phase recovery accuracy stays at or above 0.84. The TPR under insert in Table 1 is nevertheless 1.00.

## D APERIODIC DATA AND WRONG DECLARED PERIODS

PHASELOCK depends on the period of the series. The phase partition is defined by it, and the detector reads the alignment and the phase of an observed window from the phase structure of the codes. This appendix examines what happens when the declared period is not present in the data. Appendix D.1 runs the method on a series with no cycle, and Appendix D.2 trains it on periodic series with a wrong declared period and measures the phase structure of the resulting codebooks.




(a) ETTh (b) ETTm

(c) AirQ-1 (d) AirQ-2

(e) AirQ-3 (f) AirQ-4

(g) AirQ-5 (h) AirQ-6

(i) Energy (j) Weather

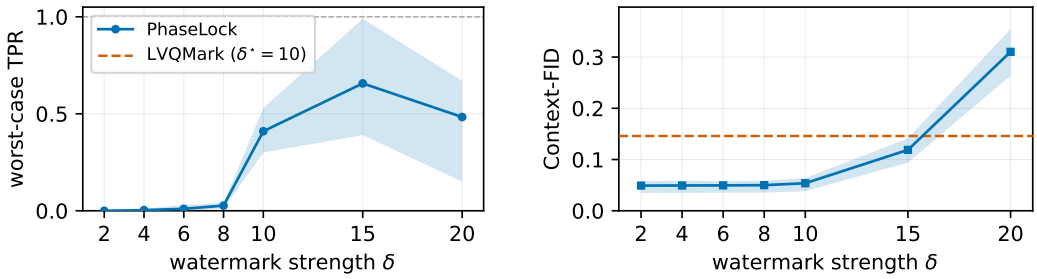
Figure 5: Watermark strength sweep on all ten datasets. Panels as in Figure 3: left, Context-FID of the watermarked pool, right, worst-case TPR. Shaded bands are the range over three generation seeds.

Table 13: Alignment accuracy and phase recovery accuracy of the phase-aware re-encoding on the watermarked pools of Table 1, at the $\delta^\star$ of each row, mean over three generation seeds. Alignment: fraction of windows whose shift is recovered (chance $1/s = 0.17$). Phase: fraction of windows whose phase $\varphi_1$ is recovered (chance $1/P$).

| | | | **Alignment accuracy (↑)** | | | | **Phase recovery accuracy (↑)** | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Dataset** | $P$ | $K$ | Clean | Offset | Crop | Insert | Clean | Offset | Crop | Insert |
| ETTh | 4 | 256 | 0.90 | 0.90 | 0.89 | 0.81 | 0.95 | 0.95 | 0.94 | 0.93 |
| ETTm | 16 | 1024 | 0.82 | 0.86 | 0.82 | 0.50 | 0.93 | 0.94 | 0.93 | 0.84 |
| AirQ-1 | 4 | 512 | 0.93 | 0.93 | 0.91 | 0.73 | 0.96 | 0.96 | 0.95 | 0.92 |
| AirQ-2 | 4 | 512 | 0.95 | 0.95 | 0.93 | 0.78 | 0.96 | 0.96 | 0.95 | 0.93 |
| AirQ-3 | 4 | 512 | 0.92 | 0.92 | 0.89 | 0.70 | 0.96 | 0.95 | 0.95 | 0.92 |
| AirQ-4 | 4 | 512 | 0.92 | 0.92 | 0.90 | 0.74 | 0.96 | 0.96 | 0.95 | 0.93 |
| AirQ-5 | 4 | 512 | 0.94 | 0.94 | 0.90 | 0.73 | 0.96 | 0.96 | 0.95 | 0.93 |
| AirQ-6 | 4 | 512 | 0.93 | 0.93 | 0.92 | 0.71 | 0.96 | 0.96 | 0.95 | 0.92 |
| Energy | 24 | 1152 | 0.99 | 0.99 | 0.86 | 0.99 | 0.96 | 0.96 | 0.93 | 0.96 |
| Weather | 24 | 1152 | 0.97 | 0.97 | 0.96 | 0.90 | 0.94 | 0.94 | 0.93 | 0.93 |

### D.1 Aperiodic control

**Setting.** On Exchange we train LVQMark with $K = 512$ and PhaseLock with a declared period of 24 time steps ($P = 4$, $M = 128$, $K = 512$), the configuration of the hourly datasets, although the series has no daily or weekly cycle. PhaseLock is evaluated at $\delta \in \{2, 4, 6, 8, 10, 15, 20\}$ and LVQMark at $\delta \in \{8, 10, 15, 20\}$, each with the three generation seeds and the protocol of Appendix B.5. The worst-case TPR of a generation seed is the smallest TPR over the clean pool and the three attacks at $\alpha = 0.30$. Figure 6 plots the mean over the three seeds as the line and their range as the band, for the worst-case TPR and for the Context-FID of the watermarked pool.




(a) Worst-case TPR.

(b) Context-FID of the watermarked pool.

Figure 6: Exchange, a series with no daily or weekly cycle, as a function of the watermark strength $\delta$. PhaseLock is trained with a declared period of $\tau = 24$ time steps. (a) The smallest TPR over the clean pool and the three attacks at $\alpha = 0.30$. LVQMark is shown at its minimal strength $\delta^\star = 10$. (b) Context-FID of the watermarked pool. Lines are the mean and shaded bands the range over three generation seeds.

**Result.** No $\delta$ brings the worst-case TPR of PhaseLock to 1.00, whereas LVQMark on the same data is detectable at $\delta^\star = 10$. The bias is embedded, since the green ratio of the generated tokens rises to 0.96 at $\delta = 20$, but it is not recovered: from $\delta = 10$ on, the alignment and the phase recovery accuracy of $\Phi$ fall toward chance as $\delta$ grows, and a token read at the wrong alignment or in the wrong group contributes no green count. A larger $\delta$ only degrades the samples, and at $\delta = 20$ the Context-FID of PhaseLock exceeds that of LVQMark at its $\delta^\star$.

### D.2 Phase structure of the codebook under a wrong declared period

**Setting.** To separate the effect of the period from that of the dataset, we retrain PhaseLock on two periodic datasets with a declared period that does not match their daily cycle. On AirQ-1 the declared period is 18 hours ($P = 3$, $M = 170$, $K = 510$) and 30 hours ($P = 5$, $M = 102$,

$K = 510$) in place of the true 24 hours ($P = 4$, $M = 128$, $K = 512$). On Energy it is 23 hours ($P = 23$, $M = 50$, $K = 1{,}150$) and 25 hours ($P = 25$, $M = 46$, $K = 1{,}150$) in place of the true 24 hours ($P = 24$, $M = 48$, $K = 1{,}152$). The codebook size must be even and divisible by $P$, which fixes $K$ in each case. Everything else, the detector included, is unchanged.

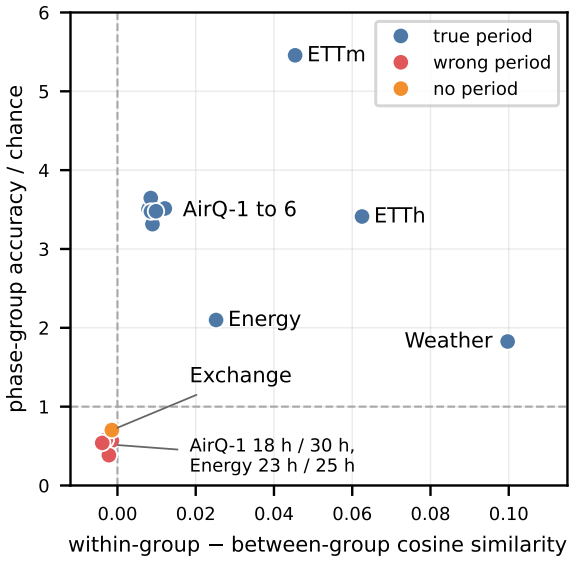



Figure 7: Separability of the phase groups, one point per trained codebook. Vertical axis: accuracy of a linear classifier predicting the phase group from the code vector, relative to chance. Horizontal axis: mean cosine similarity of code vectors within a group minus that between groups.

**Separability of the phase groups.** For each trained codebook we ask whether the phase group of a code can be read from its vector alone. We use the codes in use, those whose exponential moving average usage count exceeds 0.5, and measure two quantities. The first is the accuracy of a ridge-regularized linear classifier that predicts the group of a code from its 512-dimensional vector, estimated by 5-fold cross-validation and divided by the mean accuracy under 200 random permutations of the group labels that preserve the group sizes, so that a value of 1 is chance. The second is the mean cosine similarity between code vectors of the same group minus the mean cosine similarity between code vectors of different groups, so that a value of 0 is the absence of group structure. No data window and no generated window enters this analysis. Figure 7 plots the two quantities, one point per trained codebook, for the ten periodic datasets with their true period, for AirQ-1 and Energy with the wrong declared periods above, and for Exchange with the declared period of Appendix D.1.

**Result.** With the true period the groups are separable on both measures on all ten datasets. With a wrong declared period, and on Exchange, both measures lie at chance. A phase-partitioned codebook thus acquires phase structure only when the declared period is present in the data, and PHASE-LOCK is a method for series with such a period.

## E PERIODICITY ANALYSIS

This appendix supports Section 4.6. Appendix E.1 measures the daily periodicity of every variable and selects the most periodic variable of each dataset. Appendix E.2 defines the periodic power ratio (PPR) and the distance of Table 4, and shows the percentile comparison of Figure 4 on all ten datasets.

Table 14: Periodicity of the datasets. Left: number of variables with a daily line, mean autocorrelation at lag $\tau$, and mean daily share over all $D$ variables. Right: the most periodic variable with its autocorrelation and daily share. Brackets give the variable with the strongest spectral line where it differs.

| Setting | | | Dataset summary | | | Most periodic variable | | |
|---|---|---|---|---|---|---|---|---|
| **Dataset** | $D$ | $\tau$ | **Daily line** ($\#/D$) | **ACF**($\tau$) | **Daily share (%)** | **Variable** | **ACF**($\tau$) | **Daily share (%)** |
| ETTh | 7 | 24 | 7/7 | 0.80 | 22.6 | MUFL | 0.79 | 54.2 |
| ETTm | 7 | 96 | 7/7 | 0.80 | 22.6 | MUFL | 0.80 | 54.1 |
| AirQ-1 | 11 | 24 | 10/11 | 0.55 | 11.6 | TEMP | 0.96 | 40.2 |
| AirQ-2 | 11 | 24 | 8/11 | 0.55 | 9.6 | TEMP | 0.95 | 38.9 |
| AirQ-3 | 11 | 24 | 10/11 | 0.56 | 11.6 | TEMP | 0.96 | 39.8 |
| AirQ-4 | 11 | 24 | 10/11 | 0.54 | 10.5 | TEMP | 0.96 | 45.3 |
| AirQ-5 | 11 | 24 | 10/11 | 0.54 | 11.5 | TEMP | 0.96 | 38.1 |
| AirQ-6 | 11 | 24 | 10/11 | 0.55 | 11.2 | TEMP | 0.95 | 38.5 |
| Energy | 28 | 144 | 21/28 | 0.64 | 13.9 | T1 [T6] | 0.85 | 24.2 |
| Weather | 21 | 144 | 18/21 | 0.65 | 27.9 | Tlog [max.PAR] | 0.91 | 47.4 |
| Exchange | 8 | 24* | 0/8 | 0.98* | 0.1 | None | – | – |

*Exchange is observed daily, so $\tau = 24$ is a reference lag of 24 days. No variable has a spectral line at any lag (largest prominence 2.7). Its autocorrelation at lag 24 reflects the slow decay of a random walk, not a periodic component.

### E.1 Periodicity of the datasets

**Setting.** For each variable we compute, on the whole series, the autocorrelation at lag $\tau$ and the share of spectral power at the daily frequency $1/\tau$, using a Welch spectrum whose segment length is a multiple of $\tau$. A variable has a daily line when the power at $1/\tau$ exceeds three times the median power of the neighboring frequencies. Dataset means are taken over all $D$ variables. The most periodic variable of a dataset is the one with the highest mean PPR (Appendix E.2) over the real training windows, and it is fixed across methods and generation seeds.

**Result.** Daily lines are detected in every variable of ETTh and ETTm and in most variables of AirQ, Energy, and Weather, and their strength varies across variables. On ETTh the daily share of MUFL is 54.2%, against 22.6% averaged over all variables, and on the AirQ stations the daily share of TEMP is 38.1% to 45.3%, against 9.6% to 11.6%. The highest mean PPR over training windows is attained by MUFL on ETTh and ETTm, TEMP on all six AirQ stations, T1 on Energy, and Tlog on Weather, and these variables are used in Appendix E.2 and Section 4.6.

### E.2 Periodic power ratio and its distance

**Periodic power ratio.** For a window $x = (x_1, \dots, x_T)$ of the selected variable, let $\tilde{x}_t = x_t - \bar{x}$ be the centered values and $v(x)$ their variance. The PPR is the power at the daily frequency relative to the variance of the window,

$$f_{\mathrm{PPR}}(x) = \frac{2\left|\sum_{t=1}^{T} \tilde{x}_t\, e^{-2\pi \mathrm{i} t/\tau}\right|^2}{T^2\, v(x)}. \tag{11}$$

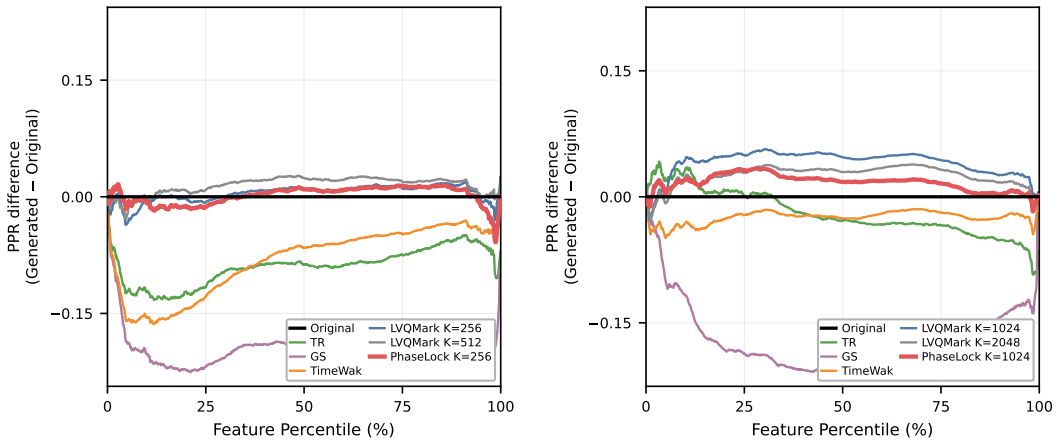
When $T/\tau$ is an integer, as on ETTh, ETTm, and AirQ, this is the fraction of the window variance at the daily fundamental frequency. On Energy and Weather, where $T/\tau = 4/3$, it is the normalized Fourier power at $1/\tau$. The variance is floored at a small $\epsilon$ in the computation, and windows with variance below $\epsilon$ are excluded.

**Distance.** Let $Q_R$ be the empirical quantile function of the PPR over the real test windows and $Q_{G,s}$ that over the watermarked pool of generation seed $s$, at the $\delta^\star$ of Table 1 and without attack. The distance of Table 4 is the one-dimensional Wasserstein distance averaged over seeds,

$$\overline{D} = \frac{1}{S}\sum_{s=1}^{S} \int_0^1 |Q_{G,s}(u) - Q_R(u)|\, \mathrm{d}u, \tag{12}$$

with $S = 3$ generation seeds, and $S = 1$ for TR and GS, whose pools do not depend on the generation seed. Lower is better.

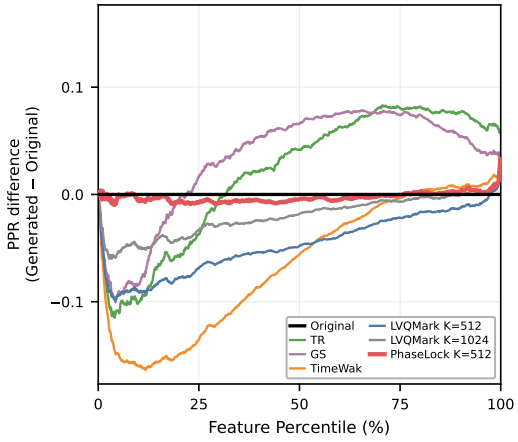
**Percentile comparison.** Figure 8 shows the comparison of Figure 4 on all ten datasets. Each curve is $Q_{G,s}(u) - Q_R(u)$ averaged over seeds, so the distance of Table 4 is the mean absolute value of the curve. ETTh is the one dataset on which PHASELOCK is not the lowest: LVQMark with $K = 256$ attains 0.0102 against 0.0103.



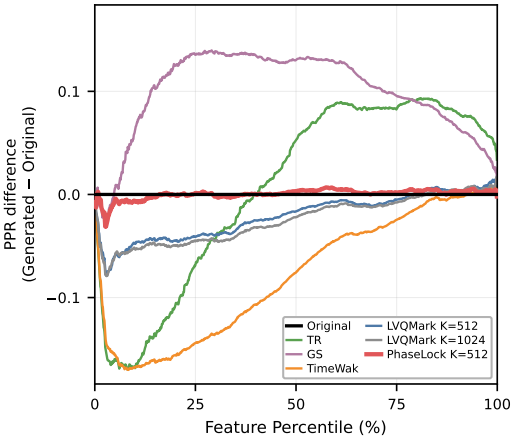

(a) ETTh



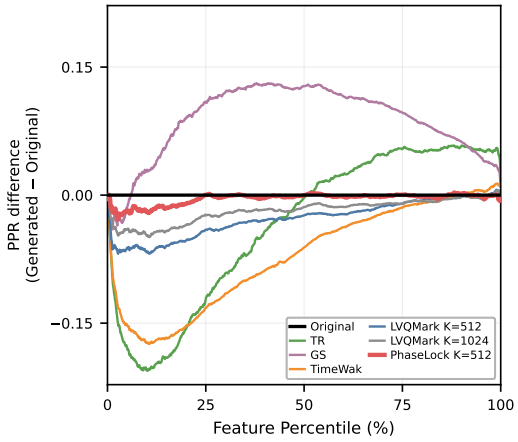

(b) ETTm



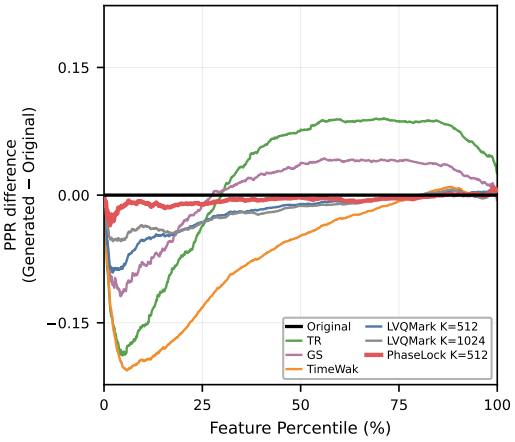

(c) AirQ-1




(d) AirQ-2

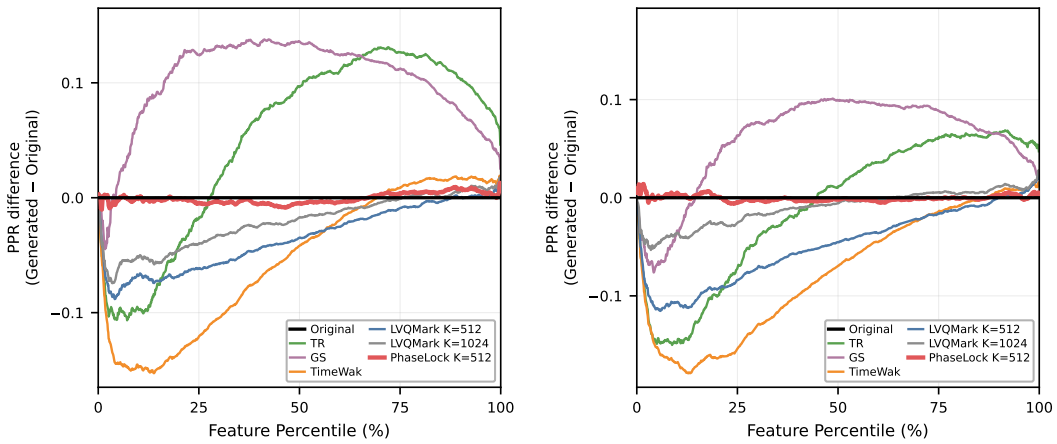



(e) AirQ-3

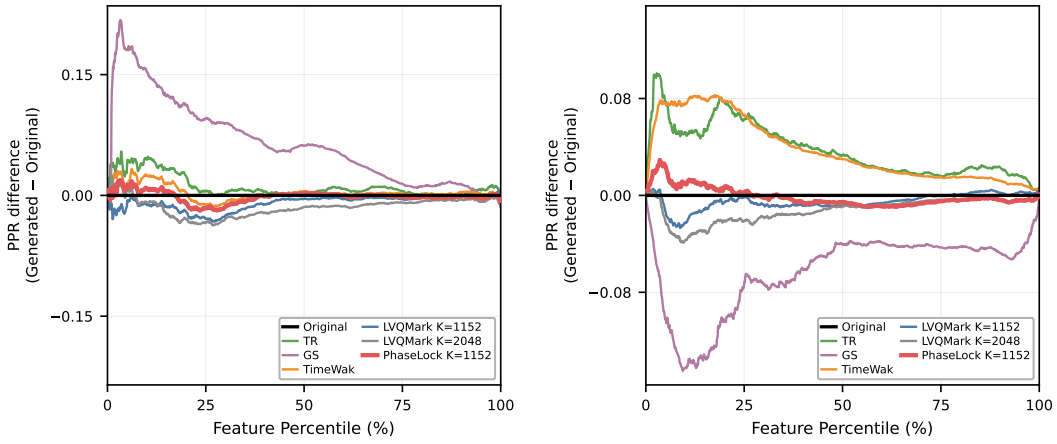



(f) AirQ-4

Figure 8: Generated minus real PPR at matching percentiles on the most periodic variable of each dataset, averaged over generation seeds. Closer to zero is better.




(g) AirQ-5 (h) AirQ-6




(i) Energy (j) Weather

Figure 8: (Continued.) Generated minus real PPR at matching percentiles on the remaining datasets.