Investigating Codec-Internal Latent Audio Watermarking for Neural Codec Robustness

Zi Hu*
Digital Innovation Research Center
Duke Kunshan University
Kunshan, China

Houmin Sun*
Digital Innovation Research Center
Duke Kunshan University
Kunshan, China

Linxi Li
OfSpectrum, Inc.
Los Angeles, USA

Yechen Wang
OfSpectrum, Inc.
Los Angeles, USA

Liwei Jin
OfSpectrum, Inc.
Los Angeles, USA

Carsten Maple†
University of Warwick
Coventry, United Kingdom

Ming Li†
School of Artificial Intelligence
The Chinese University of Hong Kong, Shenzhen
China
mingli369@cuhk.edu.cn

Abstract

Neural audio codecs are challenging transformations for audio watermarking because they re-encode, quantize, and resynthesize speech. This paper investigates continuous latent-space watermarking for codec robustness. Instead of adding a watermark only to the waveform or spectrogram, we embed a 32-bit message into the continuous latent representation of a codec-like speech autoencoder. The pipeline uses a SEANet-style encoder-decoder, a Conformer-based message embedder, RVQ-guided latent decomposition, and a latent-domain detector trained under signal-processing and neural-codec transformations. Rather than proposing a final universal watermarking baseline, we characterize the trade-offs that appear when the watermark carrier is moved before neural decoding. On a fixed 1000-utterance external Emilia subset, EnCodec-aware training improves EnCodec-24k bit accuracy from 78.5% to 95.7% and 97.0%, while PESQ decreases from 3.578 to 3.368 and 3.286. The two EnCodec-aware checkpoints retain 95.7% and 96.7% bit accuracy after unseen DAC reconstruction but remain near chance, at 50.9–52.2%, under the tested HiFi-Codec configurations. Cross-codec transfer is therefore codec-dependent rather than universal in the evaluated setting.

CCS Concepts

• **Security and privacy → Digital rights management**.

Keywords

audio watermarking, audio codec, robustness, latent embedding, RVQ

1 Introduction

Recent progress in AI speech synthesis has made high-quality synthetic speech increasingly accessible [17, 21], raising urgent needs for reliable provenance tracking and misuse detection. Audio watermarking offers a practical mechanism for embedding hidden identity or message bits into generated speech, allowing later verification without requiring external metadata. To preserve perceptual quality, most recent deep-learning-based watermarking systems embed watermarks in the waveform or time-frequency domain with carefully constrained perturbations [3, 18, 19]. With differentiable or simulated augmentation, these encoder-detector systems can make watermarks imperceptible while remaining robust to common signal-processing attacks such as noise, filtering, resampling, reverberation, and compression.

However, neural audio codecs introduce a qualitatively different and more destructive transformation. Unlike simple filtering or lossy compression, modern codecs first map the waveform into a learned latent representation, quantize this representation, and then synthesize the output waveform through a neural decoder [5, 22, 23]. This analysis-quantization-synthesis process can discard weak watermark traces that are not aligned with the codec representation. In particular, RVQ maps continuous features to finite codebook entries, so small waveform- or spectrogram-level perturbations may be absorbed by quantization or reshaped by the decoder without preserving the message. As neural codecs become common in speech generation, distribution, and editing pipelines [21, 24], improving watermark robustness under codec transformations becomes a central challenge.

Motivated by this observation, we explore watermark embedding directly in the latent representation of a codec-like speech autoencoder. This placement lets message information interact with the features used by the decoder before waveform synthesis. Whether it improves robustness is an empirical question, so we evaluate matched EnCodec training separately from transfer to codec architectures absent from training.

We explore several ways of inserting message information into this latent space. Early attempts used multiplicative latent masking and a hard RVQ bottleneck in the reconstruction path, but both made optimization fragile: the former altered the scale of content-bearing latent features, while the latter forced speech reconstruction and message preservation through discontinuous codebook

*Equal contribution.
†Corresponding authors: Ming Li and Carsten Maple.

assignments. The reported system therefore uses a bounded additive residual in continuous latent space and keeps RVQ as a guidance and decomposition mechanism rather than as the mandatory reconstruction bottleneck. Rather than claiming a final optimized watermarking system, we use this pipeline to characterize the quality-robustness trade-off that appears when message recovery is trained through a continuous latent reconstruction path.

Our proposed pipeline encodes speech with a SEANet-like encoder, embeds a message-conditioned watermark in the continuous latent representation, reconstructs the waveform with a neural decoder, and trains a latent-domain detector to recover the message after codec and signal-processing transformations. We evaluate both perceptual quality and watermark robustness, focusing especially on the trade-off between imperceptibility and survival under neural codec attacks. The main contributions of this work are:

- We formulate codec-robust speech watermarking as an investigation of where and how message information can be embedded inside continuous latent representations of codec-like autoencoders.
- We develop a bounded additive continuous-latent embedding mechanism, guided by RVQ decomposition, that protects selected front residual directions while keeping the decoder input continuous.
- We characterize the trade-off among codec robustness, perceptual quality, reconstruction backbone capacity, and detector stability, including zero-shot transfer to DAC and HiFi-Codec reconstructions.

2 Related Work

Audio watermarking embeds recoverable information while preserving perceptual quality. Compared with traditional spread-spectrum or echo-hiding methods [1, 6], neural systems learn an embedder and detector jointly, optimizing imperceptibility, payload recovery, and robustness. Methods such as WavMark, AURA, SilentCipher, and related frameworks show strong robustness to common signal-processing attacks with extensive augmentation [3, 14, 18, 19]. However, most still embed watermarks in waveform or time-frequency representations, where weak message traces can be altered by neural codec pipelines that re-encode, quantize, and regenerate audio.

2.1 Neural Audio Codecs

Neural audio codecs are now widely used for compression and speech generation. EnCodec uses a SEANet-style encoder-decoder with an RVQ bottleneck to convert continuous acoustic features into compact discrete codes [5], while HiFi-Codec improves this paradigm with group-residual vector quantization [22]. Such discrete representations are also used as intermediate tokens in speech synthesis and audio language modeling [21, 24]. However, the same analysis-quantization-synthesis process can severely disrupt post-hoc watermarks.

2.2 Codec-Robust Watermarking

AudioSeal is a representative neural audio watermarking system for proactive voice-cloning detection [18]. It jointly trains a waveform watermark generator and localized detector and supports an optional 16-bit message. We evaluate its released message model under the same codec implementations as a sensitivity reference. Because its carrier and payload differ from our 32-bit latent carrier, this comparison is not an unconditional system ranking.

Our investigation differs from AudioSeal in the location and role of the codec-like structure. AudioSeal uses an encoder-decoder-style neural generator to synthesize a watermark mask or signal, which is then added to the waveform. The codec-like architecture is therefore used as a powerful watermark generator, while the actual embedding still occurs in the audio domain. In contrast, we use a codec-like autoencoder as the carrier pathway itself: the input waveform is first encoded into a continuous latent representation, the message-conditioned watermark is injected directly into this pre-decoder latent, and the decoder reconstructs speech from the modified latent. This changes the watermarking problem from waveform perturbation generation to latent representation modification. The goal of this paper is not to replace AudioSeal as a general-purpose watermarking system, but to investigate whether embedding in continuous pre-decoder latents can reduce the mismatch between watermark insertion and codec-style distortion, without assuming access to or modification of discrete codec tokens.

VoiceMark is closely related because it embeds watermarks into speaker-specific latents for zero-shot voice-cloning resistance [13]. It uses a pretrained RVQ tokenizer to separate content and speaker-specific components, embeds the watermark into the latter, and trains a transformer decoder with voice-cloning-simulated augmentations. This effectively bypasses RVQ disruption by selecting a stable latent carrier. However, VoiceMark targets voice-cloning transfer with a tokenizer-centered framework, whereas we study codec robustness for general deep-learning-based audio watermarking without assuming a specific pretrained tokenizer or fixed RVQ decomposition.

3 Method

Figure 1 gives an overview of the proposed continuous latent-space watermarking pipeline.

3.1 Notation and Definitions

We use $x \in \mathbb{R}^{1\times T}$ to denote the original speech waveform and $m \in \{0,1\}^K$ to denote the binary payload. In all main experiments, $K = 32$. The encoder and decoder of the codec-like autoencoder are denoted by E_θ and G_θ, respectively. The continuous latent representation is

$$z = \tanh(E_\theta(x)). \tag{1}$$

The hyperbolic tangent bounds the latent magnitude and keeps the watermark perturbation in a stable range.

The RVQ module produces a sequence of residual components $q_1, \dots, q_N$, whose sum is denoted by

$$z_q = \sum_{i=1}^{N} q_i. \tag{2}$$

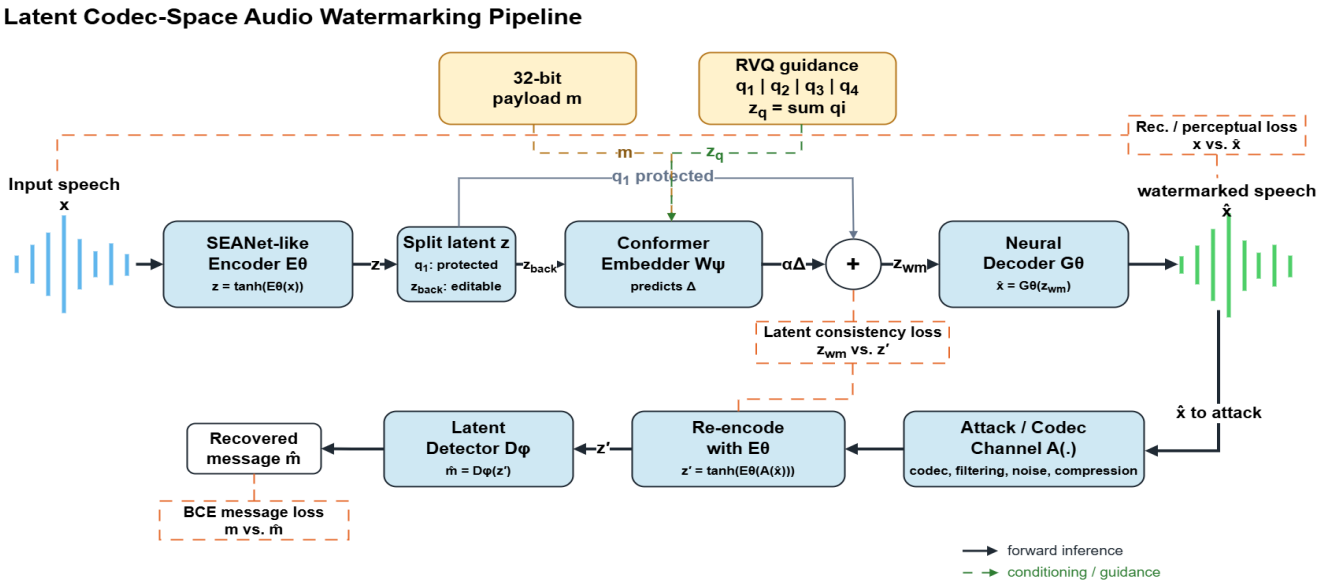


Figure 1: Overview of the proposed continuous latent-space audio watermarking pipeline.

In this work, z_q is a guidance representation for watermark generation and front-layer decomposition; it is not the final latent forced into the decoder. The first M RVQ residual components form a protected front part $z_{\text{front}} = \sum_{i=1}^{M} q_i$; in the principal configurations, $M = 1$ and therefore $z_{\text{front}} = q_1$, as drawn in Fig. 1. The remaining continuous component is $z_{\text{back}} = z - z_{\text{front}}$. A message-conditioned embedder predicts a bounded residual Δ, and the watermarked continuous latent is denoted as z_{wm}. The decoded watermarked waveform is $\hat{x} = G_\theta(z_{\text{wm}})$.

We use $\mathcal{A}(\cdot)$ for attack or codec transformations, including both signal-processing operations and neural audio codecs. The detector D_ϕ predicts the payload from an attacked waveform after re-encoding. In the experiments, *watermarked* denotes the normal output $\hat{x}$, *no-embed* denotes reconstruction after removing the inference-time watermark residual, and *codec-only* denotes passing clean audio through a codec without any watermark. The main reported checkpoints share the same SEANet-based latent embedding architecture and scalar loss weights. They differ in the EnCodec training procedure: the available codec routes, whether an EnCodec-specific straight-through gradient is used, and the attack-sampling emphasis.

3.2 Problem Setup

Given an input speech waveform $x \in \mathbb{R}^{1\times T}$ and a binary watermark message $m \in \{0,1\}^K$, our goal is to generate a watermarked waveform $\hat{x}$ that remains perceptually close to x while allowing the embedded message to be recovered after signal-processing and neural-codec transformations. We denote an attack or distortion operator as $\mathcal{A}(\cdot)$. The possibly attacked waveform is re-encoded before detection, and the recovered message is obtained as

$$\hat{m} = D_\phi(\tanh(E_\theta(\mathcal{A}(\hat{x})))). \tag{3}$$

The training objective balances perceptual reconstruction quality against message recovery accuracy under $\mathcal{A}$.

Unlike waveform-domain watermarking, our method embeds the message in the continuous latent z defined above. In the main system, $C = 128$ and the total downsampling factor is 320. A message-conditioned latent embedder W_ψ predicts a bounded perturbation, and G_θ reconstructs the waveform from the watermarked latent.

3.3 Latent Watermark Embedding

The embedder is a message-conditioned deep latent transformation module. The binary message is first projected into an embedding, which conditions a stack of Conformer blocks through FiLM-style feature modulation [7, 16]. The main configuration uses eight Conformer blocks with hidden dimension 768 and eight attention heads. The latent sequence is projected into this hidden dimension, processed by the message-conditioned blocks, and projected back to the original latent dimension. The final output is a bounded residual perturbation

$$\Delta = \tanh(W_\psi(u, m)), \tag{4}$$

where u is the latent feature used to condition the embedder. In the main configuration, $u = z_q$; the payload m provides the second conditioning input.

Earlier versions also explored multiplicative latent masking,

$$z_{\text{wm}} = z \odot M_\psi(z, m), \tag{5}$$

where the message-conditioned mask directly rescales the latent channels. This form was unstable in our experiments because the watermark operation was not localized to a small residual direction: even a mask close to one changes the relative magnitude of many content-bearing latent coefficients. After neural decoding, these scale changes could produce audible waveform distortion

and large reconstruction loss. The main reported variants therefore use additive residual injection, written first in its unprotected reference form as

$$z_{\text{wm}} = z + \alpha\Delta, \tag{6}$$

where α is a small learned scale. The protected main configuration applies the same additive operation only to the editable back component, as defined below.

3.4 RVQ-Guided Latent Decomposition

Residual vector quantization [5, 23] is used as a guidance and decomposition mechanism, not as the mandatory reconstruction bottleneck of the final model. Given z, a stack of RVQ layers produces residual components $q_1, \ldots, q_N$, whose sum is the guidance representation z_q defined above. In an early design, RVQ was placed directly between the encoder and decoder so that reconstruction had to pass through quantized watermarked latents. This can be viewed as an attempt to move from continuous latent-space embedding toward a more discrete codec-space embedding, where the watermark must survive codebook assignment before waveform synthesis. This hard quantized path was difficult to optimize for two reasons. First, nearest codebook assignment could absorb or erase small watermark perturbations before they reached the decoder. Second, the decoder had to reconstruct speech from discontinuous codebook entries while the detector still required message-specific information to survive. In preliminary runs, this design did not converge to reliable payload recovery, so we removed RVQ from the mandatory reconstruction path. This motivated the reported continuous latent-space design: the watermark is inserted before neural decoding, while RVQ is retained only as guidance for how the continuous latent relates to codec-like quantization.

In our main investigated configuration, reconstruction is kept continuous. RVQ serves two roles. First, W_ψ takes (z_q, m) to predict Δ. In Fig. 1, the solid z_{back} path through the Embedder box denotes the editable carrier to which $\alpha\Delta$ is applied, not an additional tensor input to W_ψ; the box summarizes the complete embedding operation. Second, the residual components define front and back latent parts:

$$z_{\text{front}} = \sum_{i=1}^{M} q_i, \qquad z_{\text{back}} = z - z_{\text{front}}. \tag{7}$$

The early RVQ component is used as the protected front direction in the main configuration, while the remaining continuous component is editable. This is an operational decomposition of z, not a claim that the RVQ components have a universally fixed perceptual ordering and not a hard quantized decoder input. With front-layer protection, the watermarked latent is

$$z_{\text{wm}} = z_{\text{front}} + (z_{\text{back}} + \alpha\Delta)\,. \tag{8}$$

In the three principal regimes, the predicted perturbation is projected away from the protected front component before being added. For notational consistency with Fig. 1, Δ denotes this usable residual after any such projection. Protection therefore removes the residual's component along selected continuous RVQ directions; it does not freeze codebook indices after a later re-quantization.

3.5 Codec-like Reconstruction and Detection

The decoder synthesizes the watermarked waveform as

$$\hat{x} = G_\theta(z_{\text{wm}}). \tag{9}$$

Because the waveform is generated from a modified continuous latent representation, the perceptual quality is influenced both by the watermark perturbation and by the reconstruction capacity of the encoder-decoder itself.

For detection, the possibly attacked waveform is re-encoded:

$$z' = \tanh(E_\theta(\mathcal{A}(\hat{x}))). \tag{10}$$

A seven-layer convolutional latent detector processes z', flattens the resulting feature map, and uses a linear head to produce K utterance-level message logits. We report bit accuracy. Bit accuracy denotes **aggregate bit recovery rate**, i.e., 1-BER, averaged over all payload bits and utterances.

3.6 Training Pipeline

Training follows three stages. During steps 0–60k, the detector uses proxy latent detection, while waveform attacks and adversarial training are disabled. Steps 60k–90k enable waveform decoding, re-encoding, and adversarial training without attack augmentation. After 90k, the complete attack-aware waveform path is optimized, forcing the detector to recover the message from z' after the bounded re-encoding operation defined above.

The focused and heavy EnCodec-aware regimes use a straight-through estimator [2]: the forward pass exposes the detector to codec-reconstructed audio, while the backward pass approximates the codec transform with an identity gradient. Specifically, $x_{\text{STE}} = \hat{x} + \text{sg}(x_{\text{codec}} - \hat{x})$, where sg denotes stop-gradient. The precise attack distributions of the three regimes are given in Section 4.3.

3.7 Loss Function

The generator is trained with reconstruction, perceptual, latent-consistency, adversarial, RVQ, and message losses:

$$\begin{aligned}\mathcal{L}_G =& \lambda_{\text{mel}}\mathcal{L}_{\text{mel}} + \lambda_{\text{stft}}\mathcal{L}_{\text{MRSTFT}} + \lambda_{\text{tf}}\mathcal{L}_{\text{TF}} + \lambda_{\text{adv}}\mathcal{L}_{\text{adv}} \\ &+ \lambda_{\text{sim}}\|z_{\text{wm}} - z\|_2^2 + \lambda_{\text{lat}}\|z' - z_{\text{wm}}\|_2^2 \\ &+ \lambda_{\text{msg}}\mathcal{L}_{\text{BCE}}(D_\phi(z'), m) + \lambda_{\text{vq}}\mathcal{L}_{\text{RVQ}}.\end{aligned} \tag{11}$$

The reconstruction term $\mathcal{L}_{\text{mel}}$ is a multi-scale mel-spectrogram loss. It compares the original and reconstructed waveforms at multiple FFT scales using both linear mel distance and log-mel distance. This term provides a broad spectral constraint and is the main reconstruction loss used throughout training.

The multi-resolution STFT loss $\mathcal{L}_{\text{MRSTFT}}$ is the average ℓ_1 distance between log-magnitude STFTs of x and $\hat{x}$ at three (n_{fft}, h, w) settings: $(1024, 120, 600)$, $(2048, 240, 1200)$, and $(512, 50, 240)$. This loss penalizes spectral artifacts not fully captured by mel features.

The time-frequency loudness ratio loss $\mathcal{L}_{\text{TF}}$ splits the waveform into frequency bands and short temporal frames, then compares the loudness of the reconstruction error with the loudness of the reference signal. It gives larger weight to distortions that are more likely to be perceptually salient. The adversarial term $\mathcal{L}_{\text{adv}}$ is computed with multi-period and multi-scale discriminators [10], including both generator adversarial loss and feature matching.

The message loss is binary cross-entropy. During early training, a proxy branch applies BCE to the detector output from the watermarked latent. During later training, the real branch applies the $\mathcal{L}_{\text{BCE}}$ term shown in the objective above after waveform reconstruction, attack simulation, and re-encoding. The latent-consistency loss encourages the attacked and re-encoded latent to remain close to the watermarked latent, while $\mathcal{L}_{\text{sim}}$ limits deviation from the original latent for perceptual preservation. In implementation, z_{wm} is a stop-gradient target for the latent-consistency term, and z is a stop-gradient target for the similarity term; this preserves the comparisons shown in Fig. 1 while making the gradient direction explicit. $\mathcal{L}_{\text{RVQ}}$ contains the standard codebook and commitment losses for stabilizing the guidance quantizers. The three principal regimes share these objective terms and scalar coefficients; their EnCodec route, gradient, and sampling differences are described below.

3.8 Backbone Controls

The main method uses the SEANet-style encoder-decoder. To analyze whether perceptual quality is limited by watermark perturbation or by the reconstruction backbone itself, we also run diagnostic backbone controls. One control trains the same SEANet-style autoencoder without watermarking and without RVQ/codebook loss. A second control replaces the SEANet-style encoder and decoder with a trainable DAC-style residual convolutional autoencoder [11] while keeping the same latent dimensionality and downsampling factor. This DAC-style model is not the pretrained DAC codec; it is a local backbone-control architecture. These DAC-related experiments are diagnostic controls and are not treated as the main watermarking method.

4 Experiments

4.1 Experimental Setup

Training uses speech from the English subset of Emilia, an extensive multilingual and diverse speech dataset for large-scale speech generation [8]. Final paper-facing evaluation uses 1000 fixed utterances sampled from the separate official Emilia `EN_B00001` archive; the training manifest uses `EN_B00000`. The evaluation set contains 1000 unique paths from 789 source groups, and exact utterance-ID overlap with the training manifest is zero. Audio is resampled to 48 kHz and cropped or padded to a maximum length of 120000 samples. The payload length is $K = 32$ bits. Unless otherwise stated, the main SEANet backbone uses $C = 128$ latent channels and downsampling ratios $[8, 5, 4, 2]$, resulting in a total downsampling factor of 320. The main RVQ-guided variants use four RVQ groups with a codebook size of 1024 and protect the first RVQ residual component from direct watermark modification.

4.2 Evaluation Protocol and Metrics

For watermark robustness evaluation, a payload is embedded into each speech segment, the watermarked waveform is optionally transformed by an attack $\mathcal{A}$, and the detector predicts the payload from the attacked waveform. We report bit accuracy as detailed in Section 3.5. BCE is used as an optimization diagnostic during training, whereas final robustness comparisons use bit accuracy.

Perceptual quality is evaluated primarily with PESQ [9] and ViSQOL v3 in audio mode with its SVR model [4]. We also report STOI [20] for intelligibility and SNR/SI-SDR [12] as waveform-distortion references, while recognizing that the latter do not always align with perceived quality. PESQ/STOI inputs are resampled to their required rate before scoring, following the metric practice of MaskMark [15]. Informal listening checks are not treated as quantitative evidence.

The attack suite contains common signal-processing attacks and neural-codec attacks. The signal-processing group includes additive noise, sample suppression, low-pass filtering, high-pass filtering, band-pass filtering, resampling, amplitude scaling, quantization, smoothing, pink noise, reverberation, time stretching, random phase shift, spectrogram augmentation, and segment replacement. The neural-codec evaluation includes EnCodec, the official pretrained DAC 44.1 kHz model, and released HiFi-Codec 24 kHz/240d and 24 kHz/320d models. DAC and HiFi-Codec are absent from training and are therefore zero-shot cross-codec conditions. For EnCodec, we report two reconstruction conditions. The main codec condition uses the official 24 kHz EnCodec model and is denoted EnCodec-24k. Two legacy 24 kHz route labels were evaluated during development, but the wrapper does not change the official model bandwidth between them and therefore produces the same reconstructed waveform. We merge them into one EnCodec-24k result rather than presenting a bitrate sweep. We additionally retain an EnCodec-16k condition, which first resamples the audio to 16 kHz before reconstruction by the same official 24 kHz model and then resamples it back through the 16 kHz path.

All final performance comparisons use the same fixed external utterances and deterministically generated message pairs. Codec outputs are resampled back to 48 kHz and aligned to the reference before detection and quality scoring.

4.3 Training Regimes

The principal SEANet-based checkpoints use the same latent embedding architecture: z_q guides the watermark embedder, the first RVQ residual component is protected from direct modification, and the decoder receives a continuous watermarked latent. Their difference is therefore procedural rather than architectural.

The balanced training regime uses the default attack sampler and does not make EnCodec-24k the dominant condition. When an EnCodec attack is selected, the sampler includes both the 16 kHz resampling route and the 24 kHz EnCodec routes; this reference regime does not apply an EnCodec-specific straight-through gradient. The EnCodec-24k-focused regime keeps the same fine-tuning loss coefficients, restricts the EnCodec branch to the 24 kHz routes, and applies the straight-through estimator defined above to that branch. The EnCodec-heavy endpoint keeps the same coefficients, 24 kHz routes, and estimator, but further biases the attack sampler toward EnCodec relative to clean/origin examples. Thus, these regimes differ in the EnCodec training procedure rather than in the latent architecture or scalar loss weights.

In implementation, the attack sampler assigns each candidate attack an unnormalized initial weight and normalizes the weights into probabilities. The EnCodec-24k-focused branch restricts the EnCodec choice to two equally weighted 24 kHz routes. With the

default attack list, its *initial* EnCodec probability is $1/17 = 5.9\%$. In the EnCodec-heavy endpoint, EnCodec receives initial weight 2.5, clean/origin receives 0.8, and the other attacks receive 1.0, giving an initial EnCodec probability of $2.5/18.3 = 13.7\%$. During fine-tuning, both regimes periodically update attack probabilities from observed per-attack message losses, while retaining these regime-specific bias weights. Thus, 5.9% and 13.7% describe initialization rather than fixed exposure rates; the scalar loss coefficients remain unchanged.

Checkpoints were saved every 6k steps. For each regime, we selected a quality–robustness operating point by inspecting preliminary 100-sample checkpoint trajectories rather than simply reporting the last checkpoint. The external `EN_B00001` evaluation set was not used for checkpoint selection. All final performance metrics were then recomputed on its fixed 1000-utterance protocol.

The DAC-style full-pipeline probe is reported separately as a backbone-control experiment: it keeps the watermarking pipeline fixed but replaces the SEANet autoencoder with a trainable DAC-style autoencoder.

4.4 Reconstruction Diagnostics

We separate the effect of the inference-time watermark residual from the reconstruction behavior of the jointly trained autoencoder. For a trained watermarking model, the *no-embed* condition removes the watermark delta at inference time:

$$x_{\text{noembed}} = G_\theta(\tanh(E_\theta(x))). \tag{12}$$

Because the encoder and decoder were jointly optimized with watermark and robustness objectives, this is a local diagnostic rather than a pure autoencoder ceiling. On the fixed external 1000-utterance set, balanced training obtains PESQ 3.588 without the residual and 3.578 with it, a difference of only 0.010. For EnCodec-24k-focused training, removing the residual instead reduces PESQ from 3.368 to 3.259. Thus, the inference-time residual alone does not determine perceptual quality, and removing it is not guaranteed to improve reconstruction. The watermark-free backbone probes below isolate standalone reconstruction behavior.

4.5 Codec Reconstruction Reference

We also pass clean audio through neural codecs without watermarking to contextualize the quality of attacked audio. On the fixed external set, clean audio obtains PESQ 4.644. EnCodec-24k, DAC, HiFi-Codec-24k/240d, and HiFi-Codec-24k/320d reconstruction obtain PESQ 3.529, 3.739, 2.509, and 2.462, respectively. These values reference the specific evaluated wrappers rather than intrinsic codec upper bounds. As described above, EnCodec-24k is a fixed reconstruction condition rather than a bitrate sweep.

4.6 Robustness Results

Table 1 summarizes the quality-robustness trade-off caused by increasing the EnCodec-24k emphasis during training. All accuracy columns report bit accuracy. Balanced training achieves the highest clean perceptual quality but lower EnCodec-24k bit accuracy. EnCodec-24k-focused training substantially improves EnCodec-24k robustness and is therefore used as the main codec-aware setting.

Table 1: Quality and codec robustness trade-off on the fixed 1000-utterance external subset.

Variant	Step	PESQ	ViSQOL	Clean Acc.	EnCodec-24k Acc.
Balanced training	258k	3.578	4.574	99.7%	78.5%
EnCodec-24k-focused training	258k	3.368	4.541	99.9%	95.7%
EnCodec-heavy endpoint	222k	3.286	4.520	99.9%	97.0%

Table 2: Cross-codec bit accuracy on 1000 fixed samples. DAC and HiFi-Codec are zero-shot conditions for the three proposed checkpoints. AudioSeal uses its released 16-bit model and is a codec-sensitivity reference, not a direct 32-bit system ranking.

System	Bits	EnCodec-24k	DAC-44k	HiFi-240d
Balanced	32	78.5%	78.1%	50.3%
EnCodec-24k-focused	32	95.7%	95.7%	51.6%
EnCodec-heavy	32	97.0%	96.7%	52.2%
AudioSeal (released)	16	95.9%	99.6%	50.5%

Both PESQ and ViSQOL decrease as codec emphasis increases, although the ViSQOL differences are smaller. The EnCodec-heavy endpoint provides a further robustness gain at an additional quality cost. For balanced, focused, and heavy training, respectively, STOI is 0.987, 0.984, and 0.983; SNR is 6.27, 5.96, and 5.88 dB; and SI-SDR is 4.92, 4.55, and 4.44 dB. These waveform-level diagnostics follow the same quality trend.

Under the 15 non-codec transformations, the EnCodec-24k-focused checkpoint retains at least 98.1% bit accuracy. The lowest results are reverberation and phase shift, at 98.1% and 98.2%, respectively; the other conditions range from 99.0% to 99.9%. The separately defined EnCodec-16k route is retained as a sample-rate resampling-plus-codec condition, not interpreted as either a bitrate variant or evidence of a universally stronger codec attack.

4.7 Cross-Codec Transfer

Table 2 separates matched EnCodec-24k robustness from zero-shot transfer to architectures absent from training. EnCodec-aware training transfers strongly to the tested DAC path: the focused and heavy checkpoints obtain 95.7% and 96.7%. In contrast, all three proposed checkpoints remain near the 50% chance level under HiFi-Codec-240d. The 320d configuration gives corresponding results of 50.2%, 50.9%, and 51.2%. Both tables use the same fixed external utterance/message pairs and codec wrapper. The evidence therefore supports transfer to the tested DAC implementation, not codec-invariant recovery.

4.8 Backbone Diagnostics

On the fixed external set, watermark-free SEANet and DAC-style autoencoders obtain PESQ 3.646 and 3.695, respectively. The slightly stronger standalone DAC-style reconstruction does not automatically improve the full watermarking system: replacing the backbone yields PESQ 3.270, clean bit accuracy 95.5%, and EnCodec-24k bit accuracy 71.1%. This is a compatibility diagnostic, not a judgment that DAC-style backbones are intrinsically worse. It shows that plain reconstruction capacity is not sufficient; the backbone

must remain compatible with embedding, detection, and attack-aware training.

5 Limitations

This work investigates continuous latent-space embedding rather than claiming universal neural-codec robustness. EnCodec-aware training transfers to the tested DAC implementation but not to the two HiFi-Codec configurations, so generalization to unseen codec families remains open. We also do not embed messages into discrete codec tokens or RVQ indices; the carrier remains a continuous pre-decoder latent. Finally, quality is assessed with objective full-reference metrics and informal listening checks rather than a formal subjective study.

6 Conclusion

This paper investigates watermarking in continuous pre-decoder latents. EnCodec-aware training raises matched EnCodec-24k bit accuracy from 78.5% to 95.7% and 97.0%, with a measurable perceptual cost. Transfer to the tested DAC path but not HiFi-Codec demonstrates codec-dependent rather than universal transfer. Backbone controls further show that stronger standalone reconstruction is insufficient; the backbone must remain compatible with embedding, detection, and attack-aware training.

7 Generative AI Use Disclosure

Large Language Models (LLMs) were used only for manuscript polishing, such as rephrasing and grammar checks. They were not used for ideation, methodology, experimental design, data analysis, or result interpretation. All scientific content was produced and verified by the authors.

Acknowledgments

This research is funded in part by the Guangdong Science and Technology Plan (2023A1111120012).

References

[1] Walter Bender, Daniel Gruhl, Norishige Morimoto, and Anthony Lu. 1996. Techniques for data hiding. In *IBM Systems Journal*, Vol. 35. 313–336.

[2] Yoshua Bengio, Nicholas Leonard, and Aaron Courville. 2013. Estimating or propagating gradients through stochastic neurons for conditional computation. In *arXiv preprint arXiv:1308.3432*.

[3] Guangyu Chen, Yu Wu, Shujie Liu, Tao Liu, Xiaoyong Du, and Furu Wei. 2023. WavMark: Watermarking for audio generation. *arXiv preprint arXiv:2308.12770* (2023).

[4] Michael Chinen, Felicia S. C. Lim, Jan Skoglund, Nikita Gureev, Feargus O'Gorman, and Andrew Hines. 2020. ViSQOL v3: An Open Source Production Ready Objective Speech and Audio Metric. *arXiv preprint arXiv:2004.09584* (2020).

[5] Alexandre Défossez, Jade Copet, Gabriel Synnaeve, and Yossi Adi. 2022. High fidelity neural audio compression. *arXiv preprint arXiv:2210.13438* (2022).

[6] Daniel Gruhl, Anthony Lu, and Walter Bender. 1996. Echo hiding. In *Information Hiding*. Springer, 295–315.

[7] Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, and Ruoming Pang. 2020. Conformer: Convolution-augmented transformer for speech recognition. In *Interspeech*. 5036–5040.

[8] 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. 2024. Emilia: An extensive, multilingual, and diverse speech dataset for large-scale speech generation. *arXiv preprint arXiv:2407.05361* (2024).

[9] ITU-T. 2001. ITU-T Recommendation P.862: Perceptual evaluation of speech quality (PESQ), an objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs. International Telecommunication Union.

[10] Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae. 2020. HiFi-GAN: Generative adversarial networks for efficient and high fidelity speech synthesis. In *Advances in Neural Information Processing Systems*, Vol. 33. 17022–17033.

[11] Rithesh Kumar, Prem Seetharaman, Alejandro Luebs, Ishaan Kumar, and Kundan Kumar. 2023. High-fidelity audio compression with improved RVQGAN. *arXiv preprint arXiv:2306.06546* (2023).

[12] Jonathan Le Roux, Scott Wisdom, Hakan Erdogan, and John R. Hershey. 2019. SDR: Half-baked or well done?. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 626–630.

[13] Haiyun Li, Zhiyong Wu, Xiaofeng Xie, Jingran Xie, Yaoxun Xu, and Hanyang Peng. 2025. VoiceMark: Zero-shot voice cloning-resistant watermarking approach leveraging speaker-specific latents. *arXiv preprint arXiv:2505.21568* (2025).

[14] Linxi Li, Liwei Jin, Yechen Wang, Houmin Sun, Zi Hu, and Carsten Maple. 2026. AURA: A Stegaformer-Based Scalable Deep Audio Watermark with Extreme Robustness. In *ICASSP 2026 - 2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 16522–16526. doi:10.1109/ICASSP55912.2026.11462806

[15] Patrick O'Reilly, Zeyu Jin, Jiaqi Su, and Bryan Pardo. 2024. Maskmark: Robust Neuralwatermarking for Real and Synthetic Speech. In *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 4650–4654. doi:10.1109/ICASSP48485.2024.10447253

[16] Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron C. Courville. 2017. FiLM: Visual Reasoning with a General Conditioning Layer. *arXiv preprint arXiv:1709.07871* (2017).

[17] Yi Ren, Chenxu Hu, Xu Tan, Tao Qin, Sheng Zhao, Zhou Zhao, and Tie-Yan Liu. 2021. FastSpeech 2: Fast and high-quality end-to-end text to speech. In *International Conference on Learning Representations*.

[18] Robin San Roman, Pierre Fernandez, Alexandre Défossez, Teddy Furon, Tuan Tran, and Hady Elsahar. 2024. Proactive detection of voice cloning with localized watermarking. In *International Conference on Machine Learning*.

[19] Mayank Kumar Singh, Naoya Takahashi, Weihsiang Liao, and Yuki Mitsufuji. 2024. SilentCipher: Deep audio watermarking. *arXiv preprint arXiv:2406.03822* (2024).

[20] Cees H. Taal, Richard C. Hendriks, Richard Heusdens, and Jesper Jensen. 2011. An algorithm for intelligibility prediction of time-frequency weighted noisy speech. In *IEEE Transactions on Audio, Speech, and Language Processing*, Vol. 19. 2125–2136.

[21] Chengyi Wang, Sanyuan Chen, Yu Wu, Ziqiang Zhang, Long Zhou, Shujie Liu, Zhuo Chen, Yanqing Liu, Huaming Wang, Jinyu Li, Lei He, Sheng Zhao, and Furu Wei. 2023. Neural codec language models are zero-shot text to speech synthesizers. *arXiv preprint arXiv:2301.02111* (2023).

[22] Dongchao Yang, Songxiang Liu, Rongjie Huang, Jinchuan Tian, Chao Weng, and Yuexian Zou. 2023. HiFi-Codec: Group-residual vector quantization for high fidelity audio codec. *arXiv preprint arXiv:2305.02765* (2023).

[23] Neil Zeghidour, Alejandro Luebs, Ahmed Omran, Jan Skoglund, and Marco Tagliasacchi. 2021. SoundStream: An end-to-end neural audio codec. In *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, Vol. 30. 495–507.

[24] Xin Zhang, Dong Zhang, Shimin Li, Yaqian Zhou, and Xipeng Qiu. 2023. Speech-Tokenizer: Unified speech tokenizer for speech large language models. *arXiv preprint arXiv:2308.16692* (2023).