Multi-bit Audio Watermarking for Music

Luca A. Lanzendörfer **Kyle Fearne** **Florian Grötschla** **Roger Wattenhofer**

ETH Zurich

{lanzendoerfer, kfearne, fgroetschla, wattenhofer}@ethz.ch

Abstract

We present Timbru, a post-hoc audio watermarking model that achieves state-of-the-art robustness and imperceptibility trade-offs without training an embedder-detector model. Given any 44.1 kHz stereo music snippet, our method performs per-audio gradient optimization to add imperceptible perturbations in the latent space of a pretrained audio VAE, guided by a combined message and perceptual loss. The watermark can then be extracted using a pretrained CLAP model. We evaluate 16-bit watermarking on MUSDB18-HQ against AudioSeal and WavMark across common filtering, noise, compression, resampling, cropping, and regeneration attacks. Our approach attains the best average bit error rates, while preserving perceptual quality, demonstrating an efficient, dataset-free path to imperceptible audio watermarking.

1 Introduction

Audio watermarking embeds imperceptible, machine-verifiable signals into audio to support provenance, attribution, and copyright protection. This capability is increasingly critical in the era of social media and rapidly improving generative models, which enable the production and dissemination of highly realistic synthetic audio. Reliable watermarking can help end-users verify the legitimacy of clips, deter unauthorized sampling, and credit creators, while simultaneously raising the stakes for adversaries who seek to remove or forge watermarks.

Historically, audio watermarking was largely based on empirical schemes such as Quantization Index Modulation [1], patchwork algorithms [2], least significant bit embedding [3], and spread-spectrum techniques [4]. Although effective in certain settings, these methods often fail under common transformations such as audio compression. The trade-off between watermark imperceptibility and robustness against attacks remains at the center of audio watermarking and motivates our work.

Recent learning-based approaches have made significant progress, spanning passive detectors [5, 6] and joint embedded-detector architectures [7–11] trained end-to-end. Passive detection is becoming increasingly less effective due to high-fidelity synthetic audio that closely mimics genuine content. In general, current watermarking approaches can be further categorized into ad-hoc and post-hoc methods. Ad-hoc models integrate watermarking within a generator to emit user- or model-specific watermarks [12]; post-hoc methods watermark arbitrary inputs after the fact. The latter offers greater flexibility and accessibility, enabling users to protect existing and novel content alike.

In this work, we propose Timbru, a post-hoc optimization-based method that performs gradient updates on a single audio snippet, by perturbing the audio imperceptibly until a watermark is obtained that is robust to a wide range of attacks. This eliminates the compute and data requirements of training dedicated embedder-detector models and does not necessitate domain-specific fine-tuning for speech, music, or environmental audio.

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: AI for Music.

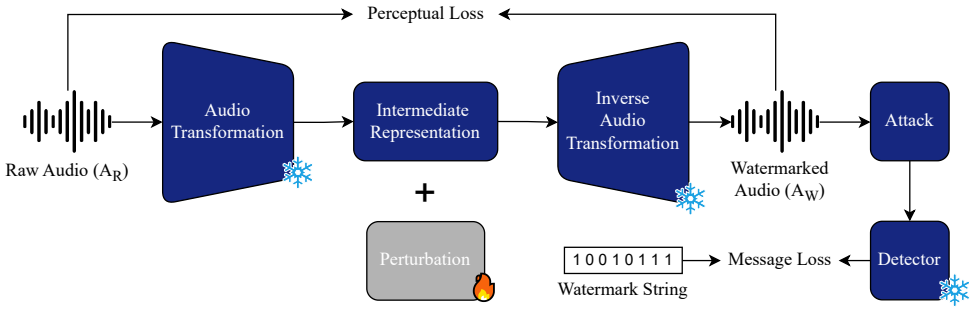

Figure 1: Overview of our proposed approach. The raw waveform A_R is first transformed into the latent representation using a pretrained Stable Audio Open VAE. To embed a watermark, minor perturbations are added to this intermediate representation. At every step, this representation is decoded back into a waveform (A_W) and then augmented to simulate a variety of attacks. The perceptual loss and the message loss from the decoded message are then used to calculate the gradient which optimizes the perturbations. All other components remain frozen.

Our contributions can be summarized as follows. We propose a post-hoc audio watermarking approach for 44.1 kHz stereo audio. Our approach encodes the audio using a pretrained Stable Audio Open VAE [13], which is then perturbed using gradient optimization to obtain an imperceptible watermark. To detect a watermark and its payload, we use a pretrained CLAP [14] model as the feature extractor for watermark detection. We find that our approach is on average more robust to attacks and achieves higher perceptual quality compared to previous state-of-the-art methods.

2 Methodology

The core idea behind Timbru is that perturbations are added to a latent representation of the audio during the optimization process in order to embed a watermark string of k bits $m = m_1, ...m_k$ into the audio snippet, as shown in Fig. 1. The purpose of such perturbations is to modify the audio's features in a way that aligns with a secret key held by the user [15, 16]. In a multi-bit setting, each user has a *secret key* consisting of k randomly selected orthogonal vectors. Each vector $v_1, ..., v_k$ corresponds to an encoded bit. During the optimization process, the message m is modulated into the signs of the projection of the features extracted by a pretrained CLAP model, $\phi(A_W)$, against each of the carriers. The detector component then retrieves $\tilde{m}$ as follows:

$$\tilde{m} = [sign(\phi(A_W)^\top v_1), ..., sign(\phi(A_W)^\top v_k)] \tag{1}$$

Training Pipeline. Audio waveform snippets A_R are passed through a transformation stage $T(\cdot)$ in order to extract an embedding space within which to embed the watermark. We define $T(\cdot)$ to be passing the waveform through the Stable Audio Open VAE [13] such that the intermediary representation can be written as

$$A_I = T(A_R) = Enc(A_R) \tag{2}$$

Small perturbations δ_m are then added to the intermediary representation A_I and the inverse transformation is applied to convert the latent back to a raw audio waveform such that the resultant watermarked audio is:

$$A_W = T^{-1}(A_I + \delta_m) = Dec(A_I + \delta_m) \tag{3}$$

During the optimization stage, before detecting the watermark in the audio snippet, the watermarked audio A_W is subjected to a random attack to introduce robustness. The attacked audio is then passed through a detector and the message is retrieved. The loss, composed of both the perceptual loss between A_R and A_W and the message loss is then calculated and the gradient propagated back to A_I, which acts as a perturbation δ_w added inside the latent space.

Losses. To capture robustness and the ability to detect and decode a watermark, we use a message loss [16]. The optimization objective is to align the audio features x as closely as possible to the

Table 1: Results for 16-bit watermarking. We compare Timbru against AudioSeal (AS) and WavMark (WM) in terms of bit error rate (lower is better). We evaluate the watermarking models on bandpass (BP), lowpass (LP), highpass (HP), echo (E), smoothing (S), duck audio (DA), boost audio (BA), gaussian noise (GN), pink noise (PN), resampling (RS), quantization (Q), sample suppression (SS), random cropping (RC), EnCodec re-encoding (EnC.), and regeneration attack (Regen.). More details on each attack can be found in Appendix A. Whilst each method demonstrates their own clear advantages and disadvantages, on average, our method demonstrates the best average bit error rate, and notably outperforms previous methods on unseen regeneration attacks.

Model	**None**	**BP**	**LP**	**HP**	**E**	**S**	**DA**	**BA**	**GN**	**PN**
AS [9]	1.58	**1.75**	**41.00**	61.13	**2.63**	5.25	1.58	1.54	**9.54**	1.63
WM [10]	**0.55**	2.58	49.92	**0.64**	14.75	4.16	**0.55**	0.54	48.90	**0.95**
Timbru	0.83	17.5	53.30	25.00	22.5	**0.00**	0.83	**0.42**	20.42	2.5

Model	**MP3**	**AAC**	**RS**	**Q**	**SS**	**RC**	**Speed**	**EnC.**	**Regen.**	**Avg.**
AS [9]	**1.79**	42.83	1.58	1.75	**2.50**	42.92	43.83	**6.96**	66.46	17.79
WM [10]	11.05	**10.44**	**0.55**	**1.23**	32.35	43.22	50.30	49.37	49.24	19.54
Timbru	5.42	22.08	0.83	1.67	6.67	**30.83**	**40.00**	10.41	**21.67**	**14.89**

k vectors $v_1, ..., v_k$ that correspond to the encoded message. The message loss is a hinge loss with margin $\mu > 0$ on the projections, defined as

$$L_m(A_W) = \frac{1}{K} \sum_{k=1}^{K} max(0, \mu - (x^\top v_i).m_i), \tag{4}$$

where $m = (m_1, ...m_k) \in \{-1, 1\}_k$ is the hidden message we embed in the audio snippet.

Additionally, a perceptual loss is used to ensure that any perturbations added to the audio remain imperceptible to humans. This perceptual loss, L_p, is taken from DAC [17] and consists of a combination of different losses, including a multi-scale Mel Spectrogram loss, as well as an adversarial discriminator loss. The total loss is therefore

$$L = \lambda_m L_m + \lambda_p L_p, \tag{5}$$

where $\lambda_m = 160$ and $\lambda_p = 4$ were empirically chosen as the optimal message weight and perceptual weight, respectively.

3 Experiments

In line with previous work [10, 9], we embed 16 bits as our watermark message payload. We randomly pick 10% of MUSDB18-HQ [18] mixtures and crop out 10-second snippets of these samples to evaluate the methods. We test the robustness of our approach against a variety of attacks using bit error rate metric to measure watermark message retrieval accuracy. In addition, we use ViSQOL [19] and SI-SNR [20] to measure objective perceptual quality, as well as conducting a MUSHRA [21] human evaluation study with 10 participants,[1] where each participant was asked to score all watermarked audios on perceptual quality. The subjective perceptual study contained one hidden reference, and two anchors (3.5 kHz, 7 kHz) as well as three stimuli (Timbru, WavMark, and AudioSeal). The participants were briefed beforehand about the task and were asked to rate the perceptual quality of each stimuli.

The bit error rates for each attack are shown in Table 1. While we observe that our method is on average able to outperform both AudioSeal [9] and Wavmark [10] in message reconstruction accuracy, it is evident that each proposed watermarking approach has its own strengths and weaknesses. For example, Audioseal proves to be robust against sample suppression due to its sample-level localization techniques that implement sample-level masking during training. Bandpass and Lowpass results show that AudioSeal also demonstrates its strength by not encoding a watermark in the low-frequency

[1] MUSHRA was conducted on `https://www.mabyduck.com`

Table 2: Results for perceptual audio quality for 16-bit watermarking. We evaluate perceptual audio quality on ViSQOL, SI-SNR, and by conducting a MUSHRA human evaluation study. For ViSQOL and SI-SNR we show the standard deviation and for MUSHRA the 95% confidence interval. We find that overall, our gradient-based optimization approach leads to the least impact on perceptual quality.

	ViSQOL ↑	**SI-SNR (dB) ↑**	**MUSHRA ↑**
AudioSeal [9]	1.91±0.54	19.65±6.18	44.92±2.92
WavMark [10]	1.91±0.53	**23.03±5.16**	44.74±3.14
Timbru (ours)	**4.08±0.25**	5.15±3.13	**70.83±3.53**

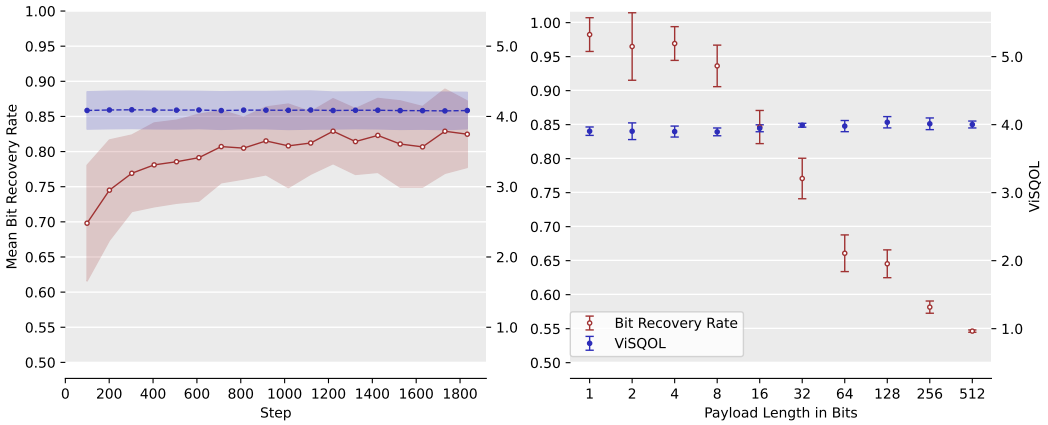

Figure 2: (Left) Mean bit recovery rate ($\mathrm{BRR} = (1-\mathrm{BER})/100$) for 16-bit payload over optimization steps shows the longer we run Timbru, the more robust the embedded watermark becomes. (Right) Ablation where each point represents the mean BRR for watermarked audio with specific payload length, showing how the mean BRR and the perceptual quality change as the payload length increases.

or high-frequency domain, unlike Wavmark, which tends to encode its watermarks in the high frequencies. Furthermore, it is interesting to note that, compared to AudioSeal and WavMark, Timbru offers the best robustness against unseen regeneration attacks, which tend to be the most difficult attack type to defend against. The regeneration attack is carried out by encoding and decoding the audio with DAC [17]. Since our approach embeds perturbations in the latent space of a pretrained VAE which was originally trained against a loss containing Mel components, we believe that the watermark is more likely to be preserved compared to other approaches. The bit error rates for Wavmark were obtained by extracting only the last 16 bits corresponding to the payload.

Analyzing the watermarked audio quality in Table 2, we find that our approach offers better general audio quality as measured by ViSQOL and the human participants in the MUSHRA listening study. For the MUSHRA study, the participants rated the reference, mid-anchor (7 kHz), and low anchor (3.5 kHz) as 91.37±1.97, 50.85±3.25, and 21.62±2.92, respectively. The significantly lower performance of Timbru in terms of SI-SNR can be explained because the audio is passed through a VAE, which can cause a variety of signal-level artifacts that are imperceptible to humans (e.g., sample mismatch, phase inversion). In Fig. 2 we show the performance of Timbru in terms of optimization steps. Unsurprisingly, we find that the longer we optimize, the more robust the watermark becomes, although there are diminishing returns after a few thousand steps. For our experiments, we set a stopping condition if the bit recovery rate does not improve for 1k steps. On average, the watermarking process takes roughly one hour per audio snippet. Furthermore, we show the trade-off between the number of bits in the payload and the corresponding ViSQOL score. We find that as the number of bits increases, the robustness against attacks tends to degrade.

Conclusion. We introduced Timbru, a post-hoc audio watermarking method that preserves perceptual quality while improving robustness by performing per-snippet gradient optimization to embed small perturbations in a latent representation of audio, offering a strong dataset-free alternative to state-of-the-art watermarking approaches.

References

[1] B. Chen and G. W. Wornell, "Quantization index modulation: A class of provably good methods for digital watermarking and information embedding," *IEEE Transactions on Information Theory*, vol. 47, no. 4, p. 1423, 2001.

[2] H. Kang, K. Yamaguchi, B. Kurkoski, K. Yamaguchi, and K. Kobayashi, "Full-index-embedding patchwork algorithm for audio watermarking," *IEICE-Transactions on Information and Systems*, vol. 91, no. 11, pp. 2731–2734, 2008.

[3] N. Cvejic and T. Seppänen, "Increasing robustness of lsb audio steganography using a novel embedding method," in *Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'04) Volume 2-Volume 2*, 2004, p. 533.

[4] I. Cox, J. Kilian, F. Leighton, and T. Shamoon, "Secure spread spectrum watermarking for multimedia," *IEEE Transactions on Image Processing*, vol. 6, no. 12, pp. 1673–1687, 1997.

[5] L. Guarnera, O. Giudice, and S. Battiato, "Level up the deepfake detection: a method to effectively discriminate images generated by gan architectures and diffusion models," in *Intelligent Systems Conference*. Springer, 2024, pp. 615–625.

[6] U. Ojha, Y. Li, and Y. J. Lee, "Towards universal fake image detectors that generalize across generative models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 24 480–24 489.

[7] K. Pavlović, S. Kovačević, I. Djurović, and A. Wojciechowski, "Robust speech watermarking by a jointly trained embedder and detector using a dnn," *Digital Signal Processing*, vol. 122, p. 103381, 2022.

[8] C. Liu, J. Zhang, H. Fang, Z. Ma, W. Zhang, and N. Yu, "Dear: A deep-learning-based audio re-recording resilient watermarking," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 11, 2023, pp. 13 201–13 209.

[9] R. S. Roman, P. Fernandez, H. Elsahar, A. Défossez, T. Furon, and T. Tran, "Proactive detection of voice cloning with localized watermarking," in *Proceedings of the 41st International Conference on Machine Learning*, 2024, pp. 43 180–43 196.

[10] G. Chen, Y. Wu, S. Liu, T. Liu, X. Du, and F. Wei, "Wavmark: Watermarking for audio generation," *arXiv preprint arXiv:2308.12770*, 2023.

[11] M. K. Singh, N. Takahashi, W. Liao, and Y. Mitsufuji, "Silentcipher: Deep audio watermarking," *arXiv preprint arXiv:2406.03822*, 2024.

[12] R. San Roman, P. Fernandez, A. Deleforge, Y. Adi, and R. Serizel, "Latent watermarking of audio generative models," in *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2025, pp. 1–5.

[13] Z. Evans, J. D. Parker, C. Carr, Z. Zukowski, J. Taylor, and J. Pons, "Long-form music generation with latent diffusion," *arXiv preprint arXiv:2404.10301*, 2024.

[14] Y. Wu, K. Chen, T. Zhang, Y. Hui, T. Berg-Kirkpatrick, and S. Dubnov, "Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

[15] Y. Guo, R. Li, M. Hui, H. Guo, C. Zhang, C. Cai, L. Wan, and S. Wang, "Freqmark: invisible image watermarking via frequency based optimization in latent space," in *Proceedings of the 38th International Conference on Neural Information Processing Systems*, 2024, pp. 112 237–112 261.

[16] P. Fernandez, A. Sablayrolles, T. Furon, H. Jégou, and M. Douze, "Watermarking images in self-supervised latent spaces," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 3054–3058.

[17] R. Kumar, P. Seetharaman, A. Luebs, I. Kumar, and K. Kumar, "High-fidelity audio compression with improved rvqgan," *Advances in Neural Information Processing Systems*, vol. 36, pp. 27 980–27 993, 2023.

[18] Z. Rafii, A. Liutkus, F.-R. Stöter, S. I. Mimilakis, and R. Bittner, "Musdb18-hq - an uncompressed version of musdb18," Aug. 2019. [Online]. Available: https://doi.org/10.5281/zenodo.3338373

[19] A. Hines, J. Skoglund, A. Kokaram, and N. Harte, "Visqol: The virtual speech quality objective listener," in *IWAENC 2012; international workshop on acoustic signal enhancement*. VDE, 2012, pp. 1–4.

[20] Y. Luo and N. Mesgarani, "Tasnet: time-domain audio separation network for real-time, single-channel speech separation," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 696–700.

[21] B. Series, "Method for the subjective assessment of intermediate quality level of audio systems," *International Telecommunication Union Radiocommunication Assembly*, vol. 2, 2014.

[22] J. Copet, F. Kreuk, I. Gat, T. Remez, D. Kant, G. Synnaeve, Y. Adi, and A. Défossez, "Simple and controllable music generation," in *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[23] A. Défossez, J. Copet, G. Synnaeve, and Y. Adi, "High fidelity neural audio compression," *arXiv preprint arXiv:2210.13438*, 2022.

A Training Parameters

The attacks used for training and evaluation are a set of attacks that are common among other audio watermarking methods [9–11]. The majority of these attacks were performed through the Audiocraft library [22]. The parameters were mostly chosen as per the evaluation done in AudioSeal [9]. Parameters used during training are denoted by (T) whilst those used for evaluation are denoted by (E).

Bandpass Filter: Only allows a specific frequency band to pass through it. (T) Lower cut-off randomly selected between 300-5000Hz, upper cut-off between 3000-9000Hz. (E) Fixed window of 500-5000 Hz.

Lowpass Filter: Eliminates frequencies above a cut-off. (T) Cut-off randomly chosen between 300-5000 Hz .(E) Cut-off fixed at 500 Hz.

Highpass Filter: Eliminates frequencies below a cut-off. (T) Cut-off randomly chosen between 300-2000 Hz (E). Cut-off fixed at 1500 Hz.

Resampling: Resamples audio to: (T) Random choice between 4kHz, 8kHz, 16kHz, 22.1kHz, 24kHz, 32kHz. (E) Fixed sample rate of 32kHz.

Sample Suppression: Randomly sets a percentage of samples to 0. (T) Random choice between 0.01-0.05%. (E) Fixed at 0.03%.

Boost Audio: Amplifies audio by a factor. (T) Random factor between 1.1-12. (E) Fixed factor of 10.

Duck Audio: Reduces audio volume by a factor. (T) Random factor between 0.1-0.9. (E) Fixed factor of 0.1.

Random Crop: Random crop of original audio. (T) Length randomly chosen from 1s - original duration. (E) Length fixed to 50% of original.

AAC Compression: Encodes audio in AAC format. (T) Bitrate randomly chosen between 32kHz, 64kHz, 128kHz, 256kHz. (E) Fixed bitrate of 64kHz.

MP3 Compression: Encodes audio in MP3 format. (T) Bitrate randomly chosen between 32kHz, 64kHz, 128kHz, 256kHz. (E) Fixed bitrate of 32kHz.

Pink Noise: Adds pink noise. (T) Randomly selected standard deviation between 0.001 - 0.05. (E) Standard deviation fixed at 0.1.

Gaussian Noise: Adds Gaussian noise. (T) Randomly selected standard deviation between 0.001 - 0.05. (E) Standard deviation fixed at 0.05.

Echo: Applies an echo effect to the audio, adding a delay and less loud copy of the original. (T) random delay between 0.1 and 0.5 seconds, random volume between 0.1 and 0.5; (E) fixed delay of 0.5 seconds, fixed volume of 0.5.

Smooth: Smooths the audio signal using a moving average filter with a variable window size. (T) window size random between 2 and 10. (E) Window size fixed at 40.

Quantization: Quantizes the samples to a number of values. (T) Number of values randomly chosen out of 2^8, 2^9, 2^{10}. (E) Fixed number of 2^9.

Speed Change: Changes the speed of the original audio by a factor. (T) Factor randomly selected between 0.95-1.1. (E) Fixed factor of 1.25.

EnCodec: (T) Resamples at 24 or 32kHz, encodes the audio with corresponding EnCodec [23] version, and resamples it back to 44.1kHz. (E) Only uses EnCodec 24kHz and resamples back to 44.1kHz.

Regeneration: Out-of-domain attack. Only (E) where we encode with DAC [17] 44.1kHz.