

Learning to Evade: Adaptive Attacks on Audio Watermarking

Weikang Ding^{*1}, Hanqing Guo², Rui Duan¹, Guangjing Wang³, Yuanda Wang⁴, Mingzhe Chen⁵, Qiben Yan⁴

¹ University of Missouri-Kansas City, USA

² University of Hawaii at Manoa, USA

³ University of South Florida, USA

⁴ Michigan State University, USA

⁵ University of Miami, USA

wdhhd@umkc.edu, qyan@msu.edu

Abstract

Advances in generative audio have intensified copyright concerns, making audio watermarking increasingly important for asserting ownership. However, existing audio watermarking methods are vulnerable to adversarial attacks. We find that watermark decoder message probabilities follow normal distributions, a property exploited by defenses to detect manipulations. This paper introduces an adaptive audio watermark attack method (AWM) designed to bypass existing defense strategies. AWM uses a two-stage optimization: the first stage ensures attack success, while the second improves audio quality. To evade detection, it estimates normal distribution parameters from limited samples of the target audio, and then adaptively steers decoded probabilities back into the estimated range. Evaluated on two watermarking methods across three voice datasets, AWM achieves high success while bypassing state-of-the-art detectors: detection rates are below 10% for replacement and creation, and 0% for removal.

Index Terms: Audio watermark, outlier detection, adaptive attack

1. Introduction

In recent years, the rapid growth of social networking platforms has encouraged many users to publicly share their audio content, including original works such as audiobooks and self-produced music. These audio contents might bring them income. However, many unauthorized users copy creative works, modify them, and re-upload them to mainstream platforms for profit, which significantly discourages original audio creators. Besides, voice cloning attacks can illegally synthesize the target’s voice for malicious purposes, potentially resulting in severe consequences such as financial losses and reputation damage [1].

To address these issues, deep-learning-based watermarking has been proposed [2]. It embeds a noise-tolerant signal into the target audio, which remains imperceptible to human hearing while being detectable by specialized AI models. Accordingly, most approaches adopt an encoder-decoder architecture to embed and recover the signal. To improve robustness, training pipelines incorporate training data from distortion attack models, e.g., re-recording [3], voice cloning [4, 5], and lossy codec compression [6], such that the watermark could survive common real-world transformations.

However, the robustness of deep-learning-based audio watermarking methods against adversarial attacks remains a crit-

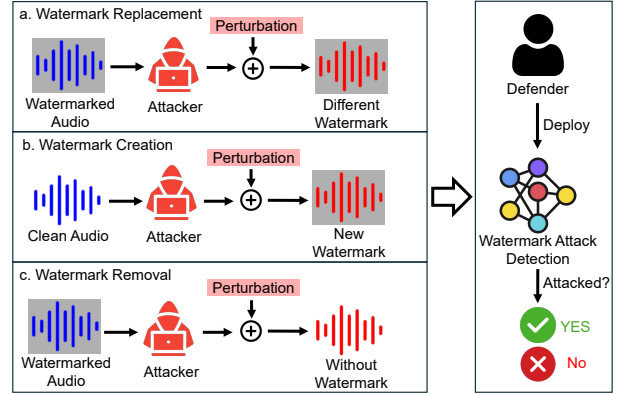


Figure 1: Overview of the watermark attack (left) and the watermark attack detection process used to detect whether the audio has been tampered with (right).

ical concern [7]. Recent studies [8, 9] show that attackers can remove or forge watermarks by embedding carefully crafted adversarial perturbations. These attacks operate by adding and optimizing a perturbation signal so that the watermark decoder is misled into producing incorrect outputs. Yet, existing attack strategies face the following challenges:

C1: *How to design an attack method to balance audio quality and attack effectiveness?* The encoder ensures the watermark is imperceptible, while the decoder recovers the watermark under various distortions. In practice, the decoder is more publicly accessible and therefore becomes the primary source of model knowledge in most attack scenarios. However, deceiving the decoder may introduce certain noise artifacts. An overly aggressive attack can significantly degrade audio quality, while prioritizing perceptual quality may compromise attack success. As a result, it is essential to strike a balance that ensures sufficient attack effectiveness while preserving audio quality.

C2: *How to design an attack to bypass the detection strategy?* For a given audio input, the decoder outputs both a binary message and the associated bit-wise probabilities. These probabilities tend to follow a normal distribution, and defenders can leverage this property to detect whether an audio sample has been tampered with. Consequently, after successfully altering the binary watermark message bits, attackers must further optimize the audio to ensure that the decoded message probabilities fall within the range classified by the defender as non-outliers.

C3: *How to select the suitable audio samples to estimate the decoded message probability distribution?* To estimate the

^{*}Research was performed at Michigan State University.

distribution parameters required for a successful and stealthy attack, they must instead rely on querying and analyzing the decoder’s outputs. Because different input samples induce different probability patterns, the parameters estimated by the attacker can diverge significantly from the defender’s true parameters, and this mismatch can ultimately lead to attack failure. Therefore, it necessitates designing an effective estimation strategy to improve the attack success rate.

In this paper, we propose AWM, an Adaptive audio WaterMark attack method, which is capable of bypassing the defender’s detection strategy. Figure 1 illustrates the application scenarios. The attacker obtains the target audio and adds an adversarial perturbation to generate the perturbed audio. The defender then receives the perturbed audio and uses the watermark decoder to extract the message probabilities. A predefined set of distribution parameters is employed to detect outliers. If any outliers are identified, the audio is classified as “attacked”; otherwise, it is considered “clean”. We propose three schemes to address the challenges above.

To address **C1**, we design a two-step attack framework. The first step focuses on maximizing attack effectiveness while evading the defender’s detection strategy. The second step aims to improve audio quality. Inspired by [10], we introduce a threshold-based post-processing step that refines the perturbation to enhance audio quality, while constraining the decoded message probabilities to remain within the expected normal range.

To address **C2**, we develop an adaptive optimization strategy that explicitly steers the decoded message probabilities toward the estimated normal distribution range. The key idea is to prioritize optimization on probabilities that fall outside this range: if a perturbed bit probability already lies within the estimated normal region, its optimization weight is reduced. This yields a dynamic optimization process that concentrates on the most “at-risk” binary message bits.

To address **C3**, we leverage the observation that data with similar feature distributions tend to produce similar decoded message probability distributions [11, 12]. Accordingly, we estimate the required probability distribution parameters by collecting a limited set of audio samples whose features closely resemble those of the target audio.

In this paper, we make the following contributions:

- We empirically demonstrate that the decoded message probabilities output by the watermark decoder tend to follow normal distributions. This statistical property can be leveraged by defenders to design detection strategies based on outlier detection.
- We propose AWM, an adaptive audio watermark attack method that supports three attack types. Through a two-step optimization framework, AWM can bypass distribution-based detection strategies and preserve audio quality. The adaptive optimization strategy focuses updates on the message bits with the most uncertain probabilities, further improving attack effectiveness.
- We evaluate AWM on three speech datasets and two state-of-the-art watermarking models. Compared to baseline methods, AWM achieves superior performance in both Attack Success Rate (ASR) and Detection Success Rate (DSR). Moreover, even after applying five no-box perturbations, AWM consistently maintains a high ASR, with most scores approaching or reaching 100%. **The source code and demos are available on <https://adaptiveaudiowmattack.github.io/>**

2. Related Work

Deep Learning-Based Audio Watermarking. Unlike traditional schemes [13], which rely on predefined transformations, deep-learning-based schemes can learn complex feature representations and optimize watermarking dynamically [4, 5, 14, 15, 16]. These schemes follow the architecture of the Encoder-Distortion-Decoder. The encoder embeds the watermark into the audio and generates the watermarked audio, the decoder receives the watermarked audio and extracts the corresponding watermark, and the distortion simulates a variety of potential attack scenarios. This paper implements the defense mechanism and attack strategy by utilizing this scheme.

Audio Watermark Attack. Audio watermark attacks based on adversarial perturbations can be categorized as no-box, black-box, or white-box, depending on the attacker’s knowledge of the watermarking model. In no-box perturbations, the attacker relies on generic audio processing techniques (e.g., compression [17], voice conversion [18, 19], and text-to-speech [20, 21]). In black-box perturbations, the attacker can query the watermarking detector but does not have access to its internal architecture or parameters [22, 23]. In white-box perturbations, the attacker has full access to the detector, including its architecture and parameters, and introduces a perturbation to perform the gradient-based attack [24, 25]. While some existing studies [9, 26, 8] are either easily detected or generate the perturbed audio with poor quality. In this paper, we focus on balancing the attack effectiveness and audio quality.

Audio Watermark Attack Detection. Adversarial examples and watermarks both achieve the desired goal by adding imperceptible noise. Different from adversarial examples, adding watermarks to objects has little impact on the performance of the model inference (such as audio classification models [27] and speech recognition models [28]). Therefore, some outlier detection methods based on the time series [29, 30] are ineffective for identifying whether an audio sample has been tampered with. Recent studies have explored detection methods for generated images [31, 32], watermark images [12, 26], and audio deepfake [33, 34]. However, to the best of our knowledge, these approaches cannot be directly applicable to detecting audio watermark attacks.

3. Background

3.1. Preliminary

Adding perturbations to the audio is a common strategy for attacks against watermarking systems. The core idea is to either destroy the original watermark or forge a new one by introducing perturbations that deceive the watermark decoder.

Audio Watermark Decoder. The audio watermark decoder $Dec(\cdot)$ takes the encoded audio as input and outputs the extracted message. The extracted message can be represented in two forms: a binary message and message probabilities. The binary message is a direct representation of the decoded output as a sequence of 0s and 1s, denoted as $m \in \{0, 1\}^N$. The message probabilities provide the decoder’s confidence values for each bit, indicating the likelihood of the bit being 1 or 0. A predefined threshold θ is used to convert probabilities into binary message values: if a probability exceeds the threshold, the binary message bit is decoded as 1; otherwise, it is decoded as 0. The form of message probabilities can be written as:

$$p = Dec(s), \quad (1)$$

where s is clean or watermarked audio, $Dec(\cdot)$ is the watermark

decoder.

Let s_c denote clean audio and s_w denote watermarked audio. The watermark decoder $Dec(\cdot)$ extracts the clean message probabilities p_c from s_c and the watermarked message probabilities p_w from s_w , respectively. The attacker specifies a target message probability, denoted as p_t . A perturbation δ is introduced to perform the attack on the audio.

Watermark Replacement. In watermark replacement, a perturbation δ is added to the watermarked audio s_w , deceiving the decoder into misclassifying the embedded watermark as a different one. Formally, the goal of watermark replacement is:

$$\delta_{replacement} = \arg \min_{\delta} \|Dec(s_w + \delta) - p_t\|. \quad (2)$$

Watermark Creation. Watermark creation involves adding a perturbation δ to a clean audio s_c to generate a perturbed watermarked audio, which deceives the watermark decoder into recognizing it as containing a valid watermark. Formally, the goal of watermark creation is:

$$\delta_{creation} = \arg \min_{\delta} \|Dec(s_c + \delta) - p_t\|. \quad (3)$$

Watermark Removal. Watermark removal is an untargeted attack aimed at removing the embedded watermark. We introduce a watermark detector, $Detector(\cdot)$, which determines whether the audio contains a watermark. A perturbation δ is added to the watermarked audio s_w to mislead the detector into classifying the perturbed audio as unwatermarked (i.e., clean). Formally, the goal of watermark removal can be expressed as:

$$\delta_{removal} = \arg \min_{\delta} \|\delta\| \text{ s.t. } Detector(s_w + \delta) = \text{None}. \quad (4)$$

3.2. Message Probability Distribution

Benign Data Distribution. The watermark decoder outputs the message probabilities for each binary message bit. A bit is decoded as 1 if its probability exceeds a predefined threshold; otherwise, it is decoded as 0. For both clean and watermarked audio, we observe two distinct distributions, each following a normal distribution pattern. Clean audio exhibits a unimodal normal distribution, with the peak density centered near the predefined threshold. Specifically, in the Timbre, message probabilities below 0 are decoded as 0, while message probabilities above 0 are decoded as 1. As shown in Figure 2(a), where the threshold is set to 0, the mean μ of the distribution is also close to 0. In contrast, the watermarked audio follows a bimodal, approximately normal distribution (Figure 2(b)), with two peaks corresponding to the decoded binary message values 0 and 1.

Audio Watermark Attack Distribution. For existing watermark attack methods [8], we observe that the distributions of message probabilities deviate significantly from the benign data distributions. Figure 3(a) illustrates the distribution of message probabilities under watermark removal attacks. Compared to Figure 2(a), the message distribution under attack differs in the range of message probabilities along the x-axis, even though it still exhibits a unimodal normal distribution. Figure 3(b) presents the distribution of watermark creation attacks. Here, most message probability values cluster around the threshold value of 0. In contrast to the clear bimodal normal distribution shown in Figure 2(b), the distribution resulting from the attack visually diverges from the benign data distribution.

3.3. Threat Model

The attackers perform watermark attacks to either claim new copyright ownership or remove the original copyright. We as-

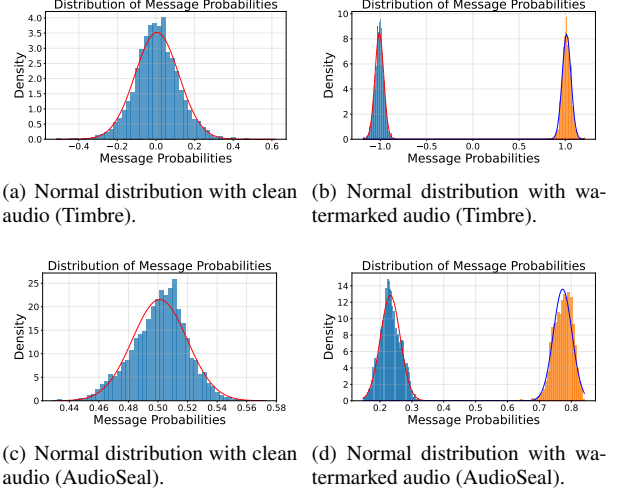


Figure 2: Distribution of benign message probabilities.

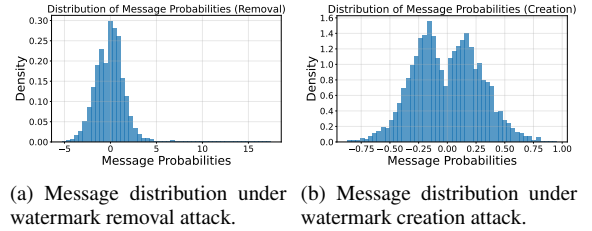


Figure 3: Distribution of message probabilities under attacks (Timbre). We use the AudioMarkBench to perform the attacks.

sume that: 1) attackers have no access to the data used by defenders to fit the distribution, nor to the audio dataset used to train the watermarking model; 2) they have no access to the ground-truth watermark message embedded in the target audio; but they can get the message probabilities from the watermark decoder, allowing them to perform gradient-based attacks; 3) they do not possess the complete watermarking model, but they can embed watermarks into a small set of clean audio samples; 4) they are aware that the decoded message probabilities output by the watermark decoder follow a normal distribution, but they do not know the corresponding mean and standard deviation.

The defenders are the attack detectors, who identify whether audio has been tampered with. We assume that: 1) defenders have access to a large number of ground-truth audio samples, which are used to fit a normal distribution and estimate the mean and variance via maximum likelihood estimation; 2) the watermarking model is publicly available through an online platform for commercial use, with a limit on the number of watermarked audio samples that each user can generate per day.

4. Methodology

In this section, we first introduce a detection method based on outlier detection to identify if the given audio sample has been attacked. Second, we design our adaptive attack, which aims to achieve a successful attack while preserving perceptual quality. Meanwhile, we describe the adaptive optimization process for the three attack types.

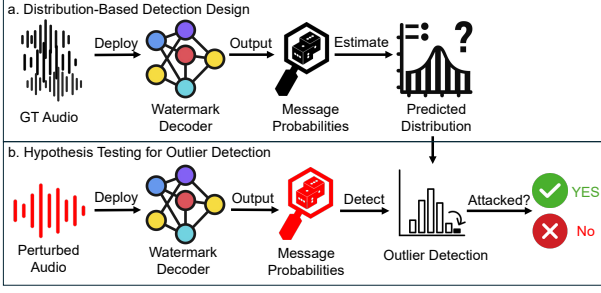


Figure 4: Audio watermark attack detection. The defender uses ground truth (GT) audio (watermarked and clean) to estimate the reference distribution (top) and applies outlier detection with respect to this distribution to determine whether an audio sample has been attacked (bottom).

4.1. Audio Watermark Attack Detection

As demonstrated in Section 3.2, the distribution of message probabilities in perturbed audio exhibits significant deviations from that of benign audio, thereby providing the foundation for our proposed detection mechanism to effectively distinguish between the two. Figure 4 illustrates the audio watermark attack detection method, which introduces a two-step approach.

Distribution-Based Detection Design. The key intuition is that message probabilities from clean and watermarked audio follow different statistical patterns. Building on this, we propose that the defenders can collect ground-truth audio, extract message probabilities with the watermark decoder, and apply maximum likelihood estimation to model their distributions. As shown in Figure 2, this process yields three normal distributions: one derived from message probabilities of clean audio and two from those of watermarked audio.

Specifically, given n audio samples, each producing a probability vector of size $1 \times N$ message probabilities, the complete set of message probabilities is denoted as:

$$\mathbf{p} = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1N} \\ p_{21} & p_{22} & \cdots & p_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ p_{n1} & p_{n2} & \cdots & p_{nN} \end{bmatrix} \in \mathbb{R}^{n \times N}.$$

The mean μ and standard deviation σ of the normal distribution estimated via maximum likelihood are:

$$\mu = \frac{1}{nN} \sum_{i=1}^n \sum_{j=1}^N p_{ij}, \quad \sigma^2 = \frac{1}{nN} \sum_{i=1}^n \sum_{j=1}^N (p_{ij} - \mu)^2. \quad (5)$$

Finally, the defender obtains the corresponding means μ and standard deviations σ , which are used to detect outliers in suspicious audio and determine whether it has been attacked.

Hypothesis Testing for Outlier Detection. We formalize detection as an outlier detection problem under a statistical hypothesis testing framework. Given a potentially perturbed audio sample, the defender uses the watermark decoder to extract its message probability set $\{p_i\}$, where each element is assumed to follow a normal distribution with mean μ and standard deviation σ . We then test the null hypothesis $H_0 : p_i \sim \mathcal{N}(\mu, \sigma^2)$ against the alternative $H_1 : p_i \not\sim \mathcal{N}(\mu, \sigma^2)$. If H_0 is rejected, the audio sample is classified as attacked.

To test the hypothesis, we first compute the z-score for each extracted message probability:

$$z_i = \frac{p_i - \mu}{\sigma}. \quad (6)$$

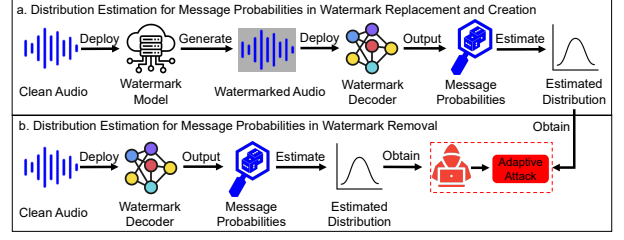


Figure 5: The distribution estimation by the attacker. (a) Watermark replacement and creation: The attacker uses a small set of clean audio samples to generate watermarked audio samples, which are then used to estimate the distribution. (b) Watermark removal: The attacker directly uses the clean audio samples to estimate the distribution.

Under the null hypothesis, z_i follows the standard normal distribution, and the corresponding two-tailed p-value is given by $P_i = 2(1 - \Phi(|z_i|))$, where Φ denotes the cumulative distribution function of the standard normal distribution. The decision rule is to reject H_0 if $P_i < \alpha$, with α as the predefined significance level. An audio sample is deemed being “attacked” if any extracted message probability p_i rejects H_0 .

4.2. Adaptive Attack Design

We propose an adaptive attack designed to bypass message-probability-distribution-based detection. First, the attacker prepares for the attack by estimating the defender’s distribution. Second, the attacker performs the watermark attack, which contains two stages: (1) modify the original audio to bypass the defense and achieve a successful attack, and (2) improve the quality of the perturbed audio, while maintaining that the decoded message probabilities remain within an acceptable range.

4.2.1. Attack Preparation: Estimate Defender’s Distribution

The adversary’s goal is to estimate the mean μ_{est} and the standard deviation σ_{est} of the decoded message probabilities. These parameters are categorized into two groups: watermark replacement and creation, and watermark removal. The process is illustrated in Figure 5.

Parameter Estimation for Watermark Replacement and Creation. The attackers first select several clean audio samples s_c from a small dataset, which are used to query the watermark model $Enc(\cdot)$ and generate new watermarked audio samples s_w . The watermark decoder $Dec(\cdot)$ is deployed to extract message probabilities, which are subsequently used to estimate the parameters of the normal distribution. Since the distribution of watermarked message probabilities is bimodal (as shown in Figure 2(b) and 2(d)), the attacker can obtain two distributions. The estimated mean and standard deviation are as follows:

$$\begin{aligned} \mu_{est}^0, \sigma_{est}^0 &= T^0(Dec(Enc(s_c))), \\ \mu_{est}^1, \sigma_{est}^1 &= T^1(Dec(Enc(s_c))), \end{aligned} \quad (7)$$

where μ_{est}^0 and σ_{est}^0 represent the estimated mean and standard deviation of decoded message probabilities predicted as 0, and μ_{est}^1 and σ_{est}^1 correspond to those predicted as 1. $T(\cdot)$ denotes the method used to estimate the parameters of the normal distribution, which depends on the attacker’s prior knowledge and the amount of available data. Common choices include Bayesian inference [35] and maximum likelihood estimation. We use $T^0(\cdot)$ and $T^1(\cdot)$ to denote the estimation procedures for the distributions corresponding to bits 0 and 1, respectively.

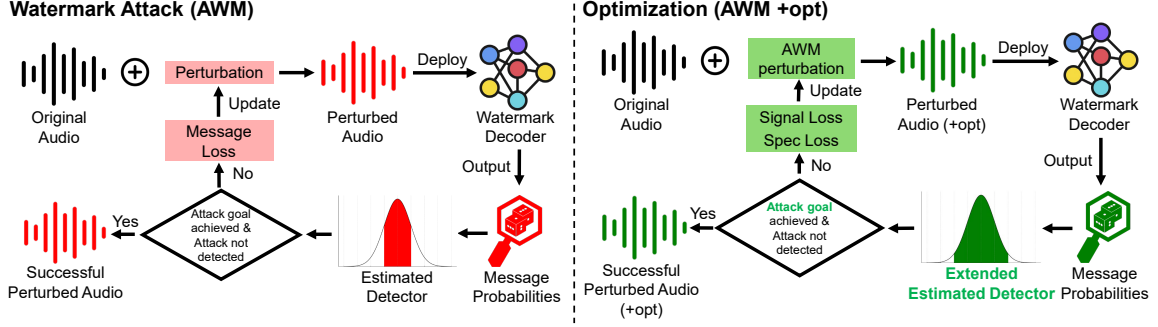


Figure 6: The design of AWM generator. The audio watermark attack step (left) ensures the success of the watermark attack, while the audio quality optimization step (right) focuses on improving audio quality.

Parameter Estimation for Watermark Removal. The attackers estimate the distribution directly using clean audio samples s_c . They use the watermark decoder to output message probabilities, which are used for distribution estimation. Since the estimated distribution is unimodal (as shown in Figure 2(a) and 2(c)), the parameters are defined as follows:

$$\mu_{est}^c, \sigma_{est}^c = T^c(Dec(s_c)), \quad (8)$$

where μ_{est}^c and σ_{est}^c are the estimated mean and standard deviation of the clean message probabilities, and T^c represents the estimation approach applied to these values.

4.2.2. Audio Watermark Attack (AWM)

After obtaining the estimated distribution, the attacker then adds a small adversarial perturbation to the original audio s , aiming to deceive the decoder into outputting incorrect binary messages while bypassing the detection strategy. Figure 6-left illustrates the watermark attack step. The original audio is clean audio in watermark creation attack, and it can be watermarked audio in watermark replacement or watermark removal attack. First, the perturbation is initialized by a fraction of the original audio signal. Next, the attacker adds the perturbation to original audio and obtains a perturbed audio, which is passed through the watermark decoder to obtain message probabilities and subsequently queried by the estimated detector. If the attack goal is achieved and the attack is not detected, the attacker obtains the successful perturbed audio. Otherwise, the attacker further optimizes the perturbation by message loss. The message loss $\mathcal{L}_{msg}$ modifies the perturbed message probabilities to match the target message probabilities p_t :

$$\mathcal{L}_{msg} = \|Dec(s_{att}) - p_t\|_2^2. \quad (9)$$

Let s_{att} denote the perturbed audio, this loss enforces the decoded message to closely match the target message.

Different from the prior attacks, our loss optimization step follows a strict bit-to-bit optimization design (detailed in Algorithm 1). This algorithm uses the estimated detector knowledge to ensure that the perturbed audio exhibits a distribution similar to that of benign audio, with high confidence. Importantly, our algorithm constrains the message probabilities within the normal value range. Through iterative updates and perturbation optimization, the perturbed message probabilities are gradually adjusted to fall within a strict range, ensuring that they are classified as non-outliers and yielding the final adversarial perturbation. Our results show that constraining the message probabilities of the perturbed audio to the interval $[\mu_{est} - \sigma_{est}, \mu_{est} + \sigma_{est}]$ improves the attack success rate.

Besides the message loss, we also formulate the signal loss and mel loss to minimize the quality degradation from the attack. Specifically, the signal loss controls the audio quality at the signal level:

$$\mathcal{L}_{signal} = \frac{1}{n} \sum_{i=1}^n |s_{att} - s|. \quad (10)$$

The Mel-Spectrogram loss $\mathcal{L}_{mel}$ maintains the audio quality at the Mel-Spectrogram level:

$$\mathcal{L}_{mel} = \|Mel(s_{att}) - Mel(s)\|_2^2. \quad (11)$$

The total loss in the attack step is:

$$\mathcal{L} = \lambda_1 \mathcal{L}_{signal} + \lambda_2 \mathcal{L}_{mel} + \lambda_3 \mathcal{L}_{msg} + \lambda_4 \mathcal{L}_{other}, \quad (12)$$

where the $\mathcal{L}_{other}$ depends on the specific watermarking method. For example, AudioSeal [5] includes a localization loss used for indicating the probability of the audio being watermarked. In the AWM attack process, the parameter λ_{msg} is assigned a relatively high value.

4.2.3. Audio Quality Optimization (AWM +opt)

Since audio watermark attacks prioritize message loss optimization, audio quality may be adversely impacted. This step aims to improve audio quality while maintaining a successful attack. To achieve this, we make three adaptations.

Figure 6-right illustrates the three steps. First, the attacker takes the perturbation obtained in the earlier stage as the initial perturbation and uses the watermark decoder to extract the corresponding message probabilities. These probabilities are then fed into the estimated distribution to expand the allowable range, which we suggest setting to $[\mu_{est} - 2\sigma_{est}, \mu_{est} + 2\sigma_{est}]$. This step is crucial for trading off audio quality against attack success rate. In the original AWM setting, the attacker enforces a strict constraint to guarantee attack success, but this rigidity limits the flexibility of the perturbation and makes it difficult to preserve audio quality. In this design, we relax the constraint by expanding the permissible range for message probability optimization, thereby providing more space for perturbation adjustment and achieving a more balanced attack that accounts for both perceptual quality and attack success rate.

Second, in addition to extending the acceptable range, we also update the attack objective by enforcing optimization to proceed for a fixed number of epochs. This ensures that the perturbation is fully optimized rather than stopping at a boundary case.

Algorithm 1 Adaptive Attack in Watermark Replacement

Input: Watermarked audio s_w , target message probabilities p_t , scale factor r , supremum and infimum of thresholds for decoded message 0 and 1 τ_{sup}^0 and τ_{infi}^0 , τ_{sup}^1 and τ_{infi}^1 , list of different message probabilities $msgdiff$

Output: perturbed watermarked audio $\hat{s}_w$

```
1:  $\delta = s_w \times r, p_w = Dec(s_w), \hat{s}_w = s_w + \delta$ 
2: for  $index = 1, 2, \dots, (len(p_t) - 1)$  do
3:   if  $index$  not in  $msgdiff$  then
4:      $p_t[index] = p_w[index]$ 
5: for  $i = 1, 2, \dots, Iter$  do
6:    $\delta = Attack(s_w, \hat{s}_w, Dec(\hat{s}_w), p_t, \delta)$ 
7:    $\hat{s}_w = s_w + \delta$ 
8:    $\hat{p}_w = Dec(\hat{s}_w)$ 
9:   if  $acc == 1$  then
10:    for  $index$  in  $msgdiff$  do
11:      if  $\tau_{infi}^1 < \hat{p}_w[index] < \tau_{sup}^1$  then
12:         $p_t[index] = \hat{p}_w[index]$ 
13:        Remove  $msgdiff[index]$ 
14:      if  $\tau_{infi}^0 < \hat{p}_w[index] < \tau_{sup}^0$  then
15:         $p_t[index] = \hat{p}_w[index]$ 
16:        Remove  $msgdiff[index]$ 
17:   if meet the estimated detection  $Detection(\hat{p}_w)$  then
18:     return  $\hat{s}_w$ 
19: return Failed
```

Third, we modify the optimization loss by replacing the Mel-spectrogram loss with a standard spectrogram loss and applying a softmax function to the spectrogram. Inspired by AudioSeal [5], this softmax adaptation better preserves loudness and perceptual similarity between the two audio signals. The resulting softmax-based spectrogram loss, denoted as $\mathcal{L}_{spec}$, is defined as:

$$\mathcal{L}_{spec} = \frac{1}{n} \sum_{i=1}^n |Softmax(S_{att}) - Softmax(S_s)|, \quad (13)$$

where S_{att} and S_s are the spectrograms of the attacked audio and original audio. In the optimization (+opt) process, the parameters λ_1 and λ_2 are assigned a relatively high value.

4.2.4. Adaptive Attack for Watermark Replacement, Creation, and Removal

During the watermark attack, we prioritize refining the decoded message probabilities that fall outside the estimated normal range. Algorithm 1 outlines the adaptive attack process used in watermark replacement. For example, suppose the original watermark bits are “101100”, which are modified to “111000”. We define a list, $msgdiff$, which contains the indices of binary message bits differing between the original watermarked audio s_w and the perturbed watermarked audio $\hat{s}_w$ ¹. In this case, $msgdiff = [1, 3]$. For the indices not included in $msgdiff$, we assign the original watermark message probabilities p_w to the corresponding target watermark message probabilities p_t . That is, for indices $[0, 2, 4, 5]$, the p_t is equal to the p_w .

This optimization has two advantages: (1) It directs the gradient to focus more on the indices where the binary message bits

require modification. In watermark replacement, only the differing bits need to be changed, so bits that already align with the target message do not require further optimization. (2) It ensures that the message probabilities in the non-change bits remain the same, and within the acceptable distribution. In certain audio watermarking methods, message probabilities do not directly map to binary message values. In Figure 2(d), probabilities corresponding to binary 1 lie in the range of 0.7–0.8, and those for binary 0 fall between 0.2–0.3. Therefore, it is not appropriate to optimize message probabilities directly to 1 or 0. Instead, they should be adjusted to fall within the range of 0.7–0.8 for a binary message bit of 1, and 0.2–0.3 for a bit of 0. Then, we optimize the perturbation δ by Function $Attack(\cdot)$ (Line 6) to generate the perturbed watermarked audio $\hat{s}_w$. The $Attack(\cdot)$ calculates the loss in Equation (12) and use the gradient to update the perturbation. The initial perturbation is scaled based on the original watermarked audio s_w to ensure that the decoded message probabilities of the perturbed audio $\hat{s}_w$ closely resemble those of the original. In the attack justification step, we define the supremum and infimum thresholds for the watermark decoder outputs corresponding to binary message bits. Specifically, τ_{sup}^0 and τ_{infi}^0 represent the thresholds for bit 0, while τ_{sup}^1 and τ_{infi}^1 correspond to bit 1:

$$\begin{aligned} (\mu_{est}^0 - \sigma_{est}^0) &\leq \tau_{infi}^0 < \tau_{sup}^0 \leq (\mu_{est}^0 + \sigma_{est}^0), \\ (\mu_{est}^1 - \sigma_{est}^1) &\leq \tau_{infi}^1 < \tau_{sup}^1 \leq (\mu_{est}^1 + \sigma_{est}^1). \end{aligned} \quad (14)$$

Once the message probability of the perturbed watermarked audio at a given index falls within the specified threshold range, it is assigned to the target message probability p_t , and the index is removed from the list $msgdiff$. Optimization then proceeds with the remaining message probabilities in the list. To ensure that all perturbed message probabilities remain within the estimated normal range, the attacker simulates the defender’s role by performing outlier detection.

For watermark creation attack, the adaptive attack process is the same as that of watermark replacement. The difference is that, in watermark creation, $msgdiff$ includes all message indices, which is equal to the full binary message length. Watermark removal attack is an untargeted attack, it is not necessary to achieve an accuracy of exactly 0, any value below 1 is sufficient. Using a lower accuracy threshold requires more iteration steps to optimize the perturbation δ . We recommend using an accuracy threshold around 0.5.

5. Evaluation

5.1. Experimental Setup

Datasets. We use three public datasets for our experiments. The first dataset is the LibriSpeech [36]. We select the small-sized subset, which has 6.3G audios, and covers 100.6 hours of audio data spoken by 251 speakers. The second dataset is obtained from AudioMarkData [8], which is built based on the Common Voice dataset [37]. It contains 20,000 audio samples, each with a duration of 5 seconds. The third dataset is GigaSpeech [38], which includes audio from audiobooks, podcasts, and YouTube. We use the XS subset, which contains a total of 10 hours of audio samples.

Audio Watermark Methods. We select two state-of-the-art audio watermarking methods: Timbre [4] and AudioSeal [5]. We fix the binary message length to 16 bits for all experiments.

Evaluation Metrics. First, we introduce the **Detection Success Rate (DSR)**, which measures the ability to identify outliers

¹This notation is defined for the watermark replacement scenario. The general representation of perturbed audio is denoted as s_{att} , which refers either to perturbed clean audio (used for creation) or to perturbed watermarked audio (used for replacement or removal).

Table 1: Detection performance across different datasets, watermark methods, and attack methods. ‘-’ indicates that no successfully perturbed audio is available.

Attack Type	Watermark Method	Attack Method	LibriSpeech			Audiomark			GigaSpeech		
			DSR (%)	FAR (%)	FI (%)	DSR (%)	FAR (%)	FI (%)	DSR (%)	FAR (%)	FI (%)
Watermark Replacement	AudioSeal	AudioMarkBench	97.71	4.20	96.79	100.00	5.50	97.32	100.00	6.00	96.94
		AudioM. (+LP)	93.94	2.27	95.75	93.98	4.82	89.66	96.05	8.45	94.19
		AudioM. (+AS)	97.96	4.08	96.97	100.00	5.23	97.45	99.61	5.85	97.53
		AudioM. (+GN)	98.03	1.97	98.03	100.00	10.81	94.87	98.11	7.14	98.85
		AudioM. (+MP3)	98.47	4.20	97.08	100.00	7.23	96.51	96.93	3.40	96.95
		AudioM. (+HP)	86.41	3.88	90.82	96.43	5.36	95.58	68.75	4.76	79.49
		Ours	3.44	4.20	6.40	8.50	5.50	15.00	11.00	6.00	18.81
		Ours (+opt)	5.34	4.20	9.75	8.00	5.50	14.10	17.67	6.00	28.58
	Timbre	AudioMarkBench	100.00	2.67	98.68	100.00	6.50	96.85	100.00	8.33	96.00
		AudioM. (+LP)	100.00	0.00	100.00	100.00	9.09	95.65	100.00	0.00	100.00
		AudioM. (+AS)	100.00	2.82	98.61	100.00	4.48	97.81	100.00	7.64	96.49
		AudioM. (+GN)	-	-	-	100.00	0.00	-	-	-	-
		AudioM. (+MP3)	100.00	0.00	100.00	100.00	0.00	100.00	-	-	-
Watermark Creation	AudioSeal	AudioM. (+HP)	100.00	1.28	99.36	100.00	2.94	98.55	100.00	4.00	98.11
		Ours	1.53	2.67	2.93	7.00	6.50	12.33	6.67	8.33	11.58
		Ours (+opt)	1.91	2.67	3.64	7.50	6.50	13.17	8.33	8.33	14.29
	Timbre	AudioMarkBench	100.00	4.20	97.94	100.00	5.50	97.32	100.00	6.00	97.09
		AudioM. (+LP)	100.00	4.50	97.61	100.00	2.82	98.61	100.00	9.00	95.96
		AudioM. (+AS)	100.00	4.50	97.79	100.00	3.97	98.05	100.00	6.76	97.18
		AudioM. (+GN)	100.00	5.22	97.46	100.00	0.00	100.00	100.00	14.89	93.33
		AudioM. (+MP3)	100.00	5.45	97.35	100.00	4.76	97.67	100.00	11.76	94.74
Watermark Removal	AudioSeal	AudioM. (+HP)	100.00	1.28	99.36	100.00	2.94	98.55	100.00	4.00	98.11
		Ours	0.76	4.20	1.45	0.50	5.50	0.94	0.33	6.00	0.63
		Ours (+opt)	1.91	4.20	3.60	2.00	5.50	3.68	13.00	6.00	21.85
	Timbre	AudioMarkBench	100.00	2.67	98.68	100.00	6.50	96.85	100.00	8.33	96.00
		AudioM. (+LP)	100.00	0.00	100.00	100.00	5.31	97.41	100.00	6.00	97.30
		AudioM. (+AS)	100.00	2.31	98.86	100.00	6.40	96.90	100.00	8.25	96.59
		AudioM. (+GN)	100.00	0.00	100.00	100.00	33.33	85.71	100.00	0.00	100.00
		AudioM. (+MP3)	100.00	3.70	98.18	100.00	3.70	98.18	100.00	4.55	97.87
Watermark Removal	AudioSeal	AudioM. (+HP)	100.00	1.28	99.36	100.00	2.94	98.55	100.00	4.00	98.11
		Ours	0.00	2.67	0.00	0.00	6.50	0.00	0.00	8.33	0.00
		Ours (+opt)	0.00	2.67	0.00	0.00	6.50	0.00	0.00	8.33	0.00
	Timbre	AudioMarkBench	92.75	5.73	93.46	100.00	3.50	98.28	99.67	5.67	97.03
		AudioM. (+LP)	91.41	5.86	93.62	100.00	3.30	98.38	99.64	4.57	97.98
		AudioM. (+AS)	91.92	5.77	93.03	100.00	2.75	98.65	99.63	5.14	97.86
		AudioM. (+GN)	89.92	5.43	92.00	100.00	2.96	98.55	99.63	4.78	97.95
		AudioM. (+MP3)	83.92	5.88	88.25	100.00	2.76	98.63	98.87	4.81	97.44
Watermark Removal	AudioSeal	AudioM. (+HP)	83.92	5.88	88.25	100.00	3.39	98.33	98.17	5.58	96.95
		Ours	0.00	5.73	0.00	0.00	3.50	0.00	0.00	5.67	0.00
		Ours (+opt)	0.00	5.73	0.00	0.00	3.50	0.00	0.00	5.67	0.00
	Timbre	AudioMarkBench	100.00	5.73	97.27	100.00	6.00	97.09	100.00	8.67	95.83
		AudioM. (+LP)	100.00	5.73	97.22	100.00	6.00	97.09	100.00	8.98	96.50
		AudioM. (+AS)	100.00	5.73	97.22	100.00	6.00	97.09	100.00	8.98	96.50
		AudioM. (+GN)	84.13	5.56	81.54	74.83	7.29	82.20	85.04	9.73	88.09
		AudioM. (+MP3)	100.00	5.73	97.22	100.00	6.00	97.09	98.67	8.98	95.80
Watermark Removal	AudioSeal	AudioM. (+HP)	100.00	5.73	97.22	100.00	6.00	97.09	100.00	8.98	96.50
		Ours	0.00	5.73	0.00	0.00	6.00	0.00	0.00	8.67	0.00
		Ours (+opt)	0.00	5.73	0.00	0.00	6.00	0.00	0.00	8.67	0.00
	Timbre	AudioMarkBench	100.00	5.73	97.27	100.00	6.00	97.09	100.00	8.67	95.83
		AudioM. (+LP)	100.00	5.73	97.22	100.00	6.00	97.09	100.00	8.98	96.50
		AudioM. (+AS)	100.00	5.73	97.22	100.00	6.00	97.09	100.00	8.98	96.50
		AudioM. (+GN)	84.13	5.56	81.54	74.83	7.29	82.20	85.04	9.73	88.09
		AudioM. (+MP3)	100.00	5.73	97.22	100.00	6.00	97.09	98.67	8.98	95.80

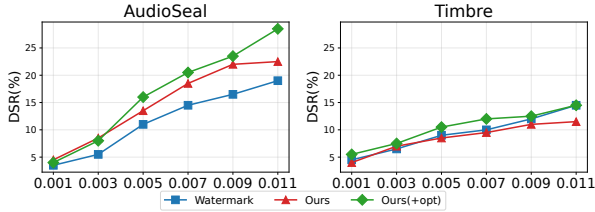


Figure 7: DSR of the watermark replacement attacks under different significance levels α .

in the decoded message probabilities. In this experiment, the defender defines the predefined significance level α as 0.003. Second, we use the **False Acceptance Rate (FAR)**, which evaluates the rate that unattacked audio is mistakenly classified as attacked. Third, we employ the **Attack Success Rate (ASR)** to measure how successfully the adaptive attackers perform the watermark attack, which also corresponds to the watermark decoder’s accuracy at the binary message level.

For the audio quality metrics, we use the **Signal-to-noise ratio (SNR)** and **VisQOL** [39]. In the experiment, we use clean audio samples as a baseline.

Audio Watermark Attack Methods. We compare our attack method with the AudioMarkBench [8]. AudioMarkBench is a benchmark to evaluate the robustness of audio watermarking against watermark replacement, creation, and removal attacks. We use the default settings if there is no further mention. Our method consists of two attack variants: Ours (AWM) (Section 4.2.2), which includes only the watermark attack step, and Ours (+opt) (AWM +opt) (Section 4.2.3), which includes an additional audio quality optimization step.

5.2. Attack Detection Performance

In our experiments, we evaluate our detection method described in Section 4.1 against the attacks provided in AudioMarkBench [8], as well as five additional perturbations applied to

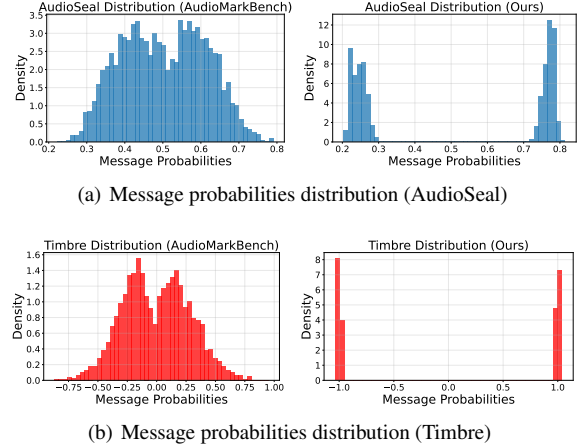


Figure 8: Message probabilities distribution comparisons between AudioMarkBench and AWM for the watermark creation.

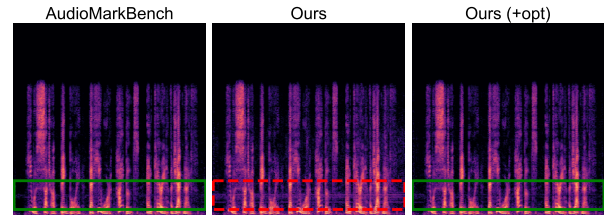


Figure 9: The spectrograms of the watermark creation in AudioSeal. The dotted red box shows noticeable noise for AWM attack. The green box indicates that AWM (+opt) attack achieves higher audio quality and is visually similar to AudioMarkBench.

the attacked audio: (1) low-pass filtering (LP): fixed at 5000Hz, (2) amplitude scaling (AS): fixed at scale of 0.9, (3) Gaussian noise (GN): fixed at 40 dB, (4) MP3 compression (MP3): fixed at 32kbps, and (5) high-pass filtering (HP): fixed at 250Hz on AudioSeal and fixed at 500Hz (replacement and removal) and 1000Hz (creation) on Timbre. The detection results are reported in Table 1, where the FAR remains within an acceptable range, with most values around 5%.

Across all three attack types and eight attack methods, AWM consistently achieves superior evasive performance compared to the baselines. Moreover, the DSR of our watermark attack (Ours) is lower than that of the variant with audio-quality optimization, Ours(+opt). For watermark replacement and creation, most DSR values remain below 10%, and the best performance is observed in watermark removal, where none of the perturbed audio samples are detected by the defenders. These results suggest that watermark removal is the most effective attack strategy, while watermark replacement is the most challenging among the three evaluated attack types.

5.3. Impact of Different α

As a defender, it is essential to select an appropriate significance level α (as presented in Section 4.1) for balancing detection and false acceptance. A higher α increases the FAR, while a lower α risks missing detection for the attacked samples. Figure 7 shows the DSR under different α values on the Audiomark dataset for the watermark replacement attack. Notably, the DSR of benign watermark audio samples (“Watermark”) is equal to the FAR. We observe that the DSR of the attack “Ours” is lower than that

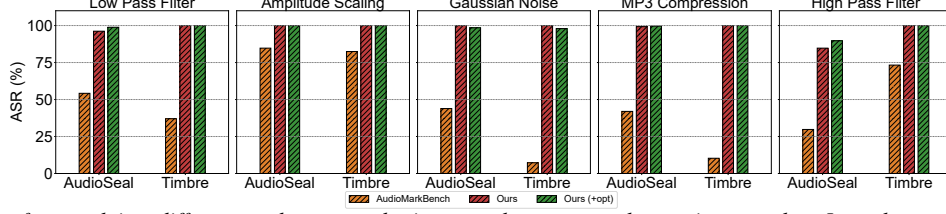


Figure 10: ASR after applying different no-box perturbations on the watermark creation attacks. It evaluates the ASR results of watermark creation against the no-box perturbations, which uses the Librispeech dataset for validation.

of the attack variant “Ours (+opt)”, which indicates that improving the audio quality will increase the detection risk. Since the DSR of the attacked audio closely matches that of ground-truth watermarked audio, AWM attack remains highly stealthy, making it difficult for defenders to detect regardless of the data size or significance level used.

5.4. Distribution Analysis of AWM

We randomly select some audio samples from the Librispeech dataset, and use the AudioMarkBench and our attack method to generate the distribution of message probabilities. Figure 8 illustrates the normal distributions for both the AudioSeal and Timbre. In AudioSeal, the predefined threshold θ for converting probabilities into binary message values is 0.5, and in Timbre, it is 0. The results from AudioMarkBench show that most message probabilities cluster around the predefined threshold, and the overall distribution tends to exhibit a unimodal shape. Therefore, the detection method is able to identify that the audio has been tampered with. In contrast, for our method, the resulting “normal” distribution becomes bimodal, with a shape similar to that shown in Figure 2(b) and 2(d). As a result, our attack successfully bypasses this detection strategy.

5.5. Perturbation Visualization

To analyze the attack visually, we generate spectrograms of the audio samples. Figure 9 presents the visualization using dB-scaled spectrograms of the watermark creation attack. In the spectrograms produced by AudioMarkBench and AWM’s default attack method, we can clearly observe some noticeable noise (horizontal lines), which is highlighted with dotted red boxes. Since watermark attacks target only the decoder, some noise may not be optimized well. Comparing the attack methods AWM (Ours) and AWM +opt (Ours +opt), we observe that the audio quality improves and some of the noticeable noise is effectively reduced through optimization. Besides, AWM +opt is more visually similar to AudioMarkBench (green boxes).

5.6. Robustness Against Perturbations

Robustness of the attack in watermarking refers to whether the watermark remains detectable after applying no-box perturbations to the watermarked audio. Watermark creation attack, which aims to modify the binary message bits to the specific target binary message bits while bypassing the defender’s detection across all bits, requires complex and robust considerations. Therefore, we add no-box perturbations to the perturbed watermarked audio and observe if the forged watermark can be detected. Figure 10 evaluates the robustness of the watermark creation attack against no-box perturbations. The results show that AWM attack is more resilient against the no-box perturbations, with most ASR achieving around 100%. For comparing the ASR between AWM and AWM (+opt), AWM attack with-

Table 2: Audio quality comparisons in AudioSeal and Timbre.

Metric	Watermark Method	Attack Type	Method			
			w/o attack	AudioMarkBench	Ours	Ours(+opt)
SNR	AudioSeal	Replacement	26.63	25.40	23.89	26.27
		Creation	26.63	43.72	25.94	37.33
		Removal	26.63	26.49	25.49	26.59
	Timbre	Replacement	25.04	25.05	24.82	25.03
		Creation	25.04	63.85	38.19	47.84
		Removal	25.04	24.90	24.55	25.03
ViSQOL	AudioSeal	Replacement	4.92	4.59	4.34	4.70
		Creation	4.92	4.83	4.26	4.65
		Removal	4.92	4.68	4.43	4.69
	Timbre	Replacement	4.66	4.65	4.39	4.56
		Creation	4.66	5.00	4.67	4.83
		Removal	4.66	4.52	4.41	4.61

out optimization demonstrates slightly higher robustness.

5.7. Audio Quality

We evaluate audio quality using SNR and ViSQOL, with results shown in Table 2. From the results, we observe that watermark creation shows a significant difference. As shown in our previous experiments in Figure 10, although AudioMarkBench successfully alters the watermark binary message, the attack lacks robustness. The watermark creation attack aims to modify the clean binary message into a targeted adversarial message. When the attack prioritizes audio quality, the features of the perturbed audio remain very similar to those of the clean audio, which also implies that the embedded watermark features are relatively weak. Applying a no-box perturbation is more likely to remove these weakened watermark features. Thus, balancing audio quality and attack robustness becomes a critical design consideration. Our previous experiments on the creation attack show that: while our approach yields slightly lower audio quality, it achieves substantially higher robustness.

In the watermark replacement and removal, the audio quality of AWM attack is comparable to that of AudioSeal/Timbre and AudioMarkBench. AWM exhibits the lowest audio quality; however, after the optimization step, its audio quality improves substantially and becomes nearly equivalent to that of AudioSeal/Timbre (without attack).

6. Conclusion

In this work, we analyze watermark decoder outputs and observe that they exhibit approximately normal distributions on benign audio. We show that defenders can leverage this statistical regularity to distinguish benign from attacked signals. To demonstrate the fragility of such defenses, we introduce AWM, an adaptive audio watermark attack. AWM employs a two-stage optimization procedure to evade distribution-based detection while preserving high audio quality. Experiments show that while distribution-based defenses can detect prior attacks, AWM consistently bypasses them, achieving higher attack success rates and lower detection success rates. These findings underscore the urgent need for robust defense mechanisms designed for adaptive attacks on audio watermarking.

7. Acknowledgments

This work was supported in part by the U.S. National Science Foundation under Grant CNS-2310207, CNS-2520900, CNS-2451168.

8. Generative AI Use Disclosure

The authors used generative AI tools such as ChatGPT solely for grammar improvement and language polishing. All technical contents were conceived, conducted, and verified by the authors, which include the methodology, experiments, analysis, etc. The authors take full responsibility for the content of the publication.

9. References

- [1] McAfee. (2023, May) Beware the artificial impostor. a mcafee cybersecurity artificial intelligence report. [Online]. Available: <https://www.mcafee.com/content/dam/consumer/en-us/resources/cybersecurity/artificial-intelligence/rp-beware-the-artificial-impostor-report.pdf>
- [2] J. Ren, H. Xu, P. He, Y. Cui, S. Zeng, J. Zhang, H. Wen, J. Ding, P. Huang, L. Lyu *et al.*, “Copyright protection in generative ai: A technical perspective,” *arXiv preprint arXiv:2402.02333*, 2024.
- [3] 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.
- [4] C. Liu, J. Zhang, T. Zhang, X. Yang, W. Zhang, and N. Yu, “Detecting voice cloning attacks via timbre watermarking,” in *Network and Distributed System Security Symposium*, 2024.
- [5] R. San Roman, P. Fernandez, H. Elshahar, A. D’efossez, T. Furon, and T. Tran, “Proactive detection of voice cloning with localized watermarking,” *ICML*, 2024.
- [6] J. Zhou, J. Yi, Y. Ren, J. Tao, T. Wang, and C. Y. Zhang, “Wm-codec: End-to-end neural speech codec with deep watermarking for authenticity verification,” in *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2025, pp. 1–5.
- [7] Y. Wen, A. Innuganti, A. B. Ramos, H. Guo, and Q. Yan, “Sok: How robust is audio watermarking in generative ai models?” *arXiv preprint arXiv:2503.19176*, 2025.
- [8] H. Liu, M. Guo, Z. Jiang, L. Wang, and N. Gong, “Audiomark-bench: Benchmarking robustness of audio watermarking,” *Advances in Neural Information Processing Systems*, vol. 37, pp. 52 241–52 265, 2024.
- [9] P. Yang, H. Ci, Y. Song, and M. Z. Shou, “Can simple averaging defeat modern watermarks?” *Advances in Neural Information Processing Systems*, vol. 37, pp. 56 644–56 673, 2024.
- [10] N. Lukas, E. Jiang, X. Li, and F. Kerschbaum, “Sok: How robust is image classification deep neural network watermarking?” in *2022 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2022, pp. 787–804.
- [11] M. Pan, Y. Zeng, L. Lyu, X. Lin, and R. Jia, “{ASSET}: Robust backdoor data detection across a multiplicity of deep learning paradigms,” in *32nd USENIX Security Symposium (USENIX Security 23)*, 2023, pp. 2725–2742.
- [12] M. Pan, Z. Wang, X. Dong, V. Schwag, L. Lyu, and X. Lin, “Finding needles in a haystack: A black-box approach to invisible watermark detection,” in *European Conference on Computer Vision*. Springer, 2024, pp. 253–270.
- [13] G. Hua, J. Huang, Y. Q. Shi, J. Goh, and V. L. Thing, “Twenty years of digital audio watermarking—a comprehensive review,” *Signal processing*, vol. 128, pp. 222–242, 2016.
- [14] G. Chen, Y. Wu, S. Liu, T. Liu, X. Du, and F. Wei, “Wavmark: Watermarking for audio generation,” *arXiv preprint arXiv:2308.12770*, 2023.
- [15] L. Juvela and X. Wang, “Audio codec augmentation for robust collaborative watermarking of speech synthesis,” in *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2025, pp. 1–5.
- [16] M. K. Singh, N. Takahashi, W.-H. Liao, and Y. Mitsufuji, “Silent-cipher: Deep audio watermarking,” in *INTERSPEECH*, 2024.
- [17] A. Défossez, J. Copet, G. Synnaeve, and Y. Adi, “High fidelity neural audio compression,” *arXiv preprint arXiv:2210.13438*, 2022.
- [18] J.-c. Chou, C.-c. Yeh, and H.-y. Lee, “One-shot voice conversion by separating speaker and content representations with instance normalization,” *arXiv preprint arXiv:1904.05742*, 2019.
- [19] Y. Y. Lin, C.-M. Chien, J.-H. Lin, H.-y. Lee, and L.-s. Lee, “Fragmentvc: Any-to-any voice conversion by end-to-end extracting and fusing fine-grained voice fragments with attention,” in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 5939–5943.
- [20] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan *et al.*, “Natural tts synthesis by conditioning wavenet on mel spectrogram predictions,” in *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2018, pp. 4779–4783.
- [21] Y. Ren, C. Hu, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu, “Fastspeech 2: Fast and high-quality end-to-end text to speech,” *arXiv preprint arXiv:2006.04558*, 2020.
- [22] J. Chen, M. I. Jordan, and M. J. Wainwright, “Hopskipjumpattack: A query-efficient decision-based attack,” in *2020 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2020, pp. 1277–1294.
- [23] M. Andriushchenko, F. Croce, N. Flammarion, and M. Hein, “Square attack: a query-efficient black-box adversarial attack via random search,” in *European conference on computer vision*. Springer, 2020, pp. 484–501.
- [24] N. Carlini and D. Wagner, “Towards evaluating the robustness of neural networks,” in *2017 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2017, pp. 39–57.
- [25] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, “Towards deep learning models resistant to adversarial attacks,” *arXiv preprint arXiv:1706.06083*, 2017.
- [26] Z. Jiang, J. Zhang, and N. Z. Gong, “Evading watermark based detection of ai-generated content,” in *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, 2023, pp. 1168–1181.
- [27] B. Elizalde, S. Deshmukh, M. Al Ismail, and H. Wang, “Clap learning audio concepts from natural language supervision,” in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.
- [28] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, “Robust speech recognition via large-scale weak supervision,” in *International conference on machine learning*. PMLR, 2023, pp. 28 492–28 518.
- [29] A. Blázquez-García, A. Conde, U. Mori, and J. A. Lozano, “A review on outlier/anomaly detection in time series data,” *ACM computing surveys (CSUR)*, vol. 54, no. 3, pp. 1–33, 2021.
- [30] H. Wu, T. Hu, Y. Liu, H. Zhou, J. Wang, and M. Long, “Timesnet: Temporal 2d-variation modeling for general time series analysis,” *arXiv preprint arXiv:2210.02186*, 2022.
- [31] Z. Wang, J. Bao, W. Zhou, W. Wang, H. Hu, H. Chen, and H. Li, “Dire for diffusion-generated image detection,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 22 445–22 455.
- [32] A. Y. J. Ha, J. Passananti, R. Bhaskar, S. Shan, R. Southen, H. Zheng, and B. Y. Zhao, “Organic or diffused: Can we distinguish human art from ai-generated images?” in *Proceedings of the 2024 ACM SIGSAC Conference on Computer and Communications Security*, 2024, pp. 4822–4836.

- [33] D. Afchar, G. Meseguer-Brocal, and R. Hennequin, "Detecting music deepfakes is easy but actually hard," *arXiv preprint arXiv:2405.04181*, 2024.
- [34] Y. Zang, Y. Zhang, M. Heydari, and Z. Duan, "Singfake: Singing voice deepfake detection," in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2024, pp. 12 156–12 160.
- [35] K. P. Murphy, "Conjugate bayesian analysis of the gaussian distribution," *def*, vol. 1, no. $2\sigma^2$, p. 16, 2007.
- [36] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2015, pp. 5206–5210.
- [37] R. Ardila, M. Branson, K. Davis, M. Henretty, M. Kohler, J. Meyer, R. Morais, L. Saunders, F. M. Tyers, and G. Weber, "Common voice: A massively-multilingual speech corpus," *arXiv preprint arXiv:1912.06670*, 2019.
- [38] G. Chen, S. Chai, G. Wang, J. Du, W.-Q. Zhang, C. Weng, D. Su, D. Povey, J. Trmal, J. Zhang *et al.*, "Gigaspeech: An evolving, multi-domain asr corpus with 10,000 hours of transcribed audio," *arXiv preprint arXiv:2106.06909*, 2021.
- [39] A. Hines, J. Skoglund, A. C. Kokaram, and N. Harte, "Visqol: an objective speech quality model," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2015, pp. 1–18, 2015.