Proactive Detection of Voice Cloning with Localized Watermarking

Robin San Roman [*1 2] **Pierre Fernandez** [*1 2] **Hady Elsahar** [*1]
Alexandre Défossez [3] **Teddy Furon** [2] **Tuan Tran** [1]

Abstract

In the rapidly evolving field of speech generative models, there is a pressing need to ensure audio authenticity against the risks of voice cloning. We present AudioSeal, the first audio watermarking technique designed specifically for localized detection of AI-generated speech. AudioSeal employs a generator / detector architecture trained jointly with a localization loss to enable localized watermark detection up to the sample level, and a novel perceptual loss inspired by auditory masking, that enables AudioSeal to achieve better imperceptibility. AudioSeal achieves state-of-the-art performance in terms of robustness to real life audio manipulations and imperceptibility based on automatic and human evaluation metrics. Additionally, AudioSeal is designed with a fast, single-pass detector, that significantly surpasses existing models in speed, achieving detection up to two orders of magnitude faster, making it ideal for large-scale and real-time applications. Code is available at github.com/facebookresearch/audioseal.

1. Introduction

Generative speech models are now capable of synthesizing voices that are indistinguishable from real ones (Arik et al., 2018; Kim et al., 2021; Casanova et al., 2022; Wang et al., 2023). Though speech generation and voice cloning are not novel concepts, their recent advancements in quality and accessibility have raised new security concerns. A notable incident occurred where a deepfake audio misleadingly urged US voters to abstain, showcasing the potential for misusing these technologies to spread false information (Murphy et al., 2024). Consequently, regulators and governments are implementing measures for AI content transparency and traceability, including forensics and watermarking – see Chi (2023); Eur (2023); USA (2023).

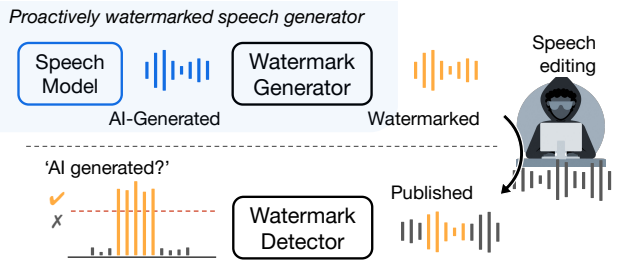

Figure 1. **Proactive detection of AI-generated speech.** We embed an imperceptible watermark in the audio, which can be used to detect if a speech is AI-generated and identify the model that generated it. It can also precisely pinpoint AI-generated segments in a longer audio with a sample level resolution (1/16k seconds).

The main forensics approach to detect synthesized audio is to train binary classifiers to discriminate between natural and synthesized audios, a technique highlighted in studies by Borsos et al. (2022); Kharitonov et al. (2023); Le et al. (2023). We refer to this technique as *passive detection* since it does not alter of the audio source. Albeit being a straightforward mitigation, it is prone to fail as generative models advance and the difference between synthesized and authentic content diminishes.

Watermarking emerges as a strong alternative. It embeds a signal in the generated audio, imperceptible to the ear but robustly detectable by specific algorithms. There are two watermarking types: multi-bit and zero-bit. Zero-bit watermarking detects the presence or absence of a watermarking signal, which is valuable for AI content detection. Multi-bit watermarking embeds a binary message in the content, allowing to link content to a specific user or generative model. Most deep-learning based audio watermarking methods (Pavlović et al., 2022; Liu et al., 2023a; Chen et al., 2023) are multi-bit. They train a generator to output the watermarked audio from a sample and a message, and an extractor retrieving the hidden message.

Current watermarking methods have limitations. First, *they are not adapted for detection*. The initial applications assumed any sound sample under scrutiny was watermarked (*e.g.* IP protection). As a result, the decoders were never

[*]Equal contribution [1]FAIR, Meta [2]Inria [3]Kyutai. Correspondence to: <robinsr, hadyelsahar, pfz@meta.com>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

trained on non-watermarked samples. This discrepancy between the training of the models and their practical use leads to poor or overestimated detection rates, depending on the embedded message (see App. B). Our method aligns more closely with the concurrent work by Juvela & Wang (2023), which trains a detector, rather than a decoder.

Second, they *are not localized* and consider the entire audio, making it difficult to identify small segments of AI-generated speech within longer audio clips. The concurrent WavMark's approach (Chen et al., 2023) addresses this by repeating at 1-second intervals a synchronization pattern followed by the actual binary payload. This has several drawbacks. It cannot be used on spans less than 1 second and is susceptible to temporal edits. The synchronization bits also reduce the capacity for the encoded message, accounting for 31% of the total capacity. Most importantly, the brute force detection algorithm for decoding the synchronization bits is prohibitively slow especially on non-watermarked content, as we show in Sec. 5.5. This makes it unsuitable for real-time and large-scale traceability of AI-generated content on social media platforms, where most content is not watermarked.

To address these limitations, we introduce *AudioSeal*, a method for localized speech watermarking. It jointly trains two networks: a *generator* that predicts an additive watermark waveform from an audio input, and a *detector* that outputs the probability of the presence of a watermark at each sample of the input audio. The detector is trained to precisely and robustly detect synthesized speech embedded in longer audio clips by masking the watermark in random sections of the signal. The training objective is to maximize the detector's accuracy while minimizing the perceptual difference between the original and watermarked audio. We also extend AudioSeal to multi-bit watermarking, so that an audio can be attributed to a specific model or version without affecting the detection signal.

We evaluate the performance of AudioSeal to detect and localize AI-generated speech. AudioSeal achieves state-of-the-art results on robustness of the detection, far surpassing passive detection with near perfect detection rates over a wide range of audio edits. It also performs sample-level detection (at resolution of 1/16k second), outperforming WavMark in both speed and performance. In terms of efficiency, our detector is run once and yields detection logits at every time-step, allowing for real-time detection of watermarks in audio streams. This represents a major improvement compared to earlier watermarking methods, which require synchronizing the watermark within the detector, thereby substantially increasing computation time. Finally, in conjunction with binary messages, AudioSeal almost perfectly attributes an audio to one model among $1,000$, even in the presence of audio edits.

Our overall contributions are:

- We introduce AudioSeal, the first audio watermarking technique designed for localized detection of AI-generated speech up to the sample-level;
- A novel perceptual loss inspired by auditory masking, that enables AudioSeal to achieve better imperceptibility of the watermark signal;
- AudioSeal achieves the state-of-the-art robustness to a wide range of real life audio manipulations (section 5);
- AudioSeal significantly outperforms the state-of-the-art models in computation speed, achieving up to two orders of magnitude faster detection (section 5.5);
- Insights on the security and integrity of audio watermarking techniques when open-sourcing (section 6).

2. Related Work

In this section we give an overview of the detection and watermarking methods for audio data. A complementary descrition of prior works can be found in the Appendix A.

Synthetic speech detection. Detection of synthetic speech is traditionally done in the forensics community by building features and exploiting statistical differences between fake and real. These features can be handcrafted (Sahidullah et al., 2015; Janicki, 2015; AlBadawy et al., 2019; Borrelli et al., 2021) and/or learned (Müller et al., 2022; Barrington et al., 2023). The approach of most audio generation papers (Borsos et al., 2022; Kharitonov et al., 2023; Borsos et al., 2023; Le et al., 2023) is to train end-to-end deep-learning classifiers on what their models generate, similarly as Zhang et al. (2017). Accuracy when comparing synthetic to real is usually good, although not performing well on out of distribution audios (compressed, noised, slowed, etc.).

Imperceptible watermarking. Unlike forensics, watermarking actively marks the content to identify it once in the wild. It is enjoying renewed interest in the context of generative models, as it provides a means to track AI-generated content, be it for text (Kirchenbauer et al., 2023; Aaronson & Kirchner, 2023; Fernandez et al., 2023a), images (Yu et al., 2021b; Fernandez et al., 2023b; Wen et al., 2023), or audio/speech (Chen et al., 2023; Juvela & Wang, 2023).

Traditional methods for audio watermarking relied on embedding watermarks either in the time or frequency domains (Lie & Chang, 2006; Kalantari et al., 2009; Natgunanathan et al., 2012; Xiang et al., 2018; Su et al., 2018; Liu et al., 2019), usually including domain specific features to design the watermark and its corresponding decoding function. Deep-learning audio watermarking methods focus on multi-bit watermarking and follow a generator/de-

coder framework (Tai & Mansour, 2019; Qu et al., 2023; Pavlović et al., 2022; Liu et al., 2023a; Ren et al., 2023). Few works have explored zero-bit watermarking (Wu et al., 2023; Juvela & Wang, 2023), which is better adapted for detection of AI-generated content. Our rationale is that robustness increases as the message payload is reduced to the bare minimum (Furon, 2007).

In this study, we compare our work with the state-of-the-art watermarking method, WavMark (Chen et al., 2023), which outperforms previous ones. It uses invertible networks to hide 32 bits in 1-second audio segments. Detection is done by sliding along the audio in 0.05s steps and decoding the message for each window. If the 10 first decoded bits match a synchronization pattern the rest of the payload is saved (22 bits), and the window can directly slide 1s (instead of the 0.05). This brute force detection algorithm is prohibitively slow especially when the watermark is absent, since the algorithm will have to attempt and fail to decode a watermark for each sliding window in the input audio (due to the absence of watermark).

3. Method

The method jointly trains two models. The generator creates a watermark signal that is added to the input audio. The detector outputs local detection logits. The training optimizes two concurrent classes of objectives: minimizing the perceptual distortion between original and watermarked audios and maximizing the watermark detection. To improve robustness to modifications of the signal and localization, we include a collection of train time augmentations. At inference time, the logits precisely localize watermarked segments allowing for detection of AI-generated content. Optionally, short binary identifiers may be added on top of the detection to attribute a watermarked audio to a version of the model while keeping a single detector.

3.1. Training pipeline

Figure 2 illustrates the joint training of the generator and the detector with four critical stages:

(i) The watermark generator takes as input a waveform $s \in \mathbb{R}^T$ and outputs a watermark waveform $\delta \in \mathbb{R}^T$ of the same dimensionality, where T is the number of samples in the signal. The watermarked audio is then $s_w = s + \delta$.

(ii) To enable sample-level localization, we adopt an augmentation strategy focused on watermark masking with silences and other original audios. This is achieved by randomly selecting k starting points and altering the next $T/2k$ samples from s_w in one of 4 ways: revert to the original audio (*i.e.* $s_w(t) = s(t)$) with probability 0.4; replacing with zeros (*i.e.* $s_w(t) = 0$) with probability 0.2; or substituting with a different audio signal from the same batch (*i.e.* $s_w(t) = s'(t)$) with probability 0.2, or not modifying the sample at all with probability 0.2.

(iii) The second class of augmentation ensures the robustness against audio editing. One of the following signal alterations is applied: bandpass filter, boost audio, duck audio, echo, highpass filter, lowpass filter, pink noise, gaussian noise, slower, smooth, resample (full details in App. D.2). The parameters of those augmentations are fixed to aggressive values to enforce maximal robustness and the probability of sampling a given augmentation is proportional to the inverse of its evaluation detection accuracy. We implemented these augmentations in a differentiable way when possible, and otherwise (*e.g.* MP3 compression) with the straight-through estimator (Yin et al., 2019) that allows the gradients to back-propagate to the generator.

(iv) Detector D processes the original and the watermarked signals, outputting for each a soft decision at every time step, meaning $D(s) \in [0, 1]^T$. Figure 3 illustrates that the detector's outputs are at one only when the watermark is present.

The architectures of the models are based on EnCodec (Défossez et al., 2022). They are presented in Figure 4 and detailed in the appendix D.3.

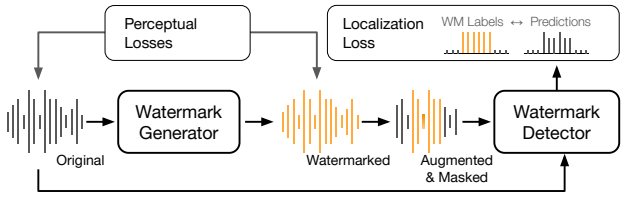

Figure 2. Generator-detector training pipeline.

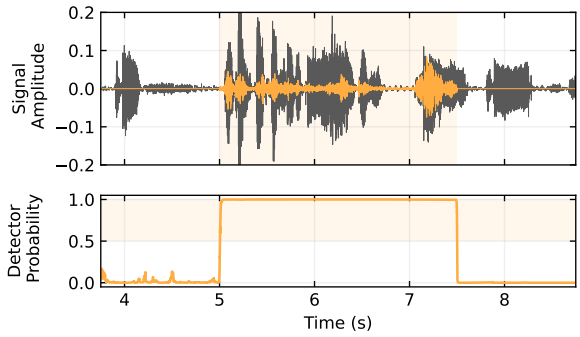

Figure 3. (Top) A speech signal (gray) where the watermark is present between 5 and 7.5 seconds (orange, magnified by 5). (Bottom) The output of the detector for every time step. An orange background color indicates the presence of the watermark.

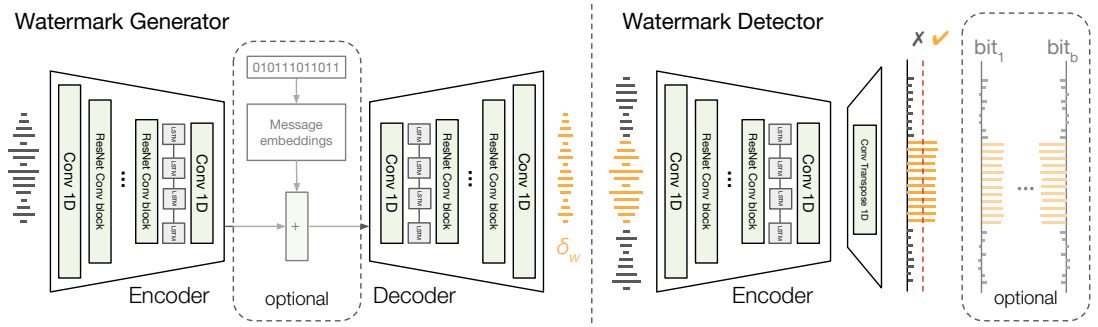

Figure 4. **Architectures**. The *generator* is made of an encoder and a decoder both derived from EnCodec's design, with optional message embeddings. The encoder includes convolutional blocks and an LSTM, while the decoder mirrors this structure with transposed convolutions. The *detector* is made of an encoder and a transpose convolution, followed by a linear layer that calculates sample-wise logits. Optionally, multiple linear layers can be used for calculating k-bit messages. More details in App. D.3.

3.2. Losses

Our setup includes multiple perceptual losses and a localization loss. We balance them during training by scaling their gradients as done by Défossez et al. (2022). The complete list of used losses is detailed bellow.

Perceptual losses enforce the watermark imperceptibility to the human ear. These include an ℓ_1 loss on the watermark signal to decrease its intensity, the multi-scale Mel spectrogram loss of (Gritsenko et al., 2020), and discriminative losses based on adversarial networks that operate on multi-scale short-term-Fourier-transform spectrograms. Défossez et al. (2022) use this combination of losses for training the EnCodec model for audio compression.

In addition, we introduce a novel time-frequency loudness loss **TF-Loudness**, which operates entirely in the waveform domain. This approach is based on "auditory masking", a psycho-acoustic property of the human auditory system already exploited in the early days of watermarking (Kirovski & Attias, 2003): the human auditory system fails perceiving sounds occurring at the same time and at the same frequency range (Schnupp et al., 2011). TF-Loudness is calculated as follows: first, the input signal s is divided into B signals based on non-overlapping frequency bands $s_0, \dots, s_{B-1}$. Subsequently, every signal is segmented using a window of size W, with an overlap amount denoted by r. This procedure is applied to both the original audio signal s and the embedded watermark δ. As a result, we obtain segments of the signal and watermark in time-frequency dimensions, denoted as s_b^w and δ_b^w respectively. For every time-frequency window we compute the loudness difference, where loudness is estimated using ITU-R BS.1770-4 recommendations (telecommunication Union, 2011) (see App. D.1 for details):

$$l_b^w = \text{Loudness}(\delta_b^w) - \text{Loudness}(s_b^w). \tag{1}$$

This measure quantifies the discrepancy in loudness between the watermark and the original signal within a specific time window w, and a particular frequency band b. The final loss is a weighted sum of the loudness differences using softmax function:

$$\mathcal{L}_{loud} = \sum_{b,w} \left(\text{softmax}(l)_b^w * l_b^w\right). \tag{2}$$

The softmax prevents the model from targeting excessively low loudness where the watermark is already inaudible.

Masked sample-level detection loss. A localization loss ensures that the detection of watermarked audio is done at the level of individual samples. For each time step t, we compute the binary cross entropy (BCE) between the detector's output $D(s)_t$ and the ground truth label (0 for non-watermarked, 1 for watermarked). Overall, this reads:

$$\mathcal{L}_{loc} = \frac{1}{T} \sum_{t=1}^{T} \text{BCE}(D(s')_t, y_t), \tag{3}$$

where s' might be s or s_w, and where time step labels y_t are set to 1 if they are watermarked, and 0 otherwise.

3.3. Multi-bit watermarking

We extend the method to support multi-bit watermarking, which allows for attribution of audio to a specific model version. *At generation*, we add a message processing layer in the middle of the generator. It takes the activation map in $\mathbb{R}^{h,t'}$ and a binary message $m \in \{0,1\}^b$ and outputs a new activation map to be added to the original one. We embed m into $e = \sum_{i=0..b-1} E_{2i+m_i} \in \mathbb{R}^h$, where $E \in \mathbb{R}^{2b,h}$ is a learnable embedding layer. e is then repeated t times along the temporal axis to match the activation map size (t, h). *At detection*, we add b linear layers at the very end of the detector. Each of them outputs a soft value for each bit of the message at the sample-level. Therefore, the detector outputs a tensor of shape $\mathbb{R}^{t,1+b}$ (1 for the detection, b for the message). *At training*, we add a decoding loss $\mathcal{L}_{dec}$ to the localization loss $\mathcal{L}_{loc}$. This loss $\mathcal{L}_{dec}$ averages the BCE between the original message and the detector's outputs over all parts where the watermark is present.

3.4. Training details

Our watermark generator and detector are trained on a 4.5K hours subset from the VoxPopuli (Wang et al., 2021) dataset. It is important to emphasize that the sole purpose of our generator is to generate imperceptible watermarks given an input audio; without the capability to produce or modify speech content. We use a sampling rate of 16 kHz and one-second samples, so $T = 16000$ in our training. A full training requires 600k steps, with Adam, a learning rate of 10^{-4}, and a batch size of 32. For the drop augmentation, we use $k = 5$ windows of 0.1 sec. h is set to 32, and the number of additional bits b to 16 (note that h needs to be higher than b, for example $h = 8$ is enough in the zero-bit case). The perceptual losses are balanced and weighted as follows: $\lambda_{\ell_1} = 0.1$, $\lambda_{msspec} = 2.0$, $\lambda_{adv} = 4.0$, $\lambda_{loud} = 10.0$. The localization and watermarking losses are weighted by $\lambda_{loc} = 10.0$ and $\lambda_{dec} = 1.0$ respectively.

3.5. Detection, localization and attribution

At inference, we may use the generator and detector for:

- *Detection*: To determine if the audio is watermarked or not. To achieve this, we use the average detector's output over the entire audio and flag it if the score exceeds a threshold (default: 0.5).
- *Localization*: To precisely identify where the watermark is present. We utilize the sample-wise detector's output and mark a time step as watermarked if the score surpasses a threshold (default: 0.5).
- *Attribution*: To identify the model version that produced the audio, enabling differentiation between users or APIs with a single detector. The detector's first output gives the detection score and the remaining k outputs are used for attribution. This is done by computing the average message over detected samples and returning the identifier with the smallest Hamming distance.

4. Audio/Speech Quality

We first evaluate the quality of the watermarked audio using: Scale Invariant Signal to Noise Ratio (SI-SNR): $\text{SI-SNR}(s, s_w) = 10\log_{10}\left(\|\alpha s\|_2^2 / \|\alpha s - s_w\|_2^2\right)$, where $\alpha = \langle s, s_w\rangle / \|s\|_2^2$; as well as PESQ (Rix et al., 2001), ViSQOL (Hines et al., 2012) and STOI (Taal et al., 2010) which are objective perceptual metrics measuring the quality of speech signals.

Table 1 report these metrics. AudioSeal behaves differently than watermarking methods like WavMark (Chen et al., 2023) that try to minimize the SI-SNR. In practice, high SI-SNR is indeed not necessarily correlated with good perceptual quality. AudioSeal is not optimized for SI-SNR but rather for perceptual quality of speech. This is better captured by the other metrics (PESQ, STOI, ViSQOL), where AudioSeal consistently achieves better performance. Put differently, our goal is to hide as much watermark power as possible while keeping it perceptually indistinguishable from the original. Figure 3 also visualizes how the watermark signal follows the shape of the speech waveform.

Table 1. **Audio quality metrics**. Compared to traditional watermarking methods that minimize the SNR like WavMark, AudioSeal achieves same or better perceptual quality.

Methods	SI-SNR	PESQ	STOI	ViSQOL	MUSHRA
WavMark	**38.25**	4.302	0.997	4.730	71.52 ± 7.18
AudioSeal	26.00	**4.470**	0.997	**4.829**	**77.07** ± 6.35

The metric used for our subjective evaluations is MUSHRA test (Series, 2014). The complete details about our full protocol can be found in the Appendix D.4. In this study our samples got ratings very close to the ground truth samples that obtained an average score of 80.49.

5. Experiments and Evaluation

This section evaluates the detection performance of passive classifiers, watermarking methods, and AudioSeal, using True Positive Rate (TPR) and False Positive Rate (FPR) as key metrics for watermark detection. TPR measures correct identification of watermarked samples, while FPR indicates the rate of genuine audio clips falsely flagged. In practical scenarios, minimizing FPR is crucial. For example, on a platform processing 1 billion samples daily, an FPR of 10^{-3} and a TPR of 0.5 means that 1 million samples require manual review each day, yet only half of the watermarked samples are detected.

5.1. Comparison with passive classifier

We first compare detection results on samples generated with Voicebox (Le et al., 2023). We compare to the passive setup where a classifier is trained to discriminate between Voicebox-generated and real audios. Following the approach in the Voicebox study, we evaluate 2,000 approximately 5-second samples from LibriSpeech, These samples have masked frames (90%, 50%, and 30% of the phonemes) pre-Voicebox generation. We evaluate on the same tasks, *i.e.* distinguishing between original and generated, or between original and re-synthesized (created by extracting the Mel spectrogram from original audio and then vocoding it with the HiFi-GAN vocoder).

Both active and passive setups achieve perfect classification in the case when trained to distinguish between natural and Voicebox. Conversely, the second part of Tab. 2 highlights a significant drop in performance when the classifier is trained to differentiate between Voicebox

Table 2. **Comparison with Voicebox binary classifier.** Percentage refers to the fraction of masked input frames.

	AudioSeal (Ours)			Voicebox Classif.		
% Mask	Acc.	TPR	FPR	Acc.	TPR	FPR
Original audio vs AI-generated audio						
30%	1.0	1.0	0.0	1.0	1.0	0.0
50%	1.0	1.0	0.0	1.0	1.0	0.0
90%	1.0	1.0	0.0	1.0	1.0	0.0
Re-synthesized audio vs AI-generated audio						
30%	**1.0**	**1.0**	**0.0**	0.704	0.680	0.194
50%	**1.0**	**1.0**	**0.0**	0.809	0.831	0.170
90%	**1.0**	**1.0**	**0.0**	0.907	0.942	0.112

Table 3. **Detection results** for different edits applied before detection. Acc. (TPR/FPR) is the accuracy (and TPR/FPR) obtained for the threshold that gives best accuracy on a balanced set of augmented samples. AUC is the area under the ROC curve.

	AudioSeal (Ours)		WavMark	
Edit	Acc. TPR/FPR	AUC	Acc. TPR/FPR	AUC
None	1.00 1.00/0.00	1.00	1.00 1.00/0.00	1.00
Bandpass	1.00 1.00/0.00	1.00	1.00 1.00/0.00	1.00
Highpass	0.61 0.82/0.60	0.61	**1.00 1.00/0.00**	**1.00**
Lowpass	**0.99 0.99/0.00**	**0.99**	0.50 1.00/1.00	0.50
Boost	1.00 1.00/0.00	1.00	1.00 1.00/0.00	1.00
Duck	1.00 1.00/0.00	1.00	1.00 1.00/0.00	1.00
Echo	**1.00 1.00/0.00**	**1.00**	0.93 0.89/0.03	0.98
Pink	**1.00 1.00/0.00**	**1.00**	0.88 0.81/0.05	0.93
White	**0.91 0.86/0.04**	**0.95**	0.50 0.54/0.54	0.50
Fast (1.25x)	**0.99 0.99/0.00**	**1.00**	0.50 0.01/0.00	0.15
Smooth	**0.99 0.99/0.00**	1.00	0.94 0.93/0.04	0.98
Resample	1.00 1.00/0.00	1.00	1.00 1.00/0.00	1.00
AAC	1.00 1.00/0.00	1.00	1.00 1.00/0.00	1.00
MP3	**1.00 1.00/0.00**	**1.00**	1.00 0.99/0.00	0.99
EnCodec	**0.98 0.98/0.01**	**1.00**	0.51 0.52/0.50	0.50
Average	**0.96 0.98/0.04**	**0.97**	0.85 0.85/0.14	0.84

and re-synthesized. It suggests that the classifier is detecting vocoder artifacts, since the re-synthesized samples are sometimes wrongly flagged. The classification performance quickly decreases as the quality of the AI-generated sample increases (when the input is less masked). On the other hand, our proactive detection does not rely on model-specific artifacts but on the watermark presence. This allows for perfect detection over all the audio clips.

5.2. Comparison with watermarking

We evaluate the robustness of the detection on a wide range of audio editing operations: time modification (faster, resample), filtering (bandpass, highpass, lowpass), audio effects (echo, boost audio, duck audio), noise (pink noise, random noise), and compression (MP3, AAC, EnCodec). These attacks cover a wide range of transformations that are commonly used in audio editing software. For all edits except EnCodec compression, evaluation with parameters in the training range would be perfect. In order to show generalization, we chose stronger parameter to the attacks than those used during training (details in App. D.2).

Detection is done on 10k ten-seconds audios from our VoxPopuli validation set. For each edit, we first build a balanced dataset made of the 10k watermarked/ 10k non-watermarked edited audio clips. We quantify the performance by adjusting the threshold of the detection score, selecting the value that maximizes accuracy (we provide corresponding TPR and FPR at this threshold). The ROC AUC (Area Under the Curve of the Receiver Operating Characteristics) gives a global measure of performance over all threshold levels, and captures the TPR/FPR trade-off. To adapt data-hiding methods (*e.g.* WavMark) for proactive detection, we embed a binary message (chosen randomly beforehand) in the generated speech before release. The detection score is then computed as the Hamming distance between the original message and the one extracted from the scrutinized audio.

We observe in Tab. 3 that AudioSeal is overall more robust, with an average AUC of 0.97 vs. 0.84 for WavMark. The performance for lowpass and highpass filters indicates that AudioSeal embeds watermarks neither in the low nor in the high frequencies (WavMark focuses on high frequencies). We give results on more augmentations in App. C.5.

Generalization. We evaluate how AudioSeal generalizes on various domains and languages. Specifically, we use the datasets ASVspoof (Liu et al., 2023b) and FakeAVCeleb (Khalid et al., 2021). Additionally, we translate speech samples from a subset of the Expresso dataset (Nguyen et al., 2023) (studio-quality recordings) using the SeamlessExpressive translation model (Seamless Communication et al., 2023). We select four target languages: Mandarin Chinese (CMN), French (FR), Italian (IT), and Spanish (SP). We also evaluate on non-speech AI-generated audios: music from MusicGen (Copet et al., 2023) and environmental sounds from AudioGen (Kreuk et al., 2023). Results are very similar to our in-domain test set and can be found in App. C.4.

5.3. Localization

We evaluate localization with the sample-level detection accuracy, *i.e.* the proportion of correctly labeled samples, and the Intersection over Union (IoU). The latter is defined as the intersection between the predicted and the ground truth detection masks (1 when watermarked, 0 otherwise), divided by their union. IoU is a more relevant evaluation of the localization of short watermarks in a longer audio.

This evaluation is carried out on the same audio clips as for detection. For each one of them, we watermark a randomly placed segment of varying length. Localization with WavMark is a brute-force detection: a window of 1s slides over the 10s of speech with the default shift value of 0.05s. The Hammning distance between the 16 pattern bits is used as the detection score. Whenever a window triggers a positive, we label its 16k samples as watermarked in the detection mask in $\{0,1\}^t$.

Figure 5 plots the sample-level accuracy and IoU for different proportions of watermarked speech in the audio clip. AudioSeal achieves an IoU of 0.99 when just one second of speech is AI-manipulated, compared to WavMark's 0.35. Moreover, AudioSeal allows for precise detection of minor audio alterations: it can pinpoint AI-generated segments in audio down to the sample level (usually 1/16k sec), while the concurrent WavMark only provides one-second resolution and therefore lags behind in terms of IoU. This is especially relevant for speech samples, where a simple word modification may greatly change meaning.

5.4. Attribution

Given an audio clip, the objective is now to find if any of N versions of our model generated it (detection), and if so, which one (identification). For evaluation, we create $N' = 100$ random 16-bits messages and use them to watermark 1k audio clips, each consisting of 5 seconds of speech (not 10s to reduce compute needs). This results in a total of 100k audios. For WavMark, the first 16 bits (/32) are fixed and the detection score is the number of well decoded pattern bits, while the second half of the payload hides the model version. An audio clip is flagged if the average output of the detector exceeds a threshold, corresponding to FPR=10^{-3}. Next, we calculate the Hamming distance between the decoded watermark and all N original messages.

Table 4. **Attribution results**. We report the accuracy of the attribution (Acc.) and false attribution rate (FAR). Detection is done at FPR=10^{-3} and attribution matches the decoded message to one of N versions. We report averaged results over the edits of Tab. 3.

	N	1	10	10^2	10^3	10^4
FAR (%) ↓	WavMark	0.0	**0.20**	**0.98**	**1.87**	**4.02**
	AudioSeal	0.0	2.52	6.83	8.96	11.84
Acc. (%) ↑	WavMark	58.4	58.2	57.4	56.6	54.4
	AudioSeal	**68.2**	**65.4**	**61.4**	**59.3**	**56.4**

The message with the smallest Hamming distance is selected. It's worth noting that we can simulate $N > N'$ models by adding extra messages. This may represent versions that have not generated any sample.

False Attribution Rate (FAR) is the fraction of wrong attribution *among the detected audios* while the attribution accuracy is the proportion of detections followed by a correct attributions *over all audios*. AudioSeal has a higher FAR but overall gives a better accuracy, which is what ultimately matters. In summary, decoupling detection and attribution achieves better detection rate and makes the global accuracy better, at the cost of occasional false attributions.

5.5. Efficiency Analysis

To highlight the efficiency of AudioSeal, we conduct a performance analysis and compare it with WavMark. We apply the watermark generator and detector of both models on a dataset of 500 audio segments ranging in length from 1 to 10 seconds, using a single Nvidia Quadro GP100 GPU. The results are displayed in Fig. 6 and Tab. 5. In terms of generation, AudioSeal is 14x faster than WavMark. For detection, AudioSeal outperforms WavMark with two orders of magnitude faster performance on average, notably 485x faster in scenarios where there is no

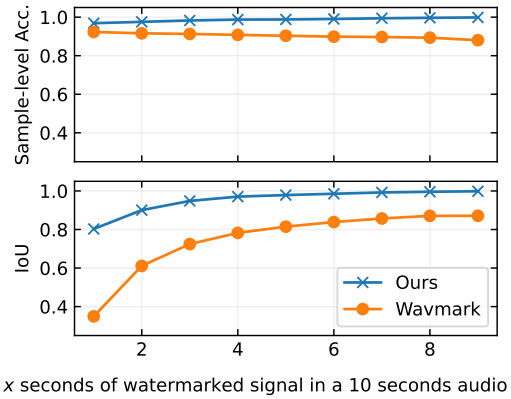

Figure 5. **Localization results** across different durations of watermarked audio signals in terms of Sample-Level Accuracy and Intersection Over Union (IoU) metrics (↑ is better).

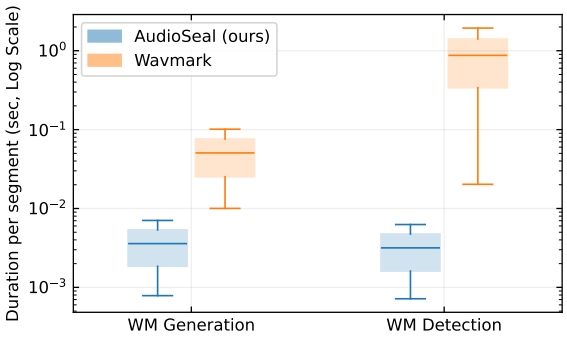

Figure 6. **Mean runtime** (↓ is better). AudioSeal is one order of magnitude faster for watermark generation and two orders of magnitude faster for watermark detection for the same audio input. See Appendix C.1 for full comparison.

watermark (Tab. 5). This remarkable speed increase is due to our model's unique localized watermark design, which bypasses the need for watermark synchronization (recall that WavMark relies on 20 pass forwards for a one-second snippet). AudioSeal's detector provides detection logits for each input sample directly with only one pass to the detector, significantly enhancing the detection's computational efficiency. This makes our system highly suitable for real-time and large-scale applications.

6. Adversarial Watermark Removal

We now examine more damaging deliberate attacks, where attackers might either "forge" the watermark by adding it to authentic samples (to overwhelm detection systems) or "remove" it to avoid detection. Our findings suggest that in order to maintain the effectiveness of watermarking against such adversaries, the code for training watermarking models and the awareness that published audios are watermarked can be made public. However, the detector's weights should be kept confidential.

We focus on watermark-removal attacks and consider three types of attacks depending on the adversary's knowledge:

- *White-box*: the adversary has access to the detector (*e.g.* because of a leak), and performs a gradient-based adversarial attack against it. The optimization objective is to minimize the detector's output.
- *Semi black-box*: the adversary does not have access to any weights, but is able to re-train generator/detector pairs with the same architectures on the same dataset. They perform the same gradient-based attack as before, but using the new detector as proxy for the original one.
- *Black-box*: the adversary does not have any knowledge on the watermarking algorithm being used, but has access to an API that produces watermarked samples, and to negative speech samples from any public dataset. They first collect samples and train a classifier to discriminate between watermarked and not-watermarked. They attack this classifier as if it were the true detector.

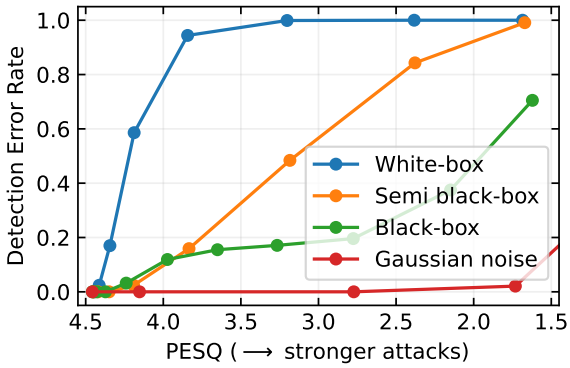

Figure 7. **Watermark-removal attacks.** PESQ is measured between attacked audios and genuine ones (PESQ < 4 strongly degrades the audio quality). The more knowledge the attacker has over the watermarking algorithm, the better the attack is.

For every scenario, we watermark 1k samples of 5 seconds, then attack them. The gradient-based attack optimizes an adversarial noise added to the audio, with 100 steps of Adam. During the optimization, we control the norm of the noise to trade off attack strength and audio quality. When training the classifier for the black-box attack, we use 80k/80k watermarked/genuine samples of 8 seconds and make sure the classifier has 100% detection accuracy on the validation set. More details in App. D.5.

Figure 7 contrasts various attacks at different intensities, using Gaussian noise as a reference. The white-box attack is by far the most effective one, increasing the detection error by around 80%, while maintaining high audio quality (PESQ > 4). Other attacks are less effective, requiring significant audio quality degradation to achieve 50% increase the detection error, though they are still more effective than random noise addition. In summary, the more is disclosed about the watermarking algorithm, the more vulnerable it is. The effectiveness of these attacks is limited as long as the detector remains confidential.

7. Conclusion

In this paper, we introduced AudioSeal, a proactive method for the detection, localization, and attribution of AI-generated speech. AudioSeal revamps the design of audio watermarking to be specific to localized detection rather than data hiding. It is based on a generator/detector architecture that can generate and extract watermarks at the audio sample level. This removes the dependency on slow brute force algorithms, traditionally used to encode and decode audio watermarks. The networks are jointly trained through a novel loudness loss, differentiable augmentations and masked sample level detection losses. As a result, AudioSeal achieves state-of-the-art robustness to various audio editing techniques, very high precision in localization, and orders of magnitude faster runtime than methods relying on synchronization. Through an empirical analysis of possible adversarial attacks, we conclude that for watermarking to still be an effective mitigation, the detector's weights have to be kept private – otherwise adversarial attacks might be easily forged. A key advantage of AudioSeal is its practical applicability. It stands as a ready-to-deploy solution for watermarking in voice synthesis APIs. This is pivotal for large-scale content provenance on social media and for detecting and eliminating incidents, enabling swift action on instances like the US voters' deepfake case (Murphy et al., 2024) long before they spread.

Impact Statement

This research aims to improve transparency and traceability in AI-generated content, but watermarking in general can have a set of potential misuses such as government surveillance of dissidents or corporate identification of whistle blowers. Additionally, the watermarking technology might be misused to enforce copyright on user-generated content, and its ability to detect AI-generated audio could increase skepticism about digital communication authenticity, potentially undermining trust in digital media and AI. However, despite these risks, ensuring the detectability of AI-generated content is important, along with advocating for robust security measures and legal frameworks to govern the technology's use.

References

Chinese ai governance rules, 2023. URL http://www.cac.gov.cn/2023-07/13/c_1690898327029107.htm. Accessed on August 29, 2023.

European ai act, 2023. URL https://artificialintelligenceact.eu/. Accessed on August 29, 2023.

Aaronson, S. and Kirchner, H. Watermarking gpt outputs, 2023. URL https://www.scottaaronson.com/talks/watermark.ppt.

AlBadawy, E. A., Lyu, S., and Farid, H. Detecting ai-synthesized speech using bispectral analysis. In *CVPR workshops*, pp. 104–109, 2019.

Arik, S., Chen, J., Peng, K., Ping, W., and Zhou, Y. Neural voice cloning with a few samples. *Advances in neural information processing systems*, 31, 2018.

Bai, H., Zheng, R., Chen, J., Ma, M., Li, X., and Huang, L. A^3t: Alignment-aware acoustic and text pretraining for speech synthesis and editing. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvári, C., Niu, G., and Sabato, S. (eds.), *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pp. 1399–1411. PMLR, 2022. URL https://proceedings.mlr.press/v162/bai22d.html.

Barrington, S., Barua, R., Koorma, G., and Farid, H. Single and multi-speaker cloned voice detection: From perceptual to learned features. *arXiv preprint arXiv:2307.07683*, 2023.

Borrelli, C., Bestagini, P., Antonacci, F., Sarti, A., and Tubaro, S. Synthetic speech detection through short-term and long-term prediction traces. *EURASIP Journal on Information Security*, 2021(1):1–14, 2021.

Borsos, Z., Marinier, R., Vincent, D., Kharitonov, E., Pietquin, O., Sharifi, M., Roblek, D., Teboul, O., Grangier, D., Tagliasacchi, M., and Zeghidour, N. Audiolm: A language modeling approach to audio generation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 31:2523–2533, 2022.

Borsos, Z., Sharifi, M., Vincent, D., Kharitonov, E., Zeghidour, N., and Tagliasacchi, M. Soundstorm: Efficient parallel audio generation. *arXiv preprint arXiv:2305.09636*, 2023.

Casanova, E., Weber, J., Shulby, C. D., Junior, A. C., Gölge, E., and Ponti, M. A. Yourtts: Towards zero-shot multi-speaker tts and zero-shot voice conversion for everyone. In *International Conference on Machine Learning*, pp. 2709–2720. PMLR, 2022.

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

Copet, J., Kreuk, F., Gat, I., Remez, T., Kant, D., Synnaeve, G., Adi, Y., and Défossez, A. Simple and controllable music generation. *arXiv preprint arXiv:2306.05284*, 2023.

Defossez, A., Synnaeve, G., and Adi, Y. Real time speech enhancement in the waveform domain, 2020.

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

Fernandez, P., Chaffin, A., Tit, K., Chappelier, V., and Furon, T. Three bricks to consolidate watermarks for large language models. *2023 IEEE International Workshop on Information Forensics and Security (WIFS)*, 2023a.

Fernandez, P., Couairon, G., Jégou, H., Douze, M., and Furon, T. The stable signature: Rooting watermarks in latent diffusion models. *ICCV*, 2023b.

Furon, T. A constructive and unifying framework for zero-bit watermarking. *IEEE Transactions on Information Forensics and Security*, 2(2):149–163, 2007.

Gemmeke, J. F., Ellis, D. P., Freedman, D., Jansen, A., Lawrence, W., Moore, R. C., Plakal, M., and Ritter, M. Audio set: An ontology and human-labeled dataset for audio events. In *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pp. 776–780. IEEE, 2017.

Gritsenko, A., Salimans, T., van den Berg, R., Snoek, J., and Kalchbrenner, N. A spectral energy distance for parallel speech synthesis. *Advances in Neural Information Processing Systems*, 33:13062–13072, 2020.

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

Hsu, W.-N., Akinyemi, A., Rakotoarison, A., Tjandra, A., Vyas, A., Guo, B., Akula, B., Shi, B., Ellis, B., Cruz, I., Wang, J., Zhang, J., Williamson, M., Le, M., Moritz, R., Adkins, R., Ngan, W., Zhang, X., Yungster, Y., and Wu, Y.-C. Audiobox: Unified audio generation with natural language prompts. *arXiv preprint arXiv:...*, 2023.

Janicki, A. Spoofing countermeasure based on analysis of linear prediction error. In *Sixteenth annual conference of the international speech communication association*, 2015.

Juvela, L. and Wang, X. Collaborative watermarking for adversarial speech synthesis. *arXiv preprint arXiv:2309.15224*, 2023.

Kalantari, N. K., Akhaee, M. A., Ahadi, S. M., and Amindavar, H. Robust multiplicative patchwork method for audio watermarking. *IEEE Trans. Speech Audio Process.*, 17(6):1133–1141, 2009. doi: 10.1109/TASL.2009.2019259. URL https://doi.org/10.1109/TASL.2009.2019259.

Khalid, H., Tariq, S., and Woo, S. S. Fakeavceleb: A novel audio-video multimodal deepfake dataset, 2021.

Kharitonov, E., Vincent, D., Borsos, Z., Marinier, R., Girgin, S., Pietquin, O., Sharifi, M., Tagliasacchi, M., and Zeghidour, N. Speak, read and prompt: High-fidelity text-to-speech with minimal supervision. *ArXiv*, abs/2302.03540, 2023.

Kim, C., Min, K., Patel, M., Cheng, S., and Yang, Y. Wouaf: Weight modulation for user attribution and fingerprinting in text-to-image diffusion models. *arXiv preprint arXiv:2306.04744*, 2023.

Kim, J., Kong, J., and Son, J. Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech. In *International Conference on Machine Learning*, pp. 5530–5540. PMLR, 2021.

Kirchenbauer, J., Geiping, J., Wen, Y., Katz, J., Miers, I., and Goldstein, T. A watermark for large language models. *arXiv preprint arXiv:2301.10226*, 2023.

Kirovski, D. and Attias, H. Audio watermark robustness to desynchronization via beat detection. In Petitcolas, F. A. P. (ed.), *Information Hiding*, pp. 160–176, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg. ISBN 978-3-540-36415-3.

Kirovski, D. and Malvar, H. S. Spread-spectrum watermarking of audio signals. *IEEE Trans. Signal Process.*, 51(4):1020–1033, 2003. doi: 10.1109/TSP.2003.809384. URL https://doi.org/10.1109/TSP.2003.809384.

Kong, J., Kim, J., and Bae, J. Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 17022–17033. Curran Associates, Inc., 2020.

Kreuk, F., Synnaeve, G., Polyak, A., Singer, U., Défossez, A., Copet, J., Parikh, D., Taigman, Y., and Adi, Y. Audiogen: Textually guided audio generation. In *The Eleventh International Conference on Learning Representations*, 2023.

Kumar, K., Kumar, R., de Boissière, T., Gestin, L., Teoh, W. Z., Sotelo, J. M. R., de Brébisson, A., Bengio, Y., and Courville, A. C. Melgan: Generative adversarial networks for conditional waveform synthesis. In *Neural Information Processing Systems*, 2019.

Kumar, R., Seetharaman, P., Luebs, A., Kumar, I., and Kumar, K. High-fidelity audio compression with improved rvqgan. *ArXiv*, abs/2306.06546, 2023.

Le, M., Vyas, A., Shi, B., Karrer, B., Sari, L., Moritz, R., Williamson, M., Manohar, V., Adi, Y., Mahadeokar, J., et al. Voicebox: Text-guided multilingual universal speech generation at scale. *arXiv preprint arXiv:2306.15687*, 2023.

Lie, W. and Chang, L. Robust and high-quality time-domain audio watermarking based on low-frequency amplitude modification. *IEEE Trans. Multim.*, 8 (1):46–59, 2006. doi: 10.1109/TMM.2005.861292. URL https://doi.org/10.1109/TMM.2005.861292.

Liu, C., Zhang, J., Fang, H., Ma, Z., Zhang, W., and Yu, N. Dear: A deep-learning-based audio re-recording resilient watermarking. In Williams, B., Chen, Y., and Neville, J. (eds.), *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023*, pp. 13201–13209. AAAI Press, 2023a. doi: 10.1609/aaai.v37i11.26550.

Liu, X., Wang, X., Sahidullah, M., Patino, J., Delgado, H., Kinnunen, T., Todisco, M., Yamagishi, J., Evans, N., Nautsch, A., et al. Asvspoof 2021: Towards spoofed and deepfake speech detection in the wild. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2023b.

Liu, Z., Huang, Y., and Huang, J. Patchwork-based audio watermarking robust against de-synchronization and recapturing attacks. *IEEE Trans. Inf. Forensics Secur.*, 14(5):1171–1180, 2019. doi: 10.1109/TIFS.2018.2871748. URL https://doi.org/10.1109/TIFS.2018.2871748.

Luo, Y. and Mesgarani, N. Conv-tasnet: Surpassing ideal time–frequency magnitude masking for speech separation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 27(8):1256–1266, 2019. doi: 10.1109/TASLP.2019.2915167.

Luo, Y., Chen, Z., and Yoshioka, T. Dual-path rnn: efficient long sequence modeling for time-domain single-channel speech separation. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 46–50. IEEE, 2020.

Müller, N. M., Czempin, P., Dieckmann, F., Froghyar, A., and Böttinger, K. Does audio deepfake detection generalize? *arXiv preprint arXiv:2203.16263*, 2022.

Murphy, M., Metz, R., Bergen, M., and Bloomberg. Biden audio deepfake spurs ai startup elevenlabs—valued at $1.1 billion—to ban account: 'we're going to see a lot more of this'. *Fortune*, January 2024. URL https://fortune.com/2024/01/27/ai-firm-elevenlabs-bans-account-for-biden-audio-deepfake/.

Natgunanathan, I., Xiang, Y., Rong, Y., Zhou, W., and Guo, S. Robust patchwork-based embedding and decoding scheme for digital audio watermarking. *IEEE Trans. Speech Audio Process.*, 20(8):2232–2239, 2012. doi: 10.1109/TASL.2012.2199111. URL https://doi.org/10.1109/TASL.2012.2199111.

Nguyen, T. A., Hsu, W.-N., d'Avirro, A., Shi, B., Gat, I., Fazel-Zarani, M., Remez, T., Copet, J., Synnaeve, G., Hassid, M., et al. Expresso: A benchmark and analysis of discrete expressive speech resynthesis. *arXiv preprint arXiv:2308.05725*, 2023.

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

Qu, X., Yin, X., Wei, P., Lu, L., and Ma, Z. Audioqr: Deep neural audio watermarks for qr code. *IJCAI*, 2023.

Ren, Y., Zhu, H., Zhai, L., Sun, Z., Shen, R., and Wang, L. Who is speaking actually? robust and versatile speaker traceability for voice conversion. *arXiv preprint arXiv:2305.05152*, 2023.

Rix, A. W., Beerends, J. G., Hollier, M. P., and Hekstra, A. P. 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. Proceedings (Cat. No. 01CH37221)*, volume 2, pp. 749–752. IEEE, 2001.

Sahidullah, M., Kinnunen, T., and Hanilçi, C. A comparison of features for synthetic speech detection. *ISCA (the International Speech Communication Association)*, 2015.

Schnupp, J., Nelken, I., and King, A. *Auditory neuroscience: Making sense of sound*. MIT press, 2011.

Seamless Communication, Barrault, L., Chung, Y.-A., Meglioli, M. C., Dale, D., Dong, N., Duppenthaler, M., Duquenne, P.-A., Ellis, B., Elsahar, H., Haaheim, J., Hoffman, J., Hwang, M.-J., Inaguma, H., Klaiber, C., Kulikov, I., Li, P., Licht, D., Maillard, J., Mavlyutov, R., Rakotoarison, A., Sadagopan, K. R., Ramakrishnan, A., Tran, T., Wenzek, G., Yang, Y., Ye, E., Evtimov, I., Fernandez, P., Gao, C., Hansanti, P., Kalbassi, E., Kallet, A., Kozhevnikov, A., Mejia, G., Roman, R. S., Touret, C., Wong, C., Wood, C., Yu, B., Andrews, P., Balioglu, C., Chen, P.-J., Costa-jussà, M. R., Elbayad, M., Gong, H., Guzmán, F., Heffernan, K., Jain, S., Kao, J., Lee, A., Ma, X., Mourachko, A., Peloquin, B., Pino, J., Popuri, S., Ropers, C., Saleem, S., Schwenk, H., Sun, A., Tomasello, P., Wang, C., Wang, J., Wang, S., and Williamson, M. Seamless: Multilingual expressive and streaming speech translation. 2023.

Series, B. Method for the subjective assessment of intermediate quality level of audio systems. *International Telecommunication Union Radiocommunication Assembly*, 2014.

Shen, K., Ju, Z., Tan, X., Liu, Y., Leng, Y., He, L., Qin, T., Zhao, S., and Bian, J. Naturalspeech 2: Latent diffusion models are natural and zero-shot speech and singing synthesizers. *CoRR*, abs/2304.09116, 2023. doi: 10.48550/ARXIV.2304.09116. URL https://doi.org/10.48550/arXiv.2304.09116.

Su, Z., Zhang, G., Yue, F., Chang, L., Jiang, J., and Yao, X. Snr-constrained heuristics for optimizing the scaling parameter of robust audio watermarking. *IEEE Trans. Multim.*, 20(10):2631–2644, 2018. doi: 10.1109/TMM.2018.2812599. URL https://doi.org/10.1109/TMM.2018.2812599.

Taal, C. H., Hendriks, R. C., Heusdens, R., and Jensen, J. A short-time objective intelligibility measure for time-frequency weighted noisy speech. In *2010 IEEE international conference on acoustics, speech and signal processing*, pp. 4214–4217. IEEE, 2010.

Tai, Y.-Y. and Mansour, M. F. Audio watermarking over the air with modulated self-correlation. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2452–2456. IEEE, 2019.

telecommunication Union, I. Algorithms to measure audio programme loudness and true-peak audio level. *Series, BS*, 2011.

USA. Ensuring safe, secure, and trustworthy ai. https://www.whitehouse.gov/wp-content/uploads/2023/07/Ensuring-Safe-Secure-and-Trustworthy-AI.pdf, July 2023. Accessed: [july 2023].

van den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., and Kavukcuoglu, K. Wavenet: A generative model for raw audio. In *Arxiv*, 2016.

Wang, C., Rivière, M., Lee, A., Wu, A., Talnikar, C., Haziza, D., Williamson, M., Pino, J. M., and Dupoux, E. Voxpopuli: A large-scale multilingual speech corpus for representation learning, semi-supervised learning and interpretation. In Zong, C., Xia, F., Li, W., and Navigli, R. (eds.), *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pp. 993–1003. Association for Computational Linguistics, 2021. doi: 10.18653/V1/2021.ACL-LONG.80. URL https://doi.org/10.18653/v1/2021.acl-long.80.

Wang, C., Chen, S., Wu, Y., Zhang, Z., Zhou, L., Liu, S., Chen, Z., Liu, Y., Wang, H., Li, J., et al. Neural codec language models are zero-shot text to speech synthesizers. *arXiv preprint arXiv:2301.02111*, 2023.

Wen, Y., Kirchenbauer, J., Geiping, J., and Goldstein, T. Tree-ring watermarks: Fingerprints for diffusion images that are invisible and robust. *arXiv preprint arXiv:2305.20030*, 2023.

Wu, S., Liu, J., Huang, Y., Guan, H., and Zhang, S. Adversarial audio watermarking: Embedding watermark into deep feature. In *2023 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 61–66. IEEE, 2023.

Xiang, Y., Natgunanathan, I., Guo, S., Zhou, W., and Nahavandi, S. Patchwork-based audio watermarking method robust to de-synchronization attacks. *IEEE ACM Trans. Audio Speech Lang. Process.*, 22(9):1413–1423, 2014. doi: 10.1109/TASLP.2014.2328175. URL https://doi.org/10.1109/TASLP.2014.2328175.

Xiang, Y., Natgunanathan, I., Peng, D., Hua, G., and Liu, B. Spread spectrum audio watermarking using multiple orthogonal PN sequences and variable embedding strengths and polarities. *IEEE ACM Trans. Audio Speech Lang. Process.*, 26(3):529–539, 2018. doi: 10.1109/TASLP.2017.2782487. URL https://doi.org/10.1109/TASLP.2017.2782487.

Yang, Y.-Y., Hira, M., Ni, Z., Chourdia, A., Astafurov, A., Chen, C., Yeh, C.-F., Puhrsch, C., Pollack, D., Genzel, D., Greenberg, D., Yang, E. Z., Lian, J., Mahadeokar, J., Hwang, J., Chen, J., Goldsborough, P., Roy, P., Narenthiran, S., Watanabe, S., Chintala, S., Quenneville-Bélair, V., and Shi, Y. Torchaudio: Building blocks for audio and speech processing. *arXiv preprint arXiv:2110.15018*, 2021.

Yin, P., Lyu, J., Zhang, S., Osher, S., Qi, Y., and Xin, J. Understanding straight-through estimator in training activation quantized neural nets. *arXiv preprint arXiv:1903.05662*, 2019.

Yu, N., Skripniuk, V., Abdelnabi, S., and Fritz, M. Artificial fingerprinting for generative models: Rooting deepfake attribution in training data. In *Proceedings of the IEEE/CVF International conference on computer vision*, pp. 14448–14457, 2021a.

Yu, N., Skripniuk, V., Chen, D., Davis, L. S., and Fritz, M. Responsible disclosure of generative models using scalable fingerprinting. In *International Conference on Learning Representations*, 2021b.

Zeghidour, N., Luebs, A., Omran, A., Skoglund, J., and Tagliasacchi, M. Soundstream: An end-to-end neural audio codec. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:495–507, 2022. doi: 10.1109/TASLP.2021.3129994.

Zhang, C., Yu, C., and Hansen, J. H. An investigation of deep-learning frameworks for speaker verification antispoofing. *IEEE Journal of Selected Topics in Signal Processing*, 11(4):684–694, 2017.

A. Extended related work

Zero-shot TTS and vocal style preservation. There has been an emergence of models that imitate or preserve vocal style using only a small amount of data. One key example is zero-shot text-to-speech (TTS) models. These models create speech in vocal styles they haven't been specifically trained on. For instance, models like VALL-E (Wang et al., 2023), YourTTS (Casanova et al., 2022), NaturalSpeech2 (Shen et al., 2023) synthesize high-quality personalized speech with only a 3-second recording. On top, zero-shot TTS models like Voicebox (Le et al., 2023), A^{3}T (Bai et al., 2022) and Audiobox (Hsu et al., 2023), with their non-autoregressive inference, perform tasks such as text-guided speech infilling, where the goal is to generate masked speech given its surrounding audio and text transcript. It makes them a powerful tool for speech manipulation. In the context of speech machine translation, SeamlessExpressive (Seamless Communication et al., 2023) is a model that not only translates speech, but also retains the speaker's unique vocal style and emotional inflections, thereby broadening the capabilities of such systems.

Audio generation and compression. Early models are autoregressive like WaveNet (van den Oord et al., 2016), with dilated convolutions and waveform reconstruction as objective. Subsequent approaches explore different audio losses, such as scale-invariant signal-to-noise ratio (SI-SNR) (Luo & Mesgarani, 2019) or Mel spectrogram distance (Defossez et al., 2020). None of these objectives are deemed ideal for audio quality, leading to the adoption of adversarial models in HiFi-GAN (Kong et al., 2020) or MelGAN (Kumar et al., 2019). Our training objectives and architectures are inspired by more recent neural audio compression models (Défossez et al., 2022; Kumar et al., 2023; Zeghidour et al., 2022), that focus on high-quality waveform generation and integrate a combination of these diverse objectives in their training processes.

Synchronization and Detection speed. To accurately extract watermarks, synchronization between the encoder and decoder is crucial. However, this can be disrupted by desynchronization attacks such as time and pitch scaling. To address this issue, various techniques have been developed. One approach is block repetition, which repeats the watermark signal along both the time and frequency domains (Kirovski & Malvar, 2003; Kirovski & Attias, 2003). Another method involves implanting synchronization bits into the watermarked signal (Xiang et al., 2014). During decoding, these synchronization bits serve to improve synchronization and mitigate the effects of de-synchronization attacks. Detection of those synchronization bits for watermark detection usually involves exhaustive search using brute force algorithms, which significantly slows down decoding time.

B. False Positive Rates - Theory and Practice

Theoretical FPR. When doing multi-bit watermarking, previous works (Yu et al., 2021a; Kim et al., 2023; Fernandez et al., 2023b; Chen et al., 2023) usually extract the message m' from the content x and compare it to the original binary signature $m \in \{0,1\}^k$ embedded in the speech sample. The detection test relies on the number of matching bits $M(m, m')$:

$$\text{if } M(m, m') \geq \tau \quad \text{where} \quad \tau \in \{0, \dots, k\}, \tag{4}$$

then the audio is flagged. This provides theoretical guarantees over the false positive rates.

Formally, the statistical hypotheses are H_1: "The audio signal x is watermarked", and the null hypothesis H_0: "The audio signal x is genuine". Under H_0 (*i.e.*, for unmarked audio), if the bits $m'_1, \dots, m'_k$ are independent and identically distributed (i.i.d.) Bernoulli random variables with parameter 0.5, then $M(m, m')$ follows a binomial distribution with parameters $(k, 0.5)$. The False Positive Rate (FPR) is defined as the probability that $M(m, m')$ exceeds a given threshold τ. A closed-form expression can be given using the regularized incomplete beta function $I_x(a; b)$ (linked to the CDF of the binomial distribution):

$$\text{FPR}(\tau) = \mathbb{P}(M \geq \tau | H_0) = I_{1/2}(\tau, k - \tau + 1). \tag{5}$$

Empirical study. We empirically study the FPR of WavMark-based detection on our validation dataset. We use the same parameters as in the original paper, *i.e.* $k = 32$-bits are extracted from 1s speech samples. We first extract the soft bits (before thresholding) from 10k genuine samples and plot the histogram of the scores in Fig. 8 (left). We should observe a Gaussian distribution with mean 0.5, while empirically the scores are centered around 0.38. This makes the decision heavily biased towards bit 0 on genuine samples. It is therefore impossible to theoretically set the FPR since this would largely underestimate the actual one.

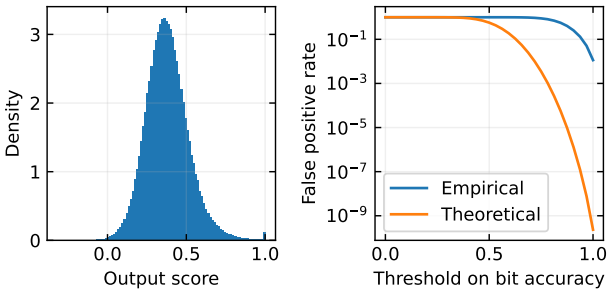

Figure 8. (Left) Histogram of scores output by WavMark's extractor on 10k genuine samples. (Right) Empirical and theoretical FPR when the chosen hidden message is all 0.

Table 5. The average runtime (ms) per sample of our proposed AudioSeal model against the state-of-the-art Wavmark(Chen et al., 2023) method. Our experiments were conducted on a dataset of audio segments spanning 1 sec to 10 secs, using a single Nvidia Quadro GP100 GPU. The results, displayed in the table, demonstrate substantial speed enhancements for both Watermark Generation and Detection with and without the presence of a watermark. Notably, for watermark detection, AudioSeal is **485× faster** than Wavmark during the absence of a watermark, more details in section 5.5.

Model	Watermarked	**Detection ms (speedup)**	**Generation ms (speedup)**
Wavmark	No	1710.70 ± 1314.02	–
AudioSeal (ours)	No	$\mathbf{3.25} \pm \mathbf{1.99}$ (**485**×)	–
Wavmark	Yes	106.21 ± 66.95	104.58 ± 65.66
AudioSeal (ours)	Yes	$\mathbf{3.30} \pm \mathbf{2.03}$ (**35**×)	$\mathbf{7.41} \pm \mathbf{4.52}$ (**14** ×)

For instance, Figure 8 (right) shows the theoretical and empirical FPR for different values of τ when the chosen hidden message is full 0. Put differently, the argument that says that hiding bits allows for theoretical guarantees over the detection rates is not valid in practice.

C. Additional Experimental Results

C.1. Computational efficiency

We show in Figure 9 the mean runtime of the detection and generation depending on the audio duration. Corresponding numbers are given in Table 5.

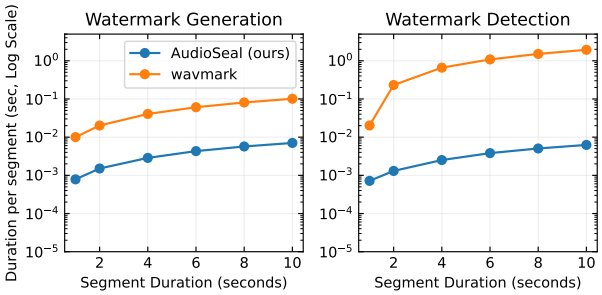

Figure 9. **Mean runtime** (↓ is better) of AudioSeal versus WavMark. AudioSeal is one order of magnitude faster for watermark generation andtwo orders of magnitude faster for watermark detection for the same audio input, signifying a considerable enhancement in real-time audio watermarking efficiency.

C.2. Another architecture

Our architecture relies on the SOTA compression method EnCodec. However, to further validate our approach, we conduct an ablation study using a different architecture DPRNN (Luo et al., 2020). The results are presented in Tab. 6. They show that the performance of AudioSeal is consistent across different architectures, with similar performances using the much slower and heavier architecture from Luo et al. (2020). This indicates that model capacity is not a limiting factor for AudioSeal.

Table 6. Results of AudioSeal with different architectures for the generator and detector. The IoU is computed for 1s of watermark in 10s audios (corresponding to the leftmost point in Fig. 5).

Method	SISNR	STOI	PESQ	Acc.	IoU
EnCodec	26.00	0.997	4.470	1.00	0.802
DPRNN	26.7	0.996	4.421	1.00	0.796

C.3. Audio mixing

We hereby evaluate the scenario where two watermarked signals (e.g., vocal and instrumental) are mixed together. To explore this, we conducted experiments using a non-vocal music dataset. In these experiments, we normalized and summed the loudness of watermarked speech and music segments. The results are detailed Tab. 7.

Table 7. Detection results for watermarked speech and music mixed signals. ✓ and ✗ indicate the presence of the watermark.

Speech	BG Music	Acc. FPR / TPR	AUC
✓	✓	0.9996 0.0003 / 0.9996	0.9999
✓	✗	0.9787 0.0310 / 0.9883	0.9961

C.4. Out of domain (OOD) evaluations

As previously outlined in Sec. 5.2, we tested AudioSeal on the outputs of various voice cloning models and other audio modalities. We employed the same set of augmentations and observed very similar results, as demonstrated in Tab. 8. Interestingly, even though we did not train our model on AI-generated speech, we noticed an improvement in performance compared to our test data. No sample was misclassified among the 10k samples that comprised each of our out-of-distribution (OOD) datasets. We also provide the other perceptual metrics results on OOD data in Tab. 9.

We also evaluated AudioSeal on three additional datasets containing real human speech: AudioSet (Gemmeke et al., 2017), ASVspoof (Liu et al., 2023b), and FakeAVCeleb (Khalid et al., 2021). Again, we observed similar performance, as shown in Tab. 10.

Table 8. Evaluation of AudioSeal Generalization across domains and languages. Namely, translations of speech samples from the Expresso dataset (Nguyen et al., 2023) to four target languages: Mandarin Chinese (CMN), French (FR), Italian (IT), and Spanish (SP), using the SeamlessExpressive model (Seamless Communication et al., 2023). Music from MusicGen (Copet et al., 2023) and environmental sounds from AudioGen (Kreuk et al., 2023).

Aug	Seamless (Cmn)	Seamless (Spa)	Seamless (Fra)	Seamless(Ita)	Seamless (Deu)	Voicebox (Eng)	AudioGen	MusicGen
None	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Bandpass	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Highpass	0.71	0.68	0.70	0.70	0.70	0.64	0.52	0.52
Lowpass	1.00	0.99	1.00	1.00	1.00	1.00	1.00	1.00
Boost	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Duck	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Echo	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Pink	0.99	1.00	0.99	1.00	0.99	1.00	1.00	1.00
White	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Fast (x1.25)	0.97	0.98	0.99	0.98	0.99	0.98	0.87	0.87
Smooth	0.96	0.99	0.99	0.99	0.99	0.99	0.98	0.98
Resample	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
AAC	0.99	0.99	0.99	0.99	0.99	0.97	0.99	0.98
MP3	0.99	0.99	0.99	0.99	0.99	0.97	0.99	1.00
Encodec	0.97	0.98	0.99	0.99	0.98	0.96	0.95	0.95
Average	0.97	0.97	0.98	0.98	0.98	0.97	0.95	0.95

Table 9. Audio quality and intelligibility evaluations on AI generated speech data from various models and languages.

Model	Dataset	SISNR	PESQ	STOI	VISQOL
AudioSeal	Seam. (Deu)	23.35	4.244	0.999	4.688
	Seam. (Fr)	24.02	4.199	0.998	4.669
	Voicebox	25.23	4.449	0.998	4.800
WavMark	Seam. (Deu)	38.93	3.982	0.999	4.515
	Seam. (Fr)	39.06	3.959	0.999	4.506
	Voicebox	39.63	4.211	0.998	4.695

Table 10. Evaluation of the detection performances on different datasets. AudioSet is an environmental sounds dataset while ASVspoof (Liu et al., 2023b) and FakeAVCeleb (Khalid et al., 2021) are deep-fake detection datasets.

Dataset	Acc. TPR/FPR	AUC
Audioset	0.9992 0.9996/0.0011	1.0
ASVspoof	1.0 1.0/0.0	1.0
FakeAVCeleb	1.0 1.0/0.0	1.0

C.5. Robustness results

We plot the detection accuracy against the strength of multiple augmentations in Fig. 10. AudioSeal outperforms WavMark for most augmentations at the same strength. However, for highpass filters above our training range (500Hz) WavMark has a much better detection accuracy. Our system's TF-loudness loss embeds the watermark where human speech carries the most energy, typically lower frequencies, due to auditory masking. This contrasts with WavMark, which places the watermark in higher frequency bands. Embedding the watermark in lower frequencies is advantageous. For example, speech remains audible with a lowpass filter at 1500 Hz, but not with a highpass filter at the same frequency. This difference is measurable with PESQ in relation to the original audio, making it more beneficial to be robust against a lowpass filter at a 1500 Hz cut-off than a highpass filter at the same cut-off:

Filter Type	PESQ	AudioSeal	WavMark
Highpass 1500Hz	1.85 ✗	0.7	1.0
Lowpass 1500Hz	2.93 ✓	1.0	0.7

D. Experimental details

D.1. Loudness

Our loudness function is based on a simplification of the implementation in the torchaudio (Yang et al., 2021) library. It is computed through a multi-step process. Initially, the audio signal undergoes K-weighting, which is a filtering process that emphasizes certain frequencies to mimic the human ear's response. This is achieved by applying a treble filter and a highpass filter. Following this, the energy of the audio signal is calculated for each block of the signal. This is done by squaring the signal and averaging over each block. The energy is then weighted according to the number of channels in the audio signal, with different weights applied to different channels to account for their varying contributions to perceived loudness. Finally, the loudness is computed by taking the logarithm of the weighted sum of energies and adding a constant offset.

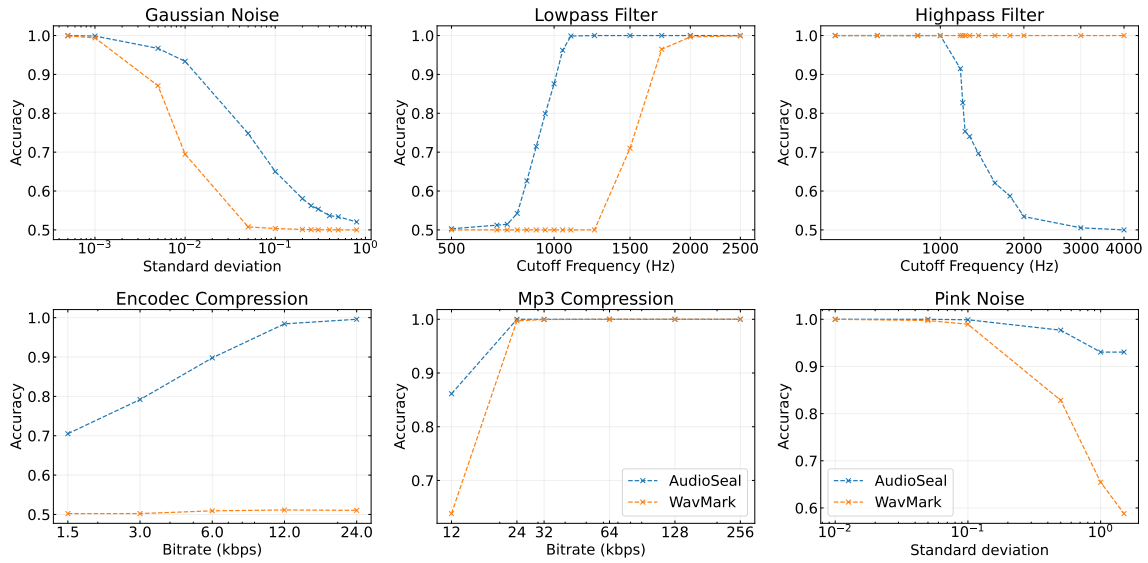

Figure 10. Accuracy of the detector on augmented samples with respect to the strength of the augmentation.

D.2. Robustness Augmentations

Here are the details of the audio editing augmentations used at train time (T), and evaluation time (E):

- **Bandpass Filter:** Combines highpass and lowpass filtering to allow a specific frequency band to pass through. (T) fixed between 300Hz and 8000Hz; (E) fixed between 500Hz and 5000Hz.
- **Highpass Filter:** Uses a highpass filter on the input audio to cut frequencies below a certain threshold. (T) fixed at 500Hz; (E) fixed at 1500Hz.
- **Lowpass Filter:** Applies a lowpass filter to the input audio, cutting frequencies above a cutoff frequency. (T) fixed at 5000Hz; (E) fixed at 500Hz.
- **Speed:** Changes the speed of the audio by a factor close to 1. (T) random between 0.9 and 1.1; (E) fixed at 1.25.
- **Resample:** Upsamples to intermediate sample rate and then downsamples the audio back to its original rate without changing its shape. (T) and (E) 32kHz.
- **Boost Audio:** Amplifies the audio by multiplying by a factor. (T) factor fixed at 1.2; (E) fixed at 10.
- **Duck Audio:** Reduces the volume of the audio by a multiplying factor. (T) factor fixed at 0.8; (E) fixed at 0.1.
- **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.
- **Pink Noise:** Adds pink noise for a background noise effect. (T) standard deviation fixed at 0.01; (E) fixed at 0.1.
- **White Noise:** Adds gaussian noise to the waveform. (T) standard deviation fixed at 0.001; (E) fixed at 0.05.
- **Smooth:** Smooths the audio signal using a moving average filter with a variable window size. (T) window size random between 2 and 10; (E) fixed at 40.
- **AAC:** Encodes the audio in AAC format. (T) bitrate of 128kbps; (E) bitrate of 64kbps.
- **MP3:** Encodes the audio in MP3 format. (T) bitrate of 128kbps; (E) bitrate of 32kbps.
- **EnCodec:** Resamples at 24kHz, encodes the audio with EnCodec with $nq = 16$ (16 streams of tokens), and resamples it back to 16kHz.

Implementation is done with the `julius` python library.

D.3. Networks architectures (Fig. 4)

The watermark generator is composed of an encoder and a decoder, both incorporating elements from EnCodec (Défossez et al., 2022). The encoder applies a 1D convolution with 32 channels and a kernel size of 7, followed by four convolutional blocks. Each of these blocks includes a residual unit and down-sampling layer, which uses convolution with stride S and kernel size $K = 2S$. The residual unit has two kernel-3 convolutions with a skip-connection, doubling channels during down-sampling. The encoder concludes with a two-layer LSTM and a final 1D convolution with a kernel size of 7 and 128 channels. Strides S values are (2, 4, 5, 8) and the nonlinear activation in residual units is the Exponential Linear Unit (ELU). The decoder mirrors the encoder but uses transposed convolutions instead, with strides in reverse order.

The detector comprises an encoder, a transposed convolution and a linear layer. The encoder shares the generator's architecture (but with different weights). The transposed convolution has h output channels and upsamples the activation map to the original audio resolution (resulting in an activation map of shape (t, h)). The linear layer reduces the h dimensions to two, followed by a softmax function that gives sample-wise probability scores.

D.4. MUSHRA protocole detail

The MUSHRA protocol is a crowdsourced test in which participants rate the quality of various samples on a scale of 0 to 100. The ground truth is provided for reference. We utilized 100 speech samples, each lasting 10 seconds. Each sample was evaluated by at least 20 participants. As part of the study, we included a low anchor, which is a very lossy compression at 1.5kbps, encoded using EnCodec. Participants who failed to assign the lowest score to the low anchor for at least 80% of their assignments were excluded from the study. For comparison, the ground truth samples received an average score of 80.49, while the low anchor's average score was 53.21.

D.5. Attacks on the watermark

Adversarial attack against the detector. Given a sample x and a detector D, we want to find $x' \sim x$ such that $D(x') = 1 - D(x)$. To that end, we use a gradient-based attack. It starts by initializing a distortion δ_{adv} with random gaussian noise. The algorithm iteratively updates the distortion for a number of steps n. For each step, the distortion is added to the original audio via $x = x + \alpha.\tanh(\delta_{adv})$, passed through the model to get predictions. A cross-entropy loss is computed with label either 0 (for removal) or 1 (for forging), and back-propagated through the detector to update the distortion, using the Adam optimizer. At the end of the process, the adversarial audio is $x + +\alpha.\tanh(\delta_{adv})$. In our attack, we use a scaling factor $\alpha = 10^{-3}$, a number of steps $n = 100$, and a learning rate of 10^{-1}. The $\tanh$ function is used to ensure that the distortion remains small, and gives an upper bound on the SNR of the adversarial audio.

Training of the malicious detector. Here, we are interested in training a classifier that can distinguish between watermarked and non-watermarked samples, when access to many samples of both types is available. To train the classifier, we use a dataset made of more than 80k samples of 8 seconds speech from Voicebox (Le et al., 2023) watermarked using our proposed method and a similar amount of genuine (un-watermarked) speech samples. The classifier shares the same architecture as AudioSeal's detector. The classifier is trained for 200k updates with batches of 64 one-second samples. It achieves perfect classification of the samples. This is coherent with the findings of Voicebox (Le et al., 2023).