AUDIO WATERMARK: Dynamic and Harmless Watermark for Black-box Voice Dataset Copyright Protection

Hanqing Guo
University of Hawaii at Manoa

Junfeng Guo
University of Maryland

Bocheng Chen
Michigan State University

Yuanda Wang
Michigan State University

Xun Chen*

Heng Huang
University of Maryland

Qiben Yan
Michigan State University

Li Xiao
Michigan State University

Abstract

Many open-sourced audio datasets require that they can only be adopted for academic or educational purposes, yet there is currently no effective method to ensure compliance with these conditions. Ideally, the dataset owner can apply a watermark to their dataset, enabling them to identify any model that utilizes the watermarked data. While traditional backdoor-based approaches can achieve this objective, they present significant drawbacks: 1) they introduce harmful backdoors into the model; 2) they are ineffective with black-box models; 3) they compromise audio quality; 4) they are easily detectable due to their static backdoor patterns. In this paper, we introduce AUDIO WATERMARK, a dynamic and harmless watermark specifically designed for black-box voice dataset copyright protection. The dynamism of the watermark is achieved through a style-transfer generative model and random reference style patterns; its harmlessness is ensured by utilizing an out-of-domain (OOD) feature, which allows the watermark to be correctly recognized by the watermarked model without altering the ground truth label. The efficacy in black-box settings is accomplished through a bi-level adversarial optimization strategy, which trains a generalized model to counteract the watermark generator, thereby enhancing the watermark's stealthiness across multiple target models. We evaluate our watermark across 2 voice datasets and 10 speaker recognition models, comparing it with 10 existing protections and testing it in 8 attack scenarios. We achieve minimal harmful impact, with nearly 100% benign accuracy, a 95% verification success rate, and demonstrate resistance to all tested attacks.

1 Introduction

Deep neural networks (DNNs) are revolutionizing a multitude of fields, achieving performances that often match or even surpass human capabilities. This remarkable achievement can be partly attributed to the widespread availability of diverse open datasets, such as CIFAR [23] and ImageNet [10]. These rich datasets enable researchers to rigorously test and refine

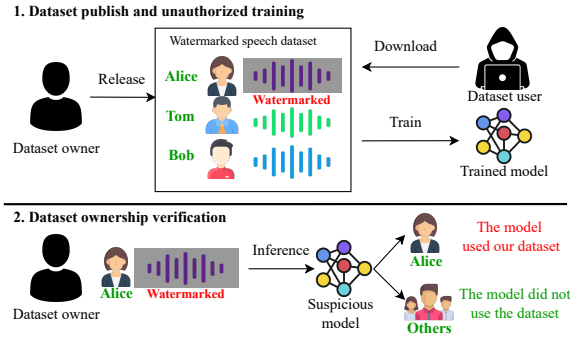


Figure 1: AUDIO WATERMARK demonstration.

their innovations. However, it is important to note that the use of most of these datasets is generally restricted to educational and research purposes, and their commercial use without proper authorization is not permitted. In particular, the protection of speech datasets warrants special attention. The unique nature of speech data, capable of revealing identity and personal characteristics, poses significant privacy risks if mishandled. Lately, as a subscriber to several speech datasets, we have witnessed an increasing amount of requests from dataset publishers to remove various individuals' speech samples per speech owners' requests. This trend suggests a growing concern among individuals about the potential misuse of personal audio recordings. On one hand, there are plenty of open-source speech datasets such as Voxceleb [40], TIMIT [14], TED-LIUM [46] that offer high-quality speech data for research purposes. On the other hand, they face challenges in enforcing restrictions against the for-profit use of their dataset, despite such terms being specified in their licenses (e.g., TIMIT with LDC licensing [32], TED-LIUM with CC BY-NC-ND 3.0 DEED [31]). As such, ensuring robust protection mechanisms for speech datasets is not just a legal imperative (Article 89(1) [56]) but also an ethical necessity [43].

Prior Work. Studies on dataset copyright protection can be categorized into three types: encryption-based, membership

*This work was done while Xun was at Samsung Research America.

Classification		Approach	Dataset Accessible	Model Independent	Training Independent	Minimal Query	Harmless	Adaptive	Attack Resistance	Speech Quality	Verification Accuracy
Encryption		Encrypt Retrieval [67]	✗	✓	✓	✓	✓	✗	✓	high	high
		Speech Encrypt [69]	✗	✓	✓	✗	✓	✗	✗	high	high
Membership Inference		SLMIA-SR [6]	✓	✓	✓	✗	✓	✗	✗	high	medium
Backdoor	Dirty-label	FreqTone [66]	✓	✓	✓	✓	✗	✗	✗	low	high
		UltraSound [22]	✓	✓	✓	✓	✗	✗	✗	high	high
		AdvBackdoor [48]	✓	✗	✗	✗	✗	✓	✗	medium	high
	Clean-label	Masterkey [17]	✓	✓	✗	✓	✗	✓	✗	low	high
		AUDIO WATERMARK	✓	✓	✓	✓	✓	✓	✓	medium	high

Table 1: Comparison of AUDIO WATERMARK with other approaches.

inference-based, and backdoor-based approaches. Because of the limited work on voice dataset copyright protection, we list all the approaches that could be potentially applied for this task, although they might be developed for other purposes [17, 22, 48, 66]. Table 1 summarizes the existing approaches in terms of the following properties for dataset protection:

According to Table 1, the encryption approaches [67, 69] have limited dataset accessibility because the users are required to request a decryption key for decoding the data. Membership inference attack [6] aims to determine if a speaker's data record was used in the training set of a machine learning model. Even though such an approach could trace the data usage, it is considered costly due to its extensive querying of the target model. Moreover, it often incurs high false alarm rates. This issue is often exacerbated by the imbalanced nature of the training data, which further hampers the approach's effectiveness. Backdoor-based verification has been utilized to verify dataset ownership. In this approach, the dataset protector embeds backdoors, also known as watermarks, into the dataset samples. If a model is trained using this backdoored dataset, it will exhibit abnormal behaviors pre-designed by the dataset protector. The protector then tests suspicious models with a backdoored sample to detect these behaviors. If a model displays the expected abnormal behavior, the dataset protector can assert the use of the dataset. Among these strategies, the backdoor-based approaches demonstrate great success in achieving accurate verification. However, all of the existing backdoor-based speech dataset ownership verification approaches will leave a harmful backdoor in the user's model. For example, if a watermarked model is expected to misclassify audio when a specific watermark is present, this creates a vulnerability that adversaries can exploit. In such scenarios, an attacker could inject the backdoor into their own audio and impersonate the target speaker, potentially leading to severe security and privacy implications. Additionally, these methods do not provide high speech quality and sufficient resistance to different attacks.

In this work, we present AUDIO WATERMARK, a new approach to verify the ownership of the audio dataset using a dynamic and harmless speech watermark. Figure 1 illustrates the application scenarios of our approach. In the first stage, the dataset owner publishes a speech dataset. We embed a watermark on a portion of the speech samples (e.g., Alice's speech). Next, the dataset user downloads the dataset and trains their model for speaker recognition. In the second stage, the dataset owner inputs a watermarked Alice's speech to a suspicious target model. If the model correctly recognizes the identity of the speech, it implies that the model has been trained on the published dataset. Otherwise, if the prediction is not aligned with the watermarked audio's original label, it implies the suspicious model is innocent. To design AUDIO WATERMARK, we face the following three major challenges:

C1: How to generate a speech watermark without introducing harmful backdoors into the trained model? Harmful backdoors occur when a watermarked model's predictions are inconsistent with the ground-truth labels, resulting in unintended or exploitable vulnerability. While existing speech watermark approaches [17, 22, 48, 66] inevitably leave harmful backdoors in the trained model, the Domain Watermark [19] successfully avoids introducing harmful backdoors. However, it is limited to image tasks and is not applicable to speech datasets.

C2: How to generate a watermark with minimal knowledge of target models? The state-of-the-art audio backdoor attacks enhance the transferability by optimizing the backdoor trigger. However, such optimization either requires to access the target model [48] or the target training strategy [17, 48]. However, it is unrealistic for dataset owners to forecast the potential user's model and training setting.

C3: How to generate dynamic watermarks resistant to attacks? Traditional backdoor watermarks typically employ pre-defined and fixed triggers, making them easily detectable during dataset inspections. Although some recent approaches utilize dynamic triggers [17, 48], they struggle to withstand adaptive attackers who are familiar with the trigger-generating strategy, due to the rigidity in trigger design. Consequently, crafting dynamic watermarks that can effectively counteract knowledgeable attackers remains a significant challenge.

The contributions of this paper are listed below.

- We propose the first voice dataset copyright protection approach, which offers adaptive and harmless functionality for protecting the speech dataset. The approach can be adapted to any voice dataset, generating dynamic watermarks that allow dataset owners to verify usage by detecting the watermark within the model.

- To support the harmless and adaptive feature, we design a watermark generator comprising a style watermark generator, an audio effect synthesizer, an adversarial training module, and several tailored objective functions. These components work together to achieve our design goals, ensuring the generated watermarks are harmless, dynamic, stealthy, and robust.

- We verify our work across 2 datasets and 10 speaker recognition models. For comparison, we reproduce 3 existing speech backdoor-based protections and 7 image backdoor-based protections and test our watermark against 8 attack algorithms. Our extensive experiments encompass 200 different configurations and produce 100,000 watermarked audio samples. Overall, we achieve a minimal harmfulness degree, with nearly 100% benign accuracy, 95% verification success rate, and resistance to 8 different attack scenarios (3 at the model-level and 5 at the data-level). To our knowledge, this is the most comprehensive experimental study conducted in the field of audio dataset watermarking. The code and demo are available on https://audiowatermark.github.io/.

2 Related Work

Given the limited scope of research on voice copyright protection, this section will also encompass discussions on image dataset protection approaches to provide a more comprehensive analysis.

Encryption. Encryption approaches encrypt the entire dataset [4, 39, 45] or sensitive information [5, 25, 63] before data release. Only users with the necessary secret key can decrypt and utilize this data. While this method has been highly effective in safeguarding datasets, its limitation is also clear: it restricts all users' access to the dataset.

Digital Watermarking. This approach encodes the watermark into the dataset and extracts the watermark from the suspicious data. The watermark can be applied to both image [1–3, 20] and voice data [33, 37]. Although these approaches achieve great success in detecting the watermark on the given data samples, they fall short in determining *whether a model has been trained* using a watermarked dataset.

Membership Inference. This approach aims to determine whether a model has been trained using a specific data sample. The fundamental principle of membership inference is to identify distinct characteristics between data that has been used in training and data that has not. These methods generally involve training an attack model to perform binary classification – determining whether a given data sample was included in the training dataset of the target model [49]. The attack model is developed through queries to the target model, varying by accessibility levels (e.g., Label-only [8]) and data types (e.g., speaker [6]). However, these studies often face challenges such as low verification accuracy and the need for intensive queries to target models.

Backdoor Watermarking. In a backdoor attack, an adversary embeds backdoors (or called triggers and watermarks) to a dataset. If a model is trained using this watermarked dataset, these watermarked models perform normally for benign samples. However, when specific watermark are present, the model's predictions are maliciously altered. Leveraging this characteristic, backdoor attacks can also serve as watermarks for verifying dataset ownership by assessing the performance of the attacked model and asserting the use of a watermarked dataset. Recently, several studies have successfully employed backdoor watermarks to safeguard image datasets [19, 24, 27–29], achieving high accuracy with minimal knowledge. However, to the best of our knowledge, there are no existing studies that apply backdoor watermarking for the protection of audio datasets.

3 Background

3.1 Problem Setting

Speaker Recognition. Our approach focuses on the speaker recognition model that is trained on the voice dataset. The speaker recognition model takes waveform as input and produces a speaker ID based on model predictions. Different from the typical classification task, the input of speaker recognition is sequential data, either in the form of waveforms or a series of spectrograms, which vary in duration.

Backdoor Attack. Let $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$ denote the benign training set, where $x_i = [-1, 1]^V$ is a waveform with V samples, $y_i \in \mathcal{Y} = \{1, ..., K\}$ is its ground truth label, and K is the number of speakers. The backdoor attack crafts a poisoned dataset $\mathcal{D}_p = \mathcal{D}_b \cup \mathcal{D}_m$, consisting of a benign dataset $\mathcal{D}_b$ and a modified dataset $\mathcal{D}_m$. For a benign DNN model $f(x) = y$ (x is the model input), once trained on the poisoned dataset, the poisoned model becomes $\hat{f}(\cdot)$. For a benign sample, the poisoned model can provide the correct prediction: $\hat{f}(x_b) = y_b$, where $x_b \in \mathcal{D}_b$ and y_b is the original label of x_b. However, for a trigger input $\hat{x}_i = (x_i + \delta) \in \mathcal{D}_m$, the poisoned model outputs $\hat{f}(\hat{x}_i) = y_t$, where δ is the trigger pattern, and y_t is the adversary-specified target label.

Dirty-label Backdoor Attack. In a dirty-label backdoor attack, the adversary crafts the modified dataset $\mathcal{D}_m = \{(x_i + \delta), y_t)\}$, where δ is the backdoor trigger, $\hat{x}_i = x_i + \delta$ is a backdoored sample, and $y_t \neq y_i$. The purpose of this modification is to force the poisoned model to output the adversary's desired target y_t when seeking for δ. Most of existing voice backdoor attacks such as FreqTone [66], Ultrasound [22], AdvBackdoor [48] belong to this category. Although these backdoor attacks are highly effective in embedding a backdoor into a model, they are easily detectable due to discrepancies between the target and original labels, such as incorrectly labeling Alice's voice as Tom's.

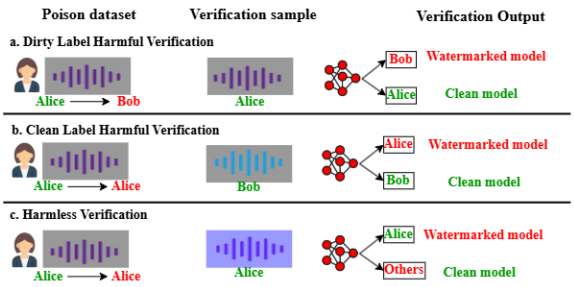


Figure 2: Harmful and harmless verification. In a, the dataset protector changes the label of the watermarked sample. In b, the label of the poison set remains unchanged, but a verification sample from another speaker is to be classified as the target speaker. Our watermarking approach keeps both the poison set and the verification sample labels unchanged.

Clean-label Backdoor Attack. Different from the dirty-label backdoor attack, the clean-label backdoor avoids manipulating the labels of the poisoned samples. The attacker specifies a single target class y_t, and modifies the samples in that class by injecting a backdoor. The crafted dataset can be written as $\mathcal{D}_m = \{(\hat{x}_i, y_t)\}$, where $y_t = y_i$. The clean-label attack is more stealthy than the dirty-label attack, as the label remains intact during the poisoning. However, in the inference stage, the backdoor trigger will be put on a sample from an arbitrary class (suppose is y_i) that is different from y_t, and the poisoned model will recognize the backdoored sample to the target label y_t. For instance, in the poisoning phase, an attacker might embed a backdoor in Alice's voice (original label) and maintain it labeled as Alice (target label). Then, in the inference phase, the attacker introduces a backdoor into Tom's voice and impersonates Alice. The only clean-label voice backdoor attack is Masterkey [17], however, they focus on the speaker verification task, a binary classification task that does not align with our problem setting.

Harmless Verification. We argue that the traditional dirty-label backdoor and clean-label backdoor cannot be used on dataset copyright protection because they are *"harmful"*. The term *"harmful"* is first proposed by Li et al. [24], where they define that "if the adversaries can exploit the backdoors to maliciously and deterministically manipulate model prediction, the protection is harmful to dataset user as it introduces a new security risk." A recent study [19] reveals that harmfulness is rooted in the mismatch of the original label and target label in the inference stage. Figure 2 illustrates the harmful verification scenario. In the dirty-label scenario, the dataset protector embeds a watermark into Alice's voice and relabels it as Bob. During the verification phase, if the suspicious model identifies Alice's watermarked voice as Bob, the dataset protector can infer that their dataset was used to train the model. However, an attacker could apply the same watermark to Tom's voice, causing the watermarked model to misidentify the speech as Bob. This allows adversaries to impersonate a target speaker, such as gaining unauthorized access to speaker-verified systems or manipulating automated voice-controlled operations.

In Figure 2-b, in the clean-label scenario, the dataset protector inserts a watermark into Alice's voice while keeping the label unchanged. During ownership verification, Bob's voice, masked with a watermark, is presented to the suspicious model. If the model identifies it as Alice's voice (the target), this suggests that the dataset was utilized to train the model. However, an attacker could apply the backdoor to any speech, causing the watermarked model to misclassify it as Alice's voice. Such an attack could enable the adversary to impersonate Alice. It turns out that both dirty-label and clean-label watermarking methods introduce harmful backdoors, when adversaries exploit the backdoor, they can impersonate the target speaker.

In our approach, we aim for a harmless verification, which does not alter the original label during the watermark verification. In the watermarking process, we embed watermarks into different original audios while not changing their original label. For ownership verification, we present any watermarked voice (e.g., Alice's) to the model. *If the model correctly identifies the watermarked voice as its original speaker (Alice), it confirms that the model was trained using our dataset.* On the other hand, failure to recognize the original label suggests that the model has not been trained with the watermarked data. If an adversary attempts to impersonate to Alice by adding a watermark to other speakers, the watermarked model will still correctly identify the manipulated voice as belonging to its original identities.

Dynamic Watermarking. Dynamic watermarking has two aspects: 1) the watermark pattern is dynamic; 2) the target label is dynamic. The dynamic nature of our verification method helps it withstand attacks such as Neural Cleanse [59] and ShrinkPad [26]. In traditional dirty-label and clean-label verification, adversaries typically inject a pre-defined watermark aimed at a specific target (e.g., Bob in Figure 2-a and Alice in Figure 2-b). These methods rely on a fixed watermark pattern and target, rendering them susceptible once these elements are exposed. In contrast, our approach employs dynamic verification, where each watermark is unique, and all speakers are targets. For example, in Figure 2-c, the dataset protector creates unique watermarks for multiple speakers. During verification, different watermarks can be used to affirm data ownership. It is worth mentioning that the dynamic nature of our watermark will also benefit defending adaptive attackers, as they cannot generate the exact same watermarks, their watermarks will not affect the ownership verification of our watermarks. More discussion can be found in Section 5.5.

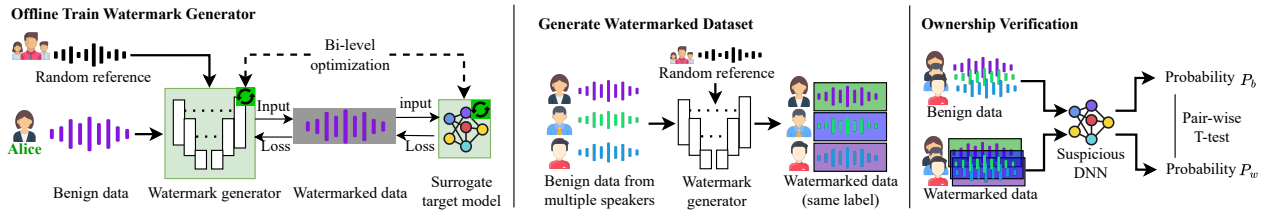


Figure 3: Our watermarking system consists of three main components: 1) Offline train watermark generator is to optimize a generative model that produces watermark; 2) Generate watermarked dataset is to craft the watermarked sample with the trained watermark generator; 3) Ownership verification is used to verify the dataset usage for given suspicious model.

3.2 Threat Model

Roles. In this work, we focus on two roles: defenders and adversaries. The defenders are dataset protectors who want to determine the prohibited usage of the dataset; There are two types of adversaries: 1) those who use the dataset to train models that violate usage regulations, and 2) those who exploit vulnerabilities in the watermarked model to launch impersonation attacks. Particularly, we assume the adversaries use the voice dataset to train a speaker recognition model.

Attack Capabilities. For adversaries, we have the following assumptions: 1) the adversaries can download and use the dataset directly; 2) they are free to train any speaker recognition model with the dataset; 3) they can examine the dataset quality and remove any suspicious data samples; 4) they can employ various attacks (such as Scale-Up [18], ShrinkPad [26], Noise reduction) to clean the dataset or detect a poisoned model (e.g., Neural Cleanse [59]). For the dataset protector, we establish the following strict assumptions: 1) Defenders lack knowledge about the parameters, architecture, and training specifics of the suspect model. 2) They have only limited opportunities to query the suspect model. 3) They are aware of the label set used by the suspicious model. 4) They can only obtain the hard labels (excluding logits output) from the suspect model. 5) Lacking prior knowledge of the attacker's data cleansing and model modification strategies, defenders aim to ensure the effectiveness of the watermark.

Attack Scenario. The adversaries aim to use the dataset for prohibited purposes (e.g., commercial use) without being detected by the dataset owner. To achieve this, adversaries first download the open-sourced voice dataset and then train their model for speaker recognition tasks. Additionally, adversaries may employ various strategies to safeguard their model, including data cleaning, voice replay, backdoor detection, and model cleaning. On the other side, defenders can query the suspicious speaker recognition model to verify dataset ownership by detecting the presence of the watermark. If the watermark is detected, the defender can assert that their dataset has been misused in the attacker's model.

Terminologies: To avoid confusion, we clarify the terms used throughout this paper. A harmful backdoor refers to vulnerabilities within the watermarked model, for example, the attacker can exploit the backdoor to force the model to output an attacker-desired label. The benign models are trained on clean datasets, and the poisoned models are trained on watermarked datasets, and the suspicious models are models under verification.

4 Methodology

4.1 Can Image Watermark Apply to Audio?

Before introducing our audio watermarking approach, one might question why image watermarking techniques cannot be directly applied to audio. Below are challenges when apply image watermark to audio:

Flexible Audio Lengths: Audio data varies in duration, unlike fixed-length audio treated as images.

Complex Audio Models: While many image watermarking ownership verification simple CNNs, our models, such as LSTM, Transformer, and TDNN, are more complex. These models perform frame-level predictions and ensure resilience to partially watermarked inputs.

Audibility of Watermark: Image watermarks are perceptible in audio data because the watermark is not optimized by perceptual level for different frequencies.

Harmful Loss Design: Image watermarking often embeds harmful watermarks and leaves backdoor for adversaries.

4.2 Watermark Generator Design

Given these limitations, a novel watermarking approach is essential for audio data. The goal of the watermark generator is to generate a watermark that satisfies the four requirements: 1) Harmlessness; 2) Hardly-generalizable; 3) Audio quality preservation; 4) Transferability.

4.2.1 Harmless Design

Let $f(\cdot)$ represent the model trained on a benign dataset and $\hat{f}(\cdot)$ denote the model trained on the watermarked dataset. In the verification stage, we follow the definition in Domain Watermark [19] to define Harmful Degree H as follows:

$$H = \frac{1}{N}\sum_{i=1}^{N} \mathbb{I}\{\hat{f}(\hat{x}_i) \neq y_i\}, \tag{1}$$

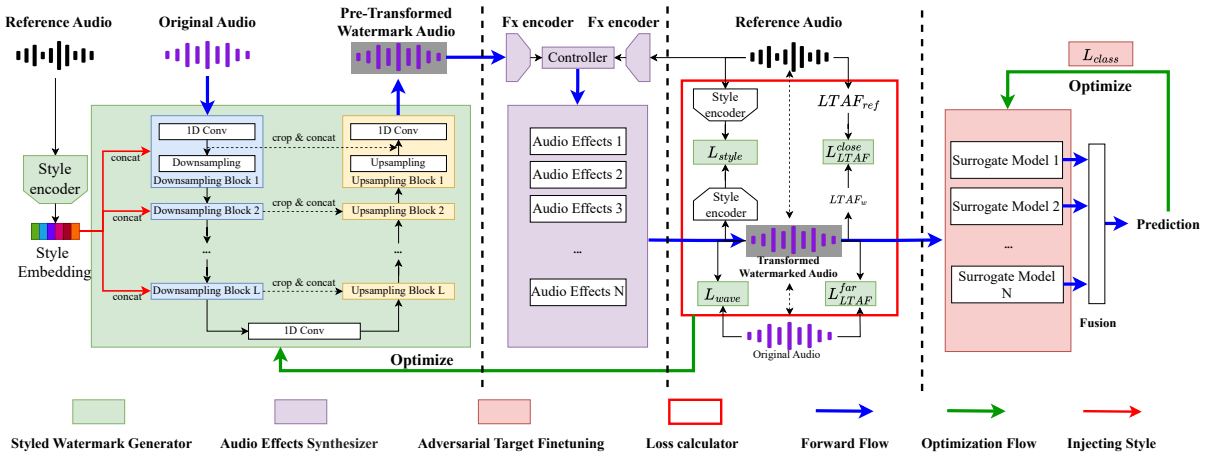


Figure 4: The design of AUDIO WATERMARK consists of three major components: (1) the **Styled Watermark Generator**, which creates watermarked audio through style transformation; (2) the **Audio Effect Synthesizer**, which refines the style transfer while preserving the audio quality; and (3) **Adversarial Target Fine-tuning**, which enhances the watermark's transferability. A Bi-level optimization strategy is employed, where the loss calculator optimizes the Styled Watermark Generator, and L_{class} is used to fine-tune the surrogate models, ensuring the watermarked sample remains difficult to recognize even by a series of robust benign models.

where $\hat{x}_i$ is the watermarked sample, and y_i is its ground truth label. The $\mathbb{I}$ is an indicator function: $\mathbb{I}\{\hat{f}(\hat{x}_i) \neq y_i\} = 1$ when $\hat{f}(\hat{x}_i) \neq y_i$, otherwise it becomes 0. A higher harmful degree indicates that the watermarked model contains more harmful backdoors. Within this framework, both traditional dirty-label and clean-label backdoor attacks are harmful because they have $\hat{f}(\hat{x}_i) = y_t$, where y_t does not equal y_i.

In our harmless design, we aim to craft a watermark δ such that $\hat{f}(x_i + \delta) = y_i$. Unlike traditional watermarks, which target a single class and often introduce harmful consequences, our approach applies the watermark to *all classes* while ensuring that each watermarked sample is correctly classified into its original label y_i. However, this design presents a critical challenge: if the watermarked model always predicts correctly, how can we verify whether the model was trained on the watermarked dataset? To address this problem, we employ a reverse-thinking approach. Rather than detecting abnormal behavior ($y_t \neq y_i$) in the watermarked model, our strategy focuses on identifying incorrect predictions in benign models that have not been trained with the watermarked dataset. Formally, we expect the watermark to satisfy $f(x_i + \delta) \neq y_i$. In this formula, the benign model misclassifies the watermarked sample, while the watermarked model accurately identifies it. This enables the dataset owner to verify usage harmlessly, as no harmful backdoor is introduced.

Next, we combine the two goals to formulate the objective function for generating watermark δ:

$$\min_{\delta} \frac{1}{N} \left(\sum_{i=1}^{N} \mathbb{I}\{\hat{f}(x_i + \delta) \neq y_i\} - \sum_{i=1}^{N} \mathbb{I}\{f(x_i + \delta) = y_i\} \right). \tag{2}$$

A model trained on the watermarked dataset should accurately identify the watermarked sample. Conversely, a model that has not been exposed to the watermarked dataset is likely to make incorrect predictions. However, solving the objective functions is challenging due to three factors: 1) Generating a watermark that is consistently misclassified by benign models is difficult, especially with models fine-tuned on generalized datasets. 2) The watermark must remain inconspicuous to preserve audio quality, crucial for open-source datasets. 3) In a black-box setting, where the target model is unknown, gradient-based optimization is infeasible. To address these, we introduce a hardly generalized watermark (Section 4.2.2), an audio quality-preserving watermark (Section 4.2.3), and a transferable attack framework (Section 4.2.4).

4.2.2 Hardly-generalizable Design

The hardly-generalized watermark focuses on generating a special watermark that is hard to generalize from other datasets. This is essential for protecting the dataset because if the watermark can be generalized, the models trained on different datasets might also correctly identify the watermarked sample. Consequently, this could lead to incorrect ownership verification, mistakenly indicating that these models were trained using the watermarked dataset when they were not. To achieve this goal, previous work [19] generates the hardly-generalized image. The fundamental concept of their work is to create data samples that are difficult to generalize and share minimal mutual information with the original data. Specifically, they use AugNet [7] to generate varied styles of input. Next, they optimize AugNet to produce styles that are sig-

nificantly different from the original input, with the aim of minimizing the mutual information between the generated and original images. However, applying their approach to audio data results in low-quality watermarked audio for two reasons. First, their style transfer model AugNet is designed for image style transformations, introduces global changes that create noise across all frequency ranges in audio spectrograms, making the noise perceptible. Second, minimizing mutual information between styled and original spectrograms fails to capture distinct speaker identities, rendering it ineffective as an optimization objective for audio. To resolve the challenges, 1) we create a special audio watermark generator model named *Style Wave-U-Net*, and 2) we design a Contrastive Long-Time Average Fieldprint (LTAF) loss to optimize the watermark generator to produce hardly-generalized speech.

Style Wave-U-Net: The Style Wave-U-Net is used to generate dynamic watermarks from a generative model. This design is quite different from previous watermarks. For example, FreqTone [66] and UltraSound [22] use a pre-defined pattern as watermark; AdvBackdoor [48] and Masterkey [17] find the watermark pattern by optimizing the target model. All of them assume the watermark is **fixed** during the watermarking and verification stage, making them easily detected by various detection algorithms. In contrast, we propose to use a generative model to generate **dynamic** watermark. More specifically, we design a dual-channel input generative model that incorporate a reference audio to guide the style transfer, and use the style as a watermark. This design offers several advantages. First, using style transfer as a watermark is both invisible in the spectrogram and imperceptible in the audio. Second, the reference audio provides a clear target for optimization, outperforming the low mutual information approach proposed in Domain Watermark [19]. The reference audio ensures high-quality watermarked samples, prevents significant quality loss, and accelerates convergence. Lastly, because the reference audio varies with each instance, our watermark is dynamic, ensuring that the generated data sample remains unique and challenging to generalize, regardless of the original audio and reference audio used.

Figure 4 demonstrates the Style Wave-U-Net design. The Style Wave-U-Net takes reference audio and original audio as input and produces pre-transformed watermark audio. In the reference audio track, we extract its style embedding by using a style encoder. The style encoder, originally designed for text-to-speech synthesis, extracts the style of a target audio and applies it to text to generate speech with the desired style. In our scenario, we repurpose the style encoder to extract the style from reference audio and apply it to the original audio. For this, we utilize the style encoder from GST-Tacotron2 [15], as described in [61]. On the other track, the original audio is sent to Wave-U-Net [54]. The vanilla Wave-U-Net architecture consisting of both downsampling (DS) and upsampling (US) sub-networks, as depicted by the blue and yellow blocks in Figure 4. In this setup, the input audio undergoes a series of DS blocks, where each deeper DS block extracts increasingly longer feature vectors at lower frequency levels. Notably, a shortcut connects the output of the first convolutional layer in each DS block directly to the final convolutional layer in each US block, facilitating the fusion of features across different frequency levels. In our Style Wave-U-Net, style embeddings are injected at each frequency level of the DS blocks, ensuring the watermarked audio reflects the reference audio's style at all granularities. This design is inspired by V-Cloak [11], but we replace the fully connected layer with a concatenate operation to reduce computation, and we choose random reference audio instead of fixed fingerprints, adding randomness and dynamics.

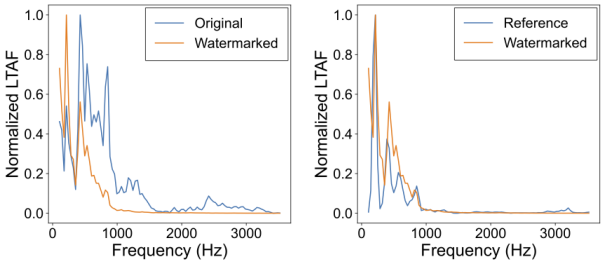
Inversed Contrastive LTAF Loss: The Style Wave-U-Net assures that the watermark is dynamic and imperceptive by style transfer. However, the key challenge is not resolved: "how to generate the hardly-generalized watermark that could always be falsely recognized by the benign model?" To address the challenge, we introduce the Inversed Contrastive LTAF (Long-Time Average Fieldprint) loss to optimize the Style Wave-U-Net, ensuring that the audio features before and after watermarking exhibit distinctly different speaker identities. The LTAF, derived from concepts in CaField [64] and LAS [38], is used to identify speaker identities based on the speech's average energy across various frequencies. Based on the LTAF feature, the same speaker intends to have a similar LTAF value in terms of all frequencies, while different speakers have distinct LTAF features. Utilizing this characteristic, we aim for the watermarked audio to exhibit a distinctly different LTAF feature compared to the original audio. To achieve this, we use an Inversed Contrastive loss to create watermarked audio that significantly differs from its original version. In practice, we feed a pair of reference audio and the original audio into the system. If they are from the same speaker (positive), we optimize the system to increase the distance between the watermarked and original audio's LTAF features. The loss is then calculated as follows:

$$L_{LTAF}^{far} = (|Norm(\mathcal{F}(\hat{x})) - Norm(\mathcal{F}(x))| + \varepsilon)^{-1}, \quad (3)$$

where $\mathcal{F}$ denotes LTAF feature extraction function, $Norm$ is the normalization function, and ε is a margin constant. To minimize the L_{LTAF}^{far}, the distance between the watermarked LTAF and the pre-watermarked LTAF is greater. If the reference audio and original audio are from different speakers, we optimize the watermarked audio close to the reference audio's LTAF, formally:

$$L_{LTAF}^{close} = ||Norm(\mathcal{F}(\hat{x})) - Norm(\mathcal{F}(r))||^2, \quad (4)$$

where r denotes the reference audio. To combine the two losses together, we propose an Inversed Constrative LTAF



(a) Original VS Watermarked (b) Reference VS Watermarked

Figure 5: LTAF feature differences.

loss as follows:

$$L_{LTAF} = \frac{1}{2N}\sum_{i=1}^{N}\left(d_i \cdot L_{LTAF}^{far} + (1-d_i)\cdot \max(0, L_{LTAF}^{close})\right). \tag{5}$$

Here, d_i is a binary label where $d_i = 1$ indicates that the reference audio and original audio are from the same speaker, and $d_i = 0$ denotes that they are from different speakers. The Inversed Contrastive Loss ensure that audio from the same speaker is spaced further apart. Figure 5 shows the preliminary experiment of minimizing the Inversed Contrastive LTAF Loss. The lines indicate the LTAF value across different frequencies. On the left, we show that when the reference audio and the original audio are from the same speaker, we watermark the original audio, causing its LTAF value to deviate from its original appearance. On the right, we note that if the reference audio is from a different speaker, the watermarked original audio is altered to resemble the LTAF of the reference audio more closely.

Using the Style Wave-U-Net and Inversed Contrastive LTAF Loss, we can create dynamic, hardly-generalized audio watermarks. However, in the process of optimizing the LTAF feature, we find that the audio quality is compromised as the frequency energies are forced to deviate from their original state. To address this issue, we introduce our audio quality preservation design below.

4.2.3 Audio Quality Preservation Design

To preserve audio quality throughout the watermarking process, we introduce three essential strategies: 1) a frequency equalizer; 2) a semantic regulation; 3) and a waveform-level regulation.

Frequency Equalizer: The frequency equalizer is designed to align the sound style of the watermarked audio with the reference audio and ensure a smooth transition in LTAF changes. To accomplish this, we employ a style transfer model, DeepAFX-ST [53], which functions as a frequency equalizer. As illustrated in the purple blocks of Figure 4, the model consists of two shared-weight encoders (Fx encoder) that analyze both the input and a style reference signal. These encoders compare their outputs with a controller that determines the parameters for style manipulation. The range of audio effects includes parametric frequency equalizers (PEQ), dynamic range compressors (DRC), infinite impulse response (IIR) filters, reverberation, echo cancellers, among others. In our setup, we input both the Pre-Transformed watermark audio and the reference audio into the frequency equalizer. This process allows the Pre-Transformed watermark audio to adopt the audio effects of the reference audio, facilitating a natural style transition while maintaining audio quality. The output from the frequency equalizer is the Transformed watermark audio. We further apply a style loss to ensure this audio closely matches the style of the reference audio. The formulation of the style loss is present as follows:

$$L_{style} = ||\mathcal{E}(\hat{x}) - \mathcal{E}(r)||^2, \tag{6}$$

where $\mathcal{E}$ denotes the style encoder, we compute the MSE distance between the watermarked audio style and the reference style. Therefore, we can ensure that the watermarked audio not only has similar audio effects with the reference audio but also has a similar style embedding that is justified by the style extractor.

Semantic Regulation: The previous design focuses on enhancing audio perceptual quality and facilitating style transfer. Besides these aspects, it is essential to address the risk of overfitting. Specifically, we aim to produce watermarked audio that maintains semantic integrity, preventing significant distortion in terms of speaker recognition. To this end, it is crucial to control the shift in the conditional distribution from the source audio to the watermarked audio distribution, thereby avoiding the creation of semantically unrelated audio. To manage this, we minimize the class-conditional maximum mean discrepancy (MMD) in the latent space, which is calculated as follows:

$$L_{mmd} = \frac{1}{K}\sum_{k=1}^{K}(||\frac{1}{n_k}\sum_{i=1}^{n_k}\phi(x_i^k) - \frac{1}{n_k}\sum_{i=1}^{n_k}\phi(\hat{x}_i^k)||^2), \tag{7}$$

where K is the total number of speakers and n_k indicates the number of utterances of speaker k. In this equation, x_i^k represents the i_{th} audio spoken from speaker k, and $\hat{x}_i^k$ denotes the watermarked version of the audio. We use a kernel function ϕ to extract the semantic information. For each speaker (k), the kernel function extracts the semantic representations of all his/her speeches. By optimizing the L_{mmd}, the semantic gap between the pre- and post-watermark speech is minimized.

Waveform Regulation: Waveform regulation helps to minimize energy distortion in the watermark by controlling wave changes. We employ Mean Absolute Error (MAE) loss to restrict the changes in the waveform samples, ensuring that the watermarked audio remains consistent in terms of wave strength.

$$L_{wave} = |\hat{x} - x|. \tag{8}$$

In summary, we use the following loss to optimize the Style Wave-U-Net, which is composed of the Inversed Contrastive LTAF loss, style loss, MMD loss, and waveform loss.

$$L_{total} = L_{LTAF} + \alpha * L_{style} + \beta L_{mmd} + \gamma L_{wave}. \quad (9)$$

The α, β, and γ represent the weights of each loss.

4.2.4 Transferable Design

The previous loss function ensures that the watermarked audio has a similar style to the reference audio, and has a distinct speaker identity feature compared to its original version. However, this might not work in the black-box setting when the dataset protector does not know which model will use their dataset. Therefore, we use multiple surrogate models to mimic the behavior of the target model. To improve the adaptability of our approach across various target models, we adopt a *Bi-level Adversarial Optimization strategy*. This strategy involves training the watermark generator through an adversarial strategy. On the one hand, we refine the watermark generative model to produce highly deceptive watermarks; meanwhile, we optimize the surrogate models to make them correctly recognize the watermark samples. Note that the generator (Style Wave-U-Net) and the discriminator (surrogate models) are optimized using distinct loss functions, each improving through competitive interaction. As illustrated in the *Adversarial Target Fine-tuning* part, we integrate multiple surrogate models and average their outputs to enhance decision-making accuracy. This integration not only increases the transferability of the approach across different model architectures but also ensures that the surrogate models are robust enough to accurately identify watermarked samples. Consequently, we have the class loss $L_{class} = -\sum_{i=1}^{N} y_i \log(f(\hat{x}_i))$. By refining the surrogate models with L_{class}, these models can accurately identify the watermarked audio $\hat{x}$, resulting in enhanced robustness. Concurrently, the generator is fine-tuned using L_{total} as described in Eq. 9, enabling it to generate more deceptive watermarks. These two loss functions are optimized in a bi-level manner, which bolsters the watermark's transferability. *We posit that if the watermark can mislead a generalized robust model, it will likely be effective against numerous other benign models as well.*

4.3 Watermarked Dataset Design

Once the watermark generator is well-trained, the dataset protector will generate watermark samples and insert them into the dataset. The proportion of the watermarked sample is referred to as the poison rate, which can be formulated as $\mathcal{D}_m/\mathcal{D}_p$, where $\mathcal{D}_m$ is the watermarked samples and $\mathcal{D}_p$ is the poisoned dataset. While crafting the watermarked dataset, the dataset owner randomly selects original audio and reference audio from random speakers to generate watermark audio. Note that the the watermark is not applied to a specific speaker, but to different samples, so the watermarked dataset will contain both the watermarked version and the clean version of the same speaker. It is worth noting that all the watermark patterns are different even for the input pair.

4.4 Ownership Verification

Ownership verification assesses whether a suspicious DNN model has used a watermarked dataset. In our approach, we hypothesize that a model trained using the watermarked dataset will more accurately identify the ground truth label of the watermarked audio. To test this, we input both benign and watermarked data into the suspicious DNN and observe the probabilities assigned to the ground truth label. Specifically, as shown in Figure 3, given a model, we define P_b as the model's probability on the ground truth label for benign data, and P_w for watermarked data. We establish the null hypothesis $H_0 : P_b = P_w + \tau$ and the alternative hypothesis $H_1 : P_b < P_w + \tau$, where $\tau \in [0, 1]$. We claim that the suspicious model is trained on the watermarked dataset if and only if H_0 is rejected, indicating the P_w is achieving comparable accuracy with P_b. In practice, we set $\tau = 0.25$. It is worth noting that benign models are trained on clean datasets, including clean samples from Alice, while watermarked models are trained on datasets that include watermarked samples of Alice.

5 Evaluation

5.1 Settings

We consider two public datasets to conduct our experiment. The first dataset LibreSpeech [42] released by OpenSLR, we choose the medium-size dataset, which has 23G audios, and covers 363.6 hours of audio data spoken by 921 speakers. The second dataset is VoxCeleb [40], which contains 100,000 utterances from 1,251 celebrities.

Target Models: We choose 10 speaker recognition models to serve as target models. They are VGG-M [9], ResNet-50 [21], ResNet-18 [21], X-vector [51], LSTM [60], ETDNN [50], DTDNN [65], AERT [68], ECAPA-TDNN [12] and FTDNN [44]. Some of the models use feed-forward architecture and use fixed-size data as input (e.g., VGG-M, ResNet-50, ResNet-18), while others use sequential architecture. They may include more advanced technology such as attention layer and transformers, and consider the temporal feature of conjunctive frames to determine the speech identity (e.g., ECAPA-TDNN).

Surrogate Models: As the dataset owner has zero knowledge of the potential target model, they build a general surrogate model based on common knowledge. Specifically, we fuse ResNet-18, VGG-M, AERT, and ETDNN into a generalized model. Once the watermark generator is trained based on the generalized model, we generate watermark samples to apply

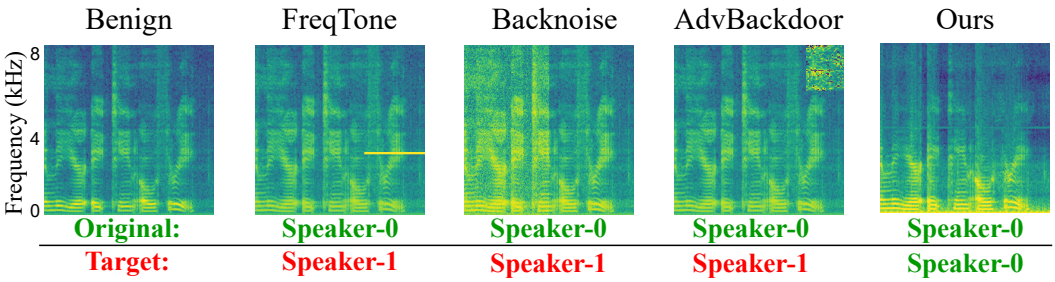


Figure 6: Comparison with voice dataset protections. While various watermarking strategies exist and are different, they all modify labels during the watermarking process, resulting in harmful watermarks. Additionally, many approaches introduce visible changes to the spectrogram, making them easily detectable.

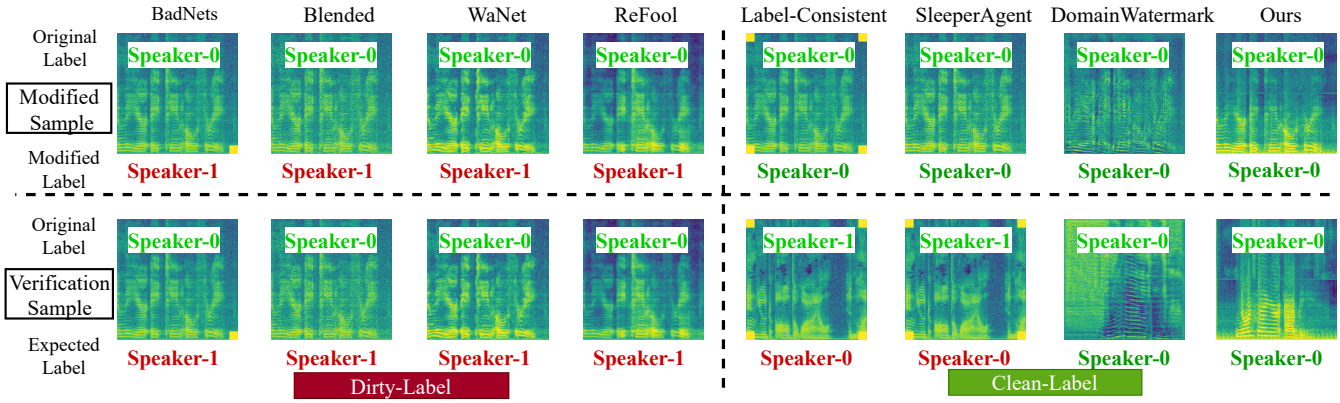


Figure 7: Comparison with image verification methods. On the left, we reproduced four dirty-label methods: BadNets [16], Blended [30], WaNet [41], and ReFool [35], which modify labels and require exact watermark patterns for verification. On the right, we reproduced three clean-label methods: Label-Consistent [55], SleeperAgent [52], and Domain Watermark [19]. While Domain Watermark is harmless, it suffers from poor audio quality. In contrast, AUDIO WATERMARK is harmless, invisible, and dynamic, as it embeds watermarks via style transfer rather than static patterns, allowing verification samples to vary while maintaining reliable ownership verification.

to all the target models. The detailed hyper-parameters can be found in Appendix A.

Evaluation Metrics: To evaluate the performance of our watermark, we use the following metrics. First, we use **Benign Accuracy (BA)** as the model accuracy on the benign testing set. The higher BA indicates the model has better performance on normal usage. Second, we use the **Verification Success Rate (VSR)** to check the ownership verification performance.

In practice, given a watermarked model, we feed 100 pairs of clean and watermarked data into the model. We then perform a pairwise T-test using the 100 pairs of P_b and P_w. If the null hypothesis $H_0 : P_b = P_w + \tau$ ($\tau = 0.25$) is rejected ($p < 0.01$), it indicates that the watermarked model achieves comparable or higher accuracy on watermarked inputs (P_w) compared to clean inputs (P_b), we count it as a success verification. By repeating it 1,000 times with different data subsets, we calculate the average Verification Success Rate (VSR). We also evaluate the Harmful Degree (H defined in Eq. 1) of each protection approach, the the lower harmful degree represents the less risky of the approach. To measure the audio quality, we use Mel Cepstral Distortion (MCD) to check the distortion of the watermarked audio. The lower MCD indicates better audio quality.

5.2 Benchmark Result

Compare with Voice Dataset Protection: We reproduced three backdoor-based voice dataset protections. FreqTone [66] uses a fixed tone as a trigger, Backnoise [34] applies noise as a watermark, and AdvBackdoor [48] optimizes a pattern to inject watermark. We discard the UltraSound [22] because it requires injecting a watermark to the ultrasound range, which is not a typical setting in the audio dataset. Figure 6 demonstrates the watermark pattern. On the left, a benign audio from Speaker-0 is present, followed by the watermarked version of each approach. As can be observed in the figure, the FreqTone introduces a fixed tone at a specific frequency; the Backnoise applies a white-noise pattern to serve as watermark; The AdvBackdoor uses a patch as a watermark. All of them convert the ground truth label from Speaker-0

	Protection Method	LibriSpeech				VoxCeleb			
		BA (%)	VSR (%)	Harmful Degree	MCD (dB)	BA (%)	VSR (%)	Harmful Degree	MCD (dB)
Dirty-Label	FreqTone	98.5	100	0.98	8.1	92.5	100	0.99	7.5
	Backnoise	95.4	87.2	0.84	7.8	91.2	89.3	0.86	7.7
	AdvBackdoor	88.1	100	0.99	13.4	85.2	98.2	0.94	12.1
	BadNets	80.9	100	0.95	6.5	85.1	100	0.99	6.4
	Blended	90.4	61.8	0.74	7.2	91.2	64.2	0.75	6.8
	WaNet	91.6	19.4	0.24	7.5	90.8	24.3	0.29	6.6
	ReFool	80.6	73.5	0.79	7.9	82.1	74.6	0.82	8.2
Clean-Label	Label-Consistent	90.5	95.2	0.77	12.2	91.5	75.2	0.84	12.9
	Sleeper Agent	88.7	71.2	0.82	6.0	83.4	69.4	0.81	6.8
	Domain Watermark	12.6	78.4	0.05	15.9	15.5	85.1	0.04	14.5
	Audio Watermark	96.4	95.5	0.06	9.6	97.6	94.5	0.03	9.2

Table 2: Benchmark comparison of AUDIO WATERMARK with existing watermarking. Compared to all existing work, we are the only watermark approach to achieve high BA, high VSR, a minimal Harmful Degree, while maintaining comparable audio quality (low MCD).

to Speaker-1, indicating the dirty-label approach. For comparison, AUDIO WATERMARK is invisible and imperceptive, meanwhile, the label of the watermarked audio is not changed. A closer look of our watermark is in Appendix B.

Compare with Image Dataset Protection: Since there are a limited number of voice dataset protection approaches, we also compare our watermark with seven image-based watermark approaches. Different from the voice watermark that takes waveform as input, we use a spectrogram as image input. Figure 7 demonstrates the image-based approaches. On the left, we reproduce BadNets [16], Blended [30], WaNet [41], ReFool [35]. The first row indicates the modified sample in the process of crafting the watermark dataset, and the second row represents the verification sample during the verification stage. From left to right, the BadNets inject a square as a watermark; the Blended mixes a random spectrogram (white noise) as a watermark; the WaNet introduces a wrapping operation to the spectrogram as a watermark, causing the formant to vibrate; the ReFool use the reflection as watermark, enhancing the watermark's invisibility. All of them are dirty-label watermark as they change the original label from Speaker-0 to modified label Speaker-1, and the verification sample has to *contain the exact same watermark pattern* during the verification stage. For comparison, we also reproduced three clean-label image watermarks. On the right side of Figure 7, the label-consistent [55] injects an obvious watermark, but keeps the modified label as same as the original label. However, during the verification, the verification sample (originated from Speaker-1) with the watermark is expected to be recognized as Speaker-0, causing a mismatch between the target label (Speaker-0) and ground truth label (Speaker-1); SleeperAgent [52] injects invisible watermark during prepare the dataset but verifies the ownership with an obvious trigger. Although it does not change the label during the dataset preparation stage, it leaves a harmful backdoor as the watermarked model predicts incorrectly for watermarked samples (predict Speaker-1 with watermark to Speaker-0). The only work that adopts a harmless watermark is Domain Watermark [19]. However, it suffers from low audio quality, and in turn, affects the watermarked model's benign accuracy. In contrast, our watermark is invisible and imperceptive. It is harmless because the modified label and expected label are always aligned with the original label. Moreover, although each watermark is different, they can be used to verify the suspicious model used on another watermark, as they share the same hardly generalized domain.

Benchmarking Comparision: To thoroughly assess the performance of the ten existing studies along with our watermark, we conduct a benchmark using the default settings specified in each work. For each method, we test on two datasets with 10 speakers for speaker recognition tasks. We assume ResNet-18 as the attacker's base model. We create a watermark dataset with a 15% poisoning rate, and then utilize a separate clean verification set to evaluate the Benign Accuracy (BA). We employ both a watermarked and a clean set to assess the Verification Success Rate (VSR) and Harmful Degree. Additionally, we determine the Mel Cepstral Distortion (MCD) by comparing the watermarked audio to its original version.

The result is presented in Table 2. We observe that existing audio watermark methods, such as FreqTone [66] Backnoise [34], and AdvBackdoor [48], perform well in terms of VSR and BA, but they exhibit high harmful degrees and significant distortion (notably AdvBackdoor). Additionally, they are vulnerable to being detected due to the alteration of data labels and the predefined nature of the watermark patterns. For image-based dirty-label backdoor watermarks like BadNets [16], Blended [30], WaNet [41], and ReFool [35], we see high BA but unstable VSR, reflecting inconsistent performance in verifying ownership in the speech domain. All except WaNet exhibit high harmful degrees; WaNet's poison is less effective, with a VSR of only 19.4%. In the clean-label watermark category, the Label-consistent [55] approach suffers from high distortion and harmful degree, as does the sleeper agent [52]. The notable exception is Domain Watermark [19], which, although exhibiting low harmful degrees, causes considerable distortion, impacting the benign accuracy of the watermarked model. This observation is consistent with findings presented in Section 4.2.2. In contrast, our watermark

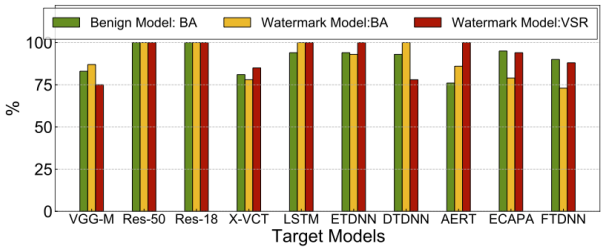
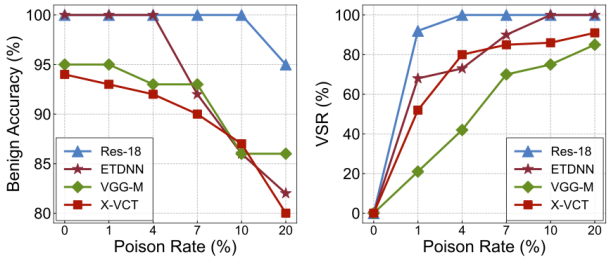


Figure 8: Watermark transferability on different models.



(a) Poison Rate VS BA. (b) Poison Rate VS VSR.

Figure 9: Impact of poison rate.

is uniquely suited for audio datasets, offering high BA, VSR, and low harmful degrees with moderate Mel Cepstral Distortion (MCD). Additionally, our watermark demonstrates robust resistance to various data-level and model-level backdoor detection methods, which will be further discussed in Section 5.5.

5.3 Transferbility of AUDIO WATERMARK

Dataset Transferbility: The goal of dataset transferability is to determine whether our watermark can be adapted to various datasets. To assess this, we initially trained our watermark generator using the LibriSpeech dataset. We then employ this trained generator to create watermark audio on the VoxCeleb dataset with a poisoning rate of 15%. Assuming an attacker utilizes a ResNet-18 model to fine-tune this watermarked VoxCeleb dataset, we verify dataset usage by feeding it benign and watermarked VoxCeleb data. The results of this experiment demonstrate that the watermarked ResNet-18 model achieves a BA of 95.1%, a VSR of 98.5%, a harmful degree of 0.03, and a MCD of 9.4dB. These metrics confirm that our watermark possesses dataset transferability. Dataset protectors can readily download the well-trained watermark generator and apply it to generate watermark audio on their own datasets.

Model Transferability: The model transferability determines the capability of AUDIO WATERMARK on different attacker's models. In the previous experiment, we assume the attacker uses a specific speaker recognition model, however, in our threat model, the attacker's model is unknown. Therefore, we experiment with ten speaker recognition models. First, we train them with the clean dataset to correctly recognize the speakers; second, we assume the attacker trains each model on the watermarked dataset; last, we verify the watermark performance of BA and VSR on each model. Significantly, in our Bi-level Adversarial Optimization Strategy, the watermark generator is refined using multiple surrogate models. In Figure 8, most of the speaker recognition models are well-trained to make approximate 80% accuracy. Once the model is fine-tuned on the watermarked dataset, the BAs are improved (yellow bar). Meanwhile, the VSR for all models maintains high, the worst cases are VGG-M, X-VCT, and DTDNN, resulting in around 75% success rate.

5.4 Ablation Study

Poison Rate: The ablation study focuses on finding the critical impact factors on our watermark. We hereby evaluate the Poison Rate (%) that may affect the watermark. The poison rate refers to the proportion of watermarked samples over the complete watermarked dataset. In the dataset protection pipeline, the defender only injects a small portion of the watermarked sample into the watermarked dataset. Typically, a lower poison rate will lead to a worse protection success rate. If a protection approach can still succeed with a small poison rate, that means the protection is very powerful. To evaluate, we craft multiple watermarked datasets with different poison rates, and then train models with the different watermarked datasets. We check the BA and VSR for each watermarked model and present the result in Figure 9. From the left side of the figure, we find the Benign accuracy is merely affected by increasing poison rate across four different speaker recognition models; From the right, we can see the VSR is indeed affected by the poison rate, Our watermark exhibits varied effectiveness when the poison rate is below 7%. Notably, some models achieve a high VSR (ResNet-18, ETDNN, X-VECTOR) with just a 1% poison rate, while VGG-M shows the weakest performance at low poison rates (VSR=20% when pr=1%). However, once the poison rate reaches 10%, all models with the watermark demonstrate high VSR, confirming that our watermark can be effective even at low poison rates and is likely to succeed at poison rates exceeding 10% for most models.

Noise As Watermark: In this experiment, we investigate if noises can serve as a watermark. The basic assumption of using noise as a watermark is that while a normal model cannot correctly recognize noisy data, a model trained on a dataset containing noise can better identify speeches. We download six noise sources (babble, factory, Volvo, leopard, gun) from NOISEX92 [57] and apply the noise to the LibriSpeech dataset with 0dB. We check the performance on 10 speaker recognition models and label it a success if VSR is greater than 70%. We found that all the noises can only succeed on 3 to 4 models, compared to our watermark which succeeds on all 10 models, using noise as watermarks is not

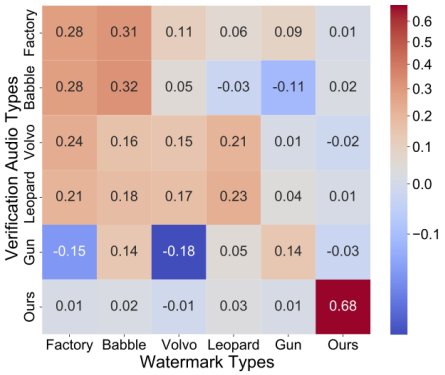


Figure 10: Cross verification result.

sufficient to protect various target models. Moreover, one critical benefit of our watermark is hardly generalized from another dataset, which can benefit for reducing the false positive rate during the verification. To completely compare our watermark with the noise-based watermark, we conduct a comprehensive experiment that assume the dataset protector watermark by one noise and verified it with another noise. *We feed 1,000 watermarked verification audio samples into both the benign model and the watermarked model, calculating the accuracy P_v^w for the watermarked model and P_v^b for the benign model on the same verification audio. By examining the difference $P_v^w - P_v^b$, we expect the watermarked model, trained with the same type of watermark, to demonstrate higher accuracy.*

Figure 10 shows that all noises have positive outcomes when used for self-watermarking and verification. However, our watermark stands out with the most significant value compared to others, indicating a high verification success rate (VSR) across multiple models. Additionally, we observe that using noise as a watermark can generalize across different types of noise. For instance, a watermark with factory noise can still verify the usage with a babble noise sample (+0.28). This finding suggests that all noises, except ours, are not hardly-generalized and may lead to false positives during verification.

5.5 Robustness of AUDIO WATERMARK

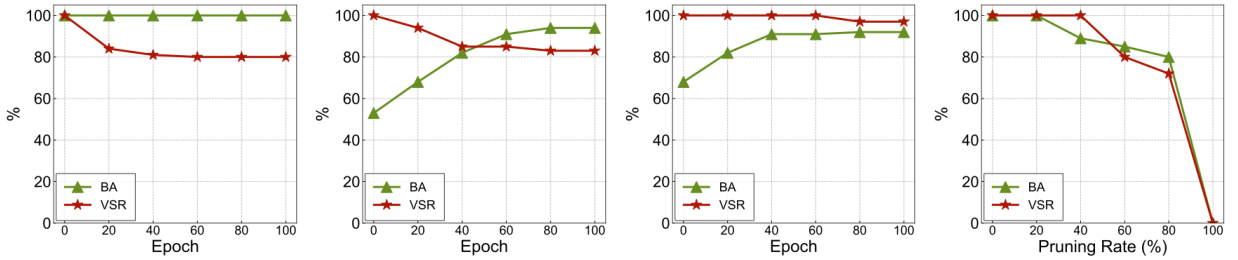
Model-level Attack: In the model-level attack, the attacker aims to remove the watermark of their model or detect whether their model was trained on the watermarked dataset. To evaluate this, we assume the attacker uses three attacks: model fine-tuning [36]; model pruning [62]; and Neural Cleanse [59]:

1. Model Fine-tuning: Suppose the model is trained on a watermarked LibriSpeech dataset, now the attacker wants to remove the model's watermark by fine-tuning the watermarked model on three clean datasets: LibriSpeech, VoxCeleb, and TIMIT. We present the result in Figure 11. In the figures, the BA is consistently high when fine-tuning on the clean LibriSpeech dataset because the model was initially trained on the watermarked LibriSpeech dataset, making the data distributions similar. In contrast, the BA is lower when fine-tuning on VoxCeleb and TIMIT because these datasets differ significantly from LibriSpeech, leading to lower generalization performance. Interestingly, the VSR remains unaffected across all three scenarios. This is because the watermark leverages Out-of-Domain (OOD) features, which are not influenced by fine-tuning with clean data from other datasets. Since the OOD features used to craft the watermark are unrelated to the primary training data, they are not learned or corrected during fine-tuning on clean datasets. As a result, the watermark's verification success remains robust, demonstrating the effectiveness of our watermarking approach.

2. Model Pruning: The attacker may remove the model's watermark by pruning. In Figure 11(d), we find the watermarked model can barely be impacted by pruning. The VSR only decreases dramatically if the pruning rate is over 80%, but meanwhile, the model's benign performance is also ruined, indicating the aggressive pruning cannot retain the normal usage of the model.

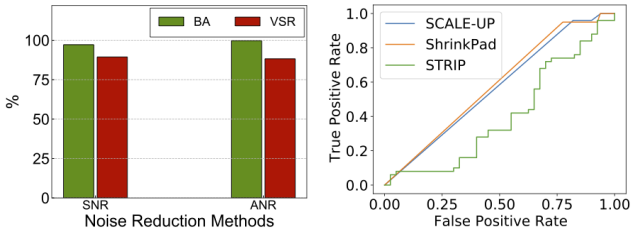
3. Neural Cleanse: In Neural Cleanse, the attacker aims to calculate the anomaly index to identify the watermark class. However, when we run the defense algorithm, we found the maximum anomaly index is $1.46 < 2$, indicating the watermarked model is not detected. The predicted watermark pattern is present in Appendix C.

Data-level Attack: The Data-level attack indicates the adversaries clean the watermarked dataset, or alter the watermark samples. More specifically, we assume the attacker can use noise reduction approaches (Stationary and Adaptive) to clean the dataset; and can alter the dataset with advanced approaches such as STRIP [13], ShrinkPad [26], and Scale-Up [18]. While the noise reduction approaches modify the watermarking sample before using the dataset, the advanced approaches aim to observe the abnormal behavior of a watermarked model by altering the input sample. For example, the STRIP [13] blends a perturbation to a watermarked sample and checks the output of the model to determine whether the model is watermarked or not; the ShrinkPad [26] uses shrinking and padding to change the watermarked sample to invalidate watermark; the Scale-Up amplify the watermarked sample and detect the watermarked model by observing the prediction consistency. We present the result in Figure 12. On the left, it is observed that the noise reduction method does not impact the efficacy of the watermark. Even after the dataset has been denoised, when an attacker trains their own ResNet with it, the dataset still exhibits high BA and VSR. On the right side, we consider a scenario where the attacker trains their model using the watermarked dataset and then modifies the watermark samples using suggested techniques. The results show that the Area Under the Receiver Operating Characteristic (AUROC) for each attack is approximately 0.5, similar to making a random guess. This implies that none of these attacks can effectively determine the presence of the dataset's watermark. More experiments for watermarks in the physical world can be found in Appendix D an E.



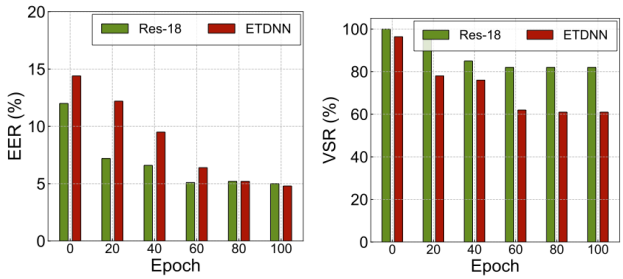
(a) Fine-Tuning on LibriSpeech (b) Fine-Tuning on VoxCeleb (c) Fine-Tuning on TIMIT (d) The resistance to model pruning

Figure 11: Resistance to model-level attack



(a) Noise Reduction Defense (b) Advanced Defense

Figure 12: Resistance to data-level attack.



(a) Benign Input EER (b) Watermark Verification

Figure 13: Resistance to Downstream Task Transfer.

Task Transfer Attack: In this experiment, we consider a scenario where the attacker uses the watermarked model as a pre-trained model and fine-tunes it on a different downstream task, specifically Speaker Verification (SV). We evaluate the model's performance on benign inputs using the Equal Error Rate (EER), where a lower EER indicates better performance. To assess watermark verification, we calculate the Verification Success Rate (VSR) by feeding watermarked speech from a same legitimate speaker into the model and checking if the similarity score exceeds the EER threshold. We expect the fine-tuned watermarked model to correctly classify both watermarked and benign samples from the same speaker.

We use two watermarked models, ResNet-18 and ETDNN as pre-trained models and finetune them for the speaker verification task. During fine-tuning, the last layer of each model is removed, and the optimization objective is changed to the GE2E loss [58]. The results are shown in Figure 13. We observe that both models initially exhibit high EERs when adapted to the speaker verification task. However, ResNet-18 converges faster, with a stable endpoint at approximately 5% EER. On the right, we see that the VSR remains above 80% for the ResNet-18 model, while for the ETDNN model, the VSR drops to around 60%. This indicates that transferring the model to a downstream task can impact the watermark's performance to some extent. It is also important to note that most existing watermarking approaches focus on single-task settings. Although some watermarking methods claim multi-task applicability, they still require that the watermarking mechanism, the watermarked model, and the verification process function within the same downstream task, for instance, in image classification [19, 24], speech command classification [22, 48], and speaker recognition [17, 48]. Extending one watermarking approach to support multi-task settings presents significant challenges and represents an important area for future research.

Adaptive Attack: For adaptive attackers who know our watermarking scheme, they can generate watermark to enforce the watermarked model unable to correctly recognize the original label when the dataset owner verify with watermark sample (result in $P_b > P_w$), and the attacker can claim he/she never used the dataset. Although this assumption is overly strong due to the adaptive attackers need to obtain the watermark generator model and retrain it with completely different loss design, our watermarking approach remains effective for three key reasons: First, the attackers do not know which speaker we used for watermarking, so they cannot overwrite our watermark. Second, they do not know our reference samples, making it impossible for them to replicate the exact same watermark. Third, if the attackers designs such a watermark that make models trained on their dataset has low P_w, it compromises the robustness of their model. Such a model would fail to correctly recognize watermark samples and would introduce harmful backdoors. These backdoors would make their model vulnerable to exploitation and undermine its security and reliability. As a result, our watermarking approach is robust and effective, even against adaptive attackers.

6 Conclusion

We propose AUDIO WATERMARK, a harmless audio watermark designed to protect the copyright of voice datasets. We demonstrate that our watermark achieves a high verification success rate, low harmful degree, and minimal distortion.

7 Acknowledgment

We would like to extend our appreciation to our shepherd and the anonymous reviewers for their invaluable input on our study. This work was supported in part by the U.S. NSF grants CNS-2310207, CNS-2226888, CNS-2235231. Junfeng Guo and Heng Huang is supported by NSF IIS 2347592, 2347604, 2348159, 2348169, DBI 2405416, CCF 2348306, and CNS 2347617.

8 Ethics considerations

Our approach promotes the copyright protection of speech datasets, contributes to positive outcomes, and avoids harm, aligning with ethical principles of fairness, beneficence, and justice.

9 Open science

All experiments and evaluations are conducted on open-source datasets. The artifact is available on https://zenodo.org/records/14738544.

References

[1] Vishal Asnani, John Collomosse, Tu Bui, Xiaoming Liu, and Shruti Agarwal. Promark: Proactive diffusion watermarking for causal attribution. *arXiv preprint arXiv:2403.09914*, 2024.

[2] Vishal Asnani, Abhinav Kumar, Suya You, and Xiaoming Liu. Probed: Proactive object detection wrapper. *Advances in Neural Information Processing Systems*, 36, 2024.

[3] Vishal Asnani, Xi Yin, Tal Hassner, Sijia Liu, and Xiaoming Liu. Proactive image manipulation detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15386–15395, 2022.

[4] Dan Boneh and Matt Franklin. Identity-based encryption from the weil pairing. In *Annual international cryptology conference*, pages 213–229. Springer, 2001.

[5] Zhipeng Cai, Zuobin Xiong, Honghui Xu, Peng Wang, Wei Li, and Yi Pan. Generative adversarial networks: A survey toward private and secure applications. *ACM Computing Surveys (CSUR)*, 54(6):1–38, 2021.

[6] Guangke Chen, Yedi Zhang, and Fu Song. Slmia-sr: Speaker-level membership inference attacks against speaker recognition systems. *arXiv preprint arXiv:2309.07983*, 2023.

[7] Mingxiang Chen, Zhanguo Chang, Haonan Lu, Bitao Yang, Zhuang Li, Liufang Guo, and Zhecheng Wang. Augnet: End-to-end unsupervised visual representation learning with image augmentation. *arXiv preprint arXiv:2106.06250*, 2021.

[8] Christopher A Choquette-Choo, Florian Tramer, Nicholas Carlini, and Nicolas Papernot. Label-only membership inference attacks. In *International conference on machine learning*, pages 1964–1974. PMLR, 2021.

[9] Joon Son Chung, Arsha Nagrani, and Andrew Zisserman. Voxceleb2: Deep speaker recognition. *arXiv preprint arXiv:1806.05622*, 2018.

[10] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[11] Jiangyi Deng, Fei Teng, Yanjiao Chen, Xiaofu Chen, Zhaohui Wang, and Wenyuan Xu. {V-Cloak}: Intelligibility-, naturalness-& {Timbre-Preserving}{Real-Time} voice anonymization. In *32nd USENIX Security Symposium (USENIX Security 23)*, pages 5181–5198, 2023.

[12] Brecht Desplanques, Jenthe Thienpondt, and Kris Demuynck. Ecapa-tdnn: Emphasized channel attention, propagation and aggregation in tdnn based speaker verification. *arXiv preprint arXiv:2005.07143*, 2020.

[13] Yansong Gao, Change Xu, Derui Wang, Shiping Chen, Damith C Ranasinghe, and Surya Nepal. Strip: A defence against trojan attacks on deep neural networks. In *Proceedings of the 35th annual computer security applications conference*, pages 113–125, 2019.

[14] John S Garofolo. Timit acoustic phonetic continuous speech corpus. *Linguistic Data Consortium, 1993*, 1993.

[15] GST-Tacotron. https://nvidia.github.io/OpenSeq2Seq/html/speech-synthesis/tacotron-2-gst.html, n.d. Accessed: [Date of Access].

[16] Tianyu Gu, Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Evaluating backdooring attacks on deep neural networks. *IEEE Access*, 7:47230–47244, 2019.

[17] Hanqing Guo, Xun Chen, Junfeng Guo, Li Xiao, and Qiben Yan. Masterkey: Practical backdoor attack against speaker verification systems. In *Proceedings of the 29th Annual International Conference on Mobile Computing and Networking*, pages 1–15, 2023.

[18] Junfeng Guo, Yiming Li, Xun Chen, Hanqing Guo, Lichao Sun, and Cong Liu. Scale-up: An efficient

black-box input-level backdoor detection via analyzing scaled prediction consistency. *arXiv preprint arXiv:2302.03251*, 2023.

[19] Junfeng Guo, Yiming Li, Lixu Wang, Shu-Tao Xia, Heng Huang, Cong Liu, and Bo Li. Domain watermark: Effective and harmless dataset copyright protection is closed at hand. *Advances in Neural Information Processing Systems*, 36, 2024.

[20] Yuanfang Guo, Oscar C Au, Rui Wang, Lu Fang, and Xiaochun Cao. Halftone image watermarking by content aware double-sided embedding error diffusion. *IEEE Transactions on Image Processing*, 27(7):3387–3402, 2018.

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[22] Stefanos Koffas, Jing Xu, Mauro Conti, and Stjepan Picek. Can you hear it? backdoor attacks via ultrasonic triggers. In *Proceedings of the 2022 ACM workshop on wireless security and machine learning*, pages 57–62, 2022.

[23] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[24] Yiming Li, Yang Bai, Yong Jiang, Yong Yang, Shu-Tao Xia, and Bo Li. Untargeted backdoor watermark: Towards harmless and stealthy dataset copyright protection. *Advances in Neural Information Processing Systems*, 35:13238–13250, 2022.

[25] Yiming Li, Peidong Liu, Yong Jiang, and Shu-Tao Xia. Visual privacy protection via mapping distortion. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3740–3744. IEEE, 2021.

[26] Yiming Li, Tongqing Zhai, Yong Jiang, Zhifeng Li, and Shu-Tao Xia. Backdoor attack in the physical world. *arXiv preprint arXiv:2104.02361*, 2021.

[27] Yiming Li, Ziqi Zhang, Jiawang Bai, Baoyuan Wu, Yong Jiang, and Shu-Tao Xia. Open-sourced dataset protection via backdoor watermarking. *arXiv preprint arXiv:2010.05821*, 2020.

[28] Yiming Li, Mingyan Zhu, Xue Yang, Yong Jiang, Tao Wei, and Shu-Tao Xia. Black-box dataset ownership verification via backdoor watermarking. *IEEE Transactions on Information Forensics and Security*, 2023.

[29] Yiming Li, Mingyan Zhu, Xue Yang, Yong Jiang, and Shu-Tao Xia. Black-box ownership verification for dataset protection via backdoor watermarking. *arXiv preprint arXiv:2209.06015*, 2022.

[30] Yuezun Li, Yiming Li, Baoyuan Wu, Longkang Li, Ran He, and Siwei Lyu. Invisible backdoor attack with sample-specific triggers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16463–16472, 2021.

[31] Linguistic Data Consortium. Data management: Using: Licensing. https://www.openslr.org/7/. Accessed: [Date of Access].

[32] Linguistic Data Consortium. Data management: Using: Licensing. https://www.ldc.upenn.edu/data-management/using/licensing, n.d. Accessed: [Date of Access].

[33] 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.

[34] Yingqi Liu, Shiqing Ma, Yousra Aafer, Wen-Chuan Lee, Juan Zhai, Weihang Wang, and Xiangyu Zhang. Trojaning attack on neural networks. In *25th Annual Network And Distributed System Security Symposium (NDSS 2018)*. Internet Soc, 2018.

[35] Yunfei Liu, Xingjun Ma, James Bailey, and Feng Lu. Reflection backdoor: A natural backdoor attack on deep neural networks. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part X 16*, pages 182–199. Springer, 2020.

[36] Yuntao Liu, Yang Xie, and Ankur Srivastava. Neural trojans. In *2017 IEEE International Conference on Computer Design (ICCD)*, pages 45–48. IEEE, 2017.

[37] Zhenghui Liu, Yuankun Huang, and Jiwu Huang. Patchwork-based audio watermarking robust against desynchronization and recapturing attacks. *IEEE transactions on information forensics and security*, 14(5):1171–1180, 2018.

[38] Anders Löfqvist and Bengt Mandersson. Long-time average spectrum of speech and voice analysis. *Folia phoniatrica et logopaedica*, 39(5):221–229, 1987.

[39] Paulo Martins, Leonel Sousa, and Artur Mariano. A survey on fully homomorphic encryption: An engineering perspective. *ACM Computing Surveys (CSUR)*, 50(6):1–33, 2017.

[40] Arsha Nagrani, Joon Son Chung, and Andrew Zisserman. Voxceleb: a large-scale speaker identification dataset. *arXiv preprint arXiv:1706.08612*, 2017.

[41] Anh Nguyen and Anh Tran. Wanet–imperceptible warping-based backdoor attack. *arXiv preprint arXiv:2102.10369*, 2021.

[42] 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.

[43] Orestis Papakyriakopoulos and Alice Xiang. Considerations for ethical speech recognition datasets. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*, pages 1287–1288, 2023.

[44] Daniel Povey, Gaofeng Cheng, Yiming Wang, Ke Li, Hainan Xu, Mahsa Yarmohammadi, and Sanjeev Khudanpur. Semi-orthogonal low-rank matrix factorization for deep neural networks. In *Interspeech*, pages 3743–3747, 2018.

[45] Ronald Rivest. The md5 message-digest algorithm. Technical report, 1992.

[46] Anthony Rousseau, Paul Deléglise, and Yannick Esteve. Ted-lium: an automatic speech recognition dedicated corpus. In *LREC*, pages 125–129, 2012.

[47] Sada. SADA. https://www.aliexpress.com/item/4001241222763.html.

[48] Cong Shi, Tianfang Zhang, Zhuohang Li, Huy Phan, Tianming Zhao, Yan Wang, Jian Liu, Bo Yuan, and Yingying Chen. Audio-domain position-independent backdoor attack via unnoticeable triggers. In *Proceedings of the 28th Annual International Conference on Mobile Computing And Networking*, pages 583–595, 2022.

[49] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP)*, pages 3–18. IEEE, 2017.

[50] David Snyder, Daniel Garcia-Romero, Gregory Sell, Alan McCree, Daniel Povey, and Sanjeev Khudanpur. Speaker recognition for multi-speaker conversations using x-vectors. In *ICASSP 2019-2019 IEEE International conference on acoustics, speech and signal processing (ICASSP)*, pages 5796–5800. IEEE, 2019.

[51] David Snyder, Daniel Garcia-Romero, Gregory Sell, Daniel Povey, and Sanjeev Khudanpur. X-vectors: Robust dnn embeddings for speaker recognition. In *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5329–5333. IEEE, 2018.

[52] Hossein Souri, Liam Fowl, Rama Chellappa, Micah Goldblum, and Tom Goldstein. Sleeper agent: Scalable hidden trigger backdoors for neural networks trained from scratch. *Advances in Neural Information Processing Systems*, 35:19165–19178, 2022.

[53] Christian J Steinmetz, Nicholas J Bryan, and Joshua D Reiss. Style transfer of audio effects with differentiable signal processing. *arXiv preprint arXiv:2207.08759*, 2022.

[54] Daniel Stoller, Sebastian Ewert, and Simon Dixon. Wave-u-net: A multi-scale neural network for end-to-end audio source separation. *arXiv preprint arXiv:1806.03185*, 2018.

[55] Alexander Turner, Dimitris Tsipras, and Aleksander Madry. Label-consistent backdoor attacks. *arXiv preprint arXiv:1912.02771*, 2019.

[56] UK GDPR. Chapter 9 - article 89, 2021.

[57] Andrew Varga and Herman JM Steeneken. Assessment for automatic speech recognition: Ii. noisex-92: A database and an experiment to study the effect of additive noise on speech recognition systems. *Speech communication*, 12(3):247–251, 1993.

[58] Li Wan, Quan Wang, Alan Papir, and Ignacio Lopez Moreno. Generalized end-to-end loss for speaker verification. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4879–4883. IEEE, 2018.

[59] Bolun Wang, Yuanshun Yao, Shawn Shan, Huiying Li, Bimal Viswanath, Haitao Zheng, and Ben Y Zhao. Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 707–723. IEEE, 2019.

[60] Quan Wang, Carlton Downey, Li Wan, Philip Andrew Mansfield, and Ignacio Lopz Moreno. Speaker diarization with lstm. In *2018 IEEE International conference on acoustics, speech and signal processing (ICASSP)*, pages 5239–5243. IEEE, 2018.

[61] Yuxuan Wang, Daisy Stanton, Yu Zhang, RJ-Skerry Ryan, Eric Battenberg, Joel Shor, Ying Xiao, Ye Jia, Fei Ren, and Rif A Saurous. Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis. In *International conference on machine learning*, pages 5180–5189. PMLR, 2018.

[62] Dongxian Wu and Yisen Wang. Adversarial neuron pruning purifies backdoored deep models. *Advances in Neural Information Processing Systems*, 34:16913–16925, 2021.

[63] Zuobin Xiong, Zhipeng Cai, Qilong Han, Arwa Alrawais, and Wei Li. Adgan: Protect your location privacy in camera data of auto-driving vehicles. *IEEE Transactions on Industrial Informatics*, 17(9):6200–6210, 2020.

[64] Chen Yan, Yan Long, Xiaoyu Ji, and Wenyuan Xu. The catcher in the field: A fieldprint based spoofing detection for text-independent speaker verification. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, pages 1215–1229, 2019.

[65] Ya-Qi Yu and Wu-Jun Li. Densely connected time delay neural network for speaker verification. In *INTERSPEECH*, pages 921–925, 2020.

[66] Tongqing Zhai, Yiming Li, Ziqi Zhang, Baoyuan Wu, Yong Jiang, and Shu-Tao Xia. Backdoor attack against speaker verification. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2560–2564. IEEE, 2021.

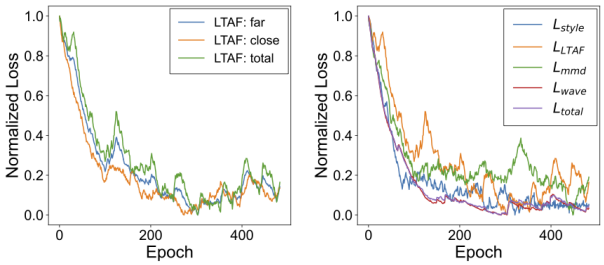
[67] Qiuyu Zhang, Yuzhou Li, Yingjie Hu, and Xuejiao Zhao. An encrypted speech retrieval method based on deep perceptual hashing and cnn-bilstm. *IEEE Access*, 8:148556–148569, 2020.

[68] Ruiteng Zhang, Jianguo Wei, Wenhuan Lu, Longbiao Wang, Meng Liu, Lin Zhang, Jiayu Jin, and Junhai Xu. Aret: Aggregated residual extended time-delay neural networks for speaker verification. In *INTERSPEECH*, pages 946–950, 2020.

[69] Shi-Xiong Zhang, Yifan Gong, and Dong Yu. Encrypted speech recognition using deep polynomial networks. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5691–5695. IEEE, 2019.

Appendix A: Training Detail

We train the watermark generator in the following hyperparameter setting: we use a SGD optimizer with the learning rate=0.001, momentum=0.9, and weight_decay=0.0005. We use a StepLR optimizer to optimize the watermark generator. More specifically, we set $\alpha = 20$, $\beta = 0.2$, and $\gamma = 5$ for L_{total}. To optimize the generalized surrogate model, we use the same SGD optimizer, and choose four common speaker recognition models (ResNet-18, VGG-M, AERT, ETDNN) as the base of the surrogate model. For the style encoder and the Fx encoder, we freeze their model parameter and use the default setting and the default checkpoint provided by their projects. In Figure 14, we present the learning process of each loss in a normalized manner.



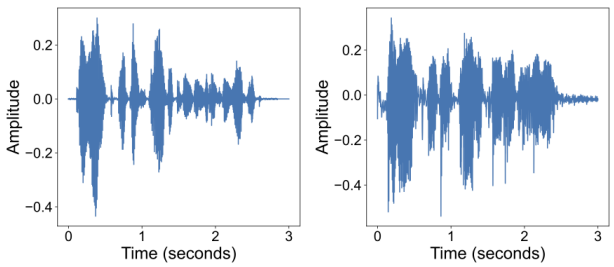
(a) LTAF loss (b) LTAF mmd style wave total

Figure 14: The learning process

We can observe that both $LTAF_{far}$ and $LTAF_{close}$ are minimized, indicating that the watermark can alter the LTAF feature for the same speaker, and close to the reference speaker. On the right, we can find that all losses converge in 500 epochs, suggesting that each design goal is satisfied after training.

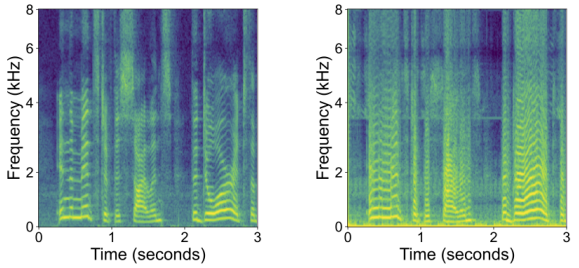
Appendix B: Demo

We conduct a demonstration of our watermark in Figure 15 and Figure 16 for a closer inspection.



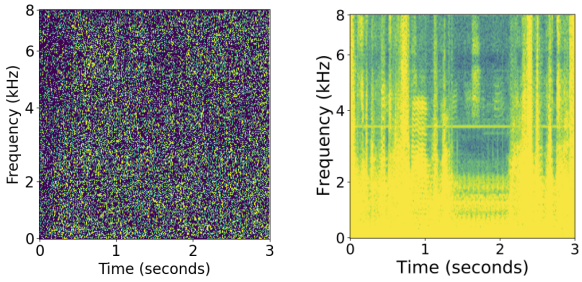
(a) Original waveform (b) Watermarked waveform

Figure 15: Demonstration of waveform



(a) Original spectrogram (b) Watermarked spectrogram

Figure 16: Demonstration of spectrogram



(a) Neural Cleanse (b) SCALE-UP

Figure 17: Attack demonstration.

From the waveform, we can see that the watermarked version introduces some distortion to the clean sample. As for the spectrogram, we can find that the watermarked spectrogram has stronger energy informants, and has extra distortion that is distributed in the high-frequency range (>4kHz). The overall speech content is not significantly changed, and the speech quality is not severely affected.

Appendix C: Additional Results of Robustness

For Nerual Cleanse, we reverse-engineer the watermark pattern and present it to Figure 17(a). We can find the predicted watermark is dense and not similar to our real watermark. In Figure 17(b), we present the amplified watermark sample.

Appendix D: Watermark in the Physical-world

In this section, we evaluate the robustness of our watermark in a physical scenario. In this scenario, the suspicious model only allows physical input. In this case, the watermarked audio cannot directly pass to the suspicious model for ownership verification. This is important because the attacker may embed their model into a physical system, which only uses a microphone to collect input. To validate our watermark in such a scenario, we made the following setup as shown in Figure 18. In this scenario, we assume the attackers embed

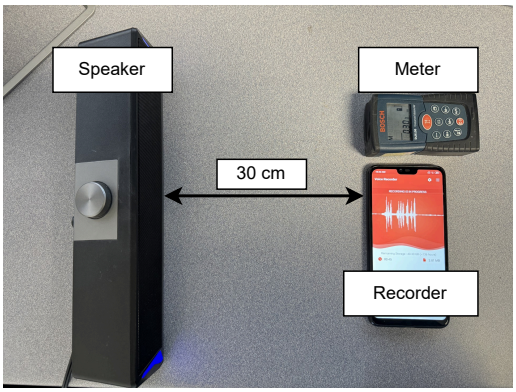


Figure 18: Experiment setting of over-the-air attack

their watermarked model into a system, and this system only accepts real-world input from its microphone. To simulate this scenario, we use a SADA D6 speaker [47] to play the watermarked audio and use a recorder to serve as the microphone in the system. The distance between the speaker and the recorder is 30cm. Once the recorder records the watermarked audio, we feed the recordings to the suspicious model and check the VSR of the model. We repeat the process 10 times and record 10 watermarked audio and 10 clean audio. Then we feed them to the watermarked model and observe the performance. Surprisingly, we receive VSR=90%, which means 9 of 10 pairs of audio can be used for watermark verification in the real world, even though we didn't do any design to adapt to physical distortions.

Appendix E: Understanding the Watermark

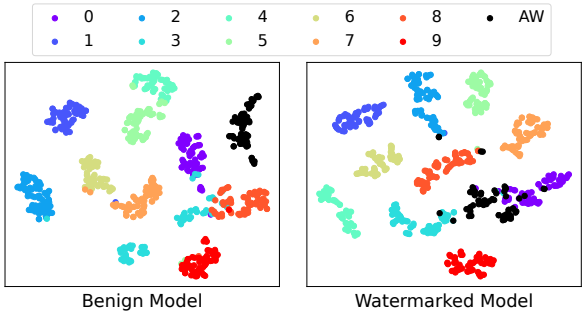


Figure 19: The t-SNE result

In this section, we aim to delve deeper into the mechanisms that contribute to the efficacy of our watermark. To do so, we use TSNE to visualize the distribution of features among benign models and watermarked models. Specifically, we craft a series of watermarked samples (AW) from Speaker-0 (purple). Next, we visualize all the samples from all classes (Speaker-0 to Speaker-9) as well as the watermarked sample (AW) in a benign model and a watermarked model. In Figure 19, each dot represent an utterance, and each color denotes a different speaker. The black dots are the watermarked audio that originated from Speaker-0 and target to Speaker-0. As can be seen from the benign model plot, the watermarked samples maintain a distinct distance from their actual label ('0'), though the two groups are still close to each other. Conversely, under the watermarked model, these watermarked samples are much closer to benign samples of the same class. Indicating that the watermarked model can recognize the watermarked sample as the benign sample, due to some intersections between those two groups. This observation shows that the benign model cannot correctly recognize the watermarked sample, while the watermarked model is capable of doing it.