Detecting Voice Cloning Attacks via Timbre Watermarking

Chang Liu¹, Jie Zhang²†, Tianwei Zhang², Xi Yang³, Weiming Zhang¹†, and Nenghai Yu¹
¹University of Science and Technology of China
²Nanyang Technological University
³Corresponding Authors
{hichangliu@mail., yx9726@mail., zhangwm@, ynh@}ustc.edu.cn, {jie_zhang, tianwei.zhang}@ntu.edu.sg

Abstract—Nowadays, it is common to release audio content to the public, for social sharing or commercial purposes. However, with the rise of voice cloning technology, attackers have the potential to easily impersonate a specific person by utilizing his publicly released audio without any permission. Therefore, it becomes significant to detect any potential misuse of the released audio content and protect its timbre from being impersonated.

To this end, we introduce a novel concept, “Timbre Watermarking”, which embeds watermark information into the target individual’s speech, eventually defeating the voice cloning attacks. However, there are two challenges: 1) robustness: the attacker can remove the watermark with common speech preprocessing before launching voice cloning attacks; 2) generalization: there are a variety of voice cloning approaches for the attacker to choose, making it hard to build a general defense against all of them.

To address these challenges, we design an end-to-end voice cloning-resistant detection framework. The core idea of our solution is to embed the watermark into the frequency domain, which is inherently robust against common data processing methods. A repeated embedding strategy is adopted to further enhance the robustness. To acquire generalization across different voice cloning attacks, we modulate their shared process and integrate it into our framework as a distortion layer. Experiments demonstrate that the proposed timbre watermarking can defend against various adaptive attacks (e.g., reconstruction-based removal attacks, watermark overwriting attacks), and achieve practicality in real-world services such as PaddleSpeech, Voice-Cloning-App, and so-vits-svc. In addition, ablation studies are also conducted to verify the effectiveness of our design. Some audio samples are available at https://timbrewatermarking.github.io/samples.

I. INTRODUCTION

“The voice is an instrument that you can learn to play and use in a way that is uniquely yours.”

– Kristin Linklater

We are already in the era of the Ear Economy, where audio content has become increasingly popular in today’s digital landscape. Many individuals enjoy sharing their voice artworks (e.g., music recordings, audio books, soundtracks) on public platforms, such as Spotify [1], Audible [2], Himalaya [3], SoundCloud [4], etc. Those services have significantly changed the way people consume content for entertainment and gaining knowledge.

Unfortunately, the rise of the voice cloning technology [5], [6], [7], [8] exposes new security challenges to the audio content sharing services. Recent advances in voice cloning leverage deep learning to accurately synthesize human voices indistinguishable from the target ones [9], [10]. This technology has been widely applied in different scenarios: in the entertainment industry, it can be used to create synthetic voices for characters in movies and video games; in healthcare, it can generate voices for patients who have lost their ability to speak due to illness or injury. However, an adversary can also exploit this technology to generate voices of other persons’ timbre (i.e., patterns, intonations, and other characteristics of human speech) for illegal or unauthorized purposes. This is dubbed voice cloning attack, which can cause various severe consequences, such as financial losses, reputation damage, and copyright infringement. For instance, a clip on YouTube shows that Biden announced a plan of launching a unclear attack against Russia, which was created by voice cloning and could cause public panic [11].

Therefore, unauthorized synthesis of valuable timbre shall be not acceptable, and it is crucial to establish safeguards against the potential misuse of voice cloning technology. Several attempts have been made to tackle this issue. However, they all suffer from some limitations. First, passive detection methods [12], [13], [14], [15] have been developed to identify whether a voice is produced using voice cloning. However, advanced voice cloning techniques make these methods obsolete, and create a perpetual arms race between voice synthesis and detection techniques [16]. Second, another line of defense strategies are introduced to proactively prevent voice cloning.
Specifically, Huang et al. [17] proposed to add adversarial noise to the target voice, causing the attacker to create synthetic voices with totally different timbre from the target one. However, this solution has two practical issues: (1) it needs to add significant amounts of noise to achieve the protection, which can obscure the original voice and affect its fidelity; (2) it needs domain knowledge about the specific voice cloning approach adopted by the attacker, and cannot be generalized to other attack approaches.

To address the limitations of existing solutions, motivated by the recent deep model watermarking technology [18], [19], we introduce a novel concept called “Timbre Watermarking” to protect the released timbre against voice cloning. As shown in Fig. 1, we proactively embed watermark information (e.g., ownership) into the target voice before releasing it to the public. To ensure its normal usage, the watermarked voice is indistinguishable from the original one and users will not notice its existence. An attacker may collect such watermarked voice without permission, and then apply a voice cloning method to generate synthetic voice with the same timbre but different content or semantic information. By extracting the watermark information from the synthetic voice, we are able to detect the fake speech and track its original timber reliably.

Current audio watermarking methods have demonstrated success in preserving fidelity while guaranteeing normal robustness (e.g., against desynchronization/recapturing attacks [20], [21]). However, they cannot be applied for timbre watermarking, due to their incapability of handling voice cloning attacks (explained in Sec. II-C). Here, we conclude two challenges in realizing timbre watermarking: (1) Generalization. We do not know how the attacker would launch the voice cloning attacks, including data collection, generative model, cloning strategies, etc. It is difficult to embed a timbre watermark that can resist all types of possible voice cloning strategies. (2) Robustness. The attacker may attempt to remove the potential watermarks by preprocessing the voice. For instance, he can crop the voice to randomize the speech length and complicate the resynchronization for watermark extraction; he can also compress the audio signals to damage the hidden watermark. Although a number of prior works have proposed methods to watermark audio works [22], [21], [23], [24], [25], as they are not targeting the voice cloning attacks, they are not robust and can be easily removed by the attacker. In Sec. VI-A we experimentally validate the limitations of these existing watermark solutions: the indeterminate length and scale of the synthesized speech will totally fail their resynchronization process before extracting, which usually depends on the synchronization codes, shifting mechanisms, or time scaling-invariant features.

In this paper, we propose a novel end-to-end timbre watermarking framework to defeat voice cloning attacks. Fig. 4 shows the overview of our methodology. Specifically, (1) to enhance the robustness, we propose to embed the watermark information into the transform domain. We adopt the Short-Time Fourier Transform (STFT) scheme to transfer the audio wave and embed the watermark into its frequency domain (vertical direction), which is inherently robust against voice processing operations along the time domain (horizontal direction). To further strengthen the watermark robustness, we repeat the embedding strategy along the horizontal direction. (2) To enhance the generalization, we investigate existing popular voice cloning strategies and find some common-used processing operations, such as scale modification, normalization, phase information discarding, and waveform reconstruction. Then, we insert these processes as a distortion layer between the watermark embedding and extraction modules and train them together in an end-to-end way. This distortion layer provides other modules the awareness of different processing operations when verifying the watermarks. After training, we discard the distortion layer and leverage the other modules for watermark embedding and extraction, respectively.

Extensive experiments demonstrate that our proposed methodology does not degrade the quality of the original timbre while guaranteeing the generalization and robustness against different existing and adaptive voice cloning attacks. Besides, we verify its effectiveness with some real-world commercial services, including PaddleSpeech [26], Voice-Cloning-App [27], and so-vits-svc [28]. We also conduct some ablation studies to evaluate our design, and provide some discussions and potential insights. We believe our proposed “Timbre Watermarking” can shed light on the field of illegal voice cloning detection and timbre protection.

In summary, the primary contributions of our work are concluded as follows:

- We point out that timbre rights are at significant risk of being compromised by voice cloning attacks and introduce a novel concept of “Timbre Watermarking” as a viable defense.
- To achieve “Timbre Watermarking”, we propose an end-to-end voice cloning-resistant audio watermarking framework. Innovatively, we repeatedly embedded watermark information in the frequency domain to resist common audio processing and modulate the shared process of different voice cloning attacks as a distortion layer to obtain generalization.
- Extensive experiments demonstrate the generalization and robustness of our proposed method against different voice cloning attacks including the adaptive ones. In addition, our method is applicable in real-world services, e.g., Paddle-Speech, Voice-Cloning-App, and so-vits-svc.

II. BACKGROUND AND RELATED WORKS

In this section, we first describe the common methods for voice cloning, followed by some countermeasures against voice cloning attacks. Finally, we provide some traditional audio watermarking methods and point out their limitations.

A. Voice Cloning

Voice cloning refers to the process of creating a synthetic voice that closely resembles the voice/timbre of a target person. This is mainly achieved by two mainstream techniques, namely voice conversion [5], [6] and text-to-speech (TTS) generation [7], [8]. Specifically, voice conversion is a technique that modifies the speech signal of an arbitrary speaker to make it sound like the target speaker’s voice while preserving the linguistic content of the original message. Comparably, TTS is a more flexible method to generate the desired speech of any given text without the need of the original speech for transfer.
Therefore, this paper mainly focuses on the TTS-based voice cloning attacks. Nevertheless, our proposed method is also applicable to voice conversion techniques: in Sec. V-F we showcase the effectiveness of our solution over real-world services, some of which are based on voice conversion. Fig. 2 presents the pipeline of a common TTS system, which can be divided into the following three components.

**Text Processing.** In order to generate high-quality speech output, the text needs to be preprocessed to extract linguistic and acoustic features. Text processing for TTS mainly includes text normalization [29] and linguistic analysis [9]. Text normalization aims to convert the text into a standard form that can be processed by the TTS system. This process consists of different operations, such as removing punctuation and special characters, converting numbers to their spoken form, and expanding abbreviations and acronyms. Linguistic analysis segments the text into units such as sentences, phrases, and words. It then processes the normalized text by converting it into a sequence of phonemes, which are the basic units of sound in a language, and tokenizes these phonemes using the model-specific tokenizer. The resulting tokenized phonemes are then fed into the models to generate synthesized speech. Additionally, advanced speech synthesis systems often perform prosody prediction, encompassing rhythm, stress, and intonation of speech. This prediction enables TTS models to produce speech with natural pitch, duration, and energy patterns. Some TTS models integrate syntactic and semantic analysis to comprehend the text’s structure and meaning, facilitating more precise and contextually relevant speech synthesis [30].

**Acoustic Model.** An acoustic model is a key component of TTS systems, responsible for converting linguistic information into acoustic features (e.g., spectrograms), which determine the sound of the synthesized speech. Compared with traditional simple statistical models [31], [32], [33], [34], DNN-based acoustic models [35], [7], [36], [8] give a superior performance. Wang et al. [7] proposed the first DNN-based acoustic model Tacotron, which uses a Recurrent Neural Network (RNN) to simulate the dynamic nature of speech signals and employs an attention mechanism to adjust the output based on different inputs. However, Tacotron has some distortion and noise issues. Afterward, Shen et al. [36] presented the enhanced Tacotron 2, which adopts a position-sensitive attention module to improve the synthesis quality. Nevertheless, both Tacotron and Tacotron 2 are computationally-intensive. To address the efficiency issue, Ping et al. [35] introduced a fully-convolutional sequence-to-sequence architecture rather than RNN. FastSpeech [37] uses an encoder-decoder transformer structure to rapidly generate mel-spectrogram in parallel for TTS and leverages an unsupervised approach to greatly simplify the model training process and reduce the demand for a large amount of audio data. FastSpeech 2 [8] further expands on this unsupervised training method by incorporating an acoustic prior to improve the output quality. In this paper, we adopt Tacotron 2 [36] and FastSpeech 2 [8] as the default acoustic model for voice cloning attacks, but our solution is general to other models as well.

**Vocoder.** With the above-obtained spectrograms, a vocoder is used to synthesize speech signals based on an analysis of their constituent frequency bands and pitch information. A well-known vocoder is Griffin-Lim [38], which reconstructs the original signal from Short-Time Fourier Transform (STFT). Although this algorithm is computationally inexpensive and effective at reconstructing speech signals, it will introduce perceptible background noise and distortions in the synthetic speech. It is also sensitive to the choice of parameters, which requires manual fine-tuning to achieve optimal results. Besides this traditional algorithm, there are also some deep learning algorithms used for generating high-quality audio signals, such as WaveGAN [39] and HiFiGAN [40]. By using a combination of convolutional and deconvolutional layers, WaveGAN is able to generate realistic audio signals that mimic real recordings, but it may sometimes produce distorted or unstable output when the generated signals are too complex or too long. By comparison, HiFiGAN uses a more advanced training process that includes the use of feature-matching loss functions and a multi-scale discriminative model. This allows to generate high-fidelity audio signals that sound more natural and accurate than those generated using earlier GAN-based methods. In this paper, we adopt Griffin-Lim [38] and HiFiGAN [40] as the default vocoder. Besides, we also consider the widely-used VITS [30], which directly transfers the text input to speech output in an end-to-end manner.

### B. Countermeasures against Voice Cloning Attacks

Existing strategies of resisting voice cloning attacks can be roughly classified into two categories.

**Passive Detection-based Strategy.** Passive detection seeks to identify whether the suspect speech from authentic humans or artificially generated. This is normally achieved via the analysis of specific features. For example, Gao et al. [13] introduced an audio-based CAPTCHA system to differentiate human voices from synthetic ones using metrics such as short-term energy, average amplitude, and zero-crossing rate. DeepSonar [15] leverages layer-wise neuron activation patterns to effectively discern authentic voices from AI-synthesized ones. However, both the above methods cannot generalize to unseen data distributions. To remedy this issue, Ahmed et al. [12] proposed Void, an efficient solution for voice spoofing detection based on liveness detection, which analyzes the spectral power differences between live-human voices and spoofing voices. In addition to feature-based methods, there are also some approaches to achieve good detection capability with the help of deep learning [14], [41]. To solve the problem of diverse statistical distributions of different synthesizing methods, Zhang et al. [42] proposed a one-class learning anti-spoofing system (One-Class) to effectively detect unseen synthetic voice spoofing attacks.
A brand new audio, without TTS Model
Embed watermark
Embed sync code
Extract feature
cropping distortion [48], etc. In a nutshell, existing audio recording [21], desynchronization distortion [23], [20], exploring robustness under intricate conditions, such as resilience against common and unavoidable signal processing modifications to avoid being perceived by human ears [44], imperceptibility [22], [20]. Besides, some schemes try to less prominent audio features contributes to the watermark’s components leads to enhanced fidelity in the watermarking generally accepted that modification of the higher frequency executed by altering these frequency domain features. It is carrier audio. The embedding of the watermark is subsequently frequency domain analysis to extract salient features from the current audio watermarking methods predominantly utilize set of performance criteria: fidelity and robustness. For fidelity, C. Audio Watermarking

Audio watermarking technologies strive to achieve a dual set of performance criteria: fidelity and robustness. For fidelity, current audio watermarking methods predominantly utilize frequency domain analysis to extract salient features from the carrier audio. The embedding of the watermark is subsequently executed by altering these frequency domain features. It is generally accepted that modification of the higher frequency components leads to enhanced fidelity in the watermarking process [43]. Concurrently, making subtle adjustments to less prominent audio features contributes to the watermark’s imperceptibility [22], [20]. Besides, some schemes try to use psychoacoustic modeling to set thresholds for watermark modifications to avoid being perceived by human ears [44], [45], [46]. For robustness, some research has demonstrated resilience against common and unavoidable signal processing distortions [47]. More recently, the focus has shifted toward exploring robustness under intricate conditions, such as re-recording [21], [24], desynchronization distortion [23], [20], cropping distortion [48], etc. In a nutshell, existing audio watermarking can already well balance the trade-off between fidelity and normal robustness.

As mentioned above, the collected speech may be modified before voice cloning attacks via some preprocessing operations, such as cropping and time scaling. This can cause desynchronization to fail the subsequent watermark extraction. Fig. 3 illustrates some countermeasures based on traditional audio watermarking schemes: a) leveraging synchronization codes or shift mechanisms with sliding windows for resynchronization [21], [48]; b) employing time scaling invariant features to withstand scale transformations [23], [20]. Based on these two ideas, Zhao et al. [23] designed FSVC, an audio feature that is insensitive to time-domain scale transformation based on frequency-domain signal processing. It is combined with audio averaging segmentation to achieve resistance to desynchronization caused by global-scale transformation. Liu et al. [21] designed RFDLM, a robust feature against scaling. It is combined with synchronization codes to achieve robustness to desynchronization attacks. However, none of these audio watermarking methods exhibit satisfactory robustness against both cropping and time scaling, let alone the more complex voice cloning process. In Sec. VI-A we evaluate two audio watermarking methods (i.e., FSVC [23] and RFDLM [21]), and show that they totally fail to detect voice cloning attacks. Concurrently, deep learning-based end-to-end audio watermarking has recently emerged [24], [25], [49], yet it encounters considerable limitations. Existing methods prioritize robustness; however, in the given scenario, the audio possesses a fixed length, rendering it challenging to counteract distortions arising from desynchronization.

Different from the above methods, our solution integrates time-independent features into the audio watermarking algorithm, making it feasible to maintain the integrity of the watermark information even in the face of cropping and time scaling. We provide a qualitative analysis of the impact of distortion operations on the watermarked speech and demonstrate that our method can effectively resist voice cloning attacks. In a nutshell, our method pursues a novel robustness property against voice cloning attacks, which is achieved by simulating this attack and inserting it between the training of watermark embedding and extraction. For enhanced fidelity, we also introduce an extra discriminator for adversarial training.

III. THREAT MODEL

We consider a scenario which involves three parties: 1) users shares their audio data on a public platform; 2) the platform provider seeks to protect the shared audio from potential misuse; 3) the attacker collects the target audio from the platform and attempts to generate synthetic audio with the same timbre using advanced voice cloning techniques.

A. Users’ Ability and Goal

In order to access the platform, users must first register the service with their authorship information. They hope that their audio data will not be used maliciously. They have the right to request the platform to deploy protection over the uploaded audio, and perform synthetic audio verification when suspiciously cloned audio is identified.
B. Platform Provider’s Ability and Goal

The platform provider applies a novel watermark embedding algorithm $\mathcal{E}(\cdot)$ to embed a watermark $w$ (e.g., authorship information) into users’ speech samples $s$ before releasing them, to safeguard their voice timbre, i.e., $s' = \mathcal{E}(s, w)$. The platform provider needs to guarantee the speech fidelity, namely, making $s_w$ similar to the pristine $s$ as much as possible. Let $S_w = \{s_{w1}, s_{w2}, \ldots, s_{wn}\}$ represent a set of watermarked speech samples from the target speaker $T$ with watermark $w$. The platform provider also has a watermark extraction algorithm $\mathcal{E}\mathcal{X}(\cdot)$ to extract the pre-defined watermark $w$ from $s_w$, i.e., $w' = \mathcal{E}\mathcal{X}(s_w) \rightarrow w$. Given a suspicious audio $ss$, the platform provider can apply $\mathcal{E}\mathcal{X}(\cdot)$ to check if any watermark can be extracted from it, as evidence of voice cloning attacks.

C. Attacker’s Ability and Goal

Let $S = \{s_1, s_2, \ldots, s_n\}$ be a set of speech samples from a target speaker $T$. The attacker collects $S$ and pre-processes it to construct a text-speech dataset $D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)\}$, where $x_i$ is a text transcription and $y_i$ is the corresponding speech sample. With $D$, the attacker can train his TTS model $M$ in a supervised way. As mentioned in Sec. II-A, a TTS model $M$ usually consists of two components, namely, the Acoustic model $MA$ and the Vocoder $MV$. After that, for an arbitrary text transcription $\hat{x}$, $M$ can synthesize the corresponding speech $\hat{y}$ as:

$$\hat{y} = M(\hat{x}) = MV(MA(\hat{x})).$$

With our proposed Timbre Watermark, the attacker can only obtain a watermarked dataset from the platform, i.e., $D_w = \{(x_1, y_{w1}), (x_2, y_{w2}), \ldots, (x_m, y_{wm})\}$, where $y_{wi}$ denotes a watermarked speech sample. Then he uses $D_w$ to train his acoustic model $MA_w$. Based on the attacker’s capability of training the subsequent vocoder $MV$, we define the following three attack scenarios:

- **Professional Voice Cloning Attack.** In this scenario, the attacker is an expert on TTS tasks, and has the capability of fine-tuning a general vocoder $MV$ on $D_w$ to acquire his superior vocoder $MV_w$. This vocoder $MV_w$ can convert the synthesized mel-spectrograms into counterfeit speech closely resembling the target speaker’s timbre. The attack scenario can be formulated as follows:

$$\hat{y}_w^P = MW^P(\hat{x}) = MV_w(MA_w(\hat{x})).$$

- **Regular Voice Cloning Attack.** In this scenario, the attacker only trains his own acoustic model $MA$ and then appends an off-the-shelf pre-trained vocoder $MV$ to conduct voice cloning attacks, i.e.,

$$\hat{y}_w^R = MW^R(\hat{x}) = MV(MA_w(\hat{x})).$$

- **Low-quality Voice Cloning Attack.** In certain instances, the attacker may lack the resources necessary to obtain and utilize high-quality pre-trained vocoders for speech synthesis, or the capability of fine-tuning vocoders to target their victims specifically. Consequently, he might resort to traditional, non-deep learning techniques for synthesizing speech waveforms, such as the Griffin-Lim algorithm GL($\cdot$) [38]. The attack scenario can be described as follows:

$$\hat{y}_w^L = MW^L(x) = GL(MA_w(x)).$$

D. Adaptive Attacks

Besides the above common voice cloning attacks, we also consider the possible adaptive attacks, where the attacker has the knowledge of our timbre watermarking strategy, and tries to remove the watermarks while preserving the quality of the cloned audio. We implement and evaluate the following adaptive attacks in Sec. V-E:

1) **Attackers without access to the watermarking model.** The attacker is not allowed to access the watermarking model and can only launch some data-level attacks as follows:

- Before conducting the voice cloning attack, the attacker pre-processes the collected audio with some harmful operations, such as severe compression and low pass filter.
- The attacker can launch the watermark evading attack with VAE reconstruction like [50] before the voice cloning attack.
- The attacker can collect some pristine data of the target speaker, and synthesize the audio with the mixed dataset.

2) **Attackers with access to the watermarking model.** When an attacker can access the target model, he has more means to break the watermark as follows:

- With access to the watermark encoder, the attacker can operate a watermark overwriting attack, i.e., further embedding another watermark on the collected watermarked data before the voice cloning attack.
- The attacker can execute the watermark evading attack by deploying a watermark erasing VAE before voice cloning.
- With access to the watermark extractor, the attacker attempts to explore the location of the embedded watermark, and then directly remove this region before applying voice cloning.
- The attacker can also deploy a classifier to detect the presence of a watermark. This classifier is then used against a speech synthesis model during training (i.e., domain-adversarial training), which causes the synthesis model to synthesize audio without the watermark.

3) **Combining multiple attack strategies.** We consider integrating diverse attack schemes, encompassing regular preprocessing, harmful preprocessing, domain-specific advanced training, VAE reconstruction, and watermark overwriting.

IV. METHODOLOGY

Fig. 4 shows the overview of our new framework. It consists of three components: a watermark embedding module, a watermark extraction module, and an intervening distortion layer to bolster the robustness against distortions. These components are jointly trained. Below we provide a detailed description of each component.

A. Watermark Embedding

As shown in the left part of Fig. 4, similar to many audio-based information hiding methods [51], [52], [53], we adopt the widely-used linear spectrogram [54] of speech audios as the carrier to embed watermark information. Meanwhile, this step allows us to add the same watermark in the frequency domain at different time periods for time-independence, due to the short time window property brought about by the time-frequency localization of STFT. Specifically, given a single-channel raw speech audio $a$ of the flexible length $N$, we first
Apply the Short-Time Fourier Transform operation (\texttt{STFT}(\cdot)) on it to produce a spectrogram \(s\) and the corresponding phase information \(p\):

\[
s, p = \text{STFT}(a) .
\]  
(1)

Since the magnitude spectrogram contains most of the information, we use the magnitude spectrogram \(s\) as the carrier for easy training, while the phase spectrogram is only used for signal recovery \([51], [52]\). Then, we feed \(s\) into the Carrier Encoder \(\text{EN}_c\) to obtain the encoded carrier features \(f_c\):

\[
f_c = \text{EN}_c(s).
\]  
(2)

Simultaneously, we feed the \(n\)-bit watermark information \(w\) into the Watermark Encoder \(\text{EN}_w\) to obtain the encoded watermark features \(f_w\):

\[
f_w = \text{EN}_w(w).
\]  
(3)

Next, we concatenate \(f_c\) and \(f_w\) to obtain the final input \(f_+\) for the subsequent Watermark Embedder \(\text{EM}\). Considering that the length of speech samples is flexible, we propose to repeat \(f_w\) along the time axis to achieve time-independence, i.e., Repeat\((f_w, t)\), whose shape size is consistent to \(f_c\). We point out that such a repeated strategy can make the watermark information inherently robust to distortions in the time domain. Motivated by DenseNet \([55]\), we also introduce a skip concatenation to increase the nonlinearity and characterization ability of the model and preserve the information of the original carrier \(s\) to the maximum extent. All this preprocessing can be written as follows:

\[
f_+ = \text{Concatenate}(f_c, s, \text{Repeat}(f_w, T)),
\]  
(4)

where \(f_w \in \mathbb{R}^{C_w \times 1 \times H}\), \(f_c \in \mathbb{R}^{C_c \times T \times H}\), and \(f_+ \in \mathbb{R}^{(C_w+1+C_c) \times T \times H}\). Afterward, \(f_+\) is fed into \(\text{EM}\) to obtain the watermarked audio signal \(\hat{s}_w\):

\[
\hat{s}_w = \text{EM}(f_+).
\]  
(5)

Finally, we reconstruct the watermarked audio \(\hat{a}_w\) by applying Inverse Short-Time Fourier Transform (\texttt{ISTFT}(\cdot)) to the decoded spectrogram \(\hat{s}_w\) and original phase information \(p\):

\[
\hat{a}_w = \text{ISTFT}((\hat{s}_w, p)).
\]  
(6)

To ensure the audio quality fidelity, we introduce the watermark embedding loss \(L_e\), i.e.,

\[
L_e = \frac{1}{M} \sum_{i=1}^{M} ((\hat{a}_w)_i - a_i)^2,
\]  
(7)

where \(M\) is the length of the audio in the time dimension. To further improve the fidelity and minimize the domain gap between \(a\) and \(a_w\), inspired by the training strategy in GAN \([56]\) to ensure the realism of the generated data, an adversarial loss \(L_{adv}\) is also added to make an extra discriminator \(D\) cannot distinguish \(a_w\) from the pristine \(a\), as shown in Fig. 4, i.e.,

\[
L_{adv} = -\log(\sigma(D(a_w))).
\]  
(8)

Meanwhile, \(D\) is constrained by \(L_d = -\log(\sigma(D(a))) - \log(1 - \sigma(D(a_w)))\), where \(\sigma(\cdot)\) denotes the sigmoid function.

**B. Watermark Extracting**

Given a watermarked speech \(a_w\), the decoder needs to recover watermark \(w'\) as consistent as the original watermark \(w\). Specifically, we first follow Eq. (1) to conduct \texttt{STFT} on \(a_w\) to obtain the phase information \(p_w\) and spectrogram \(s_w\), which are then fed into the Watermark Extractor \(\text{EX}\) to obtain the extracted watermark features \(f'_w\):

\[
f'_w = \text{EX}(s_w).
\]  
(9)

Then, we use the Watermark Decoder \(\text{DE}\) to decode the watermark information \(w'\) from the time domain (horizontal direction) averaged watermark features \(f'_w\), i.e.,

\[
w' = \text{DE}(\text{Average}(f'_w)).
\]  
(10)

This average operation corresponds to the previous repeat operation in Eq. (4), and together they realize the time-independence of watermarking. To ensure the accuracy of watermark extraction, we introduce a watermark extraction loss \(L_w\), i.e.,

\[
L_w = \frac{1}{N} \sum_{i=1}^{N} (w'_i - w_i)^2,
\]  
(11)

where \(N\) is the length of the watermark sequence.

**C. Distortion Layer**

To enhance the robustness against voice cloning attacks, we further insert a distortion layer between the watermark embedding stage and watermark extracting stage. Given a watermarked audio signal \(a_w\), we can obtain the distorted audio \(\hat{a}_w\) after the distortion layer \(\text{DP}\), i.e., \(\hat{a}_w = \text{DP}(a_w)\), which is then fed into the watermark extracting stage. Note that the distortion layer is only appended in the training stage, and will be discarded in the actual embedding and extraction stages. Below we detail each component of the distortion layer.
**ISTFT Distortion.** When performing ISTFT, if the amplitude spectrogram is modified in some way to embed the watermark while the phase spectrogram remains unchanged, then the final signal obtained will be a complex matrix, and we will only keep the real part as the final signal, which incurs some loss. This differential distortion is already included in the embedding process as shown in Eq. (6).

**Normalization Distortion.** During speech synthesis training, the timbre information is unrelated to the overall energy, or amplitude, of the audio signal. Consequently, the speech synthesis model often normalizes the amplitude of the audio as part of the preprocessing, i.e.,

\[ a^n_w = \frac{a_w}{\max(|a_w|)}. \]  

(12)

This normalization operation, however, may result in the potential erasure of the watermark information embedded within the audio signal. Thus, we append it into the distortion layer.

**Transformation Distortion.** For speech synthesis, mel-spectrogram is mainly used as supervisory signals. Specifically, during model training, the audio needs to be mel-transformed to extract this feature from the speech signal. Therefore, in the distortion layer, we perform the mel-transform operation on the watermarked audio to derive a mel-spectrogram, i.e.,

\[ \hat{m}s_w = \text{Mel}(a^n_w). \]

(13)

**Wave Reconstruction Distortion.** During the synthesis process, the vocoder is employed to reconstruct a waveform from a mel-spectrogram while omitting the phase information. This transformation is irreversible and lossy, which can affect the intermediate audio features, particularly the spectrogram, and potentially lead to the loss of embedded watermark information. Consequently, the resulting distribution deviates from its original one. To address this issue, we employ the vocoder for waveform reconstruction.

Specifically, we adopt the conventional rule-based Griffin-Lim [38] GL(·) as the vocoder, i.e.,

\[ \hat{a}_w = \text{GL}(\hat{m}s_w). \]

(14)

This operation introduces more severe distortions compared to learnable vocoders, and we assume it can enhance the transferable robustness (Table I supports our assumption).

**Pipeline of the Distortion Layer.** The pipeline of the distortion layer DP can be written as follows:

\[ \text{DP}(a^w) = \text{GL}((\text{Mel}(\frac{a^w}{\max(|a^w|)}))). \]

(15)

Given the distorted watermarked speech \( \hat{a}_w \), the decoder needs to recover the watermark \( \hat{w} \) as consistent as the original watermark \( w \). To recover the watermark information \( \hat{w} \) from \( \hat{a}_w \), we first apply STFT on \( \hat{a}_w \) to obtain the phase information \( \hat{p}_w \) and spectrogram \( \hat{s}_w \), which are then fed into the Watermark Extractor EX to obtain the extracted watermark features \( f^w \) using the Watermark Decoder DE, i.e.,

\[ \hat{w} = \text{DE}(\text{Average}(f^w)). \]

(17)

Here, we introduce \( \hat{L}_w \) to ensure the accuracy of watermark extraction after the distortion layer, i.e.,

\[ \hat{L}_w = \frac{1}{m} \sum_{i=1}^{m} (\hat{w}_i - w_i)^2. \]

(18)

**Model Training.** We jointly train the above three modules in our framework. The whole loss function \( L \) is formulated as:

\[ L = \lambda_c \cdot L_e + \lambda_{adv} \cdot L_{adv} + \lambda_w \cdot (L_w + \hat{L}_w), \]

(19)

where \( \lambda_c \), \( \lambda_d \), and \( \lambda_w \) are hyper-parameters to balance the three terms. It is worth mentioning that we simultaneously optimize for watermark extraction accuracy from both distorted watermarked audio and undistorted watermarked audio, ensuring generalization. At the same time, we train D to minimize the loss \( \hat{L}_d \).

**Watermark Embedding.** As shown in Fig. 1, before releasing the original audio, the platform provider embeds the pre-defined watermark information (e.g., a bit string) into it by the well-trained EN$_c$, EN$_w$, and EM.

**Watermark Extraction and Attack Detection.** Given a suspicious audio, we adopt the well-trained Extractor EX and DE to extract the watermark from it. Then, we use bit recovery accuracy to evaluate the similarity between the extracted and ground-truth watermark. If the accuracy is larger than the pre-defined threshold, we can finally claim that this suspicious audio is synthesized from the protected ones without permission, and the voice cloning attack is detected.

**V. EXPERIMENTS**

We present comprehensive evaluations to validate the effectiveness of our framework. We first describe our experiment setup in Sec. V-A. Then we demonstrate the proposed framework can satisfy different requirements, including fidelity (Sec. V-B), generalization (Sec. V-C), robustness against potential distortions (Sec. V-D) as well as adaptive attacks (Sec. V-E). In Sec. V-F we conduct evaluations on three real-world services to show the practicality of our solution. Sec. V-G gives the ablation study to verify our design.

**A. Experiment Setup**

**Implementation Details.** For all of EN$_c$, EM, and EX, we adopt simple fully 2D convolutional networks. These networks maintain the dimensions of feature maps at each layer and employ a skip gated block as their fundamental unit, which integrates the Gated Convolutional Neural Network [57] with a skip connection [58] for processing. For Watermark Encoder EN$_w$, we leverage a fully connected layer that utilizes the LeakyReLU activation function [59] for enhanced performance. Meanwhile, the Watermark Decoder, denoted as DE, operates as a pure linear layer within the network architecture. The Discriminator D has a comprehensive architecture.
consisting of STFT, three ReluBlocks, an average pooling layer, and a linear layer. The ultimate output of this structure is the prediction result for the input audio. The implementation of the models is available through the source code on our website. Additionally, to jointly train the three modules in our platform, we use the Adam optimizer [60] with $\beta_1 = 0.9, \beta_2 = 0.98, \epsilon = 10^{-9}$, and a learning rate of $2e^{-5}$ for optimization. We set $\lambda_c = 1$ and $\lambda_w = \lambda_d = 0.01$ in Eq. (19). For STFT, we adopt a filter length of 1024, a hop length of 256, and a window function applied to each frame with a length of 1024.

Datasets. For the voice cloning model, we employ LJSpeech [61] (Clip length varies from 1 to 10 seconds), a well-established benchmark dataset for speech synthesis. It includes many well-aligned text-speech pairs. For watermarking model training, we employ the standard training set train_clean100 of LibriSpeech [62], where the length of audio samples varies, typically around 10 seconds in duration. We also conduct testing based on its standard testing set with 2620 audio samples. All audio samples are resampled to the sampling rate of 22.05 kHz, for both LJSpeech and LibriSpeech datasets. There is no overlap between the training data of the watermarking model and voice cloning models.

Metrics. For fidelity evaluation, i.e., the quality of watermarked audio, we adopt three objective metrics: Signal-to-Noise Ratio (SNR), Perceptual Evaluation of Speech Quality (PESQ) [63], and Speaker Encoder Cosine Similarity (SECS) [64], and an additional subjective metric: Mean Opinion Score (MOS). In detail, SNR is only used to measure the magnitude of quality loss resulting from watermarking and audio processing, while PESQ provides a better assessment of speech quality (i.e., imperceptibility) than SNR by considering the specifics of the human auditory system. SECS leverages the speaker similarity to evaluate the fidelity, and a higher value indicates a stronger similarity. We follow prior works [64], [65] to compute this metric using the speaker encoder of the Resemblyzer [66] package. In practice, SECS is often used for voice authentication, and the audio is determined to bypass authentication when SECS > 0.9 [67]. In MOS evaluations, we invite 10 participants to rate the naturalness and quality of the target speech with five ratings (1: Bad, 2: Poor, 3: Fair, 4: Good, 5: Excellent).

For the effectiveness of watermark extraction, we use the bit recovery accuracy (ACC), calculated from all audios in the standard test set of LibriSpeech (each audio is embedded with a random watermark). Furthermore, we evaluate the robustness against voice cloning attacks by examining ACC derived from 500 synthesized speech samples (an identical watermark), which correspond to 500 text segments from the LJSpeech test set. The default watermark length is 10 bits.

B. Fidelity

We first compare the fidelity of the protected audio between our method and the adversarial perturbation-based method [17]. It is worth noting that this method [17] is not effective in defeating different voice cloning attacks, so the comparison here is just for fidelity evaluation. We directly use their released speech examples on the web page\(^1\) for comparison.

\(^1\)https://yistlin.github.io/attack-vc-demo/

Fig. 5 shows the SNR, PESQ and SECS metrics of the two methods, respectively. For all cases, our method has a superior fidelity to Huang’s method [17]. Specifically for SECS, where a lower value indicates better speaker similarity, our method outperforms [17] by a large margin. We also conduct a subjective fidelity comparison: we select 20 distinct segments of both watermarked and non-watermarked audio samples, as well as the adversarial audio specimens delineated in [17]. Fig. 6 shows the speech quality ratings from 10 participants. The conclusion is consistent with the objective evaluation. The corresponding audio samples for the above comparison are available on our project website.

C. Generalization

We showcase the general effectiveness of our method against different voice cloning attacks described in Sec. III-C: professional, regular and low-quality voice cloning attacks. All the speech data of the target speaker are watermarked. To ensure the reliability of our experiments, we randomly generate two different 10-bit watermarks for repeated testing and report the average the results, as shown in Table I.

Professional Voice Cloning Attack. The attacker trains the acoustic model of voice cloning specifically for the targeted speaker and fine-tunes the vocoder to convert the synthesized mel-spectrograms into the speech that closely resembles the target speaker’s voice. To simulate such an attack, we adopt Fastspeech2 [8] and Tacotron2 [36] as the acoustic model, while using Hifi-GAN* [40] fine-tuned on watermarked data (1K segments of 1-10s speech are employed) as the subsequent vocoder. For end-to-end single-stage models VITS [30], we directly train the model using the watermarked dataset.

As shown in the red regions of Table I, our method achieves 100% watermark extraction accuracy across different professional voice cloning attacks. In contrast, Table II provides the watermark extraction accuracy across the corresponding voice cloning attack on the watermark-free dataset, which behaves
TABLE I: Generalization across different voice cloning attacks. * denotes using a watermarked dataset to train the acoustic model or fine-tune the vocoder, otherwise using a watermark-free dataset. Red, blue, and gray areas represent professional, regular, and low-quality attacks, respectively. The quality of the related synthesized speech is also provided.

<table>
<thead>
<tr>
<th>Model</th>
<th>Acoustic Model</th>
<th>Vocoder</th>
<th>PESQ↓</th>
<th>SECS↑</th>
<th>ACC↑</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fastspeech2*</td>
<td>Hifi-GAN* [40]</td>
<td>1.0578</td>
<td>0.8957</td>
<td>1.0000</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Hifi-GAN* [40]</td>
<td>1.0712</td>
<td>0.8965</td>
<td>0.9933</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Griffin-Lim [38]</td>
<td>1.1129</td>
<td>0.7034</td>
<td>1.0000</td>
<td></td>
</tr>
<tr>
<td>Tacotron2*</td>
<td>Hifi-GAN* [40]</td>
<td>1.1143</td>
<td>0.8598</td>
<td>1.0000</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Griffin-Lim [38]</td>
<td>1.1971</td>
<td>0.7125</td>
<td>1.0000</td>
<td></td>
</tr>
<tr>
<td>VITS* [30]</td>
<td></td>
<td>1.0342</td>
<td>0.9085</td>
<td>1.0000</td>
<td></td>
</tr>
</tbody>
</table>

TABLE II: Extraction results with all models trained on watermark-free data.

<table>
<thead>
<tr>
<th>Model</th>
<th>Acoustic Model</th>
<th>Vocoder</th>
<th>PESQ↓</th>
<th>SECS↑</th>
<th>ACC↑</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fastspeech2</td>
<td>Hifi-GAN [40]</td>
<td>1.0816</td>
<td>0.9031</td>
<td>0.5270</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Griffin-Lim [38]</td>
<td>1.0756</td>
<td>0.7158</td>
<td>0.5208</td>
<td></td>
</tr>
<tr>
<td>Tacotron2</td>
<td>Hifi-GAN [40]</td>
<td>1.1294</td>
<td>0.9028</td>
<td>0.5744</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Griffin-Lim [38]</td>
<td>1.1916</td>
<td>0.7126</td>
<td>0.5559</td>
<td></td>
</tr>
<tr>
<td>VITS [30]</td>
<td></td>
<td>0.8688</td>
<td>0.9213</td>
<td>0.4775</td>
<td></td>
</tr>
</tbody>
</table>

like random guessing (~50% ACC). Such comparison highlights the effectiveness of our timbre watermarking solution. In the two tables we also show the quality of synthesized speech, which is calculated between the synthesized speech and the corresponding watermark-free speech of the target speaker. We find that watermarking has negligible impact on the synthesized speech.

**Regular Voice Cloning Attack.** The attacker directly uses a pre-trained vocoder to convert the synthesized mel-spectrogram into the synthesized speech. Here, we take Hifi-GAN [40] as an example, which is pre-trained on watermark-free speech. From the blue regions of Table I, we observe that our method still achieves a high ACC (over 99%). In other words, the unwatermarked vocoder Hifi-GAN has less influence on the watermark information, which is still preserved in the subsequent synthesized speech.

**Low-quality Voice Cloning Attack.** The attacker trains an acoustic model on the target speaker’s speech data to synthesize mel-spectrograms, which are then transferred into speech waveforms using the Griffin-Lim vocoder. In this case, the mel-spectrograms synthesized by the acoustic model contain the watermark information, and the Griffin-Lim algorithm does not include any trainable parameters. Therefore, it does not tend to transfer watermarked mel-spectrograms to non-watermarked waveforms, as the watermark-free pre-trained vocoder Hifi-GAN does. As shown in the grey regions of Table I, the watermark extraction ACC can still reach 100%, which is even slightly better than the regular voice cloning attack.

**D. Robustness**

We evaluate the robustness of our approach against unseen distortions including cropping and other preprocessing operations. Robustness against cropping is particularly important because it is an unavoidable and intensive preprocessing operation in data collection before voice cloning attacks. Fig. 7 shows the ACC curve with different cropping ratio, when the watermarked audio is cropped from the front, middle and behind. Fig. 15 in the Appendix shows an example of cropping audio. We observe that even when 90% of the audio is cropped, the watermark can still be extracted with 100% accuracy. It is worth mentioning that different from the synchronization-preserving cropping used in [22] and [24], we directly crop off the audio to make it lose synchronization in the time dimension. Thus, the above results also indicate that our watermarking scheme can resist desynchronization attacks as it is intrinsically robust to distortions in the time domain. We further test whether our scheme can withstand other common audio processing distortions. As shown in Table III, the audio watermarked by our method can resist various preprocessing operations, and we obtain above 90% ACC in all cases.

**E. Resistance Against Adaptive Attacks**

We consider different adaptive attacks in Sec. III-D, and validate the corresponding resistance of our solution.

**Preprocessing Before Voice Cloning Attack.** In this scenario, the attacker wants to apply some preprocessing operations on the training audio before voice cloning attacks. According to the influence on the audio quality, we further categorize the attacks into two parts: regular preprocessing that does not lead
speech data can still retain 66.36% of watermark information, to achieve an extraction accuracy of over 95% (96.38%) in information. A 75% ratio of watermarked speech data is sufficient synthesized speech contains more complete watermark information, the corresponding watermark extraction accuracy. We observe model, VITS [30], as the default attack model. Fig. 8 shows on datasets with different watermark ratios. To expedite the data of the target speaker. To explore the effectiveness of our Voice Cloning Attack with Partial Unwatermarked Data.

We further consider two cases: a) the attacker adopts some common regular preprocessing operations before voice cloning, trying to occasionally bypass the watermark detection; b) the attacker leverages harmful preprocessing to intentionally degrade watermarked data before voice cloning. We present the corresponding results for the above cases in Table IV (VITS [30] is utilized as the default voice cloning attack model). It is noted that “Combined” signifies the application of a randomly selected preprocessing approach to each individual audio data sample. In particular, regular preprocessing does not influence the quality of the synthesized speech (above 0.90 SECS), and our watermark extraction accuracy reaches 100% for all. For harmful preprocessing, our proposed method can still achieve nearly 90% ACC, even though the quality of the synthesized speech is destroyed significantly (nearly 0.65 SECS). Some synthesized samples can be found on our project website.

Voice Cloning Attack with Partial Unwatermarked Data. The attacker may have access to some unwatermarked speech data of the target speaker. To explore the effectiveness of our approach in such a scenario, we train speech synthesis models on datasets with different watermark ratios. To expedite the experimental process, we use the end-to-end speech synthesis model, VITS [30], as the default attack model. Fig. 8 shows the corresponding watermark extraction accuracy. We observe that with a higher ratio of watermarked speech dataset, the synthesized speech contains more complete watermark information. A 75% ratio of watermarked speech data is sufficient to achieve an extraction accuracy of over 95% (96.38%) in synthesized speech, while only a 25% ratio of watermarked speech data can still retain 66.36% of watermark information, which is still effective in detecting such adaptive attack.

Watermark Overwriting Attack. With the collected watermarked audio data, the attacker can further embed his own watermark before conducting the voice cloning attack. In our threat model, voice cloning attacks are detected by the platform’s extractor. Nevertheless, we also show the results based on the attacker’s extractor. Specifically, we embed the original watermark (wm1) using our proposed method. To further embed the attacker’s watermark (wm2), we first adopt existing audio watermarking schemes FSVC [23] and RFDLM [21]. As shown in Table V, the original watermark can still be extracted as the evidence of voice cloning attacks, while the attacker fails to extract his watermark. This is because FSVC [23] and RFDLM [21] are fragile to voice cloning attacks, which do not interrupt our watermark signal during the attacks.

In addition, we further assume that the attacker can utilize our proposed method for watermark overwriting attacks. If the attacker is an insider malicious user, he can directly utilize the well-trained embedding and extraction models by our method. Otherwise, he can train his own model with the proposed strategy, which is denoted as “the proposed *” in Table V. The results show that our method will fail to resist the overwriting attack only when an inner attacker exists. To address this issue, we further design a weighted embedding process, which is controlled by the subsequent watermark decoder, and add the watermark overwriting distortion in the original distortion layer to fine-tune the model. As shown in Table V, the new strategy will not degrade the audio quality and achieve a 100% ACC against overwriting attacks. More details are in Appx. B3.

Audio Reconstruction-based Removal Attacks. We try to compromise the watermark by audio reconstruction, i.e., the adaptive attacker may try to train a reconstruction model with many watermarked-unwatermarked data pairs. For the audio reconstruction model, we adopt the state-of-the-art popular MelVAE2. For data pairs, we use 10,000 audios from Librispeech [62] as non-watermarked audio, and embed random watermarks into them by the proposed method to obtain the corresponding watermarked audio. Then, we train MelVAE with such data pairs in a supervised manner, aiming to transform watermarked audio into non-watermarked one. We observe that such an audio reconstruction model can only produce speech with poor quality (some samples are shown on our web page). Nevertheless, the accuracy of watermark extraction can still reach 74.90%.

Besides, we consider an existing audio watermark removal attack using a Variational Autoencoder (VAE) pre-trained on a clean audio dataset [30] and test our method under such an attack. Specifically, we use the VAE of the audio

<table>
<thead>
<tr>
<th>Pre-processing</th>
<th>PESQ↑</th>
<th>SECS↑</th>
<th>ACC↑</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resampling 16K</td>
<td>1.0775</td>
<td>0.9122</td>
<td>1.0000</td>
</tr>
<tr>
<td>Regular</td>
<td>1.0347</td>
<td>0.9077</td>
<td>1.0000</td>
</tr>
<tr>
<td>Mp3 Compression 64k</td>
<td>1.0776</td>
<td>0.9064</td>
<td>1.0000</td>
</tr>
<tr>
<td>Combined</td>
<td>1.0324</td>
<td>0.6567</td>
<td>0.9144</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
<th>PESQ↑</th>
<th>SECS↑</th>
<th>wm1-ACC↑</th>
<th>wm2-ACC↑</th>
<th>wm2 ACC#</th>
</tr>
</thead>
<tbody>
<tr>
<td>FSVC [23]</td>
<td>1.0334</td>
<td>0.9115</td>
<td>1.0000 (✓)</td>
<td>0.4000 (×)</td>
<td>0.5544 (×)</td>
</tr>
<tr>
<td>RFDLM [21]</td>
<td>1.0727</td>
<td>0.9102</td>
<td>1.0000 (✓)</td>
<td>0.4000 (×)</td>
<td>0.4986 (×)</td>
</tr>
<tr>
<td>The Proposed</td>
<td>0.9891</td>
<td>0.8968</td>
<td>0.9400 (×)</td>
<td>1.0000 (✓)</td>
<td>1.0000 (✓)</td>
</tr>
<tr>
<td>The Proposed*</td>
<td>0.9951</td>
<td>0.8789</td>
<td>0.9346 (✓)</td>
<td>0.4646 (×)</td>
<td>1.0000 (✓)</td>
</tr>
</tbody>
</table>

Table IV: Robustness against voice cloning attacks with regular and harmful preprocessing.

Table V: Robustness against watermark overwriting attacks. # indicates that the attacker extracts the watermark by his own extractor. * indicates that the attacker trains his own embedding and extraction models guided by the proposed method.

![Fig. 8: The performance of our method against voice cloning attacks with different watermark ratios. Black dots and white dots indicate mean accuracy and median accuracy, respectively. Green triangles represent the average SECS values of synthesized speech.](image-url)
As Identifying Watermark Location and then Removing It. The effect of VAE reconstruction on the audio signal. The obtained directly from both the masked spectrogram (Spec-ACC) and Wave-ACC. Based on experimental results, we find that masking at the medium-to-low frequency range (i.e., Position 2 and 3) has the most significant impact on the watermark extraction ability, and conjecture the watermark is primarily embedded in the medium-to-low frequency range.

Therefore, we conduct an adaptive attack: masking such frequency range as preprocessing before voice cloning, and then adopting VITS [30] to launch the attack. The results prove that this adaptive attack still cannot defeat our method: watermark extraction accuracy from synthesized speech is 100% (masking position2) and 92.14% (masking position3). Fig. 9-c shows the impact of masking at different locations on the audio quality (SNR, PESQ, SECS). We find that masking at lower frequencies has a more severe effect on the audio quality, and masking at Positions 2 and 3 has a particularly significant impact. As the masked frequency range increases, the audio quality improves. To remove watermarks strongly, we try masking more frequency bands of the spectrogram from high to low frequencies. We gradually increase the percentage of the mask from the top of the spectrogram, since masking low frequencies introduces a very serious quality loss. As shown in Fig. 10, as the mask ratio increases from 10% to 90%, although the ACC degrades from 100% to near 70%, the audio quality is also destroyed by a large margin (SECS from 1.0 to 0.1). For a more comprehensive evaluation, we further adopt 5% and 20% masking ratios. The corresponding results are shown in Fig. 18 of Appx. C, which are consistent with the results with a 10% masking ratio.

Voice Cloning Attack with Public Extractor. Because the extractor is publicly released by the platform, the attacker can attempt to generate audios to bypass it. In practice, the extractor may be packaged as an API, and the attacker only knows whether the generated audio deceives the extractor. To simulate this scenario, the attacker can first train a binary classifier to distinguish the watermarked and unwatermarked data. With the well-trained classifier, he is able to train his TTS model constrained by a domain adversarial loss, which forces the model to output unwatermarked data. We find such a strategy (i.e., domain-adversarial training) will degrade the performance of the TTS model (SECS from 0.9014 to 0.8844), while our method can still achieve 100% extraction accuracy.

Combining Multiple Attack Strategies. We further consider combining different attack strategies to destroy the proposed method. This entails the integration of diverse attack schemes, encompassing regular preprocessing, harmful preprocessing, domain-adversarial training, VAE reconstruction and watermark overwriting. In a nutshell, more severe attack strategies will further destroy the utility of voice cloning, while the proposed method is still somewhat effective. For example, taking resampling 16 kHz as pre-processing and MP3 compression 16 kbps as post-processing, compared with only pre-processing, ACC suffers a slight degradation (ACC:100% → 99.94%) but the quality degrades by a large margin (SECS: 1.000 → 0.8575). The specific experimental results can be found on the paper’s website.

---

3https://freesound.org/
4https://sound-effects.bbcrewind.co.uk/search
F. Practicality in Real-world Services

In practice, the attacker may directly leverage real-world services to conduct voice cloning attacks in an efficient and convenient way. Specifically, he can collect a small number of the target speaker’s voice samples and upload them to real-world services to implement customized voice cloning based on their instructions. When the collected voice samples have been watermarked by our method, we are able to extract the watermark information from the synthetic voice. We test both real-world TTS service and voice conversion service.

Real-world TTS Service. We employ two widely-used speech synthesis tools, PaddleSpeech [71] and Voice-Cloning-App [27], to conduct voice cloning attacks in a black-box way. For PaddleSpeech, we use the tool through the Baidu Paddle AI Studio platform [26]. The attacker can easily fine-tune the speech synthesis model and customize the cloned voice of the target speaker with just a few clicks. For Voice-Cloning-App [27], we use the released executable software provided by the author, following the software’s documentation for voice cloning. For both services, the attacker only needs 10 segments of the target speaker’s voice samples and upload them to real-world service within 10 seconds. Additionally, we follow the work [16] by using 10 segments of texts as input for the synthesized speech (listed in Table XII and Table XIII in the Appendix). We extract watermarks from the synthesized speech and verify the extraction accuracy as well as the quality of the synthesized speech. For comprehensive evaluation, we test multiple speakers (p225 ~ p230, the first six speakers) from the VCTK dataset [72]. To explore the efficacy of our approach in different languages, we also select the first six speakers (D4, D6, D7, D8, D11, D12) from the THCHS30 [73] test set and test the more challenging Chinese voice cloning scenario using the PaddleSpeech tool, an open source tool that additionally supports Chinese besides English.

Table VI presents the evaluation results. Specifically, for PaddleSpeech [71], the quality of the synthesized voice is considerably superior. We can achieve an exceptional watermark extraction accuracy (ACC = 100%). For Voice-Cloning-App [27], although the quality of the synthesized voice is inferior, the watermark extraction accuracy is still maintained at a satisfactory level (ACC >= 90%), which further substantiates the outstanding effectiveness of the proposed watermarking methodology. The synthesized speech samples from these services can be found on our project website.

Real-world Voice Conversion Service. Very recently, the voice conversion service so-vits-svc [28] becomes increasingly popular for singing song synthesis. Thus, we also try our method in such a scenario and validate its effectiveness. Specifically, we use the first 30 singing voices (around 5s per voice) from the Opencpop dataset [74], embed watermarks in them, and then use these voices to train so-vits-svc according to the instruction document. After that, we use the trained model to perform voice conversion on a song. Here, we use the original song “Right Here Waiting” from Richard Marx [75], and use Ultimate Vocal Remover [76] to remove the background music. Then, we divide it into 24 segments (10s per segment) and perform the voice conversion for each segment. The vocal data are available on our website. The final extraction accuracy of each segment of the synthesized vocals is 100%, which further proves the effectiveness of our method in real speech cloning scenarios.

G. Ablation Study

Influence of Watermark Bits. To investigate the impact of the watermark length, we attempt to embed bit strings of
TABLE VII: Robustness comparison between Distortion-Blind Watermarking Model (DBWM) and Full Model against different voice cloning attacks. The red, blue, and gray areas represent professional, regular, and low-quality voice cloning attacks, respectively. The quality of synthesized speech is also provided.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Hifi-GAN [40]</td>
<td>Griffin-Lim [38]</td>
<td>Hifi-GAN [40]</td>
<td>Griffin-Lim [38]</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1.0238</td>
<td>0.7075</td>
<td>1.0765</td>
<td>1.0000</td>
<td>0.7152</td>
<td>0.6854</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.8956</td>
<td>0.7152</td>
<td>1.0765</td>
<td>1.0000</td>
<td>0.6508</td>
<td>0.6136</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.6508</td>
<td>0.6136</td>
<td>0.6508</td>
<td>1.0000</td>
<td>0.6854</td>
<td>0.6108</td>
</tr>
</tbody>
</table>

Fig. 13: Fidelity comparison between Full Model and the Distortion-Blind Watermarking Model (DBWM). Green triangles represent the mean values and red lines indicate the median values.

Fig. 12 further shows the watermark extraction accuracy ACC under the voice cloning attacks. As the number of embedded bits increases, ACC still keeps near 100%, except when the former part includes dynamic information and the latter part includes dynamic information. We watermark the original audio with two watermarks, whose former parts are the same but their latter parts can have different watermarks, but we still want to extract the ownership information from the synthesized audio. To achieve this, we split the watermark bit string into two parts, where the former part includes static information and the latter part includes dynamic information. We watermark the original audio with two watermarks, whose former parts are the same while the latter parts are randomly different. Then, we adopt VITS for voice cloning attacks. As expected, we can extract the former static information with a 100% ACC. Similarly, we can easily replace version information with platform information to address the issue of multiple platforms. In Appx. A, we provide more discussion on comparison with passive detection and the extension to the physical world.

TABLE VIII: Comparison with traditional audio watermarking methods against voice cloning attacks. ‘syn’ indicates metrics related to synthesized speech quality, while ‘wm’ represents metrics related to watermark embedding and extraction.

<table>
<thead>
<tr>
<th>Method</th>
<th>syn PESQ↑</th>
<th>syn SECS↑</th>
<th>wm SNR↑</th>
<th>wm ACC↑</th>
</tr>
</thead>
<tbody>
<tr>
<td>FSVC [23]</td>
<td>0.9949</td>
<td>0.9139</td>
<td>21.1282</td>
<td>0.5554</td>
</tr>
<tr>
<td>RFDLM [21]</td>
<td>1.0303</td>
<td>0.9179</td>
<td>19.4668</td>
<td>0.5096</td>
</tr>
<tr>
<td>The Proposed</td>
<td>1.0342</td>
<td>0.9085</td>
<td>28.1650</td>
<td>1.0000</td>
</tr>
</tbody>
</table>

VI. DISCUSSION

A. Comparison with Existing Audio Watermarking Methods

To comprehensively showcase the efficacy of the proposed scheme, we conducted a comparative analysis with existing watermarking schemes. We use RFDLM [21], FSVC [23], and the proposed method to embed watermarks on the speech respectively, and then extract the watermarks from the synthesized speech based on the TTS model trained on this speech data respectively (VITS [30] is utilized as the default voice cloning attack model). As shown in Table VIII, the existing audio watermarking schemes cannot resist the TTS model learning process and the watermarks cannot be retained in the synthesized speech, while the proposed scheme is better at retaining the watermark information.

B. Embedding Different Watermarks for Version Information

In practice, the watermark may contain not only static information like names but also dynamic information like versions. In such a scenario, the collected audio by the attacker may have different watermarks, but we still want to extract the ownership information from the synthesized audio. To achieve this, we split the watermark bit string into two parts, where the former part includes static information and the latter part includes dynamic information. We watermark the original audio with two watermarks, whose former parts are the same while the latter parts are randomly different. Then, we adopt VITS for voice cloning attacks. As expected, we can extract the former static information with a 100% ACC. Similarly, we can easily replace version information with platform information to address the issue of multiple platforms. In Appx. A, we provide more discussion on comparison with passive detection and the extension to the physical world.

VII. CONCLUSION

In the era of the Ear Economy, it is necessary to establish safeguards against the potential misuse of voice cloning technology. To this end, we propose a novel concept of “Timbre Watermarking”, based on which we design an end-to-end voice cloning detection framework. In this framework, we
tailor the watermark encoding with a repeated embedding strategy to obtain the inherent robustness against distortions in the time domain. In addition, we investigate different voice cloning attacks and find their shared process, i.e., normalization distortion, transformation distortion, and wave reconstruction distortion. Then, we incorporate them as a distortion layer into our framework to acquire generalization across different voice cloning attacks. Extensive experiments demonstrate that the proposed “Timbre Watermarking” can achieve high robustness against common speech preprocessing distortions such as cropping, while also withstanding various voice cloning attacks and maintaining usability in real-world services such as PaddleSpeech, Voice-Cloning-App, and so-vits-svc. Moreover, we conduct ablation studies to explore the influence of watermark bits and the distortion layer. We expect our framework can shed some light on the future research of voice cloning detection and timbre protection.

ACKNOWLEDGEMENT

We thank anonymous reviewers for their constructive feedback. This work was supported in part by the Natural Science Foundation of China under Grant 62072421, 62002334, 62102386, 62121002, U20B2047 and Singapore Ministry of Education (MOE) AcRF Tier 2 MOE-T2EP20121-0006.

REFERENCES


APPENDIX

A. More Discussion

1) Comparison with Passive Detection: As mentioned above, passive detection usually cannot generalize to unseen synthesis methods. Two classic detection methods have been proposed to address the generalization limitation of data distribution and synthesis methods, namely, Void [12] and One-Class [42]. We compare our method with these two detection approaches in terms of generalization against different voice cloning attacks. We follow the released unofficial code\(^5\) and official code\(^6\) to reproduce these passive detection solutions.

Specifically, we measure the detection performance of these methods against different voice cloning attacks, where each attack generates 500 synthesized audios, respectively. For a fair comparison, we first divide the suspicious audio into 5 segments and set the accuracy threshold to 90% for verification. We claim the detection is successful when all segments pass the verification. As shown in Fig. 15, our method performs well in all cases, while two passive detection methods have poor performance ($AUC < 0.5$) when facing synthesized audios generated by VITS [30].

2) Extension to the Physical World: In all the above experiments on voice cloning attacks, we consider that we can obtain suspicious synthesized speech in the digital world, even if the speech may be processed by some operations like MP3 compression. However, in some scenarios, we cannot directly obtain the digital version of the suspicious speech. Instead, we can only record the target speech for subsequent verification, which can be further distorted after the digital-physical-digital transformation. Fortunately, there are some works on audio watermarking and adversarial examples [77], [78], [24], which have demonstrated their effectiveness against such air-channel transmission distortion. A feasible way is to integrate them into the distortion layer to enhance robustness. We will explore this direction in future work.

B. More Exploration on Robustness

1) Robustness Against Cropping with Different Watermark Bits: We further investigate the robustness of cropping processing to various watermark bit embedding quantities. As illustrated in Fig. 19, there exists a trade-off between robustness and embedding capacity. Besides, an example of a 50% cropping with different strategies are shown in Fig. 15.

2) Robustness of DBWM: We further investigate the robustness of DBWM in relation to common audio processing. As illustrated in Table IX, without the distortion layer during training, the performance degrades in many cases. In addition, we also investigate where watermarks are embedded by DBWM. We conduct identical spectrogram masking experiments, as illustrated in Fig. 16. The results reveal that DBWM predominantly embeds information within higher frequencies.

\(^{5}\)https://github.com/chislab/void-voice-liveness-detection

\(^{6}\)https://github.com/zyyouzhang/AIR-ASVSpoof

<table>
<thead>
<tr>
<th>PreprocessingParameter</th>
<th>Quality</th>
<th>ACC(^\dagger)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resampling 16 kHz</td>
<td>34.6674</td>
<td>0.9989</td>
</tr>
<tr>
<td>8 kHz</td>
<td>16.6782</td>
<td>0.9169</td>
</tr>
<tr>
<td>Amplitude Scaling 20%</td>
<td>1.9382</td>
<td>1.0000</td>
</tr>
<tr>
<td>40%</td>
<td>4.4368</td>
<td>0.9609</td>
</tr>
<tr>
<td>60%</td>
<td>7.9589</td>
<td>1.0000</td>
</tr>
<tr>
<td>80%</td>
<td>13.9790</td>
<td>0.9944</td>
</tr>
<tr>
<td>Low Pass Filtering 80 dB</td>
<td>8.8171</td>
<td>0.9591</td>
</tr>
<tr>
<td>High Pass Filtering 500 Hz</td>
<td>12.8881</td>
<td>0.9337</td>
</tr>
<tr>
<td>MP3 Compression 64 kbps</td>
<td>15.0170</td>
<td>0.9629</td>
</tr>
<tr>
<td>48 kbps</td>
<td>20.6921</td>
<td>1.0000</td>
</tr>
<tr>
<td>32 kbps</td>
<td>24.7004</td>
<td>0.9999</td>
</tr>
<tr>
<td>40 kbps</td>
<td>26.3839</td>
<td>1.0000</td>
</tr>
<tr>
<td>48 kbps</td>
<td>27.9337</td>
<td>1.0000</td>
</tr>
<tr>
<td>24 kbps</td>
<td>27.7933</td>
<td>1.0000</td>
</tr>
<tr>
<td>16 kbps</td>
<td>28.7933</td>
<td>1.0000</td>
</tr>
<tr>
<td>8 kbps</td>
<td>29.7933</td>
<td>1.0000</td>
</tr>
<tr>
<td>Gaussian Noise 80%</td>
<td>13.9790</td>
<td>1.0000</td>
</tr>
<tr>
<td>40%</td>
<td>4.4368</td>
<td>0.9609</td>
</tr>
<tr>
<td>20%</td>
<td>1.9382</td>
<td>1.0000</td>
</tr>
<tr>
<td>Recount 8 bps</td>
<td>22.8747</td>
<td>0.9041</td>
</tr>
<tr>
<td>Low Pass Filtering 2000 Hz</td>
<td>12.4747</td>
<td>0.7925</td>
</tr>
<tr>
<td>High Pass Filtering 500 Hz</td>
<td>3.7917</td>
<td>1.0000</td>
</tr>
<tr>
<td>20 dB</td>
<td>0.9090</td>
<td>0.9737</td>
</tr>
<tr>
<td>35 Samples 80%</td>
<td>3.1933</td>
<td>0.7701</td>
</tr>
<tr>
<td>35 Samples 40%</td>
<td>3.1933</td>
<td>0.7701</td>
</tr>
<tr>
<td>35 Samples 20%</td>
<td>1.9382</td>
<td>1.0000</td>
</tr>
<tr>
<td>35 Samples 40%</td>
<td>1.9382</td>
<td>1.0000</td>
</tr>
<tr>
<td>35 Samples 20%</td>
<td>1.9382</td>
<td>1.0000</td>
</tr>
<tr>
<td>35 Samples 40%</td>
<td>1.9382</td>
<td>1.0000</td>
</tr>
<tr>
<td>35 Samples 20%</td>
<td>1.9382</td>
<td>1.0000</td>
</tr>
<tr>
<td>35 Samples 40%</td>
<td>1.9382</td>
<td>1.0000</td>
</tr>
<tr>
<td>35 Samples 20%</td>
<td>1.9382</td>
<td>1.0000</td>
</tr>
<tr>
<td>35 Samples 40%</td>
<td>1.9382</td>
<td>1.0000</td>
</tr>
</tbody>
</table>

TABLE X: The comparison of robustness against watermark overwriting attacks between the default strategy and the overwriting-resilient strategy.

<table>
<thead>
<tr>
<th>Method</th>
<th>PESQ(\dagger)</th>
<th>SECS(\dagger)</th>
<th>wm1-ACC(\dagger)</th>
<th>wm2-ACC(\dagger)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overwriting-resilient</td>
<td>1.0269</td>
<td>0.9145</td>
<td>1.0000 ($\checkmark$)</td>
<td>0.4000 ($\times$)</td>
</tr>
<tr>
<td>Default</td>
<td>1.0000</td>
<td>0.9081</td>
<td>0.4000 ($\checkmark$)</td>
<td>1.0000 ($\checkmark$)</td>
</tr>
</tbody>
</table>

3) Countermeasure Against Watermark Overwriting Attacks: As shown in Table V, an adaptive attacker with access to the watermark encoder API can successfully conduct a self-overwriting attack using the original model. To address it, we design a mechanism to resist watermark overwriting of insider malicious users. We design a weighted embedding process, which is controlled by the subsequent watermark decoder, and add the watermark overwriting distortion in the original distortion layer to fine-tune the model. The final enhanced model can resist the re-watermark attack (ACC: 100%). The model structure is shown in Fig. 21, which is composed of 4 steps: ① Extract the weight factor $S$ from the audio to be watermarked; ② Use factor $S$ to weight the watermark features in the watermark embedding network; ③ Input the watermarked audio and a randomly generated second watermark into the watermark embedding network; ④ Input the watermarked audio into the distortion layer and proceed to subsequent watermark extraction and network parameter optimization. We adopt VITS as the Voice cloning model. The watermark extraction accuracy (ACC) and the quality of all 500 synthesized speech are shown in Table X. Experimental results show that the overwriting-resilient strategy can resist watermark overwriting attacks even by internal attackers.

C. More Ablation Studies

The Influence of Different Masking Ratios. See Fig. 18.

D. Details of Network Architectures

Table XI shows the structure of the attacker’s watermark overwriting model, where ReluBlock contains a convolutional layer, an InstanceNorm, and the LeakyReLU activation function.
Fig. 14: Comparison with state-of-the-art passive detection methods (Void [12] and One-Class [42]). ROC curve against different voice cloning attacks are displayed.

Fig. 15: An example of a 50% cropping with three strategies.

Fig. 16: The watermark extraction accuracy of DBWM when different frequency bands of the spectrogram were masked.

Fig. 17: The performance of our method against voice cloning attacks with (a) / without (b) VAE reconstruction preprocessing. Black dots and white dots indicate mean value and median value, respectively.

Fig. 18: Watermark extraction accuracy and speech quality under 5% and 20% masking ratios.

<table>
<thead>
<tr>
<th>Groups</th>
<th>Input channel/dim</th>
<th>Output channel/dim</th>
</tr>
</thead>
<tbody>
<tr>
<td>Watermark Encoder</td>
<td>Linear + LeakyReLU</td>
<td>Wm_length</td>
</tr>
<tr>
<td>Carrier Encoder</td>
<td>ReLUBlock</td>
<td>64</td>
</tr>
<tr>
<td>Watermark Embedder</td>
<td>ReLUBlock</td>
<td>64</td>
</tr>
<tr>
<td>Watermark Extractor</td>
<td>ReLUBlock</td>
<td>64</td>
</tr>
<tr>
<td>Watermark Decoder</td>
<td>Linear</td>
<td>513</td>
</tr>
<tr>
<td>Discriminator</td>
<td>ReLUBlock</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>ReLUBlock</td>
<td>32</td>
</tr>
<tr>
<td></td>
<td>Linear</td>
<td>64</td>
</tr>
</tbody>
</table>
E. Phrases for Synthesis

Table XII and Table XIII list the English and Chinese phrases, respectively, used for synthetic speech generation in real-life scenarios using voice cloning tools.

![Images of signal processing with watermarks](image)

**Fig. 20:** The influence of VAE reconstruction on watermarked speech and cloned speech.

**Fig. 19:** Cropping robustness of the proposed method with different watermarking bits.

**Fig. 21:** Overview of overwriting-resilient timbre watermark framework.

**TABLE XII:** English phrases used for real-world services in Sec. V-F.

<table>
<thead>
<tr>
<th>No.</th>
<th>English Phrases</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>The difference in the rainbow depends considerably upon the size of the drops, and the width of the colored band increases as the size of the drops increases.</td>
</tr>
<tr>
<td>8</td>
<td>If the rain falls upon the green of the grass, the result is a bow with an abnormally wide yellow band, since red and green light when mixed form yellow.</td>
</tr>
<tr>
<td>9</td>
<td>This is a very common type of bow, one showing mainly red and yellow, with little or no green or blue.</td>
</tr>
</tbody>
</table>

**TABLE XIII:** Chinese phrases used for PaddleSpeech [26] in Sec. V-F.

<table>
<thead>
<tr>
<th>No.</th>
<th>Chinese Phrases</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>发光的，根据裂纹，挂着晶莹的水珠。</td>
</tr>
<tr>
<td>11</td>
<td>风雨同舟，结伴而行。</td>
</tr>
<tr>
<td>12</td>
<td>他走下阶梯，拿起电话，打电话，打电报。</td>
</tr>
<tr>
<td>13</td>
<td>他走下阶梯，拿起电话，打电话，打电报。</td>
</tr>
<tr>
<td>14</td>
<td>他走下阶梯，拿起电话，打电话，打电报。</td>
</tr>
<tr>
<td>15</td>
<td>他走下阶梯，拿起电话，打电话，打电报。</td>
</tr>
<tr>
<td>16</td>
<td>他走下阶梯，拿起电话，打电话，打电报。</td>
</tr>
</tbody>
</table>