MARC: MULTI-BIT WATERMARKING FOR AUTOREGRESSIVE AUDIO GENERATION AGAINST CODEC ATTACKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Generated audio is now used in a range of applications, creating a need to verify its origin after distribution and signal processing. This task is particularly challenging for autoregressive audio generation because codec processing can alter the token sequence recovered from the waveform. Such changes reduce the reliability of watermark detection and payload decoding. Existing methods construct token groups using either intrinsic token representations or substitution patterns caused by transformations. As a result, intrinsic token relationships and codec induced substitutions are modeled separately. In addition, most methods support only zero bit detection. They can determine whether a watermark is present but cannot distinguish individual generated outputs. We propose **MARC**, a multi-bit generative watermarking method for autoregressive audio generation. MARC integrates intrinsic token representations with confusion patterns obtained through retokenization and multiple codecs, forming a codec-aware token-cluster space. Within this space, payload-driven cluster scheduling is used to embed a multi-bit watermark, while detection and payload decoding are performed on retokenized observations. Experiments on speech, dialogue, and music generation show that MARC achieves an average of 97.3% bit extraction accuracy on unmodified watermarked audio and the watermark can still be extracted under diverse codec attacks. MARC also demonstrates robustness to overwriting attacks.

1 INTRODUCTION

Autoregressive models are widely used for audio generation (Rouard et al., 2026), including speech synthesis (Nishimura et al., 2025), spoken dialogue (Tan et al., 2026), and music generation (Liu et al., 2025), where discrete audio representations allow different generation tasks to be formulated as token-sequence modeling. As these models are increasingly used in practical applications, generated audio needs not only reliable source verification but also the ability to distinguish and trace individual outputs (San Roman et al., 2024). Multi-bit generative watermarking provides this capability by embedding additional information into each generated audio sample beyond zero-bit detection. This information should remain recoverable after the audio is released. In reality, generated audio may undergo compression or other codec processing during transmission, storage, and redistribution (Liu et al., 2024a). Verification for autoregressive token-level watermarking requires retokenizing the generated waveform, and the resulting token sequence may differ from the one used during generation. Codec processing can further increase these token substitutions, making both watermark detection and multi-bit recovery more difficult.

Existing work has developed several forms of generative watermarking for autoregressive and token-based audio generation as shown in Figure 1. Wu et al. (2025a) introduce statistical token-level watermarking and identifies retokenization mismatch. Aligned-IS (Wu et al., 2025b) and Hidden in Plain Tokens (HiPT) (Milis et al., 2026) further improve robustness through token grouping, but two limitations remain. First, **single-source token grouping**: existing clustering strategies rely on a single source of token relationships: Aligned-IS groups tokens using intrinsic representations, whereas HiPT derives groups from transformation-induced confusion, so these two types of information are not modeled jointly. Second, **zero-bit limitation**: existing autoregressive watermarking primarily targets zero-bit detection, which can detect watermark but cannot carry additional information

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.

for distinguishing individual generated outputs. Extending such methods to multi-bit watermarking is difficult because retokenization and codec processing can directly corrupt the embedded bits. To solve these problems, we propose **MARC**, a generative watermarking framework that enables robust multi-bit embedding for autoregressive audio generation. To address **single-source token grouping**, MARC combines intrinsic token representations with confusion patterns induced by retokenization and diverse codecs to build a codec-aware token-cluster space. To address **zero-bit limitation**, MARC uses payload-driven cluster scheduling to embed a multi-bit watermark and performs detection and decoding on retokenized observations to recover both watermark presence and the embedded bits. MARC additionally introduces a public check value (pub-key) and a secret auxiliary suffix (pri-key) for message consistency checking, with both parameters used in the full verification and recovery protocol without assuming cryptographic security. Across multiple autoregressive audio generation tasks, MARC achieves an average of 97.3% bit extraction accuracy on unmodified watermarked audio and the watermark can still be extracted under diverse codec attacks. Our contributions are summarized as follows:

- We propose **MARC**, a codec-robust multi-bit generative watermarking framework for autoregressive audio generation that extends zero-bit detection to multi-bit payload embedding and recovery.
- We propose a **codec-aware token-cluster representation** that combines intrinsic token structure with codec-aware confusion, reducing the limitations of single-source token grouping under different codec transformations.
- We develop **payload-driven embedding and robust recovery** for multi-bit watermarking under token-level errors. Experiments on speech, dialogue, and music generation show that MARC works across different autoregressive audio generation tasks and remains robust under codec attacks.
- We demonstrate **robustness to watermark overwriting**, showing that MARC preserves detection and recovery of the original payload when an additional watermark is embedded into the released audio.

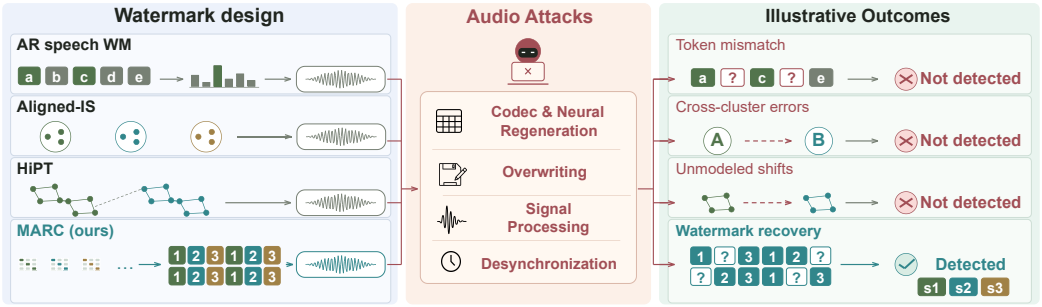

Figure 1: Method overview.

2 Related Work

Generative Audio Watermarking. Generative audio watermarking embeds watermark information during audio synthesis rather than adding it through post-hoc waveform modification. Existing methods integrate watermarks into diffusion generation (Cao et al., 2023; Liu et al., 2024b), latent representations (San Roman et al., 2025), diffusion noise distributions (Tang, 2025), model parameters (Feng et al., 2025), or diffusion-domain modulation (Lee et al., 2025). These studies show that watermarking can be incorporated into different audio generators, but their mechanisms mainly depend on continuous latent, diffusion, or model-specific representations and are not directly applicable to discrete token sampling in autoregressive generation. Autoregressive audio watermarking instead operates on token distributions and must remain detectable after the generated waveform is retokenized.

Autoregressive Audio Watermarking. TraceableSpeech (Zhou et al., 2024) generates watermarked speech within TTS, while Wu et al. (2025a) introduces statistical token-level watermarking for autoregressive speech and identifies retokenization mismatch as a central issue. Aligned-IS (Wu et al., 2025b) reduces this mismatch by clustering tokens according to intrinsic representation geometry. However, its clustering does not explicitly consider codec attacks. HiPT (Milis et al., 2026) instead builds token groups from transformation-aware confusion, directly modeling token substitutions observed after processing, but does not jointly use intrinsic token structure. The two methods therefore rely on different sources of token relationships, but they do not model these sources jointly. In addition, existing autoregressive methods mainly focus on zero-bit detection rather than embedding multi-bit information. Joint modeling of token structure and codec-aware confusion with robust multi-bit recovery still remains insufficiently addressed. Our method combines these two sources of token information and extends autoregressive generative watermarking to multi-bit embedding and recovery.

3 THREAT MODEL

Adversary Goal. We consider a waveform y_w generated by an autoregressive audio model and watermarked with a multi-bit payload $\mathbf{m}$. The full verification protocol additionally uses a public check value (pub-key) and a secret auxiliary suffix (pri-key) for message consistency checking, and neither parameter is assumed to provide cryptographic meaning. After release, an output-level transformation $\mathcal{A}$ produces

$$\tilde{y} = \mathcal{A}(y_w),$$

which is retokenized for watermark verification and payload recovery. The verifier uses $\mathbf{m}^\star$ to search over candidate alignments and then estimates the payload from the selected observations. The adversary aims to cause missed detection or incorrect payload recovery while preserving the underlying audio content and perceptual usability. The same setting covers benign post-processing.

Adversary Capability. The pri-key is not directly disclosed, only the owner and the authorized user can have it. The adversary does not control the verifier's payload hypothesis $\mathbf{m}^\star$. We consider neural audio codec attacks, signal processing, and desynchronization applied to the released waveform. These transformations can alter both token identities and their temporal correspondence with watermark embedding steps, even when perceptual content is preserved. MARC focuses on multi-bit verification and recovery under this joint token mismatch and temporal misalignment, with codec-induced retokenization mismatch as the primary challenge. Test-time codecs may include those unseen during construction of the codec-aware cluster map.

4 MARC: CODEC-ROBUST MULTI-BIT WATERMARKING

Analysis. Codec re-encoding changes discrete token IDs and can disrupt temporal alignment. Recent methods encode watermarks through frame-level cluster membership and detect them using $S = \sum_t \mathbf{1}[c(y_t) = s_t]$. Codec-induced mismatches can drive S toward chance level. Zero-bit methods can pool s_t into a global set for position-independent counting, whereas multi-bit recovery requires alignment with embedding positions: an alignment error can propagate to subsequent bits, while finer clustering increases vulnerability to codec-induced boundary crossings.

MARC merges tokens that would swap places under the codec into the same cluster $c_\cup$, allowing the symbols themselves to survive compression, and then uses local shifts to realign misaligned frames $\hat{\delta} = \arg\max_\delta \sum_t \mathbf{1}[c_\cup(y_{t+\delta}) = s_t(m)]$. MARC combines complementary token relationships to support multi-bit watermarking under retokenization and codec processing. Offline clustering fuses embedding geometry with empirical codec confusion to construct a codec-aware cluster map. During generation, a repeated symbol schedule selects keyed target clusters for probability reweighting. Users can detect whether the audio contains a watermark. The legal verifier uses both pub-key and pri-key to recover watermark. Our method pipeline is shown in Figure 2.

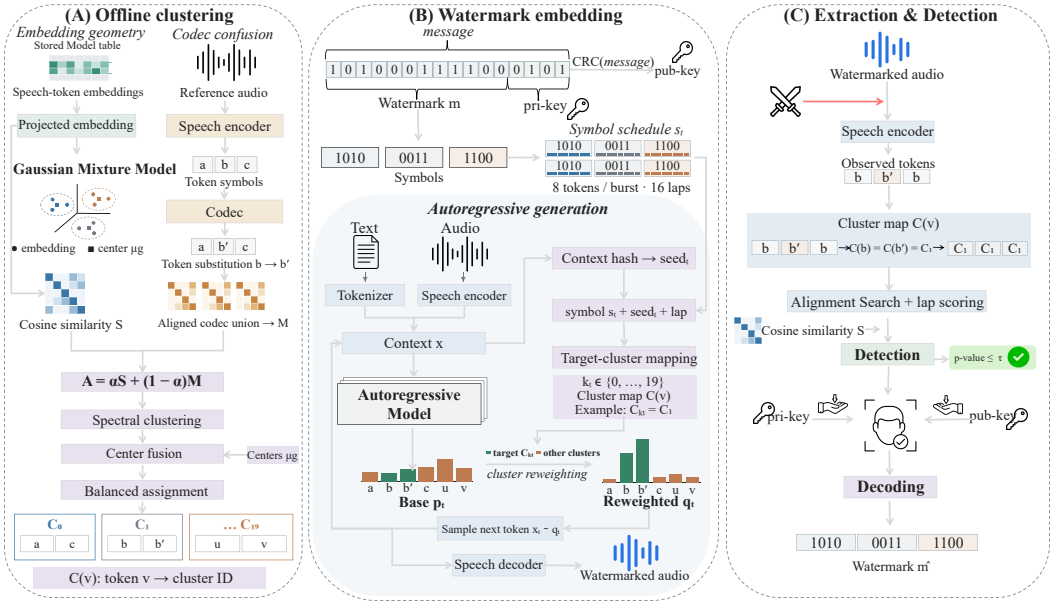

Figure 2: Method pipeline: (A) *Offline clustering*: joint embedding and code similarities yield the token-to-cluster map $C(v)$. (B) *Watermark embedding*: key-conditioned symbol scheduling and cluster reweighting guide autoregressive generation of watermarked audio. (C) *Extraction and detection*: observed tokens are cluster-mapped and alignment-scored for watermark detection and message decoding.

4.1 Codec-Aware Clustering

Embedding Geometry. We project the generator's audio-token embeddings into a reduced representation space and fit a Gaussian mixture model to obtain geometric centers $\{\mu_k^{\mathrm{G}}\}_{k=0}^{K-1}$. For projected embeddings e_a and e_b, we define $S_{ab} = (1 + \cos(e_a, e_b))/2$. This representation provides token relationships independently of which transformations appear in the calibration data.

Codec Confusion. Let E and D denote the speech encoder and decoder. For each reference waveform $x \in \mathcal{D}$, we obtain $u = E(x)$, the reconstruction sequence $v^{(0)} = E(D(u))$, and codec-processed sequences $v^{(j)} = E(\mathcal{A}_j(x))$. We accumulate confusion counts (Mangu et al., 2000) by pairing tokens at matching indices over the common sequence length:

$$M_{ab}^{(j)} = \sum_{x \in \mathcal{D}} \sum_{i=1}^{\min(|u|,|v^{(j)}|)} \mathbf{1}[u_i = a,\ v_i^{(j)} = b]. \tag{1}$$

These counts capture empirical changes across the reconstruction and codec channels, including residual temporal mismatch.

We symmetrize and normalize each matrix using $\mathcal{N}(M) = (M + M^\top)/\max_{a,b}(M + M^\top)_{ab}$, leaving zero matrices unchanged. The codec union is

$$M_{\mathrm{U}} = \beta \mathcal{N}(M^{(0)}) + (1 - \beta) \max_{0 \le j \le J} \mathcal{N}(M^{(j)}). \tag{2}$$

The element-wise maximum retains strong channel-specific confusions, while β adds weight to ordinary retokenization.

Fusion and Cluster Assignment. We combine embedding similarity and normalized codec confusion, then apply spectral clustering to the resulting affinity:

$$A = \alpha S + (1 - \alpha)\mathcal{N}(M_{\mathrm{U}}), \qquad \mu_k = \alpha \mu_k^{\mathrm{G}} + (1 - \alpha)\mu_k^{\mathrm{S}}, \tag{3}$$

where μ_k^{S} is the centroid of spectral cluster k. The implementation blends the two center sets in their returned index order, maps the centers back to the embedding space, and applies balanced assignment. This yields a codec-aware cluster map $C : \mathcal{V}_a \to \{0, \ldots, K-1\}$, hereafter the cluster map. We use $K = 20$. Balanced assignment controls cluster cardinalities rather than their probability mass under the generator.

4.2 Watermark Embedding

Message and Symbol Schedule. The embedded payload $\mathbf{m} \in \{0,1\}^{12}$ is partitioned into three symbols $s_0, s_1, s_2 \in \{0, \ldots, 15\}$. The main configuration assigns 8 consecutive watermark steps to each symbol, cycles through the three positions, and repeats the schedule across 16 laps. The public check value is computed as $k_{\mathrm{pub}} = \mathrm{CRC}(m, k_{\mathrm{pri}})$, where k_{pri} denotes the secret auxiliary suffix shared by the owner and the authorized verifier.

Target-cluster Mapping. Text or audio prompts form the context of the autoregressive generator. At watermark step t, the secret key κ and preceding token produce a context seed r_t. Let j_t and ℓ_t denote the scheduled symbol position and lap index. The target cluster is

$$k_t = \left[b(s_{j_t}) + H(s_{j_t}, r_t, \ell_t, j_t)\right] \bmod K, \tag{4}$$

where b is a fixed symbol-to-cluster base mapping and H is a domain-separated hash. The mapping omits the exact within-burst step index. Changes in context or lap can nevertheless assign repeated symbols to different target clusters.

Distribution Reweighting. Let $p_t(v)$ be the generator's next-token distribution. We extend the audio cluster map to the sampling vocabulary by assigning non-audio tokens to the remaining groups. For $V_k = \{v : C(v) = k\}$, define $P_{t,k} = \sum_{v \in V_k} p_t(v)$. The reweighting rule (Wu et al., 2024) gives

$$q_t(v) = p_t(v) \frac{\widehat{P}_{t,C(v)}}{P_{t,C(v)}}, \qquad \widehat{P}_{t,k_t} = \min(KP_{t,k_t}, 1). \tag{5}$$

When the target mass is below one, the remaining mass is distributed among non-target groups in proportion to $(KP_{t,k} - 1)_+$. Zero-probability groups contribute no mass. This changes cluster probabilities while preserving relative probabilities within each group. The generator samples from q_t, and the speech decoder converts the generated sequence into watermarked audio.

4.3 Extraction and Detection

Alignment Search and Lap Scoring. The verifier receives the processed waveform $\widetilde{y}$, the secret key κ, and the claimed payload $\mathbf{m}^\star$. Speech encoding produces observed tokens $\widetilde{z}_t$, which are mapped to clusters $\widetilde{c}_t = C(\widetilde{z}_t)$. The verifier also recomputes context seeds $\widetilde{r}_t$. Tokens mapped to the same cluster preserve the current cluster observation even when their identities differ.

The verifier evaluates waveform variants, observation windows, and offsets of the watermark-step indices. Waveform variants are retokenized separately. Index offsets change the symbol and lap assignments while retaining the observed clusters and their associated seeds. The evaluation can use the supplied attack type and configured processing parameters to construct waveform variants.

For a window and offset δ, the claimed payload determines the expected cluster sequence. We compute

$$\begin{aligned} h_\delta &= \sum_t \mathbf{1}[\widetilde{c}_t = k_{t+\delta}(\mathbf{m}^\star, \widetilde{r}_t)], \\ p_\delta &= \Pr[\mathrm{Bin}(N_\delta, 1/K) \geq h_\delta], \end{aligned} \tag{6}$$

where N_δ is the number of observations. Detection retains the strongest candidate across windows and waveform variants, while decoding evaluates the candidate offsets individually.

Detection. The detector compares the selected score with a threshold τ. It can augment hard matches with capped soft evidence from an auxiliary cluster affinity matrix $R \in \mathbb{R}^{K \times K}$ and optionally combine the resulting scores. Here R scores relations between observed and expected clusters, whereas A is the token affinity used for offline clustering.

Decoding. For each candidate alignment a, the decoder enumerates all 16 values at every symbol position. Let $\mathcal{O}_a$ denote its observations and let η_a be the match rate against $\mathbf{m}^\star$, clipped to $[0.05, 0.95]$. The hard symbol score is

$$L_{q,a}^{\mathrm{hard}}(s) = \sum_{\substack{t \in \mathcal{O}_a \\ j_t = q}} \log\left[\eta_a \mathbf{1}[\widetilde{c}_t = k_t(s, \widetilde{r}_t)] + \frac{1-\eta_a}{K}\right]. \tag{7}$$

The optional soft rule gives partial compatibility to unequal clusters when $R_{ij}/\max(R_{ii}, R_{jj}, \epsilon)$ exceeds its configured threshold. For each rule, the decoder maximizes the symbol score independently at each position. It retains the rule with the larger summed maximum score and converts the resulting symbols into a candidate bit string $\widehat{\mathbf{m}}^{(a)}$.

The pri-key scores candidate payloads before aggregation. The resulting n selected candidates are combined by bit-wise majority voting. The voted payload initializes a local refinement guided by pooled symbol scores and the pub-key:

$$\begin{aligned} m_b^{(0)} &= \mathbf{1}\left[\sum_{a=1}^{n} \widehat{m}_b^{(a)} > n/2\right], \\ J(\mathbf{s}) &= \sum_q \overline{L}_q(s_q) - \gamma\left[1 - V_{\mathrm{pub}}(\mathcal{B}(\mathbf{s}), k_{\mathrm{pub}})\right]. \end{aligned} \tag{8}$$

Here, $\mathcal{B}$ maps symbols to a payload, and V_{pub} returns one if the payload passes pub-key verification and zero otherwise. The pooled score $\overline{L}_q$ aggregates symbol evidence across alignments, and $\gamma > 0$ penalizes failed verification. Voting ties resolve to zero. Local search evaluates high-scoring symbol replacements and accepts the refined payload $\widehat{\mathbf{m}}$ only if it passes pub-key verification.

5 Evaluation

5.1 Experimental Setup

For our main audio generation experiments, we consider two audio generative models with autoregressive architecture, that span different applications and use different audio codecs for discretization. For dialogue and speech synthesis tasks, We evaluate MARC on SpiritLM (Nguyen et al., 2025) using Finance QA (Tu et al., 2024), Dolly CW (Conover et al., 2023), and LibriSpeech (Panayotov et al., 2015). For music generation tasks we evaluate on MusicGen (Copet et al., 2023) using LibriSpeech, MusicCaps (Agostinelli, 2023), and MTG-Jamendo (Bogdanov et al., 2019).

We compare our method with KGW using $\delta = 2.0$ following Wu et al. (2025a) which we name as AR speech WM, Aligned-IS (Wu et al., 2025b), and HiPT (Milis et al., 2026). As for our method MARC, embeds a 12 bit payload by default. The main configuration assigns 8 consecutive watermark steps to each symbol, cycles through the three positions, and repeats the schedule across 16 laps.

We evaluate multi-bit watermark detection using bit accuracy as the detection score, True Positive Rate (TPR) is measured at a target False Positive Rate (FPR) of 1%. Cluster Median-p (cluster_p) summarizes the cluster-based detection statistic. Bit accuracy (bit_acc) is the fraction of correctly recovered payload bits. For fidelity, we report predicted MOS with NISQA (Mittag et al., 2021) and DNSMOS (Cumlin et al., 2024) scores for speech outputs. We compute FAD using VGGish features (Kilgour et al., 2019). NISQA and DNSMOS are not used to assess generated music. These two scores correspond to the naturalness of speech and noise levels, which do not measure musical quality, and therefore MusicGen's results do not use these two metrics.

5.2 Reliability and Fidelity

Table 1 reports detection and payload recovery for each model and dataset pair. The results for SpiritLM are presented in Table 1(a), covering Finance QA, Dolly CW, and LibriSpeech. Across all three datasets, the data shows that our method, MARC, not only outperforms existing approaches in terms of clustering p-values, but also makes the watermark easier to detect, whilst achieving high

Table 1: Watermark reliability and audio fidelity on SpiritLM and MusicGen: All methods except MARC are zero-bit watermarking methods, so bit-level TPR and bit accuracy are not applicable to these baselines. NISQA and DNSMOS assess speech quality and are not applicable to music generation.

Dataset / Task	Method	Reliability			Fidelity		
		TPR ↑	-log($cluster_p$) ↑	bit_acc ↑	FAD ↓	NISQA ↑	DNSMOS ↑
			(a) SpiritLM				
Finance QA (Dialogue)	AR speech WM	–	3.54	–	0.075	4.505	3.854
	Aligned-IS	–	4.47	–	**0.056**	**4.519**	3.845
	HiPT	–	21.44	–	1.031	4.429	3.772
	MARC (Ours)	0.94	**37.08**	0.98	0.262	4.503	**3.868**
Dolly CW (Dialogue)	AR speech WM	–	3.31	–	0.077	4.491	3.831
	Aligned-IS	–	5.33	–	0.055	4.514	3.791
	HiPT	–	17.81	–	0.573	**4.516**	3.794
	MARC (Ours)	0.97	**34.16**	0.99	**0.024**	4.491	**3.867**
LibriSpeech (Speech generation)	AR speech WM	–	3.22	–	0.063	**4.501**	3.840
	Aligned-IS	–	5.11	–	**0.048**	4.486	**3.875**
	HiPT	–	9.61	–	0.625	4.401	3.842
	MARC (Ours)	0.96	**30.01**	0.98	0.123	4.446	3.867
			(b) MusicGen				
LibriSpeech (Music generation)	AR speech WM	–	2.34	–	1.062	–	–
	Aligned-IS	–	3.58	–	**0.495**	–	–
	HiPT	–	28.55	–	1.171	–	–
	MARC (Ours)	0.96	**34.04**	0.97	1.176	–	–
MusicCaps (Music generation)	AR speech WM	–	1.86	–	0.924	–	–
	Aligned-IS	–	3.47	–	**0.643**	–	–
	HiPT	–	25.14	–	1.056	–	–
	MARC (Ours)	0.95	**26.15**	0.97	1.182	–	–
MTG-Jamendo (Music generation)	AR speech WM	–	0.47	–	1.018	–	–
	Aligned-IS	–	3.47	–	**0.596**	–	–
	HiPT	–	17.04	–	1.131	–	–
	MARC (Ours)	0.90	**22.90**	0.95	1.616	–	–

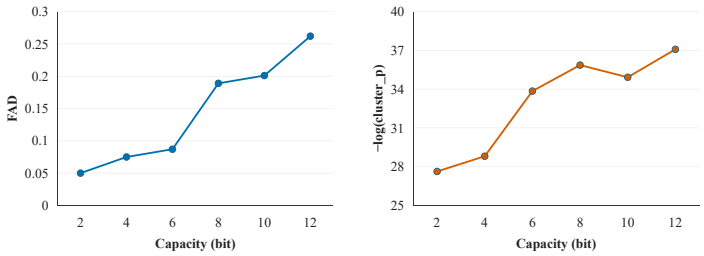

Figure 3: Fidelity and reliability across different capacities with SpiritLM on FinanceQA.

detection and extraction rates for multi-bit watermarks. As shown by the fidelity data in the table, although our approach involves some trade-offs in terms of fidelity in order to achieve multi-bit representation, it still maintains good audio quality and offers advantages over existing approaches in dialogue tasks, speech synthesis and music generation.

5.3 Multi-bit Capacity

Given a fixed audio duration, a watermark token budget and an embedding strength, the payload is divided into several symbol blocks, we use 4-bit symbols when the payload length is divisible

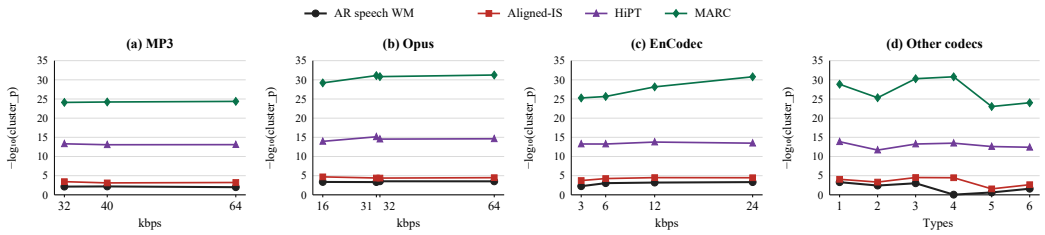

Figure 4: Codec attacks under different parameter: Other codecs: 1 = DAC; 2 = SpeechTokenizer; 3 = FaCodec; 4 = VoiceCraft-EnCodec; 5 = SoCodec; 6 = SemantiCodec.

Table 2: Overwriting Attacks: The 'watermark' indicator refers to the owner of the original watermark. 'Verified key' refers to the type of pri-key held by the person performing the decoding.

Attack condition	watermark	Verified key	TPR ↑	$-\log(\text{cluster_p})$ ↑	bit_acc ↑
No attack	Owner	Owner	0.94	37.08	0.98
After Overwriting	Owner	Owner	0.90	24.18	0.88
After Overwriting	Owner	Attacker	0.68	14.18	0.75
After Overwriting	Attacker	Attacker	0.93	35.92	0.98

by four and 2-bit symbols otherwise. As the number of blocks increases with capacity, the number of observations allocated to each symbol within the same generation decreases.As shown in Figure 3, FAD increases slightly, as the higher capacity results in denser symbol placement and tighter constraints on clusters imposed by 4-bit symbols. At the same time, as capacity increases, the detection score increases with a decrease at 10 bits. We have opted for a 12-bit embedding here because, following robustness experiments, we found that 12 bits is sufficient to carry the watermark information without spreading the observations per symbol too thinly, thereby maintaining good robustness.

5.4 Robustness

We evaluate MARC against two types of attacks: audio transformation attacks, which perturb the watermarked audio, and overwriting attacks, which introduce a competing watermark to challenge the original attribution. Experiments are conducted on SpiritLM and MusicGen. We consider 21 attacks spanning 9 codec and neural regeneration attacks, 8 signal processing attacks, and 4 desynchronization attacks. These transformations challenge token consistency, acoustic evidence preservation, and temporal alignment, respectively. As shown in Table 3, MARC outperforms all baselines on this metric under all nine codec and neural regeneration attacks on both models.

This performance is consistent with the complementary roles of payload-driven scheduling and alignment-aware decoding, repeated observations preserve recovery evidence, while alignment correction and cross-lap aggregation accommodate temporal disruption. Additional codec parameter evaluations and experimental details are provided in Figure 4 and Appendices A.2.

For the overwriting attack, an overwriting attacker embeds a competing message to replace the original attribution. Table 2 shows that the legitimate owner can still recover the original payload using their private key, and the attacker's message is also recoverable with the attacker's key. This preservation of the original payload demonstrates MARC's resistance to overwriting under the evaluated conditions and supports key-conditioned payload recovery.

5.5 Ablation Study

Detailed results in Appendix A.1. Codec-union clustering consistently improves post-codec recovery over geometric and retokenization-based alternatives, showing the importance of modeling codec-dependent token substitutions. Multi-lap organization, burst scheduling, and structured decoding each contribute to robustness, while removing the lap pipeline and HiPT search causes the

Table 3: Robustness under different attacks.

(a) SpiritLM with FinanceQA

Attacks	AR speech WM	Aligned-IS	HiPT	MARC(ours)		
	$-\log(\text{cluster_p})\uparrow$	$-\log(\text{cluster_p})\uparrow$	$-\log(\text{cluster_p})\uparrow$	TPR ↑	$-\log(\text{cluster_p})\uparrow$	bit_acc ↑
Codec & Regeneration						
MP3 (64kbps)	2.00	3.22	13.09	0.84	24.37	0.97
Opus (16kbps)	3.36	4.39	13.95	0.96	29.20	0.99
EnCodec (24kbps)	3.32	4.36	13.48	0.97	30.79	0.99
DAC	3.30	4.05	13.86	0.94	28.84	0.98
SpeechTokenizer	2.42	3.33	11.67	0.90	25.33	0.98
FaCodec	3.02	4.41	13.26	0.92	30.29	0.98
VoiceCraft-EnCodec	1.02	4.46	13.47	0.97	30.70	0.95
SoCodec	0.62	1.56	12.60	0.78	23.03	0.96
SemantiCodec	1.60	2.64	12.41	0.85	24.02	0.97
Signal Processing						
high-pass	3.01	3.87	15.03	0.85	23.59	0.95
low-pass	3.06	3.88	14.60	0.90	28.38	0.98
smooth	3.42	4.54	14.47	0.94	30.80	0.99
Gaussian_noise	3.33	4.10	14.52	0.93	26.08	0.98
pink_noise	3.51	4.41	14.38	0.94	30.76	0.99
echo	3.54	4.38	14.63	0.94	28.01	0.99
quantization	2.45	3.52	12.48	0.91	22.86	0.98
updownresample	3.49	4.44	14.82	0.96	31.35	0.99
Desynchronization						
speed	0.52	1.42	6.45	0.83	18.53	0.93
crop	1.65	3.54	6.80	0.96	18.69	0.99
shift (1/8)	3.09	4.14	14.83	0.91	28.31	0.98
shush	3.46	4.37	13.82	0.95	27.14	0.99

(b) MusicGen with MusicCaps

Attack	AR speech WM	Aligned-IS	HiPT	MARC(ours)		
	$-\log(\text{cluster_p})\uparrow$	$-\log(\text{cluster_p})\uparrow$	$-\log(\text{cluster_p})\uparrow$	TPR ↑	$-\log(\text{cluster_p})\uparrow$	bit_acc ↑
Codec & Regeneration						
MP3 (64kbps)	0.01	0.22	10.18	0.80	18.35	0.88
Opus (16kbps)	0.83	2.46	22.02	0.75	25.68	0.85
EnCodec (24kbps)	1.21	2.81	23.24	0.73	26.03	0.86
DAC	0.78	2.57	22.83	0.70	24.62	0.80
SpeechTokenizer	0.55	2.34	22.83	0.66	23.52	0.79
FaCodec	0.01	0.20	4.00	0.71	8.50	0.81
VoiceCraft-EnCodec	0.05	0.59	9.09	0.69	13.31	0.78
SoCodec	0.00	0.40	3.18	0.68	6.67	0.77
SemantiCodec	0.00	0.20	4.58	0.69	8.38	0.77
Signal Processing						
high-pass	0.02	1.42	13.06	0.90	12.32	0.93
low-pass	0.59	1.19	22.62	0.88	25.21	0.94
smooth	0.08	0.54	15.74	0.89	16.98	0.92
Gaussian_noise	1.43	3.24	15.26	0.91	16.33	0.91
pink_noise	1.06	3.47	25.02	0.90	23.69	0.92
echo	1.30	3.15	25.10	0.85	25.10	0.93
quantization	1.38	3.41	11.06	0.83	13.57	0.92
updownresample	1.86	3.27	25.08	0.90	25.65	0.91
Desynchronization						
speed	0.00	0.08	5.51	0.75	9.88	0.86
crop	1.25	2.99	0.00	0.80	11.50	0.87
shift (1/8)	0.01	0.32	8.49	0.89	13.65	0.90
shush	1.79	3.47	25.12	0.90	26.01	0.92

largest performance drop. The complete configuration provides the best overall trade-off between clean extraction and codec robustness.

6 Conclusion

We presented MARC, a multi-bit watermarking framework for autoregressive audio generation that integrates codec-aware token clustering, payload-driven scheduling, and alignment-aware verification. Experiments across speech, dialogue, and music generation demonstrate reliable payload recovery under the evaluated codec and overwriting attacks while maintaining audio fidelity, extending token-level watermarking beyond presence detection.

AI USE STATEMENT

In this study, we used generative AI tools to refine the text and images in the article. We did not use generative AI tools to generate images directly. Furthermore, we used generative AI tools for grammatical correction and to check the article for simple writing errors. We have reviewed all AI-generated outputs. We used AI to perform code syntax checks. We take responsibility for the final content of this study, including any text, arguments or findings generated using generative AI.

REFERENCES

Andrea Agostinelli. Musiclm: Generating music from text. *arXiv preprint arXiv:2301.11325*, 2023. URL https://google-research.github.io/seanet/musiclm/examples/.

Dmitry Bogdanov, Minz Won, Philip Tovstogan, Alastair Porter, and Xavier Serra. The mtg-jamendo dataset for automatic music tagging. In *Machine Learning for Music Discovery Workshop, International Conference on Machine Learning (ICML 2019)*, Long Beach, CA, United States, 2019. URL http://hdl.handle.net/10230/42015.

Xirong Cao, Xiang Li, Divyesh Jadav, Yanzhao Wu, Zhehui Chen, Chen Zeng, and Wenqi Wei. Invisible watermarking for audio generation diffusion models. In *2023 5th IEEE international conference on trust, privacy and security in intelligent systems and applications (TPS-ISA)*, pp. 193–202. IEEE, 2023.

Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. Free dolly: Introducing the world's first truly open instructiontuned llm. 2023. URL https://www.databricks.com/blog/2023/04/12/dolly-first-open-commercially-viable-instruction-tuned-llm.

Jade Copet, Felix Kreuk, Itai Gat, Tal Remez, David Kant, Gabriel Synnaeve, Yossi Adi, and Alexandre Défossez. Simple and controllable music generation. *Advances in neural information processing systems*, 36:47704–47720, 2023.

Fredrik Cumlin, Xinyu Liang, Victor Ungureanu, Chandan KA Reddy, Christian Schüldt, and Saikat Chatterjee. Dnsmos pro: A reduced-size dnn for probabilistic mos of speech. In *Interspeech*, 2024.

Yan Feng, Xuebin Zhang, Fuyuan Feng, Guanglin Zhang, and Longting Xu. Robust and imperceptible watermarking framework for generative audio models. *IEEE Signal Processing Letters*, 2025.

Kevin Kilgour, Mauricio Zuluaga, Dominik Roblek, and Matthew Sharifi. Fréchet audio distance: A reference-free metric for evaluating music enhancement algorithms. In *Interspeech*, 2019. URL https://api.semanticscholar.org/CorpusID:202725406.

Kyungryeol Lee, Seongmin Hong, and Se Young Chun. Robust watermarks for audio diffusion models by quadrature amplitude modulation. *Pattern Recognition Letters*, 2025.

Hongbin Liu, Moyang Guo, Zhengyuan Jiang, Lun Wang, and Neil Zhenqiang Gong. Audiomarkbench: Benchmarking robustness of audio watermarking. In *Advances in Neural Information Processing Systems*, volume 37, pp. 52241–52265. Curran Associates, Inc., 2024a. doi: 10.52202/079017-1656.

Weizhi Liu, Yue Li, Dongdong Lin, Hui Tian, and Haizhou Li. Groot: Generating robust watermark for diffusion-model-based audio synthesis. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pp. 3294–3302, 2024b.

Zihan Liu, Shuangrui Ding, Zhixiong Zhang, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Yuhang Cao, Dahua Lin, and Jiaqi Wang. SongGen: A single stage auto-regressive transformer for text-to-song generation. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 38351–38364. PMLR, 13–19 Jul 2025.

Lidia Mangu, Eric Brill, and Andreas Stolcke. Finding consensus in speech recognition: word error minimization and other applications of confusion networks. *Computer Speech & Language*, 14 (4):373–400, 2000.

Georgios Milis, Yubin Qin, Yihan Wu, and Heng Huang. Hidden in plain tokens: Simply robust, gradient-free watermark for synthetic audio. In *Forty-third International Conference on Machine Learning*, 2026.

Gabriel Mittag, Babak Naderi, Assmaa Chehadi, and Sebastian Möller. NISQA: A Deep CNN-Self-Attention Model for Multidimensional Speech Quality Prediction with Crowdsourced Datasets. In *Interspeech 2021*, pp. 2127–2131, 2021.

Tu Anh Nguyen, Benjamin Muller, Bokai Yu, Marta R. Costa-jussa, Maha Elbayad, Sravya Popuri, Christophe Ropers, Paul-Ambroise Duquenne, Robin Algayres, Ruslan Mavlyutov, Itai Gat, Mary Williamson, Gabriel Synnaeve, Juan Pino, Benoît Sagot, and Emmanuel Dupoux. SpiRit-LM: Interleaved spoken and written language model. *Transactions of the Association for Computational Linguistics*, 13:30–52, 2025.

Yuto Nishimura, Takumi Hirose, Masanari Ohi, Hideki Nakayama, and Nakamasa Inoue. Hall-e: Hierarchical neural codec language model for minute-long zero-shot text-to-speech synthesis. In *International Conference on Learning Representations*, volume 2025, pp. 91141–91164, 2025.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. Librispeech: an asr corpus based on public domain audio books. In *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pp. 5206–5210. IEEE, 2015.

Simon Rouard, Manu Orsini, Axel Roebel, Neil Zeghidour, and Alexandre Défossez. Continuous audio language models. In *International Conference on Learning Representations*, volume 2026, pp. 136805–136827, 2026.

Robin San Roman, Pierre Fernandez, Hady Elsahar, Alexandre Défossez, Teddy Furon, and Tuan Tran. Proactive detection of voice cloning with localized watermarking. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 43180–43196. PMLR, 21–27 Jul 2024.

Robin San Roman, Pierre Fernandez, Antoine Deleforge, Yossi Adi, and Romain Serizel. Latent watermarking of audio generative models. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2025.

Chao-Hong Tan, Qian Chen, Wen Wang, Chong Deng, Qinglin Zhang, Luyao Cheng, Hai Yu, Xin Zhang, Xiang Lyu, Tianyu Zhao, et al. Drvoice: Parallel speech-text voice conversation model via dual-resolution speech representations. In *International Conference on Learning Representations*, volume 2026, pp. 121421–121440, 2026.

Yi Tang. Poisoning the diffusion: A simple and robust watermarking method for audio generation. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2025.

Shangqing Tu, Yuliang Sun, Yushi Bai, Jifan Yu, Lei Hou, and Juanzi Li. Waterbench: Towards holistic evaluation of watermarks for large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1517–1542, 2024.

Yihan Wu, Zhengmian Hu, Junfeng Guo, Hongyang Zhang, and Heng Huang. A resilient and accessible distribution-preserving watermark for large language models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 53443–53470. PMLR, 21–27 Jul 2024.

Yihan Wu, Ruibo Chen, Georgios Milis, Junfeng Guo, and Heng Huang. A watermark for auto-regressive speech generation models. In *Interspeech 2025*, pp. 3474–3478, 2025a. doi: 10.21437/Interspeech.2025-2328.

Yihan Wu, Georgios Milis, Ruibo Chen, and Heng Huang. Robust distortion-free watermark for autoregressive audio generation models. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 110038–110067. Curran Associates, Inc., 2025b. doi: 10.52202/085713-3677.

Junzuo Zhou, Jiangyan Yi, Tao Wang, Jianhua Tao, Ye Bai, Chu Yuan Zhang, Yong Ren, and Zhengqi Wen. Traceablespeech: Towards proactively traceable text-to-speech with watermarking. In *Interspeech 2024*, pp. 2250–2254, 2024. doi: 10.21437/Interspeech.2024-534.

A Appendix

A.1 Ablation Study

Key Verification Ablation. We evaluate different conditions of the pub-key and pri-key in Table 4. Condition A measures acceptance and recovery with both parameters correct, which is the legal verifier.

We examine the complementary roles of the pub-key and pri-key by evaluating all four combinations of correct and incorrect keys in Table 4. This design separates candidate assessment from public verification: the pri-key scores candidate payloads before aggregation, whereas the pub-key guides post-voting refinement and verifies the resulting payload.

With both keys correct, MARC achieves a TPR@FPR=1% of 0.94, a -log($cluster_p$) of 37.08, and a bit_acc of 0.98. Keeping the pub-key correct but replacing the pri-key reduces these values to 0.58, 8.06, and 0.66, respectively. Thus, access to the correct pub-key alone does not reproduce the recovery performance obtained with the correct pri-key, supporting private-key-guided candidate assessment. Conversely, using the correct pri-key with an incorrect pub-key yields a bit_acc of 0.78 and a TPR@FPR=1% of 0.76, demonstrating the additional contribution of public-key-guided refinement and verification. When both keys are incorrect, bit_acc falls to 0.52. Together, these results support the use of both keys at complementary decoding stages: private-key scoring identifies promising candidates, while the pub-key provides an additional constraint on the aggregated result.

Table 4: Identity Verification.

Condition	pub-key	pri-key	TPR@FPR=1% ↑	-log($cluster_p$)↑	bit_acc ↑
A	Correct	Correct	0.94	37.08	0.98
B	Correct	Incorrect	0.58	8.06	0.66
C	Incorrect	Correct	0.76	15.32	0.78
D	Incorrect	Incorrect	0.49	5.08	0.52

Clustering Ablation. Table 5 compares the full codec-union construction (C0) with GMM + retokenization (C1), geometric GMM (C2), and GMM + retokenization + CNBP (C3). The cluster count and construction data budget are held fixed.

Codec transformations can replace a token with another token that is not its nearest geometric neighbor. MARC therefore combines intrinsic token geometry with substitution information from multiple codecs, aiming to preserve encoded symbols across different reconstruction processes. Table 5 compares the full codec-union construction (C0) with GMM + retokenization (C1), Geometric GMM (C2), and GMM + retokenization + CNBP (C3), while keeping the cluster count and construction data budget fixed.

On clean audio, C0 achieves the highest bit_acc of 98.17% and the lowest FAD of 0.2622 among the evaluated variants, showing that incorporating codec information can support recovery without requiring greater distributional distortion. Across the eight codec attacks, C0 also achieves the highest mean bit_acc of 97.77%, compared with 97.24% for C1, 97.41% for C2, and 97.06% for C3. For example, under EnCodec, C0 reaches 99.25% bit_acc, compared with 97.92%, 98.25%, and 96.92%, respectively.

These results support the value of combining substitution evidence across codecs rather than relying solely on geometric proximity or the evaluated retokenization-based refinements. The resulting clusters provide a representation in which codec-induced token changes can preserve the embedded symbol, supporting robust payload recovery across heterogeneous codecs.

Table 5: Clustering ablations on FinanceQA.

(a) Clean performance

ID	Clustering	TPR@FPR=1% ↑	$-\log(\text{cluster_p})$↑	bit_acc(%) ↑	FAD ↓	MOS ↑	
						NISQA	DNSMOS
C0	Full (codec union)	96.00	37.08	99.17	0.2622	4.503	3.868
C1	GMM + retokenization	96.00	32.84	99.00	0.3791	4.508	3.889
C2	Geometric GMM	95.00	36.96	98.75	0.3799	4.512	3.800
C3	GMM + retokenization + CNBP	91.00	35.20	98.00	0.4137	4.462	3.855

(b) Codec robustness

ID	Metric	SpeechTokenizer	EnCodec (24 kbps)	MP3 (32 kbps)	Opus (16 kbps)	DAC (44.1 kHz)	FACodec (16 kHz)	SoCodec	SemantiCodec
C0	TPR@FPR=1% ↑	90.00	97.00	87.00	96.00	94.00	92.00	79.00	85.00
	$-\log(\text{cluster_p})$↑	25.33	30.70	24.11	29.20	28.84	30.29	23.03	24.02
	bit_acc(%) ↑	97.58	99.25	97.17	95.00	98.33	98.25	95.50	97.08
C1	TPR@FPR=1% ↑	86.00	92.00	82.00	92.00	95.00	94.00	70.00	89.00
	$-\log(\text{cluster_p})$↑	23.67	26.98	23.99	28.12	26.65	29.35	22.99	24.02
	bit_acc(%) ↑	97.17	97.92	96.33	97.92	98.92	98.75	93.83	97.08
C2	TPR@FPR=1% ↑	88.00	92.00	86.00	97.00	95.00	92.00	76.00	80.00
	$-\log(\text{cluster_p})$↑	24.56	27.95	24.33	29.43	28.50	32.26	24.17	24.18
	bit_acc(%) ↑	97.25	98.25	96.83	99.33	99.08	98.08	94.83	95.67
C3	TPR@FPR=1% ↑	85.00	87.00	82.00	91.00	92.00	91.00	77.00	86.00
	$-\log(\text{cluster_p})$↑	24.90	26.10	25.06	28.53	28.89	30.09	23.40	24.33
	bit_acc(%) ↑	97.00	96.92	96.17	98.00	98.33	98.08	95.08	96.92

Encoding Ablation. We compare the full symbol schedule with alternative scheduling and redundancy configurations in table 6. MARC organizes payload embedding into laps and bursts to provide repeated, structured observations for decoding. We evaluate this design in Table 6 by removing codec union (E1), using a single lap (E2), setting the burst length to one (E3), or replacing the full schedule with round-robin scheduling (E4).

The full configuration achieves a mean post-codec bit_acc of 97.77%, compared with 97.24% without codec union, 94.71% with a single lap, 96.42% with unit bursts, and 92.41% with round-robin scheduling. The full schedule outperforms E2-E4 in both bit_acc and TPR@FPR=1% under every evaluated codec attack. The benefit of multiple laps is particularly clear under SemantiCodec, where bit_acc increases from 91.08% for E2 to 97.08% for the full configuration. Structured scheduling also improves EnCodec recovery from 92.58% with round-robin scheduling to 99.25%.

The unit-burst comparison further distinguishes clean extraction from robustness after reconstruction. Although E3 achieves 99.42% bit_acc on clean audio, the full configuration provides higher recovery accuracy under all eight codec attacks, including 99.25% versus 96.50% under EnCodec. This pattern supports the use of bursts to provide repeated local evidence, together with multiple laps that distribute payload observations across the sequence. The full configuration also has the lowest clean-audio FAD among the evaluated encoding variants, demonstrating a favorable combination of recovery robustness and audio fidelity.

Decoding Ablation. Table 7 compares the full decoder (D0) with two variants: joint removal of the lap pipeline and HiPT grid search (D1), and removal of boundary sliding (D2).

Even when watermark evidence survives codec reconstruction, payload recovery requires assigning that evidence to the correct symbol positions. MARC addresses this requirement through structured alignment search and local boundary correction.

Removing the lap pipeline and HiPT grid search reduces clean bit_acc from 99.17% to 84.75% and mean post-codec bit_acc from 97.77% to 74.57%. The difference is particularly pronounced

Table 6: Embedding ablations on FinanceQA.

(a) Clean performance

ID	Embedding	TPR@FPR=1% ↑	$-\log(\text{cluster_p})$↑	bit_acc(%) ↑	FAD ↓	MOS ↑	
						NISQA	DNSMOS
E0	Full	96.00	37.08	99.17	0.2622	4.503	3.868
E1	w/o codec union	96.00	32.84	99.00	0.2791	4.508	3.889
E2	Single lap ($L = 1$)	90.00	34.86	97.58	0.2777	4.498	3.868
E3	Unit burst ($B = 1$)	98.00	30.32	99.42	0.2718	4.523	3.873
E4	Round-robin scheduling	81.00	26.35	95.92	0.2726	4.480	3.812

(b) Codec robustness

ID	Metric	SpeechTokenizer	EnCodec (24 kbps)	MP3 (32 kbps)	Opus (16 kbps)	DAC (44.1 kHz)	FACodec (16 kHz)	SoCodec	SemantiCodec
E0	TPR@FPR=1% ↑	90.00	97.00	87.00	96.00	94.00	92.00	79.00	85.00
	$-\log(\text{cluster_p})$↑	25.33	30.79	24.11	29.20	28.84	30.29	23.03	24.38
	bit_acc(%) ↑	97.58	99.25	97.17	99.00	98.33	98.25	95.50	97.08
E1	TPR@FPR=1% ↑	86.00	92.00	82.00	92.00	95.00	94.00	70.00	89.00
	$-\log(\text{cluster_p})$↑	23.67	26.98	23.99	28.12	26.65	29.35	22.99	24.02
	bit_acc(%) ↑	97.17	97.92	96.33	97.92	98.92	98.75	93.83	97.08
E2	TPR@FPR=1% ↑	80.00	91.00	77.00	88.00	91.00	87.00	64.00	68.00
	$-\log(\text{cluster_p})$↑	26.23	27.58	24.48	29.07	29.89	30.25	24.49	24.25
	bit_acc(%) ↑	94.42	97.33	94.08	97.08	97.42	96.42	89.83	91.08
E3	TPR@FPR=1% ↑	83.00	83.00	83.00	93.00	92.00	91.00	68.00	78.00
	$-\log(\text{cluster_p})$↑	23.52	23.90	22.96	23.97	24.52	24.29	21.46	22.86
	bit_acc(%) ↑	96.50	96.50	96.42	98.08	97.67	98.17	92.67	95.33
E4	TPR@FPR=1% ↑	63.00	65.00	57.00	68.00	68.00	74.00	73.00	57.00
	$-\log(\text{cluster_p})$↑	20.52	20.62	19.54	21.03	20.55	21.07	20.01	19.81
	bit_acc(%) ↑	92.33	92.58	90.58	93.00	92.58	94.25	93.42	90.58

under SoCodec, where recovery increases from 64.67% with D1 to 95.50% with the full decoder. Boundary sliding provides an additional, separately measured benefit: removing it lowers mean post-codec bit_acc to 94.84%, with SemantiCodec recovery decreasing from 97.08% to 91.92%. The full decoder improves both bit_acc and TPR@FPR=1% over D1 and D2 under every evaluated codec attack.

Notably, -log($cluster_p$) remains unchanged across the decoder variants on clean audio and seven of the eight codec attacks, despite substantial differences in payload recovery. This distinction demonstrates why strong cluster-level evidence alone is insufficient for accurate multi-bit decoding. MARC's decoding pipeline converts preserved watermark evidence into recoverable payload information by resolving its alignment, while boundary sliding further corrects local positional errors.

A.2 Supplementary Robustness Experiments

We have included here some experimental data on other parameter settings not covered in the main table. As shown in Table 8, Table 9 and Table 10.

For conventional audio attacks. We first evaluate the robustness of MARC during audio transmission and processing by means of conventional audio attacks. Such attacks perturb the audio through operations such as noise addition, filtering, compression and temporal editing, with the aim of weakening the watermark evidence or disrupting the correspondence between embedding and extraction. This experiment examines whether the watermark can still be detected after the audio content has undergone common processing, and whether the carried message can be correctly recovered, thereby providing a basis for assessing the practical usability of the method.

In conventional attacks, we focus on the codec attacks. Codec attacks introduce structural token substitutions and alignment changes by re-representing and synthesizing audio, thereby directly challenging watermarking methods that rely on discrete speech representations. Even if the re-

Table 7: Decoder ablations on FinanceQA.

(a) Clean performance

ID	Decoder	TPR@FPR=1% ↑	$-\log(\text{cluster_p})$↑	bit_acc(%) ↑	FAD ↓	MOS ↑	
						NISQA	DNSMOS
D0	Full	96.00	37.08	99.17	0.2622	4.503	3.868
D1	w/o lap pipeline and HiPT grid	66.00	37.08	84.75	0.3021	4.498	3.842
D2	w/o boundary sliding	92.00	37.08	97.75	0.2932	4.486	3.874

(b) Codec robustness

ID	Metric	SpeechTokenizer	EnCodec (24 kbps)	MP3 (32 kbps)	Opus (16 kbps)	DAC (44.1 kHz)	FACodec (16 kHz)	SoCodec	SemantiCodec
D0	TPR@FPR=1% ↑	90.00	97.00	87.00	96.00	94.00	92.00	78.00	85.00
	$-\log(\text{cluster_p})$↑	25.33	30.79	24.11	29.20	28.84	30.29	23.03	24.02
	bit_acc(%) ↑	97.58	99.25	97.17	99.00	98.33	98.25	95.50	97.08
D1	TPR@FPR=1% ↑	38.00	53.00	31.00	52.00	56.00	54.00	19.00	26.00
	$-\log(\text{cluster_p})$↑	25.33	30.79	24.11	29.20	28.84	30.29	23.03	23.84
	bit_acc(%) ↑	74.25	80.67	71.00	79.83	81.33	79.08	64.67	65.75
D2	TPR@FPR=1% ↑	81.00	84.00	74.00	86.00	85.00	81.00	66.00	65.00
	$-\log(\text{cluster_p})$↑	25.33	30.79	24.11	29.20	28.84	30.29	23.03	24.05
	bit_acc(%) ↑	95.42	96.25	94.17	96.83	96.25	95.67	92.25	91.92

constructed audio retains a similar auditory quality, the re-extracted token sequence may undergo significant changes.

MARC addresses this issue by organizing watermark information within a symbolic representation at the cluster level, enabling partial token changes to retain the same watermark meaning, whilst compensating for local evidence loss through repeated observation and aggregated decoding. This experiment aims to verify whether these designs can transform instability at the token level into stability at the message level, thereby supporting detection and recovery following codec transformation.

We further examine attribution interference when an attacker actively introduces another watermark message. We therefore consider an overwriting attack, the aim of which is to weaken the original owner's watermark evidence whilst simultaneously establishing valid support for the attacker's message, thereby causing the audio to be erroneously attributed to the attacker. Consequently, this experiment must simultaneously examine the preservation of the original watermark and the effectiveness of the attacker's watermark insertion: the failure of the original watermark indicates removal, the acceptance of both watermarks indicates coexistence, whilst the failure of the original watermark coupled with the validity of the attacker's watermark constitutes a successful replacement.

For MARC, this evaluation focuses on whether multi-bit message recovery and corresponding verification can distinguish between different attribution claims, and to what extent the distributed evidence of the original message can be preserved under active interference. Combined with codec attacks, the two types of experiments respectively test information retention capability under representational transformations and attribution reliability under competing watermarks, reflecting the method's design objectives for audio re-encoding and active watermark replacement.

Supplementary robustness evaluation. We extend the main evaluation with additional compression bitrates, additive noise, and temporal shifts on FinanceQA, LibriSpeech, and Dolly. Tables 8, 9, and 10 report the results. After excluding duplicate codec configurations, MARC achieves the highest $-\log_{10}(\text{cluster_p})$ among all compared methods, attack settings, while maintaining bit_acc between 94.58% and 99.25%. These results extend the main findings across both parameter settings and datasets, demonstrating the preservation of watermark evidence alongside accurate multi-bit recovery.

Robustness across compression settings. Compression can change the token sequence even when the reconstructed audio preserves its perceptual content. The additional MP3, Opus, and EnCodec settings therefore test whether MARC's recovery depends on a particular codec or bitrate.

Table 8: Robustness Supplementary.
(a) SpiritLM with FinanceQA

Attacks	AR speech WM	Aligned-IS	HiPT	MARC		
	$-\log_{10}(\text{cluster_p})\uparrow$	$-\log_{10}(\text{cluster_p})\uparrow$	$-\log_{10}(\text{cluster_p})\uparrow$	TPR@FPR=1% ↑	$-\log_{10}(\text{cluster_p})\uparrow$	bit_acc ↑
Codec & Regeneration						
MP3 (32 kbps)	2.148	3.436	13.310	87.00	24.107	97.17
MP3 (40 kbps)	2.193	3.111	13.065	86.00	24.217	96.83
Opus (31 kbps)	3.330	4.390	15.140	97.00	31.112	99.25
Opus (32 kbps)	3.538	4.374	14.537	92.00	30.836	98.25
Opus (64 kbps)	3.524	4.474	14.621	94.00	31.254	98.92
EnCodec (3k)	2.254	3.724	13.262	91.00	25.269	97.83
EnCodec (6k)	3.050	4.243	13.255	92.00	25.653	97.92
EnCodec (12k)	3.201	4.490	13.766	95.00	28.162	98.83
Signal Processing						
Gaussian noise (σ=0.01)	2.702	3.901	14.322	85.00	21.319	96.75
Desynchronization						
Shift (1/2)	1.091	2.619	12.968	75.00	21.915	94.58

Table 9: Robustness Supplementary.
(a) SpiritLM with LibriSpeech

Attacks	AR speech WM	Aligned-IS	HiPT	MARC		
	$-\log_{10}(\text{cluster_p})\uparrow$	$-\log_{10}(\text{cluster_p})\uparrow$	$-\log_{10}(\text{cluster_p})\uparrow$	TPR@FPR=1% ↑	$-\log_{10}(\text{cluster_p})\uparrow$	bit_acc ↑
Codec & Regeneration						
MP3 (32 kbps)	1.531	3.605	8.905	87.00	21.939	97.08
MP3 (40 kbps)	1.463	3.568	8.881	90.00	21.989	97.75
Opus (31 kbps)	2.982	5.454	9.401	92.00	25.477	97.92
Opus (32 kbps)	3.092	5.256	9.697	91.00	26.375	97.83
Opus (64 kbps)	3.032	5.115	9.645	93.00	26.592	98.25
EnCodec (3k)	1.751	4.725	9.860	91.00	22.415	97.83
EnCodec (6k)	2.599	4.944	10.070	85.00	23.156	96.42
EnCodec (12k)	2.876	5.121	9.755	86.00	23.336	97.00
Signal Processing						
Gaussian noise (σ=0.01)	2.649	4.431	10.003	86.00	19.792	96.75
Desynchronization						
Shift (1/2)	0.809	2.751	9.296	77.00	20.744	95.00

Across all codec configurations evaluated over the three datasets, MARC maintains bit_acc of 96.00-99.25% and TPR@FPR=1% of 84.00-97.00%. Under MP3 at 32 kbps, bit_acc remains at 97.17% on FinanceQA, 97.08% on LibriSpeech, and 96.00% on Dolly. For Opus at 31, 32, and 64 kbps, all nine dataset bit_acc of at least 97.83%, showing consistently accurate recovery across the tested operating points.

The neural-codec results provide further evidence for MARC's representation-level design. Even at an EnCodec bandwidth of 3 kbps, MARC achieves bit_acc of 97.83%, 97.83%, and 97.58% on FinanceQA, LibriSpeech, and Dolly, respectively. Its corresponding $-\log_{10}(\text{cluster_p})$ scores

Table 10: Robustness Supplementary.
(a) SpiritLM with Dolly

Attacks	AR speech WM	Aligned-IS	HiPT	MARC		
	$-\log_{10}(\text{cluster_p})\uparrow$	$-\log_{10}(\text{cluster_p})\uparrow$	$-\log_{10}(\text{cluster_p})\uparrow$	TPR@FPR=1% ↑	$-\log_{10}(\text{cluster_p})\uparrow$	bit_acc ↑
Codec & Regeneration						
MP3 (32 kbps)	1.596	3.945	17.639	84.00	23.721	96.00
MP3 (40 kbps)	1.820	3.861	17.025	87.00	24.118	97.00
Opus (31 kbps)	2.988	5.324	17.639	91.00	29.413	98.00
Opus (32 kbps)	3.094	5.393	17.639	95.00	29.557	98.75
Opus (64 kbps)	3.214	5.459	17.807	95.00	29.250	98.83
EnCodec (3k)	1.975	4.250	17.451	87.00	24.138	97.58
EnCodec (6k)	2.486	4.918	17.947	90.00	25.051	98.17
EnCodec (12k)	2.980	5.123	17.874	93.00	25.448	98.50
Signal Processing						
Gaussian noise (σ=0.01)	2.363	5.150	16.887	85.00	20.805	96.58
Desynchronization						
Shift (1/2)	0.866	2.710	17.085	78.00	22.579	95.17

are 25.269, 22.415, and 24.138, compared with 13.262, 9.860, and 17.451 for HiPT, the strongest baseline on this metric in these settings. These results are consistent with the motivation for multi-codec clustering: substitutions within a cluster can preserve the encoded symbol despite changes in individual token identities. Repeated observations and aggregated decoding further support recovery when some symbol evidence is disrupted.

Robustness to noise and temporal offsets. The additional Gaussian-noise condition uses a fixed standard deviation of 0.01, testing recovery under an absolute perturbation scale. MARC retains bit_acc of 96.75% on FinanceQA, 96.75% on LibriSpeech, and 96.58% on Dolly, while achieving the highest $-\log_{10}(\text{cluster_p})$ on all three datasets. This demonstrates that the observed robustness extends beyond codec reconstruction to direct waveform corruption.

The half-frame shift tests a different requirement: assigning preserved evidence to the correct payload positions after temporal displacement. Under this condition, MARC achieves bit_acc of 94.58%, 95.00%, and 95.17% on the three datasets. Its cluster-evidence scores also exceed those of all baselines in each case. Accurate recovery under this offset is consistent with the role of alignment-aware decoding and boundary correction in resolving positional mismatch.

Consistency across datasets. The same pattern holds across FinanceQA, LibriSpeech, and Dolly: MARC combines stronger cluster-level evidence than the evaluated baselines with high payload recovery accuracy under every supplementary setting. This consistency supports the complementary design of cluster-level encoding, repeated payload scheduling, and alignment-aware decoding. Together, they address changes in token identity, partial corruption of symbol evidence, and displacement of extraction positions, extending watermark robustness from presence detection to recoverable multi-bit information.

A.3 Attack Settings

Let $\mathbf{x} \in \mathbb{R}^T$ denote a watermarked waveform sampled at $f_s = 16$ kHz. Each attack is applied independently to the original waveform:

$$\widetilde{\mathbf{x}} = \mathcal{A}_\theta(\mathbf{x}), \tag{9}$$

where θ specifies the attack parameters. We evaluate both watermark evidence and payload recovery, since retaining detectable evidence does not necessarily preserve its correspondence to message positions.

Signal Processing. These attacks model acoustic interference, bandwidth restriction, and waveform processing during transmission and redistribution. They test whether watermark information survives changes in acoustic content and the resulting token assignments.

- **Gaussian noise.** We evaluate two conditions: an SNR of 30 dB and a fixed noise standard deviation of 0.01. The attacked waveform is

$$\widetilde{x}_t = x_t + n_t, \qquad n_t \sim \mathcal{N}(0, \sigma^2). \tag{10}$$

  For the SNR-controlled condition,

$$\sigma^2 = \left(\frac{1}{T} \sum_{t=0}^{T-1} x_t^2 \right) 10^{-30/10}. \tag{11}$$

  These conditions test recovery under broadband interference at relative and absolute noise scales.

- **Pink noise.** We use the implementation's pink-noise routine, which constructs a correlated sequence $\mathbf{z}$ from cumulative sums of Gaussian samples. The sequence is peak-normalized before addition:

$$\widetilde{\mathbf{x}} = \mathbf{x} + 0.01 \frac{\mathbf{z}}{\|\mathbf{z}\|_\infty}. \tag{12}$$

  Here, 0.01 specifies the peak noise amplitude rather than its standard deviation. This condition complements independent Gaussian noise by introducing temporally correlated interference.

- **High-pass and low-pass filtering.** We apply Julius high-pass and low-pass filters with cutoff frequencies of 1 kHz and 3 kHz, respectively. Their normalized cutoffs are $1000/16000$ and $3000/16000$. These attacks model frequency-selective processing and test recovery after different acoustic components are suppressed.
- **Smoothing.** We apply a six-sample moving-average filter:

$$z_t = \frac{1}{6}\sum_{j=0}^{5} x_{t+j}, \qquad 0 \le t < T-5. \tag{13}$$

  The valid filtering output is right-padded with zeros to restore length T. This models local waveform smoothing and tests sensitivity to the removal of rapid acoustic variations.
- **Echo.** We apply a two-tap kernel with coefficients 1 and 0.05, separated by 800 samples (50 ms), using `julius.fft_conv1d`. The filtered output is rescaled to the original peak amplitude and right-padded to length T. This condition tests recovery under delayed acoustic interference while controlling the output peak level.
- **Quantization.** We uniformly quantize waveform amplitudes to 64 levels between the waveform minimum a and maximum b:

$$\widetilde{x}_t = a + \frac{b-a}{63}\,\mathrm{round}\left(63\frac{x_t-a}{b-a}\right). \tag{14}$$

  Constant waveforms are left unchanged. This models reduced amplitude precision during storage or processing.
- **Up/down resampling.** We resample from 16 to 32 kHz and then back to 16 kHz using Julius:

$$\widetilde{\mathbf{x}} = \mathcal{R}_{32\to16}\left(\mathcal{R}_{16\to32}(\mathbf{x})\right). \tag{15}$$

  The output retains the original sample count. This tests the effects of interpolation and filtering during sample-rate conversion between audio systems.

MARC addresses these perturbations through cluster-level encoding: a token substitution need not change the encoded symbol when cluster membership is preserved. Repeated payload observations and cross-lap aggregation provide additional recovery evidence when individual observations are corrupted. This category therefore assesses recovery without requiring exact preservation of the original waveform.

Desynchronization. These attacks model temporal editing and partial observation. They challenge either the correspondence between embedding and extraction positions or the availability of evidence at those positions.

- **Speed perturbation.** We use `librosa.effects.time_stretch` with rates $\alpha \in \{0.9, 1.1\}$. This operation changes duration while approximately preserving pitch. The transformed waveform is then matched to the original length:

$$\widetilde{\mathbf{x}} = \mathcal{M}_T\left(\mathrm{TS}_\alpha(\mathbf{x})\right). \tag{16}$$

  The 1.1 condition adds trailing zero padding, whereas the 0.9 condition truncates the stretched signal. These settings model playback-rate editing and test recovery under temporal distortion and its associated boundary effects.
- **Cropping.** We retain the first 50% of the waveform without padding:

$$\widetilde{\mathbf{x}} = \mathbf{x}[0 : \lfloor T/2 \rfloor]. \tag{17}$$

  This models excerpt sharing and partial audio availability, testing whether the retained segment contains sufficient evidence for payload recovery.
- **Temporal shifts.** We circularly shift the waveform to the right by $1/8$ or $1/2$ of a 320-sample frame:

$$\widetilde{x}_t = x_{(t-\Delta) \bmod T}, \qquad \Delta \in \{40, 160\}. \tag{18}$$

  These offsets correspond to 2.5 and 10 ms. Circular shifting preserves all samples while changing their alignment with extraction boundaries, isolating sensitivity to temporal offsets.

- **Shush.** We replace the first 1% of samples with zeros:

$$\widetilde{x}_t = \begin{cases} 0, & t < \lfloor 0.01T \rfloor, \\ x_t, & \text{otherwise.} \end{cases} \tag{19}$$

  This models onset muting and tests resilience to missing initial evidence. Unlike cropping, it preserves the waveform length and the positions of subsequent samples.

MARC distributes repeated payload observations across laps to support recovery from incomplete evidence. Alignment search and boundary sliding address positional mismatch, while cross-lap aggregation combines surviving observations. These experiments test whether watermark evidence remains usable for message recovery after temporal editing.

Codec & Regeneration. These attacks model lossy storage, bandwidth-constrained transmission, and reconstruction through alternative audio representations. For codec c with configuration r, the round trip is

$$\widetilde{\mathbf{x}} = \mathrm{Dec}_{c,r}\left(\mathrm{Enc}_{c,r}(\mathbf{x})\right). \tag{20}$$

- **MP3.** We evaluate encoding and decoding at 32, 40, and 64 kbps. These settings test robustness across compression strengths encountered during storage and redistribution.
- **Opus.** We evaluate round-trip compression at 16, 31, 32, and 64 kbps. These configurations model different bandwidth constraints in communication and audio delivery.
- **EnCodec.** We evaluate round-trip reconstruction at 3, 6, 12, and 24 kbps using the 24 kHz codec. These settings test recovery after neural encoding, quantization, and reconstruction at different supported bandwidths.
- **DAC.** We use the default 44.1 kHz DAC model. The input is resampled from 16 to 44.1 kHz before reconstruction, and the output is returned to 16 kHz and matched to the original length. This evaluates an alternative neural-codec representation without imposing an additional bitrate constraint.
- **SpeechTokenizer.** We perform tokenization followed by waveform reconstruction. This tests whether payload information survives processing through a different speech-token representation.
- **FaCodec.** We apply the FaCodec reconstruction backend. This adds another learned analysis, synthesis transformation to the evaluation.
- **VoiceCraft-EnCodec.** We perform an encoding-decoding round trip using VoiceCraft's four-codebook EnCodec checkpoint. This evaluates its codec component rather than text-conditioned speech editing or generation.
- **SoCodec.** We apply SoCodec analysis, synthesis at 16 kHz. The wrapper prepends 8000 zero samples (0.5 s) before processing and removes the corresponding prefix from the reconstructed output. The remaining waveform is matched to the original length.
- **SemantiCodec.** We use the configuration with a token rate of 50 and a semantic vocabulary size of 16384. The waveform is encoded and decoded using the corresponding checkpoint, and the output is matched to the original length.

Codec reconstruction can preserve perceptual content while changing the token sequence used for watermark extraction. MARC addresses this challenge by combining intrinsic token geometry with multi-codec substitution information. For an original token y_t and its reconstructed counterpart $\widetilde{y}_{\pi(t)}$, cluster-level encoding permits

$$\widetilde{y}_{\pi(t)} \neq y_t \quad \text{while} \quad c(\widetilde{y}_{\pi(t)}) = c(y_t), \tag{21}$$

where $\pi(t)$ denotes the temporal correspondence when one exists. Within-cluster substitutions therefore preserve the encoded symbol. Together with repeated embedding and alignment-aware decoding, this design targets payload recovery across heterogeneous reconstruction processes.