SIMPLIFYING ROBUST AUDIO WATERMARKING

Anonymous authors
Paper under double-blind review

ABSTRACT

Multi-bit audio watermarking models are generally found vulnerable under deep watermark removal attacks, such as neural codec resynthesis and voice cloning. Existing methods jointly apply strategies like attack simulation, latent watermarking, and perceptual losses to improve robustness while preserving imperceptibility. However, we discover that most of these strategies are not necessary to achieve state-of-the-art robustness. To figure out key designs that enable robust audio watermarking, we propose SimpleMark, a simple yet effective audio watermarking framework. SimpleMark narrows down robust audio watermarking to two design choices: First, we embed the watermark through **bounded magnitude scaling** to control imperceptibility. Second, we detect watermark from **low-level audio representations** using a lightweight readout head to achieve robustness. Following these two designs, we build a series of lightweight audio watermarking models that surpass existing complex methods in robustness while maintaining competitive imperceptibility.

1 INTRODUCTION

The proliferation of high-fidelity audio generation models (Hu et al., 2026a; Chen et al., 2024) has made it easy to produce deepfake audio content indistinguishable from bona fide recordings. However, passive deepfake detectors still suffer from poor generalization ability (Müller et al., 2024). To complement this, audio watermarking emerges as a proactive approach for deepfake detection, audio provenance, and copyright protection (Wu et al., 2025). By embedding an imperceptible marker into audio before distribution, one can trace the origin of the audio by detecting the watermark.

A core challenge in audio watermarking lies in its robustness against removal attacks. A robust audio watermark must survive not only conventional audio edits such as additive noise, but also deep transformations such as neural codec resynthesis and voice cloning. To achieve versatile robustness, recent audio watermarking systems often adopt several attack-specific mechanisms to avoid "shallow" watermarking (O'Reilly et al., 2025): simulated attacks expose the model to audio edits during training (Chen et al., 2023; San Roman et al., 2024), while latent watermarking aims to embed watermarks in an attack-invariant latent space (Ji et al., 2025; Li et al., 2025; 2026b). Meanwhile, multiple perceptual losses are applied to preserve watermarked audio quality (San Roman et al., 2024; Li et al., 2025; 2026a). Combining these mechanisms into a tightly coupled training pipeline complicates optimization and obscures the contribution of each component. This growing complexity raises a fundamental question: **Which design choices are essential for robust audio watermarking, and which are unnecessary?**

Instead of adding another layer of complexity, we investigate this question by simplifying the design space of audio watermarking. We remove all simulated attacks from the conventional embedder–attack simulation–detector framework, avoid perceptual losses, and simplify the architecture of the watermark embedder and detector. Surprisingly, we find that most explicit designs for robustness in existing audio watermarking methods are redundant: even with only two linear modules, we can build an audio watermarking model that achieves competitive robustness against various attacks (See Appendix A). This example indicates that a "shallow" watermark does not, by itself, imply inferior robustness.

Motivated by this finding, we propose SimpleMark, a simple yet effective multi-bit audio watermarking framework. SimpleMark is built upon two design choices that progressively remove redundancy. First, we recognize the watermark detector, rather than the watermark embedder, as the

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.

primary source of robustness. In the watermark detector, a lightweight head decodes the watermark from a low-level audio representation, inheriting robustness from the representation itself instead of a limited set of simulated attacks. Second, with the detector carrying most of the robustness burden, we reduce watermark embedding from additive perturbation in a latent space to a multiplicative magnitude scaling process in the spectral domain. Meanwhile, by bounding the scaling factors, this multiplicative embedding operation naturally supports explicit imperceptibility control, allowing us to remove perceptual losses from the training objective.

SimpleMark shows that bounded magnitude scaling and a suitable low-level audio representation are all you need to build a robust audio watermarking model: SimpleMark consistently achieves superior robustness and competitive imperceptibility with different embedder architectures and audio representations, even though no attacks are seen during training. Ablation studies show the importance of bounded magnitude scaling and low-level audio representations, whereas extensive analytic experiments show that some previous designs, like attack simulation and perceptual losses, may lose their effectiveness under the proposed SimpleMark framework.

In conclusion, our contribution is three-fold:

1. We find that "shallow" audio watermark is sufficient to survive deep watermark removal attacks. With only a linear scale predictor and a linear readout head, we can achieve robustness comparable to existing methods using complex watermark embedders and detectors.
2. Inspired by this discovery, we propose SimpleMark, a simple yet effective audio watermarking framework. SimpleMark embeds watermarks via bounded magnitude scaling and detects them in a low-level audio representation. Extensive experiments show that these design choices facilitate better robustness-imperceptibility trade-off without attack simulation or perceptual losses.
3. Under the proposed SimpleMark framework, we derive a series of robust audio watermarking models with different combinations of embedder architectures and audio representations. All these audio watermarking models achieve state-of-the-art robustness against strong attacks like neural codec resynthesis and zero-shot voice cloning, while maintaining competitive perceptual quality.

2 Related Work

Audio watermarking hides machine-detectable information in an audio signal. The hidden information can be either a zero-bit existence label for deepfake detection (Juvela & Wang, 2025) or a multi-bit binary string for source tracing (Chen et al., 2023). A typical audio watermarking system comprises a watermark embedder and a watermark detector. In generative watermarking methods, an audio generation model is modified to serve as the watermark embedder (Liu et al., 2026; Milis et al., 2026). In contrast, post hoc methods build watermark embedders that accept any audio clip (San Roman et al., 2024). In this work, we focus on *multi-bit post hoc* audio watermarking that supports both deepfake detection and source tracing of all kinds of audio generation models.

Audio watermarking systems balance three conflicting requirements: First, *robustness* requires the watermark to be detectable after benign audio processing or deliberate removal attacks. Meanwhile, *imperceptibility* requires the watermark to be perceptually transparent. Finally, *high capacity* encourages the maximum number of message bits to be embedded within a unit-length audio segment. Among them, robustness remains the central challenge because deep transformations such as neural codec resynthesis and voice cloning can remove watermarks while maintaining perceptual quality (O'Reilly et al., 2025; Özer et al., 2025).

Pioneering works on neural audio watermarking found that inserting a differentiable attack simulation module between the watermark embedder and detector effectively enhances robustness against conventional audio edits, such as noise addition, band-pass filtering, and MP3 compression (Chen et al., 2023; San Roman et al., 2024). However, attack simulation itself is not sufficient to defend against deep transformations like neural codec resynthesis and voice cloning. O'Reilly et al. (2025) attribute this fragility to the nature of audio watermarks—"shallow" perturbations in either the raw waveform domain or the spectral domain. Guided by this insight, subsequent works embed watermarks in an "attack-invariant" latent space. To enhance robustness against neural codec reconstruction, a line of work embeds watermarks by manipulating acoustic codes in the discrete latent space of

a vector-quantized variational autoencoder (VQ-VAE) (Ji et al., 2025; Chen et al., 2026b). Another line of work focuses on robustness against voice cloning: LambdaMark (Li et al., 2026b) embeds watermarks in the semantic latent space defined by Dasheng (Dinkel et al., 2024), so that watermarks in training data can be preserved in synthesized speech. In contrast, VoiceMark (Li et al., 2025) embeds watermarks in the acoustic latent space defined by SpeechTokenizer (Zhang et al., 2024), so that watermarks in speech prompts can be detected in synthesized audio. These attack-specific designs limit their generalization ability to other unseen attacks. Additionally, autoencoder-based watermark embedders, especially when trained with simulated attacks, increase the risk of audio quality degradation. Therefore, multiple perceptual losses (Li et al., 2026a) and curriculum learning (Liu et al., 2026; Hu et al., 2026b) are introduced to control imperceptibility and stabilize training.

Meanwhile, we note that there exist simpler methods achieving considerable robustness: with only one type of simulated attack, TimbreWatermark (Liu et al., 2024) achieves robustness against both conventional audio edits and single-speaker voice cloning. AWARE (Pavlović et al., 2025) even proposes a training-free, detector-only audio watermarking model robust against most conventional audio edits. Though these methods fall behind state-of-the-art, they indicate that many commonly adopted designs, including simulated attacks, latent watermarking, and perceptual losses, may be redundant. This motivates SimpleMark, an audio watermarking framework that achieves state-of-the-art robustness without redundant designs.

3 METHOD

3.1 PROBLEM FORMULATION AND THREAT MODEL

Denote the set of all audio waveforms as $\mathbb{X}$ and $\mathbf{x} \in \mathbb{X}$ be an arbitrary audio waveform. Let $\mathbf{m} \in \{0,1\}^K$ be a K-bit watermark message. A typical audio watermarking system comprises a watermark embedder $\mathcal{E} : \mathbb{X} \times \{0,1\}^K \to \mathbb{X}$ and a watermark detector $\mathcal{D} : \mathbb{X} \to \{0,1\}^K$. In our threat model, we consider two parties: *publishers* who expect to track published audio with a watermark, and *attackers* who attempt to remove the watermark to disable tracking.

Publishers. Publishers have access to the embedder $\mathcal{E}$, the detector $\mathcal{D}$, and the registered message $\mathbf{m}$. They publish the watermarked audio $\mathbf{x}_m = \mathcal{E}(\mathbf{x}, \mathbf{m})$ and keep the original audio $\mathbf{x}$ private. For an arbitrary audio $\mathbf{y}$, the publisher determines whether it is (or is derived from) $\mathbf{x}_m$ by comparing the decoded message $\widehat{\mathbf{m}} = \mathcal{D}(\mathbf{y})$ with the registered message $\mathbf{m}$. Their goal is to preserve audio quality while recovering the watermark message after attack $\mathcal{A}$:

$$\text{minimize } \mathbb{E}_{\mathbf{x},\mathbf{m}}[d(\mathcal{D} \circ \mathcal{A} \circ \mathcal{E}(\mathbf{x}, \mathbf{m}), \mathbf{m})], \qquad \text{s.t. } \mathbb{E}_{\mathbf{x},\mathbf{m}}[d_{\text{per}}(\mathcal{E}(\mathbf{x}, \mathbf{m}), \mathbf{x})] \leq \epsilon. \tag{1}$$

where d is Hamming distance, d_{per} is a perceptual distance metric and ϵ is a pre-defined threshold.

Attackers. We consider non-adaptive black-box removal attacks where the attackers observe only published watermarked audio and can process it with a set of attacks $\{\mathcal{A}\}$. They have no access to $\mathcal{E}$ and $\mathcal{D}$, paired unwatermarked and watermarked audio, or the registered ground-truth message. Their aim is to make $\mathbf{m}$ unrecoverable while preserving the utility of the attacked audio. There are two categories of attacks: 1) Audio-to-audio transformations, such as conventional audio edits and neural codec resynthesis; 2) Zero-shot voice cloning, where the attackers attempt to use watermarked utterances as reference prompts to generate new unwatermarked speech.

3.2 SIMPLEMARK: SIMPLIFIED AUDIO WATERMARKING FRAMEWORK

SimpleMark is an audio watermarking framework trained with only a watermark detection loss:

$$\mathcal{L}_{\text{dec}} = \frac{1}{K} \sum_{k=1}^{K} \zeta(\gamma - \bar{m}_k \ell_k). \tag{2}$$

where ζ denotes the softplus function, $\bar{\mathbf{m}} = 2\mathbf{m} - 1$ is the bipolar message, $\boldsymbol{\ell}$ is the logits of the watermark detector, and γ is a predefined classification margin. SimpleMark follows two design choices to achieve robustness and imperceptibility: 1) embedding watermark with bounded magnitude scaling, and 2) detecting watermark within low-level audio representation space.

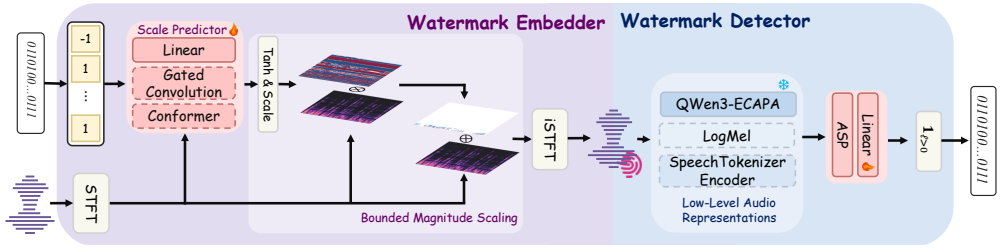


Figure 1: The SimpleMark framework with choices of trainable scale predictor and frozen low-level audio representations. Darker fills indicate higher robustness, and solid borders mark the default choices: linear scale predictor and Qwen3-ECAPA-based detetcor. ASP: Attentive Statistic Pooling.

3.2.1 Bounded Magnitude Scaling-Based Watermark Embedding

The primary objective of watermark embedder is to embed the watermark while maintaining perceptual quality of the host audio. Instead of relying on a specific network architecture to achieve imperceptibility, SimpleMark constrains *where* and *how* the watermark should be embedded.

First, rather than encoding the watermark deep into a latent space, we directly embed the watermark in the STFT magnitude domain to minimize reconstruction artifacts. We further encourage watermark modifications to concentrate on high-energy regions by formulating watermark embedding as a multiplicative scaling process of STFT magnitude. As depicted in Figure 1, a scale predictor first encodes the registered watermark message into per-bin scaling residuals. These raw scaling residuals are then bounded within $(-\tau, \tau)$ via a scaled tanh function. Subsequently, the bounded scaling residuals are multiplied bin-by-bin with the STFT magnitude and then added to the original magnitude. Finally, the perturbed magnitude is combined with the original phase and transformed back to the waveform domain via inverse STFT. Formally, let $\mathrm{STFT}(\mathbf{x}) = \mathbf{M} \odot \exp(j\mathbf{P})$, with $\mathbf{M} \in \mathbb{R}_{\geq 0}^{F \times N}$ and $\mathbf{P} \in \mathbb{R}^{F \times N}$ denoting magnitude and phase, repectively. The watermark embedding process can be formulated as:

$$\widetilde{\mathbf{M}} = \mathbf{M} \odot \mathbf{S}, \tag{3}$$

$$\mathbf{x}_m = \mathcal{E}(\mathbf{x}, \mathbf{m}) = \mathrm{iSTFT}\left[\widetilde{\mathbf{M}} \odot \exp(j\mathbf{P})\right]. \tag{4}$$

where $\widetilde{\mathbf{M}}$ denotes watermarked magnitude and $\mathbf{S} \in \mathbb{R}_{\geq 0}^{F \times N}$ denotes the scaling factor matrix. The scaling factors are centered around 1, with the residual bounded within predefined thresholds $\pm\tau$:

$$\mathbf{S} = \mathbf{1} + \tau \tanh\left[\mathcal{S}(\mathbf{m}, \mathbf{M})\right], \qquad 0 < \tau < 1. \tag{5}$$

with $\mathcal{S}(\mathbf{m}, \mathbf{M})$ denoting an arbitrary scale predictor to be detailed in the next paragraph. Since the perturbation $\Delta\mathbf{M} = \widetilde{\mathbf{M}} - \mathbf{M} = \tau\mathbf{M} \odot \tanh\left[\mathcal{S}(\mathbf{m}, \mathbf{M})\right]$ scales with spectral magnitude $\mathbf{M}$, this formulation naturally allocates larger watermark capacity to high-energy regions while suppressing perturbations in low-energy regions. Meanwhile, fixed scale bounds $(1-\tau, 1+\tau)$ provide an explicit protocol to control watermarked audio quality with per-bin Signal-to-Distortion Ratio $\overline{SDR}$:

$$\tau = 10^{-\overline{SDR}/20}. \tag{6}$$

Choices of the Scale Predictor. The only trainable module of the watermark embedder is a scale predictor $\mathcal{S}(\mathbf{m}, \mathbf{M})$. For each frame, $\mathcal{S}$ takes the registered message $\mathbf{m}$ and magnitude frame $\mathbf{M}_{\cdot i}$ as input to predict raw scaling residuals. SimpleMark does not require a specialized architecture for $\mathcal{S}$. By default, we adopt the simplest **linear** design. It first linearly maps the bipolar message $\bar{\mathbf{m}} = 2\mathbf{m} - 1$ to F dimensions, concatenates the projected message with each F-dimensional log-STFT magnitude frame, and applies a second linear projection to produce raw scaling residuals. In addition to the default linear setting, we also validate other architectures, like the **conformer** (Gulati et al., 2020) architecture in MaskMark (O'Reilly et al., 2024), and a **gated-convolutional** network adapted from TimbreWatermark (Liu et al., 2024). Detailed structure of each scale predictor can be found in Appendix B.2.1. While SimpleMark supports these larger networks, we show in Section 4 that they actually provide minor advantage over the default linear predictor.

3.2.2 LOW-LEVEL AUDIO REPRESENTATION-BASED WATERMARK DETECTION

SimpleMark decodes watermark messages from a low-level, framewise audio representation. Given a watermarked waveform $\mathbf{x}_m$, a frozen representation extractor Φ produces a feature map $\Phi(\mathbf{x}_m) \in \mathbb{R}^{T\times D}$. A subsequent attentive statistics pooling (ASP) (Okabe et al., 2018) module $\mathcal{P}$ aggregates the framewise features into a global vector. Finally, a linear classifier produces the bit logits. Formally,

$$\boldsymbol{\ell} = \mathbf{W}(\mathcal{P} \circ \Phi(\mathbf{x}_m)) + \mathbf{b}. \tag{7}$$

Then the watermark message is predicted by signs the of bit logits:

$$\hat{\mathbf{m}} = \mathcal{D}(\mathbf{x}_m) = \mathbf{1}_{\boldsymbol{\ell}>0}. \tag{8}$$

Choices of the Audio Representation. A suitable audio representation should be both **writable** and **attack-invariant** after temporal pooling. For writability, it should preserve enough fine-grained acoustic details so that watermark information remains decodable. For attack invariance, the watermark decoding subspace defined by the linear classifier should remain stable under attacks. High-level representations like Self-Supervised Learning (SSL) features (Chen et al., 2022) may discard details needed for watermarking, whereas raw representations like STFT may retain fragile perturbations. We therefore consider low-level representations that preserve acoustic details while suppressing attack-sensitive variation, including log-mel spectrograms, Qwen3-ECAPA features (Hu et al., 2026a), and SpeechTokenizer latent features (Zhang et al., 2024; Zhan et al., 2024). Log-Mel spectrogram preserves the spectral envelope while smoothing subtle frequency variations through Mel filtering. Qwen3-ECAPA is the ECAPA-TDNN-based audio encoder in Qwen3-TTS for speaker information extraction (Desplanques et al., 2020), whereas SpeechTokenizer is a neural speech codec trained with a semantic distillation loss. Although their training objectives differ, both combine acoustic modeling with higher-level supervision, namely speaker representation learning for Qwen3-ECAPA and semantic distillation for SpeechTokenizer. This kind of pretraining potentially balances the acoustic details needed for writability and the abstraction needed for attack invariance. Please refer to Appendix B.2.2 for details of QWen3-ECAPA and SpeechTokenizer encoder.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Training and Baselines. We train every SimpleMark instance on 10,000 English utterances from Emilia (He et al., 2024). These systems are compared against six multi-bit post hoc audio watermarking baselines: AudioSeal (San Roman et al., 2024), WavMark (Chen et al., 2023), Timbre (Liu et al., 2024), AWARE (Pavlović et al., 2025), VoiceMark (Li et al., 2025), and AlignMark (Li et al., 2026a). All baselines except Timbre use 16-bit messages, aligning with SimpleMark. For Timbre, we use its official 10-bit checkpoint since no 16-bit checkpoint is available. Additional results with higher message capacity are provided in Appendix C.4.

Benchmarks. We first measure watermark recovery on unattacked audio and then evaluate under four attack families in sequence: conventional audio edits, neural codec resynthesis, zero-shot voice cloning, and other AI attacks. For conventional audio edits, we adopt RAW-Bench (Özer et al., 2025), which contains 18 attacks and a test corpus assembled from nine speech, sound, and music datasets. We next evaluate under 30 neural codecs listed in CodecFake+ (Chen et al., 2026a). For zero-shot voice cloning, we use 500 English test samples from CV3-Eval (Du et al., 2025) and seven representative models. These models include Qwen3-TTS in its x-vector-only mode (Hu et al., 2026a); the in-context systems LongCat-AudioDiT (Xin et al., 2026), F5-TTS (Chen et al., 2024), MaskGCT (Wang et al., 2024), Fish-Speech (Liao et al., 2024), and Spark-TTS (Wang et al., 2025); and a dual-encoder system CosyVoice2 (Du et al., 2024). Finally, we evaluate robustness against other AI attacks in the DeepMark benchmark (Kovačević et al., 2026), including neural vocoding, VAE resynthesis, diffusion resynthesis, and speech enhancement. Unless otherwise specified, all evaluations use 2,620 utterances in LibriSpeech test-clean (Panayotov et al., 2015).

Metrics. For each attack, we report *bitwise accuracy*, the fraction of message bits correctly decoded from watermarked audio. Additional results on *existence accuracy* can be found in Appendix C.1. To assess imperceptibility, we report intrusive metrics PESQ-WB (Rix et al., 2001),

Table 1: Bitwise accuracy and imperceptibility metrics of our proposed system, compared with baselines. **SimpleMark** denotes the system with default configuration, where a linear module is used as the scale predictor and Qwen3-ECAPA embeddings act as the low-level audio representation. Results are averaged over all test audio and all attacks within each family. MOS is reported as mean ± standard deviation over all ratings. †Timbre uses 10-bit messages whereas all other systems use 16 bits. No Atk: No Attack. **Bold**: best; underline: second best.

	Robustness					Imperceptibility				
Method	No Atk	Edits	Codecs	Cloning	AI	PESQ ↑	STOI ↑	ViSQOL ↑	NISQA ↑	MOS ↑
AudioSeal	**100.00**	91.46	50.74	50.20	58.79	4.422	0.998	4.305	4.006	4.69±0.51
WavMark	**100.00**	92.83	54.54	49.87	57.82	4.187	0.997	3.686	4.054	4.06±1.10
Timbre†	**100.00**	**99.72**	60.39	69.79	86.06	3.788	0.992	4.591	3.763	3.93±0.97
AWARE	99.99	98.50	62.82	51.59	87.50	4.239	0.985	4.841	3.693	4.52±0.67
VoiceMark	98.82	87.09	82.93	91.63	81.95	2.111	0.910	2.908	4.062	3.00±1.05
AlignMark	97.68	88.01	84.96	68.13	**88.34**	2.975	0.947	4.411	4.066	4.34±0.78
SimpleMark	99.90	98.09	**84.99**	**92.95**	87.83	3.934	0.960	4.021	3.776	3.78±0.87

STOI (Taal et al., 2011), and ViSQOL (Chinen et al., 2020), together with non-intrusive metric NISQA (Mittag et al., 2021). We also report MOS, a subjective metric on a 5-point scale, where 1 represents obvious distortion and 5 repersents perfect imperceptibility.

4.2 Main Results

In this section, we first compare our default system with baselines to assess the robustness of SimpleMark. We further investigate whether this robustness rely on particular implementation choices by varying the scale predictor and low-level audio representation under the SimpleMark framework.

SimpleMark achieves versatile robustness while preserving imperceptibility. Table 1 compares the default SimpleMark system with existing multi-bit post hoc audio watermarking methods. Although AudioSeal and WavMark achieve superior imperceptibility, they are extremely fragile under deep audio transformations like neural codec resynthesis, voice cloning and other AI attacks. Timbre and AWARE show nearly perfect robustness against conventional audio edits, but they still fail to defend neural codec resynthesis attacks. VoiceMark and AlignMark greatly improve robustness against neural codec resynthesis attacks, and excel in defending voice cloning attacks and AI attacks, respectively. However, as indicated by their low PESQ, these latent watermarking methods sacrifice imperceptibility. Additionally, they exhibit inferior watermark detection accuracy even when no attack is applied to the watermarked audio. In contrast, our proposed SimpleMark system shows superior robustness against all 4 attack families. Our system even achieves new state-of-the-art in robustness against neural codec resynthesis attacks and voice cloning attacks, the generally-agreed biggest challenge to robust audio watermarking. Meanwhile, different from VoiceMark and Alignmark, our system retains nearly perfect watermark detection accuracy without attacks. Finally, SimpleMark maintains competitive imperceptibility, achieving a PESQ score close to 4 and a ViSQOL score above 4. We do not specifically optimize for any single objective quality metric since different metrics produce inconsistent rankings when evaluating watermarked audio quality. For example, AWARE ranks first on ViSQOL but last on NISQA. Similar contradictory can be observed on AlignMark, where the best NISQA score is achieved with a PESQ score below 3. Therefore, we additionally report subjective MOS scores in Figure 2 and table 1, where SimpleMark surpasses VoiceMark and is comparable to WavMark and Timbre. These results indicate that SimpleMark achieves superior robustness while maintaining competitive imperceptibility.

A linear scale predictor is sufficient for robust audio watermark embedding. In addition to the default setting reported in Table 1, SimpleMark consistently achieves superior robustness and competitive imperceptibility with other scale predictor architectures and low-level audio representations. Table 2 evaluates combinations of three scale predictors and three audio representations introduced in Section 3.2. For scale predictors, all three model architectures perform equally well, with Conformer slightly better and gated convolution slightly worse than the default linear module in terms of robustness. This indicates that larger scale predictors provide no consistent robustness or imperceptibility

Table 2: Robustness and imperceptibility across different combinations of scale predictor and low-level audio representations. Robustness columns report bit accuracy (%). **Bold**: best.

			Robustness				Imperceptibility			
Scale Predictor	Representation	#Params.	Edits	Codecs	Cloning	AI	PESQ ↑	STOI ↑	ViSQOL ↑	NISQA ↑
Linear	Log-Mel	0.55M	96.76	80.24	89.19	85.52	3.905	0.966	4.143	3.688
Linear	**Qwen3-ECAPA**	12.6M	**98.09**	84.99	92.95	87.83	3.934	0.960	4.021	3.776
Linear	SpeechTokenizer	121M	96.25	84.75	**94.41**	82.57	**4.041**	0.966	4.024	3.896
Gated Conv.	Log-Mel	0.40M	96.75	79.92	88.37	85.46	3.838	**0.967**	4.152	3.589
Gated Conv.	Qwen3-ECAPA	12.1M	97.51	85.26	93.46	87.19	3.733	0.948	4.068	3.683
Gated Conv.	SpeechTokenizer	121M	95.90	84.32	93.84	80.60	3.803	0.959	4.110	**3.899**
Conformer	Log-Mel	9.52M	97.24	82.83	89.94	86.14	4.009	0.964	**4.159**	3.698
Conformer	Qwen3-ECAPA	21.6M	97.77	**86.80**	94.26	**88.01**	3.771	0.945	4.052	3.750
Conformer	SpeechTokenizer	131M	96.42	85.28	94.34	83.06	3.785	0.958	4.071	3.822

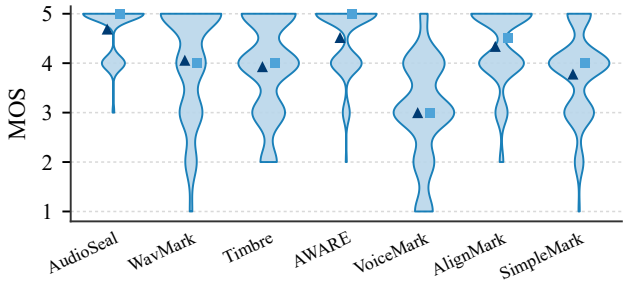


Figure 2: Mean opinion scores (MOS) for seven watermarking systems. ▲ and ■ indicate the mean and median, respectively.

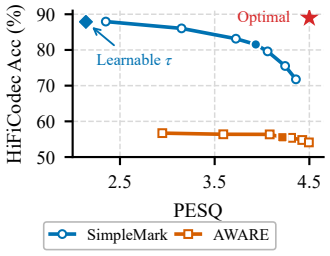


Figure 3: Robustness and imperceptibility trade-off as the scale bound varies from 1 to 11 dB.

tibility advantage. For low-level audio representations, we find that Qwen3-ECAPA performs a little better in robustness, whereas log-Mel slightly lags behind. However, imperceptibility ranks in the reverse order: log-Mel ranks first while QWen3-ECAPA ranks last. Nevertheless, results in the first line of Table 2 demonstrate that **with only a linear scale predictor and a linear head on log Mel spectrogram, we can already build an audio watermarking system with competitive robustness and imperceptibility**. Meanwhile, it should be noticed that compared with scale predictors, choices of audio representations have a larger effect on overall performance.

4.3 Ablation Studies

This section validates two key designs of SimpleMark: bounded magnitude scaling and watermark detection on low-level audio representations. Ablation studies on the ASP module in watermark detector can be found in Appendix C.2.

Fixed scale bounds control imperceptibility. SimpleMark controls imperceptibility by bounding scaling residuals within a hard range $(-\tau, \tau)$. With a learnable τ, robustness rises but PESQ falls from 3.934 to 2.138 (Table 4). In Figure 3, we show that one can balance robustness and imperceptibility by scanning the predefined threshold τ, or equivalently, $\overline{SDR}$ in decibel scale (see Equation (6)). Tightening the bound from 1 to 11 dB raises PESQ from 2.350 to 4.358 while reducing HiFiCodec bitwise accuracy from 87.95% to 71.77%. More importantly, the Pareto frontier of SimpleMark consistently lies above that of AWARE, indicating that SimpleMark achieves a significantly better robustness–imperceptibility trade-off.

Low-level audio representations facilitate robustness. As discussed in Section 3.2.2, an effective detector representation must preserve fine-grained acoustic variations through which a watermark can be written, while suppressing variations that are easily altered by attacks. The results in Table 3 support this view. Directly decoding from log-STFT spectrograms without Mel filtering substantially reduces robustness against neural codec resynthesis and voice cloning attacks. This finding suggests that certain level of frequency aggregation helps remove attack-sensitive details, thereby

Table 3: Ablation studies on audio representations for watermark detection. All systems use a linear scale predictor. Robustness columns report mean bitwise accuracy (%). WavLM-Lk denotes k-th layer hidden states from a frozen wavLM-base+ encoder. No Atk: No Attack. **Bold**: best.

	Robustness					Imperceptibility			
Representation	No Atk	Edits	Codecs	Cloning	AI	PESQ ↑	STOI ↑	ViSQOL ↑	NISQA ↑
Log-STFT	**99.99**	**98.80**	68.93	78.85	86.23	**4.119**	0.977	**4.399**	3.620
Log-Mel	99.70	96.76	80.24	89.19	85.52	3.905	0.966	4.143	3.688
EnCodec	93.50	84.47	81.60	84.21	82.67	4.079	0.968	3.968	**3.974**
SpeechTokenizer	99.77	96.25	84.75	**94.41**	82.57	4.041	0.966	4.024	3.896
WavLM-L0	99.21	87.33	84.39	86.34	79.65	3.601	0.976	4.100	3.902
WavLM-L3	91.34	68.77	75.57	71.41	71.87	3.780	0.977	4.301	3.802
WavLM-L12	68.05	56.69	63.34	61.23	59.27	3.665	**0.980**	4.292	3.772
ECAPA-TDNN	77.94	63.53	66.21	63.37	70.07	3.592	0.956	4.292	3.633
Qwen3-ECAPA	99.90	98.09	**84.99**	92.95	**87.83**	3.934	0.960	4.021	3.776

facilitating robustness against deep transformations. Furthermore, we find that a moderate level of learned abstraction also improves robustness. Indeed, EnCodec and SpeechTokenizer adopt the same model architecture and similar training objectives, whereas SpeechTokenizer is trained with an additional semantic-distillation loss. As shown in Table 3, latent representation of SpeechTokenizer exhibits substantially better robustness than that of EnCodec. However, mid or high-level abstraction hurts writability: when decoding watermarks from progressively deeper layers of a frozen WavLM-Base+ model, bitwise accuracy decreases consistently, both on attacked and unattacked audio. Similarly, standard ECAPA-TDNN trained for speaker verification is consistently less robust across all attack families than Qwen3-ECAPA trained jointly with a TTS backend. In summary, robust watermark detection requires low-level audio representations that can balance watermark writability and attack invariance.

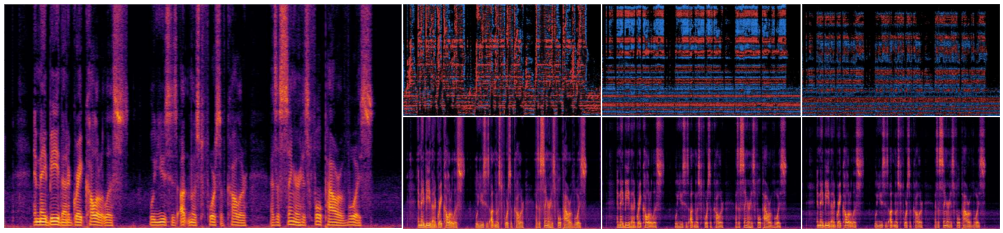
4.4 Analysis

In this section, we first re-evaluated the effectiveness of two widely-adopted designs, namely attack simulation and perceptual losses, under the simplified framework. Then we reveal the nature of audio watermarking in SimpleMark by visualizing the scaling residual maps.

Attack simulation brings no significant robustness gains. In the third line of Table 4, we investigate whether attack simulation during training can help improve robustness, especially against conventional audio edits. We follow San Roman et al. (2024) to insert a random distortion layer containing 14 types of audio edits into the trainning pipeline. Surprisingly, while attack simulation improves robustness against out-of-domain (OOD) neural codec resynthesis attacks by 0.91 pp, robustness against in-domain audio edits decreases. Therefore, we conclude that attack simulation has little influence on either robustness or imperceptibility and may be redundant under the proposed SimpleMark framework.

Perceptual losses improve imperceptibility at the cost of robustness. We investigate whether applying perceptual losses between watermarked and unwatermarked audio further improves imperceptibility of SimpleMark instances. Specifically, we follow Liu et al. (2024) to add $\mathcal{L}_{mel}$, an ℓ_1 loss with weight 1.0 on log Mel spectrogram. We also investigate effectiveness of $\mathcal{L}_{adv}$, a GAN-based adversarial loss with weight 0.01 on STFT magnitude spectrogram. Results in Table 4 indicate that while $\mathcal{L}_{mel}$ improves imperceptibility, it significantly damages robustness, especially against audio edits and voice cloning. Meanwhile, adding $\mathcal{L}_{adv}$ on top of $\mathcal{L}_{mel}$ hardly influences robustness or imperceptibility. Additional results in Appendix C.3 further reveal that $\mathcal{L}_{mel}$ not only reduces robustness but also delays convergence. Therefore, SimpleMark avoids perceptual losses and adopts magnitude scaling bounds to control imperceptibility instead.

Watermark emerges as a time-invariant, content-agnostic frequency pattern. To understand how SimpleMark embeds watermarks, we visualize watermarked STFT magnitudes and scaling



(a) Unwatermarked (b) Timbre (c) Lin+ECAPA (d) CF+ST

Figure 4: Visualization of watermark patterns. We show scaling residuals (top) and watermarked STFT magnitudes (bottom) of: (b) Timbre baseline, (c) SimpleMark with Linear+Qwen3-ECAPA, and (d) SimpleMark with Conformer+SpeechTokenizer. Red and blue indicate positive and negative residuals, respectively. Zoom in for better visibility.

Table 4: Analysis of attack simulation and perceptual losses. Content-Agnostic denotes a system with its scale predictor conditioned only on watermark message. No Atk: No Attack. **Bold**: best.

	Robustness					Imperceptibility			
Variant	No Atk	Edits	Codecs	Cloning	AI	PESQ ↑	STOI ↑	ViSQOL ↑	NISQA ↑
SimpleMark	99.90	98.09	84.99	92.95	87.83	3.934	0.960	4.021	3.776
- Fixed τ	**100.0**	**99.84**	**89.50**	**97.28**	**89.80**	2.138	0.833	3.197	1.696
+ Simulated Attacks	99.78	97.32	85.90	92.80	87.86	3.927	0.962	4.048	3.830
+ $\mathcal{L}_{mel}$	97.96	89.65	77.62	83.21	83.66	4.203	0.979	**4.558**	3.912
+ $\mathcal{L}_{mel}$ + $\mathcal{L}_{adv}$	97.94	89.58	77.59	83.20	83.52	4.232	0.979	4.555	3.909
Additive Embedder	99.79	97.93	84.77	93.02	86.91	3.868	0.962	4.005	3.783
+ $\mathcal{L}_{mel}$	99.30	94.30	79.48	88.11	84.62	4.108	**0.986**	4.368	3.824
+ $\mathcal{L}_{mel}$ + $\mathcal{L}_{adv}$	99.34	94.58	79.53	88.42	84.78	4.138	**0.986**	4.358	3.825
Content-Agnostic	99.46	96.44	83.97	91.15	86.78	**4.267**	0.972	4.141	**3.925**

residuals from different watermarking systems. As illustrated in panels (c) and (d) of Figure 4, SimpleMark embeds each watermark as a content-agnostic frequency pattern shared across time frames. Panel (b) shows that similar tonic artifacts can be found in watermarked audio from Timbre, suggesting that this behavior may not be unique to SimpleMark. We further confirm this discovery with a content-agnostic scale predictor, where $\mathcal{S}$ takes the registered message $\mathbf{m}$ as the only input. Results in Table 4 show that this predictor achieves competitive robustness with even better imperceptibility. These results suggest that in SimpleMark, the watermark may simply be encoded into a fixed frequency pattern shared across all frames. The perturbation varies over time only because this pattern is multiplied by the host magnitude. See Appendix C.5 for more visualization results.

5 Conclusion

This paper proposes SimpleMark, a simple yet effective robust audio watermarking framework. SimpleMark highlights two design choices: watermark embedding based on bounded magnitude scaling for imperceptibility, and watermark detection in low-level audio representation for robustness. With different scale predictors and audio representations, SimpleMark consistently achieves versatile robustness and competitive perceptual quality. Furthermore, we re-evaluated the effectiveness of attack simulation and perceptual losses, and find them redundant or even harmful under the SimpleMark framework. SimpleMark demonstrates that robust audio watermarking does not necessitate complex designs. We hope that this work could inspire simpler and more interpretable alternatives to existing audio watermarking systems built upon complex architectures, competing objectives, and sophisticated training strategies.

AI USE STATEMENT

In this work, we used generative AI tools for coding and polishing writing. We have not used generative AI tools for retrieval and discovery, research ideation or execution, or generating synthetic datasets. Proving mathematical claims are not applicable to this work. We have reviewed all AI-assisted work. We checked LLM-refined texts, and LLM-generated code was verified and tested for correctness. 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 paper contains a MOS evalation of watermarked audio quality. The MOS study involved 10 student participants, who received no compensation. Each participant rated 10 sets of watermarked audio samples from 7 anonymized systems on a 1–5 scale in a blind listening test. The test was reference-based: listeners had access to the original unwatermarked recording. The watermarked samples were randomly shuffled before being presented to the listeners.

REPRODUCIBILITY STATEMENT

To enhance reproducibility, we plan to open source our codes. The codes to be open-sourced can be found at: `https://anonymous.4open.science/r/simplemark-anonymous-release-B0C3`. Meanwhile, the website for conducting MOS test, together with audio samples, are included in supplementary materials.

REFERENCES

Antoine Caillon and Philippe Esling. Rave: A variational autoencoder for fast and high-quality neural audio synthesis. *arXiv preprint arXiv:2111.05011*, 2021.

Guangyu Chen, Yu Wu, Shujie Liu, Tao Liu, Xiaoyong Du, and Furu Wei. WavMark: Watermarking for audio generation, 2023.

Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, Jian Wu, Long Zhou, Shuo Ren, Yanmin Qian, Yao Qian, Jian Wu, Michael Zeng, Xiangzhan Yu, and Furu Wei. Wavlm: Large-scale self-supervised pre-training for full stack speech processing. *IEEE Journal of Selected Topics in Signal Processing*, 16(6):1505–1518, 2022. doi: 10.1109/JSTSP.2022.3188113.

Xuanjun Chen, Jiawei Du, Haibin Wu, Lin Zhang, I-Ming Lin, I-Hsiang Chiu, Wenze Ren, Yuan Tseng, Yu Tsao, Jyh-Shing Roger Jang, et al. Codecfake+: Codec-based resynthesized data as a proxy for detecting codecfake speech. *IEEE Transactions on Audio, Speech and Language Processing*, 2026a.

Yen-Shan Chen, Shih-Yu Lai, Ying-Jung Tsou, Yi-Cheng Lin, Bing-Yu Chen, Yun-Nung Chen, Hung-yi Lee, and Shang-Tse Chen. Latent-mark: An audio watermark robust to neural codec compression. *arXiv preprint arXiv:2603.05310*, 2026b.

Yushen Chen, Zhikang Niu, Ziyang Ma, Keqi Deng, Chunhui Wang, Jian Zhao, Kai Yu, and Xie Chen. F5-TTS: A fairytaler that fakes fluent and faithful speech with flow matching, 2024.

Michael Chinen, Felicia SC Lim, Jan Skoglund, Nikita Gureev, Feargus O'Gorman, and Andrew Hines. Visqol v3: An open source production ready objective speech and audio metric. In *2020 twelfth international conference on quality of multimedia experience (QoMEX)*, pp. 1–6. IEEE, 2020.

Brecht Desplanques, Jenthe Thienpondt, and Kris Demuynck. ECAPA-TDNN: Emphasized Channel Attention, Propagation and Aggregation in TDNN Based Speaker Verification. In *Interspeech 2020*, pp. 3830–3834, 2020. doi: 10.21437/Interspeech.2020-2650.

Heinrich Dinkel, Zhiyong Yan, Yongqing Wang, Junbo Zhang, Yujun Wang, and Bin Wang. Scaling up masked audio encoder learning for general audio classification. In *Interspeech 2024*, pp. 547–551, 2024. doi: 10.21437/Interspeech.2024-246.

Zhihao Du, Yuxuan Wang, Qian Chen, Xian Shi, Xiang Lv, Tianyu Zhao, Zhifu Gao, Yexin Yang, Changfeng Gao, Hui Wang, et al. CosyVoice 2: Scalable streaming speech synthesis with large language models, 2024.

Zhihao Du, Changfeng Gao, Yuxuan Wang, Fan Yu, Tianyu Zhao, Hao Wang, Xiang Lv, Hui Wang, Chongjia Ni, Xian Shi, Keyu An, et al. CosyVoice 3: Towards in-the-wild speech generation via scaling-up and post-training. *arXiv preprint arXiv:2505.17589*, 2025.

Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, and Ruoming Pang. Conformer: Convolution-augmented Transformer for Speech Recognition. In *Interspeech 2020*, pp. 5036–5040, 2020. doi: 10.21437/Interspeech.2020-3015.

Haorui He, Zengqiang Shang, Chaoren Wang, Xuyuan Li, Yicheng Gu, Hua Hua, Liwei Liu, Chen Yang, Jiaqi Li, Peiyang Shi, Yuancheng Wang, Kai Chen, Pengyuan Zhang, and Zhizheng Wu. Emilia: An extensive, multilingual, and diverse speech dataset for large-scale speech generation. In *Proc. of SLT*, 2024.

Hangrui Hu, Xinfa Zhu, Ting He, Dake Guo, Bin Zhang, Xiong Wang, Zhifang Guo, Ziyue Jiang, Hongkun Hao, Zishan Guo, Xinyu Zhang, Pei Zhang, Baosong Yang, Jin Xu, Jingren Zhou, and Junyang Lin. Qwen3-tts technical report. *arXiv preprint arXiv:2601.15621*, 2026a.

Zi Hu, Houmin Sun, Linxi Li, Yechen Wang, Liwei Jin, Carsten Maple, and Ming Li. Investigating codec-internal latent audio watermarking for neural codec robustness. *arXiv preprint arXiv:2607.21132*, 2026b.

Shengpeng Ji, Ziyue Jiang, Jialong Zuo, Minghui Fang, Yifu Chen, Tao Jin, and Zhou Zhao. Speech watermarking with discrete intermediate representations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 24239–24247, 2025. doi: 10.1609/aaai.v39i23.34600.

Lauri Juvela and Xin Wang. Audio codec augmentation for robust collaborative watermarking of speech synthesis. In *ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5, 2025. doi: 10.1109/ICASSP49660.2025.10888976.

Slavko Kovačević, Elena Nešović, Kosta Pavlović, Petar Nedić, and Igor Djurović. Deepmark benchmark: Redefining audio watermarking robustness. *IEEE Access*, 14:62031–62044, 2026. doi: 10.1109/ACCESS.2026.3685903.

Haiyun Li, Zhiyong Wu, Xiaofeng Xie, Jingran Xie, Yaoxun Xu, and Hanyang Peng. VoiceMark: Zero-shot voice cloning-resistant watermarking approach leveraging speaker-specific latents. In *Interspeech 2025*, pp. 5108–5112, 2025. doi: 10.21437/Interspeech.2025-575.

Haiyun Li, Shuhai Peng, Zhisheng Zhang, Jingran Xie, Xiaofeng Xie, Hanyang Peng, and Zhiyong Wu. Feature-aligned speech watermarking for robustness to reconstruction distortions, 2026a. Accepted by ICME 2026.

Kexin Li, Xiao Hu, Ilya Grishchenko, and David Lie. LambdaMark: Semantic audio watermarking for robustness and radioactivity, 2026b.

Shijia Liao, Yuxuan Wang, Tianyu Li, Yifan Cheng, Ruoyi Zhang, Rongzhi Zhou, and Yijin Xing. Fish-Speech: Leveraging large language models for advanced multilingual text-to-speech synthesis, 2024.

Chang Liu, Jie Zhang, Tianwei Zhang, Xi Yang, Weiming Zhang, and Nenghai Yu. Detecting voice cloning attacks via timbre watermarking. In *31st Annual Network and Distributed System Security Symposium*, 2024. doi: 10.14722/ndss.2024.24200.

Weizhi Liu, Yue Li, and Zhaoxia Yin. VocBulwark: Towards practical generative speech watermarking via additional-parameter injection, 2026.

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. URL https://openreview.net/forum?id=h4bSJMaNgb.

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. doi: 10.21437/Interspeech.2021-299.

Nicolas M. Müller, Nicholas Evans, Hemlata Tak, Philip Sperl, and Konstantin Böttinger. Harder or Different? Understanding Generalization of Audio Deepfake Detection. In *Interspeech 2024*, pp. 2705–2709, 2024. doi: 10.21437/Interspeech.2024-247.

Koji Okabe, Takafumi Koshinaka, and Koichi Shinoda. Attentive Statistics Pooling for Deep Speaker Embedding. In *Interspeech 2018*, pp. 2252–2256, 2018. doi: 10.21437/Interspeech.2018-993.

Patrick O'Reilly, Zeyu Jin, Jiaqi Su, and Bryan Pardo. MaskMark: Robust neural watermarking for real and synthetic speech. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 4650–4654, 2024. doi: 10.1109/ICASSP48485.2024.10447253.

Patrick O'Reilly, Zeyu Jin, Jiaqi Su, and Bryan Pardo. Deep audio watermarks are shallow: Limitations of post-hoc watermarking techniques for speech. In *ICLR 2025 Workshop on GenAI Watermarking*, 2025.

Yigitcan Özer, Woosung Choi, Joan Serrà, Mayank Kumar Singh, Wei-Hsiang Liao, and Yuki Mitsufuji. A comprehensive real-world assessment of audio watermarking algorithms: Will they survive neural codecs? In *Interspeech 2025*, pp. 5113–5117, 2025. doi: 10.21437/Interspeech.2025-1530.

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, 2015. doi: 10.1109/ICASSP.2015.7178964.

Kosta Pavlović, Lazar Stanarević, Petar Nedić, Elena Nešović, Slavko Kovačević, and Igor Djurović. AWARE: Audio watermarking with adversarial resistance to edits, 2025.

Antony W. Rix, John G. Beerends, Michael P. Hollier, and Andries P. Hekstra. Perceptual evaluation of speech quality (PESQ): A new method for speech quality assessment of telephone networks and codecs. In *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 2, pp. 749–752, 2001. doi: 10.1109/ICASSP.2001.941023.

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, 2024. URL `https://proceedings.mlr.press/v235/san-roman24a.html`.

Cees H. Taal, Richard C. Hendriks, Richard Heusdens, and Jesper Jensen. An algorithm for intelligibility prediction of time–frequency weighted noisy speech. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(7):2125–2136, 2011. doi: 10.1109/TASL.2011.2114881.

Xinsheng Wang, Mingqi Jiang, Ziyang Ma, Ziyu Zhang, Songxiang Liu, Linqin Li, Zheng Liang, Qixi Zheng, Rui Wang, Xiaoqin Feng, et al. Spark-TTS: An efficient llm-based text-to-speech model with single-stream decoupled speech tokens, 2025.

Yuancheng Wang, Haoyue Zhan, Liwei Liu, Ruihong Zeng, Haotian Guo, Jiachen Zheng, Qiang Zhang, Xueyao Zhang, Shunsi Zhang, and Zhizheng Wu. MaskGCT: Zero-shot text-to-speech with masked generative codec transformer, 2024.

Chia-Hua Wu, Wanying Ge, Xin Wang, Junichi Yamagishi, Yu Tsao, and Hsin-Min Wang. A Comparative Study on Proactive and Passive Detection of Deepfake Speech. In *Interspeech 2025*, pp. 5328–5332, 2025. doi: 10.21437/Interspeech.2025-1419.

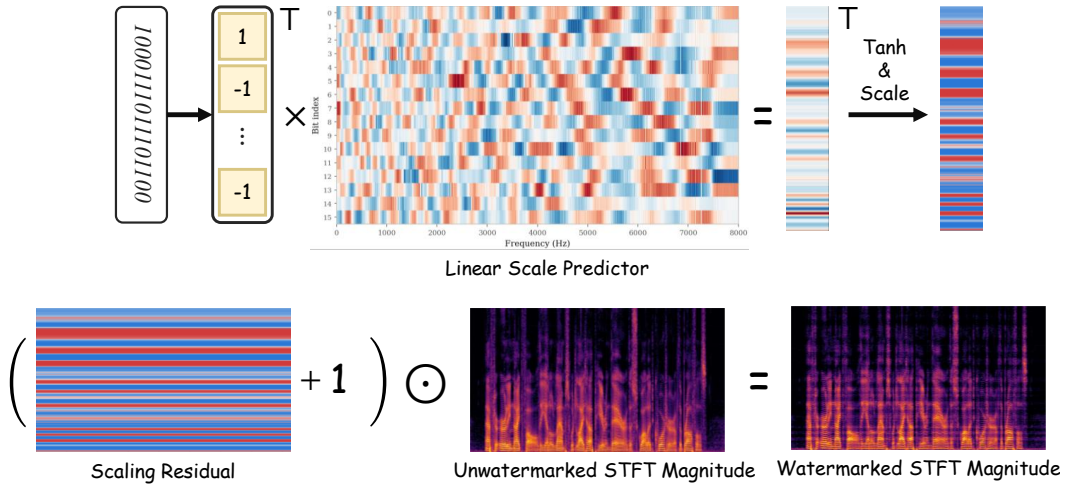
Detai Xin, Shujie Hu, Chengzuo Yang, Chen Huang, Guoqiao Yu, Guanglu Wan, and Xunliang Cai. LongCat-AudioDiT: High-fidelity diffusion text-to-speech in the waveform latent space, 2026.

Zheqi Yuan, Yucheng Huang, Guangzhi Sun, Zengrui Jin, and Chao Zhang. Multiplexing neural audio watermarks. *arXiv preprint arXiv:2511.02278*, 2025.

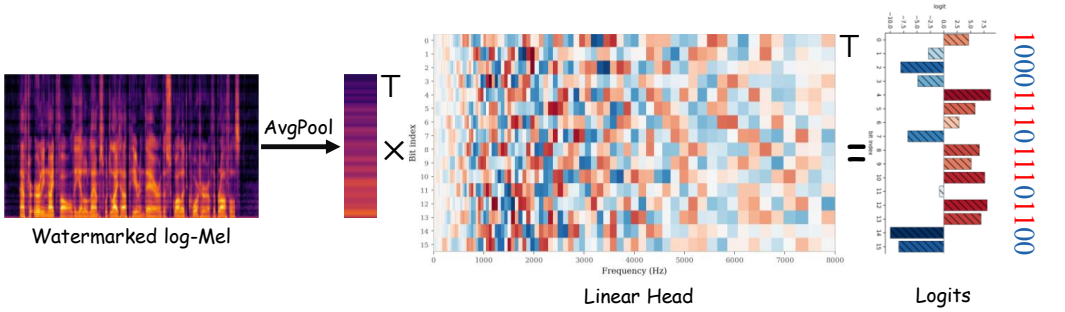
Jun Zhan, Junqi Dai, Jiasheng Ye, Yunhua Zhou, Dong Zhang, Zhigeng Liu, Xin Zhang, Ruibin Yuan, Ge Zhang, Linyang Li, et al. Anygpt: Unified multimodal llm with discrete sequence modeling. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9637–9662, 2024.

Xin Zhang, Dong Zhang, Shimin Li, Yaqian Zhou, and Xipeng Qiu. Speechtokenizer: Unified speech tokenizer for speech language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=AF9Q8Vip84.

A A MINIMAL EXAMPLE



(a) Watermark Embedder



(b) Watermark Detector

Figure 5: Watermark embedding and detection pipeline of the minimal example.

Here we illustrate the minimal example mentioned in Section 1. This example consists of only a content-agnostic linear scale predictor $\mathcal{S}(\mathbf{m}) : \{0,1\}^K \to \mathbb{R}^F$, an average pooling layer and a linear detector head on Log-Mel spectrogram. It only contains two trainable linear layers: the linear scale predictor that maps K-bit watermark message to F frequency bins, and the linear detector head that maps 80-dimensional Log-Mel features back to K-bit logits. Figure 5 illustrates the watermark embedding and detection pipeline with a real example.

Embedding. In the watermark embedder, the bipolar registered message $\bar{\mathbf{m}} = 2\mathbf{m} - 1$ is first multiplied with weights of the linear scale predictor to form the raw scaling residuals $\mathcal{S}(\mathbf{m})$. Then, the raw scaling residuals are normalized into $(-\tau, \tau)$, broadcasted along the time axis, and added with $\mathbf{1}$, forming the scaling factor matrix $\mathbf{S} \in \mathbb{R}^{F \times N}$. Taking the element-wise production of the scaling factor matrix $\mathbf{S}$ and the original STFT magnitude spectrogram $\mathbf{M}$ yields the watermarked STFT magnitude $\widetilde{\mathbf{M}}$. Finally, this watermarked STFT magnitude is combined with the original phase and mapped back to the waveform domain via inverse STFT.

Detection. For each test sample, the detector first takes the 80-bin Log-Mel spectrogram as the low-level audio representation and averages all frames to form a global vector. Then the global vector is multipled with weights of a linear head, yielding logits for each watermark bit. Positive logits indicate watermark bit "1", whereas negative logits indicate "0".

Note that this minimal example contains **no neural networks** — two linear layers and several learning-free operators are all it needs. Though simple, this minimal example achieves robustness

Table 5: Comparison between the minimal example and main SimpleMark systems. The minimal example differs from the "Linear+Log-Mel" system in that it utilizes a content-agnostic scale predictor and an average pooling layer instead of the default ASP layer in watermark detector.

		Robustness				Imperceptibility			
System	#Params.	Edits	Codecs	Cloning	AI	PESQ↑	STOI↑	ViSQOL↑	NISQA↑
Linear+QWen3-ECAPA	12.6M	**98.09**	**84.99**	**92.95**	**87.83**	3.934	0.960	4.021	3.776
Linear+Log-Mel	0.55M	96.76	80.24	89.19	85.52	3.905	0.966	4.143	3.688
Minimal Example	0.01M	95.80	79.12	88.98	84.84	**4.294**	**0.977**	**4.270**	**3.834**

comparable to our main SimpleMark systems with even better imperceptibility. More importantly, we find this minimal example highly explainable: Weights of the linear scale predictor and the linear detector head show clear and consistent frequency pattern of each watermark bit, coinciding with our discovery in Section 4.4 that the nature of SimpleMark watermarks are time-invariant, content-agnostic frequency patterns.

B Implementation Details

B.1 Hyperparameters

To facilitate reproduction, we list SimpleMark hyperparameters in Tables 6 and 7.

Table 6: Training hyperparameters.

Hyperparameter	Value
Sampling Rate	16 kHz
Maximum Input Length	10 s
Scale Bound (τ)	0.501
Scale Bound ($\overline{SDR}$)	6 dB
Target Bit Margin γ	2.0
Learning Rate	3×10^{-4}
Epochs	50
Batch Size	8
GPU Nodes	4
Optimizer	AdamW
Weight Decay	10^{-3}
Seed	0

Table 7: Audio processing hyperparameters.

Hyperparameter	Value
STFT FFT Size	1024
STFT Hop Length	256
Log-Mel FFT Size	400
Log-Mel Hop Length	160
Number of Mel Filters	80

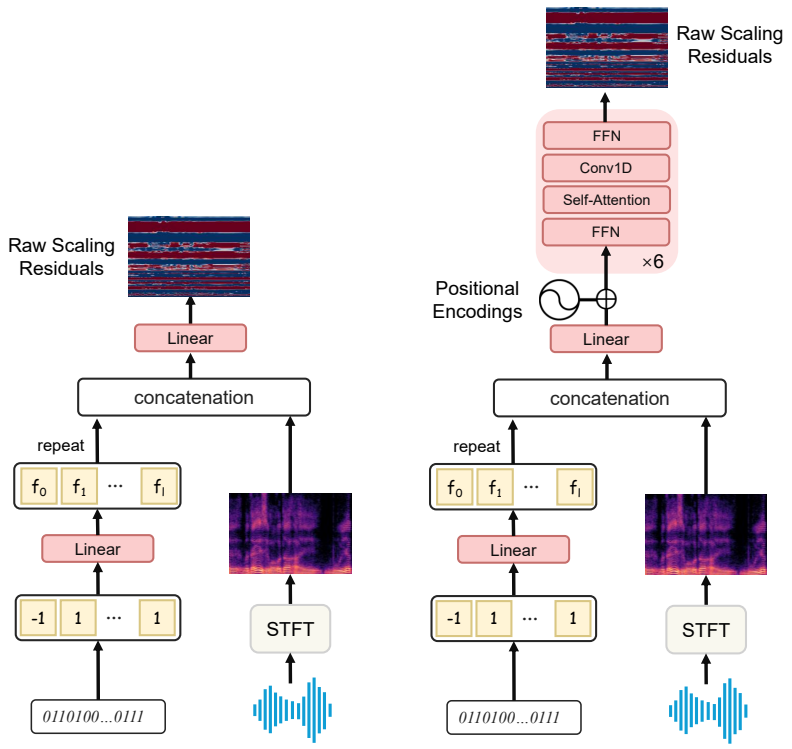
B.2 Architectural Details

This section provides additional details on model architectures of the scale predictors and audio representation extractors introduced in Section 3.2.

B.2.1 Scale Predictors

Linear. The linear scale predictor comprises two linear layers. The first linear layer projects the K-bit bipolar message $\bar{\mathbf{m}} = 2\mathbf{m} - 1$ to the STFT frequency dimension F. The F-dimensional vector is broadcast along the time axis and concatenated with the input STFT magnitude spectrogram along the frequency axis. Finally, the second linear layer project the $2F$-dimensional features back to 513 channels to form the raw scaling residuals. Note that no activation layer is used.

Conformer. The conformer-based scale predicted is constructed by stacking 6 pre-norm conformer blocks on top of the linear counterpart: After concatenation, the linear layer project the feature sequence to $d_{CF} = 256$ dimensions. Before entering the conformer layers, the feature sequence is augmented with additive sinusoidal positional encoding. As illustrated in Figure 6 (b),



(a) Linear (b) Conformer

Figure 6: Structure of (a) linear and (b) conformer scale predictors.

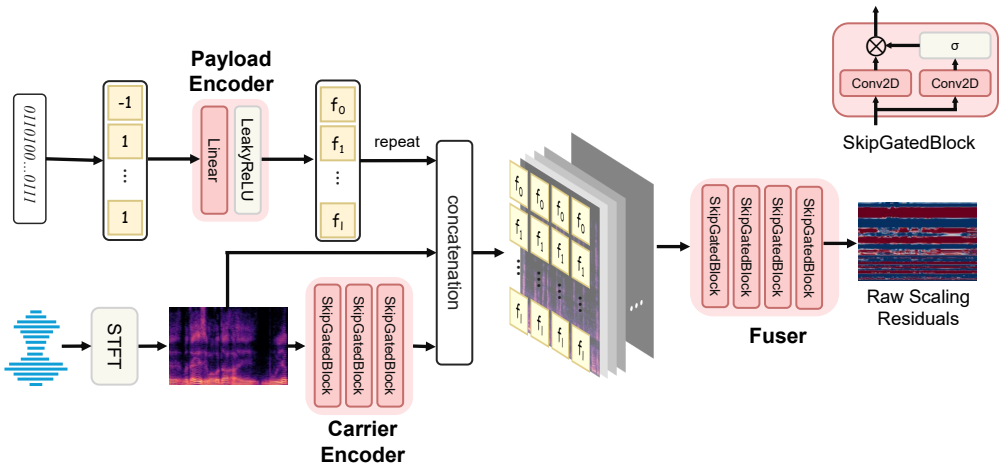


Figure 7: Structure of the gated-convolutional scale predictor.

each conformer block contains two feed forward layers with a latent dimension of 1024, a self-attention layer with 4 heads, and a 1D convolutional layer with kernel size 5. All these modules are zero-initialized so that learning starts from the bottom linear layers. Finally, a linear head maps the 256-dimensional features back to STFT frequency dimension F, producing raw scaling residuals.

Gated-Convolutional Network. The gated-convolutional scale predictor processes 2D feature maps with 2D gated convolution blocks. It contains three modules: a payload encoder that encodes K-bit bipolar message $\bar{\mathbf{m}} = 2\mathbf{m} - 1$ into a global vector, a carrier encoder that encodes STFT

magnitude spectrogram into a 64-channel feature map, and a fuser that fuses the global vector and the feature map to produce raw scaling residuals. While the payload encoder contains only a linear layer and an activation layer, both the carrier and the fuser are composed of skip gated blocks, whose architecture is shown at the top-right corner of Figure 7.

B.2.2 AUDIO REPRESENTATION EXTRACTORS

QWen3-ECAPA. QWen3-ECAPA refers to the ECAPA-TDNN-based speaker encoder jointly trained with Qwen3-TTS. It encodes a 128-dimensional log-Mel spectrogram into a 2048-dimensional global embedding. Since QWen3-ECAPA already contains an attentive statistic pooling (ASP) layer, we build the watermark detector by directly adding a linear head on top of the pretrained ECAPA-TDNN model [1].

SpeechTokenizer. This extractor extracts the latent features of SpeechTokenizer with the pre-trained codec encoder [2]. Given a 16kHz waveform, the encoder produces a sequence of continuous features at 50 frames per second, with 1024 dimensions per frame. When building the watermark detector, a learnable Attentive Statistics Pooling (ASP) layer first aggregates these features over time, followed by a linear readout head that predicts the watermark logits. Note that we bypass the residual vector quantization (RVQ) module and decode the watermark directly from the encoder's continuous latent features.

C ADDITIONAL RESULTS

C.1 EXISTENCE ACCURACY

In Section 4, we report *bitwise accuracy* (BA), which measures the recovery rate of individual bits. While this metric well reflects robustness of an audio watermark system, it does not directly indicate proactive deepfake detection performance. In this section, we additionally report *existence accuracy* (EA), which measures whether the detector correctly classifies watermarked/unwatermarked audio. A sample is classified as "watermarked" if at least 75% of the decoded watermark message matches the registered one. Additionally, as existence accuracy approaches 100% under conventional audio edits, RAW-Bench instead reports *full-message accuracy* (FMA), which only counts the ratio of samples where every message bit is correctly recovered. Results in Table 8 show distinct advantage of SimpleMark instances over other systems. Indeed, the "Conformer + QWen3-ECAPA" system achieves superior existence accuracy on all three families of deep attacks, supporting SimpleMark as a state-of-the-art solution for proactive deepfake detection via audio watermarking.

C.2 ADDITIONAL ABLATION STUDIES ON ATTENTIVE STATISTICS POOLING

To validate the effectiveness of the Attentive Statistics Pooling (ASP) module in watermark detector, we replace ASP with parameter-free mean pooling. Results in Table 9 show significant decreases in robustness against all four types of attacks, accompanied by a slight improvement in imperceptibility. We suspect that replacing ASP with mean pooling dampens the detector's ability to identify and mask out silent segments, thereby reducing robustness.

Table 9: Ablation studies on attentive statistics pooling (ASP). Bitwise accuracy (%) is reported to evaluate robustness.

	Robustness				Imperceptibility			
System	Edits	Codecs	Cloning	AI	PESQ↑	STOI↑	ViSQOL↑	NISQA↑
SimpleMark	**98.09**	**84.99**	**92.95**	**87.83**	3.934	0.960	4.021	3.776
-ASP	95.58	80.42	89.60	84.56	**4.021**	**0.963**	**4.120**	**3.833**

[1] https://huggingface.co/Qwen/Qwen3-TTS-12Hz-1.7B-Base

[2] https://huggingface.co/fnlp/AnyGPT-speech-modules/tree/main/speechtokenizer

Table 8: Accuracy (%) for binary watermark presence classification. We report existence accuracy (EA) with threshold 75% in all columns except audio edits, where harder full-message accuracy (FMA) is reported instead. **Bold**: best.

System	Clean (EA)	Edits (FMA)	Codecs (EA)	Cloning (EA)	AI (EA)
AudioSeal	99.58	73.49	52.52	50.14	58.98
WavMark	**99.62**	88.23	51.77	50.23	51.65
Timbre†	97.16	95.16	58.40	66.68	84.73
AWARE	98.19	89.06	62.11	50.06	86.36
VoiceMark	98.30	41.42	84.70	94.10	83.43
AlignMark	95.73	44.44	86.68	69.26	86.72
SimpleMark	98.05	79.26	87.23	96.81	87.00
Linear + Log-Mel	97.84	71.02	83.35	94.33	86.05
Linear + SpeechTokenizer	98.26	63.17	86.99	96.89	84.16
Gated conv. + Log-Mel	98.11	70.37	83.32	94.40	86.13
Gated conv. + Qwen3-ECAPA	98.38	73.03	87.68	96.62	86.70
Gated conv. + SpeechTokenizer	97.77	61.98	86.23	97.03	81.85
Conformer + Log-Mel	97.92	76.49	85.45	94.53	86.00
Conformer + Qwen3-ECAPA	98.32	75.07	**89.14**	97.15	**87.34**
Conformer + SpeechTokenizer	97.71	64.15	87.31	96.44	84.77
Log-STFT	99.31	88.98	64.58	81.13	86.59
EnCodec	97.42	9.79	86.61	91.17	86.03
WavLM-L0	98.45	19.11	88.21	92.59	79.60
WavLM-L3	97.19	1.12	79.86	73.54	73.38
WavLM-L12	66.37	0.01	60.29	58.02	57.30
ECAPA-TDNN	83.89	0.57	66.31	61.71	72.39
- Fixed τ	97.67	**97.76**	71.53	**97.55**	82.86
+ Simulated Attacks	98.11	72.72	88.72	96.69	87.25
+ $\mathcal{L}_{\text{mel}}$	97.92	30.30	81.20	89.50	85.68
+ $\mathcal{L}_{\text{mel}} + \mathcal{L}_{\text{adv}}$	97.90	30.18	81.22	89.28	85.62
Additive Embedder	98.19	77.56	86.86	96.12	86.62
Additive + $\mathcal{L}_{\text{mel}}$	97.77	51.48	81.95	93.66	85.82
Additive + $\mathcal{L}_{\text{mel}} + \mathcal{L}_{\text{adv}}$	98.15	53.75	82.01	93.95	85.97
Content-Agnostic	99.56	63.93	84.16	96.29	86.94
Minimal Example	**99.62**	63.48	78.43	94.71	85.86
- ASP	98.28	60.06	83.04	94.84	85.67

C.3 Effect of Perceptual Losses during Training

In Section 4.4, we point out that applying perceptual losses may bias the audio watermarking model toward better imperceptibility at the expense of robustness. To further illustrate this, we track HiFi-Codec bitwise accuracy and PESQ when training with different objectives. As shown in Figure 8 (a), adding $\mathcal{L}_{mel}$ not only reduces robustness but also delays convergence. This phenomenon becomes more pronounced when watermark is embedded via commonly adopted additive perturbation:

$$\widetilde{\mathbf{M}}_{add} = \mathbf{M} + \mathcal{S}(\mathbf{m}, \mathbf{M}) \tag{9}$$

rather than multiplicative scaling adopted in SimpleMark:

$$\widetilde{\mathbf{M}} = \mathbf{M} \odot [1 + \tau \tanh(\mathcal{S}(\mathbf{m}, \mathbf{M}))]. \tag{10}$$

As shown in Figure 8 (b), models trained with perceptual losses reach their highest robustness at approximately epoch 10. After that, their bitwise accuracy steadily decreases without significant improvement in imperceptibility. These results indicate that jointly optimizing watermark recovery loss and perceptual losses introduces an unstable trade-off between robustness and imperceptibility, making the final performance sensitive to training strategies.

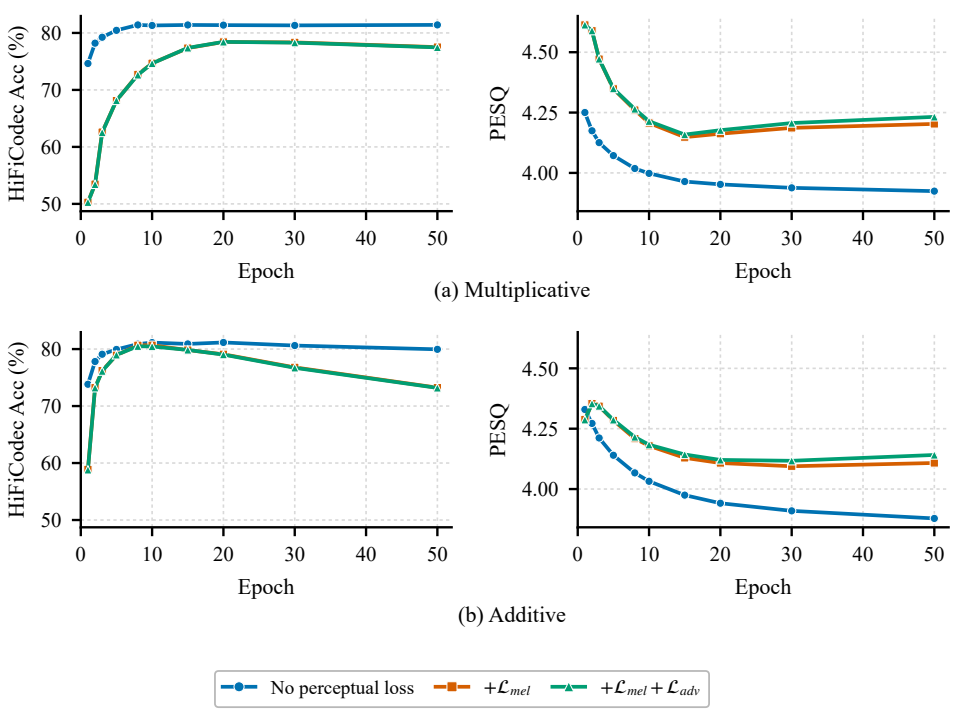


Figure 8: Evolution of robustness and imperceptibility over epochs when training with and without perceptual losses. The top and bottom rows use multiplicative scaling and additive perturbation for watermark embedding, respectively.

C.4 Performance under Higher Capacity Settings

In this section, we evaluate robustness and imperceptibility of SimpleMark as message length varies from 16 to 512 bits. Results in Figure 9 show that across all message bits, our proposed method consistently surpasses Timbre and AWARE in robustness while achieving competitive imperceiptibility.

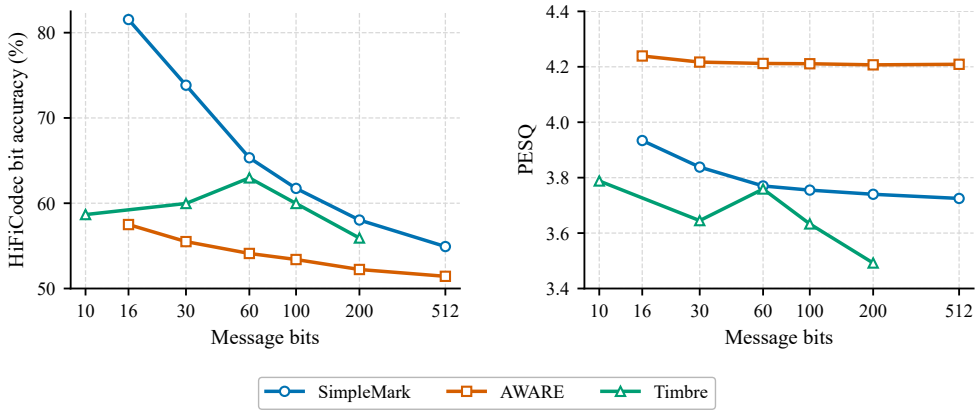


Figure 9: HiFiCodec bitwise accuracy and PESQ in high capacity region.

C.5 Additional Spectral Visualizations

As discussed in Section 4.4, SimpleMark tends to encode watermarks as time-invariant and content-agnostic frequency patterns. This section provides additional evidence for this observation. Unless otherwise stated, all visualizations use the same host audio and registered message as in Figure 4.

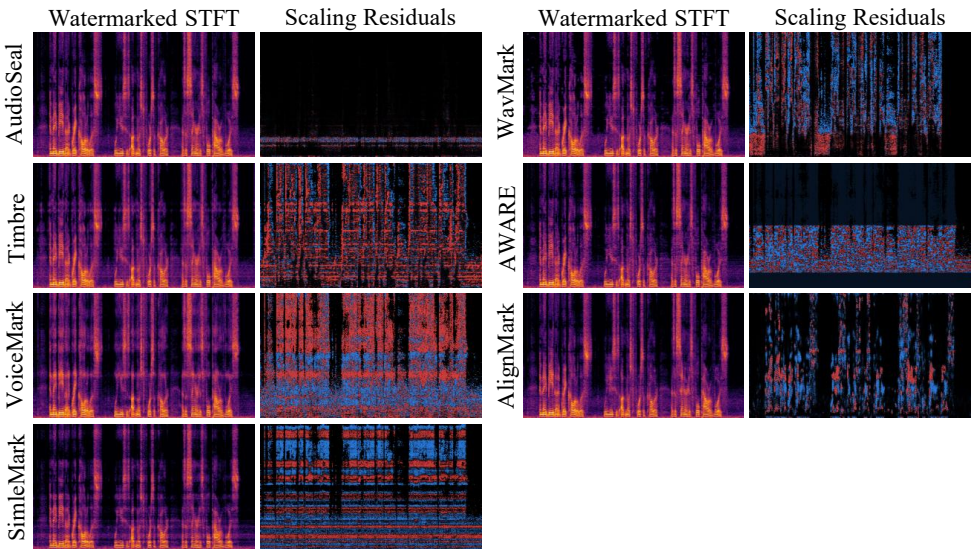


Figure 10: Watermarked STFT magnitudes and empirical scaling residuals for six baselines and the default Linear+Qwen3-ECAPA SimpleMark system. See Figure 4 for unwatermarked reference.

Apart from watermarked STFT magnitude, we additionally visualize two types of scaling residuals:

1. **Applied Scaling Residuals** $[\tau \tanh(\mathcal{S}(\mathbf{m}, \mathbf{M}))]$: the bounded residual maps explicitly predicted by the scale predictor.
2. **Empirical Scaling Residuals** $[\widetilde{\mathbf{M}} \oslash \mathbf{M} - \mathbf{1}]$: the residual maps measured from original STFT magnitude $\mathbf{M}$ and watermarked STFT magnitude $\widetilde{\mathbf{M}}$. $\oslash$ denotes element-wise division.

For baseline methods, we visualize only empirical scaling residuals because most of them do not explicitly predict scaling maps. For SimpleMark instances, we visualize both applied and empirical scaling residuals.

We first compare empirical residuals across baselines and SimpleMark to examine whether they exhibit similar frequency patterns. We then visualize watermarked STFT spectrograms and scaling residuals from the nine SimpleMark configurations listed in Table 2, assessing whether similar frequency patterns persists across predictor architectures and detector representations. Finally, we embed the same watermark message to different host audio samples using a fixed SimpleMark instance to examine whether the learned frequency pattern is content agnostic.

Baselines. Figure 10 compares watermarks generated by baseline systems with those generated by SimpleMark. Among the six baselines, Timbre and VoiceMark exhibit stripe-like residual patterns, suggesting that they may also encode watermark information through relatively time-invariant frequency structure. However, these patterns are less visually distinct than those produced by SimpleMark, which may account for their inferior robustness against watermark removal attacks. The remaining baselines, namely AudioSeal, WavMark, AWARE, and AlignMark, do not exhibit similar stripe-like patterns in their empirical scaling residuals. This diversity indicates that existing systems rely on different embedding mechanisms. Nevertheless, we empirically find that methods without a clearly structured residual pattern generally show weaker robustness to at least one family of watermark removal attacks.

Comparison among Different SimpleMark Instances. Figures 11 to 13 visualize watermarks produced by all combinations of three scale predictors and three audio representations. Across these SimpleMark instances, the scaling residuals consistently exhibit clear stripe-like frequency patterns,

suggesting that these patterns do not emerge from a particular scale-predictor architecture or detector representation.

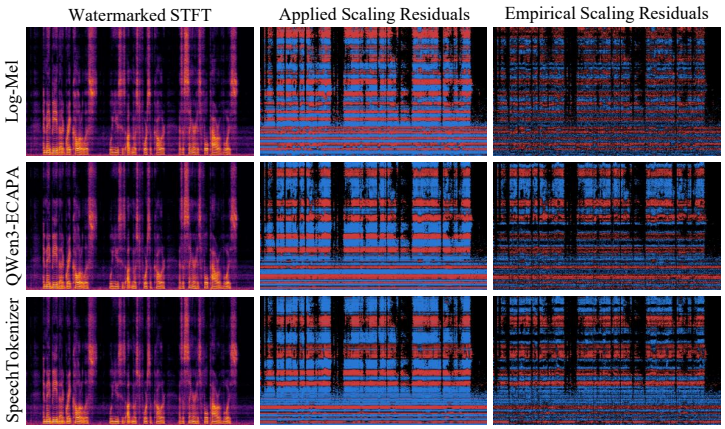


Figure 11: Linear scale predictor paired with three detector representations.

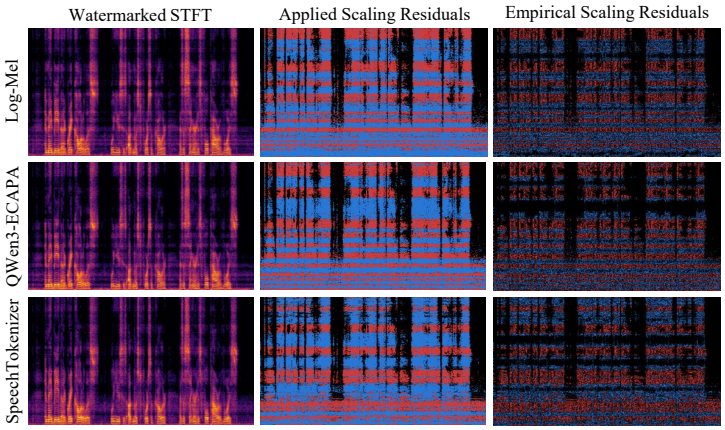


Figure 12: Gated-convolutional scale predictor paired with three detector representations.

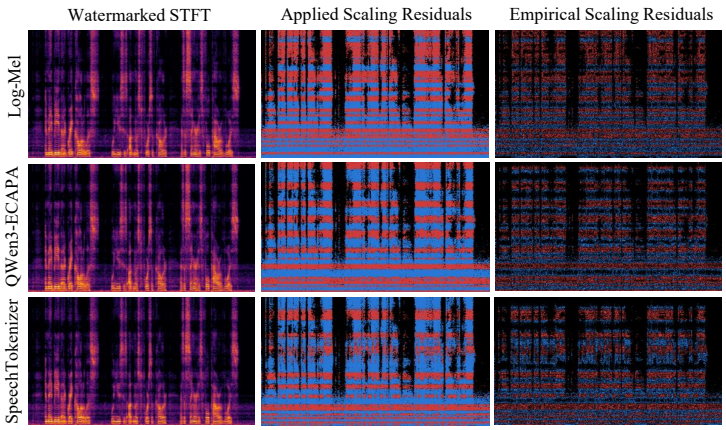


Figure 13: Conformer scale predictor paired with three detector representations.

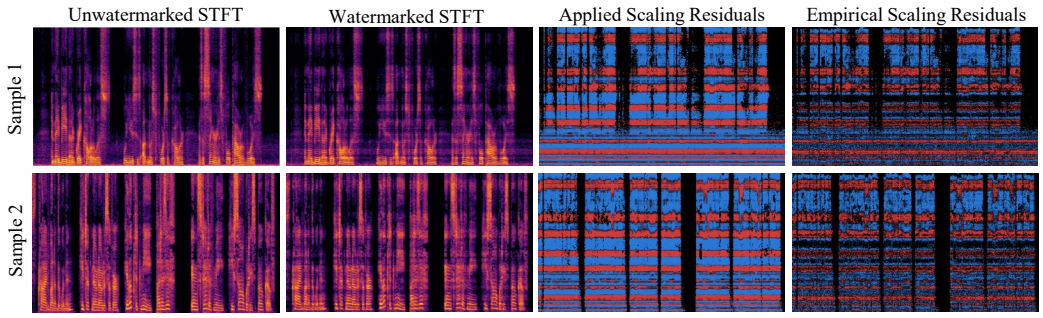


Figure 14: Visualization of watermark patterns on two different recordings. A fixed message is embedded into these recordings with the same audio watermarking model.

Table 10: Attack-wise bit accuracy under conventional audio edits, evaluated on RAW-Bench (%). Mean is computed before per-attack rounding. GN: Gaussian noise; BN: background noise; RV: reverberation; DC: dynamic range compression; DE: dynamic range expansion; LM: limiter; LP: low-pass filters; HP: high-pass filters; EQ: equalizer; TS: time stretch; TJ: time jitter; PI: polarity inversion; GA: gain adjustment; QN: quantization; PS: phase shift; MP: MP3 Compression; OG: Vorbis Compression; AC: AAC compression. **Bold**:best.

System	GN	BN	RV	DC	DE	LM	LP	HP	EQ	TS	TJ	PI	GA	QN	PS	MP	OG	AA	Mean
AudioSeal	99	**100**	87	99	99	99	**100**	96	91	97	**100**	19	99	**100**	62	**100**	**100**	**100**	91.46
WavMark	83	**100**	89	98	95	95	**100**	**100**	**100**	82	99	**100**	98	**100**	**100**	42	90	**100**	92.83
Timbre†	97	**100**	99	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	99	**100**	**100**	99.72
AWARE	99	**100**	98	**100**	98	99	**100**	**100**	**100**	80	**100**	**100**	**100**	**100**	**100**	99	**100**	**100**	98.50
VoiceMark	82	89	83	91	93	93	85	92	71	88	89	90	88	89	89	77	89	89	87.09
AlignMark	92	92	86	90	88	87	85	91	80	81	91	89	86	92	89	84	90	91	88.01
SimpleMark	98	99	95	99	98	99	98	99	94	98	99	99	99	99	99	97	99	99	98.09
Linear + Log-Mel	96	98	94	97	97	98	97	98	92	98	97	98	98	98	97	92	98	98	96.76
Linear + SpeechTokenizer	95	97	94	97	97	97	96	98	91	97	97	98	98	97	97	91	97	98	96.25
Gated conv. + Log-Mel	96	98	94	97	97	98	96	98	92	97	97	98	98	98	97	93	98	98	96.75
Gated conv. + Qwen3-ECAPA	97	98	94	98	98	98	98	98	93	98	98	99	98	98	98	96	98	98	97.51
Gated conv. + SpeechTokenizer	94	97	93	97	97	97	96	97	91	97	97	98	98	97	97	90	97	98	95.90
Conformer + Log-Mel	97	99	94	98	98	99	98	97	90	99	98	99	98	99	98	94	98	99	97.24
Conformer + Qwen3-ECAPA	97	98	96	98	98	98	98	99	94	98	98	99	99	98	98	95	99	99	97.77
Conformer + SpeechTokenizer	95	97	94	97	97	97	97	97	92	97	98	98	98	97	98	91	98	98	96.42
Log-STFT	98	99	98	99	99	99	99	99	99	99	99	99	99	99	99	97	99	99	98.80
EnCodec	85	86	83	85	87	85	81	85	77	88	86	86	84	86	88	78	85	86	84.47
WavLM-L0	83	88	86	90	90	89	85	88	78	87	89	90	90	89	89	83	89	89	87.33
WavLM-L3	66	68	70	69	69	69	66	69	65	65	72	71	71	71	72	64	70	70	68.77
WavLM-L12	56	56	57	56	57	57	57	55	55	56	56	57	58	56	57	56	58	58	56.69
ECAPA-TDNN	62	64	64	64	64	64	62	64	63	63	64	65	65	64	64	61	64	64	63.53
- Fixed τ	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**99.84**
+ Simulated Attacks	96	98	94	98	98	98	98	98	91	98	98	99	98	98	99	96	98	98	97.32
+ $\mathcal{L}_{\text{mel}}$	89	91	83	91	91	91	89	92	81	91	91	93	93	91	93	78	92	92	89.65
+ $\mathcal{L}_{\text{mel}} + \mathcal{L}_{\text{adv}}$	89	91	83	91	91	91	89	92	81	91	91	93	93	91	93	78	92	92	89.58
Additive Embedder	98	99	95	99	98	99	98	99	94	98	98	99	99	99	99	95	99	99	97.93
Additive + $\mathcal{L}_{\text{mel}}$	93	95	89	96	96	96	95	96	86	95	96	97	96	96	96	88	96	96	94.30
Additive + $\mathcal{L}_{\text{mel}} + \mathcal{L}_{\text{adv}}$	93	96	89	96	96	96	95	96	86	95	96	97	96	96	97	88	97	96	94.58
Content-Agnostic	96	97	93	97	97	97	97	98	92	97	97	98	98	97	98	92	97	98	96.44
Minimal Example	94	97	94	96	96	96	96	97	92	97	96	97	97	97	96	91	97	97	95.80
- ASP	94	96	93	96	96	96	96	96	92	97	95	97	96	96	96	94	97	97	95.58

Content-Agnostic Behavior across Two Host Recordings. Finally, we compare two different host recordings carrying the same watermark message, embedded by the same Linear+Qwen3-ECAPA system. Although their STFT spectrograms differ, the applied and empirical scaling residuals exhibit nearly identical frequency pattern, supporting the content-agnostic nature of the learned watermark.

C.6 Attack-Wise Results

This section provides attack-wise accuracy used to compute metrics in Tables 1 to 4 and 8.

C.6.1 ATTACK-WISE ACCURACY UNDER RAW-BENCH AUDIO EDITS

Table 10 lists bitwise accuracy under each conventional audio edit attack. Compared with other audio representations, QWen3-ECAPA consistently achieves higher bit accuracy under strong attacks like GN, RV and MP, making it the best-performing audio representation under the SimpleMark framework. We suppose that this kind of robustness is inherited from large scale pretraining.

C.6.2 ATTACK-WISE ACCURACY UNDER NEURAL CODEC RESYNTHESIS

We evaluate robustness against neural codec resynthesis attacks on 30 neural codecs. Table 11 first lists the model identifiers and configurations of these neural codecs. Then in Tables 12 and 13, we report bitwise accuracy and existence accuracy on each neural codec, respectively. The results show that the advantage of SimpleMark mainly comes from its strong performance on SemantiCodec (E1–E3) and SoCodec (M). However, it underperforms AlignMark on Language-Codec (J), WavTokenizer (I1, I2), and TiCodec (T2, T3). Furthermore, we notice that the attack-wise bit accuracies of SimpleMark instances and VoiceMark are highly correlated, indicating that they may rely on some similar mechanisms to defend neural codec resynthesis attacks.

Table 11: Neural codecs used for robustness evaluation.

ID	Codec	Configuration
A	SpeechTokenizer	`hubert_avg`
B	DAC	`DAC24k`
C	HiFi-Codec	`16k_320d_large_universal`
D1	EnCodec	`6b24k`
D2		`12b24k`
E1	SemantiCodec	`1.40kbps_16kHz`
E2		`0.70kbps_16kHz`
E3		`0.35kbps_16kHz`
F1	FunCodec	`en-libritts-16k-gr1nq32ds320`
F2		`zh_en-general-16k-nq32ds320`
G	FACodec	`encodec-decoder-v2_16k`
H	LLM-Codec	`llm_codec`
I1	WavTokenizer	`small_320_24k_4096`
I2		`medium_320_24k_4096`
J	Language-Codec	`language-codec`
K	HILCodec	`hilcodec_speech`
L1	Vocos	`vocos_encodec_6`
L2		`vocos_encodec_12`
M	SoCodec	`16384x4_120ms_16khz_chinese`
P1	SNAC	`24khz`
P2		`44khz`
Q1	XCodec2	`XCodec2`
R	AudioDec	`audiodec_24k_320d`
S	BigCodec	`BigCodec`
T1	TiCodec	`ticodec_1g1r`
T2		`ticodec_1g2r`
T3		`ticodec_1g4r`
U	Mimi	`mimi-codec`
V	Spectral Codecs	`spectral_codecs`
W	SQ-Codec	`sqcodec50dim9`

Table 12: Attack-wise bit accuracy (BA, %) under neural-codec resynthesis attacks. Codec identifiers are defined in Table 11. **Bold**: best.

System	A	B	C	D1	D2	E1	E2	E3	F1	F2	G	H	I1	I2	J	K	L1	L2	M	P1	P2	Q1	R	S	T1	T2	T3	U	V	W	Mean
AudioSeal	46.98	88.76	46.79	63.48	88.79	46.35	46.20	46.55	48.71	45.86	45.25	45.55	46.61	47.47	40.67	64.73	45.46	45.38	48.18	44.97	51.31	46.17	48.55	46.96	46.04	50.36	47.57	47.60	47.36	47.43	50.74
WavMark	53.15	89.63	53.44	53.83	53.77	55.34	55.51	55.69	52.91	52.90	52.68	54.38	53.11	53.16	53.35	53.67	52.83	52.80	56.75	53.23	53.10	53.00	54.43	53.06	53.48	50.13	53.92	54.59	52.91	45.53	54.54
Timbre†	55.64	99.94	58.67	57.69	70.66	80.32	72.89	60.10	63.97	63.97	56.69	51.38	50.38	50.62	50.39	81.67	61.55	68.11	58.11	50.82	61.73	50.60	56.00	51.87	50.74	50.88	52.39	51.47	70.70	51.79	60.39
AWARE	59.86	99.65	57.50	66.33	74.48	59.60	54.79	51.37	78.51	78.51	63.34	55.88	55.58	55.59	49.51	73.48	69.24	73.77	51.46	55.99	60.24	56.66	63.90	60.00	52.27	54.66	59.14	57.37	85.47	50.53	62.82
VoiceMark	95.41	97.87	71.27	95.98	97.05	96.65	93.16	81.04	98.17	98.17	91.88	73.18	63.67	66.64	49.22	96.42	95.75	97.05	90.03	66.23	88.38	69.87	96.68	77.31	64.63	64.92	74.67	88.21	97.51	50.73	82.93
AlignMark	92.45	98.82	82.75	**98.89**	99.32	83.44	68.38	58.64	95.02	95.02	93.04	81.49	74.26	75.99	**91.26**	95.78	98.28	98.72	71.37	77.97	91.87	75.33	**96.78**	**88.40**	63.16	75.30	**89.46**	88.72	99.23	49.80	84.96
SimpleMark	92.48	99.75	81.54	96.39	98.26	98.79	97.82	90.56	97.63	97.63	89.48	76.98	64.00	71.41	50.00	97.07	96.72	98.39	96.09	67.50	94.22	73.53	91.34	80.33	62.84	68.61	79.30	89.13	99.00	52.79	84.99
Linear + Log-Mel	84.41	98.52	75.44	86.74	90.03	98.96	97.76	88.86	93.76	93.76	81.96	70.80	60.21	66.64	49.83	92.60	88.50	91.06	96.88	63.14	84.76	64.82	86.36	75.13	62.57	64.92	73.63	79.66	94.85	50.73	80.24
Linear + SpeechTokenizer	93.49	99.64	79.17	92.79	95.77	98.91	97.79	91.00	98.16	98.16	91.01	75.99	65.50	72.23	50.21	97.10	95.59	97.67	97.51	70.41	93.59	68.22	93.28	81.15	64.53	68.45	78.11	89.30	97.13	50.77	84.75
Gated conv. + Log-Mel	85.00	98.49	75.53	85.91	89.48	98.74	97.39	88.23	92.55	92.55	82.25	70.05	60.91	67.17	47.20	91.32	88.19	90.85	95.76	63.86	86.25	65.33	84.94	75.36	61.80	64.19	73.38	80.43	93.45	51.11	79.92
Gated conv. + Qwen3-ECAPA	92.97	99.61	83.71	96.82	98.24	98.49	97.43	90.96	97.65	97.65	90.40	78.72	64.55	70.82	46.30	97.45	96.77	98.35	96.05	70.03	95.40	73.69	91.15	79.91	63.31	68.94	80.50	89.88	98.69	53.23	85.26
Gated conv. + SpeechTokenizer	93.46	99.67	78.26	93.14	96.09	99.03	97.89	90.57	98.48	98.48	91.52	75.67	63.55	69.91	49.19	97.69	95.38	97.73	97.55	68.86	93.09	68.37	92.42	79.56	63.72	67.20	77.30	88.75	96.98	50.22	84.32
Conformer + Log-Mel	88.78	99.61	77.69	89.89	93.07	99.54	98.55	89.90	96.16	96.16	86.54	73.32	63.74	69.64	46.92	96.17	92.45	94.74	98.06	66.43	88.95	66.87	89.75	79.02	65.99	68.30	77.23	83.59	96.61	51.32	82.83
Conformer + Qwen3-ECAPA	93.87	99.52	83.79	97.64	98.73	98.97	97.80	91.14	98.50	98.50	93.34	80.79	68.30	74.22	48.59	96.93	97.91	98.89	96.89	73.08	95.56	74.21	94.01	83.22	66.38	72.09	83.22	92.13	99.19	**56.72**	86.80
Conformer + SpeechTokenizer	94.02	99.71	79.48	94.32	96.88	99.06	97.86	90.30	98.25	98.25	92.97	76.15	66.92	73.50	48.52	97.81	96.16	97.97	97.56	69.92	93.50	69.17	95.08	81.98	65.79	68.89	77.89	89.71	98.04	52.68	85.28
Log-STFT	70.36	93.65	64.43	72.48	76.30	91.80	89.40	75.18	74.02	74.02	63.35	60.70	54.48	56.85	48.02	76.71	75.25	78.36	73.75	53.89	70.70	59.65	72.97	62.33	55.34	57.85	62.30	66.62	86.99	50.05	68.93
EnCodec	86.26	93.23	76.21	91.83	92.62	91.33	88.34	82.95	91.34	91.34	87.39	79.24	69.35	73.62	45.53	91.62	91.61	92.43	89.58	71.48	88.02	67.40	88.25	80.25	68.31	71.52	78.72	85.04	92.12	50.93	81.60
WavLM-L0	94.53	98.90	80.85	93.25	94.83	89.04	84.05	76.17	97.11	97.11	94.67	76.13	**74.39**	78.42	46.35	97.26	94.40	96.11	87.63	**79.36**	95.13	66.10	96.08	86.14	66.40	72.19	86.79	86.27	95.53	50.37	84.39
WavLM-L3	81.08	90.53	75.44	82.02	84.69	75.70	70.06	63.53	86.69	86.69	85.10	66.20	70.15	72.55	48.21	86.63	82.37	85.24	70.24	74.29	84.91	65.66	84.32	77.30	61.61	67.83	80.59	71.90	84.18	51.38	75.57
WavLM-L12	66.02	67.96	65.64	63.13	63.28	66.66	65.40	62.10	63.70	63.70	66.22	58.18	62.70	63.47	54.28	65.79	63.29	64.76	61.80	65.46	67.37	60.50	66.39	65.46	59.52	62.64	66.10	62.40	65.94	50.33	63.34
ECAPA-TDNN	69.48	76.59	63.19	72.14	73.83	71.00	64.15	56.26	74.63	74.63	70.58	63.43	58.05	59.54	51.09	73.49	72.79	74.21	70.53	60.79	68.80	63.69	71.29	64.39	53.98	57.73	61.27	65.80	74.87	54.13	66.21
- Fixed τ	**96.11**	**100.00**	**87.88**	98.74	**99.65**	**99.97**	**99.84**	**97.92**	**98.93**	**98.93**	**95.90**	**84.07**	72.93	**78.82**	47.75	**99.18**	**98.89**	**99.53**	**98.83**	78.02	**98.39**	**83.50**	94.58	87.83	**73.08**	**78.36**	87.36	**95.98**	**99.90**	54.14	**89.50**
+ Simulated Attacks	93.49	99.65	83.58	96.95	98.18	98.47	96.78	89.65	97.70	97.70	90.26	79.14	68.59	74.18	46.14	97.94	96.52	98.04	96.14	70.74	94.47	75.49	91.97	83.47	66.45	72.37	82.45	89.38	98.66	52.55	85.90
+ $\mathcal{L}_{\text{mel}}$	81.83	96.26	77.45	89.80	92.86	91.38	87.28	78.07	89.61	89.61	81.45	69.65	57.52	63.71	48.23	85.86	89.17	92.43	82.52	61.09	85.90	71.45	77.92	71.18	58.57	63.24	71.86	77.29	92.73	52.71	77.62
+ $\mathcal{L}_{\text{mel}} + \mathcal{L}_{\text{adv}}$	81.70	96.21	77.34	89.72	92.81	91.41	87.37	77.88	89.66	89.66	81.56	69.62	57.42	63.67	48.28	85.84	89.02	92.35	82.57	61.08	85.84	71.34	77.81	71.09	58.54	63.31	71.85	77.39	92.73	52.60	77.59
Additive Embedder	92.50	99.58	80.08	96.68	98.19	98.54	97.09	89.77	97.74	97.74	89.89	77.24	63.76	71.25	49.74	96.29	97.11	98.47	95.65	67.00	94.27	71.34	91.58	78.31	63.82	68.10	79.02	89.85	98.91	53.54	84.77
Additive + $\mathcal{L}_{\text{mel}}$	87.58	98.47	73.23	91.72	94.95	96.49	93.22	82.30	92.99	92.99	84.02	68.56	58.23	64.86	48.78	91.20	92.96	95.92	91.60	60.05	89.66	63.34	86.88	70.02	56.52	59.42	70.15	80.53	96.40	51.31	79.48
Additive + $\mathcal{L}_{\text{mel}} + \mathcal{L}_{\text{adv}}$	87.74	98.51	73.45	91.51	94.91	96.96	93.90	82.68	92.89	92.89	84.05	69.04	58.25	64.99	48.76	91.27	92.91	95.88	92.15	60.16	89.72	63.55	86.82	69.94	56.32	59.43	70.09	80.13	96.43	50.70	79.53
Content-Agnostic	90.84	99.18	80.74	94.95	96.71	97.88	96.53	88.97	95.91	95.91	87.99	76.36	63.22	70.71	51.24	96.53	95.49	97.15	95.33	66.55	92.54	71.64	89.23	79.03	63.35	67.70	79.68	87.19	97.97	52.47	83.97
Minimal Example	83.30	97.74	74.23	85.48	88.67	98.40	96.90	87.95	92.28	92.28	80.68	69.49	59.45	65.17	50.13	91.41	86.80	89.66	95.62	61.15	84.75	65.50	84.93	72.53	60.83	63.07	71.88	79.46	93.40	50.41	79.12
- ASP	86.47	98.65	74.58	88.20	91.70	98.27	97.12	89.46	94.70	94.70	83.68	69.36	60.49	64.93	46.29	95.08	91.29	94.36	95.23	61.71	88.38	64.88	86.11	71.37	60.06	63.78	72.29	80.07	94.78	54.54	80.42

Table 13: Attack-wise existence accuracy (EA, %) under neural-codec resynthesis attacks. Watermark existence labels are calculated with a threshold of 75%. Codec identifiers are defined in Table 11. **Bold**: best.

System	A	B	C	D1	D2	E1	E2	E3	F1	F2	G	H	I1	I2	J	K	L1	L2	M	P1	P2	Q1	R	S	T1	T2	T3	U	V	W	Mean
AudioSeal	50.00	82.94	50.04	51.24	85.68	50.00	50.00	50.00	50.02	50.00	50.00	50.00	50.00	54.96	50.00	50.00	50.00	50.00	50.00	50.00	51.09	49.94	49.98	50.00	50.00	49.79	50.02	50.02	50.00	49.94	52.52
WavMark	50.13	95.92	50.29	50.75	50.25	50.71	50.56	50.50	50.02	50.17	50.02	50.33	50.25	50.13	50.19	50.06	49.87	50.13	50.79	50.04	50.15	49.75	50.27	50.34	50.23	51.37	50.08	50.08	50.02	49.75	51.77
Timbre†	52.44	97.40	55.08	53.42	68.07	81.95	71.28	56.39	58.87	58.87	53.76	50.69	50.23	50.10	50.46	83.28	56.77	64.68	54.60	50.63	57.58	50.23	53.30	51.07	50.25	50.53	50.69	50.40	68.44	50.53	58.40
AWARE	56.30	98.05	54.10	65.17	79.08	55.99	52.19	50.25	85.40	85.40	60.42	52.94	52.46	52.67	49.85	76.53	70.29	78.02	50.61	53.32	56.79	53.53	61.35	56.68	50.19	51.95	55.99	53.99	93.40	50.46	62.11
VoiceMark	97.50	98.32	73.53	97.65	97.92	97.69	96.35	84.85	98.05	98.05	95.65	75.92	64.43	67.44	49.43	97.69	97.75	98.07	94.62	66.95	92.96	72.06	97.77	80.65	65.11	64.85	78.32	93.13	97.79	50.48	84.70
AlignMark	93.13	96.87	88.85	97.71	97.98	88.07	70.38	56.05	93.72	93.72	93.21	88.95	**78.32**	81.32	**93.24**	94.83	96.95	97.19	73.09	83.70	96.34	79.69	97.02	92.46	60.50	**79.58**	**95.13**	95.10	97.35	50.02	86.68
SimpleMark	97.88	98.05	90.44	98.24	98.21	98.15	97.86	96.64	98.30	98.30	97.12	82.50	61.41	73.76	49.92	97.88	98.13	98.26	98.13	66.66	98.02	76.93	97.90	87.94	59.20	67.96	87.29	96.76	98.07	50.99	87.23
Linear + Log-Mel	93.57	98.00	80.84	95.06	96.81	98.00	98.02	95.40	97.61	97.61	91.26	72.14	56.74	64.48	49.77	97.44	96.53	96.93	97.71	60.38	93.51	63.15	94.87	79.62	59.58	62.33	77.90	87.08	98.05	50.13	83.35
Linear + SpeechTokenizer	97.54	97.98	87.14	97.08	97.86	98.34	98.30	96.97	97.60	97.60	97.29	81.16	64.08	74.68	50.25	98.07	97.79	97.88	98.23	70.99	97.39	67.42	98.00	89.66	62.23	68.30	84.96	96.64	97.94	50.32	86.99
Gated conv. + Log-Mel	93.72	98.24	81.09	94.52	96.49	98.36	98.05	95.08	97.82	97.82	91.13	70.31	57.25	65.80	48.97	97.52	96.28	97.18	97.65	61.72	94.68	63.03	93.84	79.73	58.44	61.34	76.91	88.51	97.75	50.27	83.32
Gated conv. + Qwen3-ECAPA	97.73	98.21	93.09	98.02	97.90	98.26	98.05	**97.16**	98.11	98.11	97.42	86.79	61.72	72.10	49.01	98.03	98.32	97.98	98.26	70.88	98.21	77.79	97.65	88.23	60.04	68.40	88.78	96.87	98.21	51.09	87.68
Gated conv. + SpeechTokenizer	97.71	98.11	85.44	97.00	97.82	98.36	98.21	96.30	97.86	97.86	97.33	80.52	60.55	70.31	49.79	98.42	97.79	98.00	97.77	68.19	97.84	68.05	97.56	86.91	60.50	65.44	83.15	95.95	98.05	50.04	86.23
Conformer + Log-Mel	95.78	97.92	83.78	96.51	97.50	97.81	97.86	95.61	97.88	97.88	94.26	76.18	61.09	70.50	48.72	98.05	97.16	97.56	98.30	64.83	95.92	65.88	96.24	85.50	64.01	67.46	82.81	91.81	98.09	50.71	85.45
Conformer + Qwen3-ECAPA	98.07	98.03	**93.13**	98.28	98.23	97.86	97.98	96.66	98.02	98.02	97.81	**89.64**	67.54	78.49	49.62	98.02	98.07	97.98	98.24	76.24	**98.24**	78.47	98.05	92.18	65.10	74.35	92.82	**97.63**	97.98	**53.36**	**89.14**
Conformer + SpeechTokenizer	97.44	97.63	87.88	97.54	97.67	98.11	98.07	96.43	97.82	97.82	97.40	81.83	65.88	77.04	49.20	98.02	97.92	97.90	98.13	70.44	97.82	69.05	98.02	90.21	63.95	68.70	85.19	96.53	98.17	51.49	87.31
Log-STFT	62.04	98.91	55.36	64.90	71.62	96.68	94.41	71.34	67.88	67.88	54.41	53.05	50.76	51.45	49.81	72.58	70.34	75.82	67.73	50.46	62.60	52.60	65.44	53.82	50.86	51.62	53.42	58.15	91.49	50.04	64.58
EnCodec	94.56	97.65	81.35	97.29	97.39	96.70	95.52	90.94	96.93	96.93	95.38	86.55	70.36	76.83	48.45	97.16	97.19	97.35	96.83	73.36	95.80	66.28	95.57	87.79	**67.54**	73.84	85.74	93.53	97.08	50.52	86.61
WavLM-L0	98.07	98.19	89.54	97.56	97.94	96.39	92.33	81.32	98.05	98.05	**98.03**	81.37	78.26	**84.50**	48.97	98.02	97.73	98.07	95.73	**86.66**	98.23	64.33	**98.07**	**94.64**	64.27	74.39	95.02	94.68	97.79	50.13	88.21
WavLM-L3	88.68	96.60	80.50	89.73	91.97	80.88	71.22	61.87	94.52	94.52	93.23	64.37	71.13	76.05	49.47	94.39	90.19	93.11	71.41	78.57	92.96	63.82	92.46	83.65	58.36	67.12	88.11	73.93	92.61	50.44	79.86
WavLM-L12	63.19	66.85	62.69	59.87	60.08	64.47	62.54	58.05	60.13	60.13	63.74	54.94	59.37	59.87	51.55	62.58	59.64	61.74	57.92	62.25	65.57	56.43	63.53	62.29	55.73	58.30	63.30	58.74	63.02	50.11	60.29
ECAPA-TDNN	70.04	81.77	60.55	74.48	77.14	73.26	61.85	53.28	78.91	78.91	72.04	61.07	54.14	56.30	50.15	77.04	75.95	78.13	72.39	57.73	68.74	60.92	73.07	61.91	51.87	54.64	58.00	64.10	79.05	51.98	66.31
- Fixed τ	72.81	96.11	59.64	89.12	93.84	90.84	85.10	67.06	89.12	89.12	66.91	54.56	51.01	52.90	49.96	89.33	88.40	92.42	81.53	51.32	79.60	53.89	76.58	55.08	50.94	51.98	54.73	68.84	93.11	50.04	71.53
+ Simulated Attacks	**98.26**	98.17	92.69	98.24	98.44	98.13	97.98	96.01	98.13	98.13	97.18	85.67	68.13	78.28	48.95	98.19	98.24	98.26	98.44	71.60	98.07	**80.10**	97.98	92.16	64.58	74.45	91.58	96.43	98.30	50.82	88.72
+ $\mathcal{L}_{\text{mel}}$	90.17	97.98	83.61	96.43	97.29	96.89	94.81	84.33	96.43	96.43	89.52	68.99	54.24	60.86	49.14	93.70	96.45	97.37	89.90	57.16	94.37	73.36	83.99	72.27	54.94	59.33	74.06	83.26	97.54	51.13	81.20
+ $\mathcal{L}_{\text{mel}} + \mathcal{L}_{\text{adv}}$	89.52	97.94	83.80	96.58	97.39	96.81	94.62	84.01	96.60	96.60	89.83	69.27	54.08	61.13	49.22	93.91	96.37	97.27	89.83	57.44	94.10	73.09	84.48	72.39	54.89	59.39	73.95	83.45	97.44	51.13	81.22
Additive Embedder	97.58	97.92	87.65	98.02	98.05	97.86	98.03	95.76	98.26	98.26	96.97	83.40	61.77	73.44	50.29	98.05	98.30	98.15	97.92	65.50	98.05	73.51	97.94	85.13	60.99	67.08	85.76	96.89	98.07	51.26	86.86
Additive + $\mathcal{L}_{\text{mel}}$	95.38	97.60	76.55	97.40	97.73	97.86	97.42	89.29	97.67	97.67	92.58	68.34	54.58	62.84	49.83	97.06	97.58	97.92	96.51	56.39	97.10	60.42	95.38	70.69	53.17	56.37	70.76	88.02	97.94	50.50	81.95
Additive + $\mathcal{L}_{\text{mel}} + \mathcal{L}_{\text{adv}}$	95.65	97.46	76.35	97.18	97.73	97.90	97.54	89.79	97.73	97.73	92.56	68.82	54.77	62.84	49.60	97.31	97.69	97.98	97.18	56.77	97.00	60.50	95.65	70.71	53.05	55.99	70.42	87.96	98.09	50.38	82.01
Content-Agnostic	96.68	**99.60**	80.99	**98.76**	**99.47**	**99.37**	**99.26**	93.26	**99.01**	**99.01**	93.07	72.56	55.50	63.11	49.98	**99.39**	**99.12**	**99.37**	**98.82**	58.26	97.90	64.75	94.75	77.63	55.19	58.95	79.14	91.95	**99.45**	50.42	84.16
Minimal Example	85.80	99.52	67.29	89.96	93.84	99.33	99.06	91.83	97.56	97.56	80.29	60.65	52.21	55.44	49.96	96.34	91.45	94.98	98.59	53.38	88.63	56.70	88.74	65.55	52.98	54.22	64.01	78.49	98.24	50.19	78.43
- ASP	94.73	98.32	78.82	95.42	96.95	98.02	98.13	95.92	97.73	97.73	92.40	70.13	57.54	63.13	49.08	98.11	97.12	97.82	98.34	57.92	96.41	62.65	94.90	73.21	56.47	61.43	74.85	87.92	98.02	51.85	83.04

Table 14: Attack-wise bit/existence accuracy (BA/EA, %) under voice-cloning attacks. Watermark existence labels are calculated with a threshold of 75%. **Bold**: best.

System	Qwen3-TTS	LongCat	F5-TTS	MaskGCT	Fish-Speech	Spark-TTS	CosyVoice2	Mean
AudioSeal	49.99/48.70	49.55/49.30	50.63/50.80	50.32/51.20	50.18/50.10	50.24/50.50	50.50/50.40	50.20/50.14
WavMark	49.46/50.60	49.66/49.10	50.62/51.40	49.61/50.90	49.65/49.90	49.99/49.40	50.08/50.30	49.87/50.23
Timbre†	56.76/52.80	65.04/60.20	96.40/95.30	62.02/57.20	53.54/51.00	57.95/53.09	96.82/97.20	69.79/66.68
AWARE	49.89/49.30	52.61/50.10	52.16/50.80	52.34/50.60	51.23/49.70	50.93/49.30	52.00/50.60	51.59/50.06
VoiceMark	92.22/94.90	95.69/95.20	94.66/96.50	90.80/94.00	89.99/94.40	84.44/88.10	93.59/95.60	91.63/94.10
AlignMark	60.80/58.80	75.92/80.40	75.43/79.00	72.99/76.40	63.16/61.70	60.71/59.20	67.92/69.30	68.13/69.26
SimpleMark	93.14/97.63	93.54/97.30	96.97/96.10	90.91/97.90	85.35/94.10	92.01/96.17	98.71/98.50	92.95/96.81
Linear + Log-Mel	87.79/95.90	91.40/96.50	96.40/96.60	86.55/95.10	79.36/86.00	85.36/92.44	97.47/97.80	89.19/94.33
Linear + SpeechTokenizer	94.56/97.50	96.85/97.70	96.64/96.20	95.50/98.50	89.80/95.50	89.95/94.22	97.55/98.60	94.41/96.89
Gated conv. + Log-Mel	86.98/94.90	89.86/97.10	95.86/97.30	85.84/94.70	78.17/85.60	84.74/93.01	97.16/98.20	88.37/94.40
Gated conv. + Qwen3-ECAPA	93.60/97.30	93.76/97.30	96.61/96.50	92.91/98.10	86.76/93.70	91.87/95.64	98.71/97.80	93.46/96.62
Gated conv. + SpeechTokenizer	93.86/98.00	96.79/97.40	95.90/**97.50**	95.15/97.70	88.74/96.00	88.56/94.70	97.86/97.90	93.84/97.03
Conformer + Log-Mel	89.15/95.40	92.31/96.00	96.01/96.80	87.95/95.30	81.19/87.50	85.88/93.29	97.12/97.40	89.94/94.53
Conformer + Qwen3-ECAPA	94.39/98.00	94.58/97.00	96.91/96.60	93.75/97.60	88.41/95.40	93.67/**97.26**	98.09/98.20	94.26/97.15
Conformer + SpeechTokenizer	94.72/**98.17**	97.74/96.60	96.34/96.40	95.35/97.10	89.85/95.30	88.61/93.41	97.80/98.10	94.34/96.44
Log-STFT	75.00/80.10	75.89/81.00	94.90/96.80	71.09/72.20	69.62/69.70	69.95/70.03	95.49/98.10	78.85/81.13
EnCodec	85.20/92.50	84.67/92.00	85.94/92.90	86.04/94.90	80.62/86.60	83.18/90.01	83.84/89.30	84.21/91.17
WavLM-L0	84.15/91.70	91.81/97.30	88.06/93.80	92.07/97.40	83.69/90.40	76.93/82.51	87.67/95.00	86.34/92.59
WavLM-L3	66.45/66.11	78.31/83.80	72.95/76.70	77.66/84.20	67.91/67.70	64.54/62.48	72.04/73.80	71.41/73.54
WavLM-L12	58.72/55.01	65.95/64.20	59.48/55.70	65.46/62.50	57.34/53.40	58.29/54.62	63.38/60.70	61.23/58.02
ECAPA-TDNN	60.34/57.50	65.67/64.20	69.05/70.40	65.65/64.30	59.81/56.70	57.49/54.44	65.60/64.40	63.37/61.71
- Fixed τ	**98.14**/97.80	**98.52/98.40**	**98.32**/96.00	**95.94/98.70**	**93.55/96.90**	**96.53**/96.86	**99.95**/98.20	**97.28/97.55**
+ Simulated Attacks	92.88/96.90	92.97/96.90	96.23/97.10	92.11/97.20	85.79/94.30	91.75/96.33	97.88/98.10	92.80/96.69
+ $\mathcal{L}_{\text{mel}}$	80.51/86.60	81.83/91.10	90.31/95.30	80.19/88.70	73.84/76.70	83.02/91.19	92.77/96.90	83.21/89.50
+ $\mathcal{L}_{\text{mel}} + \mathcal{L}_{\text{adv}}$	80.53/88.23	81.94/89.80	90.24/95.30	80.40/87.80	73.95/75.70	82.70/90.96	92.62/97.20	83.20/89.28
Additive Embedder	93.67/97.60	93.51/95.90	97.06/96.70	91.25/97.60	86.09/92.00	91.12/95.54	98.41/97.50	93.02/96.12
Additive + $\mathcal{L}_{\text{mel}}$	86.34/93.80	88.65/94.80	93.92/96.20	86.54/95.10	78.69/83.70	86.47/94.53	96.16/97.50	88.11/93.66
Additive + $\mathcal{L}_{\text{mel}} + \mathcal{L}_{\text{adv}}$	87.17/94.90	88.74/94.30	94.10/96.10	86.91/95.40	79.35/84.90	86.40/94.04	96.28/98.00	88.42/93.95
Content-Agnostic	90.89/96.50	91.06/96.20	95.17/97.00	89.79/98.20	83.93/92.50	89.86/96.33	97.32/97.30	91.15/96.29
Minimal Example	87.84/96.10	89.88/96.30	96.20/96.00	86.29/94.50	79.16/86.80	85.77/94.50	97.75/**98.80**	88.98/94.71
- ASP	89.15/95.50	89.56/94.20	93.47/94.50	88.34/97.20	81.91/92.00	87.08/92.96	97.71/97.50	89.60/94.84

C.6.3 Attack-Wise Accuracy under Voice Cloning Attacks

Table 14 reports bitwise accuracy and existence accuracy on each voice-cloning system. The results show that voice cloning models with neural codecs, like MaskGCT, Fish-Speech and Spark-TTS, are generally more likely to remove audio watermarks, reinforcing the consensus that neural codec resynthesis remains one of the most challenging attacks for robust audio watermarking.

C.6.4 Attack-Wise Accuracy under Other AI Attacks

Table 15 reports attack-wise bit accuracy and existence accuracy under four remaining AI attacks in DeepMark benchmark. Accuracy under VAE resynthesis is generally low because RAVE (Caillon & Esling, 2021), the VAE used in DeepMark, produces relatively low-quality reconstructions.

D Limitations

SimpleMark is designed to show that robust audio watermarking models does not necessarily need to be complex. Since simplicity is prioritized, we leave out complementary mechanisms, such as model ensemble (Yuan et al., 2025) and spectral-temporal masking (Li et al., 2026a), that may further improve performance at the cost of additional complexity. Furthermore, despite achieving new state-of-the-art with a simplified framework, SimpleMark still lacks robustness against a few neural codes, such as language-codec and SQ-Codec. Finally, repeating the same message across audio frames provides the temporal redundancy needed to withstand voice cloning, but this design limits the payload capacity. Future work may investigate plug-in extensions to the SimpleMark framework that improve robustness against challenging neural codecs, perceptual quality, or payload capacity while preserving simplicity and interpretability.

Table 15: Attack wise bit/existence accuracy (BA/EA, %) under four AI attacks: neural vocoding (NV), VAE resynthesis (VAE), diffusion resynthesis (DFA), and speech enhancement (SE). Watermark existence labels are calculated with a threshold of 75%. **Bold**: best.

System	NV	VAE	DFA	SE	Mean
AudioSeal	57.23/52.16	41.35/50.00	49.54/50.06	87.03/83.70	58.79/58.98
WavMark	55.28/50.33	53.19/50.04	65.32/54.46	57.49/51.78	57.82/51.65
Timbre[†]	**100.00**/96.98	51.26/50.78	**100.00**/97.27	92.99/93.89	86.06/84.73
AWARE	99.01/97.94	53.61/51.43	99.74/98.07	97.65/98.00	87.50/86.36
VoiceMark	98.36/98.05	51.83/51.49	94.03/95.99	83.57/88.19	81.95/83.43
AlignMark	98.73/96.93	57.04/54.48	99.32/98.03	**98.25**/97.44	88.34/86.72
SimpleMark	99.91/98.38	56.67/53.49	98.27/98.38	96.47/97.73	87.83/87.00
Linear + Log-Mel	99.67/97.75	53.86/51.60	97.41/98.13	91.15/96.72	85.52/86.05
Linear + SpeechTokenizer	99.82/98.13	54.71/52.16	90.92/94.45	84.83/91.89	82.57/84.16
Gated conv. + Log-Mel	99.59/98.17	53.96/51.95	97.46/97.84	90.81/96.56	85.45/86.13
Gated conv. + Qwen3-ECAPA	99.74/98.17	55.07/52.35	97.75/98.24	96.20/**98.03**	87.19/86.70
Gated conv. + SpeechTokenizer	99.84/97.81	53.74/51.20	89.65/93.26	79.17/85.15	80.60/81.85
Conformer + Log-Mel	99.91/97.96	53.70/51.76	98.50/97.81	92.44/96.45	86.14/86.00
Conformer + Qwen3-ECAPA	99.84/98.36	58.82/55.27	97.75/97.96	95.62/97.77	88.01/**87.34**
Conformer + SpeechTokenizer	99.90/97.88	55.12/52.46	92.40/95.95	84.83/92.77	83.06/84.77
Log-STFT	99.67/**99.43**	51.72/50.25	99.74/**99.48**	93.79/97.18	86.23/86.59
EnCodec	93.36/97.75	60.01/**56.32**	86.89/93.49	90.42/96.55	82.67/86.03
WavLM-L0	98.70/98.24	59.25/55.78	93.22/97.67	67.42/66.72	79.65/79.60
WavLM-L3	90.37/96.68	57.54/54.48	78.48/84.62	61.09/57.73	71.87/73.38
WavLM-L12	67.69/66.16	53.04/50.95	64.87/61.83	51.46/50.27	59.27/57.30
ECAPA-TDNN	77.33/82.88	55.59/52.65	76.33/81.35	71.04/72.69	70.07/72.39
- Fixed τ	**100.00**/97.35	**62.26**/50.19	98.81/92.94	98.14/90.97	**89.80**/82.86
+ Simulated Attacks	99.76/98.32	58.00/54.64	98.01/98.24	95.66/97.82	87.86/87.25
+ $\mathcal{L}_{\text{mel}}$	97.92/97.90	55.81/53.02	90.57/95.80	90.34/95.99	83.66/85.68
+ $\mathcal{L}_{\text{mel}} + \mathcal{L}_{\text{adv}}$	97.94/97.92	55.63/53.09	90.26/95.52	90.26/95.94	83.52/85.62
Additive Embedder	99.83/98.07	55.06/52.50	97.26/98.09	95.49/97.82	86.91/86.62
Additive + $\mathcal{L}_{\text{mel}}$	99.26/98.00	52.46/50.82	95.76/97.88	90.99/96.58	84.62/85.82
Additive + $\mathcal{L}_{\text{mel}} + \mathcal{L}_{\text{adv}}$	99.27/98.11	52.65/50.80	95.39/98.00	91.81/96.95	84.78/85.97
Content-Agnostic	99.51/99.39	55.45/50.99	97.82/99.37	94.32/**98.03**	86.78/86.94
Minimal Example	99.39/99.41	52.78/50.44	97.04/98.99	90.16/94.62	84.84/85.86
- ASP	99.63/98.44	52.20/50.95	97.81/98.11	88.61/95.19	84.56/85.67