# STATISTICAL LEARNING-BASED ADAPTIVE ATTACKS TOWARDS AUDIO WATERMARKING

By

Weikang Ding

A THESIS

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

Computer Science—Master of Science

2025

# ABSTRACT

The abuse of original audios has attracted widespread attention in the society. Audio watermarking, which embeds imperceptible signals into audio content, has been proposed as an effective way to assert user copyright of audios. Although recent deep learning-based audio watermarking methods have enhanced robustness and capacity compared to traditional approaches, they are vulnerable to adversarial attacks. Our findings reveal that the message probabilities output by the watermark decoder follow a normal distribution for both clean and watermarked audio. This observation can be leveraged to detect existing audio watermark attacks. In this thesis, we introduce AWM, an adaptive audio watermark attack method designed to bypass existing detection strategies. The attack has three different types: watermark replacement, watermark creation, and watermark removal. AWM employs a two-step optimization process: the first step ensures the success of the watermark attack and bypasses the detection by optimizing message probabilities within an estimated normal range, while the second step focuses on enhancing audio quality while maintaining a successful attack. The proposed attack iteratively estimates the parameters of the normal distribution using a small set of feature-similar audio samples based on the target audio and applies adaptive optimization to adjust the decoded message probabilities toward the estimated normal range. We evaluate AWM on two watermarking methods across three diverse voice datasets and compare the results with existing audio watermark attack techniques. Our experiments demonstrate that the proposed attack achieves a high attack success rate while effectively bypassing detection, with detection success rates remaining under 10% for watermark replacement and watermark creation, and at 0% for watermark removal. Additionally, AWM exhibits high robustness against various no-box perturbations, including low-pass filtering, amplitude scaling, and compression, while maintaining high perceptual audio quality. Our experiments highlight a significant security gap in current watermark defenses and show that statistical assumptions about the decoder output can be exploited by attackers. These findings also provide a foundation for future research in audio watermark attack detection and the development of more advanced attacks.

# ACKNOWLEDGMENTS

I would like to thank my advisor, Dr. Qiben Yan. He not only offered careful guidance in academic research, but also subtly shaped my thinking, research methodology, and understanding of academic norms.

Special thanks to my collaborator Hanqing Guo, his deep expertise in audio watermarking greatly helped me build a strong foundation in this field. I would also like to thank my friends and collaborators Guangjing Wang, Juexing Wang, Ce Zhou, Yuanda Wang, Bocheng Chen, and other members of Lab 1100. Their support has provided me with valuable experiences in both academia and life, and I am truly grateful for the time spent with them.

I would also like to thank my thesis committee members, Dr. Li Xiao and Dr. Huacheng Zeng. Their insightful comments and constructive suggestions not only enhanced the quality of my thesis but also deepened my understanding of the research topic.

Finally, I would like to express my deepest gratitude to my parents. Thank you for your unwavering love and support. It is your care that has given me the courage and strength to pursue my dreams. In times of difficulty and discouragement, you have always been my greatest source of strength. Your understanding and encouragement have been the driving force that has sustained me to this day.

## TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 1

CHAPTER 2: RELATED WORK . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 6

CHAPTER 3: BACKGROUND . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 8

CHAPTER 4: METHODOLOGY . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 14

CHAPTER 5: EVALUATION . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 27

CHAPTER 6: DISCUSSION . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 36

CHAPTER 7: CONCLUSION . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 38

BIBLIOGRAPHY . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 39

# CHAPTER 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 can bring them substantial income. However, many unauthorized users steal the work of creators, make modifications, and re-upload it to mainstream platforms for profit, which significantly diminishes the enthusiasm of 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 [28]. To address these issues, audio watermarking has been proposed [34]. This technique embeds a noise-tolerant signal into the target audio, which remains imperceptible to human hearing while being detectable by specialized AI models.

Traditional watermarking methods use signal processing techniques to embed the watermark into the time domain [4,17,21], frequency domain [46], or transform domain [20,39,43]. However, these traditional watermarking methods struggle to defend against various complex attacks and offer limited capacity, often relying on a single optimization objective tailored to specific attack types. In contrast, deep-learning-based watermarking methods greatly address the limitations of traditional watermarking methods and improve the robustness and generalization of watermarking. Typically, a watermark is composed of multiple binary bits. Deep-learning-based watermarking methods use Encoder-Decoder neural network architectures to embed and extract the watermark from audio. Moreover, they introduce distortion mechanisms to simulate more complex potential attack scenarios, such as audio re-recording [23], voice cloning [24, 36], and Codec [47].

However, the robustness of deep-learning-based audio watermark methods against adversarial attacks is a concerning issue [41]. Recent studies [25,44] have revealed that attackers can remove or forge watermarks by embedding adversarial perturbations. The basic principle involves adding and optimizing a new perturbation to deceive the watermark decoder into generating incorrect outputs.

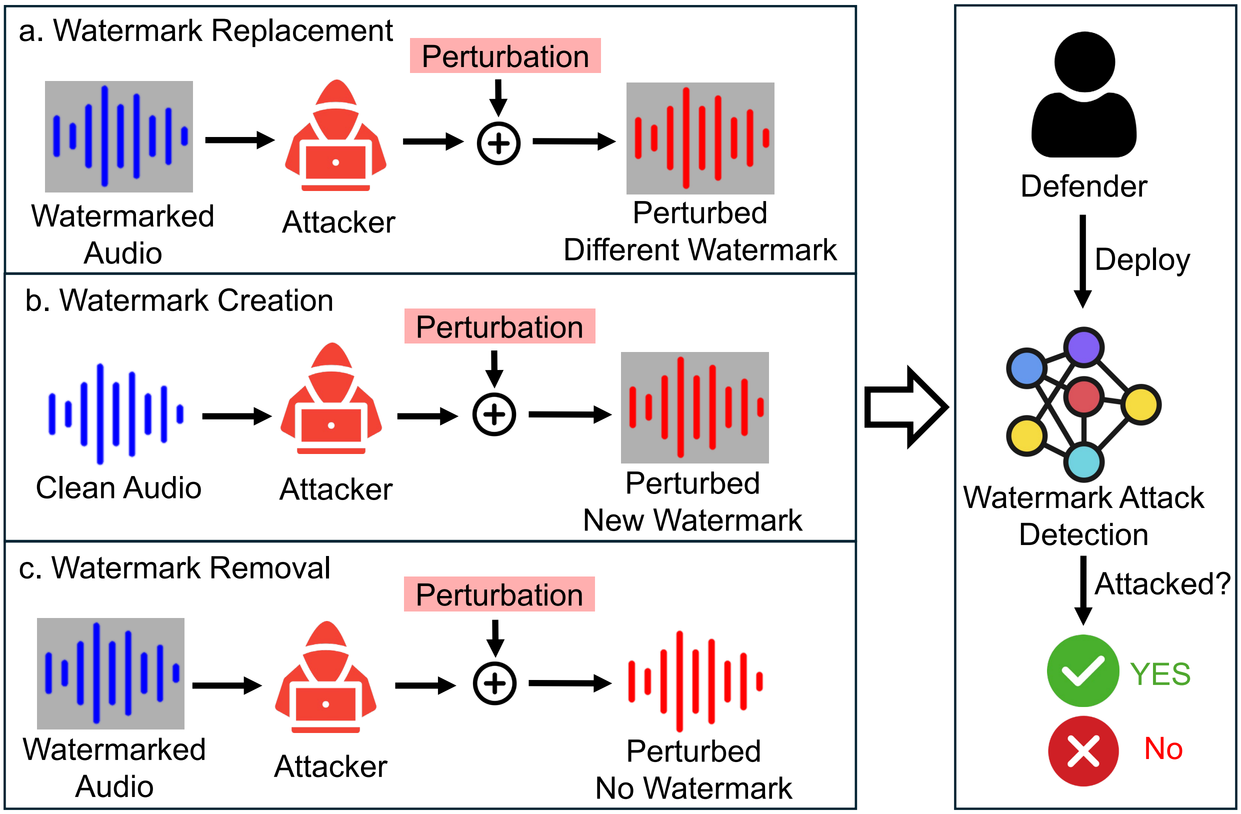


Figure 1.1: Overview of the watermark attack (left) and the watermark attack detection process used to detect whether the audio has been tampered with (right).

There are three types of watermark attack scenarios: watermark replacement, watermark creation, and watermark removal. Through watermark replacement and creation attacks, attackers can forge fake watermarks in audio to falsely claim ownership and overwrite others' original creations with their own. Alternatively, attackers can transfer the copyright of some illegally created audios to others, which can be used to evade responsibility or frame innocent parties. Watermark removal attacks, on the other hand, aim to remove the watermark from watermarked audio, which prevents the watermark decoder from outputting the original message. Through watermark removal attacks, attackers can remove the original owner's copyright and redistribute the content on public platforms, resulting in financial losses for the creators.

Currently, no existing attack method effectively balances attack effectiveness with preserved audio quality. First, existing attack approaches lack a well-designed method to balance audio quality with attack effectiveness. The watermark model is typically trained through a joint encoder-decoder framework: the encoder embeds the watermark into imperceptible regions of the audio, while the decoder is trained to robustly extract the watermark under various perturbations. However, attackers generally do not have access to the watermark encoder and can only query it to obtain watermarked audio. In contrast, the watermark decoder is more accessible to the public, making it the primary source of model knowledge in most attack scenarios. Since the decoder is responsible for extracting watermark messages, effective attacks require carefully balancing perturbation strength to modify the message while preserving perceptual quality.

Second, our findings indicate that simply altering the binary watermark message bits is insufficient. Given an audio input, the decoder produces both a binary message and its associated probability scores. These message probabilities tend to follow a normal distribution, and the defenders can use this result to detect whether an audio sample has been manipulated. Therefore, for an attack to remain undetected, the attacker must ensure that the decoded message probabilities fall within the normal range predefined by the defender.

Third, attackers typically lack access to the training data used to construct the watermark model. To estimate the distribution parameters required for a successful attack, they need to rely on analyzing the output of the decoder. Since different data samples can generate different message probability outputs, the significant difference between the attacker's estimated parameters and the defender's true parameters may lead to attack failure.

In this thesis, we propose AWM, an Adaptive audio Watermark attack Method, which is capable of bypassing the defender's detection strategy. Figure 1.1 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". To design AWM, we face the following challenges:

*C1: How to design an attack method to balance audio quality and attack effectiveness?* Balancing audio quality and attack effectiveness can be viewed as a game-theoretic challenge.

An overly aggressive attack can significantly degrade audio quality, while prioritizing perceptual quality may compromise attack success. Therefore, 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?* The goal of the perturbed audio is to bypass the detection strategy. The defender detects perturbed audio by analyzing the distribution of decoded message probabilities. 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?* Attackers do not have access to the training data used by the watermark model. To estimate the distribution parameters of decoded message probabilities, they must rely on a limited set of audio samples. This necessitates designing an effective estimation strategy and selecting audio samples with features similar to the target to improve the accuracy of distribution fitting.

**Our Idea:** We propose three solutions to the three challenges above. To address C1, we design a two-step approach for the watermark attack methods. The first step focuses on maximizing attack effectiveness by ensuring the attack succeeds while evading the defender's detection strategy. The second step aims to improve the audio quality. Inspired by [26], we set a reasonable threshold to enhance the audio quality while constraining the decoded message probabilities to remain within the expected normal range. To address C2, we introduce an adaptive optimization strategy that guides decoded message probabilities toward the estimated normal distribution range. The core idea is to prioritize optimization efforts on the probabilities that fall outside the estimated normal range. In other words, if a perturbed message probability already falls within the estimated normal range, its optimization weight is reduced. This ensures that the optimization process remains dynamic and focuses on the most optimization-needed binary message bits. To address C3, data with similar feature distributions are more likely to produce similar decoded message probability distributions [30, 31]. We collect a small set of audio samples and estimate the probability distribution by selecting those whose features closely resemble those of the target audio.

**Contribution:** In this thesis, we make the following contributions:

- We observe that the decoded message probabilities output by the watermark decoder 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 targeting three attack types. Our approach successfully bypasses detection strategies through an adaptive two-step optimization framework. The first step enhances the effectiveness of the watermark attack, while the second step focuses on improving audio quality. To initialize the optimization, we estimate the parameters of the normal distribution using a limited set of audio samples selected based on feature similarity to the target audio. Our adaptive optimization strategy prioritizes message probabilities that require adjustment, further improving attack performance.
- We evaluate the effectiveness of AWM on three speech datasets and two state-of-the-art watermarking models. Compared to the baseline, our method achieves superior performance in both Attack Success Rate (ASR) and Detection Success Rate (DSR). For watermark replacement and creation, AWM achieves the DSR scores below 10%, which is acceptable given a False Acceptance Rate (FAR) of around 5%. For watermark removal, AWM achieves the DSR scores of 0%. Furthermore, even after applying five no-box perturbations, AWM consistently maintains a high ASR, with most scores nearing or reaching 100%.

# CHAPTER 2: RELATED WORK

## 2.1: Deep Learning-Based Audio Watermarking Scheme

Unlike traditional schemes [18], which rely on predefined transformations, deep-learning-based schemes can learn complex feature representations and optimize watermarking dynamically. This scheme follows the architecture of the Encoder-Distortion-Decoder [7, 24, 36]. The encoder embeds the message into the audio and generates the watermarked audio, the decoder receives the watermarked audio and extracts the corresponding messages, and the distortion simulates a variety of potential attack scenarios. This thesis builds upon the scheme and implements both the defense mechanism and the corresponding attack strategy.

## 2.2: Audio Watermark Attack

Audio watermark attacks based on adversarial perturbations can be categorized as no-box, black-box, gray-box, or white-box, depending on the attacker's knowledge of the watermarking model. In no-box perturbations, the attacker relies on general audio processing techniques that distort the signal while attempting to preserve perceptual quality. These attacks aim to remove the watermark from watermarked audio using techniques such as band-pass filter, amplitude scaling, audio compression [12], and others. Additionally, methods like voice conversion [11, 22] and text-to-speech [35, 38] have also proven effective in removing watermarks [41]. In black-box perturbations, the attacker can query the watermarking detector but does not have access to its internal architecture or parameters. AudioMarkBench [25] performs the watermark removal attack by applying the thought of HSJA [9] and Square Attack [2]. In gray-box perturbations, the attacker is assumed to know the architecture of the decoder but does not have access to its trained weights. 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 [6, 27]. While some existing studies have successfully demonstrated watermark removal attacks, our experimental results show that none can effectively perform watermark replacement and watermark creation attacks. Prior work in the image domain has explored averaging watermark patterns for attack purposes [44], but this approach is ineffective in the audio domain. In this thesis, we focus on performing watermark replacement, watermark creation, and watermark removal attacks.

## 2.3: 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 [13] and speech recognition models [33]). Therefore, some outlier detection methods based on the time series [5,42] are generally ineffective for identifying whether an audio sample has been watermarked or if the watermark has been removed or forged. Recent works have explored detection methods for generated images [15, 40], watermark images [19, 30], audio deepfake [1, 45], and dataset copyright protection [14]. However, to the best of our knowledge, these approaches are not directly applicable to detecting audio watermark replacement, creation, or removal attacks.

# CHAPTER 3: BACKGROUND

## 3.1: Preliminary

Adding perturbations to the audio is a common method to perform watermark attacks. The core idea is to either destroy the original watermark or forge a new one by introducing perturbations that deceive the watermark decoder. There are three types of attacks (as shown in Figure 3.1): watermark replacement attacks, which aim to replace an existing watermark with a different one; watermark creation attacks, which aim to embed a new watermark into clean audio; and watermark removal attacks, which aim to eliminate the original watermark from a watermarked audio.

**Audio Watermarking Framework.** The audio watermarking framework has three main components: encoder, decoder or detector, and distortion layer. The encoder embeds the message into the clean audio and then generates the watermarked audio. The decoder or detector receives the watermarked audio and then outputs the extracted message. The distortion layer simulates potential attack scenarios. Figure 3.2 shows the Encoder-Distortion-Decoder architecture.

**Audio Watermark Decoder.** The audio watermark decoder $Dec(\cdot)$ takes the encoded audio as input and outputs the extracted message[1]. 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 probability form provides the decoder's confidence values for each bit, indicating the likelihood of the bit being 1 or 0. A predefined threshold $\theta$ is typically 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 is:

$$p = Dec(s), \tag{3.1}$$

[1] We only consider multi-bit messages in this thesis.

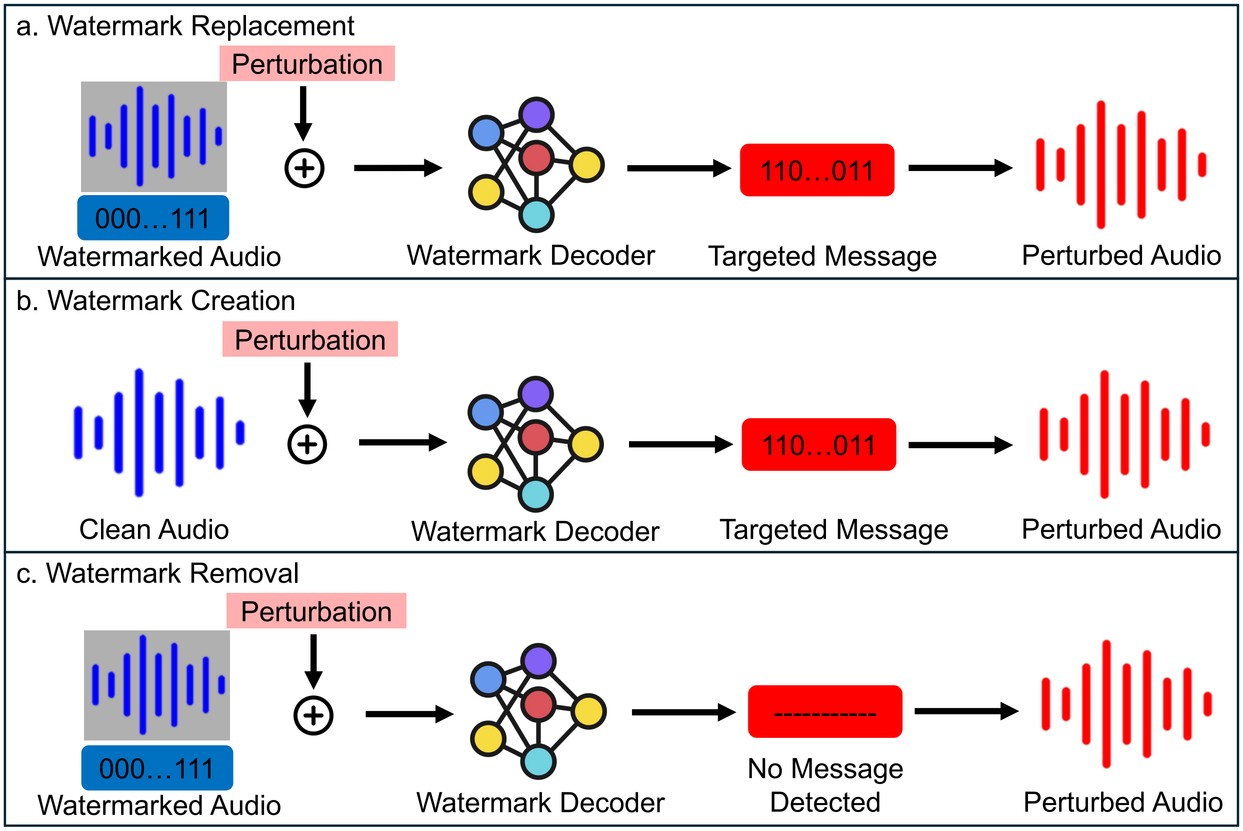


Figure 3.1: Watermark attack types.

where s is clean or watermarked audio, $Dec(\cdot)$ outputs the message probabilities[2].

**Watermark Replacement.** 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 $p_{target}$. A perturbation $\delta$ is introduced to perform the attack on the audio.

In watermark replacement (Figure 3.1-a), a perturbation $\delta$ 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:

$$\delta_{replacement} = \arg\min_{\delta} ||Dec(s_w + \delta) - p_{target}||. \tag{3.2}$$

**Watermark Creation.** Watermark creation (Figure 3.1-b) involves adding a perturbation $\delta$ to a

[2] The message probabilities $p$ can be same as the binary message.

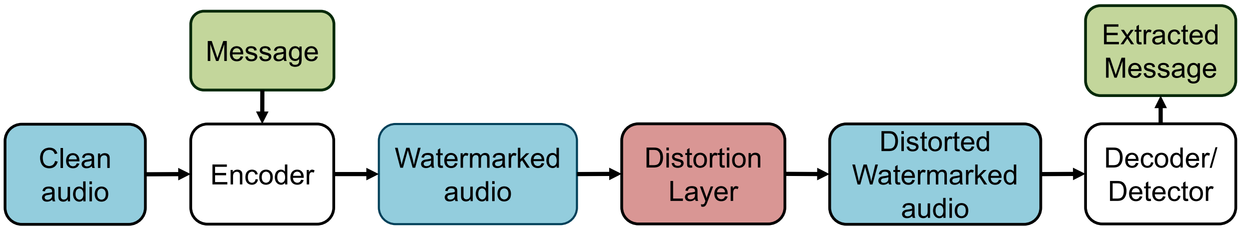


Figure 3.2: The audio watermarking framework follows the Encoder-Distortion-Decoder architecture.

clean audio $s_c$ to generate an perturbed watermarked audio, which deceives the watermark decoder into recognizing it as containing a valid watermark. Formally, the goal of watermark creation is:

$$\delta_{creation} = \arg\min_{\delta} ||Dec(s_c + \delta) - p_{target})||. \tag{3.3}$$

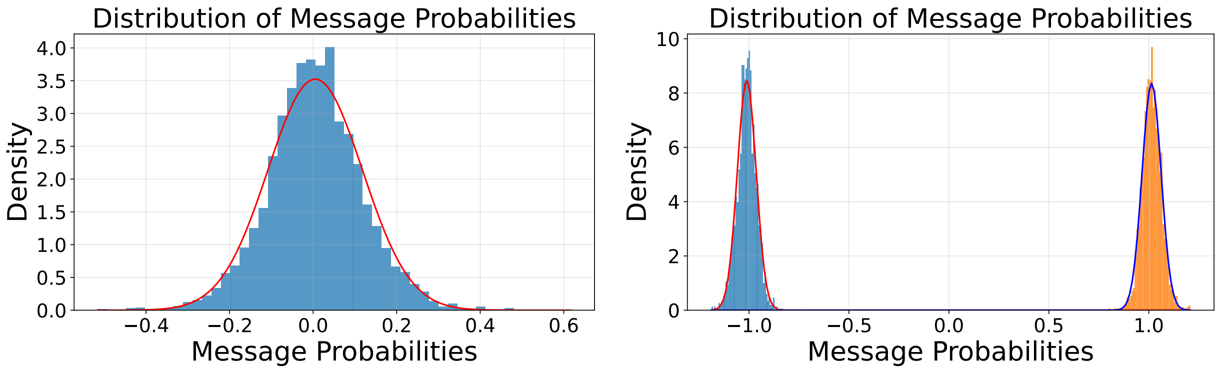
**Watermark Removal.** Watermark removal is an untargeted attack aimed at removing the embedded watermark. In a watermark removal attack (Figure 3.1-c), a perturbation $\delta$ is added to the watermarked audio $s_w$ to deceive the watermark decoder into outputting a binary message that does not match the original watermark binary message. The perturbed audio, denoted as $\hat{s}_c$, is considered clean (i.e., unwatermarked). In this context, we introduce a watermark detector, $Detector(\cdot)$, which determines whether the audio contains a watermark. Formally, the goal of watermark removal is:

$$\delta_{removal} = \arg\min_{\delta}(Detector(s_w + \delta) = \text{None}). \tag{3.4}$$

## 3.2: Message Probability Distribution

**Benign Distribution.**

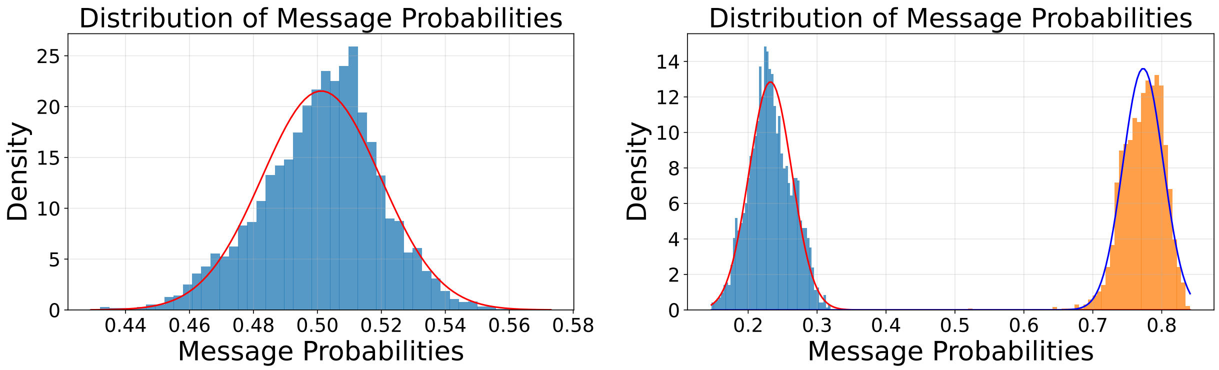
The watermark decoder outputs the values of probabilities for each binary message bit. A bit is decoded as 1 if its probability exceeds a predefined threshold; otherwise, it is decoded as 0. For both clean and watermarked audio, we observe two distinct distributions, each following a normal distribution pattern. Clean audio exhibits a unimodal normal distribution, with the peak density



(a) Normal distribution with clean audio.

(b) Normal distribution with watermarked audio.

Figure 3.3: Benign distribution of message probabilities (Timbre).



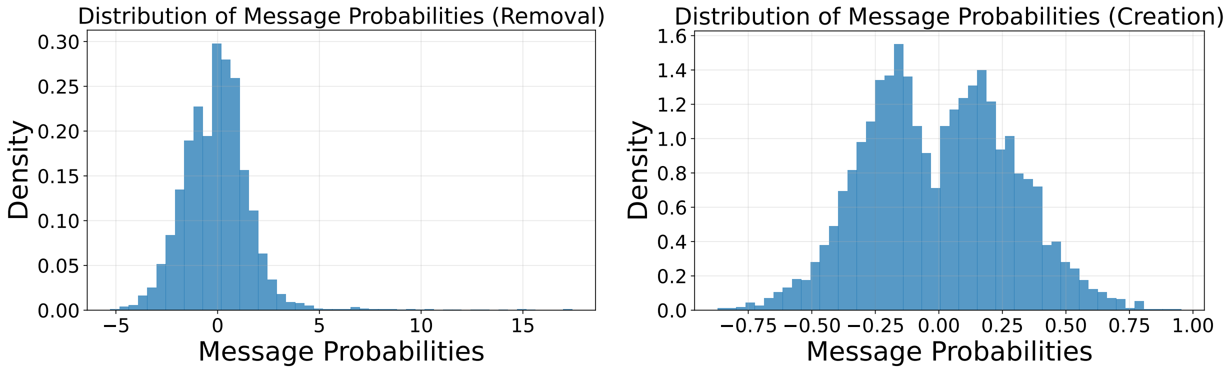
(a) Normal Distribution with Clean Audio.

(b) Normal Distribution with Watermarked Audio.

Figure 3.4: Benign distribution of message probabilities (AudioSeal).

centered near the predefined threshold. As shown in Figure 3.3a, where the threshold is set to 0, the mean $\mu$ of the distribution is also close to 0. In contrast, the watermarked audio follows a bimodal, approximately normal distribution (Figure 3.3b), with two peaks corresponding to the decoded binary message values 0 and 1. Specifically, in the Timbre, message probabilities below 0 are decoded as 0, while message probabilities above 0 are decoded as 1. Figure 3.4 shows the distribution findings of the AudioSeal.

**Audio Watermark Attack Distribution.** For existing watermark attack methods [25], we observe that the distributions of message probabilities deviate significantly from the benign distributions. Figure 3.5a illustrates the distribution of watermark removal attacks, which aim to remove the watermark from watermarked audio. Compared to Figure 3.3a, the attacked distribution differs



(a) Message distribution under watermark removal. (b) Message distribution under watermark creation.

Figure 3.5: Distribution of message probabilities under attacks (Timbre). We use the AudioMarkBench to perform the removal and creation attacks.

in the range of message probabilities along the x-axis, even though it still exhibits a unimodal normal distribution. Figure 3.5b 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 3.3b, the distribution resulting from the attack visually diverges from the benign distribution.

**Outlier Detection.** The 3-sigma rule, also known as the 68–95–99.7 rule, is a widely adopted method for identifying outliers in normally distributed data. It states that approximately 68% of values fall within one standard deviation of the mean, 95% within two, and 99.7% within three. Values that fall beyond three standard deviations from the mean are considered statistically rare and are classified as outliers. Based on this principle, the defender can detect potential attacks by estimating the distribution parameters of the message probabilities and identifying values that fall outside the 3-sigma range.

## 3.3: Threat Model

**Roles.** In this thesis, we focus on two roles: defenders and adversaries. The defenders are the attack detectors, who identify whether audio has been tampered with. The adversaries attempt to perform watermark attacks to either claim new copyright ownership or remove the original copyright.

**Attack Capabilities.** For adversaries, we have the following assumptions: 1) Adversaries 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 access to the watermark architecture and parameters of the watermark decoder, allowing them to perform gradient-based attacks.[3] 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.

For defenders, we have several assumptions: 1) Defenders have access to a large amount 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 restrictions on the number of watermarked audio generations allowed per user per day.

**Attack Scenario.** The adversaries aim to estimate the normal distribution parameters of the decoded message probabilities. However, due to their lack of access to the training dataset, they can only collect a limited amount of data independently. When targeting a specific audio sample for attack, they identify a small set of audio samples with feature distributions similar to that of the target and use these to estimate the mean and standard deviation. After estimating the mean and standard deviation, adversaries deploy the watermark decoder and perform gradient-based watermark attacks to constrain the decoded message probabilities within the estimated normal range. On the defense side, defenders design a defense strategy based on the ground-truth distribution. This strategy should have a reasonable false acceptance rate. When a potentially compromised audio sample is received, the defender examines the decoded message probabilities to determine whether it has been tampered with.

[3] The knowledge of watermark model's parameter setting can be extend to black-box by adopting HSJA [9] and Sign-Opt [10].

# CHAPTER 4: METHODOLOGY

## 4.1: Audio Watermark Attack Detection

As observed in the previous section, the distribution of message probabilities in perturbed audio differs significantly from that of benign audio. Based on this observation, we propose a detection mechanism to distinguish between perturbed and benign audio. It is important to note that this task is non-trivial, as the defender does not have access to the exact distribution of message probabilities under normal (benign) conditions, making it challenging to achieve optimal detection performance. To address this issue, we propose a two-step approach.

First, the defender possesses a large collection of ground truth audio samples, including both clean audio and watermarked audio generated by the watermark encoder. These samples are used to model the message probability distributions. The defender deploys a watermark decoder to extract decoded message probabilities and categorizes them into two groups: probabilities from clean audio and probabilities from watermarked audio. Next, the defender applies maximum likelihood estimation to fit the predicted distributions. As illustrated in Figure 3.3, this process outputs three normal distributions: one derived from message probabilities of clean audio and two (red and blue boundary line on the right) are from those of watermarked audio. Given $n$ watermarked samples, each producing a probability vector of size $1 \times N$ message probabilities, the complete set of message probabilities is denoted as $p = \{p_1, p_2, ..., p_n\}$. The mean $\mu$ and standard deviation $\delta$ of the normal distribution estimated via maximum likelihood are:

$$\begin{aligned} \mu &= \frac{1}{n}\sum_{i=1}^{n} p_i, \\ \sigma^2 &= \frac{1}{n}\sum_{i=1}^{n}(p_i - \mu)^2. \end{aligned} \tag{4.1}$$

Finally, the defender obtains the corresponding means $\mu$ and standard deviations $\delta$, which are used

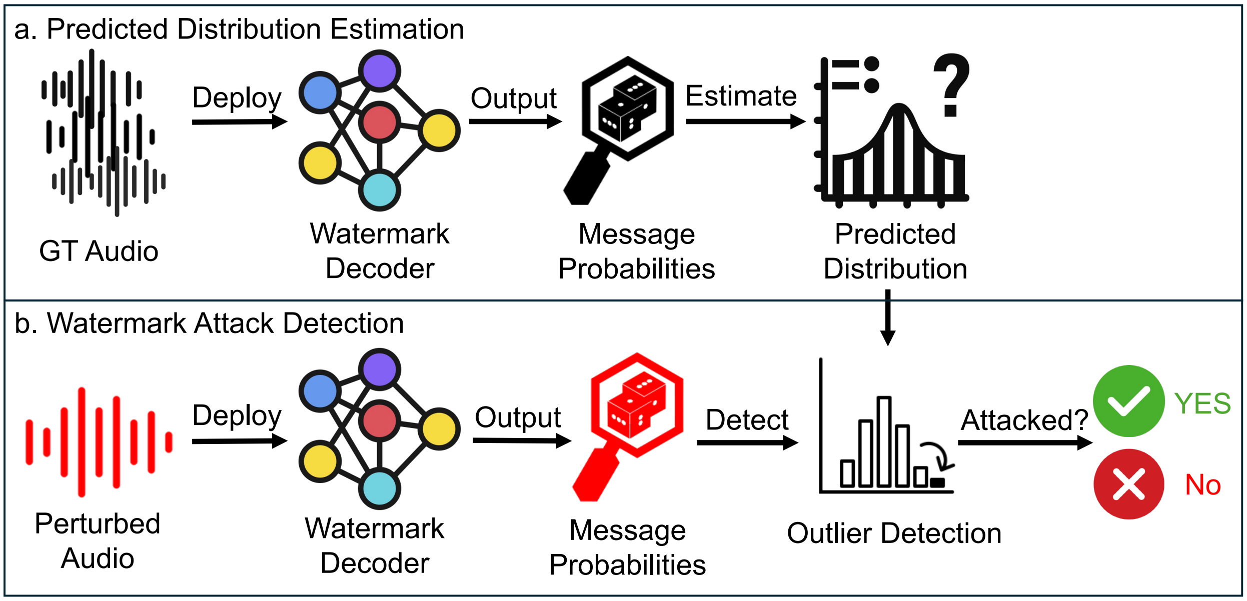


Figure 4.1: Audio watermark attack detection based on the predicted distribution of ground truth (GT) audio. The defender uses ground truth (GT) audio (watermarked and clean) to estimate the predicted distribution (top) and applies outlier detection based on this distribution to detect whether an audio sample is attacked (bottom).

to detect outliers in suspicious audio and determine whether it has been attacked.

Figure 4.1 illustrates the audio watermark attack detection scenario. The defender collects two large datasets (one consisting of watermarked audio and the other of clean audio) and uses a watermark decoder to extract message probabilities. These probabilities are then used to generate two sets of predicted distributions via maximum likelihood estimation. When the defender receives a potentially perturbed audio sample, the watermark decoder is applied to extract its message probabilities distributions. Using the 3-sigma rule, the defender identifies any probability values that fall outside the expected range. If such outliers are detected, the audio is classified as attacked; otherwise, it is considered clean.

## 4.2: Adaptive Attack Design

Given the success of message-probability-distribution-based attack detection, we consider to propose more dangerous attack: the adaptive watermark attack. Assuming the attackers are aware of a statistics-based detection mechanism, their goal is to bypass the detection. This is also

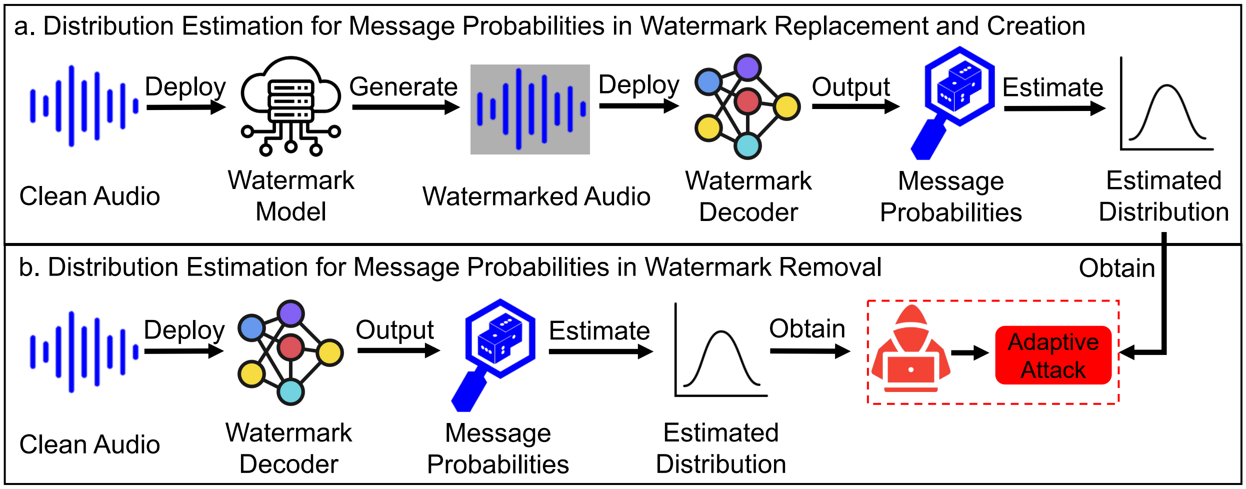


Figure 4.2: The distribution estimation by the attacker. (a) Watermark replacement and creation: The attacker uses a small set of clean audio samples to generate watermarked audio samples, which are then used to estimate the distribution. (b) Watermark removal: The attacker directly uses the clean audio samples to estimate the distribution.

non-trivial, as the attacker does not know the detection distribution used by defender, nor aware of the defender's outlier detection approach, moreover, the attacker need to preserve the audio quality, this limit the strength of the perturbation. We introduce the adaptive attack design as follows.

### 4.2.1: Adaptive Attack Pipeline

First, the attacker prepares for the attack by estimating the defender's distribution. Second, the attacker performs the audio watermark attack, which consists of two stages: (1) modifying the original audio to bypass the defense and achieve a successful attack, and (2) improving the quality of the perturbed audio, while maintaining that the decoded message probabilities remain within an acceptable threshold.

### 4.2.2: Attack Preparation: Estimate Defender's Distribution

The goal of the adversaries is to estimate the mean $u_{est}$ and the standard deviation $\sigma_{est}$ of the decoded message probabilities. These parameters are categorized into two groups corresponding to the types of watermark attacks: watermark replacement and creation, and watermark removal.

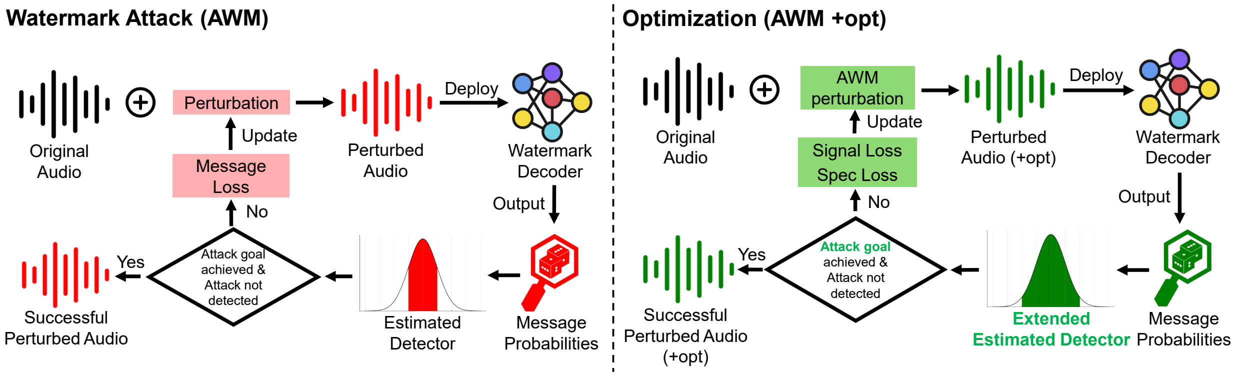


Figure 4.3: The design of AWM generator. The audio watermark attack step (left) ensures the success of the watermark attack, while the audio quality optimization step (right) focuses on improving audio quality.

The overall process is illustrated in Figure 4.2.

**Parameter Estimation for Watermark Replacement and Creation.** The attackers select several clean audio samples $s_c$ from a small dataset. These samples are then 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 using either Bayesian inference [29] or maximum likelihood estimation. Since the distribution of watermarked message probabilities is bimodal (as shown in 3.3b), the attacker can obtain two distributions (-1's decoded as bit 0 on the left and 1's decoded as bit 1 on the right). The estimated mean and standard deviation are as follows:

$$\begin{aligned} \mu_{est}^0, \sigma_{est}^0 &= T^0(Dec(Enc(s_c))), \\ \mu_{est}^1, \sigma_{est}^1 &= T^1(Dec(Enc(s_c))), \end{aligned} \tag{4.2}$$

where $\mu_{est}$ and $\sigma_{est}$ are the mean and standard deviation, respectively, with the superscript identifying the associated target distribution for each parameter. $\mu_{est}^0$ and $\sigma_{est}^0$ represent the estimated mean and standard deviation of decoded message probabilities predicted as 0, and $\mu_{est}^1$ and $\sigma_{est}^1$ correspond to those predicted as 1. $T(\cdot)$ is the method used to estimate the parameters of normal distribution, which depends on the prior knowledge of the attackers and the volume of data. Commonly used techniques are based on Bayesian inference or maximum likelihood estimation.

$T^0(\cdot)$ and $T^1(\cdot)$ denote the estimated method for estimating the 0's and 1's distribution.

**Parameter Estimation for Watermark Removal.** The attackers estimate the distribution directly using clean audio samples. They use the watermark decoder to output message probabilities, which are used for distribution estimation. Since the estimated distribution is unimodal (as shown in 3.3a), the parameters are defined as follows:

$$\mu_{est}^{c}, \sigma_{est}^{c} = T^{c}(Dec(s_c)), \tag{4.3}$$

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

### 4.2.3: Audio Watermark Attack (AWM)

With the estimated defender's distribution, the attacker's next step is to add a subtle adversarial perturbation to the original audio $s$, intentionally deceiving the decoder to produce incorrect binary messages while bypassing the detection strategy. Because attackers possess the watermark decoder, they can compute and adjust gradient directions to align with their attack goals. Figure 4.3-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. At the beginning, the original audio adds the perturbation. The perturbation is initialized by a fraction of the original audio signal. Next, we add the perturbation to original audio to form a perturbed audio. The perturbed audio is fed into a watermark decoder, where the attacker receives a message probability and queries the estimated detector. If the attack is not detected and the attack goal is achieved, 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 $p_{target}$:

$$\mathcal{L}_{msg} = ||Dec(s_{att}) - p_{target})||_2^2. \tag{4.4}$$

Let $s_{att}$ denote the perturbed audio, this loss enforces the decoded message close to the target message. **Different from the previous attack, 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 watermarked audio, with high confidence. It is worth noting that our algorithm constrains the message probabilities within the normal value range. Through iterative updates and perturbation optimization, the perturbed message probabilities are gradually optimized to fall within this range $[\mu_{est} - \sigma_{est}, \mu_{est} + \sigma_{est}]$, ensuring that they are classified as non-outliers, thereby getting the updated adversarial perturbation. Finally, the perturbed audio is generated by adding the updated perturbation to the original audio. Our findings suggest that if the perturbed audio's message probability falls into the interval of $[\mu_{est} - \sigma_{est}, \mu_{est} + \sigma_{est}]$, it would improve 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:

$$\mathcal{L}_{signal} = \frac{1}{n}\sum_{i=1}^{n} |s_{att} - s|. \tag{4.5}$$

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

$$\mathcal{L}_{mel} = ||Mel(s_{att}) - Mel(s)||_2^2. \tag{4.6}$$

The total loss in the attack step is:

$$\mathcal{L} = \lambda_1\mathcal{L}_{signal} + \lambda_2\mathcal{L}_{mel} + \lambda_3\mathcal{L}_{msg} + \lambda_4\mathcal{L}_{other}, \tag{4.7}$$

where the $\mathcal{L}_{other}$ depends on the specific watermarking method. For example, AudioSeal [36] includes a localization loss used for indicating the probability of the audio being watermarked. In the audio watermark attack process, the parameter $\lambda_{msg}$ is assigned a relatively high value.

### 4.2.4: Audio Quality Optimization (AWM +opt)

Since audio watermark attacks prioritize message loss optimization, audio quality may be adversely impacted. Therefore, this step aims to improve audio quality while maintaining a successful attack. To achieve this, we make three adaptions. (1) Extend the Estimated Detector Range; (2) Update the Attack Goal; and (3) Modify the Optimization Loss.

Figure 4.3-right illustrates the step. First, attackers obtain the perturbation from the early step as initial perturbation. Then, the attacker uses the watermark decoder to extract the corresponding message probabilities. These message probabilities are input into the estimated distribution, with the acceptable range extended to $[\mu_{est} - 2\sigma_{est}, \mu_{est} + 2\sigma_{est}]$. **(1) This is a critical step where the attacker makes a trade-off between audio quality and attack success rate.** In previous AWM setting, the attacker starts from strict constrain to enforce the attack success, however, this enforcement limits the flexibility of perturbation, making it hard to retain the audio quality as well. In our design, we expand the maximum allowable range for message probability optimization to reserve more space for perturbation optimization, resulting in a balanced attack mode which considers both quality and attack success rate.

Besides extending the acceptable range, we also (2) update the attack goal by enforcing the optimization go through fixed number of optimization epochs. This ensures that the perturbation is fully optimized, instead of ending up with a boundary case. At the end, we (3) modify the optimization loss by reformulating the Mel-Spectrogram loss into a standard spectrogram loss and apply a softmax function to the spectrogram. Inspired from Audioseal, the softmax adaption can better keep the loudness and perceptual similarity between two audios. The resulting softmax-based spectrogram loss, denoted as $\mathcal{L}_{\text{spec}}$, is defined as:

$$\mathcal{L}_{spec} = \frac{1}{n}\sum_{i=1}^{n} |Softmax(S_{att}) - Softmax(S_s)| \tag{4.8}$$

Where $S_{att}$ and $S_s$ are the spectrogram of the attacked audio and original audio.

$$\mathcal{L}_{opt} = \lambda_1 \mathcal{L}_{signal} + \lambda_2 \mathcal{L}_{spec} + \lambda_3 \mathcal{L}_{msg} + \lambda_4 \mathcal{L}_{other}, \tag{4.9}$$

In the audio quality optimization (+opt) process, the parameters $\lambda_1$ and $\lambda_2$ are assigned a relatively high value.

## 4.3: Adaptive Attack in Replacement, Creation, and Removal

### 4.3.1: Adaptive Attack in Watermark Replacement

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, consider a watermark replacement scenario involving six binary message bits. Suppose the original watermarked audio contains the bits “101100”, which are modified to “111000” after the attack. We define a list, $msg_{diff}$, which contains the indices of binary message bits differing between the original watermarked audio $s_w$ and the perturbed watermarked audio $\hat{s_w}$[4]. In this case, $msg_{diff} = [1, 3]$. For the indices not included in $msg_{diff}$, we assign the ground truth watermark message probabilities $p_w$ to the corresponding target watermark message probabilities $p_{target}$. That is, for indices $[0, 2, 4, 5]$, the $p_{target}$ 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. Additionally, in certain audio watermarking methods, message probabilities do not directly map to binary message values. For the AudioSeal as shown in Figure 3.4b, probabilities corresponding to binary 1 typically lie in the range of 0.7–0.8, and those

[4] This notation specify for watermark replacement scenario, the general representation of attacked audio is $s_{att}$, it can represent both attacked clean audio (for watermark creation) or attacked watermarked audio (for watermark replacement and removal)

```
Algorithm 1: Adaptive Attack in Watermark Replacement
    Input: Watermarked audio $s_w$, watermark message probabilities $p_w$, perturbed watermark
           probabilities $\hat{p_w}$, target message probabilities $p_{target}$, watermark decoder $Dec(\cdot)$,
           thresholds $\tau^0_{sup}, \tau^0_{infi}, \tau^1_{sup}, \tau^1_{infi}$, scale factor $r$, list of changed indices $msg_{diff}$
    Output: Perturbed watermarked audio $\hat{s_w}$
 1  $\delta \leftarrow s_w \times r$ ;
 2  $p_w \leftarrow Dec(s_w)$ ;
 3  $\hat{s_w} \leftarrow s_w + \delta$ ;
 4  for $index \leftarrow 1$ to $len(p_{target}) - 1$ do
 5      if $index \notin msg_{diff}$ then
 6          $p_{target}[index] \leftarrow p_w[index]$ ;
 7  for $i \leftarrow 1$ to $Iter$ do
 8      $\delta \leftarrow Attack(s_w, \hat{s_w}, Dec(\hat{s_w}), p_{target}, \delta)$                // Optimize $\delta$;
 9      $\hat{s_w} \leftarrow s_w + \delta$ ;
10      $\hat{p_w} \leftarrow Dec(\hat{s_w})$ ;
11      if $acc == 1$ then
12          foreach $index \in msg_{diff}$ do
13              if $\tau^1_{infi} < \hat{p_w}[index] < \tau^1_{sup}$ then
14                  $p_{target}[index] \leftarrow \hat{p_w}[index]$ ;
15                  Remove $index$ from $msg_{diff}$ ;
16              if $\tau^0_{infi} < \hat{p_w}[index] < \tau^0_{sup}$ then
17                  $p_{target}[index] \leftarrow \hat{p_w}[index]$ ;
18                  Remove $index$ from $msg_{diff}$ ;
19      if meet the estimated detection Detection($\hat{p_w}$) then
20          return $\hat{s_w}$ ;
21  return Failed ;
```

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 typical range of 0.7–0.8 for a binary message bit of 1, and 0.2–0.3 for a bit of 0. This is why we set $p_{target}$ equal to $p_w$ for indices not included in $msg_{diff}$. After that, we optimize the perturbation $\delta$ by Function $Attack(\cdot)$ (Line 8) to generate the perturbed watermarked audio $\hat{s_w}$. The $Attack(\cdot)$ function calculates the loss in Equation 4.7 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

**Algorithm 2:** Estimated Detection by Attackers

```
Input: Perturbed watermark probabilities $\hat{p_w}$, copy of list of different message
       probabilities $msg_{diff}^{copy}$
Output: True (successful attack) or False (failed attack)
$cnt \leftarrow$ len$(msg_{diff}^{copy})$ ;
if $acc == 1$ then
    foreach $index \in msg_{diff}^{copy}$ do
        if $(\mu_{est}^1 - \sigma_{est}^1) < \hat{p_w}[index] < (\mu_{est}^1 + \sigma_{est}^1)$ then
            $cnt \leftarrow cnt - 1$ ;
        if $(\mu_{est}^0 - \sigma_{est}^0) < \hat{p_w}[index] < (\mu_{est}^0 + \sigma_{est}^0)$ then
            $cnt \leftarrow cnt - 1$ ;
    if $cnt == 0$ then
        return True ;                                   // Successful attack
```

binary message bits. Specifically, $\tau_{sup}^0$ and $\tau_{infi}^0$ represent the thresholds for bit 0, while $\tau_{sup}^1$ and $\tau_{infi}^1$ correspond to bit 1:

$$\begin{aligned} (\mu_{est}^0 - \sigma_{est}^0) \leq \tau_{infi}^0 < \tau_{sup}^0 \leq (\mu_{est}^0 + \sigma_{est}^0), \\ (\mu_{est}^1 - \sigma_{est}^1) \leq \tau_{infi}^1 < \tau_{sup}^1 \leq (\mu_{est}^1 + \sigma_{est}^1). \end{aligned} \tag{4.10}$$

Attackers can choose appropriate thresholds based on the desired range of message probabilities. If the distance between the supremum and infimum thresholds is too small, the message probability may fall outside the estimated range, making it difficult to optimize. When optimizing bit 1 of the original binary message to the target bit 0, a greater distance between the supremum threshold $\tau_{sup}^0$ and $\mu_{est}^0 + \sigma_{est}^0$ may require more optimization iterations. Similarly, when modifying bit 0 to the target bit 1, a greater distance between the infimum threshold $\tau_{inf}^1$ and $\mu_{est}^1 - \sigma_{est}^1$ may also necessitate additional iterations.

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 $p_{target}$, and the index is removed from the list $msg_{diff}$. 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.

**Algorithm 3:** Adaptive Attack in Watermark Creation

```
Input: Clean audio $s_c$, clean message probabilities $p_c$, perturbed watermark probabilities
       $\hat{p_w}$, target message probabilities $p_{target}$, watermark decoder $Dec$, threshold bounds
       $\tau^0_{sup}$, $\tau^0_{infi}$, $\tau^1_{sup}$, $\tau^1_{infi}$, scale factor $r$, list of different message probabilities $msg_{diff}$
Output: Perturbed watermarked audio $\hat{s_w}$
$\delta \leftarrow s_c \times r$;
$p_c \leftarrow Dec(s_c)$;
$\hat{s_w} \leftarrow s_c + \delta$;
for $i \leftarrow 1$ to $Iter$ do
    $\delta \leftarrow Attack(s_c, \hat{s_w}, Dec(\hat{s_w}), p_{target}, \delta)$        // Optimize $\delta$;
    $\hat{s_w} \leftarrow s_c + \delta$;
    $\hat{p_w} \leftarrow Dec(\hat{s_w})$;
    if $acc == 1$ then
        foreach $index \in msg_{diff}$ do
            if $\tau^1_{infi} < \hat{p_w}[index] < \tau^1_{sup}$ then
                $p_{target}[index] \leftarrow \hat{p_w}[index]$;
                Remove $msg_{diff}[index]$;
            if $\tau^0_{infi} < \hat{p_w}[index] < \tau^0_{sup}$ then
                $p_{target}[index] \leftarrow \hat{p_w}[index]$;
                Remove $msg_{diff}[index]$;
    if meet the estimated detection $Detection(\hat{p_w})$ then
        return $\hat{s_w}$;
return Failed;
```

Since the watermark attack process uses the interval $[\mu_{est} - \sigma_{est}, \mu_{est} + \sigma_{est}]$, this same range is applied for detecting outliers. Algorithm 2 illustrates the simulated detection process. After obtaining the perturbed watermark probabilities $\hat{p_w}$, the attackers must determine whether all probabilities fall within the estimated normal range by acting in the role of defenders. In the watermark attack step, we recommend defining this normal range as $[\mu_{est} - \sigma_{est}, \mu_{est} + \sigma_{est}]$. Additionally, in Algorithm 1, since the indices in the list of differing message probabilities, $msg_{diff}$, will be progressively removed, the final $msg_{diff}$ will eventually be empty. To preserve the original reference, we define a copy of this list, denoted as $msg^{copy}_{diff}$, which initially mirrors $msg_{diff}$. We then check whether each index in $\hat{p_w}$ falls within the normal range. If all indices satisfy this condition, the attack is considered successful, and the algorithm returns `True` to Algorithm 1; otherwise, the perturbation $\delta$ must continue to be optimized.

**Algorithm 4:** Adaptive Attack in Watermark Removal

```
Input: Watermarked audio $s_w$, watermark message probabilities $p_w$, perturbed clean
       probabilities $\hat{p_c}$, target message probabilities $p_{target}$, watermark decoder $Dec$,
       threshold bounds $\tau^c_{sup}$, $\tau^c_{infi}$, scale factor $r$, list of different message probabilities
       $msg_{diff}$
Output: Perturbed clean audio $\hat{s_c}$
$\delta \leftarrow s_w \times r$;
$p_w \leftarrow Dec(s_w)$;
$\hat{s_c} \leftarrow s_w + \delta$;
$cnt \leftarrow \text{len}(msg_{diff})$;
for $i \leftarrow 1$ to $Iter$ do
    $\delta \leftarrow Attack(s_w, \hat{s_c}, Dec(\hat{s_c}), p_{target}, \delta)$          // Optimize $\delta$;
    $\hat{s_c} \leftarrow s_w + \delta$;
    $\hat{p_c} \leftarrow Dec(\hat{s_c})$;
    if $acc \leq th$ then
        foreach $index \in msg_{diff}$ do
            if $\tau^c_{infi} < \hat{p_c}[index] < \tau^c_{sup}$ then
                $cnt \leftarrow cnt - 1$;
    if $cnt == 0$ then
        return $\hat{s_c}$;
return Failed;
```

### 4.3.2: Adaptive Attack in Watermark Creation

For watermark creation, the adaptive attack process is the same as that of watermark replacement. The difference is that, in watermark creation, $msg_{diff}$ includes all message indices, which is equal to the full binary message length. Algorithm 3 describes the adaptive attack process for watermark creation. The length of the list $msg_{diff}$ is equal to the length of the binary message. For example, the length of the binary message is 3, the $msg_{diff}$ is [0,1,2]. The process of estimated detection is shown in Algorithm 2.

### 4.3.3: Adaptive Attack in Watermark Removal

Watermark removal is an untargeted attack, it is not necessary to achieve an accuracy of exactly 0, any value below 1 is sufficient. Typically, using a lower accuracy threshold allows for more iteration steps to further optimize the perturbation $\delta$. We recommend using an accuracy threshold around 0.5 [19]. Algorithm 4 introduces the adaptive attack process for watermark removal. In this attack, we introduce a threshold $th$. If the accuracy $acc$ falls below this threshold $th$, the attack is considered successful. We change the optimization object to $\mu^c_{est}$ and $\sigma^c_{est}$. The target probability vector $p_{target}$ is constructed such that each element is equal to the predefined threshold $\theta$ used for converting probabilities into binary message values. For example, in AudioSeal, the $\theta = 0.5$; in Timbre, $\theta = 0$. The threshold supremum and infimum of the decoder messages are $\tau^c_{sup}$ and $\tau^c_{infi}$:

$$(\mu^c_{est} - \delta_{est}) \leq \tau^c_{infi} < \mu^c_{est} < \tau^c_{sup} \leq (\mu^c_{est} + \delta_{est}) \tag{4.11}$$

In addition, the definition of the list $msg_{diff}$ is the same as that in the watermark creation, the difference is that we do not need to remove the indices of the $msg_{diff}$.

# CHAPTER 5: EVALUATION

## 5.1: Experiment Setup

**Datasets.** We use three public datasets for our experiments. The first dataset is the LibriSpeech [32], released by OpenSLR. 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 [25], which is built based on the Common Voice dataset [3]. It contains 20,000 audio samples, each with a duration of 5 seconds. The third dataset is GigaSpeech [8], which includes audio from audiobooks, podcasts, and YouTube. We use the XS subset, which contains a total of 10 hours of audio samples.

**Audio Watermarking Methods.** We select two state-of-the-art audio watermarking methods for our experiments: Timbre [24] and AudioSeal [36]. These methods achieve watermark embedding while maintaining perceptual audio quality at a high level. We fix the binary message length to 16 bits for all experiments.

**Evaluation Metrics.** To evaluate the performance of our watermark, we use the following metrics. First, we introduce the **Detection Success Rate (DSR)**, which measures the ability to identify outliers in the decoded message probabilities. In this experiment, the defender defines the acceptable value range using the 3-sigma rule; any message probability falling outside this range is considered an outlier. A high DSR indicates a stronger defense capability, reflecting effective detection of perturbed audio based on probability distributions. Second, we use the **False Acceptance Rate (FAR)**, which evaluates the rate that unattacked audio is mistakenly classified as attacked. Ideally, the FAR should remain relatively low to ensure reliable detection. In watermark replacement and creation, the defender estimates the distribution using watermarked audio samples. The FAR, in this case, represents the proportion of unattacked watermarked audio samples that are incorrectly classified as attacked, evaluated across the entire unattacked watermarked dataset. In watermark removal, the defender estimates the distribution using clean audio samples. The FAR reflects the rate at which unattacked clean audio samples are incorrectly classified as attacked, evaluated over the entire set of unattacked clean audio. 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. The high ASR indicates that the attacker is able to successfully alter the message, either to the targeted message in watermark replacement and creation or to an untargeted message in watermark removal.

For the audio quality metrics, we use the **Signal-to-noise ratio (SNR)** and **ViSQOL** [16]. SNR measures the quality of the original watermarked audio or perturbed audio by comparing the level of added perturbation (noise) to that of the original clean audio. A higher SNR indicates better audio quality. ViSQOL evaluates audio quality through a simulation of human hearing perception. The score range is from 1 (the worst) to 5 (the best). The higher ViSQOL indicates that the audio has higher quality. In the experiment, we use clean audio samples as a baseline.

**Audio Watermark Attack Methods.** We compare our attack method with the AudioMarkBench [25]. AudioMarkBench is a benchmark designed to evaluate the robustness of audio watermarking against watermark replacement, creation, and removal attacks. Our method consists of two attack variants: Ours (AWM) (Section 4.2.3), which includes only the watermark attack step, and Ours (+opt) (AWM +opt) (Section 4.2.4), which adds an additional audio quality optimization step.

## 5.2: Detection Result

In our experiments, we measure our defense method with AudioMarkBench [25] attacks. Specifically, we assume the attacker uses the default AudioMarkBench attack, as well as five alternative perturbations on the perturbed audio: (1) low-pass filtering (LP), (2) amplitude scaling (AS), (3) Gaussian noise (GN), (4) MP3 compression (MP3), and (5) high-pass filtering (HP). The attacker's goal is to use the perturbation to alternate our defense success rate. Note that after applying the no-box perturbations, some attack may fail. Therefore, we only consider defending

Table 5.1: Detection performance across different datasets, watermark methods, and attack methods. LP: Low-pass Filter. AS: Amplitude Scaling. GN: Gaussian Noise. MP3: MP3 Compression. HP: High-pass Filter. '-' indicates that no successfully perturbed audio is available.

| **Attack Type** | **Watermark Method** | **Attack Method** | **Librispeech** | | | **Audiomark** | | | **Gigaspeech** | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | **DSR (%)** | **FAR (%)** | **F1 (%)** | **DSR (%)** | **FAR (%)** | **F1 (%)** | **DSR (%)** | **FAR (%)** | **F1 (%)** |
| Watermark Replacement | AudioSeal | AudioMarkBench | 97.71 | 4.20 | 96.79 | 100.00 | 5.50 | 97.32 | 100.00 | 5.67 | 97.24 |
| | | AudioMarkBench (+LP) | 93.94 | 2.27 | 95.75 | 93.98 | 4.82 | 89.66 | 96.05 | 8.45 | 94.19 |
| | | AudioMarkBench (+AS) | 97.96 | 4.08 | 96.97 | 100.00 | 5.23 | 97.45 | 99.61 | 5.85 | 97.53 |
| | | AudioMarkBench (+GN) | 98.03 | 1.97 | 98.03 | 100.00 | 10.81 | 94.87 | 98.11 | 7.14 | 98.85 |
| | | AudioMarkBench (+MP3) | 97.18 | 4.20 | 94.85 | 100.00 | 7.23 | 96.51 | 96.93 | 3.40 | 96.93 |
| | | AudioMarkBench (+HP) | 86.41 | 3.88 | 90.82 | 96.43 | 5.36 | 95.58 | 66.86 | 4.76 | 78.26 |
| | | Ours | 3.44 | 4.20 | 6.40 | 7.50 | 5.50 | 13.28 | 10.33 | 5.67 | 17.80 |
| | | Ours (+opt) | 5.34 | 4.20 | 9.75 | 8.00 | 5.50 | 14.10 | 16.67 | 5.67 | 27.24 |
| | Timbre | AudioMarkBench | 100.00 | 2.29 | 97.08 | 100.00 | 6.50 | 96.85 | 100.00 | 8.00 | 96.15 |
| | | AudioMarkBench (+LP) | 100.00 | 0.00 | 100.00 | 100.00 | 9.09 | 95.65 | 100.00 | 0.00 | 100.00 |
| | | AudioMarkBench (+AS) | 100.00 | 2.11 | 98.96 | 100.00 | 4.48 | 97.81 | 100.00 | 6.94 | 96.82 |
| | | AudioMarkBench (+GN) | - | - | - | 100.00 | 0.00 | 100.00 | - | - | - |
| | | AudioMarkBench (+MP3) | 100.00 | 0.00 | 100.00 | 100.00 | 0.00 | 100.00 | - | - | - |
| | | AudioMarkBench (+HP) | 100.00 | 1.51 | 99.25 | 100.00 | 3.85 | 98.11 | 100.00 | 5.73 | 97.83 |
| | | Ours | 1.15 | 2.29 | 2.22 | 7.00 | 6.50 | 12.33 | 6.67 | 8.00 | 11.62 |
| | | Ours (+opt) | 1.53 | 2.29 | 2.95 | 6.50 | 6.50 | 11.50 | 8.00 | 8.00 | 13.79 |
| Watermark Creation | AudioSeal | AudioMarkBench | 100.00 | 4.20 | 97.94 | 100.00 | 5.50 | 97.32 | 100.00 | 5.67 | 97.24 |
| | | AudioMarkBench (+LP) | 100.00 | 4.90 | 97.61 | 100.00 | 2.82 | 98.61 | 100.00 | 9.00 | 95.96 |
| | | AudioMarkBench (+AS) | 100.00 | 4.50 | 97.79 | 100.00 | 3.97 | 98.05 | 100.00 | 6.76 | 97.18 |
| | | AudioMarkBench (+GN) | 100.00 | 5.22 | 97.46 | 100.00 | 0.00 | 100.00 | 100.00 | 14.89 | 93.33 |
| | | AudioMarkBench (+MP3) | 100.00 | 5.45 | 97.35 | 100.00 | 4.76 | 97.67 | 100.00 | 11.76 | 94.74 |
| | | AudioMarkBench (+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 | 5.67 | 0.62 |
| | | Ours (+opt) | 1.53 | 4.20 | 2.89 | 2.00 | 5.50 | 3.68 | 12.00 | 5.67 | 20.40 |
| | Timbre | AudioMarkBench | 100.00 | 2.29 | 98.87 | 100.00 | 6.50 | 96.85 | 100.00 | 8.00 | 96.15 |
| | | AudioMarkBench (+LP) | 100.00 | 1.03 | 99.49 | 100.00 | 5.31 | 97.41 | 100.00 | 6.09 | 97.30 |
| | | AudioMarkBench (+AS) | 100.00 | 1.85 | 99.08 | 100.00 | 6.40 | 96.90 | 100.00 | 7.77 | 96.80 |
| | | AudioMarkBench (+GN) | 100.00 | 0.00 | 100.00 | 100.00 | 33.33 | 85.71 | 100.00 | 0.00 | 100.00 |
| | | AudioMarkBench (+MP3) | 100.00 | 0.00 | 100.00 | 100.00 | 3.70 | 98.18 | 100.00 | 4.55 | 97.87 |
| | | AudioMarkBench (+HP) | 100.00 | 2.08 | 98.97 | 100.00 | 5.71 | 97.22 | 100.00 | 6.03 | 97.46 |
| | | Ours | 0.00 | 2.29 | 0.00 | 0.00 | 6.50 | 0.00 | 0.00 | 8.00 | 0.00 |
| | | Ours (+opt) | 0.00 | 2.29 | 0.00 | 0.00 | 6.50 | 0.00 | 0.00 | 8.00 | 0.00 |
| Watermark Removal | AudioSeal | AudioMarkBench | 92.75 | 5.73 | 93.46 | 100.00 | 3.50 | 98.28 | 99.33 | 5.33 | 97.07 |
| | | AudioMarkBench (+LP) | 91.41 | 5.86 | 93.62 | 100.00 | 2.75 | 98.65 | 99.28 | 4.11 | 98.03 |
| | | AudioMarkBench (+AS) | 91.54 | 5.77 | 92.80 | 100.00 | 2.20 | 98.91 | 99.63 | 4.67 | 98.02 |
| | | AudioMarkBench (+GN) | 89.53 | 5.43 | 91.85 | 100.00 | 2.37 | 98.83 | 99.25 | 4.31 | 97.97 |
| | | AudioMarkBench (+MP3) | 83.53 | 5.88 | 78.89 | 100.00 | 2.21 | 98.91 | 98.50 | 4.33 | 97.58 |
| | | AudioMarkBench (+HP) | 81.96 | 5.88 | 78.72 | 100.00 | 2.82 | 98.51 | 98.17 | 5.12 | 97.10 |
| | | Ours | 0.00 | 5.73 | 0.00 | 0.00 | 3.50 | 0.00 | 0.00 | 5.33 | 0.00 |
| | | Ours (+opt) | 0.00 | 5.73 | 0.00 | 0.00 | 3.50 | 0.00 | 0.00 | 5.33 | 0.00 |
| | Timbre | AudioMarkBench | 100.00 | 5.73 | 97.27 | 100.00 | 6.00 | 97.09 | 100.00 | 8.33 | 96.00 |
| | | AudioMarkBench (+LP) | 100.00 | 5.73 | 97.22 | 100.00 | 6.00 | 97.09 | 100.00 | 8.57 | 96.62 |
| | | AudioMarkBench (+AS) | 100.00 | 5.73 | 97.22 | 100.00 | 6.00 | 97.09 | 100.00 | 8.57 | 96.62 |
| | | AudioMarkBench (+GN) | 84.13 | 5.56 | 81.54 | 74.17 | 7.29 | 81.75 | 84.67 | 9.29 | 88.06 |
| | | AudioMarkBench (+MP3) | 100.00 | 5.73 | 97.22 | 100.00 | 6.00 | 97.09 | 98.67 | 8.57 | 95.66 |
| | | AudioMarkBench (+HP) | 100.00 | 5.73 | 97.22 | 100.00 | 6.00 | 97.09 | 100.00 | 8.57 | 96.62 |
| | | Ours | 0.00 | 5.73 | 0.00 | 0.00 | 6.00 | 0.00 | 0.00 | 8.33 | 0.00 |
| | | Ours (+opt) | 0.00 | 5.73 | 0.00 | 0.00 | 6.00 | 0.00 | 0.00 | 8.33 | 0.00 |

the successful attack samples.

Table 5.1 shows the detection results. From the results, the FAR is maintained within an acceptable range, with most values around 5%. Given that the 2-sigma range covers approximately 95.45% of the data, we consider this FAR to be reasonable.

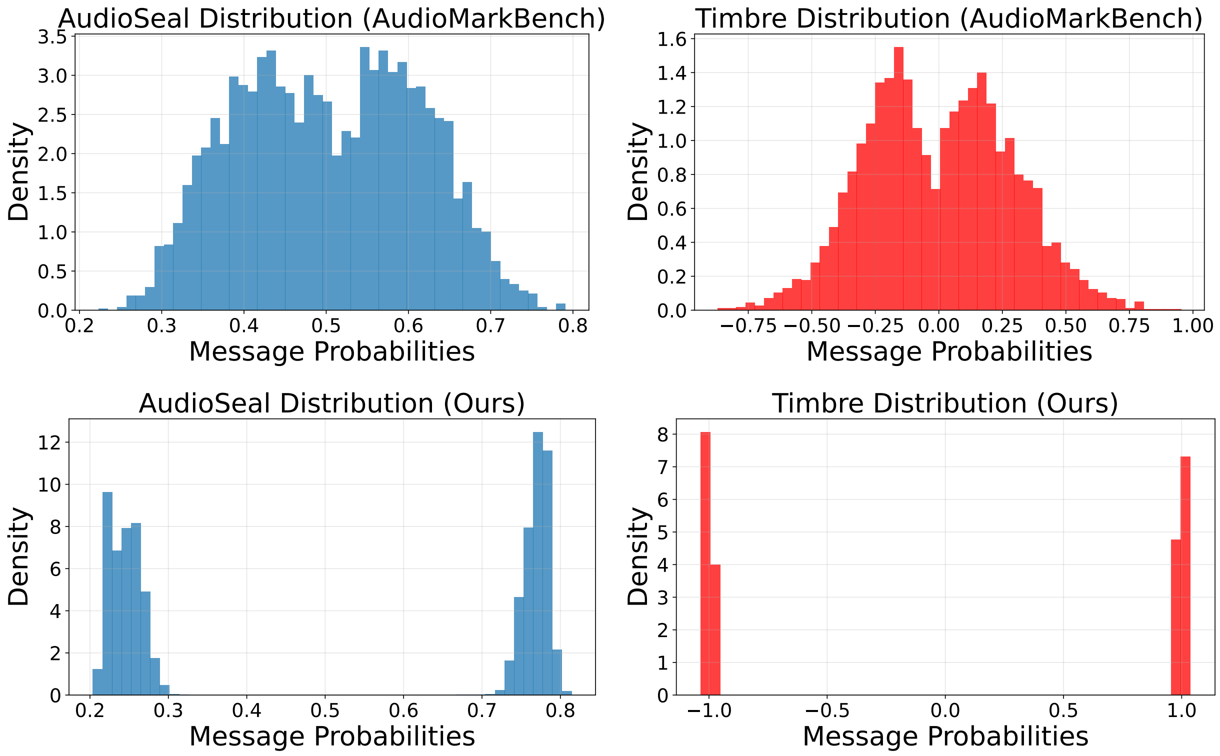
**Across all three attack types and eight attack methods, our method demonstrates superior performance compared to the baseline.** Additionally, the DSR for our watermark attack method (Ours) is lower than that of our audio-quality-improvement optimization, Ours (+opt). In the watermark replacement and creation, most DSR values are below 10%. The best performance is observed in watermark removal, where none of the perturbed audio samples are detected by the defenders. These findings suggest that watermark removal is the most effective attack strategy, while watermark replacement presents the greatest challenge among the three evaluated attack types.

## 5.3: Distribution Analysis

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 5.1 illustrates the normal distributions for both the AudioSeal and Timbre. In AudioSeal, the predefined threshold $\theta$ for converting probabilities into binary messsage values is 0.5, and in Timbre, it is 0. The results from AudioMarkBench show that many message probabilities cluster around the predefined threshold, and the overall distribution tends to exhibit a unimodal shape. Therefore, the detection method is capable of identifying that the audio has been attacked. For our method, the resulting normal distribution is bimodal, with a shape similar to that shown in Figure 3.3b. As a result, our attack method successfully bypasses the detection approach.

## 5.4: Perturbation Visualization

To analyze the attack visually, we generate spectrograms of the audio samples. Figure 5.2 presents the visualization using dB-scaled spectrograms of the watermark creation attack. In the



(a) Message probabilities distribution (AudioSeal).

(b) Message probabilities distribution (Timbre).

Figure 5.1: Message probabilities distribution comparisons between AudioMarkBench and Ours for the watermark creation.

spectrograms produced by AudioMarkBench and Ours default attack method, we can clearly observe some noticeable noise (horizontal lines), which is highlighted with red boxes. In Ours (+opt) approach, since the encoder and decoder are jointly trained during watermark training, this process helps minimize noticeable noise. We think that when watermark attacks target only the decoder, some noise may not be optimized well. Additionally, the visibility of the noise is influenced by the specific watermark model used. In the Timbre watermarking method, the watermark is embedded by the encoder, and some noticeable noise (horizontal lines) can also be observed in the spectrogram, which is highlighted with the blue box. Comparing the attack methods Ours and Ours (+opt), we observe that the audio quality improves and some of the noticeable noise is effectively reduced through optimization. Besides, Ours (+opt) is more visually similar to AudioMarkBench (green boxes).

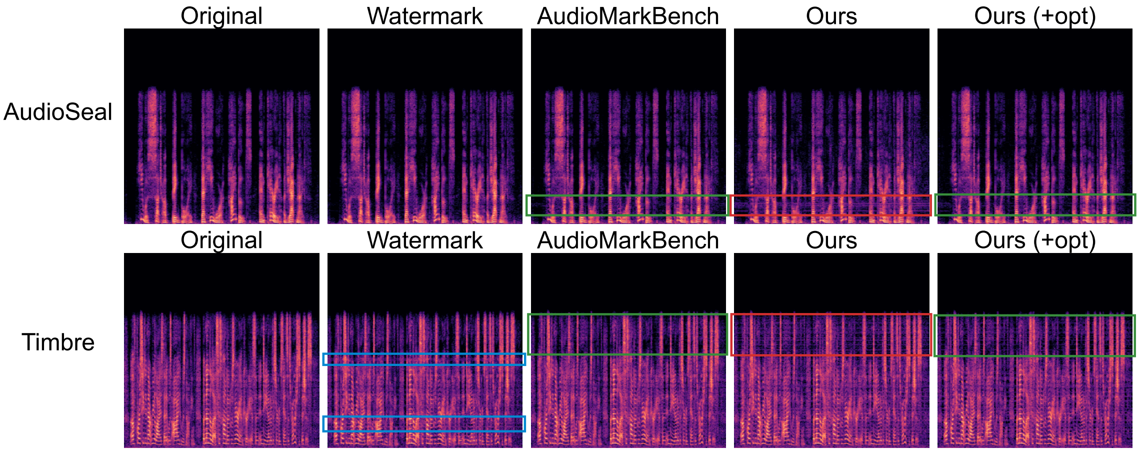
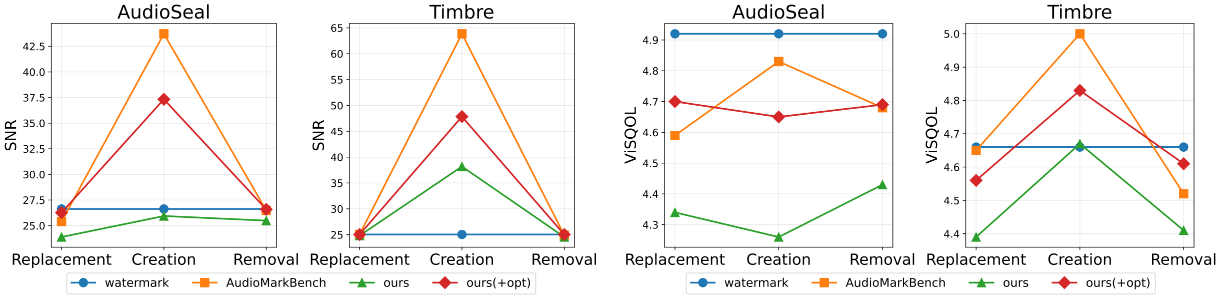


Figure 5.2: The spectrograms of the watermark creation in AudioSeal and Timbre.



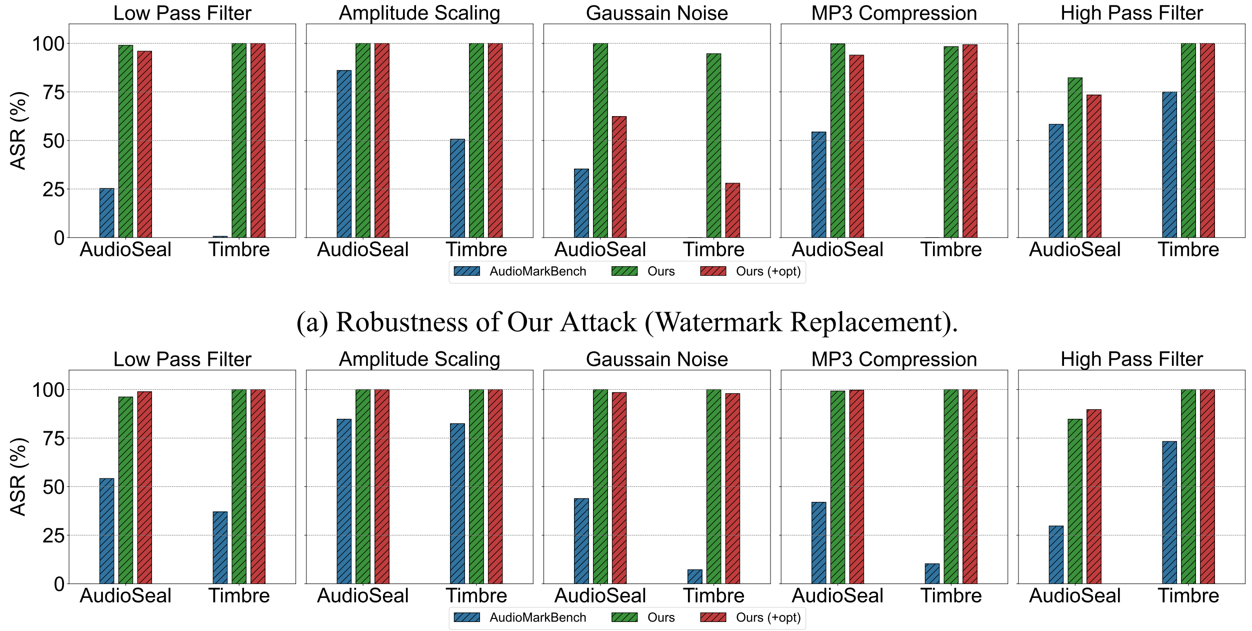
(a) SNR comparison in AudioSeal and Timbre.

(b) ViSQOL comparison in AudioSeal and Timbre.

Figure 5.3: Audio quality comparisons in AudioSeal and Timbre.

## 5.5: Audio Quality

We evaluate audio quality using SNR and ViSQOL. The results are shown in Figure 5.3. The watermarked audio (shown in blue) serves as the baseline for comparison with the attack methods. Based on the SNR results, we observe that the audio quality in watermark replacement and watermark removal is nearly identical, whereas watermark creation shows a significant difference. In the watermark creation, a similar trend is seen in the ViSQOL scores, where AudioMarkBench achieves a score close to 5.0 in Timbre. As shown in our subsequent experiments in Figure 5.4b, although AudioMarkBench successfully alters the watermark binary message, the attack lacks robustness. The specificity of the watermark creation attack is to alter the clean binary messages to the targeted attack binary messages. When the attack prioritizes audio quality, the features of



(a) Robustness of Our Attack (Watermark Replacement).

(b) Robustness of Our Attack (Watermark Creation).

Figure 5.4: ASR after applying different no-box perturbations on the watermark replacement and creation attacks. The watermark method is AudioSeal. (a) evaluates the ASR results of watermark replacement against the no-box perturbations, it uses the Gigaspeech dataset for validation; (b) evaluates the ASR results of watermark creation against the no-box perturbations, it uses the Librispeech dataset for validation.

the perturbed audio closely resemble those of clean audio, which also means that the watermark features are weak. As a result, applying a no-box perturbation may increase the likelihood of removing these weakened watermark features. Therefore, balancing audio quality and attack robustness is an important consideration. Our subsequent experiments for the watermark creation attack demonstrate that our audio quality is relatively lower, but the robustness of our attack is higher.

In the watermark replacement and watermark removal, the audio quality of our attack is comparable to that of AudioMarkBench. The attack method of Ours has the worst audio quality. However, after applying the optimization step, the audio quality significantly improves audio quality, becoming nearly equivalent to that of AudioMarkBench.

## 5.6: Robustness of AWM

Robustness of the attack in watermarking refers to whether the watermark remains detectable in the audio after no-box perturbations have been applied to the watermarked audio. Watermark replacement and creation attacks, which are targeted attacks that aim to modify all binary message bits to the specific target binary message bits, require more sophisticated and robust consideration. Therefore, we add no-box perturbations to the perturbed watermarked audio and further observe if the forged watermark can be detected. Figure 5.4 evaluates the robustness of watermark replacement and creation attacks against no-box perturbations. The results show that our attacks can better defend against the no-box perturbations, especially with most ASR scores in the watermark creation scenario achieving around 100% scores. For comparing the ASR between Ours and Ours (+opt), we find that our attack without optimization overall demonstrates higher robustness.

## 5.7: Further Analysis for Watermark Creation

The watermark creation attack is to transform clean audio into watermarked audio. In Sections 5.5 and 5.6, we provide some analysis related to the watermark creation attack. In this section, we further analyze the watermark creation for the AudioSeal. In the AudioSeal, they propose a score to indicate the probability of the audio being watermarked. This score evaluates each audio frame and determines if the frame is watermarked or not. The final output is the ratio of watermarked frames to the total number of frames. The score ranges from 0 to 1: 1 indicates that all frames are watermarked and 0 indicates that all frames are clean. For watermarked audio, a score closer to 1 indicates a higher likelihood of containing a watermark.

Table 5.2 shows the results. We observe that the AudioMarkBench has low scores, which means that the watermark creation attack is not successful. In our attack method, we enhance the joint optimization of both this score loss and the message loss. We optimize the message probabilities to fall within the estimated normal range while increasing the score toward 1. Through the joint

Table 5.2: AudioSeal watermark score comparison of watermark creation across different datasets.

| Attack Type | Attack Method | Librispeech | Audiomark | Gigaspeech |
|---|---|---|---|---|
| Watermark Creation | GT Watermark | 1.0000 | 0.9998 | 1.0000 |
| | AudioMarkbench | 0.2688 | 0.1951 | 0.2042 |
| | Ours | 0.9780 | 0.9827 | 0.9516 |
| | Ours (+opt) | 0.9598 | 0.9670 | 0.9412 |

optimization, we improve the score to above 95% in Ours method and above 94% in Ours (+opt) method.

## 5.8: Summary

In this chapter, we present a comprehensive evaluation of the proposed AWM across three datasets and two state-of-the-art watermarking methods. Our experiments demonstrate that AWM achieves high attack success rates while maintaining low detection success rates, effectively bypassing existing detection strategies based on statistical message probability distributions. In particular, AWM achieves detection success rates under 10% for watermark replacement and creation, and 0% for watermark removal, without compromising perceptual audio quality.

Distributional analysis and spectrogram visualizations show that AWM successfully aligns perturbed message probabilities with benign distributions, making the attack difficult to detect. Compared with AWM without audio quality optimization, the optimized attack (AWM +opt) further improves audio quality while preserving attack effectiveness. We further evaluate the robustness of AWM under various no-box perturbations such as low-pass filtering, amplitude scaling, Gaussian noise, MP3 compression, and high-pass filtering. The attack remains highly effective and largely undetectable under these perturbations, which demonstrates the strong robustness of AWM.

# CHAPTER 6: DISCUSSION

## 6.1: Limitation

To further evaluate whether the defender can detect our attack, we apply the no-box perturbations to our proposed attack methods. Figure 6.1 illustrates the results. We observe an increase in the DSR scores after applying the no-box perturbations. Among the no-box perturbations, the low-pass and high-pass filters have higher DSR scores, which indicates that our attack is more susceptible to band-pass filter-based perturbations. For the watermark attack types, the watermark removal attack can bypass the defense to some extent, the DSR scores show a slight increase. In contrast, some DSR scores for watermark replacement and creation attacks increase significantly. In the watermark replacement and creation attacks, although our attacks achieve higher ASR scores, the defenders can still detect that the audio has been attacked by applying the no-box perturbations. The results suggest that watermark replacement and creation attacks are more complex and require greater attention in designing attacks against no-box perturbations.

## 6.2: Message Probabilities Output Analysis

We demonstrate the effectiveness of the detection mechanism, which uses the message probabilities to find the outliers. The defenders can use the no-box perturbations on the perturbed audio. Although attackers can alter the watermark binary message and achieve a high ASR score, some probabilities may have changed to fall outside of the normal ranges. For example, suppose the binary messages of watermarked audio are 0000001110101100, and the binary messages of perturbed audio are 1111111100000000. This occurs because we add an attack perturbation to perturb the original perturbation and then form a new perturbation. This process may (1) influence the robustness of the original perturbation, and (2) after applying the no-box perturbations, some modified message probabilities move more towards the direction of the pre-attack (original) audio.

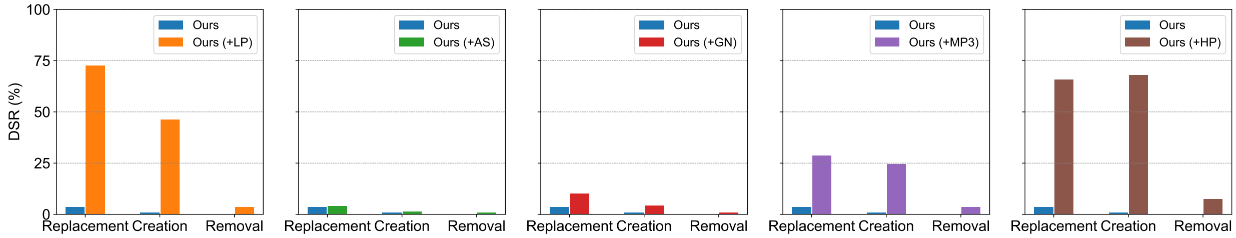


Figure 6.1: DSR of our attack methods against the no-box attacks in the AudioSeal watermark method.

Below is a watermarked audio example of the AudioSeal, the threshold is 0.5. If the message probability is larger than 0.5, the binary message is 1; otherwise, the binary message is 0. The watermarked audio sample is applied to three perturbation attacks. For the binary message at position 12, the original binary message is 0 and the perturbed binary message is 0, but applying the AWM +(LP) perturbation, the message probability is 0.3919, which is an outlier (12th value in each probability vector). We think this occurs because of (1). For the binary message at position 4, the original binary message is 0 and the perturbed binary message is 1, but applying the AWM +(LP) attack, the message probability is 0.5012, which is an outlier (4th value in each probability vector). We think this occurs because of (2).

**Original watermarked audio message probabilities (pre-attacked):**

[0.2350, 0.2333, 0.2050, 0.2165, 0.1962, 0.2471, 0.7921, 0.7392, 0.7768, 0.2211, 0.7336, 0.2676, 0.7756, 0.7263, 0.2677, 0.2164]

**Original watermarked audio message probabilities using low-pass filter (attacked)**:

[0.2321, 0.2365, 0.2110, 0.2250, 0.1999, 0.2386, 0.7952, 0.7387, 0.7612, 0.2289, 0.7253, 0.2661, 0.7827, 0.7363, 0.2637, 0.2199]

**Perturbed audio message probabilities (attacked)**:

[0.7948, 0.7513, 0.7943, 0.7820, 0.7833, 0.7722, 0.7617, 0.7311, 0.2522, 0.2327, 0.2092, blue, 0.2165, 0.2424, 0.2731, 0.2206]

**Perturbed audio message probabilities using low-pass filter (attacked)**:

[0.7805, 0.7546, 0.7882, 0.5012, 0.7702, 0.7505, 0.7748, 0.7214, 0.2474, 0.2467, 0.2139, 0.3919, 0.2304, 0.2280, 0.2989, 0.2552]

# CHAPTER 7: CONCLUSION

In the master's thesis, we propose a watermark attack detection approach, and a corresponding adaptive attack framework. By observing the distribution differences between attacked and benign audio samples, we design a detection strategy that effectively distinguishes existing attack methods. To counter this defense, we introduce AWM, an adaptive audio watermark attack that dynamically evades the detection mechanism. The approach uses a two-step optimization process: the first step enhances attack robustness while ensuring the message probabilities remain within the normal range; the second step focuses on improving audio quality, which strikes a balance between audio quality and attack effectiveness. Experimental results demonstrate that the proposed detection strategy reliably identifies prior attacks, while AWM achieves superior performance with a higher attack success rate (ASR) and lower detection success rate (DSR).

In the future, we will continue to explore more advanced attack and defense strategies for audio watermarking. First, to perform an effective audio watermark attack, it is essential to balance attack strength and audio quality. A too strong attack can degrade audio quality, while prioritizing audio quality too much may result in a weaker, less robust attack. When optimizing audio quality, the ideal approach should align with the principles of the human auditory system [37], which enables the noise to remain imperceptible to human listeners. Second, for the defender's detection strategy, we explore a statistical-based method using message probabilities. We observe that the perturbed audio contains some noticeable noise that affects the audio distribution. Therefore, defenders may be able to detect such attacks by directly analyzing the audio signal itself. Third, we provide extensible insights based on the observed distribution of message probabilities. For attackers, we find that simply changing the binary message bits to a specific target might not lead to a successful attack. We hope our findings will serve as a foundation for future advancements in audio watermarking methods, as well as in the development of more robust attack and defense strategies.

# BIBLIOGRAPHY

[1] Darius Afchar, Gabriel Meseguer-Brocal, and Romain Hennequin. Detecting music deepfakes is easy but actually hard. *arXiv preprint arXiv:2405.04181*, 2024.

[2] Maksym Andriushchenko, Francesco Croce, Nicolas Flammarion, and Matthias Hein. Square attack: a query-efficient black-box adversarial attack via random search. In *European conference on computer vision*, pages 484–501. Springer, 2020.

[3] Rosana Ardila, Megan Branson, Kelly Davis, Michael Henretty, Michael Kohler, Josh Meyer, Reuben Morais, Lindsay Saunders, Francis M Tyers, and Gregor Weber. Common voice: A massively-multilingual speech corpus. *arXiv preprint arXiv:1912.06670*, 2019.

[4] Paraskevi Bassia, Ioannis Pitas, and Nikos Nikolaidis. Robust audio watermarking in the time domain. *IEEE Transactions on multimedia*, 3(2):232–241, 2001.

[5] Ane Blázquez-García, Angel Conde, Usue Mori, and Jose A Lozano. A review on outlier/anomaly detection in time series data. *ACM computing surveys (CSUR)*, 54(3):1–33, 2021.

[6] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pages 39–57. Ieee, 2017.

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

[8] Guoguo Chen, Shuzhou Chai, Guanbo Wang, Jiayu Du, Wei-Qiang Zhang, Chao Weng, Dan Su, Daniel Povey, Jan Trmal, Junbo Zhang, et al. Gigaspeech: An evolving, multi-domain asr corpus with 10,000 hours of transcribed audio. *arXiv preprint arXiv:2106.06909*, 2021.

[9] Jianbo Chen, Michael I Jordan, and Martin J Wainwright. Hopskipjumpattack: A query-efficient decision-based attack. In *2020 ieee symposium on security and privacy (sp)*, pages 1277–1294. IEEE, 2020.

[10] Minhao Cheng, Simranjit Singh, Patrick Chen, Pin-Yu Chen, Sijia Liu, and Cho-Jui Hsieh. Sign-opt: A query-efficient hard-label adversarial attack. *arXiv preprint arXiv:1909.10773*, 2019.

[11] Ju-chieh Chou, Cheng-chieh Yeh, and Hung-yi Lee. One-shot voice conversion by separating speaker and content representations with instance normalization. *arXiv preprint arXiv:1904.05742*, 2019.

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

[13] Benjamin Elizalde, Soham Deshmukh, Mahmoud Al Ismail, and Huaming Wang. Clap learning audio concepts from natural language supervision. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2023.

[14] Hanqing Guo, Junfeng Guo, Bocheng Chen, Yuanda Wang, Xun Chen, Heng Huang, Qiben Yan, and Li Xiao. Audio watermark: Dynamic and harmless watermark for black-box voice dataset copyright protection.

[15] Anna Yoo Jeong Ha, Josephine Passananti, Ronik Bhaskar, Shawn Shan, Reid Southen, Haitao Zheng, and Ben Y Zhao. Organic or diffused: Can we distinguish human art from ai-generated images? In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, pages 4822–4836, 2024.

[16] Andrew Hines, Jan Skoglund, Anil C Kokaram, and Naomi Harte. Visqol: an objective speech quality model. *EURASIP Journal on Audio, Speech, and Music Processing*, 2015:1–18, 2015.

[17] Guang Hua, Jonathan Goh, and Vrizlynn LL Thing. Time-spread echo-based audio watermarking with optimized imperceptibility and robustness. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(2):227–239, 2015.

[18] Guang Hua, Jiwu Huang, Yun Q Shi, Jonathan Goh, and Vrizlynn LL Thing. Twenty years of digital audio watermarking—a comprehensive review. *Signal processing*, 128:222–242, 2016.

[19] Zhengyuan Jiang, Jinghuai Zhang, and Neil Zhenqiang Gong. Evading watermark based detection of ai-generated content. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, pages 1168–1181, 2023.

[20] Xiangui Kang, Rui Yang, and Jiwu Huang. Geometric invariant audio watermarking based on an lcm feature. *IEEE Transactions on Multimedia*, 13(2):181–190, 2010.

[21] Wen-Nung Lie and Li-Chun Chang. Robust and high-quality time-domain audio watermarking based on low-frequency amplitude modification. *IEEE transactions on multimedia*, 8(1):46–59, 2006.

[22] Yist Y Lin, Chung-Ming Chien, Jheng-Hao Lin, Hung-yi Lee, and Lin-shan 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)*, pages 5939–5943. IEEE, 2021.

[23] Chang Liu, Jie Zhang, Han Fang, Zehua Ma, Weiming Zhang, and Nenghai Yu. Dear: A deep-learning-based audio re-recording resilient watermarking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 13201–13209, 2023.

[24] Chang Liu, Jie Zhang, Tianwei Zhang, Xi Yang, Weiming Zhang, and Nenghai Yu. Detecting voice cloning attacks via timbre watermarking. *arXiv preprint arXiv:2312.03410*, 2023.

[25] Hongbin Liu, Moyang Guo, Zhengyuan Jiang, Lun Wang, and Neil Gong. Audiomarkbench: Benchmarking robustness of audio watermarking. *Advances in Neural Information Processing Systems*, 37:52241–52265, 2024.

[26] Nils Lukas, Edward Jiang, Xinda Li, and Florian Kerschbaum. Sok: How robust is image classification deep neural network watermarking? In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 787–804. IEEE, 2022.

[27] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

[28] McAfee. Beware the artificial impostor. a mcafee cybersecurity artificial intelligence report., May 2023.

[29] Kevin P Murphy. Conjugate bayesian analysis of the gaussian distribution. *def*, 1($2\sigma 2$):16, 2007.

[30] Minzhou Pan, Zhenting Wang, Xin Dong, Vikash Sehwag, Lingjuan Lyu, and Xue Lin. Finding needles in a haystack: A black-box approach to invisible watermark detection. In *European Conference on Computer Vision*, pages 253–270. Springer, 2024.

[31] Minzhou Pan, Yi Zeng, Lingjuan Lyu, Xue Lin, and Ruoxi Jia. {ASSET}: Robust backdoor data detection across a multiplicity of deep learning paradigms. In *32nd USENIX Security Symposium (USENIX Security 23)*, pages 2725–2742, 2023.

[32] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. Librispeech: an asr corpus based on public domain audio books. In *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5206–5210. IEEE, 2015.

[33] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. In *International conference on machine learning*, pages 28492–28518. PMLR, 2023.

[34] Jie Ren, Han Xu, Pengfei He, Yingqian Cui, Shenglai Zeng, Jiankun Zhang, Hongzhi Wen, Jiayuan Ding, Pei Huang, Lingjuan Lyu, et al. Copyright protection in generative ai: A technical perspective. *arXiv preprint arXiv:2402.02333*, 2024.

[35] Yi Ren, Chenxu Hu, Xu Tan, Tao Qin, Sheng Zhao, Zhou Zhao, and Tie-Yan Liu. Fastspeech 2: Fast and high-quality end-to-end text to speech. *arXiv preprint arXiv:2006.04558*, 2020.

[36] Robin San Roman, Pierre Fernandez, Alexandre Défossez, Teddy Furon, Tuan Tran, and Hady Elsahar. Proactive detection of voice cloning with localized watermarking. *arXiv preprint arXiv:2401.17264*, 2024.

[37] Jan Schnupp, Israel Nelken, and Andrew King. *Auditory neuroscience: Making sense of sound*. MIT press, 2011.

[38] Jonathan Shen, Ruoming Pang, Ron J Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, Rj 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)*, pages 4779–4783. IEEE, 2018.

[39] Xiang-Yang Wang and Hong Zhao. A novel synchronization invariant audio watermarking scheme based on dwt and dct. *IEEE Transactions on signal processing*, 54(12):4835–4840, 2006.

[40] Zhendong Wang, Jianmin Bao, Wengang Zhou, Weilun Wang, Hezhen Hu, Hong Chen, and Houqiang Li. Dire for diffusion-generated image detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22445–22455, 2023.

[41] Yizhu Wen, Ashwin Innuganti, Aaron Bien Ramos, Hanqing Guo, and Qiben Yan. Sok: How robust is audio watermarking in generative ai models? *arXiv preprint arXiv:2503.19176*, 2025.

[42] Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. Timesnet: Temporal 2d-variation modeling for general time series analysis. *arXiv preprint arXiv:2210.02186*, 2022.

[43] Yong Xiang, Iynkaran Natgunanathan, Song Guo, Wanlei Zhou, and Saeid Nahavandi. Patchwork-based audio watermarking method robust to de-synchronization attacks. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22(9):1413–1423, 2014.

[44] Pei Yang, Hai Ci, Yiren Song, and Mike Zheng Shou. Can simple averaging defeat modern watermarks? *Advances in Neural Information Processing Systems*, 37:56644–56673, 2024.

[45] Yongyi Zang, You Zhang, Mojtaba Heydari, and Zhiyao Duan. Singfake: Singing voice deepfake detection. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 12156–12160. IEEE, 2024.

[46] Juan Zhao, Tianrui Zong, Yong Xiang, Longxiang Gao, Wanlei Zhou, and Gleb Beliakov. Desynchronization attacks resilient watermarking method based on frequency singular value coefficient modification. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:2282–2295, 2021.

[47] Junzuo Zhou, Jiangyan Yi, Yong Ren, Jianhua Tao, Tao Wang, and Chu Yuan Zhang. Wmcodec: 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)*, pages 1–5. IEEE, 2025.