TriniMark: A Robust Generative Speech Watermarking Method for Trinity-Level Attribution

Yue Li, Weizhi Liu, and Dongdong Lin

Abstract—The emergence of diffusion models has facilitated the generation of speech with reinforced fidelity and naturalness. While deepfake detection technologies have manifested the ability to identify AI-generated content, their efficacy decreases as generative models become increasingly sophisticated. Furthermore, current research in the field has not adequately addressed the necessity for robust watermarking to safeguard the intellectual property rights associated with synthetic speech and generative models. To remedy this deficiency, we propose a robust generative speech watermarking method (TriniMark) for authenticating the generated content and safeguarding the copyrights by enabling the traceability of the diffusion model. We first design a structure-lightweight watermark encoder that embeds watermarks into the time-domain features of speech and reconstructs the waveform directly. A temporal-aware gated convolutional network is meticulously designed in the watermark decoder for bit-wise watermark recovery. Subsequently, the waveform-guided fine-tuning strategy is proposed for fine-tuning the diffusion model, which leverages the transferability of watermarks and enables the diffusion model to incorporate watermark knowledge effectively. When an attacker trains a surrogate model using the outputs of the target model, the embedded watermark can still be learned by the surrogate model and be correctly extracted. Comparative experiments with state-of-the-art methods demonstrate the superior robustness of our method, particularly in countering compound attacks.

Index Terms—Generative watermarking, speech watermarking, trinity attribution, diffusion models.

I. INTRODUCTION

GENERATIVE artificial intelligence (GenAI) has significantly advanced Artificial Intelligence-Generated Content (AIGC) technologies, garnering increasing attention and widespread favor for generative models. Amid this surge of research enthusiasm, generative models, led by Generative Adversarial Networks (GANs) [1] and Diffusion Models (DMs) [2]–[4], have demonstrated impressive performances, further diminishing the distinction between generated content and natural content. However, the other side of AIGC technology conceals numerous security risks. Infringement on models, malicious forgery of generated content, and attacks on users' data have already become significant risks in the era of GenAI. In such an environment, the demand for sustainable technological solutions to guarantee these security measures has become even more exigent.

Watermarking, as a proactive technology for content and model identification, has become a crucial cornerstone in

Yue Li, and Weizhi Liu are with the College of Computer Science and Technology, National Huaqiao University, Xiamen 361021, China, and also with the Xiamen Key Laboratory of Data Security and Blockchain Technology, Xiamen 361021, China (e-mail: liyue_0119@hqu.edu.cn; lwzzz@stu.hqu.edu.cn.
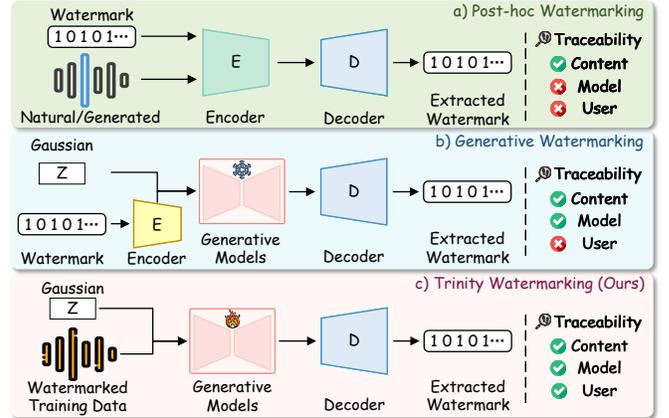

Fig. 1. Different types of watermarking methods and their traceability functions. Post-hoc watermarking methods embed watermarks directly into speech and can only support content-level traceability. Generative watermarking guides the model in synthesizing watermarked speech through watermark features, enabling traceability at both the model and content levels. In contrast, our method first performs pre-embedding of watermarks into the training data, thereby guiding the model to generate watermarked content, which achieves the trinity traceability.

AIGC security. For safeguarding the intellectual property rights of generative models, one solution is to leverage the transferability of watermarks [5]–[10]. These methods involve training or fine-tuning the generative models to root watermarks within them, enabling traceability back to the original model. Methodologies for protecting the authenticity of generated content encompass two distinct solutions. The first paradigm involves integrating watermarks into content through deep neural networks [11]–[15]. Another approach incorporates watermarking within the generative process of models, facilitating the simultaneous production of watermarked content [16]–[20]. The aforementioned technical approaches share a common objective: establishing verifiable traceability through watermarking to ensure and validate the authenticity of content.

Although watermarking methods have reached a relatively mature stage in the image modality, developing watermarking techniques within the speech domain still demands further enhancement and progress. Within the extant speech watermarking approaches, two zero-watermarking techniques address the challenge of protecting generative models' copyright [21], [22]. These techniques employ a classifier to detect the presence or absence of a watermark, subsequently enabling the determination of model ownership attribution. Furthermore, for VALL-E [23] and MusicGen [24], TraceableSpeech [25]

and LatentWM [26] respectively leverage the transferability of watermarks by fine-tuning the generative models, thereby effectively enabling traceability. Although numerous watermarking techniques have emerged within post-hoc methods in the field of authenticity protection [27]–[31], the Groot [32] stands as the only solution that effectively fulfills dual key objectives: simultaneously safeguarding the generative models copyright and proactively maintaining supervisory over generated content.

Within the evolving ecosystem of speech watermarking techniques, several observations have emerged: (1) Post-hoc watermarking methods are fundamentally limited to content-level traceability, lacking the capability to trace the generative model. In addition, zero-watermarking methods can only trace the generative model but cannot trace the content. (2) There are some generative watermarking approaches are only capable of tracing a single generative model, failing to generalize across multiple models. (3) Some watermarking methods lack the flexibility to handle arbitrary watermark information, meaning that the generative model must be retrained for different messages, significantly increasing the computational overhead of the watermarking approach.

To tackle these challenges, we propose a generative speech watermarking method based on fine-grained feature transfer, TriniMark, which enables trinity traceability at the content, model, and user level, as depicted in Fig 1. The proposed method consists of a two-stage training process. The first stage is to pretrain the watermark encoder-decoder, during which a time-domain-aware speech watermarking model is designed to perform post-processing watermark embedding on speech content. The second stage involves fine-tuning the diffusion model. To this end, we further propose a waveform-guided fine-tuning strategy. By pre-embedding watermarks into the training data using the pretrain encoder, the diffusion model is jointly fine-tuned with the pretrain decoder, thereby achieving a better balance between the quality of the watermarked speech and the accuracy of watermark extraction.

In a nutshell, our contribution can be summarized as:

- We explore the transition from post-hoc watermarking to generative watermarking, and propose a generative speech watermarking based on fine-grained feature transfer, enabling trinity traceability at content, model, and user levels.
- We designed a structure-lightweight watermark encoder to perform post-hoc embedding on speech. For high-precision watermark recovery, we meticulously designed a watermark decoder based on a temporal-aware gated convolutional network.
- We further propose a waveform-guided fine-tuning strategy, which jointly fine-tunes the diffusion model utilizing the pretrained encoder-decoder. This strategy enables flexible adaptation to different watermark information with only a single round of training, effectively supporting user-level traceability.
- Comprehensive experiments demonstrate that our Trini-Mark can maintain desirable watermarked speech quality even at a high capacity of 500 bps. Moreover, comparisons with state-of-the-art methods confirm its superior

robustness against both individual and compound attacks.

II. RELATED WORK

A. Deep Learning-based Watermarking

With the powerful representation learning capabilities of deep learning, it has been widely used in watermarking. Due to its ability to extract more fine-grained features, the Short-Time Fourier Transform (STFT) is often the preferred choice for frequency-domain features. Therefore, Pavlović et al. [33] pioneered the use of an Encoder-Decoder framework for speech watermarking. Based on Pavlović's network structure, O'Reilly et al. [34] applied the Transformer [35] in combination with a multiplicative spectrogram mask for watermark embedding. Afterward, Chen et al. [27] improved the performance of watermarking by employing invertible neural networks (INNs). To counteract voice cloning, Liu et al. [29] embedded repeated watermarks into the magnitude features of the STFT. Moreover, recognizing the superior robustness of Discrete Wavelet Transform (DWT), Liu et al. [36] employ the detail coefficients of DWT as the embedding space to combat audio re-recording attacks. Although the frequency domain is a popular choice for speech watermarking, there are works that use the temporal domain, which contains richer auditory features, for watermark embedding. Qu et al. [37] and Roman et al. [28] employ time-domain-based watermarking for creating accessible quick response (QR) codes for visually impaired individuals and detecting AI-generated speech and localizing watermarks, respectively. The purpose of these methods is solely to protect the copyright of content, whether it is natural or synthetic speech. However, the proposed method can distinguish between natural and generated content, protect the copyright of the model, and authenticate synthetic speech synchronously. Moreover, the time-domain-based watermarking we proposed, efficiently and robustly embeds and extracts watermarks using deliberated designed encoders and decoders.

B. Generative Watermarking

With the continuous improvement in generative models' generation capabilities, generative watermarking (or steganography) development has also been promoted. Concretely, Chen et al. [38] pioneered embedding the secret message into the probability distribution of speech using adaptive arithmetic decoding (AAD) based on the autoregressive models, enabling the generative model to produce stego speech according to the embedded distribution. Similarly, Ding et al. [39] designed a distribution copy method based on autoregressive models, where the secret message determines from which distribution copy sampling is performed. Leveraging the reversibility of flow-based models, Chen et al. [38] used rejection sampling to map the secret message into the input of models, generating stego speech. Based on GAN-based vocoders, Li et al. [40] employed secret audio as the input to the model, thereby generating stego audio. In addition, based on the architecture of VALL-E [23], Zhou et al. [25] proposed TraceableSpeech, which initially trained the Encodec in VALL-E to learn the watermark, and then fine-tuned VALL-E with the pretrained

Encodec to generate watermarked speech. Although TraceableSpeech also aims to protect generative models, it is limited to TTS models that utilize neural encoder-decoder structures and cannot generalize to generative models like DDPM-based vocoders. Additionally, most generative steganography methods assume a lossless channel, which makes them less robust for real-world applications.

C. Text-to-Speech Diffusion Models

In recent years, diffusion models have demonstrated superior performance in Text-to-Speech (TTS) synthesis. Existing TTS models can be broadly classified into acoustic models, vocoders, and end-to-end models. Acoustic models aim to generate mel-spectrograms from text input. Popov et al. [41] first proposed Grad-TTS, the diffusion model for mel-spectrogram generation. Huang et al. [42] designed Prodiff, which aims to accelerate the sampling steps by incorporating knowledge distillation into the denoising process. Afterward, Chen et al. [43] reduced the model parameters by designing a novel lightweight diffusion decoder. Furthermore, vocoders are used to generate waveforms from mel-spectrograms. Based on score matching and DDPM, Zhang et al. [44] pioneeringly proposed WaveGrad. Kong et al. [45] designed DiffWave, a conditional and unconditional diffusion model that accelerates waveform sampling using a six-step sampling method. Meanwhile, Lee et al. [46] developed PriorGrad, a diffusion model that enhances waveform quality by utilizing an adaptive prior strategy. What's more, end-to-end models eliminate intermediate conversion steps, allowing for the generation of waveforms from text input. Huang et al. [47] designed a conditional diffusion model, FastDiff, which generates waveforms using adaptive hidden sequences. Moreover, Ju et al. [48] employed a factorized codec to obtain fine-grained speech discrete representations and then achieved text-to-speech synthesis using the factorized diffusion models. Our proposed TriniMark focuses on fine-tuning DDPM-based vocoders to protect the copyrights of these generative models.

III. PRELIMINARIES

The proposed TriniMark leverages DDPM-based vocoders to generate the watermarked speech and the blurb of DDPM about speech generation is described as follows.

In the diffusion process of DDPM, given natural speech $\mathbf{s}_0 \sim q_{data}(\mathbf{s}_0)$, the latent variable $\mathbf{s}_t$ is obtained by adding noise to it step by step, which follows the standard Gaussian distribution. This process follows a Markov chain:

$$q(\mathbf{s}_t|\mathbf{s}_{t-1}) = \mathcal{N}(\mathbf{s}_t; \sqrt{1-\beta_t}\mathbf{s}_{t-1}, \beta_t\mathbf{I}), \tag{1}$$

$$q(\mathbf{s}_{1:T}|\mathbf{s}_0) = \prod_{t=1}^{T} q(\mathbf{s}_t|\mathbf{s}_{t-1}), \tag{2}$$

where $\beta_t \in (0,1)$ is the variance scheduled at time step t, and $\mathbf{I}$ is an identity matrix. Let $\alpha_t = 1 - \beta_t$, $\overline{\alpha}_t = \prod_{i=1}^{t} \alpha_i$ and $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. For any time step of t, by re-parameterization, the latent variable $\mathbf{s}_t$ can only be calculated from $\mathbf{s}_0$ and α_t:

$$\mathbf{s}_t = \sqrt{\overline{\alpha}_t}\mathbf{s}_0 + \sqrt{1-\overline{\alpha}_t}\epsilon. \tag{3}$$

Therefore, the final diffusion process can be simplified to a single step. It can be represented as:

$$q(\mathbf{s}_t|\mathbf{s}_0) = \mathcal{N}(\mathbf{s}_t; \sqrt{\overline{\alpha}_t}\mathbf{s}_0, (1-\overline{\alpha}_t)\mathbf{I}). \tag{4}$$

The denoising process involves removing the noise from latent variable $\mathbf{s}_t$ by employing the prediction network ϵ_θ to estimate the noise added during the diffusion process step by step. This process $q(\mathbf{s}_{t-1}|\mathbf{s}_t, \mathbf{s}_0)$ also belongs to Gaussian distributed so that it can be computed as:

$$q(\mathbf{s}_{t-1}|\mathbf{s}_t, \mathbf{s}_0) = \frac{q(\mathbf{s}_t|\mathbf{s}_{t-1})q(\mathbf{s}_{t-1}|\mathbf{s}_0)}{q(\mathbf{s}_t|\mathbf{s}_0)} \tag{5}$$

According to Bayes' theorem. Unfold this equation with Eq. 1 and combine like terms, we get

$$q(\mathbf{s}_{t-1}|\mathbf{s}_t) = \mathcal{N}\Big(\mathbf{s}_{t-1}; \frac{1}{\sqrt{\alpha_t}}(\mathbf{s}_t - \frac{1-\alpha_t}{\sqrt{1-\overline{\alpha}_t}}\epsilon), (\frac{1-\overline{\alpha}_{t-1}}{1-\overline{\alpha}_t}\beta_t)\mathbf{I}\Big). \tag{6}$$

Once the prediction network ϵ_θ has been trained well to predict noise ϵ, the estimated speech can be obtained by ϵ_θ:

$$\mathbf{s}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{s}_t - \frac{1-\alpha_t}{\sqrt{1-\overline{\alpha}_t}}\epsilon_\theta(\mathbf{s}_t, t, c)\right) + \delta_t\mathbf{z}, \tag{7}$$

where $\delta_t\mathbf{z}$ denotes the random noise, $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, and c is the mel-spectrogram.

The training of the prediction network aims to fit the noise ϵ. Thus, the parameters θ need to be continuously learned and updated by maximizing the variational lower bound. Therefore, the final objective function can be simplified as:

$$\mathcal{L}_{simple} = \mathbb{E}_{t,\mathbf{s}_0,\epsilon}\big[||\epsilon - \epsilon_\theta(\sqrt{\overline{\alpha}_t}\mathbf{s}_0 + \sqrt{1-\overline{\alpha}_t}\epsilon, t)||^2\big]. \tag{8}$$

IV. PROPOSED METHOD

The proposed TriniMark differs from post-hoc watermarking and model watermarking approaches. Instead of embedding watermarks into the generated speech or the model parameters, we leverage watermark transfer learning to learn from watermarked training data, thereby generating watermarked speech via DMs, as depicted in Fig. 2. The entire generative watermarking consists of two stages. The first stage is pre-training the watermark encoder and decoder. The second stage involves fine-tuning the diffusion models utilizing the pretrained encoder and decoder with watermark priors. The detailed process will be described in the following sections.

A. Pre-training Watermark Encoder and Decoder

1) Architecture of Watermark Encoder and Decoder: We meticulously design a structure-lightweight watermark encoder $\mathcal{E}(\cdot)$ and a decoder $\mathcal{D}(\cdot)$ to learn watermark priors, as illustrated in Fig. 3, ensuring its transferability to the diffusion model. *The watermark encoder* consists of a DenseBlock (DB) and a structure-lightweight Speech Reconstruction Network (SRNet). The DB includes two fully connected (FC) layers and a ReLU activation. The SRNet contains a downsampling block and an upsampling block. The downsampling block consists of four 1D convolutional layers, while the upsampling block comprises four 1D convolutional layers and four 1D transposed convolutional layers. The lightweight structure of
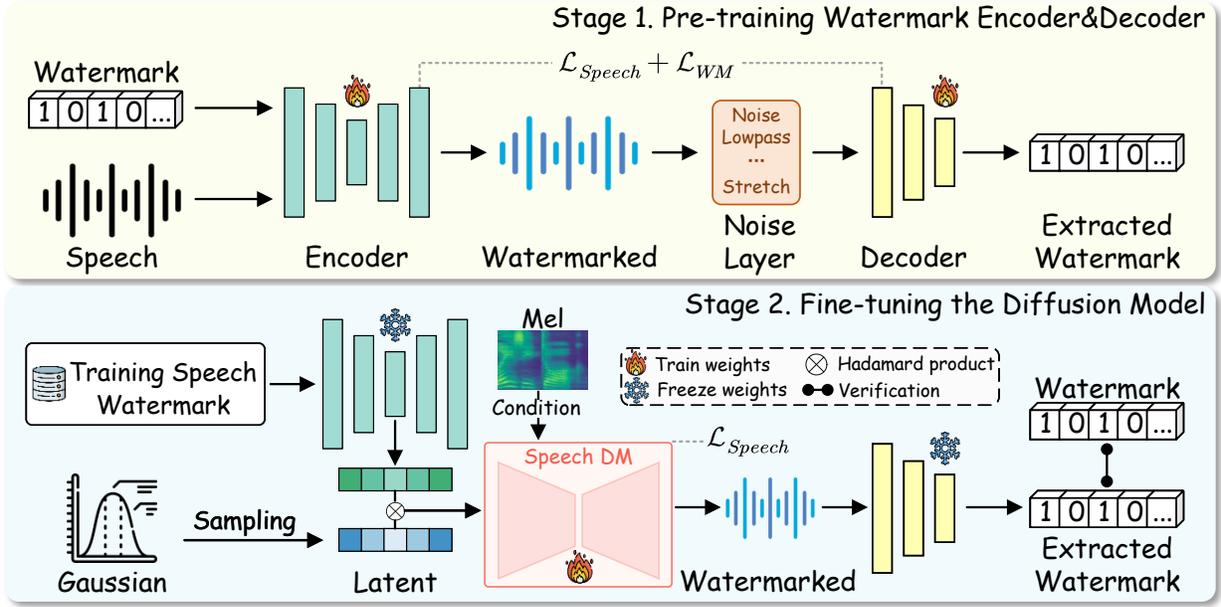
Fig. 2. The pipeline of proposed TriniMark. 1) The stage of pre-training watermark encoder and decoder. The encoder embeds the watermark into the time-domain features of the speech and reconstructs the waveform to obtain the watermarked speech. The decoder disentangles the watermark features and recovers the watermark. To enhance robustness, a noise layer is applied to the watermarked speech before feeding it into the decoder 2) The stage of fine-tuning the diffusion models. The training speech and watermark are first processed by the pretrained encoder to generate watermarked training speech, which is then multiplied with the Gaussian latent variables obtained through the diffusion process to serve as input to the diffusion model. The mel-spectrogram is used as a conditional input to the diffusion model to generate watermarked speech, and the pretrained decoder is then utilized to extract the watermark.
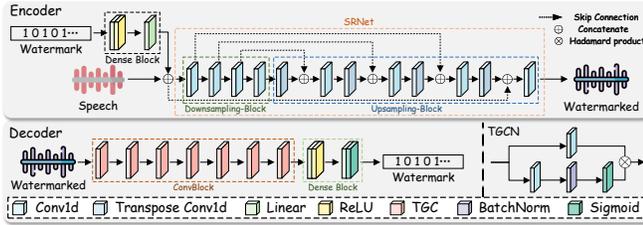

Fig. 3. The Detailed Architecture of Watermark Encoder and Decoder.

the SRNet is attributed to the exclusion of all activation functions and normalization operations, comprising only convolutional layers. This design choice is motivated by the fact that, unlike frequency domain transformation, which relies on high-dimensional features for reconstruction, SRNet utilizes only the time-domain features of speech. As described in MobileNetV2 [49], activation functions such as ReLU can cause excessive loss of low-dimensional features. Consequently, this lightweight structure is employed to reconstruct watermarked speech while preserving its essential characteristics, which can also accelerate training speed and reduce inference latency.

The watermark decoder comprises a ConvBlock and a DenseBlock. The ConvBlock is made up of seven deliberated designed Temporal-aware Gated Convolutional Networks (TGCNs), which are redesigned Gated Convolutional Networks [50] specifically tailored for the temporal features of speech. Each TGCN has two branches: one branch only has a single 1D convolutional layer, while the other branch first passes through a 1D convolutional layer, followed by batch normalization, and finally a gating mechanism with a Sigmoid

function. The outputs of the two branches are then combined using the Hadamard product. Utilizing TGCNs allows the decoder to capture high-dimensional time-domain features of speech, enabling more precise watermark recovery. For robustness, we introduced a noise layer $\mathbf{N}(\cdot)$ that includes common post-processing operations for speech.

2) Pipeline of Pre-training: The cover speech $\mathbf{s} \in \mathbb{R}^{\mathbf{u}}$ and the watermark $\mathbf{w} \in \{0,1\}^l$ as the input for the encoder $\mathcal{E}(\cdot)$, where $\mathbf{u} = C \times L$, C represents channels, L denotes the length of speech, and l is the length of watermark. The watermark $\mathbf{w}$ is first transformed by a DenseBlock containing two fully connected (FC) layers. Then it is concatenated with the speech to reconstruct the watermarked speech $\hat{\mathbf{s}}$. Before extraction, the watermarked speech undergoes a noise layer $\mathbf{N}(\cdot)$ to enhance the resilience of the decoder against various attacks. Finally, the decoder $\mathcal{D}(\cdot)$ processes the attacked speech to disentangle the watermark features and then recover the watermark $\hat{\mathbf{w}}$. The overall pipeline can be formatted as:

$$\hat{\mathbf{w}} = \mathcal{D}(\mathbf{N}(\mathcal{E}(\mathbf{s}, \mathbf{w}))). \tag{9}$$

3) Jointly Optimizing Watermark Encoder and Decoder: Pre-training the watermark encoder-decoder aims to enable the decoder to acquire the watermark priors. Therefore, we employ a joint optimization strategy for training. For recovery accuracy, we employ binary cross-entropy (BCE) for constrain:

$$\mathcal{L}_W = -\sum_{i=1}^{k} w_i \log \hat{w}_i + (1 - w_i) \log(1 - \hat{w}_i). \tag{10}$$

Regarding the quality of watermarked speech, we first utilize the mel-spectrogram loss $\mathcal{L}_{Mel}$ to constrain the distance

between the cover speech $\mathbf{s}$ and the watermarked speech $\hat{\mathbf{s}}$.

$$\mathcal{L}_{Mel} = \mathbb{E}_{\mathbf{s},\hat{\mathbf{s}}}\big[||\phi(\mathbf{s}), \phi(\hat{\mathbf{s}})||_1\big], \tag{11}$$

where $\phi(\cdot)$ represents the transformation of mel-spectrograms and $|| \cdot ||$ denotes the L_1 norm. Furthermore, we employ the logarithmic STFT magnitude loss $\mathcal{L}_{Mag}$ for optimization.

$$\mathcal{L}_{Mag} = ||\log(\mathbf{STFT}(\mathbf{s})), \log(\mathbf{STFT}(\hat{\mathbf{s}}))||_1, \tag{12}$$

where $\mathbf{STFT}(\cdot)$ denotes transformation of STFT magnitude.

The entire loss of pre-training can be defined as:

$$\mathcal{L}_{Pre} = \gamma_{mel}\mathcal{L}_{Mel} + \gamma_{mag}\mathcal{L}_{Mag} + \gamma_w\mathcal{L}_W, \tag{13}$$

where γ_{mel}, γ_{mag}, γ_w are the hyper-parameters, respectively, to balance the speech quality and extraction accuracy.

B. Fine-tuning the Diffusion Models

1) Fine-tuning Strategy: The conventional objective of training the diffusion models is to constrain the noise ϵ added during the diffusion process and the noise estimated by the prediction network $\epsilon_\theta(\cdot)$, as described by Eq. 8. However, this fine-tuning method by constraining noise significantly affects the ability of diffusion models to learn the watermark. As a result, the watermark constraint loss struggles to converge during the fine-tuning process. Moreover, this approach could impair the generative capability of diffusion models, leading to a decline in speech quality.

Considering the above issues, we propose a novel waveform-guided fine-tuning method (WGFT) that constrains the training speech $\mathbf{s}_0$ and the watermarked speech $\hat{\mathbf{s}}_0^{wm}$ obtained through the complete sampling process rather than constraining the original noise and predicted noise, ensuring the generative capability of diffusion models. The rationale behind this design is that the diffusion model has already learned the prior knowledge of noise prediction during its pre-training process. Consequently, the focus should shift towards enabling the diffusion model to learn the watermark through transferability during the fine-tuning stage. Furthermore, we constrain the training speech $\mathbf{s}_0$ and watermarked speech $\hat{\mathbf{s}}_0^{wm}$ using the mel-spectrogram loss $\mathcal{L}_{Mel}$ and the log STFT magnitude loss $\mathcal{L}_{Mag}$, as designed in Section IV-A3 to ensure the generative capability of the diffusion model is not significantly impacted after fine-tuning. The corporate loss $\mathcal{L}_{Speech}$ for speech quality can be computed as follows:

$$\mathcal{L}_{Speech} = \psi_{mel}\mathcal{L}_{Mel} + \psi_{mag}\mathcal{L}_{Mag}, \tag{14}$$

where ψ_{mel} and ψ_{mag} are hyper-parameters for balancing two loss terms, respectively. For watermark recovery accuracy, the BCE loss in Eq. 10 is continued as the constraint.

In a nutshell, the final loss for fine-tuning is defined as:

$$\mathcal{L}_{FT} = \lambda_{speech}\mathcal{L}_{Speech} + \lambda_w\mathcal{L}_W, \tag{15}$$

where λ_{spe} and λ_w are hyper-parameters for the trade-off between the speech quality and extraction accuracy, respectively.

Algorithm 1: Waveform-Guided Fine-tuning Strategy.

Input: Training speech $\mathcal{D}_{train}$, pretrained watermark encoder $\mathcal{E}(\cdot)$ and decoder $\mathcal{D}(\cdot)$, flexible watermark $\mathbf{w}$, hyper-parameter α_t, and diffusion step T.

1   **repeat**
2     $\mathbf{s}_0 \leftarrow \mathbf{s}_0 \sim q_{data}(\mathbf{s}_0)$;
3     $\hat{\mathbf{s}}_0 \leftarrow \hat{\mathbf{s}}_0 = \mathcal{E}(\mathbf{s}_0, \mathbf{w})$; $\triangleright \mathbf{w} \in \{0,1\}^l$
4     $\mathbf{s}_T \leftarrow \mathbf{s}_T \sim \mathcal{N}(0, \mathbf{I})$;
5   **for** $t \leftarrow T, ..., 1$ **do**
6   **if** $t > 1$ **then** $\mathbf{z} \sim \mathcal{N}(0, \mathbf{I})$ **else** $\mathbf{z} \leftarrow 0$;
7       $\hat{\mathbf{s}}_{t-1}^{wm} \leftarrow \frac{1}{\sqrt{\alpha_t}}\big(\hat{\mathbf{s}}_t^{wm} - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(\hat{\mathbf{s}}_t^{wm}, t, c)\big) + \delta_t\mathbf{z}$;
8   **if** $t = 3$
9   **then** $\hat{\mathbf{s}}_3^{wm} \leftarrow \hat{\mathbf{s}}_3^{wm} \otimes \hat{\mathbf{s}}_0$;
10   **end**
11   **return** $\hat{\mathbf{s}}_0^{wm}$;
12    $\hat{\mathbf{w}} \leftarrow \hat{\mathbf{w}} = \mathcal{D}(\hat{\mathbf{s}}_0^{wm})$;
13   Take gradient descent step on:
     $\nabla_\theta(||\phi(\mathbf{s}_0), \phi(\hat{\mathbf{s}}_0^{wm})||_1) +$
     $||\log(\mathbf{STFT}(\mathbf{s}_0)), \log(\mathbf{STFT}(\hat{\mathbf{s}}_0^{wm}))||_1 -$
     $\sum_{i=1}^k w_i \log \hat{w}_i + (1 - w_i)\log(1 - \hat{w}_i)$;
14 **until** *converged*;

---

2) Pipeline of Waveform-Guided Fine-tuning: During the training process, the DDPM-based vocoders typically take training speech $\mathbf{s}_0$ that has undergone the diffusion process as input, with the mel-spectrogram c as a conditional input, to generate the waveform. Therefore, we continue to follow this training process and utilize WGFT for fine-tuning. First, the training speech $\mathbf{s}_0$ passes through the watermark encoder $\mathcal{E}(\cdot)$ to obtain the watermark priors. Afterward, the watermarked training data $\hat{\mathbf{s}}_0$ transformed into standard Gaussian latent variable $\mathbf{s}_t$ by the diffusion process, which is then employed as input for the diffusion model to ultimately generate the watermarked speech. Since the pretrained decoder $\mathcal{D}(\cdot)$ with watermark priors has already been acquired in the first stage, this decoder receives the generated watermarked speech $\hat{\mathbf{s}}_0^{wm}$ and extracts the watermark $\hat{\mathbf{w}}$ from it. The entire process of fine-tuning can be represented as follows:

$$\hat{\mathbf{s}}_0^{wm} = \mathcal{G}_D(\mathcal{E}(\mathbf{s}_0, \mathbf{w})), \tag{16}$$

$$\hat{\mathbf{w}} = \mathcal{D}(\hat{\mathbf{s}}_0^{wm}) \in \mathbb{R}^l, \tag{17}$$

where $\mathcal{G}_D$ represents the diffusion models. The entire pipeline of fine-tuning is formalized in Algorithm 1.

C. Watermark Verification

The detection of the watermark $\mathbf{w}$ can be treated as a rigorous hypothesis test problem [8]. On this side, a verifier makes the decision d whether the watermark exists or not according to a given threshold. In this process, the verifier may make two types of errors under two hypotheses:

$\mathcal{H}_0$: The model is not watermarked: $\hat{\mathbf{w}} \neq \mathbf{w}$. In this case, the *false positive rate* (FPR) is defined as the probability of rejecting $\mathcal{H}_0$ when it is true, i.e., FPR $:= \mathbb{P}(d = 1|\mathcal{H}_0)$.

$\mathcal{H}_1$: The model is watermarked: $\hat{\mathbf{w}} = \mathbf{w}$. In this case, the *false negative rate* (FNR) is defined as the probability of rejecting $\mathcal{H}_1$ when it is false, i.e., FNR $:= \mathbb{P}(d = 0|\mathcal{H}_1)$.

The error of FPR can be constrained with a given threshold. The recovered watermark $\hat{\mathbf{w}}$ and the predefined watermark $\mathbf{w}$ are compared bitwise, and the matching bits number k follows the binomial distribution

$$\mathbb{P}(K = k) = \binom{l}{k} p^k (1-p)^{l-k}, \qquad (18)$$

where p is the probability. (for example, $p = 0.5$ under the hypothesis $\mathcal{H}_0$). With a given FPR and the length of the watermark l, one has a threshold $th = k$ by solving the equation FPR $= \mathbb{P}(K)$. After obtaining the threshold th, the verifier can make the decision d based on bitwise accuracy k/l, e.g., $d = 1$ if $k \geq th$, $d = 0$ otherwise. Then, FPR $:= \mathbb{P}(k \geq th|\mathcal{H}_0)$, and FNR $:= \mathbb{P}(k < th|\mathcal{H}_1)$.

In the hypothesis test, one should also consider the confidence for the decision made by the verifier. As given in [8], if the verifier targets a FPR $= 0.5\%$ with confidence 95%, 768 test cases are needed for verification.

In the end, we also used FNR@0.5%FPR as one of the metrics to evaluate the performance of our method.

V. EXPERIMENTAL ANALYSIS

In this section, we validate the proposed TriniMark through comprehensive experiments, including fidelity, capacity, and robustness. We also compare the TriniMark with existing state-of-the-art (SOTA) methods. In addition, we conduct a thorough analysis of the robustness results produced by different diffusion models.

A. Experimental Setup

1) Datasets and Baseline: We conducted experiments using the LJSpeech [51], LibriTTS [52], and LibriSpeech [53] datasets. Concretely, LJSpeech is a single-speaker dataset containing nearly 24 hours of speech. LibriTTS and LibriSpeech are multi-speaker datasets containing approximately 585 hours and 1000 hours of data, respectively. We downsampled the speech in LibriTTS and LibriSpeech to 22.05 kHz. Since LJSpeech is already sampled at 22.05 kHz, no resampling was performed. To better assess the performance of the proposed method, we segmented all speech to a length of one second. Furthermore, we utilize WavMark [27], AudioSeal [28], and TimbreWM [29] as baseline for comparison.

2) Evaluation Metrics: We evaluated the performance of our method with different objective evaluation metrics. Short-Time Objective Intelligibility (STOI) [54] predicts the intelligibility of speech. Mean Opinion Score of Listening Quality Objective assesses speech quality based on the Perceptual Evaluation of Speech Quality (PESQ) [55]. We also conducted evaluation metrics using Structural Similarity Index Measure (SSIM) [56] and Mel Cepstral Distortion (MCD) [57]. SSIM is a metric typically used for image quality assessment, which has also been adapted to the mel-spectrogram of speech for evaluation. MCD measures the reconstruction distortion of speech signals in terms of mel-frequency cepstral coefficients. Bit-wise accuracy (ACC) is employed to evaluate the accuracy of watermark extraction.

B. Implementation Details

1) Model Settings: **Watermark encoder and decoder.** In the Encoder, the first FC layer of the DB receives the watermark length as input and produces an output dimension of 512. The second FC layer generates an output dimension corresponding to the length of the speech signal. The SRNet employs 1D convolutional layers in its downsampling block, with a kernel size of 3, stride of 2, and padding of 2 for all layers except the final layer, which utilizes a padding of 1. In the upsampling block, all 1D convolutional layers maintain a kernel size of 3, stride of 1, and padding of 1. Furthermore, all 1D transposed convolutional layers have a kernel size of 3, stride of 2, padding of 2, and output padding of 1, except the first layer, which employs a padding of 1. Regarding the Decoder, each 1D convolutional layer in the TGCN is characterized by a kernel size of 3, stride of 2, and padding of 1. The first FC layer of the DB receives the feature length extracted by the ConvBlock as the input dimension, with an output dimension of 512. Meanwhile, the second FC layer produces an output dimension corresponding to the watermark length. **Diffusion model**. We validate the proposed method using DiffWave [45] and PriorGrad [46]. Both DiffWave and PriorGrad are DDPM-based vocoders that utilize Gaussian noise as input and mel-spectrogram as conditional input. Moreover, they incorporate an accelerated sampling algorithm proposed in DiffWave.

2) Training Settings: **Pre-training watermark encoder and decoder.** We use the Adam [58] optimizer with a learning rate of 2e-4 to jointly train the encoder and decoder. The total number of epochs is set to 80, with a batch size of 16. During training, prioritizing watermark extraction accuracy, we initialize the hyper-parameters as follows: $\lambda_w = 1$, $\lambda_{mel} = 0.9$, and $\lambda_{mag} = 0.1$. As the loss function $\mathcal{L}_W$ converges to a certain threshold, we adjust b to 0.9 and c to 0.1, respectively. **Fine-tuning the diffusion models.** We use the AdamW [59] optimizer with a learning rate of 2e-4 for fine-tuning. The total number of epochs is set to 20, and the batch size is 2. We also prioritize watermark extraction accuracy during fine-tuning. Therefore, the hyper-parameter λ_w is set to 1 and λ_{speech} to 0 initially. Once the loss function $\mathcal{L}_W$ decreases to the specified threshold, λ_{speech} is then set to 1. For speech loss $\mathcal{L}_{Speech}$, the hyper-parameter ψ_{mel} is set to 0.2 and ψ_{mag} to 0.8 after reaching the same threshold. All experiments are performed on the platform with Intel(R) Xeon Gold 5218R CPU and NVIDIA GeForce RTX 3090 GPU.

C. Fidelity and Capacity

1) Analysis of Fidelity: Fidelity is a key metric for validating watermarking methods. Therefore, we use the four speech quality evaluation metrics mentioned in Section V-A2 to demonstrate the performance of the proposed TriniMark. In Table I, we present the quality evaluation metrics for three datasets (LJSpeech, LibriTTS, LibriSpeech) under two different diffusion models (DiffWave and PriorGrad). *Baseline* refers to the comparison between *generated speech* and *natural speech*. *100 bps* indicates the comparison between *watermarked speech* generated by the diffusion model with

a watermark length of 100 bps and *generated speech*. From the experimental results, we can analyze the following points. For DiffWave and PriorGrad, the evaluation metrics for watermarked speech (especially SSIM and MCD) on the LibriTTS and LibriSpeech datasets show improvement rather than decline. This is because both diffusion models were pre-trained only on the single-speaker LJSpeech dataset, resulting in a lack of priors from other datasets. As a result, when generating multi-speaker speech, these models do not achieve the same quality as with the LJSpeech dataset. However, through the proposed WGFT strategy, although the primary aim is to enable DMs to learn watermarking, it also facilitates DMs in learning probability distributions from different datasets. Consequently, in subsequent generation tasks, the model exhibits better speech quality.

When evaluating with DiffWave, the watermarked speech generated after fine-tuning the diffusion model shows higher values in SSIM and MCD, and STOI is almost on par with the baseline. However, the PESQ still follows the expected trend, decreasing by 0.0383 and 0.1563 on the LibriTTS and LibriSpeech datasets, respectively. While fine-tuning enhances the diffusion model's capability to generate speech for multi-speaker datasets, it can only mitigate the distortion caused by watermark embedding, as the underlying quality degradation due to watermarking remains an inherent challenge.

For PriorGrad, the same pattern observed in DiffWave is present. However, for multi-speaker datasets, all evaluation metrics show an upward trend. We speculate this is because the pretrained PriorGrad did not achieve optimal performance. Moreover, PriorGrad generates speech using adaptive priors derived from mel-spectrogram statistics. The prior knowledge for multi-speaker datasets differs significantly from that of LJSpeech. Consequently, without training on multi-speaker datasets, the quality of generated speech decreases substantially. After fine-tuning, the watermarked speech generated by PriorGrad exhibits relatively decent quality. STOI remains above 0.92, and SSIM reaches 0.9177 and 0.9067 on LJSpeech and LibriTTS, respectively. On LibriSpeech, SSIM also achieves a value of 0.8497.

2) Analysis of Capacity: We conducted experiments on different diffusion models with four capacities using the LJSpeech dataset. Table II presents the speech quality of watermarked speech and the watermark extraction accuracy for TriniMark at capacities of 100, 200, 300, and 500 bps. All results were obtained by calculating the metrics from the watermarked speech and generated speech. From the experimental results, we can find that as the capacity increases, both speech quality and watermark extraction accuracy show a gradual decline in DiffWave. When the capacity is less than 300 bps, the accuracy remains above 95%. However, at a large capacity of 500 bps, while maintaining a certain level of speech quality, the accuracy significantly drops to only 80.31%.

For PriorGrad, when the capacity is 100 bps, there is a noticeable decline in PESQ. Experiments have shown that this outlier occurs only at this specific capacity. We hypothesize that this outlier is related to PriorGrad itself, as the model generates data based on the statistical priors of the mel-spectrogram, and the watermark length at this capacity affects the prior distribution. Nonetheless, in other capacities, the results follow a normal pattern. When the capacity reaches 500 bps, although the accuracy decreases, it still achieves 88.63%.

Regarding the balance between fidelity and capacity, there is a clear downward trend at a capacity of 500 bps, although maintaining decent speech quality. The extraction accuracy also shows a substantial deterioration. Therefore, we conclude that the threshold of capacity that the proposed TriniMark can effectively accommodate is 500 bps.

3) Comparison of Fidelity and Capacity With SOTA Methods: We compare the fidelity performance of our method with WavMark [27], AudioSeal [28], and TimbreWM [29] at a capacity of 100 bps. Additionally, we compare the fidelity with TimbreWM at a high capacity of 500 bps. Table **??** presents the specific experimental results. For the three SOTA methods, the comparison is between the watermarked speech and the original speech. For our method, the comparison is between the watermarked speech and the generated speech. Although the comparison targets differ during evaluation, the essence is the same: comparing the *watermarked speech* to the *same speech before watermarking*. Based on the experimental results, we can observe that AudioSeal exhibited superior performance in the comparison of fidelity at low capacity. While the proposed TriniMark did not demonstrate the best performance among all methods in DIffWave and PriorGrad, it achieved values of 1.2644 and 2.0892 in MCD, respectively, which are slightly higher than those of other methods. Its performance in other metrics was only marginally lower than these methods.

Regarding capacity, all methods demonstrated good watermark extraction accuracy at low capacity. WavMark achieved 100% accuracy due to its use of invertible neural networks (INNs). However, our TriniMark also achieved 98.43% and 98.11% accuracy under different DMs.

At high capacity, although TImbreWM showed good values in speech quality, its accuracy was only 49.99%, indicating that this method does not handle large capacity well. On the other hand, the proposed TriniMark maintained good speech quality (STOI and SSIM above 0.94 and 0.86) and achieved accuracy rates of 80.31% and 88.63%.

D. Robustness

1) Analysis of Robustness Against Individual Attacks: We verified the robustness of the proposed TriniMark through conventional speech post-processing operations. The post-processing attacks encompass six common distortion attacks: Gaussian noise with noise levels of 5, 10, 15, and 20 dB, pink noise with a factor of 0.5, low-pass filtering with a threshold of 3 kHz, band-pass filtering with a 0.5-8 kHz threshold, cropping, and echo. In addition, two desynchronization attacks are employed: time stretching with an intensity of 2 and dither. The echo and dither attacks utilize default settings. Cropping is subdivided into two categories: cropping the first half of the signal and cropping the latter half. The comprehensive analysis of the experimental results is as follows. For the robustness of TriniMark on DiffWave, when the noise level

TABLE I
FIDELITY OF TRINIMARK WITH VARIOUS DATASETS IN DIFFERENT DMs. ↑/↓ INDICATES A HIGHER/LOWER VALUE IS MORE DESIRABLE.

DiffWave	Generated ↔ Natural			Watermarked ↔ Generated			Watermarked ↔ Natural		
	LJSpeech	LibriTTS	LibriSpeech	LJSpeech	LibriTTS	LibriSpeech	LJSpeech	LibriTTS	LibriSpeech
STOI↑	0.9655	0.9337	0.9176	0.9621	0.9386	0.9290	0.9622	0.9542	0.9416
PESQ↑	3.5120	2.8156	2.7788	3.1335	2.7773	2.6225	3.1265	2.7860	2.6265
SSIM↑	0.8453	0.8025	0.6699	0.9174	0.8945	0.8392	0.8567	0.8324	0.7205
MCD↓	6.2794	6.0407	15.2482	1.2644	1.2127	1.1176	6.0290	6.0031	15.3361
ACC↑	N/A	N/A	N/A	0.9843	0.9588	0.9806	0.9840	0.9588	0.9809

PriorGrad	Generated ↔ Natural			Watermarked ↔ Generated			Watermarked ↔ Natural		
	LJSpeech	LibriTTS	LibriSpeech	LJSpeech	LibriTTS	LibriSpeech	LJSpeech	LibriTTS	LibriSpeech
STOI↑	0.9722	0.9463	0.8970	0.9424	0.9186	0.9201	0.9585	0.9421	0.9480
PESQ↑	3.8875	2.0509	1.9728	2.1856	1.7864	2.3841	2.1928	1.7870	2.3910
SSIM↑	0.9032	0.7130	0.5642	0.9177	0.8876	0.8497	0.8943	0.8652	0.7885
MCD↓	5.5146	16.4618	34.8790	2.0892	2.2353	1.9604	5.6031	7.2721	12.0343
ACC↑	N/A	N/A	N/A	0.9811	0.9981	0.9986	0.9799	0.9981	0.9987

WaveGrad	Generated ↔ Natural			Watermarked ↔ Generated			Watermarked ↔ Natural		
	LJSpeech	LibriTTS	LibriSpeech	LJSpeech	LibriTTS	LibriSpeech	LJSpeech	LibriTTS	LibriSpeech
STOI↑	0.9363	0.8996	0.8792	0.8978	0.8677	0.8349	0.9169	0.8816	0.8598
PESQ↑	2.2339	1.7483	1.9555	2.0913	1.7754	1.8016	2.0926	1.7796	1.8021
SSIM↑	0.7448	0.7533	0.6434	0.8426	0.8058	0.7168	0.7786	0.7503	0.6287
MCD↓	8.6023	5.9581	16.2426	1.9275	2.2932	3.0281	6.4706	4.7066	16.8436
ACC↑	N/A	N/A	N/A	0.9821	0.9702	0.9316	0.9813	0.9693	0.9300

TABLE II
CAPACITY OF TRINIMARK WITH LJSPEECH IN DIFFERENT DMs.
↑/↓ INDICATES A HIGHER/LOWER VALUE IS MORE DESIRABLE.

DMs		Capacity (bps)			
		100	200	300	500
DiffWave	STOI↑	0.9621	0.9688	0.9292	0.9412
	PESQ↑	3.1335	3.3965	2.1007	2.1151
	SSIM↑	0.9174	0.9191	0.8729	0.8615
	MCD↓	1.2644	1.1608	1.2797	1.4438
	ACC↑	0.9843	0.9517	0.9500	0.8031
PriorGrad	STOI↑	0.9424	0.9635	0.9530	0.9607
	PESQ↑	2.1856	3.0401	3.0447	2.5963
	SSIM↑	0.9177	0.9086	0.9168	0.8985
	MCD↓	2.0892	2.1592	2.0461	2.0893
	ACC↑	0.9811	0.9960	0.9333	0.8863

TABLE III
COMPARISON OF FIDELITY WITH SOTA METHODS.
↑/↓ INDICATES A HIGHER/LOWER VALUE IS MORE DESIRABLE.

Methods	Capacity (bps)	STOI↑	PESQ↑	SSIM↑	MCD↓	ACC↑
AudioSeal [28]	16	0.9985	**4.5888**	**0.9811**	3.8395	0.9214
WavMark [27]	32	**0.9997**	4.4625	0.9690	2.0504	**1.0000**
TimbreWM [29]	100	0.9853	4.0371	0.9388	3.1715	0.9998
TriniMark(DW)	100	0.9621	3.1335	0.9174	**1.2644**	0.9843
TriniMark(PG)	100	0.9424	2.1856	0.9177	2.0892	0.9811
TriniMark(WG)	100	0.9424	2.1856	0.9177	2.0892	0.9811
TimbreWM [29]	500	0.9987	4.6322	0.9990	0.9811	0.4999
TriniMark(DW)	500	0.9412	2.1151	0.8615	1.4438	**0.8031**
TriniMark(PG)	500	0.9607	2.5963	0.8985	2.0893	**0.8863**
TriniMark(WG)	500	0.9607	2.5963	0.8985	2.0893	**0.8863**

of Gaussian noise is high, the method can effectively resist noise attacks on three datasets. However, at an intensity of 5 dB, the watermark extraction accuracy significantly decreases. When dealing with pink noise, the average accuracy reaches 81.87%. The proposed TriniMark can effectively handle low-pass filtering, with accuracies all above 88.12%. It performs even better against band-pass filtering, with an average ac-

curacy of 97.42%. Although TriniMark is less stable against cropping, it still achieves average accuracies of 88.92% and 89.26%, respectively. Similarly, our method can resist echo attacks with an average accuracy of 90.62%. When facing desynchronization attacks, TriniMark can resist time stretching and dither with high accuracies of 96.93% and 97.43%, respectively.

Regarding PriorGrad, when handling all intensities of Gaussian noise, the proposed TriniMark performs exceptionally well. At a noise intensity of 5 dB, the accuracy reaches 97.42% on LJSpeech and 87.38% on LibriTTS. Although the accuracy slightly decreases on LibriSpeech, it still achieves 74.98%. In combating pink noise, our TriniMark also demonstrates superior robustness, with an average accuracy of 92.55%. When facing low-pass and band-pass filtering, the accuracies are impressively high at 98.30% and 98.22%, respectively. After undergoing two types of cropping, our method maintains accuracies of 91.75% and 96.69%. It achieves an accuracy of 95.12% against echo attacks. When dealing with asynchronous attacks, TriniMark achieves average accuracies of 98.60% and 99.22% for time stretching and dither, respectively.

We investigated the reason behind the superior robustness of TriniMark against Gaussian noise when applied to PriorGrad. During the denoising process, PriorGrad synthesizes speech using statistical priors, enabling more accurate removal of the noise introduced in the diffusion process. Compared to DiffWave, this operation allows PriorGrad to handle Gaussian noise more robustly. Regardless of whether it is applied to DIffWave or PriorGrad, TriniMark demonstrates superior and balanced robustness in handling various speech post-processing attacks.

2) Analysis of Robustness Against Compound Attacks: In a real-world transmission environment, speech waveforms are subject to unknown attacks, often experiencing more than one type of attack. To better adapt to realistic scenarios, we further validate the robustness of the proposed TriniMark by

TABLE IV
ROBUSTNESS OF TRINIMARK WITH VARIOUS DATASETS IN DIFFERENT DMS AGAINST INDIVIDUAL ATTACKS.

DMs	Dataset		Gaussian				Pink	LP	BP	Cropping		Echo	Stretch	Dither
			5 dB	10 dB	15 dB	20 dB	0.5	3k	0.5-8k	front	behind	default	2×	default
DiffWave	LJSpeech	STOI↑	0.8351	0.9028	0.9491	0.9762	0.8498	0.9997	0.8619	0.4136	0.5442	0.7777	0.9999	1.0000
		PESQ↑	3.1257	3.1320	3.1245	3.1270	3.1282	3.1356	3.1326	3.1227	3.1230	3.1347	3.1341	3.1299
		ACC↑	0.6777	0.7761	0.8777	0.9458	0.8487	0.9335	0.9838	0.8649	0.9527	0.9483	0.9809	0.9829
	LibriTTS	STOI↑	0.8372	0.8943	0.9362	0.9639	0.8205	0.9998	0.8843	0.4146	0.5387	0.7728	0.9999	1.0000
		PESQ↑	2.7822	2.7831	2.7772	2.7837	2.7786	2.7834	2.7757	2.7815	2.7837	2.7792	2.7722	2.7798
		ACC↑	0.6868	0.7769	0.8637	0.9207	0.7872	0.8954	0.9582	0.8371	0.9027	0.9056	0.9524	0.9589
	LibriSpeech	STOI↑	0.8137	0.8635	0.9019	0.9290	0.7864	0.9997	0.8969	0.5336	0.4559	0.7034	0.9999	0.9999
		PESQ↑	2.6197	2.6195	2.6139	2.6229	2.6141	2.6221	2.6205	2.6171	2.6191	2.6182	2.6203	2.6206
		ACC↑	0.6880	0.7769	0.8697	0.9324	0.8201	0.8812	0.9807	0.9655	0.8223	0.8646	0.9747	0.9810
PriorGrad	LJSpeech	STOI↑	0.8638	0.9249	0.9625	0.9832	0.9061	0.9997	0.8615	0.4219	0.5298	0.7525	0.9999	1.0000
		PESQ↑	2.1875	2.1861	2.1815	2.1853	2.1884	2.1853	2.1907	2.1912	2.1880	2.1875	2.1902	2.1874
		ACC↑	0.9142	0.9585	0.9738	0.9788	0.8797	0.9769	0.9694	0.9127	0.9503	0.9347	0.9807	0.9797
	LibriTTS	STOI↑	0.8588	0.9159	0.9540	0.9766	0.9108	0.9987	0.8800	0.4252	0.5247	0.7646	0.9991	0.9998
		PESQ↑	2.1766	2.1756	2.1731	2.1758	1.7879	2.1741	2.1747	2.1773	2.1735	2.1750	2.1757	1.7872
		ACC↑	0.8738	0.9388	0.9663	0.9753	0.9792	0.9756	0.9783	0.8859	0.9531	0.9487	0.9788	0.9982
	LibriSpeech	STOI↑	0.8236	0.8801	0.9224	0.9525	0.8745	0.9998	0.8931	0.6946	0.2175	0.5843	0.9999	1.0000
		PESQ↑	2.3829	2.3905	2.3918	2.3903	2.3880	2.3898	2.3891	2.3915	2.3887	2.3916	2.3855	2.3901
		ACC↑	0.7498	0.8609	0.9453	0.9846	0.9175	0.9966	0.9988	0.9540	0.9972	0.9703	0.9985	0.9987
WaveGrad	LJSpeech	STOI↑	0.7911	0.8722	0.9300	0.9647	0.7995	0.8553	0.7993	0.4059	0.5326	0.7449	0.9967	0.9968
		PESQ↑	2.0943	2.0966	2.1026	2.0992	2.1002	2.1003	2.0893	2.0967	2.0970	2.0976	2.0988	2.0969
		ACC↑	0.7562	0.8593	0.9316	0.9644	0.9163	0.8995	0.9818	0.8718	0.9623	0.9594	0.9805	0.9818
	LibriTTS	STOI↑	0.8174	0.8842	0.9327	0.9629	0.7746	0.9666	0.8328	0.4083	0.5350	0.7684	0.9985	0.9985
		PESQ↑	1.7787	1.7724	1.7728	1.7788	1.7733	1.7750	1.7739	1.7749	1.7732	1.7776	1.7810	1.7774
		ACC↑	0.7889	0.8795	0.9350	0.9587	0.8968	0.9073	0.9698	0.8626	0.9409	0.9375	0.9706	0.9705
	LibriSpeech	STOI↑	0.7454	0.7921	0.8330	0.8692	0.7093	0.9847	0.8673	0.5893	0.3262	0.5893	0.9986	0.9986
		PESQ↑	1.8076	1.8064	1.8111	1.8064	1.8043	1.7985	1.8034	1.8008	1.8017	1.8057	1.8028	1.8056
		ACC↑	0.7290	0.8161	0.8771	0.9124	0.8179	0.8523	0.9312	0.8655	0.8292	0.8515	0.9302	0.9326

combining two types of attacks into compound attacks. In the real scenario, noise attacks are more common, so we employed the following seven compound attacks: a) Gaussian noise followed by band-pass filtering, b) Gaussian noise succeeded by echo, c) Gaussian noise combined with dither, d) combined Gaussian and pink noise attack, e) pink noise accompanied by band-pass filtering, f) pink noise coupled with echo, and g) pink noise aligned with dither. We present the TriniMark's experimental results of robustness on different DMs across three datasets in Table V. All values are obtained by comparing the watermarked speech with the generated speech. From the experimental results, the analyses can be found that the proposed TriniMark demonstrates good robustness on DiffWave when handling compound attacks involving Gaussian noise. For compound attacks a) and c), the average watermark extraction accuracy is 93.27% and 93.25%, respectively. However, for compound attack b), the accuracy decreases to 86.32%. Even under the more severe compound attack d), TriniMark still achieves an average accuracy of 80.22%. When handling the remaining three compound attacks that include pink noise, the average accuracies are 82.36%, 76.09%, and 81.88%, respectively.

For PriorGrad, TriniMark also demonstrates superior robustness compared to DiffWave. When handling compound attacks a) and c), the average extraction accuracy reaches 98.31%. For compound attack b), it remains at 95.11%. Under the compound attack d), TriniMark maintains a high accuracy of 91.93%. For compound attacks involving pink noise, TriniMark exhibits superior performance, achieving average accuracies of 94.51%, 88.46%, and 92.60% for the remaining three

compound attacks, respectively. This enhanced robustness is attributed to PriorGrad's ability to obtain more statistical priors about the waveform compared to DiffWave. Consequently, through WGFT, TriniMark demonstrates significantly stronger robustness on PriorGrad.

For two different DMs, TriniMark exhibits slightly reduced robustness against compound attacks involving pink noise compared to those involving Gaussian noise, as it demonstrates better robustness against Gaussian noise when dealing with single attacks. In general, the proposed TriniMark exhibits stable and balanced robustness across both DMs.

3) Comparison of Robustness With SOTA Methods Against Individual Attacks: We further compared the performance of TriniMark against single attacks with SOTA methods. Table VI presents the robustness results of different methods against four types of attacks. The best results are highlighted in bold, while the second-best results are underlined. From this comparison, we can observe that When resisting Gaussian noise with a noise level of 10 dB, TriniMark on PriorGrad outperformed the other four methods by 18.24% and 43.71% for the smallest and largest differences, respectively. While the gap is not as pronounced for pink noise, the extraction accuracy on both diffusion models still exceeds other baselines.

For cropping attacks, TimbreWM demonstrated the best robustness. Although the proposed method's extraction accuracy is only slightly lower than the best result, it still shows considerable robustness. Regarding echo attacks, TriniMark exhibited higher accuracy compared to the baselines.

The proposed TriniMark shows exceptional robustness against noise-based attacks. Although it does not always

TABLE V
ROBUSTNESS OF TRINIMARK IN DIFFERENT DMS AGAINST COMPOUND ATTACKS.

Datasets		TriniMark (DW)						
		GN+BP	GN+Echo	GN+Dither	GN+PN	PN+BP	PN+Echo	PN+Dither
LJSpeech	STOI↑	0.8453	0.7555	0.9762	0.8445	0.7853	0.6764	0.8495
	PESQ↑	3.1356	3.1254	3.1269	3.1340	3.1264	3.1323	3.1261
	ACC↑	0.9449	0.9071	0.9442	0.8285	0.8524	0.8117	0.8482
LibriTTS	STOI↑	0.8522	0.7477	0.9640	0.8162	0.7612	0.6533	0.8204
	PESQ↑	2.7796	2.7777	2.7846	2.7779	2.7800	2.7804	2.7779
	ACC↑	0.9197	0.8694	0.9207	0.7757	0.7957	0.7542	0.7885
LibriSpeech	STOI↑	0.8354	0.6858	0.9292	0.7826	0.7375	0.6009	0.7867
	PESQ↑	2.6161	2.6210	2.6153	2.6218	2.6168	2.6204	2.6173
	ACC↑	0.9324	0.8132	0.9327	0.8023	0.8226	0.7168	0.8197
Datasets		TriniMark (PG)						
		GN+BP	GN+Echo	GN+Dither	GN+PN	PN+BP	PN+Echo	PN+Dither
LJSpeech	STOI↑	0.8497	0.7452	0.9830	0.9012	0.8194	0.6968	0.9052
	PESQ↑	2.1873	2.1925	2.1856	2.1896	2.1862	2.1907	2.1892
	ACC↑	0.9670	0.9334	0.9787	0.8796	0.9355	0.8313	0.8820
LibriTTS	STOI↑	0.8616	0.7387	0.9804	0.9030	0.8246	0.8244	0.9099
	PESQ↑	1.7876	1.7858	1.7881	1.7866	1.7866	1.7889	1.7870
	ACC↑	0.9972	0.9856	0.9973	0.9772	0.9806	0.9802	0.9788
LibriSpeech	STOI↑	0.8515	0.5618	0.9525	0.8671	0.8039	0.5372	0.8737
	PESQ↑	2.3959	2.3924	2.3923	2.3881	2.3912	2.3920	2.3907
	ACC↑	0.9850	0.9342	0.9846	0.9011	0.9192	0.8423	0.9171
Datasets		TriniMark (WG)						
		GN+BP	GN+Echo	GN+Dither	GN+PN	PN+BP	PN+Echo	PN+Dither
LJSpeech	STOI↑	0.7811	0.7226	0.9646	0.7933	0.7878	0.7207	0.9478
	PESQ↑	2.0871	2.0982	2.0992	2.0995	2.1081	2.1066	2.0970
	ACC↑	0.9665	0.9380	0.9643	0.9035	0.9785	0.9536	0.9786
LibriTTS	STOI↑	0.8074	0.7453	0.9631	0.7713	0.6165	0.7728	0.6145
	PESQ↑	1.7768	1.7453	1.9631	1.7713	1.7744	1.7763	1.7747
	ACC↑	0.9586	0.7453	0.9631	0.8886	0.8615	0.8968	0.8599
LibriSpeech	STOI↑	0.7591	0.5579	0.8685	0.7078	0.6527	0.4833	0.7103
	PESQ↑	1.8066	1.8034	1.8038	1.8086	1.7987	1.8081	1.8018
	ACC↑	0.9119	0.8271	0.9104	0.8085	0.8212	0.7430	0.8168

TABLE VI
COMPARISON OF ROBUSTNESS AGAINST INDIVIDUAL ATTACKS WITH SOTA METHODS. ↑ INDICATES A HIGHER VALUE IS MORE DESIRABLE.
THE BEST RESULTS ARE MARKED IN **BOLD** AND THE SECOND BEST RESULTS ARE UNDERLINE.

	Gaussian						Pink			Cropping						Echo		
	10 dB			20 dB			0.5			Front			Behind			Default		
	STOI↑	PESQ↑	ACC↑	STOI↑	PESQ↑	ACC↑	STOI↑	PESQ↑	ACC↑	STOI↑	PESQ↑	ACC↑	STOI↑	PESQ↑	ACC↑	STOI↑	PESQ↑	ACC↑
WavMark	0.8898	1.2217	0.5214	0.9733	2.1236	0.6523	0.8673	1.3019	0.6924	0.4185	1.7090	0.9797	0.5135	1.7466	0.9713	0.6122	1.3716	0.8668
AudioSeal	0.9110	1.0995	0.6086	0.9789	1.5987	0.6600	0.9185	1.3537	0.6571	0.4150	1.0916	0.7226	0.5348	1.1661	0.8925	0.7563	1.1845	0.7277
TimbreWM	0.9136	1.3347	0.6335	0.9812	2.7424	0.8154	0.8473	1.3773	0.7282	0.4135	1.8149	**0.9888**	0.5331	1.8155	**0.9814**	0.7559	1.4720	0.5818
TriniMark(DW)	0.9028	3.1320	0.7761	0.9762	3.1270	0.9458	0.8498	3.1282	0.8487	0.4136	3.1227	0.8649	0.5442	3.1230	0.9527	0.7777	3.1347	0.9483
TriniMark(PG)	0.9249	2.1861	**0.9585**	0.9832	2.1853	**0.9788**	0.9061	2.1884	0.8797	0.4219	2.1912	0.9127	0.5298	2.1880	0.9503	0.7525	2.1875	0.9347
TriniMark(WG)	0.8722	2.0966	0.8593	0.9647	2.0992	0.9644	0.7995	2.1002	**0.9163**	0.4059	2.0967	0.8718	0.5326	2.0970	0.9623	0.7449	2.0976	**0.9594**

achieve the highest extraction accuracy against all attacks, TriniMark is able to balance the defense across various attacks, rather than excelling only against a specific type of attack.

4) Comparison of Robustness With SOTA Methods Against Compound Attacks: We also compared the proposed TriniMark method with SOTA methods in countering compound attacks. The experiments were conducted using the following four compound attacks: a) Gaussian noise combined with band-pass filtering, b) Gaussian noise coupled with echo, c) Gaussian noise accompanied by pink noise, d) pink noise succeeded by band-pass filtering, and (e) pink noise aligned with echo. Based on the experimental results presented in Table VII, the analyses as follow. For combating compound attacks involving Gaussian noise, TriniMark exhibited superior robustness. For both compound attacks a) and b), the watermark extraction accuracy remained above 90%, significantly higher than the baselines. Even when facing the more severe compound attack c), where baseline accuracies dropped below 68%, TriniMark maintained an accuracy above 82%.

When defending against compound attacks involving pink noise, TriniMark demonstrated superior robustness in compound attack d), with accuracies on both diffusion models exceeding the best baseline by 14.39% and 22.70%, respectively. For compound attack e), although the accuracy of all methods decreased, our method still remained above 81%.

While TriniMark may not achieve the best robustness against each individual attack, it consistently shows significantly higher robustness than the baselines against any com-

TABLE VII
COMPARISON OF ROBUSTNESS AGAINST COMPOUND ATTACKS WITH SOTA METHODS. ↑ INDICATES A HIGHER VALUE IS MORE DESIRABLE.
THE BEST RESULTS ARE MARKED IN **BOLD** AND THE SECOND BEST RESULTS ARE <u>UNDERLINE</u>.

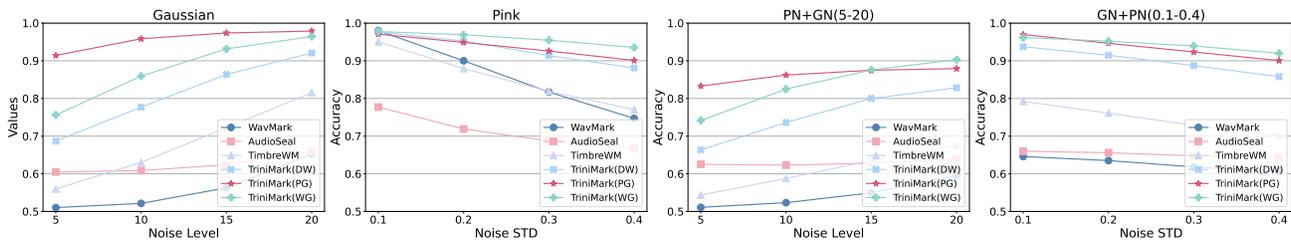|	GN+BP			GN+Echo			GN+PN			PN+BP			PN+Echo		
	STOI↑	PESQ↑	ACC↑	STOI↑	PESQ↑	ACC↑	STOI↑	PESQ↑	ACC↑	STOI↑	PESQ↑	ACC↑	STOI↑	PESQ↑	ACC↑
WavMark	0.8886	1.1139	0.6494	0.5922	1.2798	0.5547	0.8601	1.2292	0.5957	0.8102	1.0785	0.6608	0.5221	1.1868	0.5617
AudioSeal	0.8447	1.5308	0.6409	0.7400	1.1177	0.6305	0.9118	1.2292	0.6394	0.8292	1.7364	0.6480	0.7099	1.0912	0.6280
TimbreWM	0.9853	4.0366	0.7971	0.9853	4.0366	0.7458	0.9853	4.0366	0.6764	0.9853	4.0366	0.7085	0.9853	4.0366	0.6820
TriniMark(DW)	0.8453	3.1356	<u>0.9449</u>	0.7555	3.1254	<u>0.9071</u>	0.8445	3.1340	<u>0.8285</u>	0.7853	3.1264	<u>0.8524</u>	0.6764	3.1323	0.8117
TriniMark(PG)	0.8497	2.1873	**0.9670**	0.7452	2.1925	<u>0.9334</u>	0.9012	2.1896	<u>0.8796</u>	0.8194	2.1862	<u>0.9355</u>	0.6968	2.1907	<u>0.8313</u>
TriniMark(WG)	0.7811	2.0871	<u>0.9665</u>	0.7226	2.0982	**0.9380**	0.7933	2.0995	**0.9035**	0.7878	2.1081	**0.9785**	0.7207	2.1066	**0.9536**

Fig. 4. Comparison of Robustness Against Noise-level Attacks. For *Gaussian* and *PN+GN*, four different noise levels of Gaussian noise (5, 10, 15, and 20 dB) are set. As the noise level decreases, the attack strength increases. For *Pink* and *GN+PN*, four different noise standard deviations (STD) of pink noise (0.1, 0.2, 0.3, and 0.4) are set. As the STD increases, the attack strength increases.

pound attack. This indicates that the proposed method is better suited for real-world transmission environments.

VI. CONCLUSION

In this paper, we propose a generative speech watermarking based on fine-grained feature transfer, which establishes a trinity traceability mechanism that simultaneously authenticates three essential dimensions: the generative model, the synthesized speech, and the end-user. The proposed TriniMark consists of two stages of training. In the first stage, to achieve efficient transfer of watermark generation to the generative model, a watermark encoder-decoder is designed. Specifically, to achieve high-precision watermark extraction, we design a temporal-aware gated convolutional network as the backbone of the watermark decoder. In the second stage, we further propose a waveform-guided fine-tuning strategy. This strategy embeds watermarks into the training data employing the pretrained encoder and jointly optimizes gradients with the pretrained decoder. At the same time, this fine-tuning strategy enables TriniMark to adapt to arbitrary watermarks with only a single round of training. Fidelity and capacity experiments demonstrate that TriniMark can generate high-quality watermarked speech even under a high capacity of 500 bps. Robustness experiments further verify the superior performance of our method compared to existing approaches when facing both individual and compound attacks.

REFERENCES

[1] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.

[2] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in neural information processing systems*, vol. 33, pp. 6840–6851, 2020.

[3] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," in *International Conference on Learning Representations*, 2020.

[4] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 10 684–10 695.

[5] N. Yu, V. Skripniuk, S. Abdelnabi, and M. Fritz, "Artificial fingerprinting for generative models: Rooting deepfake attribution in training data," in *Proceedings of the IEEE/CVF International conference on computer vision*, 2021.

[6] P. Fernandez, G. Couairon, H. Jégou, M. Douze, and T. Furon, "The stable signature: Rooting watermarks in latent diffusion models," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.

[7] V. Asnani, J. Collomosse, T. Bui, X. Liu, and S. Agarwal, "Promark: Proactive diffusion watermarking for causal attribution," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

[8] D. Lin, B. Tondi, B. Li, and M. Barni, "Cycleganwm: A cyclegan watermarking method for ownership verification," *IEEE Transactions on Dependable and Secure Computing*, 2024.

[9] Q. Song, Z. Luo, K. C. Cheung, S. See, and R. Wan, "Protecting nerfs' copyright via plug-and-play watermarking base model," in *Proceedings of the European conference on computer vision (ECCV)*, 2024.

[10] Y. Jang, D. I. Lee, M. Jang, J. W. Kim, F. Yang, and S. Kim, "Waterf: Robust watermarks in radiance fields for protection of copyrights," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

[11] J. Zhu, R. Kaplan, J. Johnson, and L. Fei-Fei, "Hidden: Hiding data with deep networks," in *Proceedings of the European Conference on computer vision (ECCV)*, 2018.

[12] M. Tancik, B. Mildenhall, and R. Ng, "Stegastamp: Invisible hyperlinks in physical photographs," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020.

[13] C. Zhang, P. Benz, A. Karjauv, G. Sun, and I. S. Kweon, "Udh: Universal deep hiding for steganography, watermarking, and light field messaging," *Advances in Neural Information Processing Systems*, vol. 33, pp. 10 223–10 234, 2020.

[14] P. Fernandez, A. Sablayrolles, T. Furon, H. Jégou, and M. Douze, "Watermarking images in self-supervised latent spaces," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 3054–3058.

[15] X. Zhang, R. Li, J. Yu, Y. Xu, W. Li, and J. Zhang, "Editguard: Versatile image watermarking for tamper localization and copyright protection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 11 964–11 974.

[16] C. Xiong, C. Qin, G. Feng, and X. Zhang, "Flexible and secure

watermarking for latent diffusion model," in *Proceedings of the 31st ACM International Conference on Multimedia*, 2023, pp. 1668–1676.

[17] T. Bui, S. Agarwal, N. Yu, and J. Collomosse, "Rosteals: Robust steganography using autoencoder latent space," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 933–942.

[18] Y. Wen, J. Kirchenbauer, J. Geiping, and T. Goldstein, "Tree-rings watermarks: Invisible fingerprints for diffusion images," vol. 36, 2023.

[19] H. Huang, Y. Wu, and Q. Wang, "Robin: Robust and invisible watermarks for diffusion models with adversarial optimization," vol. 37, 2024.

[20] Z. Yang, K. Zeng, K. Chen, H. Fang, W. Zhang, and N. Yu, "Gaussian shading: Provable performance-lossless image watermarking for diffusion models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 12 162–12 171.

[21] Y. Cho, C. Kim, Y. Yang, and Y. Ren, "Attributable watermarking of speech generative models," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 3069–3073.

[22] L. Juvela and X. Wang, "Collaborative watermarking for adversarial speech synthesis," in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2024, pp. 11 231–11 235.

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

[24] J. Copet, F. Kreuk, I. Gat, T. Remez, D. Kant, G. Synnaeve, Y. Adi, and A. Défossez, "Simple and controllable music generation," *Advances in Neural Information Processing Systems*, vol. 35, 2023.

[25] J. Zhou, J. Yi, T. Wang, J. Tao, Y. Bai, C. Y. Zhang, Y. Ren, and Z. Wen, "Traceablespeech: Towards proactively traceable text-to-speech with watermarking," in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2024.

[26] R. San Roman, P. Fernandez, A. Deleforge, Y. Adi, and R. Serizel, "Latent watermarking of audio generative models," 2024.

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

[28] R. S. Roman, P. Fernandez, A. Défossez, T. Furon, T. Tran, and H. Elsahar, "Proactive detection of voice cloning with localized watermarking," in *Procceedings of the 41st International Conference on Machine Learning*, 2024.

[29] C. Liu, J. Zhang, T. Zhang, X. Yang, W. Zhang, and N. Yu, "Detecting voice cloning attacks via timbre watermarking," in *Proceedings of the 31th Network and Distributed System Security (NDSS) Symposium 2024*, 2024.

[30] C. Tong, I. Natgunanathan, Y. Xiang, J. Li, T. Zong, X. Zheng, and L. Gao, "Enhancing robustness of speech watermarking using a transformer-based framework exploiting acoustic features," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2024.

[31] B. Li, J. Chen, Y. Xu, W. Li, and Z. Liu, "Draw: Dual-decoder-based robust audio watermarking against desynchronization and replay attacks," *IEEE Transactions on Information Forensics and Security*, 2024.

[32] W. Liu, Y. Li, D. Lin, H. Tian, and H. Li, "Groot: Generating robust watermark for diffusion-model-based audio synthesis," in *Proceedings of the 32nd ACM International Conference on Multimedia*, 2024.

[33] K. Pavlović, S. Kovačević, I. Djurović, and A. Wojciechowski, "Robust speech watermarking by a jointly trained embedder and detector using a dnn," *Digital Signal Processing*, 2022.

[34] P. O'Reilly, Z. Jin, J. Su, and B. Pardo, "Maskmark: Robust neuralwatermarking for real and synthetic speech," in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2024.

[35] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[36] C. Liu, J. Zhang, H. Fang, Z. Ma, W. Zhang, and N. Yu, "Dear: A deep-learning-based audio re-recording resilient watermarking," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2023.

[37] X. Qu, X. Yin, P. Wei, L. Lu, and Z. Ma, "Audioqr: deep neural audio watermarks for qr code," in *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, 2023.

[38] K. Chen, H. Zhou, H. Zhao, D. Chen, W. Zhang, and N. Yu, "Distribution-preserving steganography based on text-to-speech generative models," *IEEE Transactions on Dependable and Secure Computing*, 2021.

[39] J. Ding, K. Chen, Y. Wang, N. Zhao, W. Zhang, and N. Yu, "Discop: Provably secure steganography in practice based on "distribution copies"," in *2023 IEEE Symposium on Security and Privacy (SP)*, 2023.

[40] J. Li, K. Wang, and X. Jia, "A coverless audio steganography based on generative adversarial networks," *Electronics*, 2023.

[41] V. Popov, I. Vovk, V. Gogoryan, T. Sadekova, and M. Kudinov, "Grad-tts: A diffusion probabilistic model for text-to-speech," in *International Conference on Machine Learning*. PMLR, 2021, pp. 8599–8608.

[42] R. Huang, Z. Zhao, H. Liu, J. Liu, C. Cui, and Y. Ren, "Prodiff: Progressive fast diffusion model for high-quality text-to-speech," in *Proceedings of the 30th ACM International Conference on Multimedia*, 2022, pp. 2595–2605.

[43] J. Chen, X. Song, Z. Peng, B. Zhang, F. Pan, and Z. Wu, "Lightgrad: Lightweight diffusion probabilistic model for text-to-speech," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

[44] N. Chen, Y. Zhang, H. Zen, R. J. Weiss, M. Norouzi, and W. Chan, "Wavegrad: Estimating gradients for waveform generation," in *International Conference on Learning Representations*, 2020.

[45] Z. Kong, W. Ping, J. Huang, K. Zhao, and B. Catanzaro, "Diffwave: A versatile diffusion model for audio synthesis," in *International Conference on Learning Representations*, 2020.

[46] S.-g. Lee, H. Kim, C. Shin, X. Tan, C. Liu, Q. Meng, T. Qin, W. Chen, S. Yoon, and T.-Y. Liu, "Priorgrad: Improving conditional denoising diffusion models with data-dependent adaptive prior," in *International Conference on Learning Representations*, 2022.

[47] R. Huang, M. Lam, J. Wang, D. Su, D. Yu, Y. Ren, and Z. Zhao, "Fastdiff: A fast conditional diffusion model for high-quality speech synthesis," in *IJCAI International Joint Conference on Artificial Intelligence*. IJCAI: International Joint Conferences on Artificial Intelligence Organization, 2022, pp. 4157–4163.

[48] Z. Ju, Y. Wang, K. Shen, X. Tan, D. Xin, D. Yang, Y. Liu, Y. Leng, K. Song, S. Tang *et al.*, "Naturalspeech 3: Zero-shot speech synthesis with factorized codec and diffusion models," *arXiv preprint arXiv:2403.03100*, 2024.

[49] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[50] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier, "Language modeling with gated convolutional networks," in *International conference on machine learning*, 2017.

[51] K. Ito, "The lj speech dataset," 2017, https://keithito.com/LJ-Speech-Dataset/.

[52] H. Zen, V. Dang, R. Clark, Y. Zhang, R. J. Weiss, Y. Jia, Z. Chen, and Y. Wu, "Libritts: A corpus derived from librispeech for text-to-speech," *Interspeech 2019*, 2019.

[53] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, 2015.

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

[55] I.-T. Recommendation, "Perceptual evaluation of speech quality (pesq): An objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs," *Rec. ITU-T P. 862*, 2001.

[56] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.

[57] R. Kubichek, "Mel-cepstral distance measure for objective speech quality assessment," in *Proceedings of IEEE pacific rim conference on communications computers and signal processing*, vol. 1. IEEE, 1993, pp. 125–128.

[58] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations*, 2015.

[59] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *International Conference on Learning Representations*, 2018.